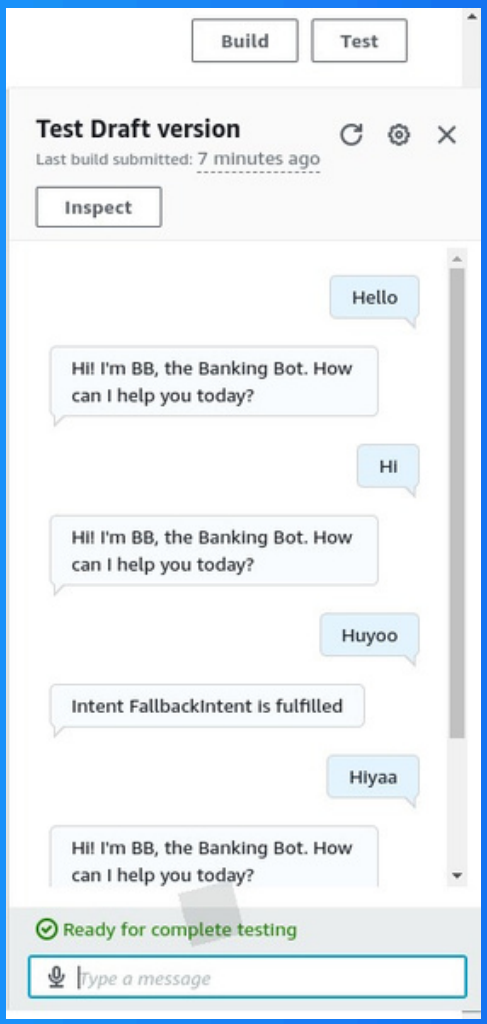
NextWork.org

Build a Chatbot with Amazon Lex

Hassan Gachoka



Hassan Gachoka
linkedin.com/gachokahassan

Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a service for building conversational interfaces with voice and text. Itś' useful for creating chatbots with natural language understanding and speech recognition, enabling seamless user interactions.

How I used Amazon Lex in this project

I used Amazon Lex in today's project to build a banking chatbot that can understand and respond to user inputs. It processes text-based interactions, providing a seamless conversational experience for users.

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

One thing I didn't expect in this project was the importance of training the chatbot with various user inputs. It required thoughtful configuration to ensure accurate intent recognition and smooth interactions.

This project took me...

This project took me 50 minutes to complete. In Part 1, I set up a basic WelcomeIntent, created lists of utterances, handled failures with FallbackIntent, defined a MessageGroup for response variations, and tested the bot with text and speech.

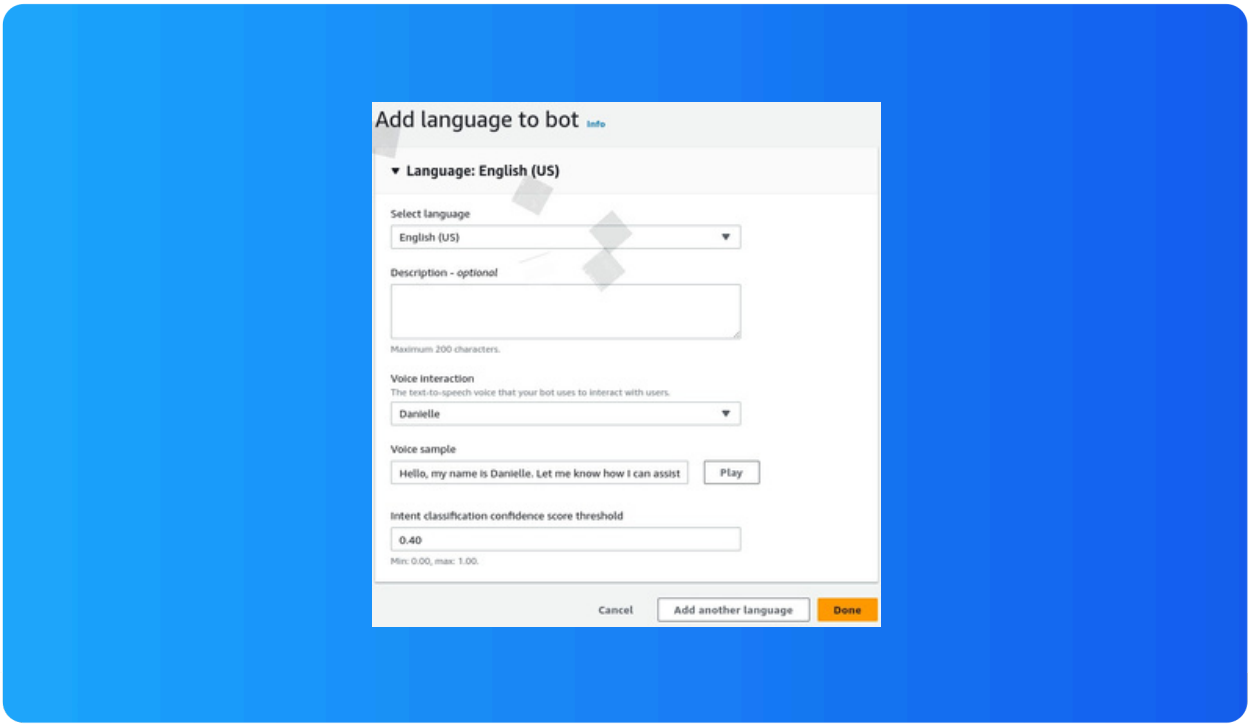
Hassan Gachoka
linkedin.com/gachokahassan

Setting up a Lex chatbot

I created my chatbot from scratch with Amazon Lex. Part 1 took me 30 minutes to design intents and configure FallbackIntent, followed by an additional 10 minutes for testing and fine-tuning to ensure seamless interactions.

While creating my chatbot, I also created a role with basic permissions because it ensures secure access to Amazon Lex and its integrated services like Lambda without exposing unnecessary resources. This minimizes potential security risks.

In terms of the intent classification confidence score, I kept the default value of 0.40. This means the chatbot triggers an intent only if the confidence score is at least 40%, ensuring accurate and relevant responses to user inputs.

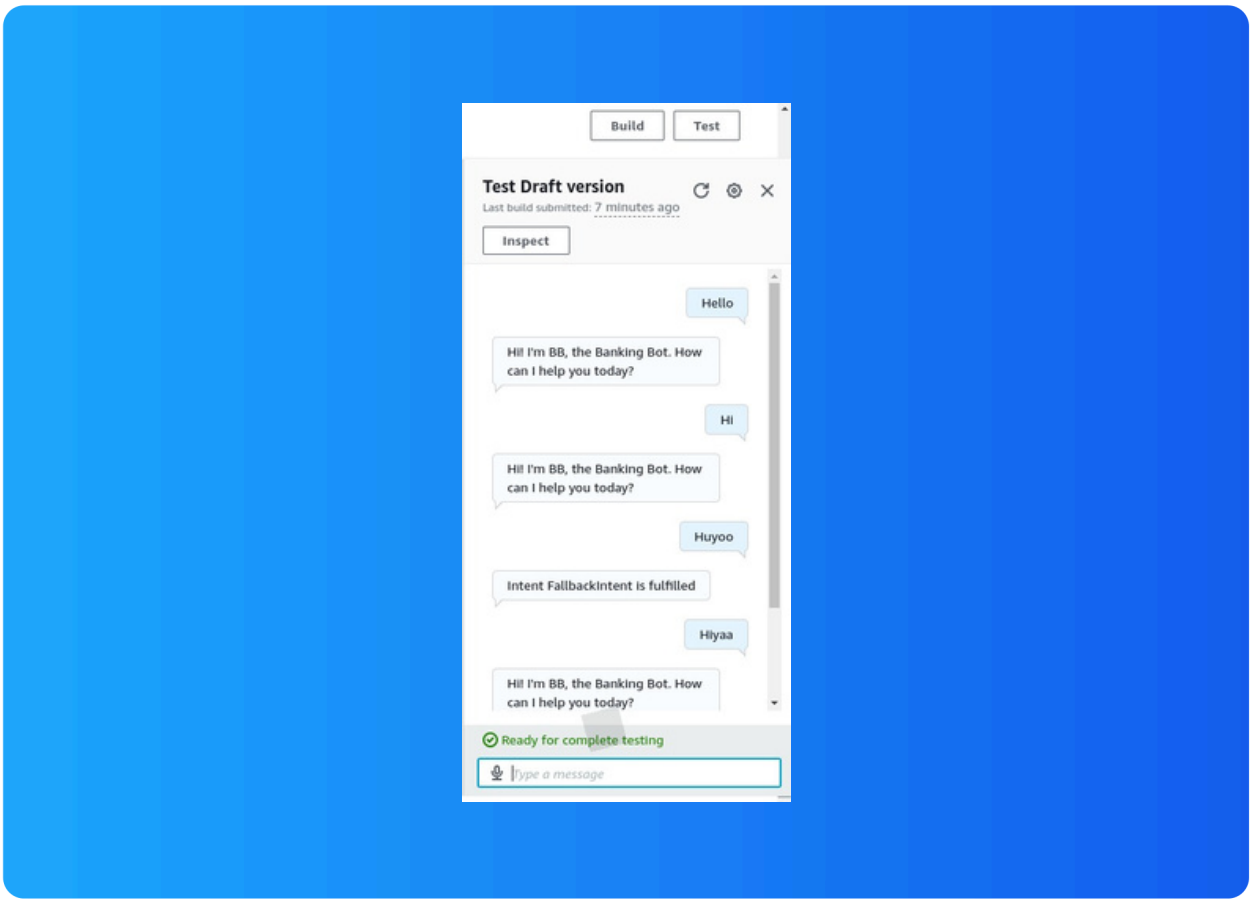


Hassan Gachoka
linkedin.com/gachokahassan

Intents

Intents are specific goals or actions users want to achieve through a chatbot. They represent the purpose behind a user's input, such as booking a ticket or checking the weather, guiding the chatbot's responses.

I created my first intent, WelcomeIntent, to greet users when they interact with the chatbot. It helps set the tone and context for the conversation, ensuring a friendly and engaging start to the interaction.

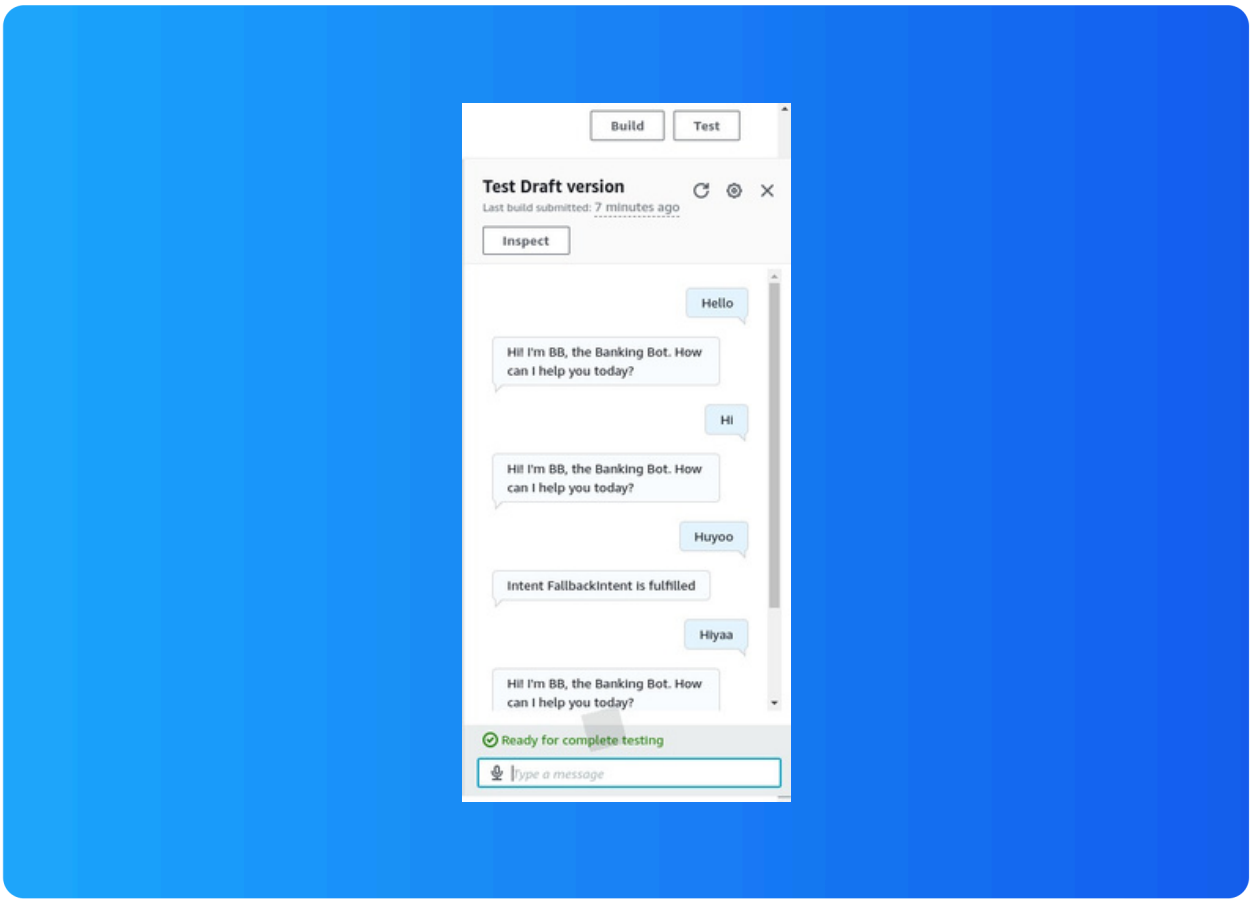


Hassan Gachoka
linkedin.com/gachokahassan

FallbackIntent

I launched and tested my chatbot, which could respond successfully if I enter greetings like "Good morning," "Hey." "Hi," or "Hello," It recognizes multiple variations to ensure a natural and user-friendly experience.

My chatbot returned the error message "Intent FallbackIntent is fulfilled" when I entered an unrecognized input like "Hiyaa" "Huyoo". This error message occurred because the chatbot didn't understand the phrase and defaulted to the fallback intent.



Hassan Gachoka
linkedin.com/gachokahassan

Configuring FallbackIntent

FallbackIntent is a default intent in every chatbot that gets triggered when the user's input doesn't match any of the predefined intents. It handles unrecognized phrases and provides a fallback response.

I wanted to configure FallbackIntent because it helps handle unrecognized inputs, ensuring my chatbot can respond appropriately when it doesn't understand the user's query.

Hassan Gachoka
linkedin.com/gachokahassan

Variations

To configure FallbackIntent, I created it as a default intent, set up sample phrases for unrecognized inputs, and defined appropriate responses to guide users when their queries couldn't be understood.

I also added variations! What this means for an end user is that the chatbot can recognize different ways of phrasing the same intent, improving its ability to understand and respond to various user inputs effectively.