VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

Department of Software and Systems Engineering

September 2022

HOTEL RESERVATION CHATBOT

AN INDUSTRIAL INTERNSHIP REPORT

Submitted in partial fulfilment for the

award of the degree of

MTech

In

Software Engineering

By

Pothuluri Sivani (19MIS0284)

TABLE OF CONTENTS

ABSTRACT

Engaging in hotel room reservation has remained a hassle even though there are few applications which provide their own services. This project aims at building a voice chatbot device that can be utilized for hotel reservation using Amazon Lex and Amazon Lambda services with the hotel website as the communication platform used. The chatbot is built by leveraging on Amazon Web Services (AWS) in the form of a service called Amazon Lex for configuring the bot with utterances and responses and Lambda Functions to validate the responses while carrying out the operations. The Lambda function runs a script that collects input in the form of plain text or by using voice recognition using the microphone connected to it, which is sent to the Amazon Lex to be processed using various services provided by Amazon Web Services. Then the chatbot sends back a suitable response for the user via the speaker which is connected to the device or it can also send the response in the form of plain text.

Sequence of actions or responses from the users are required to complete its primary tasks when it comes to complex tasks. The simplest type of chatbot is a scripted chatbot. Discussions with such chatbots can only take predetermined sequences. At each level or progression of the conversation, the user should select from the limited choices or options provided by the chatbot to reach the next level of the conversation.

LIST OF ABBREVIATIONS

ACRONYM	EXPANSION
AWS	Amazon Web Services
NLU	Natural Language Understanding
ASR	Automatic Speech Recognition
EC2	Elastic compute Cloud
OS	Operating System
DFD	Data Flow Diagram

LIST OF FIGURES

1. Introduction

In this fast generation, we need everything or any work to be done as fast as possible. So, we need good encrypted and well secured website to perform data transaction. So, I want to create a chatbot that Powered by the same deep learning technologies as Amazon Alexa, Amazon Lex is a service for building conversational interfaces into any application that uses voice and text. In this project I will show you How Hotel booking bot will answer the questions given by the users. When user types the question like hotel details, room details and when submits that then bot will give the answers accordingly. Here, the solution is scalable, using Serverless Computing technologies and designed to allow secure, anonymous access to Hotel Booking chatbot to solve the queries of the users.

Amazon Lex service is an area of Amazon Web Services (AWS) that focuses on the activities performed by smart machines, that is, they mimic human activities and reactions. Some of the activities performed through lex service include speech recognition, learning, planning and problem-solving. A chatbot is a smart device, which is designed with the specific purpose of meeting its user’s requests. Chatbots understand exactly what the request entails by inferring different information, such as focusing on certain keywords, from the conversation between the bot and each individual user. It requires planning and completion of a task before the chatbot can move onto its next task. The first advantage is that the chatbot is available 24/7 to offer service. Another advantage is that the customer waiting time is eliminated, so that customers can receive service at their own convenience.

In this project, I create an intent in Amazon Lex for Hotel Booking system. And then I have created utterances and slot types to have information for the chatbot. This project aims at building a chatbot device that can be used for hotel reservation using Amazon Lex Service. The chatbot is built by leveraging on Amazon Web Services (AWS) in the form of a service called Amazon Lex for configuring the bot with utterances and responses and Lambda Functions to validate the responses.

1.1. PROBLEM STATEMENT

As there are lot of online applications, every user has lot of queries like how to access the website, how can we book the rooms or how will be the services or what are the room types available or where will be the hotel location, etc. To know the basic information of the organizations, in websites itself they will provide customer service contacts. But they will not be available 24/7 right. So, every web application should have chatbots for basic information. So, Chatbots for invented for this usage of online applications. Here chatbots can be created using Artificial Intelligence, Aws Services like Amazon Lex and Lambda Service Function, etc. Hence, in this project, I would like to use AWS services to prepare a chatbot.

1.2. BACKGROUND

Traditional applications of information technology have become obsolete and have been substituted by new technologies such as robotics and AI, which have the capacity to influence economy and society. Chatbots were originally designed as means of entertainment. A chatbot is a program which uses typed conversation and aims at making users think that they were communicating with another individual. Chatbots are computer programs developed using AIML scripts that communicate with users using natural language and engages in a conversation with users by generating natural language as output. Speech interaction has a major role in humanising machineries in future (Yang, 2004). Speech is viewed as the most important facet of communication with regards to humans. Speech carries a lot of information and this is utilised for AI. One manner of extracting information from speech is conversion of speech to text by means of **Automatic Speech Recognition (ASR)** and excavating speech information. Chatbots can answer with appropriate messages, provide recommendations and customers can use them for shopping purposes by a product carousel, all of which are located in the messenger user interface. A chatbot can recognise the buyer’s intent and refine offerings based on the buyer’s choices and preferences. It can then facilitate the sale, order and delivery process. Chatbots use the messenger infrastructure and the usage of apps is not a necessity. The usage of AI as an agent of conversation is more cost-effective than human-aided support. Many studies have investigated the acceptance levels of customers towards AI, particularly in developed economies. According to Gaines-Ross (2016), there was great variation in the understanding of AI among consumers, particularly high among Chinese consumers, and there was a good level of exposure to AI, particularly through social media. Further, most consumers were willing to trust AI in their daily lives, for example, in handling medication reminders, travel directions, entertainment, targeted news, manual labour, and mechanics, and most consumers perceived AI's impact on society to be positive. Several articles are reports have discussed the benefits and challenges of chatbots in the hospitality industry. According to the State of Chatbots Report, 6 the three top benefits of chatbots were uninterrupted 24/7 service, instant responses and quick answers to simple questions; however, the potential blockers for chatbots were preference for real-life assistant and fear of it making a mistake. The chatbots have to be designed to interact in a conversational manner with consumers, often playing the role of a digital content tour guide, directing customers towards the content they need; this in turn drives engagement and satisfaction among customers. Chatbots can also be instructed to do work on behalf of customers e.g., some chatbots can set up a website for a customer in less than two minutes. The challenge still is that the vast majority of technical content is not ready for conversational delivery. Campbell (2017)8 reported that Holiday Inn was the first major hotel chain in Japan to adopt the latest AI chatbot concierge, Bebot, following hospitality giants such as Hilton and the Cosmopolitan Las Vegas. Chatbots can enhance visitor engagement, delivering personalised offers or options instantly and directly to the guest via chat at any point before, during or after their stay, ultimately improving customer experience and loyalty. According to World Hotels, chatbots was one of the most significant trends in hospitality in 2017, and are expected to be the future of marketing and customer support. The use of chatbots in the hospitality industry is still in its nascent stage, encompassing a wide range of services, from hotel bookings and customer service inquiries to pre/post-stay inquiries and general travel advice. hospitality industry has been one of the biggest adopters of chatbots, for a variety of operations, including hotel booking to flight schedules. Chatbots support the hospitality processes through enabling customer and employee engagement, removing language barriers, building trust through personalised service, and increasing reach and profitability of hotels.

1.3. MOTIVATION

Alexa for Hospitality is Amazon's new product, and it helps the hotels, vacation rentals and other hospitality locations, which provides voice first experience to the consumers.

Amazon Echo smart speakers will be placed in the hotel rooms where guests will be able to talk to Alexa for accessing hotel information, requests for room service, housekeeping, control lights and thermostats, play music and more. This solution is rolled out across Marriott Hotels, Westin Hotels & Resorts, St. Regis Hotels & Resorts, Aloft Hotels, and Autograph Collection Hotels.

Gia for Goibibo and Myra for MakeMyTrip are the chat bots using to solve multiple customer interactions and product problems. MakeMyTrip with the integration of business processes with WhatsApp messenger platform offering more convenience to the customers like Web CheckIn, Seat Selection, Reviews & Ratings, Food Selection and many more with a few taps.

State Bank of India (SBI) has launched ILA an interactive live assistant to address the customer queries about SBI Credit card features, benefits, services and more through chat.

Customers drop off due to long lead forms, incomplete information, and lack of an immediate touchpoint. To avoid this various business services are using chatbots. The motivation for this project is to create a chatbot which helps the regular hotel booking websites to have a chatbot for daily activities like booking a hotel room, room service, food etc. This helps to reduce the manual work done by workers daily and to increase the work effectiveness of the hotel.

1.4. PROPOSED SYSTEM

This project simulates the interaction between hotel customers of any hotel integrated with the bot with a voice-based chatbot. This simulation front-end will include a simulated front-end chatbot environment that includes a text chatbot and a separate voice enabled chatbot.

The backend will include question and answer content, natural language processing, ability to find answers to questions, and user authentication to enable the access.

The users should give his concern to book a hotel, and the bot will present the user with various number of questions asking the nights the user will be staying, the food the user prefers and the type of room the user wants to book. And the details will be confirmed and will be updated in the backend. This chatbot will be built using Amazon lex and Amazon Lambda framework.

This framework allows users to create bots that can easily process voice or text input interchangeably. Integration with the digital personal assistant Alexa is easy for a bot built with Lex. It is also simpler to use than other leading bot frameworks and developers can enjoy the benefits that come with the Amazon platform, such as low hosting costs, syncing, etc

1.5. ADVANTAGES OF THE PROPOSED SYSTEM

- **Simplicity:** It offers an easy-to-use console to create your own chatbot in minutes & predefined bots to help you get started.
- **Inbuilt Technologies:** You supply just a few example phrases, and Amazon Lex builds a complete natural language model through which the bot can interact using voice and text.
- **Seamless deployment and scaling:** As the user engagement increases, you don’t need to worry about provisioning hardware and managing infrastructure to power your bot experience.
- **Built-in integration with AWS:** Amazon Lex allows integrating with many other services on the AWS platform including AWS Lambda, Amazon CloudWatch, Amazon Cognito, and Amazon DynamoDB & many others.
- **Cost-Effective:** With Amazon Lex, there are no upfront costs or minimum fees. You will have to pay only for the text or speech requests that you make
- Promoting effective offers – Chatbots can offer products and services based on data of previous market preferences
- Reduction of contact centers – Contact Centers can be dramatically reduced as many of their functions, if not all, are available to chatbots
- Automatic Insights Capture – Once the users communicate with a company’s bot, the company can automatically get their full name, language preference, profile photo, gender, and other personal insights

- Rich Qualitive Data – As potential customers interact with chatbots, creating qualitive data can be used by the company to create better customer offers and solutions. The chatbot can gather necessary information such as users' interests during the dialog with users. Companies get to know their customers and their preferences in a new way
- Customer Satisfaction – Customers interact with bots whenever they prefer. This is quite important because the user can be helped regardless of their time zone. Also, it can solve their doubts at the moment and wait no more for an operator to be available. A further advantage is one-to-one communication. Users no longer have to search the website to find right information like services, products or prices. In case of customers inquiries or complaints, chatbots are helpful, supportive, and capable

2. TECHNOLOGIES LEARNT

I have learnt the technologies which are Amazon Lex, AWS Lambda and lambda function, Python for programming language. The technologies are briefly explained below –

2.1. AMAZON LEX

Amazon Lex V2 is an AWS service for building conversational interfaces for applications using voice and text. Amazon Lex V2 provides the deep functionality and flexibility of natural language understanding (NLU) and automatic speech recognition (ASR) so you can build highly engaging user experiences with lifelike, conversational interactions, and create new categories of products.

Amazon Lex V2 enables any developer to build conversational bots quickly. With Amazon Lex V2, no deep learning expertise is necessary—to create a bot, you specify the basic conversation flow in the Amazon Lex V2 console. Amazon Lex V2 manages the dialog and dynamically adjusts the responses in the conversation. Using the console, you can build, test, and publish your text or voice chatbot. You can then add the conversational interfaces to bots on mobile devices, web applications, and chat platforms (for example, Facebook Messenger).

Amazon Lex V2 provides integration with AWS Lambda, and you can integrate with many other services on the AWS platform, including Amazon Connect, Amazon Comprehend, and Amazon Kendra. Integration with Lambda provides bots access to pre-built serverless enterprise connectors to link to data in SaaS applications such as Salesforce.

Amazon Lex service support both voice and text and can be deployed across mobile and messaging platforms. Amazon Lex enables any developer to build conversational chatbots quickly. With Amazon Lex, no deep learning expertise is necessary—to create a bot, users can just specify the basic conversation flow in the Amazon Lex console. Amazon Lex manages the dialogue and dynamically adjusts the responses in the conversation. Using the console, you can build, test, and publish your text or voice chatbot.

2.2. AMAZON LAMBDA

Lambda is a compute service that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging. With Lambda, you can run code for virtually any type of application or backend service. All you need to do is supply your code in one of the languages that Lambda supports. You organize your code into Lambda functions. Lambda runs your function only when needed and scales automatically, from a few requests per day to thousands per second. You pay only for the compute time that you consume—there is no charge when your code is not running.

Lambda is a serverless compute service that runs your code in response to events and automatically manages the underlying compute resources for you. These events may include changes in state or an update, such as a user placing an item in a shopping cart on an ecommerce website. You can use AWS Lambda to extend other AWS services with custom logic, or create your own backend services that operate at AWS scale, performance, and security

You can invoke your Lambda functions using the Lambda API, or Lambda can run your functions in response to events from other AWS services. For example, you can use Lambda to:

- Build data-processing triggers for AWS services such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB.
- Process streaming data stored in Amazon Kinesis.
- Create your own backend that operates at AWS scale, performance, and security.

Lambda is an ideal compute service for many application scenarios, as long as you can run your application code using the Lambda standard runtime environment and within the resources that Lambda provides.

When using Lambda, you are responsible only for your code. Lambda manages the compute fleet that offers a balance of memory, CPU, network, and other resources to run your code. Because Lambda manages these resources, you cannot log in to compute instances or customize the operating system on provided runtimes. Lambda performs operational and administrative activities on your behalf, including managing capacity, monitoring, and logging your Lambda functions.

If you need to manage your own compute resources, AWS has other compute services to meet your needs. For example:

- Amazon Elastic Compute Cloud (Amazon EC2) offers a wide range of EC2 instance types to choose from. It lets you customize operating systems, network and security settings, and the entire software stack. You are responsible for provisioning capacity, monitoring fleet health and performance, and using Availability Zones for fault tolerance.
- AWS Elastic Beanstalk enables you to deploy and scale applications onto Amazon EC2. You retain ownership and full control over the underlying EC2 instances.

Lambda has features like concurrency and scaling controls, Functions defined as container images, Code Signing, Lambda Extensions, Function blueprints, Database access, File systems access.

2.3. AMAZON LAMBDA FUNCTION

WS Lambda function helps developers to focus on core product and business logic instead of managing operating system (OS) access control, OS patching, right-sizing, provisioning, scaling, etc. Users can run Python code in AWS Lambda. Lambda provides runtimes for Python that code to process events. The code runs in an environment that includes the SDK for Python (Boto3), with credentials from an AWS Identity and Access Management role that one can manage.

2.4. PYTHON – PROGRAMMING LANGUAGE

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020. Python consistently ranks as one of the most popular programming languages. It will be used for lambda service function.

2.5. Cloud Watch

Amazon CloudWatch is a monitoring and management service that provides data and actionable insights for AWS, hybrid, and on-premises applications and infrastructure resources. One can collect and access all your performance and operational data in the form of logs and metrics from a single platform rather than monitoring them in silos (server, network, or database). Cloud Watch enables users to monitor the complete stack (applications, infrastructure, and services) and use alarms, logs, and events data to take automated actions and reduce mean time to resolution (MTTR). This frees up important resources and allows you to focus on building applications and business value.

2.6. S3 Bucket

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can use Amazon S3 to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics.

2.7. SNS Service

SNS stands for Simple Notification Service. It is a web service which makes it easy to set up, operate, and send a notification from the cloud. It provides developers with the highly scalable, cost-effective, and flexible capability to publish messages from an application and sends them to other applications.

3. SYSTEM DESIGN

A flowchart is designed that shows the process flow of the entire project.

3.1. SYSTEM ARCHITECTURE

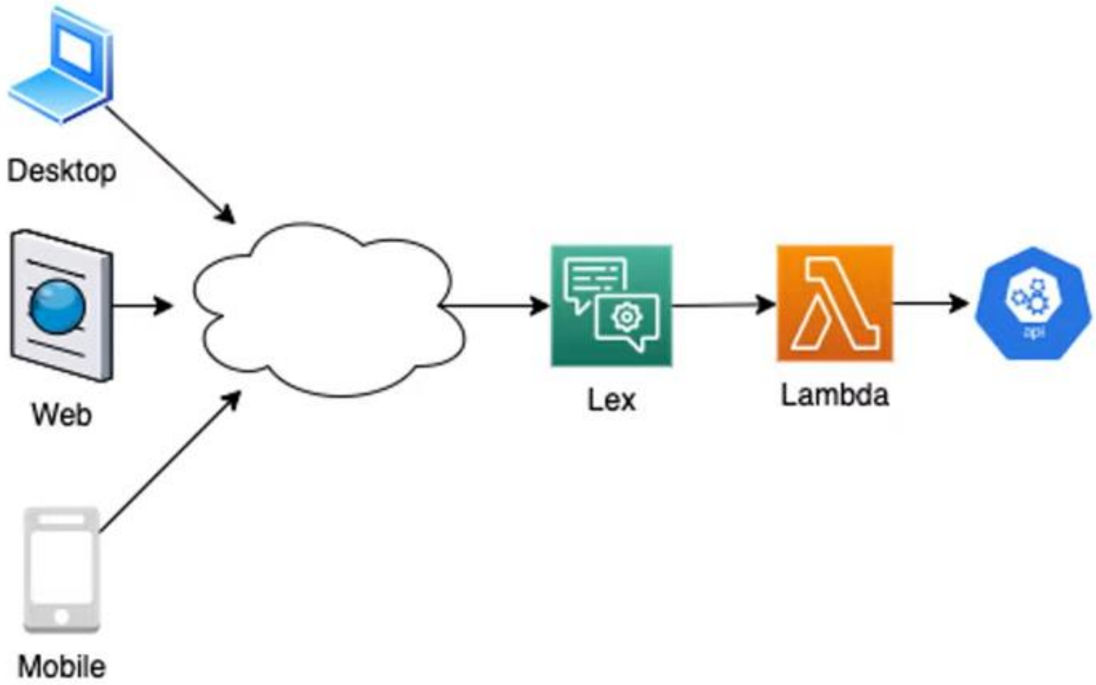


FIG.1

3.2. Module Description

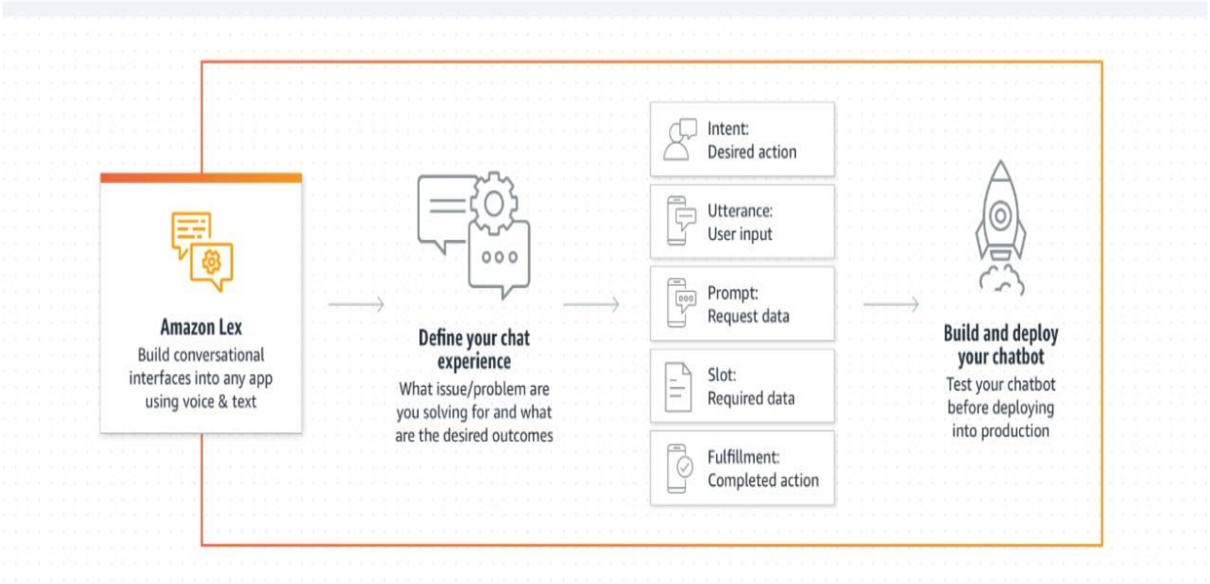

Amazon Lex
Build conversational interfaces into any app using voice & text
Define your chat experience
What issue/problem are you solving for and what are the desired outcomes
Intent:
Desired action
Utterance:
User input
Prompt:
Request data
Slot:
Required data
Fulfillment:
Completed action
Build and deploy your chatbot
Test your chatbot before deploying into production

FIG. 2

3.3. System Requirements

Software Requirements

- AWS Services like Amazon Lex, Amazon Lambda.
- Python Programming Language.
- AWS Console

Hardware Requirements

- Laptop

3.4. Detailed Design

3.4.1. DFD Diagram

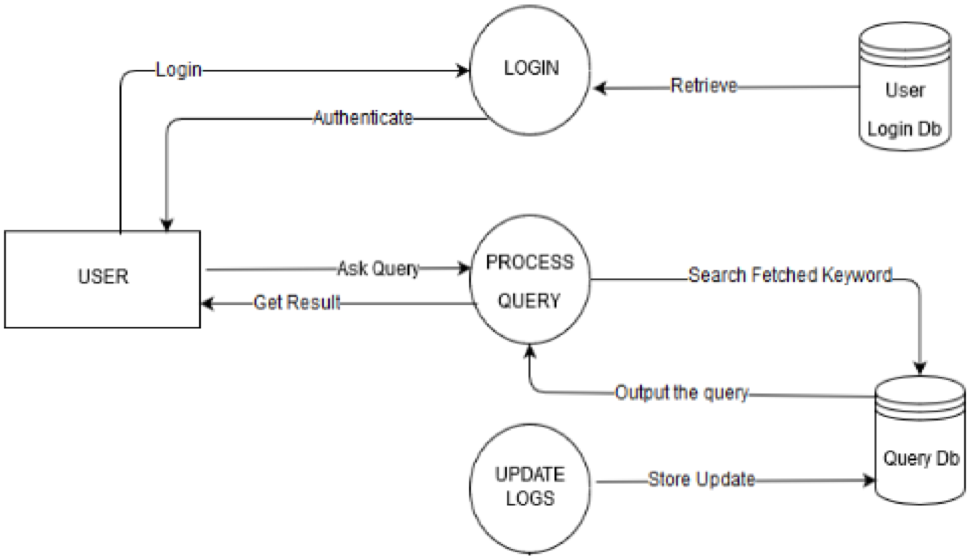


FIG.3

3.4.2 Use Case Diagram

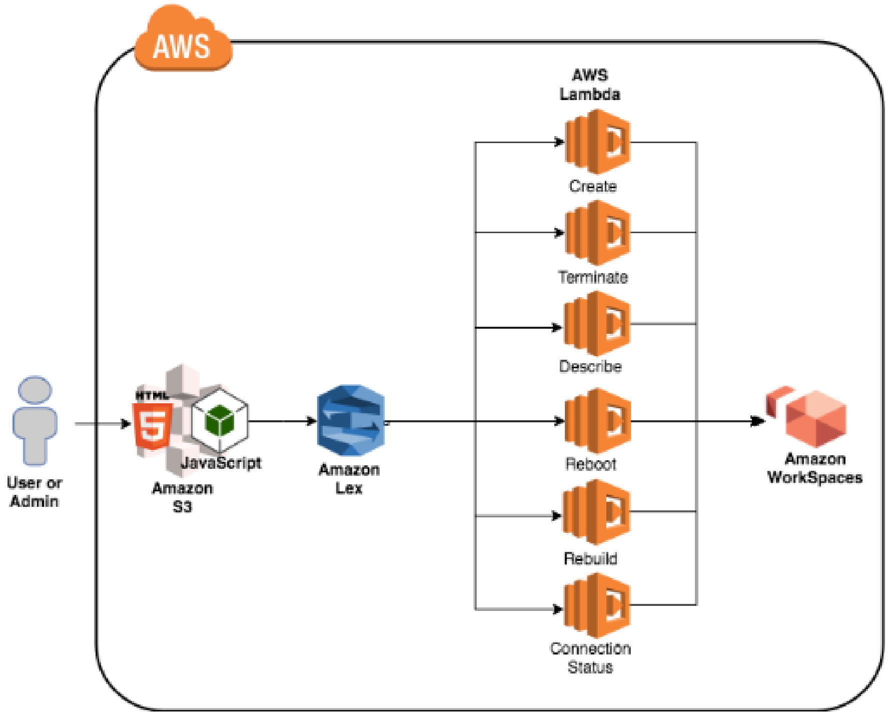


FIG.4

3.4.3 Class Diagram for Hotel Reservation

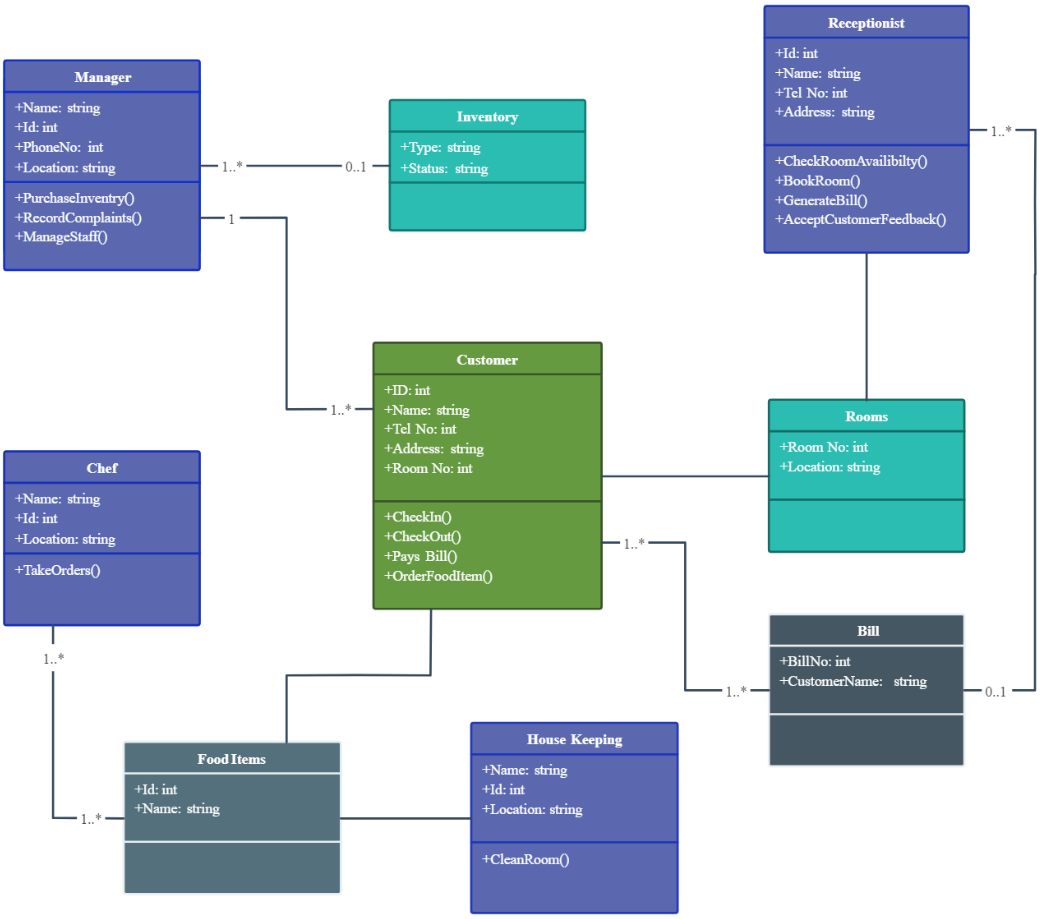
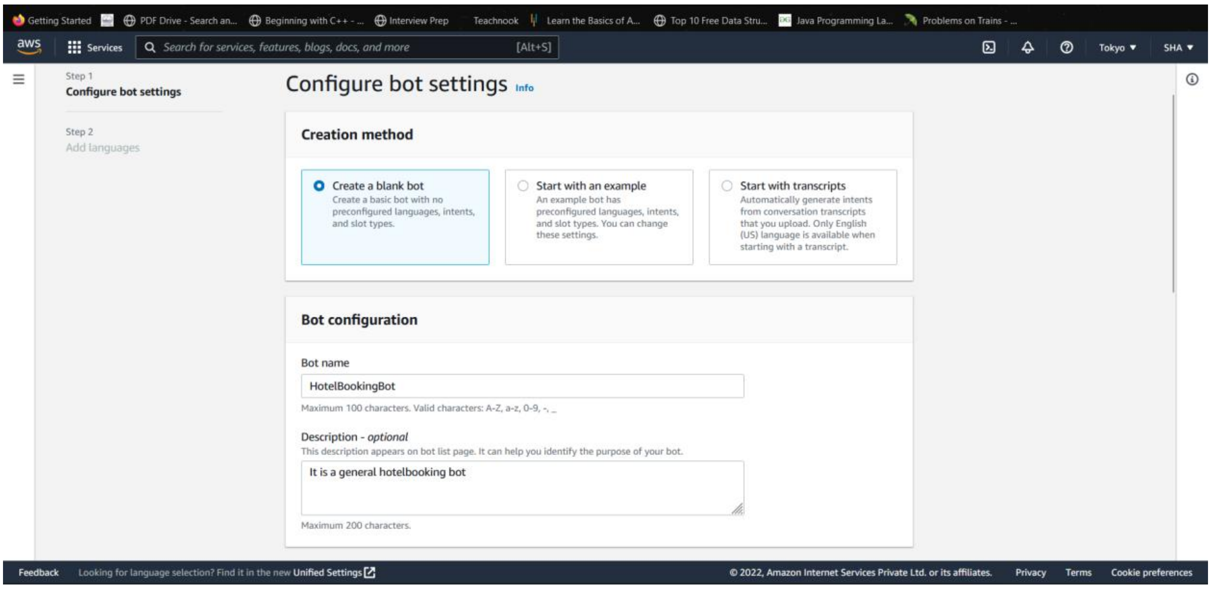


FIG.5

4. IMPLEMENTATION

The building phase of a chatbot can be done using the predefined templates that are already pre-defined by amazon web services (AWS) or can also be developed from scratch. The languages supported in the building of a chatbot are python and NodeJS with various versions that can be selected on the developer's preference. Once the code has been developed then it is tested in the lambda interface.

At first login into the AWS Console, then open the Amazon Lex Service and Click the create Bot option and fill out the fields accordingly.



Step 1
Configure bot settings

Step 2
Add languages

Configure bot settings Info

Creation method

● Create a blank bot
Create a basic bot with no preconfigured languages, intents, and slot types.

○ Start with an example
An example bot has preconfigured languages, intents, and slot types. You can change these settings.

○ Start with transcripts
Automatically generate intents from conversation transcripts that you upload. Only English (US) language is available when starting with a transcript.

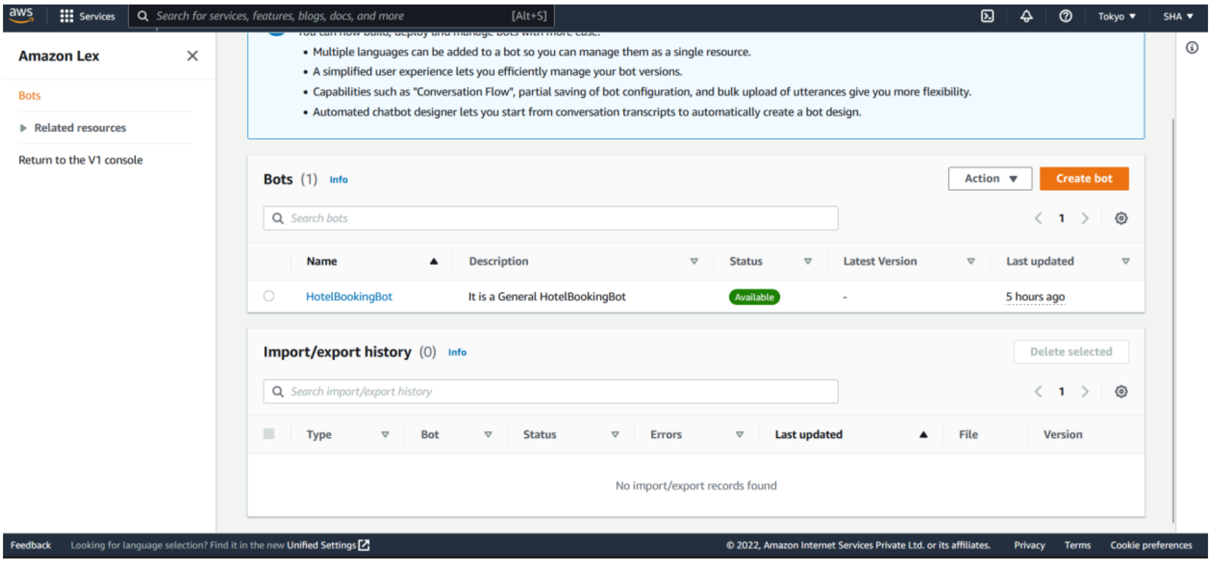
Bot configuration

Bot name
HotelBookingBot
Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, _

Description - *optional*
This description appears on bot list page. It can help you identify the purpose of your bot.
It is a general hotelbooking bot
Maximum 200 characters.

FIG.6

Then it will be visible in Bots History as, if created successfully.



Amazon Lex

Bots
Related resources
Return to the V1 console

- Multiple languages can be added to a bot so you can manage them as a single resource.
- A simplified user experience lets you efficiently manage your bot versions.
- Capabilities such as "Conversation Flow", partial saving of bot configuration, and bulk upload of utterances give you more flexibility.
- Automated chatbot designer lets you start from conversation transcripts to automatically create a bot design.

Bots (1) Info

	Name	Description	Status	Latest Version	Last updated
○	HotelBookingBot	It is a General HotelBookingBot	Available	-	5 hours ago

Import/export history (0) Info

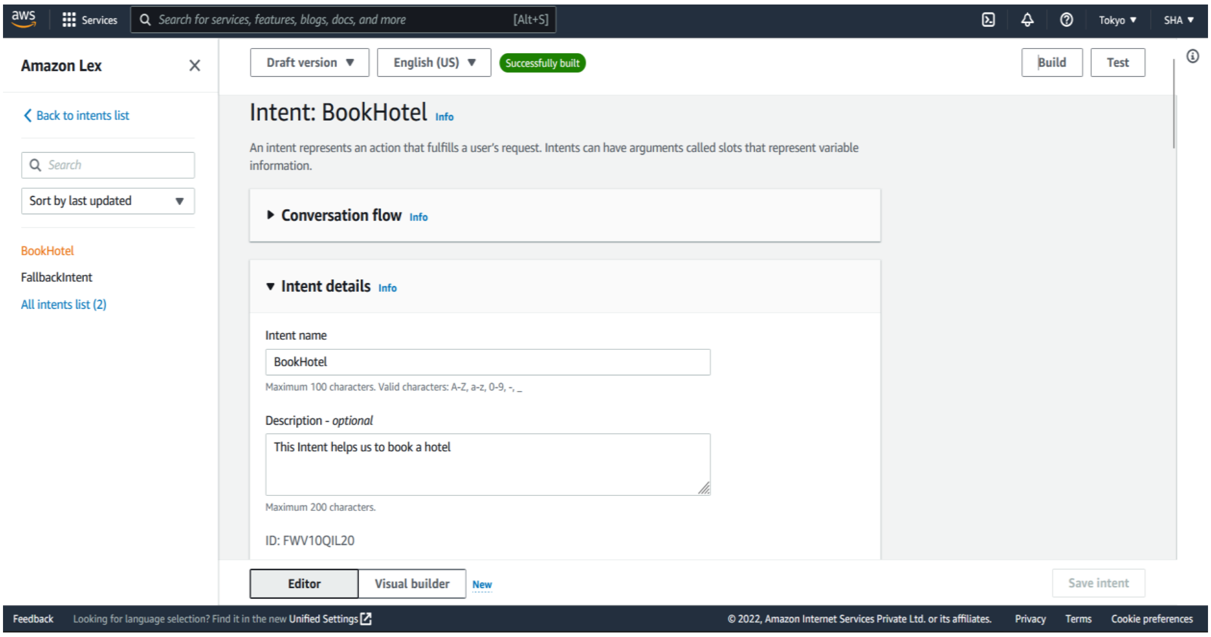
	Type	Bot	Status	Errors	Last updated	File	Version

No import/export records found

aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Tokyo ▾
SHA ▾
Action ▾
Create bot
Search bots
Delete selected
Search import/export history
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences

FIG.7

Then create an intent in the created bot and give name as “Bookhotel” and provide it with description. And then give the information required in the fields. Save the intents after going back to the bot page, without that we will lose the information.



Intent: BookHotel Info

An intent represents an action that fulfills a user's request. Intents can have arguments called slots that represent variable information.

Intent details Info

Intent name: BookHotel

Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, _

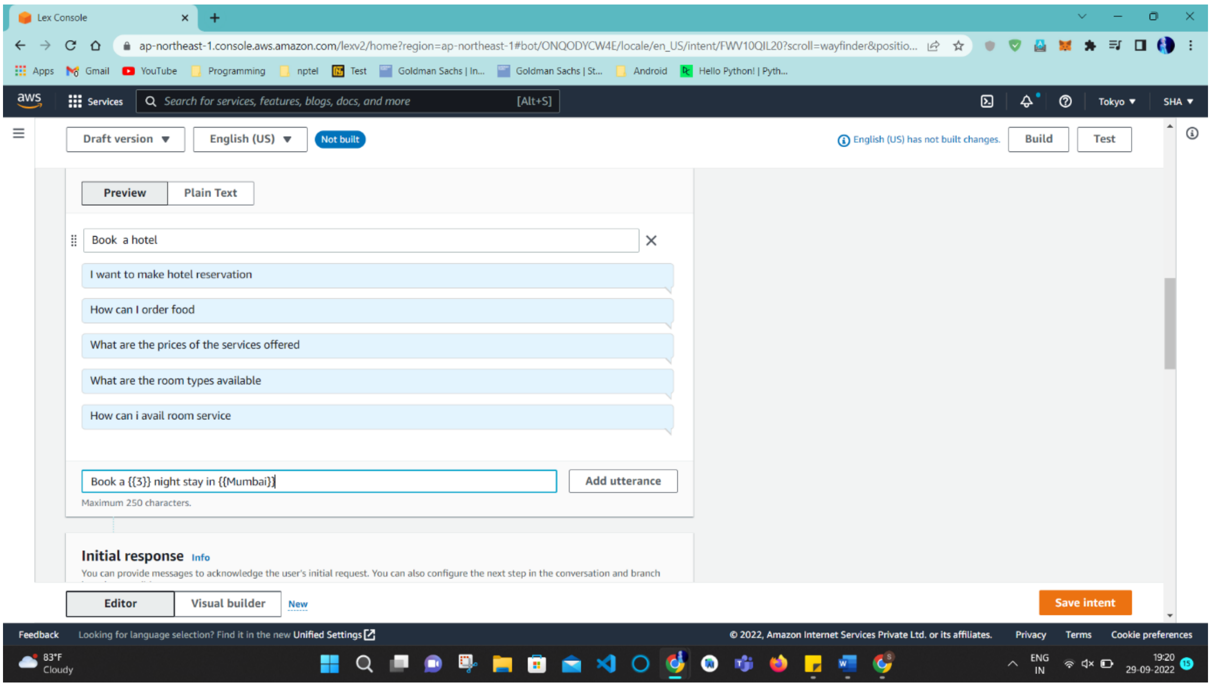
Description - *optional*: This Intent helps us to book a hotel

Maximum 200 characters.

ID: FWV10QIL20

FIG.8

Then create the sample Utterances accordingly -



Lex Console
ap-northeast-1.console.aws.amazon.com/lexv2/home?region=ap-northeast-1#bot/ONQODYCW4E/locale/en_US/intent/FWV10QIL20?scroll=wayfinder&positio...
Apps
Gmail
YouTube
Programming
nptel
Test
Goldman Sachs | In...
Goldman Sachs | St...
Android
Hello Python! | Pyth...
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Tokyo ▼
SHA ▼
Draft version ▼
English (US) ▼
Not built
English (US) has not built changes.
Build
Test
Preview
Plain Text
Add utterance
Editor
Visual builder
New
Save intent
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
83°F
Cloudy
ENG
IN
19:20
29-09-2022

Book a hotel

I want to make hotel reservation

How can I order food

What are the prices of the services offered

What are the room types available

How can i avail room service

Book a {{3}} night stay in {{Mumbai}}

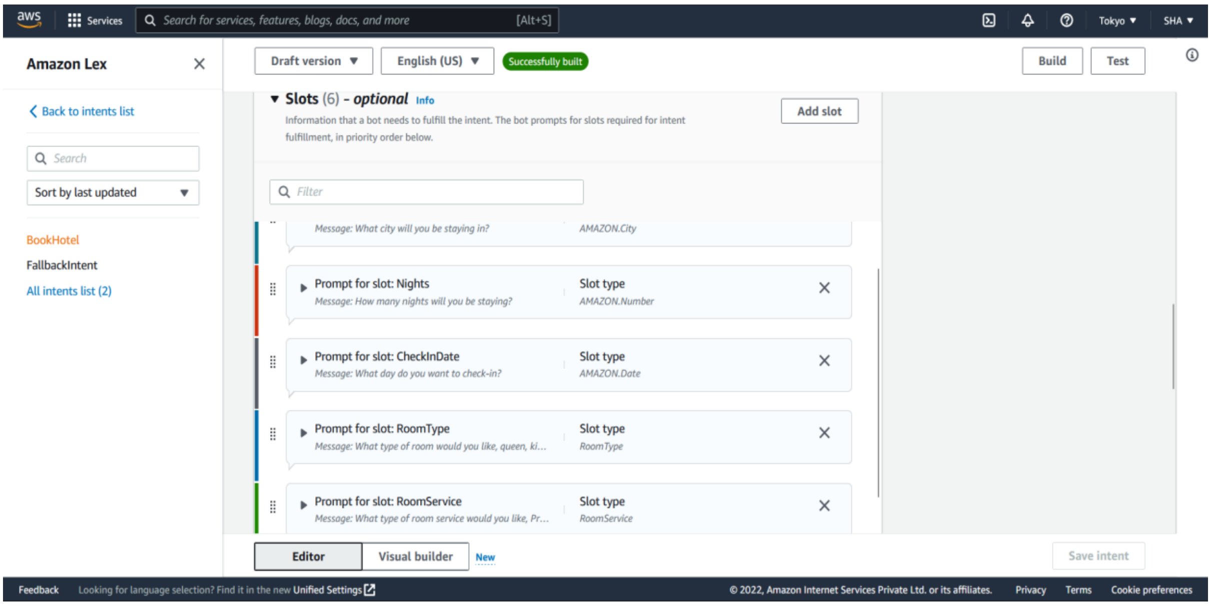
Maximum 250 characters.

Initial response Info

You can provide messages to acknowledge the user's initial request. You can also configure the next step in the conversation and branch

FIG.9

Then create the Slots as follows -



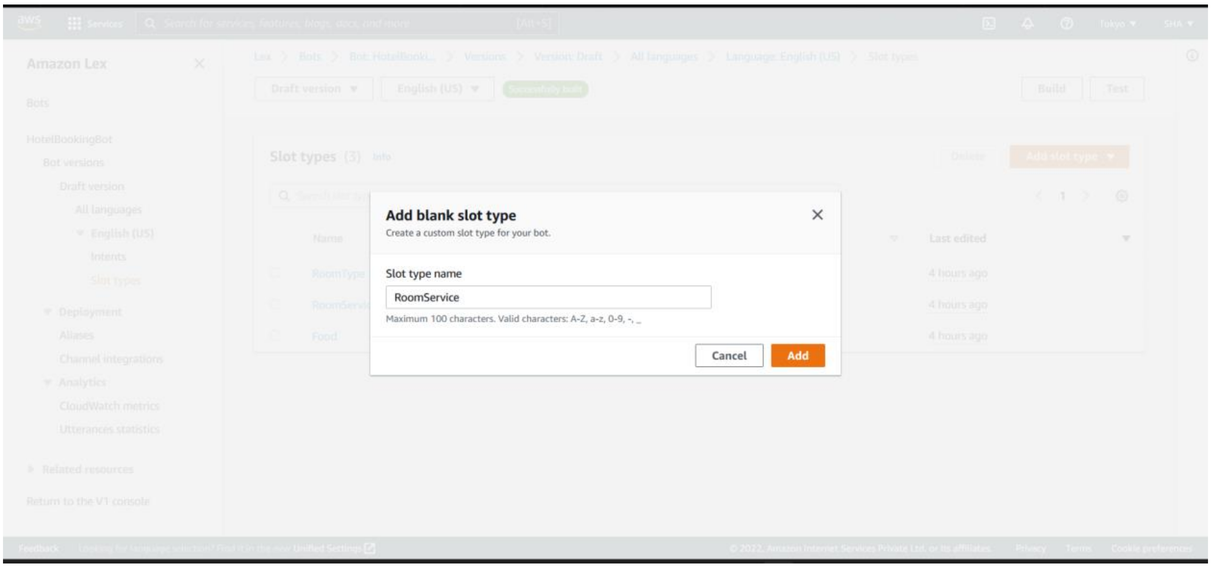
Slots (6) - *optional* Info

Information that a bot needs to fulfill the intent. The bot prompts for slots required for intent fulfillment, in priority order below.

Prompt	Slot type
Message: What city will you be staying in?	AMAZON.City
Prompt for slot: Nights Message: How many nights will you be staying?	Slot type AMAZON.Number
Prompt for slot: CheckInDate Message: What day do you want to check-in?	Slot type AMAZON.Date
Prompt for slot: RoomType Message: What type of room would you like, queen, ki...	Slot type RoomType
Prompt for slot: RoomService Message: What type of room service would you like, Pr...	Slot type RoomService

FIG.10

If we wish to have custom slot then go to intents and go to slots type and then add the slot type and give prompt message which should be visible to the user.



Add blank slot type

Create a custom slot type for your bot.

Slot type name

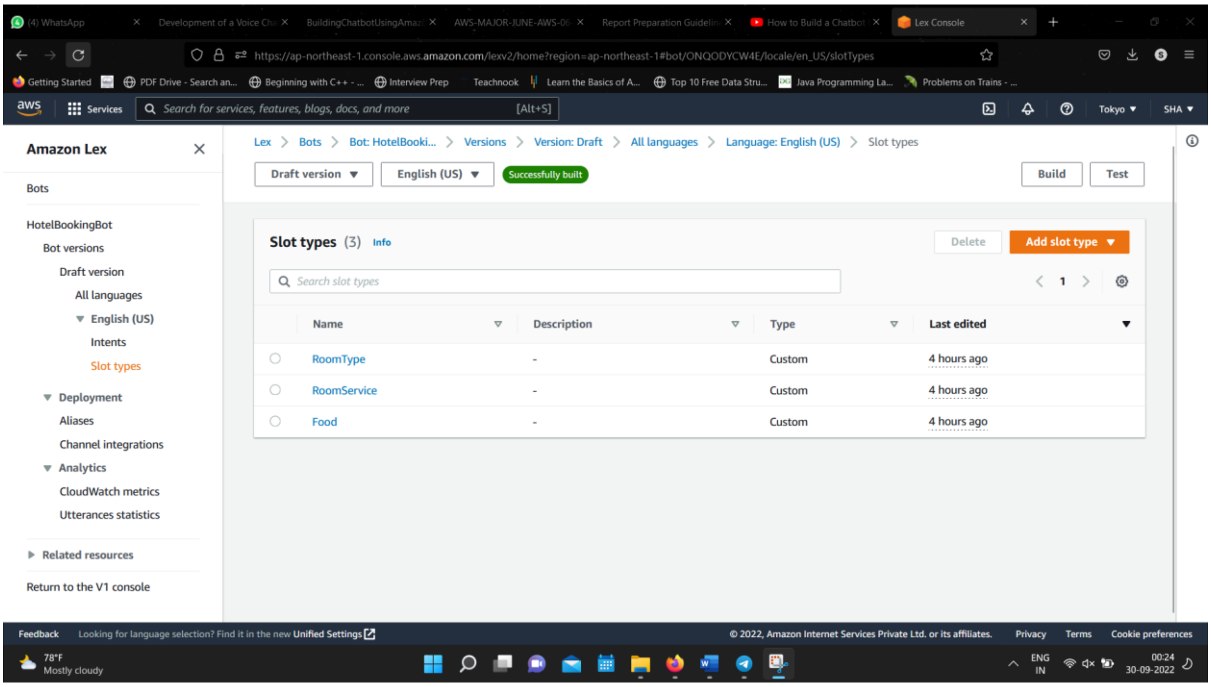
RoomService

Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, _

Cancel Add

FIG.11

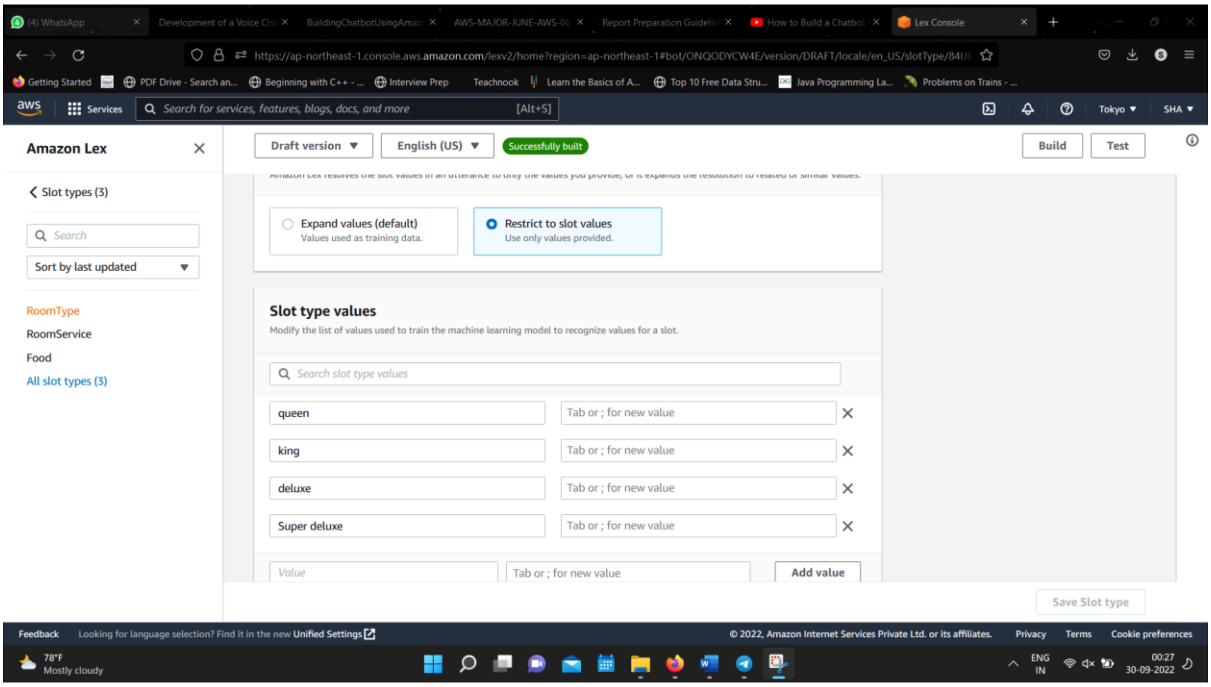
Then it will show the added slots like these in the slot types list -



Slot types (3) Info

	Name	Description	Type	Last edited
○	RoomType	-	Custom	4 hours ago
○	RoomService	-	Custom	4 hours ago
○	Food	-	Custom	4 hours ago

FIG.12



○ Expand values (default)
Values used as training data.

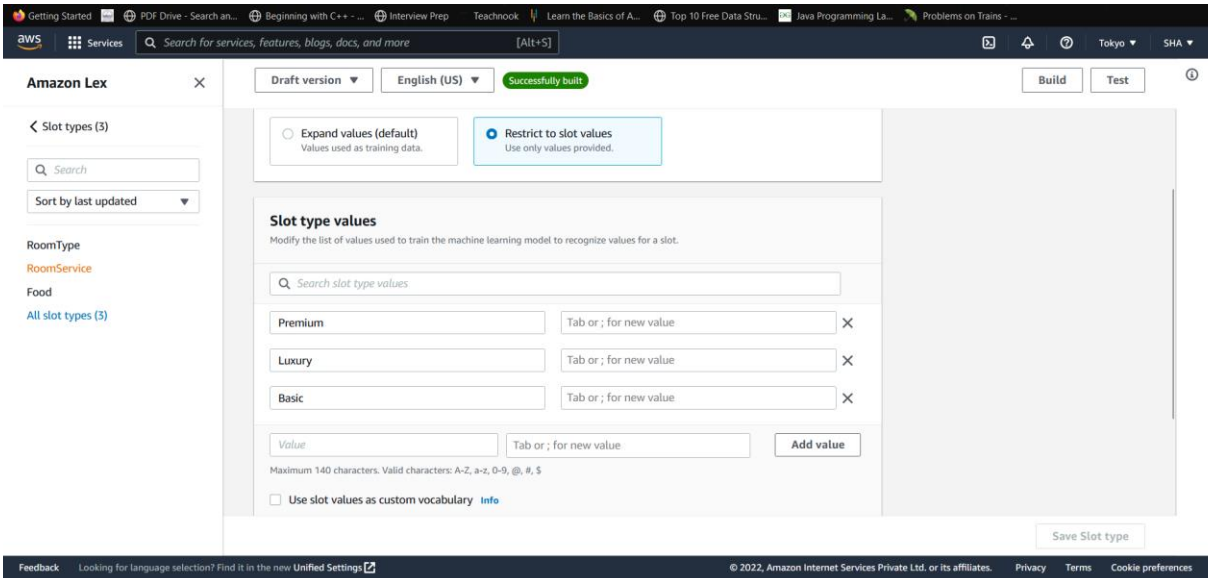
● Restrict to slot values
Use only values provided.

Slot type values

Modify the list of values used to train the machine learning model to recognize values for a slot.

Value	Synonyms
queen	Tab or ; for new value
king	Tab or ; for new value
deluxe	Tab or ; for new value
Super deluxe	Tab or ; for new value

FIG.13



○ Expand values (default)
Values used as training data.

● Restrict to slot values
Use only values provided.

Slot type values

Modify the list of values used to train the machine learning model to recognize values for a slot.

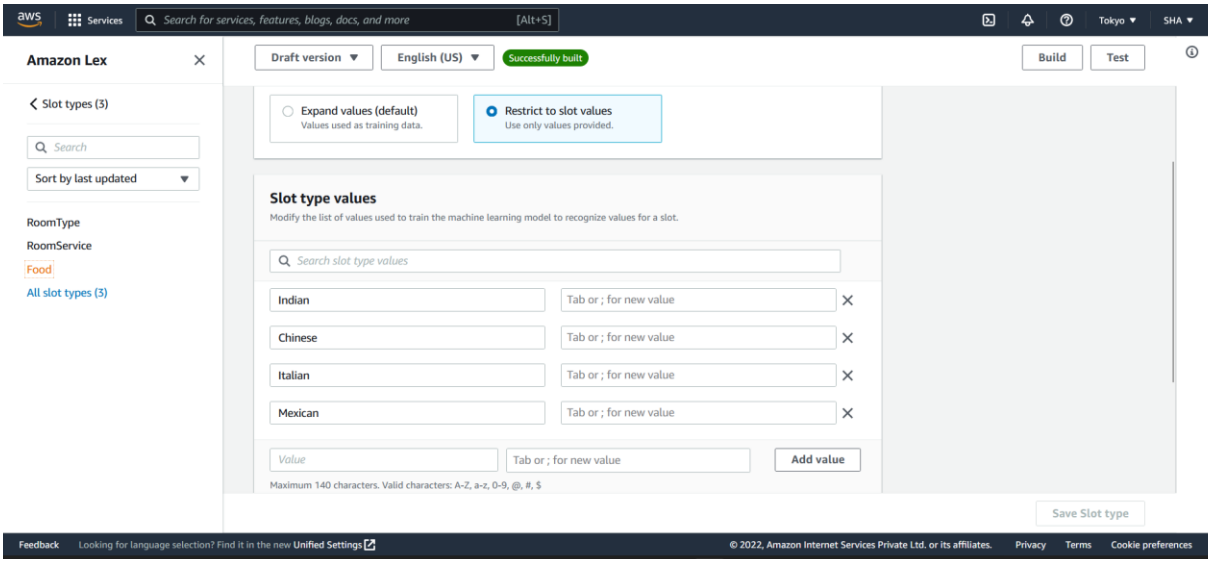
Search slot type values

Premium	Tab or ; for new value	✕
Luxury	Tab or ; for new value	✕
Basic	Tab or ; for new value	✕
Value	Tab or ; for new value	Add value

Maximum 140 characters. Valid characters: A-Z, a-z, 0-9, @, #, $

□ Use slot values as custom vocabulary Info

FIG.14



○ Expand values (default)
Values used as training data.

● Restrict to slot values
Use only values provided.

Slot type values

Modify the list of values used to train the machine learning model to recognize values for a slot.

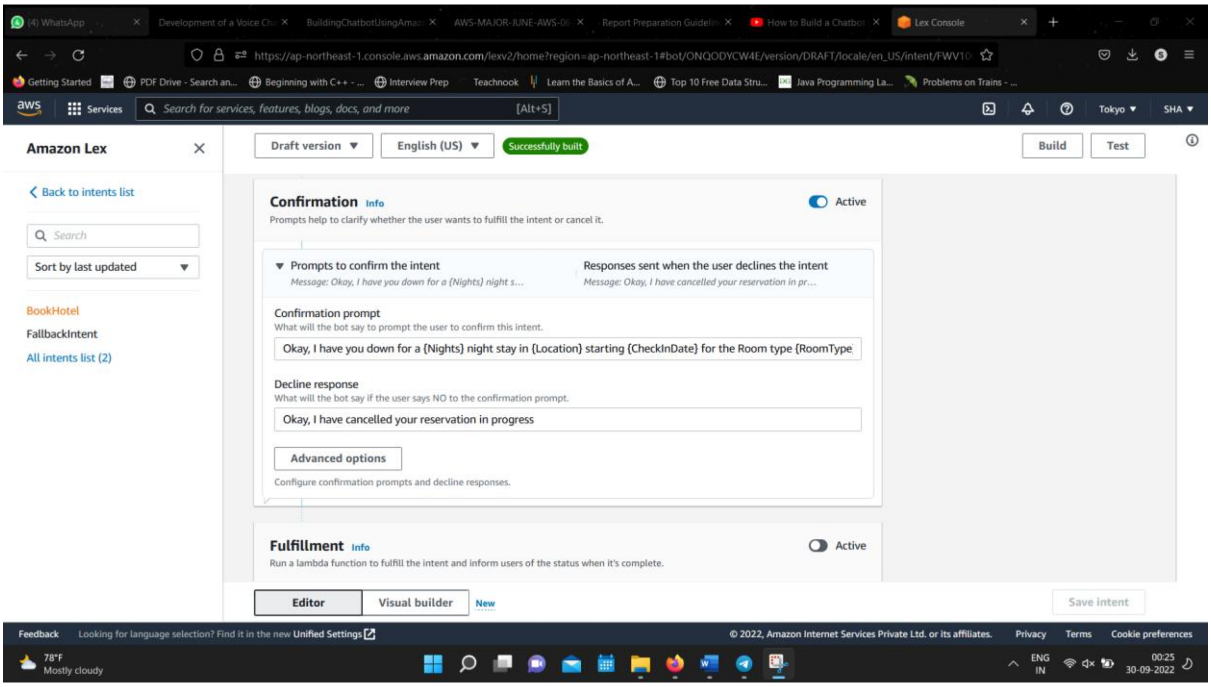
Search slot type values

Indian	Tab or ; for new value	✕
Chinese	Tab or ; for new value	✕
Italian	Tab or ; for new value	✕
Mexican	Tab or ; for new value	✕
Value	Tab or ; for new value	Add value

Maximum 140 characters. Valid characters: A-Z, a-z, 0-9, @, #, $

FIG.15

Then add confirmation messages in confirmation fields like this -



Confirmation Info

Prompts help to clarify whether the user wants to fulfill the intent or cancel it.

Prompts to confirm the intent — Message: Okay, I have you down for a {Nights} night s...

Responses sent when the user declines the intent — Message: Okay, I have cancelled your reservation in pr...

Confirmation prompt

What will the bot say to prompt the user to confirm this intent.

Okay, I have you down for a {Nights} night stay in {Location} starting {CheckInDate} for the Room type {RoomType

Decline response

What will the bot say if the user says NO to the confirmation prompt.

Okay, I have cancelled your reservation in progress

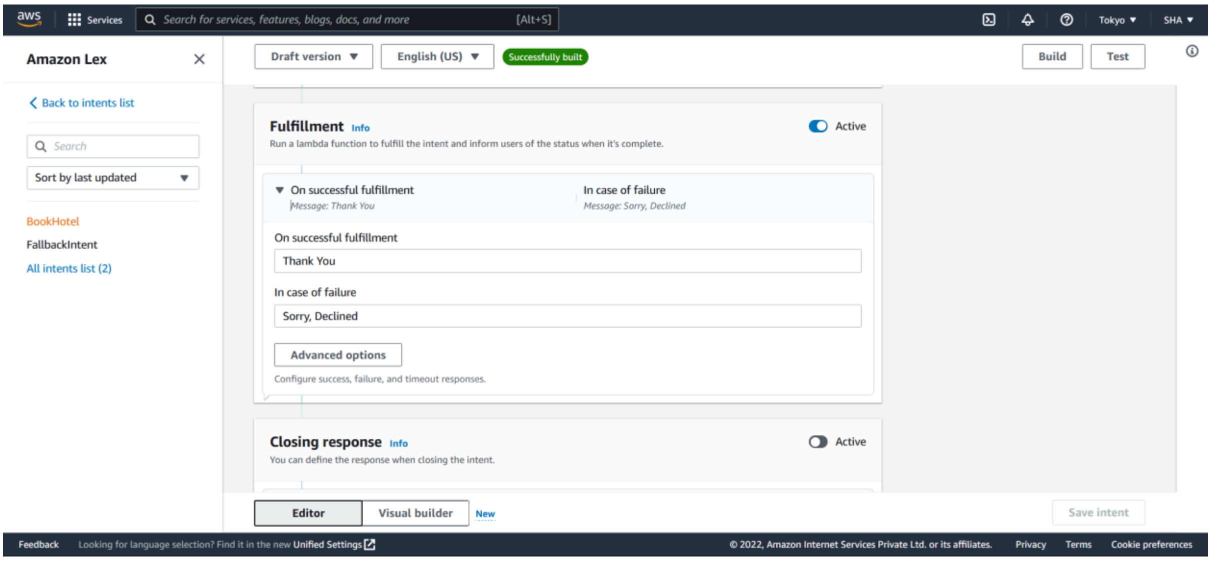
Configure confirmation prompts and decline responses.

Fulfillment Info

Run a lambda function to fulfill the intent and inform users of the status when it's complete.

FIG.16

Now fulfilment details will be given for users as –



Fulfillment Info

Run a lambda function to fulfill the intent and inform users of the status when it's complete.

On successful fulfillment — Message: Thank You

In case of failure — Message: Sorry, Declined

On successful fulfillment

Thank You

In case of failure

Sorry, Declined

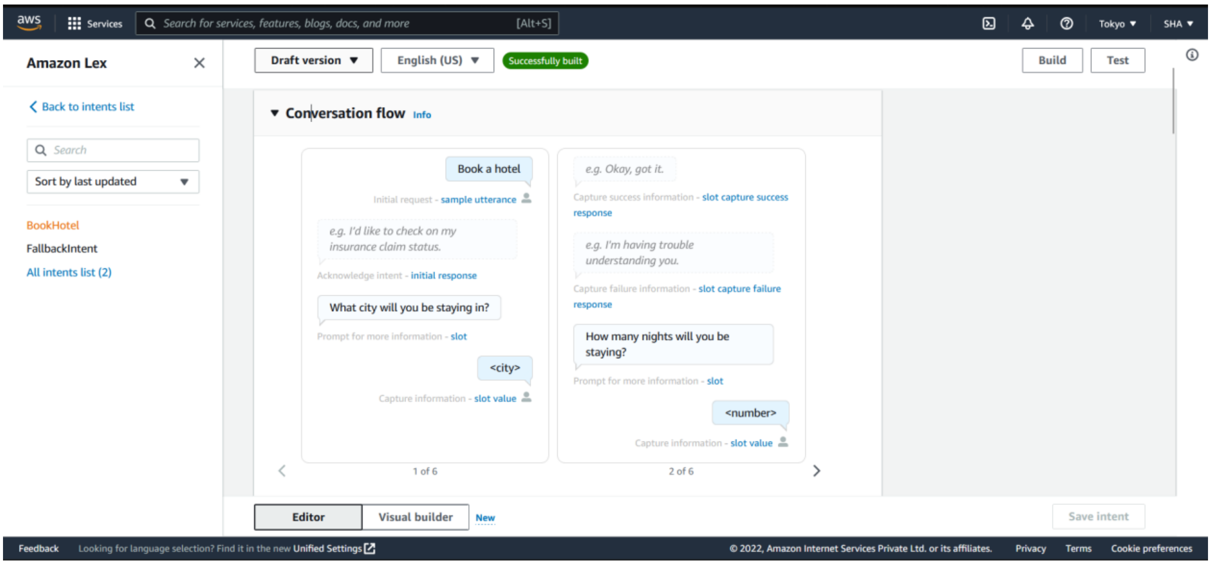
Configure success, failure, and timeout responses.

Closing response Info

You can define the response when closing the intent.

FIG.17

Now I will show the Conversational flow of the chatbot which giving the bot instructions that what should be displayed one by one question as a sequence –



Book a hotel

Initial request - sample utterance

e.g. I'd like to check on my insurance claim status.

Acknowledge intent - initial response

What city will you be staying in?

Prompt for more information - slot

<city>

Capture information - slot value

1 of 6

e.g. Okay, got it.

Capture success information - slot capture success response

e.g. I'm having trouble understanding you.

Capture failure information - slot capture failure response

How many nights will you be staying?

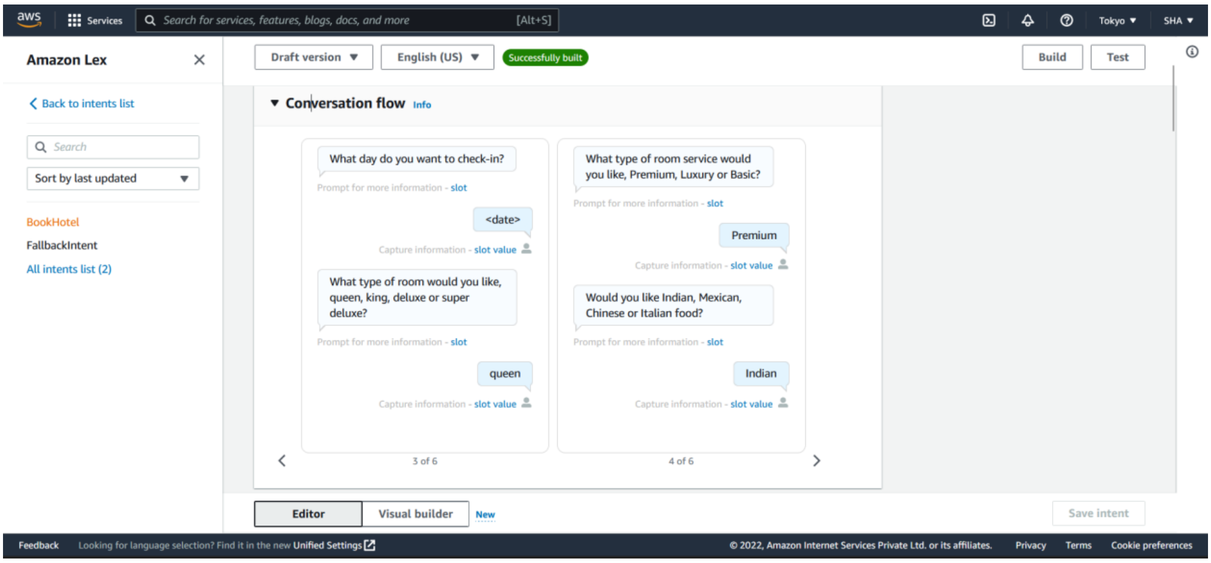
Prompt for more information - slot

<number>

Capture information - slot value

2 of 6

FIG.18



What day do you want to check-in?

Prompt for more information - slot

<date>

Capture information - slot value

What type of room would you like, queen, king, deluxe or super deluxe?

Prompt for more information - slot

queen

Capture information - slot value

3 of 6

What type of room service would you like, Premium, Luxury or Basic?

Prompt for more information - slot

Premium

Capture information - slot value

Would you like Indian, Mexican, Chinese or Italian food?

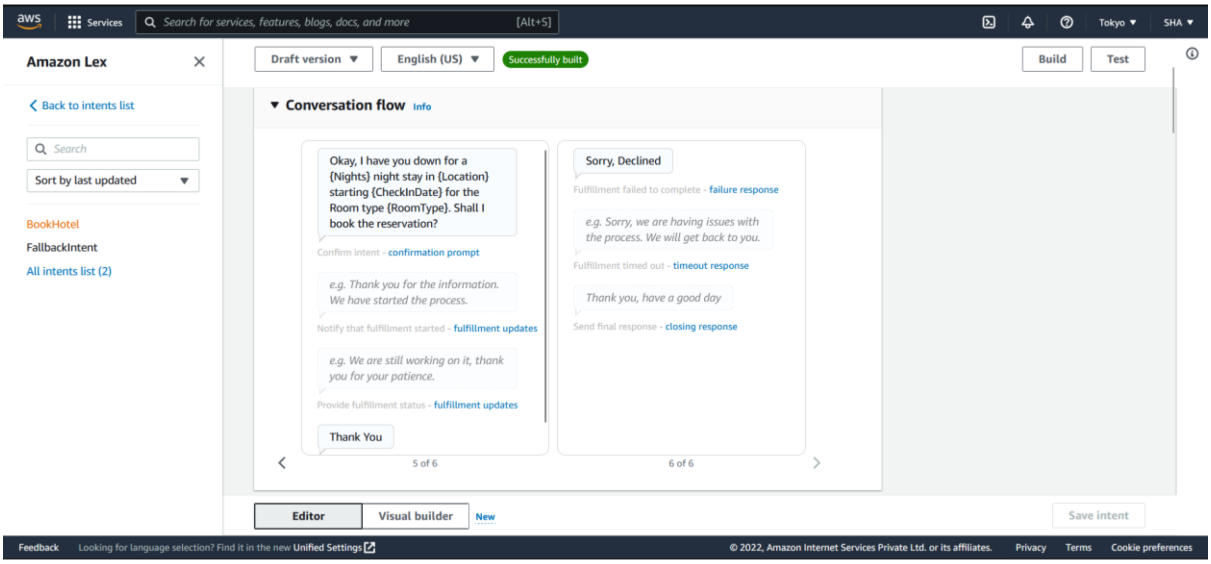
Prompt for more information - slot

Indian

Capture information - slot value

4 of 6

FIG.19



Okay, I have you down for a {Nights} night stay in {Location} starting {CheckInDate} for the Room type {RoomType}. Shall I book the reservation?

Confirm intent - confirmation prompt

e.g. Thank you for the information. We have started the process.

Notify that fulfillment started - fulfillment updates

e.g. We are still working on it, thank you for your patience.

Provide fulfillment status - fulfillment updates

Thank You

Sorry, Declined

Fulfillment failed to complete - failure response

e.g. Sorry, we are having issues with the process. We will get back to you.

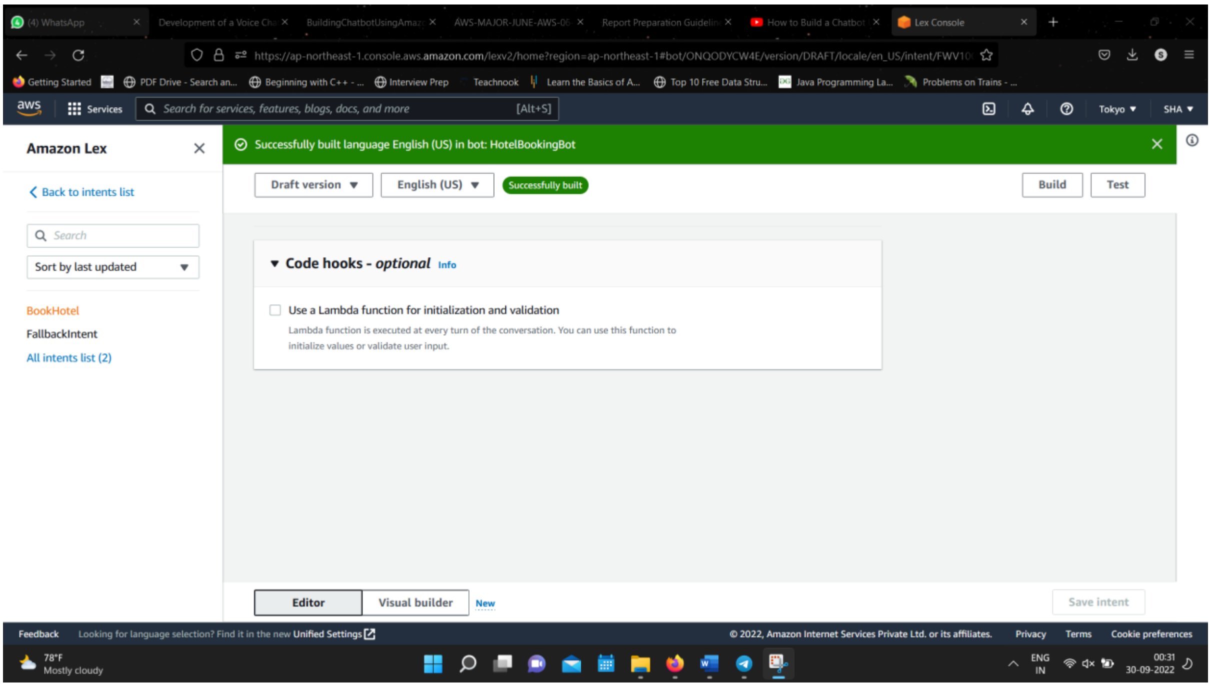
Fulfillment timed out - timeout response

Thank you, have a good day

Send final response - closing response

FIG.20

Now after adding everything as mentioned, Now save the intent finally. Now Build the Intent which will be looked after successful built –



Successfully built language English (US) in bot: HotelBookingBot

Code hooks - *optional* Info

□ Use a Lambda function for initialization and validation

Lambda function is executed at every turn of the conversation. You can use this function to initialize values or validate user input.

FIG.21

Now here I will search for Lambda Function which will be used to serve the validation mechanism or the business logic and to enable the backend to the services that we provide.

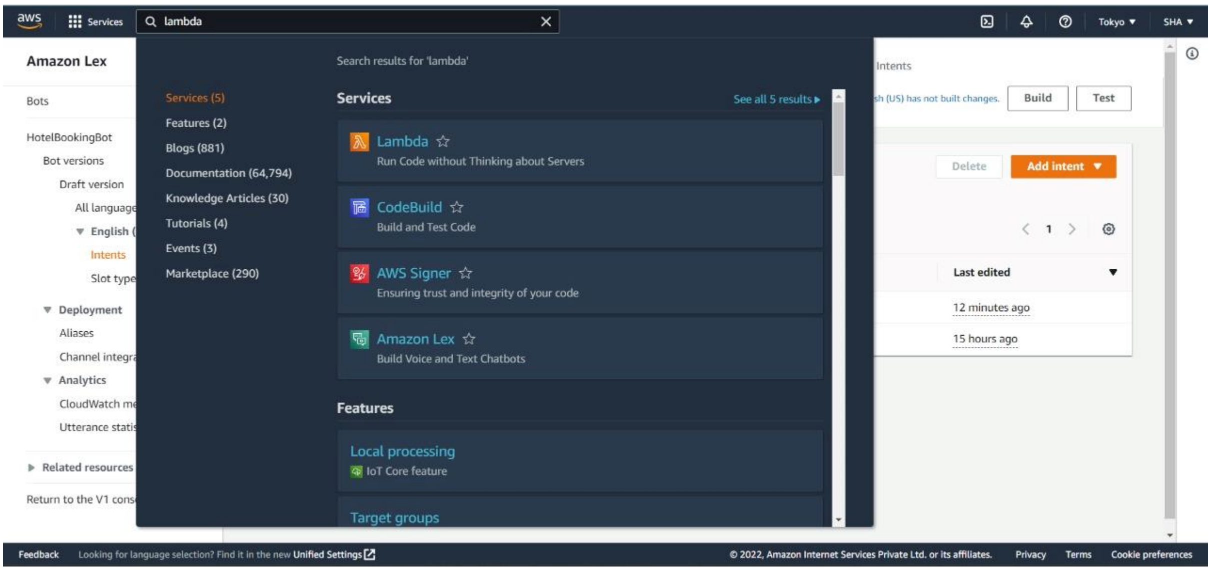


FIG.22

Create function to instruct the chatbot

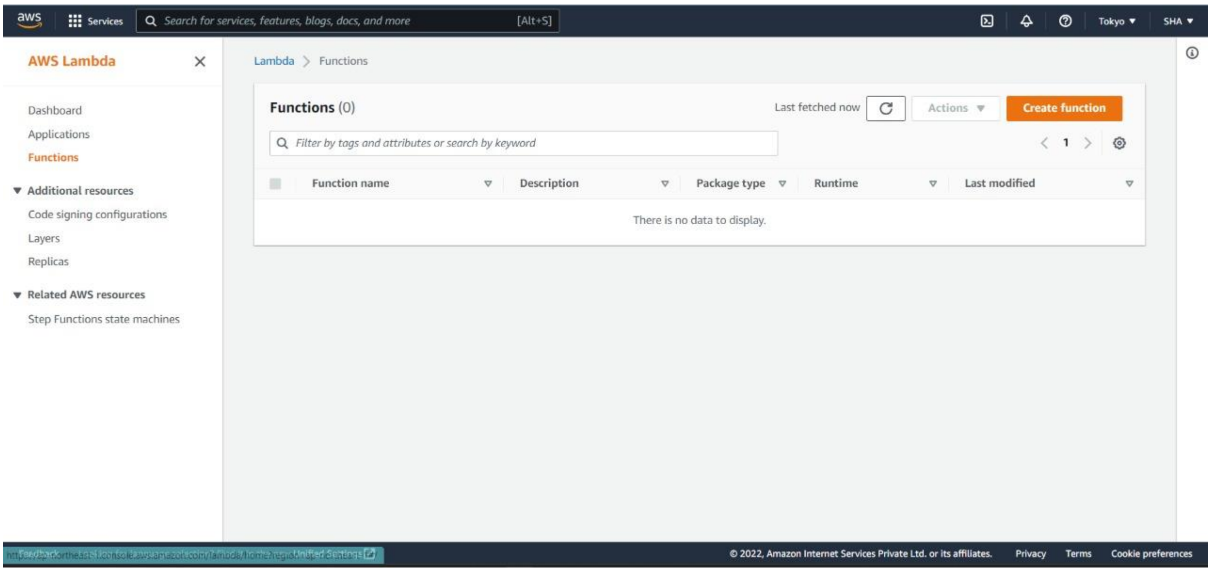
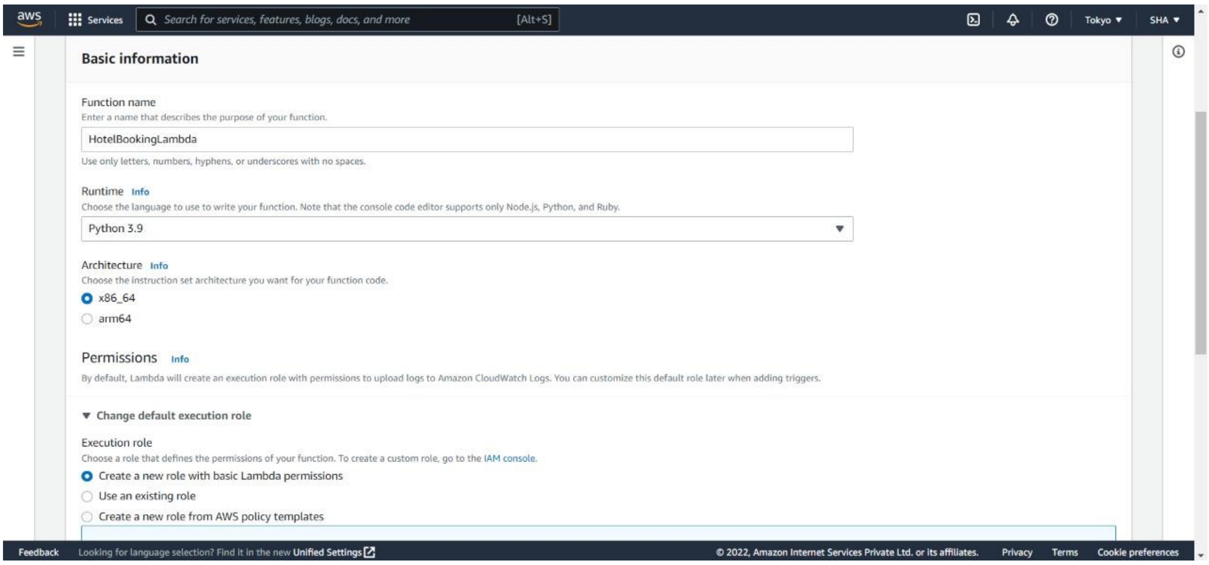


FIG.23

Fill the information accordingly in the fields


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Tokyo ▾
SHA ▾
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences

Basic information

Function name
Enter a name that describes the purpose of your function.
HotelBookingLambda
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.9

Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

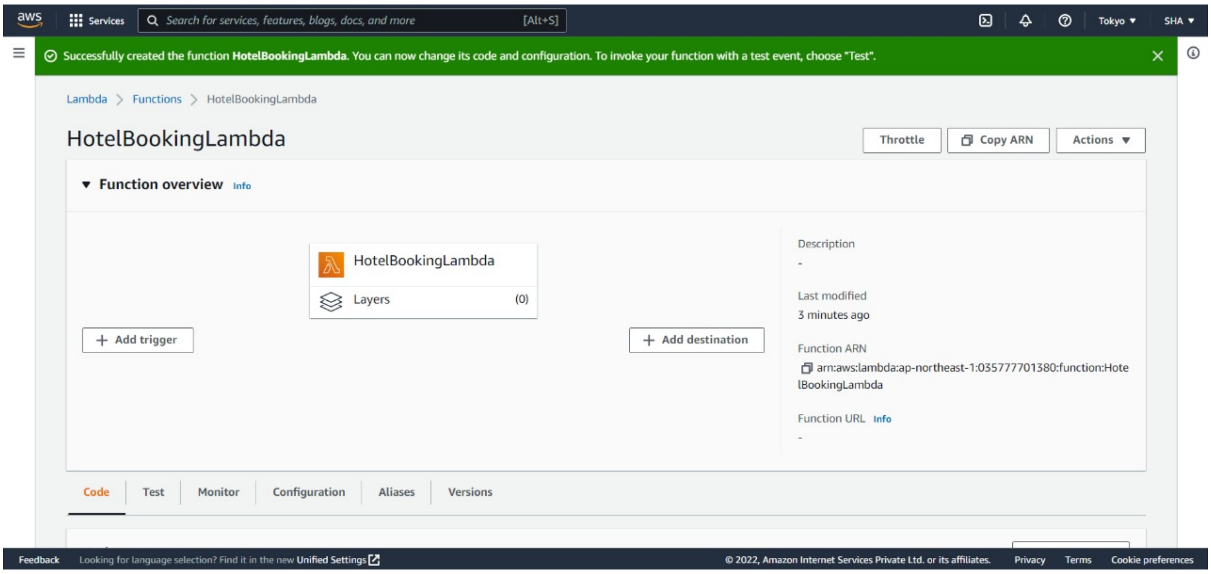
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
● Create a new role with basic Lambda permissions
○ Use an existing role
○ Create a new role from AWS policy templates

FIG.24

Function is created successfully.


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Tokyo ▾
SHA ▾
Lambda > Functions > HotelBookingLambda
Throttle
Copy ARN
Actions ▾
+ Add trigger
+ Add destination
Code
Test
Monitor
Configuration
Aliases
Versions
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences

Successfully created the function **HotelBookingLambda**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

HotelBookingLambda

▼ Function overview Info

HotelBookingLambda
Layers (0)

Description
-

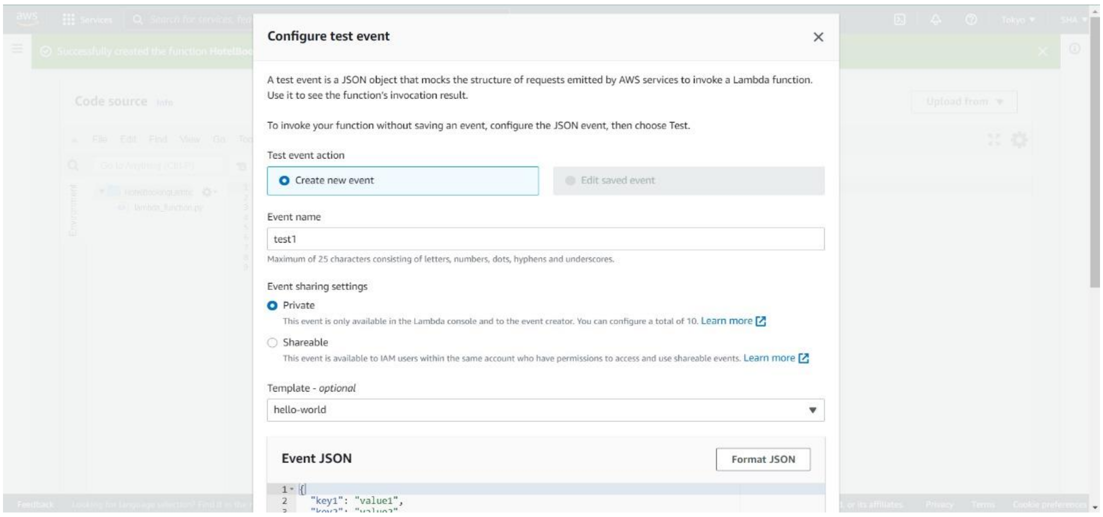
Last modified
3 minutes ago

Function ARN
arn:aws:lambda:ap-northeast-1:035777701380:function:HotelBookingLambda

Function URL Info
-

FIG.25

Creating test event



Configure test event

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

● Create new event ○ Edit saved event

Event name: test1

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

● Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more

○ Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

Template - *optional*: hello-world

Event JSON

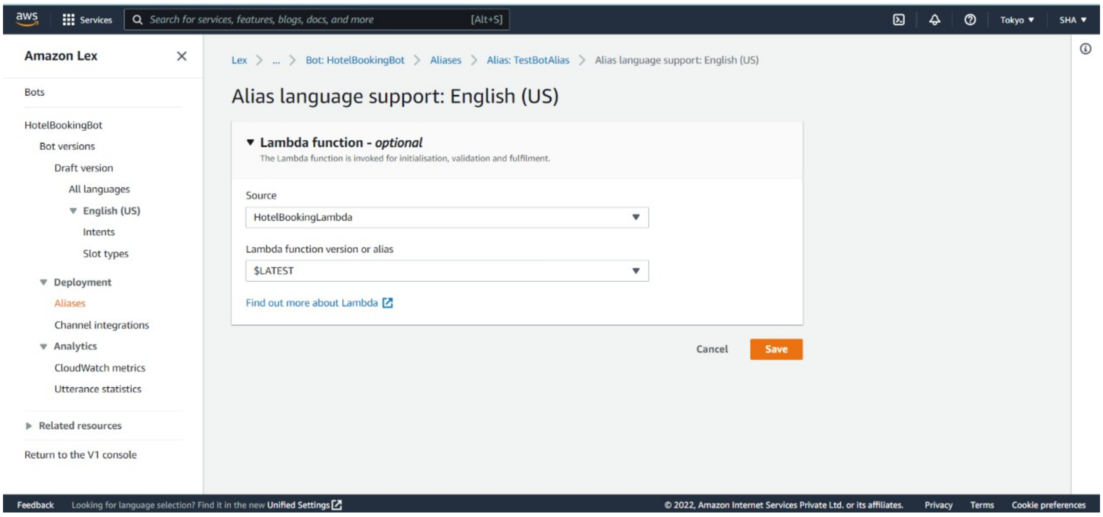
```
{
  "key1": "value1",
  "key2": "value2"
```

FIG.26

Test event created successfully and the execution results saying hello world.

We wish to trigger the lambda with the help of lex. To achieve this, we need to utilize the aliases function in the deployment section of amazon lex.

Utilizing aliases to create versions of our lambda function.



Lex > ... > Bot: HotelBookingBot > Aliases > Alias: TestBotAlias > Alias language support: English (US)

Alias language support: English (US)

Lambda function - *optional*

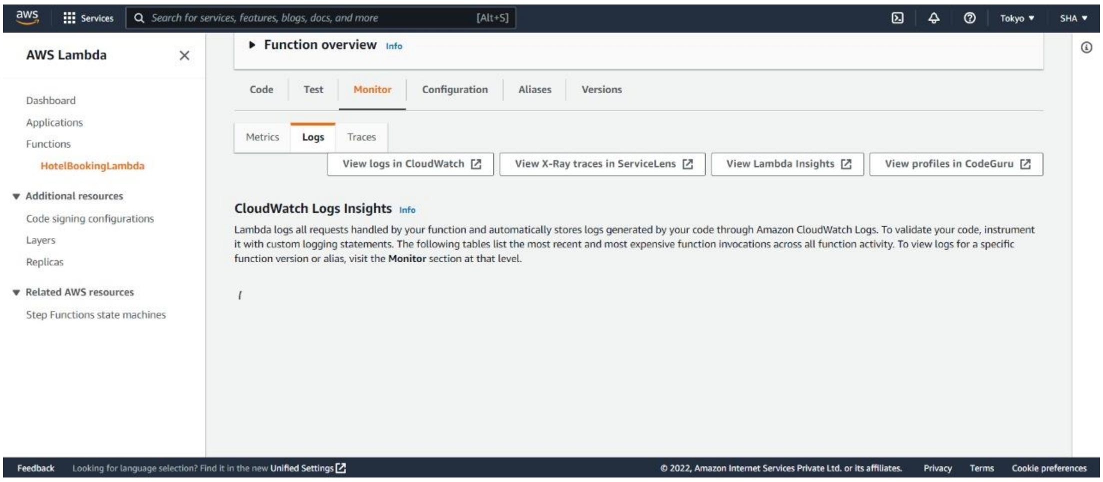
The Lambda function is invoked for initialisation, validation and fulfilment.

Source: HotelBookingLambda

Lambda function version or alias: $LATEST

FIG.27

Opening the cloud watch -



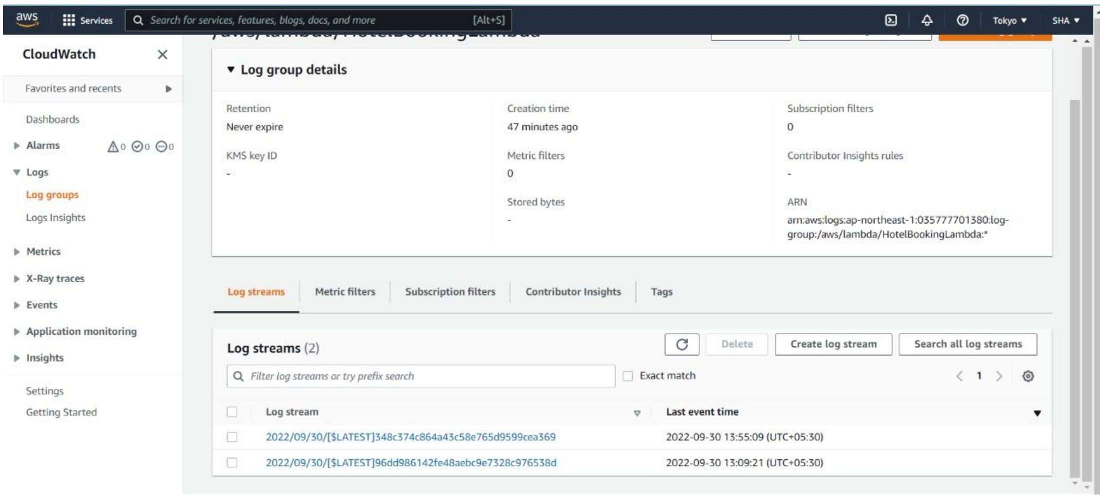
aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Tokyo ▾
SHA ▾
AWS Lambda
Dashboard
Applications
Functions
HotelBookingLambda
Additional resources
Code signing configurations
Layers
Replicas
Related AWS resources
Step Functions state machines
Function overview Info
Code
Test
Monitor
Configuration
Aliases
Versions
Metrics
Logs
Traces
View logs in CloudWatch
View X-Ray traces in ServiceLens
View Lambda Insights
View profiles in CodeGuru
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences

CloudWatch Logs Insights Info

Lambda logs all requests handled by your function and automatically stores logs generated by your code through Amazon CloudWatch Logs. To validate your code, instrument it with custom logging statements. The following tables list the most recent and most expensive function invocations across all function activity. To view logs for a specific function version or alias, visit the **Monitor** section at that level.

FIG.28

Cloud watch logs of the amazon lambda –



Log group details

Retention
Never expire

KMS key ID
-

Creation time
47 minutes ago

Metric filters
0

Stored bytes
-

Subscription filters
0

Contributor Insights rules
-

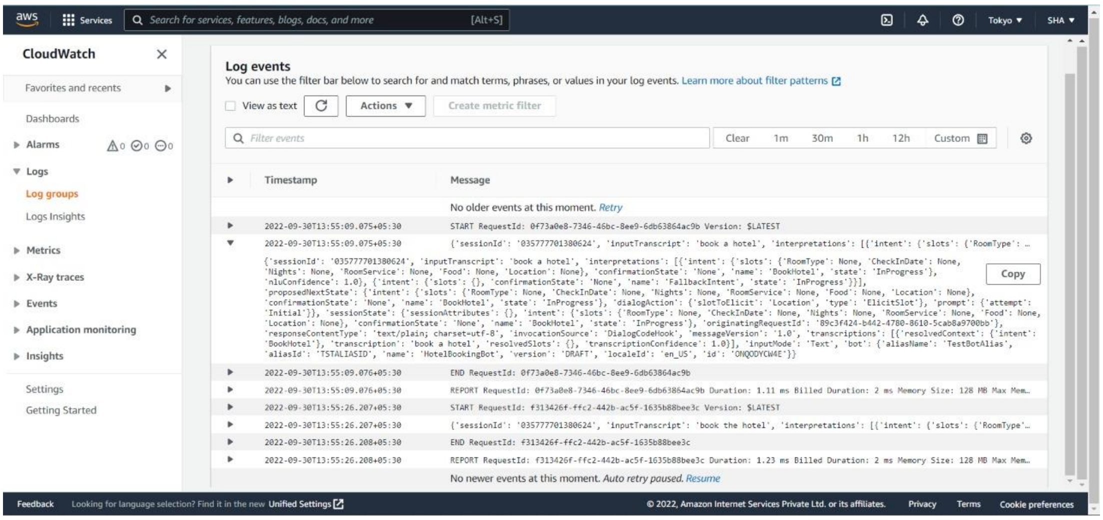
ARN
arn:aws:logs:ap-northeast-1:035777701380:log-group:/aws/lambda/HotelBookingLambda:*

Log streams (2)

	Log stream	Last event time
□	2022/09/30/[$LATEST]348c374c864a43c58e765d9599cea369	2022-09-30 13:55:09 (UTC+05:30)
□	2022/09/30/[$LATEST]96dd986142fe48aebc9e7328c976538d	2022-09-30 13:09:21 (UTC+05:30)

FIG.29

After opening first link in the log streams window will show the events in events history.



Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

```
Timestamp                         Message
                                  No older events at this moment. Retry
2022-09-30T13:55:09.075+05:30     START RequestId: 0f73a0e8-7346-46bc-8ee9-6db63864ac9b Version: $LATEST
2022-09-30T13:55:09.075+05:30     {'sessionId': '035777701380624', 'inputTranscript': 'book a hotel', 'interpretations': [{'intent': {'slots': {'RoomType': …
{'sessionId': '035777701380624', 'inputTranscript': 'book a hotel', 'interpretations': [{'intent': {'slots': {'RoomType': None, 'CheckInDate': None,
'Nights': None, 'RoomService': None, 'Food': None, 'Location': None}, 'confirmationState': 'None', 'name': 'BookHotel', 'state': 'InProgress'},
'nluConfidence': 1.0}, {'intent': {'slots': {}, 'confirmationState': 'None', 'name': 'FallbackIntent', 'state': 'InProgress'}}],
'proposedNextState': {'intent': {'slots': {'RoomType': None, 'CheckInDate': None, 'Nights': None, 'RoomService': None, 'Food': None, 'Location': None},
'confirmationState': 'None', 'name': 'BookHotel', 'state': 'InProgress'}, 'dialogAction': {'slotToElicit': 'Location', 'type': 'ElicitSlot'}, 'prompt': {'attempt':
'Initial'}}, 'sessionState': {'sessionAttributes': {}, 'intent': {'slots': {'RoomType': None, 'CheckInDate': None, 'Nights': None, 'RoomService': None, 'Food': None,
'Location': None}, 'confirmationState': 'None', 'name': 'BookHotel', 'state': 'InProgress'}, 'originatingRequestId': '89c3f424-b442-4780-8610-5cab8a9700bb'},
'responseContentType': 'text/plain; charset=utf-8', 'invocationSource': 'DialogCodeHook', 'messageVersion': '1.0', 'transcriptions': [{'resolvedContext': {'intent':
'BookHotel'}, 'transcription': 'book a hotel', 'resolvedSlots': {}, 'transcriptionConfidence': 1.0}], 'inputMode': 'Text', 'bot': {'aliasName': 'TestBotAlias',
'aliasId': 'TSTALIASID', 'name': 'HotelBookingBot', 'version': 'DRAFT', 'localeId': 'en_US', 'id': 'ONQODYCW4E'}}
2022-09-30T13:55:09.076+05:30     END RequestId: 0f73a0e8-7346-46bc-8ee9-6db63864ac9b
2022-09-30T13:55:09.076+05:30     REPORT RequestId: 0f73a0e8-7346-46bc-8ee9-6db63864ac9b Duration: 1.11 ms Billed Duration: 2 ms Memory Size: 128 MB Max Mem…
2022-09-30T13:55:26.207+05:30     START RequestId: f313426f-ffc2-442b-ac5f-1635b88bee3c Version: $LATEST
2022-09-30T13:55:26.207+05:30     {'sessionId': '035777701380624', 'inputTranscript': 'book the hotel', 'interpretations': [{'intent': {'slots': {'RoomType'…
2022-09-30T13:55:26.208+05:30     END RequestId: f313426f-ffc2-442b-ac5f-1635b88bee3c
2022-09-30T13:55:26.208+05:30     REPORT RequestId: f313426f-ffc2-442b-ac5f-1635b88bee3c Duration: 1.23 ms Billed Duration: 2 ms Memory Size: 128 MB Max Mem…
                                  No newer events at this moment. Auto retry paused. Resume
```

FIG.30

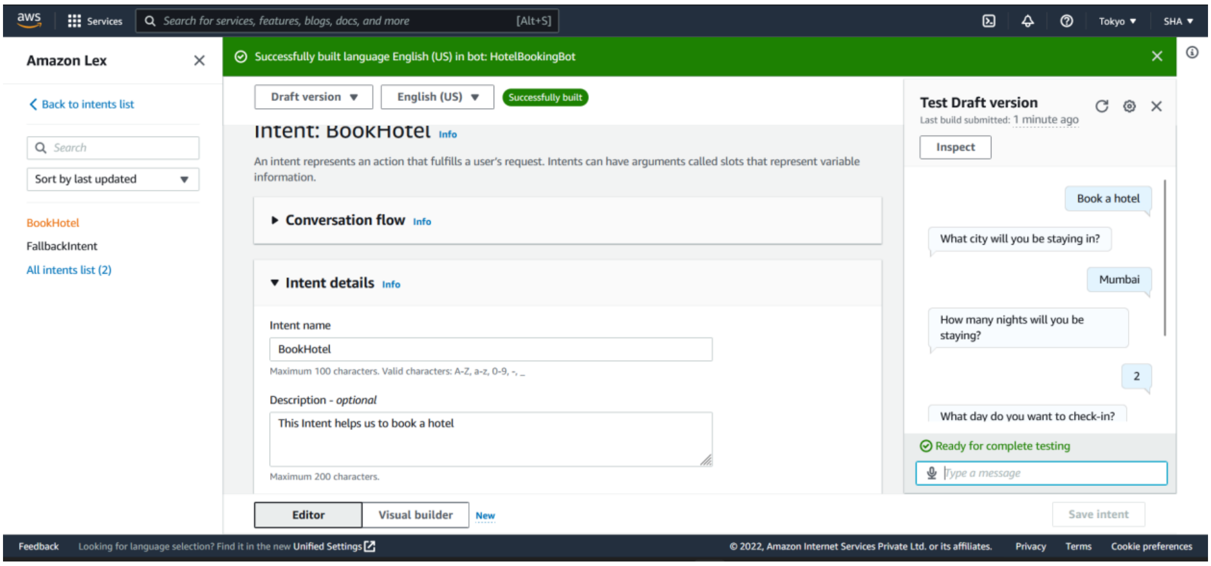
The Lambda Code for the backend of the operator

```
import json
def validate(slots):
    if not slots['Location']:
        print("Inside Empty Location")
        return {
        'isValid': False,
        'violatedSlot': 'Location'
        }
    if not slots['CheckInDate']:

        return {
        'isValid': False,
        'violatedSlot': 'CheckInDate',
    }

    if not slots['Nights']:
        return {
        'isValid': False,
        'violatedSlot': 'Nights'
    }

    if not slots['RoomType']:
        return {
        'isValid': False,
        'violatedSlot': 'RoomType'
    }
    return {'isValid': True}

def lambda_handler(event, context):

    slots = event['sessionState']['intent']['slots']
    intent = event['sessionState']['intent']['name']
    print(event['invocationSource'])
    print(slots)
    print(intent)
    validation_result = validate(slots)
    if event['invocationSource'] == 'DialogCodeHook':

    if not validation_result['isValid']:

    response = {
                "sessionState": {
                    "dialogAction": {

'slotToElicit':validation_result{'violatedSlot'},
                        "type": "ElicitSlot"
                    },
                    "intent": {
                        'name':intent,
                        'slots': slot
                        }
                }
               }
    return response
```

Then testing will be sone for the bot.

5. Test Results

Now, Test the intent for working of the Bot.. which will look like as –



Successfully built language English (US) in bot: HotelBookingBot

Intent: BookHotel Info

An intent represents an action that fulfills a user's request. Intents can have arguments called slots that represent variable information.

Intent name: BookHotel

Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, _

Description - *optional*: This Intent helps us to book a hotel

Maximum 200 characters.

Test Draft version — Last build submitted: 1 minute ago

Book a hotel

What city will you be staying in?

Mumbai

How many nights will you be staying?

2

What day do you want to check-in?

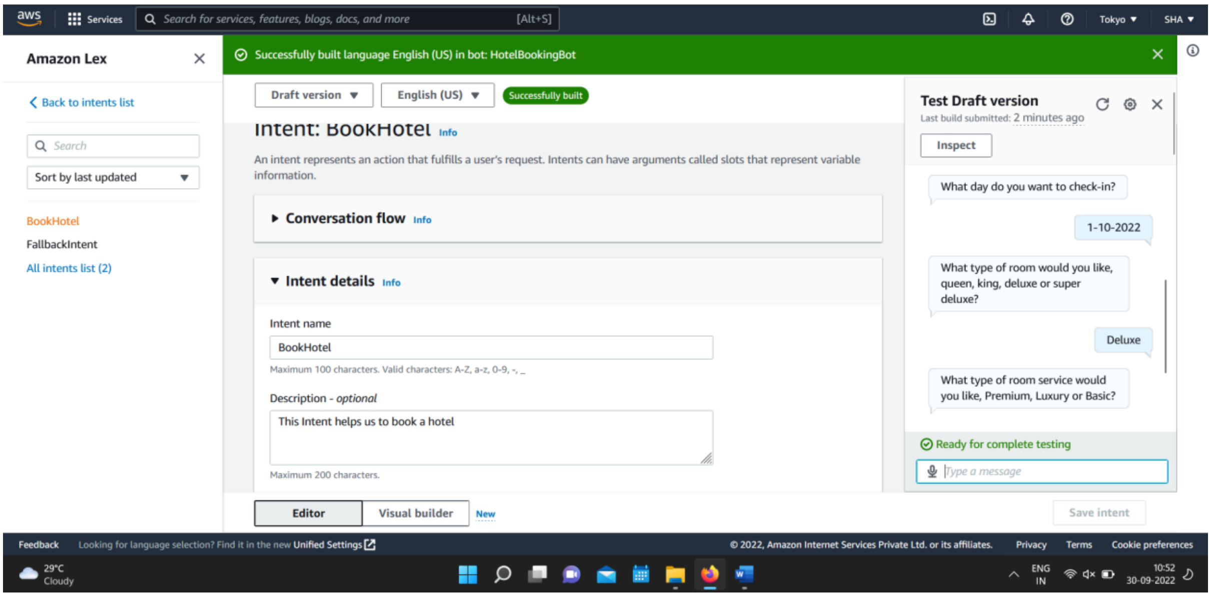
Ready for complete testing

Type a message

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

FIG. 31



Successfully built language English (US) in bot: HotelBookingBot

Intent: BookHotel Info

An intent represents an action that fulfills a user's request. Intents can have arguments called slots that represent variable information.

Intent name: BookHotel

Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, _

Description - *optional*: This Intent helps us to book a hotel

Maximum 200 characters.

Test Draft version — Last build submitted: 2 minutes ago

What day do you want to check-in?

1-10-2022

What type of room would you like, queen, king, deluxe or super deluxe?

Deluxe

What type of room service would you like, Premium, Luxury or Basic?

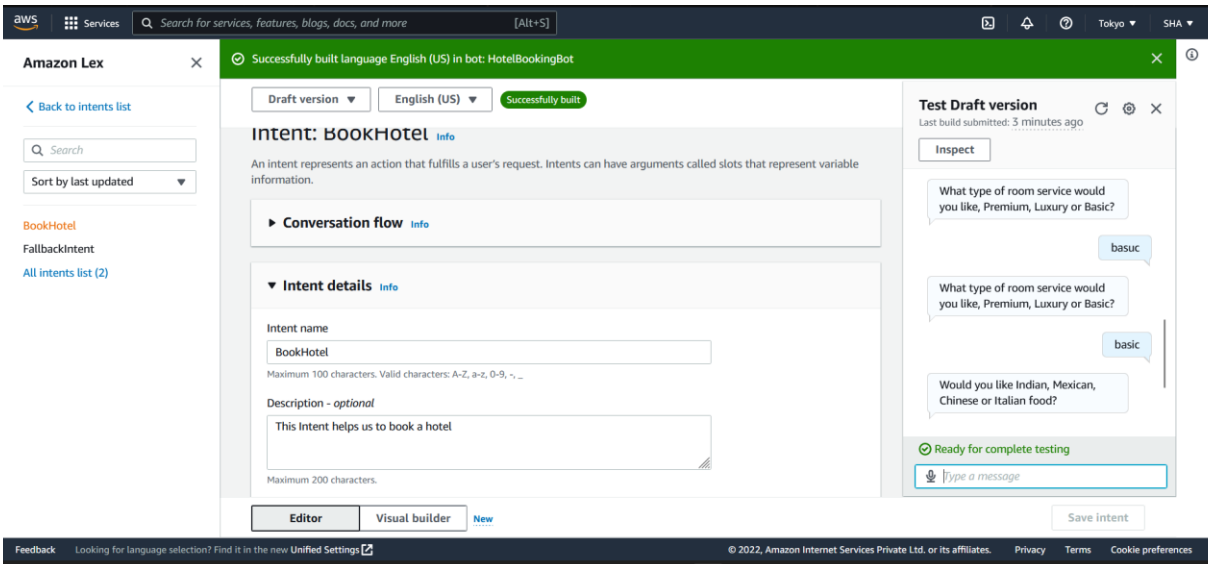
Ready for complete testing

Type a message

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

FIG.32



Successfully built language English (US) in bot: HotelBookingBot

Intent: BookHotel

An intent represents an action that fulfills a user's request. Intents can have arguments called slots that represent variable information.

Intent name: BookHotel

Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, _

Description - *optional*: This Intent helps us to book a hotel

Maximum 200 characters.

Test Draft version

Last build submitted: 3 minutes ago

What type of room service would you like, Premium, Luxury or Basic?

basuc

What type of room service would you like, Premium, Luxury or Basic?

basic

Would you like Indian, Mexican, Chinese or Italian food?

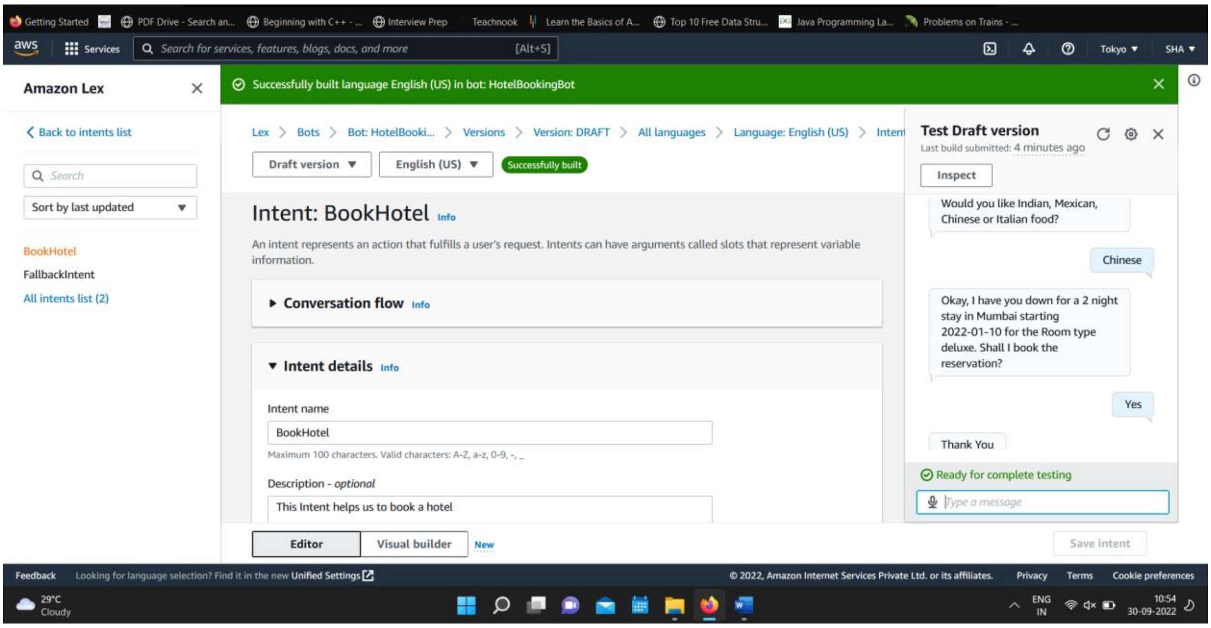
Ready for complete testing

Type a message

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

FIG.33



Successfully built language English (US) in bot: HotelBookingBot

Intent: BookHotel

An intent represents an action that fulfills a user's request. Intents can have arguments called slots that represent variable information.

Intent name: BookHotel

Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, _

Description - *optional*: This Intent helps us to book a hotel

Test Draft version

Last build submitted: 4 minutes ago

Would you like Indian, Mexican, Chinese or Italian food?

Chinese

Okay, I have you down for a 2 night stay in Mumbai starting 2022-01-10 for the Room type deluxe. Shall I book the reservation?

Yes

Thank You

Ready for complete testing

Type a message

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

FIG.34

6. Results and Discussions

Initially, the chatbot successfully provided the right output in about less number of cases. However, as the training data improved with continuous interaction with the chatbot, this accuracy has improved to a greater extent. For queries unrelated to the intents described in the chatbot, a generalized response is generated, and such questions were logged to be checked later and included to the datasets. The Amazon S3 connected to the chatbot have successfully monitored the traffic to various sections of the chatbot by capturing the general purpose of the user. This information can further be used to analyse the kind of people visiting the site and their purpose. Although people aren't always precise in their wording when they interact with a bot, we still want to provide them with a natural user experience.

This work would bring comfort and security to not just the visually impaired, but the nation as a whole. The voice chatbot system will be used as an alternative form of payment whereby users can engage in financial transactions such as sending of money just by speaking. According to the results obtained from the tests carried out, it can be seen that implementing the voice chatbot device is beneficial in numerous ways. With natural language understanding improvements and confidence scores now available on Amazon Lex, one can have additional information available to design a more intelligent conversation. Developers can couple the machine learning-based intent matching capabilities of Amazon Lex with your own business logic to zero in on your user's intent. Developers and creators can also use the confidence score threshold while testing during bot development, to determine if changes to the sample utterances for intents have the desired effect. These improvements enable you to design more effective conversation flows. So finally, Hotel Booking system chatbot will be prepared using lambda function as a backend.

7. CONCLUSION AND FUTURE WORK

7.1. Conclusion

In this I have designed the hotel Booking chatbot for having the answers of user simple queries related to booking rooms and using services available. In this I have used Amazon Services like Amazon Lex, Amazon Lambda for creating the backend. For the Lambda Service we have to use any one of the programming languages. So I have used the highly usable and easily understandable language which is Python. This project have given a practical introduction to chatbots through building increasingly complex amazon skills and lex chatbots. I have created a perfect conversation and creating flow diagrams to visualize the users conversational path with a chatbot. Using these flow diagram, I have built intents using utterances and slots that are handled in lambdas.

I have improved the features and abilities of our chatbots through the use of services like Amazon Lex, Amazon Lambda. To improve the user experience, I have also created restricted slot types which will give only available answers. If user passes a question or query like book a hotel and then bot will the questions according to the conversational flow. This is how the finally bot works perfectly without any errors.

7.2. Future Work

Chatbots have come a long way over the last decade and are now often found in households through Amazon Echo and Google Home Technologically, they have improved in leaps and bounds, with improvements in Al and machine leaming resulting in better language understanding as well as voice-to-text that power the Echo and Google Home devices. I expect that the growth of chatbots will continue and we'll start seeing them used in a huge range of application and through a wide range of devices. As they improve, they'll be trusted with increasingly complex and important tasks and will completely revolutionize a lot of industries. Industries such as customer services are already changing, with chatbots on multiple banking and retail websites and phone systems.

The future work I would like to do is bot working with spoking interactions like Amazon Alexa. As with language understanding, being able to respond to a user's request means being able to understand what they said. With voice systems, this involves converting the spoken sound waves into text. Whilst this can work brilliantly if you happen to speak clearly with a neutral accent, there are often issues when people speak very quickly or with a strong accent. When the text is generated from the users with a strong accent, it can often be misunderstood, and the text that is produced makes no sense. This then means that when it is passed into the language-understanding system, the speech can't be matched to an intent. This can be very frustrating for users with a strong accent who are unable to interact with these devices. This is a significant hurdle to overcome before voice-based chatbots become common in commercial application.

8. References

[1] Overview of AWS Services
https://d1.awsstatic.com/whitepapers/awsoverview.pdf?did=wp_card&trk=wp_card

[2] Why prefer AWS over On-Premise Hosting

https://www.fingent.com/blog/why-prefer-aws-overon-premise-hosting

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski,G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A view of cloud computing. Communications of the ACM, 53(4):50–58, 2010.

[4] I. Bermudez, S. Traverso, M. Mellia, and M. Munafo. Exploring the cloud from passive measurements: The Amazon AWS case. 2013 Proceedings IEEE INFOCOM, pages 230–234, 2013.

[5] A. Li, X. Yang, S. Kandula, and M. Zhang. Cloudcmp: Comparing public cloud providers. In Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, IMC '10, pages 1–14, New York, NY, USA, 2010. ACM.

[6] Ady, M., & Quadri-Felitti, D. (2016). Consumer research identifies how to present travel review content for more bookings. Trust You and New York University Joint Study.

[7] Amir, A. (2016). The rise of chatbots: Why brands are embracing conversation. Marketing Tech News. Retrieved from htttp://www.marketingtechnews.net/news/2016/jun/ 07/rise-and-rise-chatbots-why-brands-are-embracingconversation/fb

[8] Lee, C.-H. (2004). From knowledge-ignorant to knowledge rich modelling: A new speech research paradigm for next generation automatic speech recognition. In Proceedings of the International Conference on Spoken Language Processing (ICLSP), 109-111.

[9] Shawar, B. A., & Atwell, E. (2002). A comparison between Alice and Elizabeth chatbot system. University of Leeds, School of Computing Research Report No. 2002/19.

[10] Gaines-Ross, L. (2016). What do people - Not techies, not companies - Think about artificial intelligence?. Harvard Business Review (Digital Version), 2016/10.