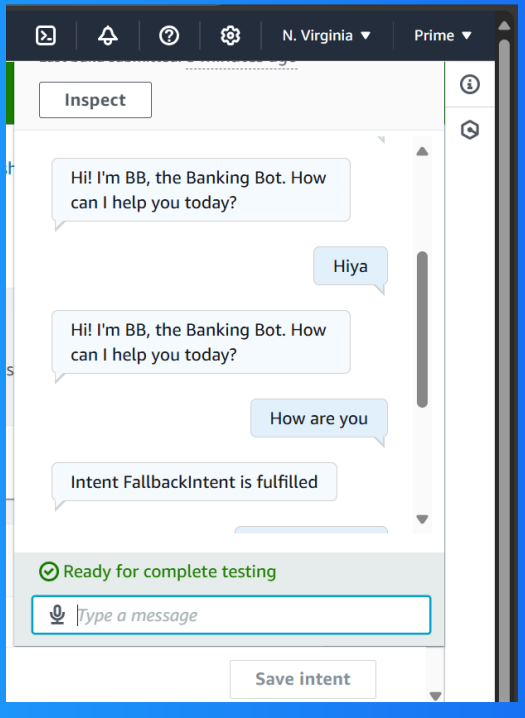
# Build a Chatbot with Amazon Lex

Unmilan Mukherjee



# Introducing Today's Project!

## What is Amazon Lex?

Amazon Lex is a chatbot service provided by AWS. It is useful as it can be integrated seamlessly with other AWS services like for example: AWS Lambda.

## How I used Amazon Lex in this project

I used Amazon Lex to set up a basic greeting conversation chatbot with a fallback message and variations.

## One thing I didn't expect in this project was...

I did not expect the level of granular control this Lexbot framework would give us in conversations. This includes variations and conversation flowcharts that are completely customziable.

## This project took me...

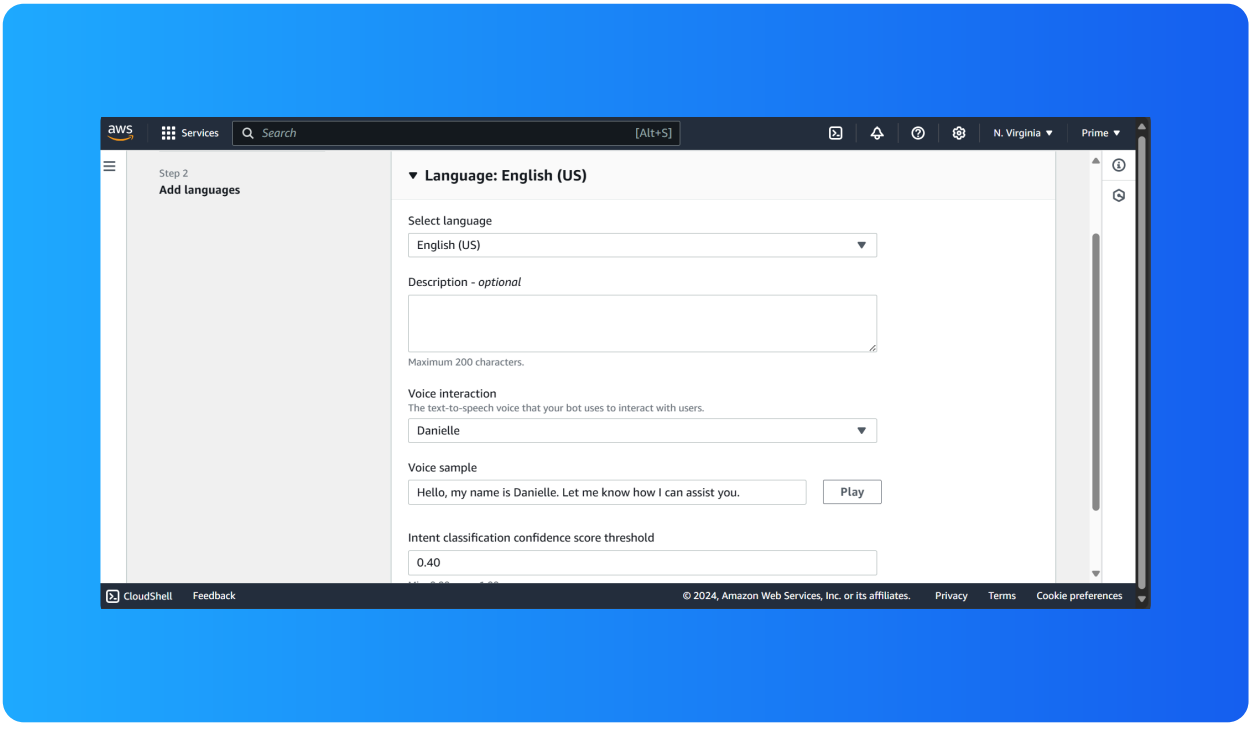
This project took me 30 minutes.

# Setting up a Lex chatbot

I created my chatbot from scratch with Amazon Lex. Setting it up took me 2 minutes.

While creating my cahtbot, I also created a role with basic permissions because Lex needs permissions to call other AWS services, for example: AWS Lambda.

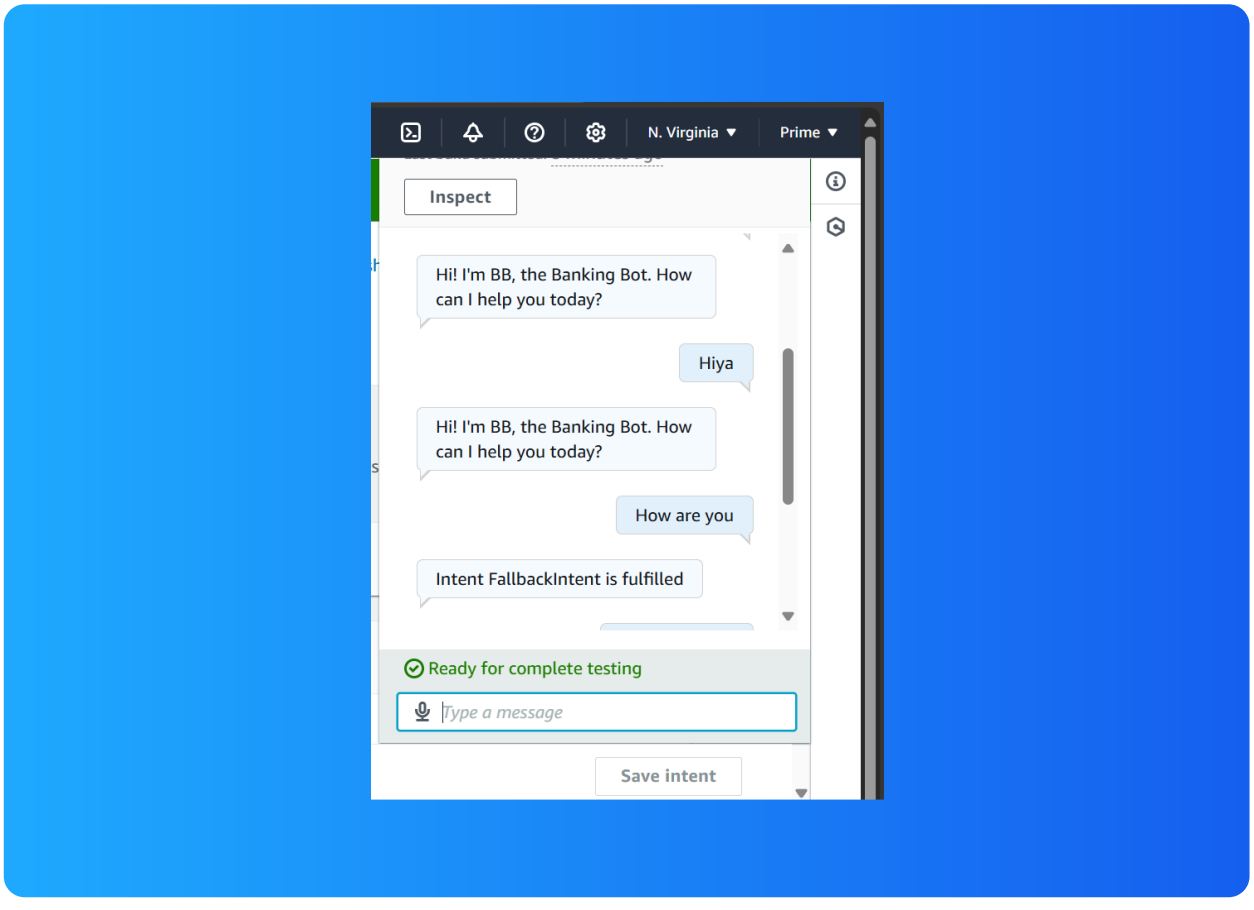
In terms of the intent classifivation confidence score, I kept the default value of 0.40. This means that in order for Lex to return a response, it needs to be atleast 40% confident that it understands the user's query and wants.



# Intents

Intents are what the user is trying to achieve. Amazon Lex uses these intents to understand what the user wants and responds accordingly. For example: Book tickets, raise complaint, etc.

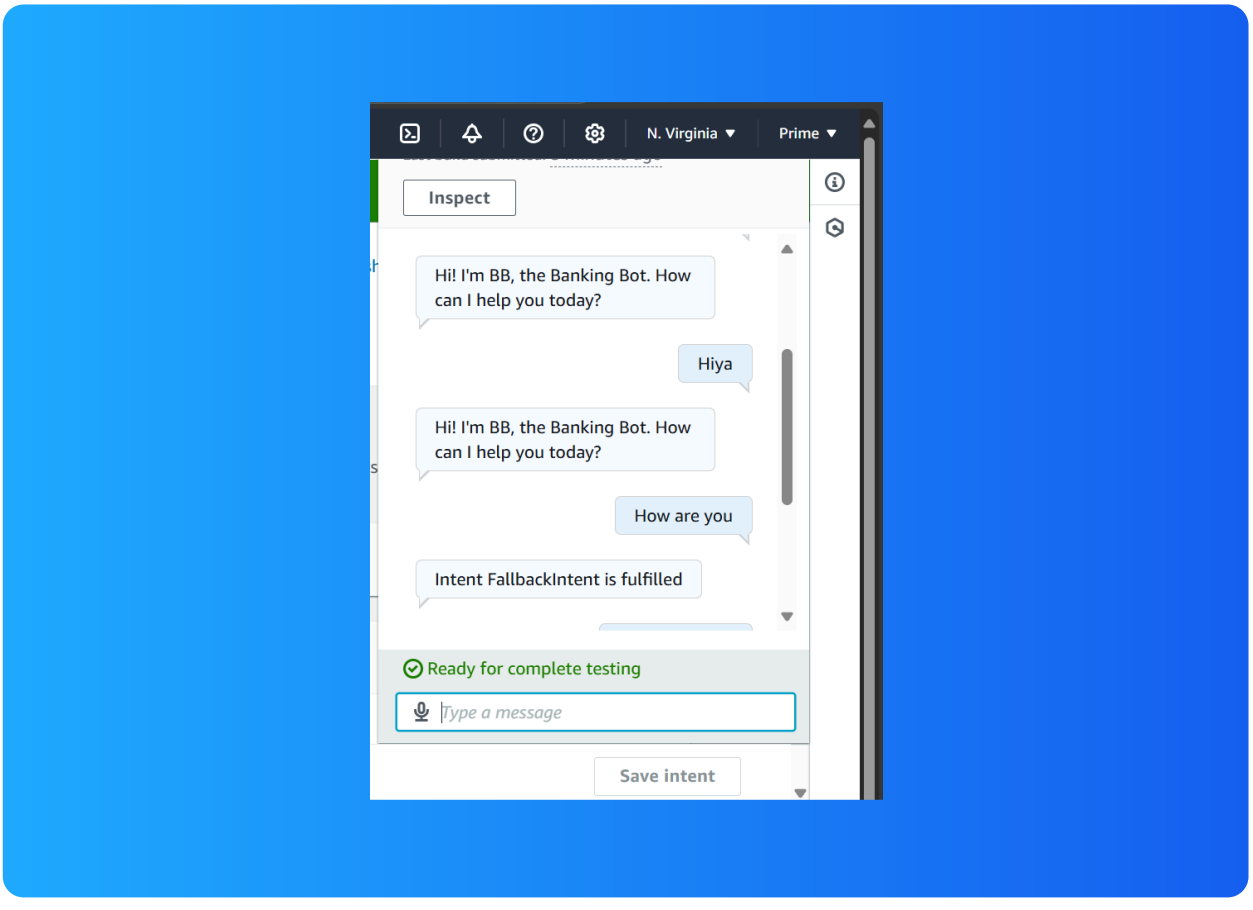
I created my first intent, WelcomeIntent, to determine when the user initiates a conversation with my chatbot and to reply with a warm welcome message to the user.



# FallbackIntent

I launches and tested my chatbot which could respond successfully if I enter any of the phrases I pre defined in my utterances tab.

My chatbot returned the error message: Intent FallbackIntent is fulfilled when I entered: Good morning or How are you. This error message occured because these are not in our utterances and the confidence scores are less than 40%.



# Configuring FallbackIntent

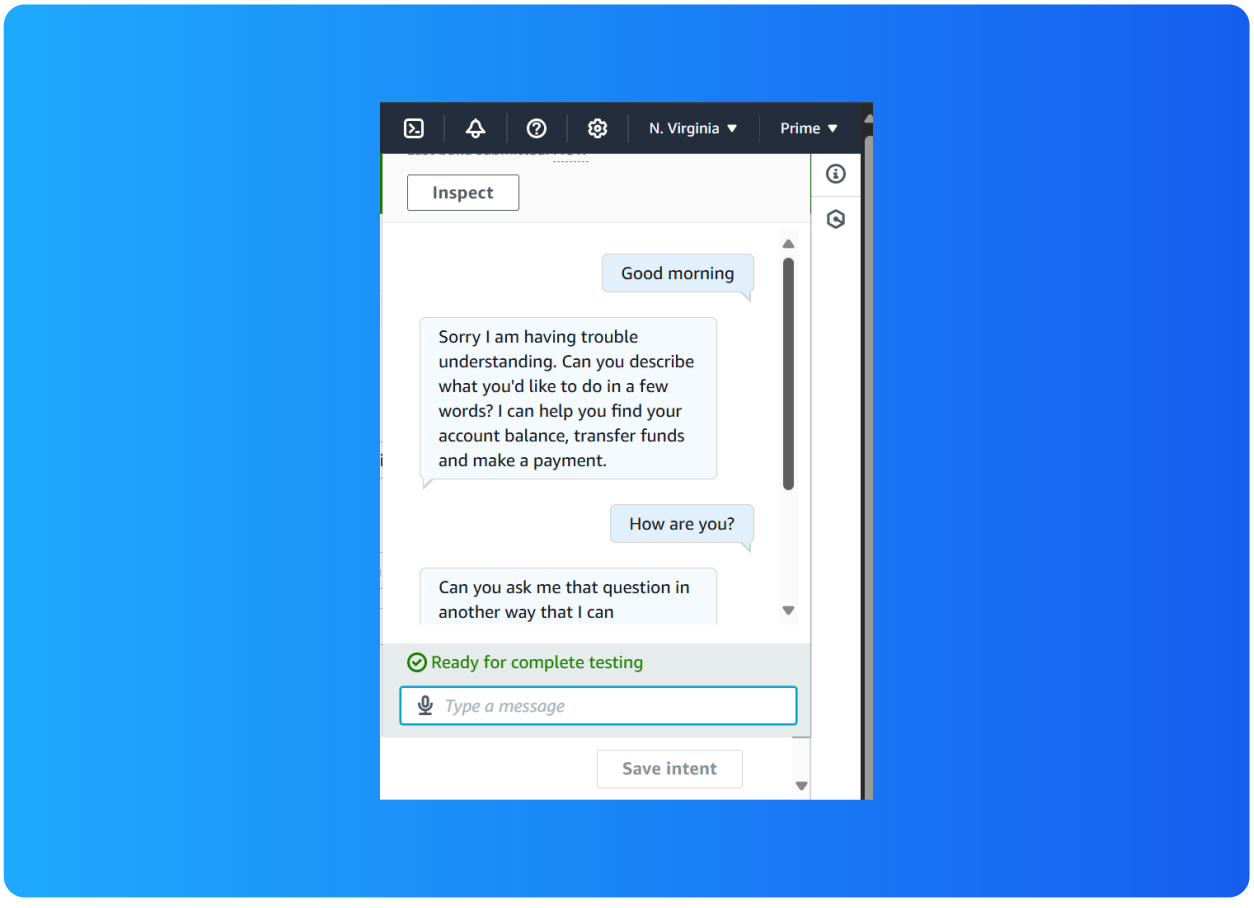
FallbackIntent is a default intent in every chatbot that gets triggered when the user's intent is not in our list of intents or utterances and the chatbot is less than 40% confident of what the user wants.

I wanted to configure FallbackIntent because I do not want the user to view our error message as it looks unprofessional and breaks the flow of natural conversation.

# Variations

To configure FallbackIntent, I selected FallbackIntent in our list of Intents and added a closing response to it.

I also added variations! What this means for an end user is that they don't get the same response from our chatbot. This will help the conversation feel more natural and less robotic.



# Initial Responses

As an extension for this project, I'm setting up initial responses, which are basically a greeting that our Lexbot gives to a user before recognising an intent for the user.

The intial response messages I set up are for the FallbackIntent. For the user, this means that the chatbot will return a kind "wait a minute" message before telling the user to give it a different prompt. This makes the conversation more human.