Building Chatbot with Amazon Lex

Summary - This project showcases a series of chatbots built using Amazon Lex, integrated with AWS Lambda, to simulate intelligent, real-time conversational interfaces. The chatbot handles greetings, fallback cases, user-specific data queries (like account balances), and fund transfers.

Objective - To build a chatbot that automates basic banking inquiries and transactions, using Amazon Lex's NLP capabilities combined with serverless computing from AWS Lambda.”

Technologies and services used

- Amazon Lex
- Amazon Lambda
- IAM
- Amazon Console
- NLP concepts like **Utterances, Slots, Fallback, Context Tags**
- Python(lambda code function)
- Cloud Formation

What is **Amazon Lex** and how is it used?

Amazon Lex is a **Natural language Processing** (understands the user input) service used for building interfaces like chatbots by using text and voice . Holds conversations with the help of **Intents** and **Slots**.

Use case of these Chatbots

1) Customer service support - Automatic answers to FAQs, creating appointment for person chat support based on responses of user.

2) Banking - account balance inquiries, fund transfer, Checking Transactions history of account.

3) E-commerce - Product recommendations, Order purchases, Order tracking, Shopping assistance by changing filters, size, colour, price etc.

These chatbots can be trained specifically to generate correct and relevant responses to the user without any person to respond.

Features Implemented

1. Intents - intent is what user is trying to achieve with a conversation to a chatbot.

An Example of intents are : - Checking Bank balance, Ordering a product,

Booking a ticket , Q and A response from FAQs etc.

2. Welcome Intent

Welcome Intent is used for Greeting users with friendly message like

“Hello I am Banking Bot how can I help you. today”

3. Fallback Intent

Fallback Intent manages unknown queries which are not related to intent of the chatbot ,

and generates a response to user like

”could you please ask me about banking details, I can help you with that”

4. Check Balance Intent

Allows to check bank balance of specified **account Type.**

Add slot types Date and Date of Birth to check whether is eligible or not.

5. Utterances

Utterances are example questions or phrases to trigger specific intent in lex chatbot.

utterances helps lex in identifying the correct intent type , even if word are slightly

different.

6. Custom Slot Types

- Added blank slot type as **account Type** slot for checking user bank balance.
- Defined Restricted slot values to control input quality in Slot resolution.
- Added **Checking** , **Savings** and **Credit** as Slot values helps lex to accept only specific topics for account type.

7. Lambda Integration

- Created an lambda function and deploy the code
- Used AWS Lambda to simulate account balance responses

- Connected Lex to Lambda using code hooks for real-time logic

8 . Context Tags for Seamless Conversations

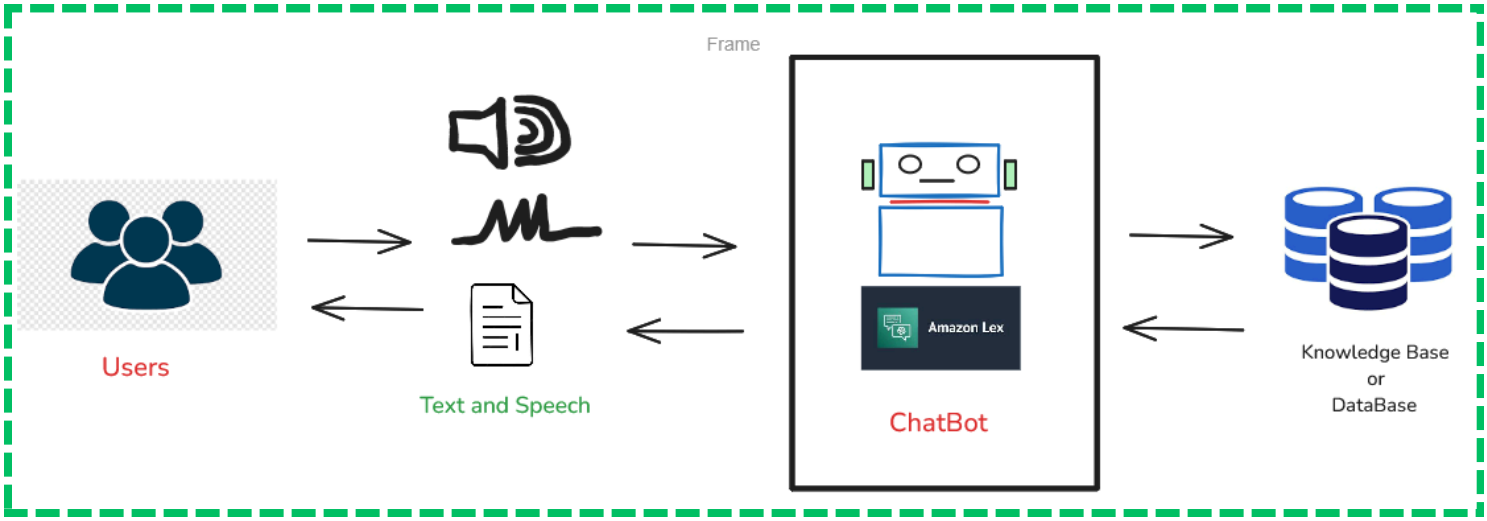
- Used output and input context tags to remember user data (e.g., Date of Birth)
- **Output context tag** will tell chatbot to remember details after conversation to save later example - Account type
- **Input context tag** will check for specific details which are already available like “Date of birth”

9 . **Multi-Slot Interaction** (Transfer Funds)

- Built **TransferFunds intent** with multiple slots
- Add slots **Source Account type, target Account type** with slot type as accountType
- Add slot **transferAmount** with slot type as Number
- Enabled confirmation prompts to validate transactions

10 . CloudFormation

- Used Infrastructure-as-Code (IaC) to deploy the full chatbot configuration with a file template and can create , update and delete stack as needed.



Workflow of Chatbot

User input - User starts a conversation through text or speech.

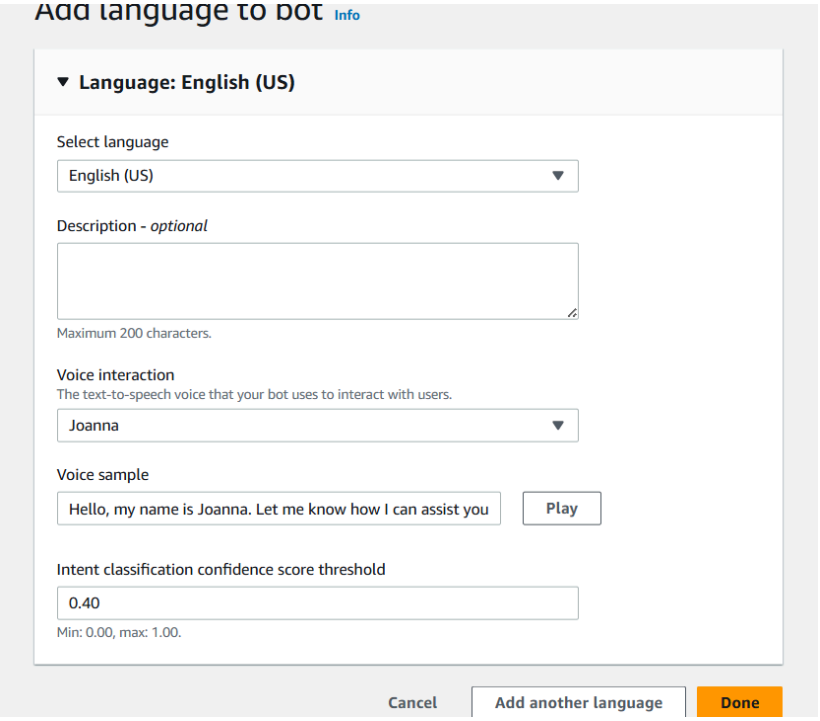
Processing - Chatbot process the input to understand user intents and extract relevant information .

Information retrieval - Here chatbot will access knowledge base or database .

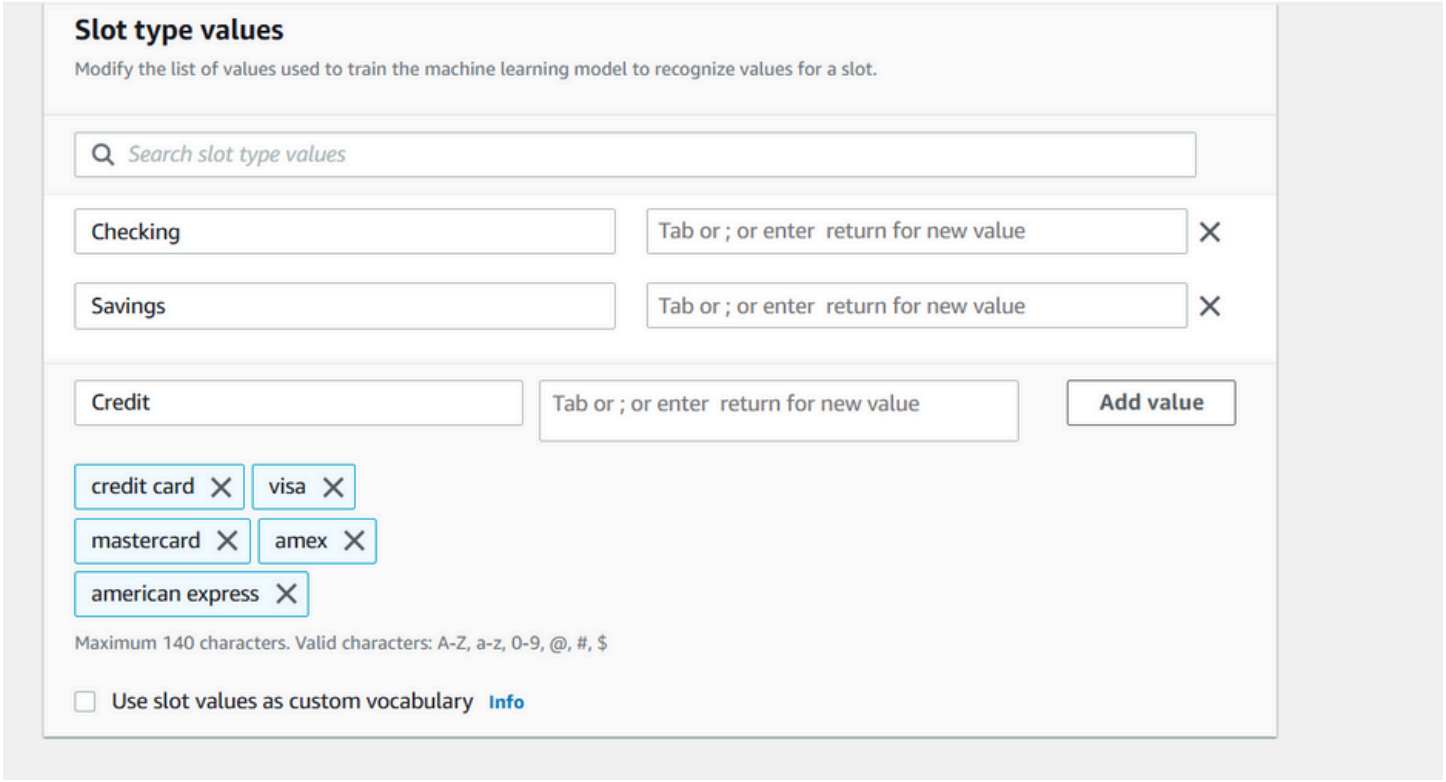
Response Generation - Chatbot will formulate a response based on the processed input and retrieved information from KB or DB .

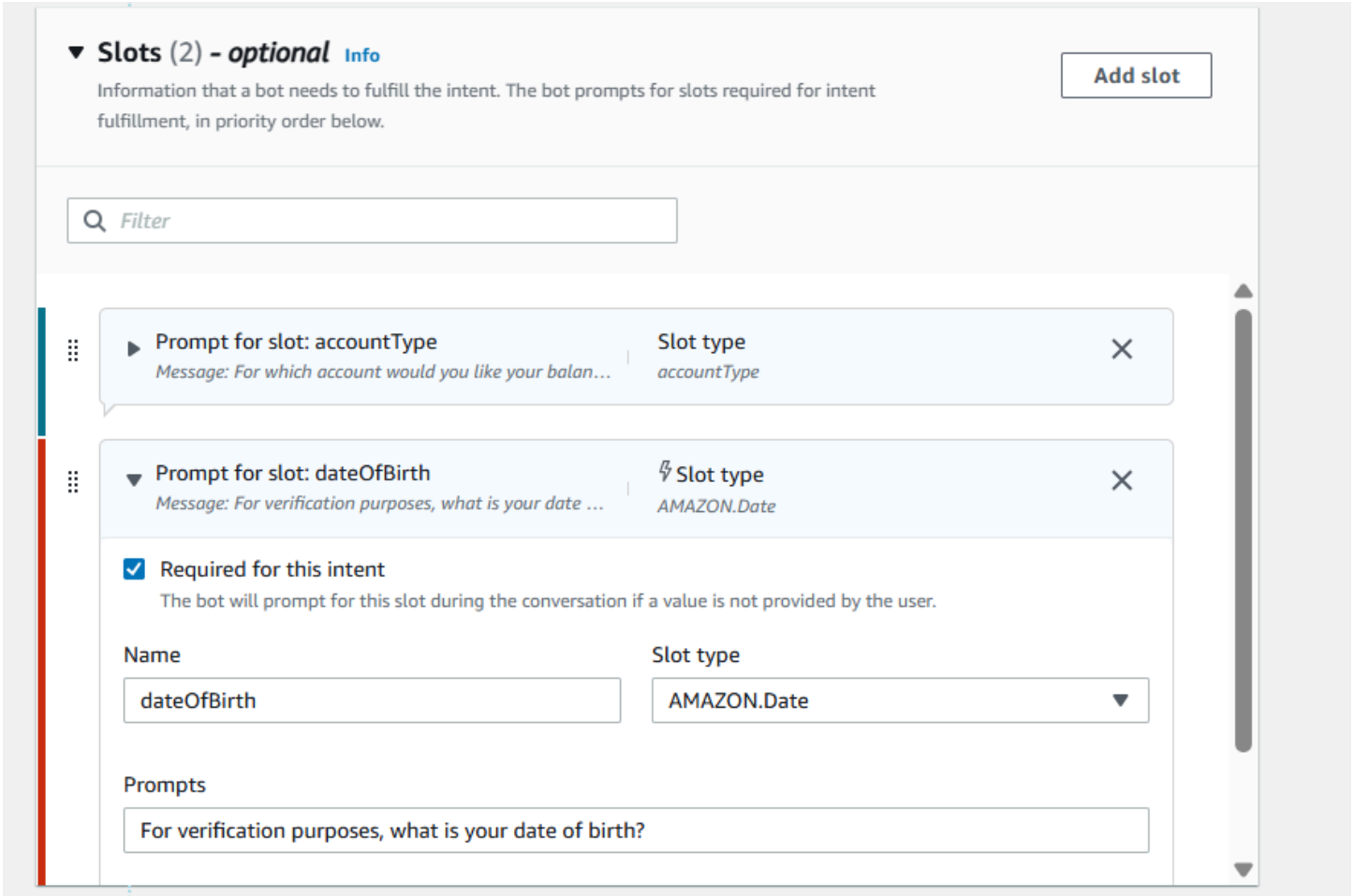
User Output - Chatbot will generates response to user in form of text or speech.

Setting up Chatbot

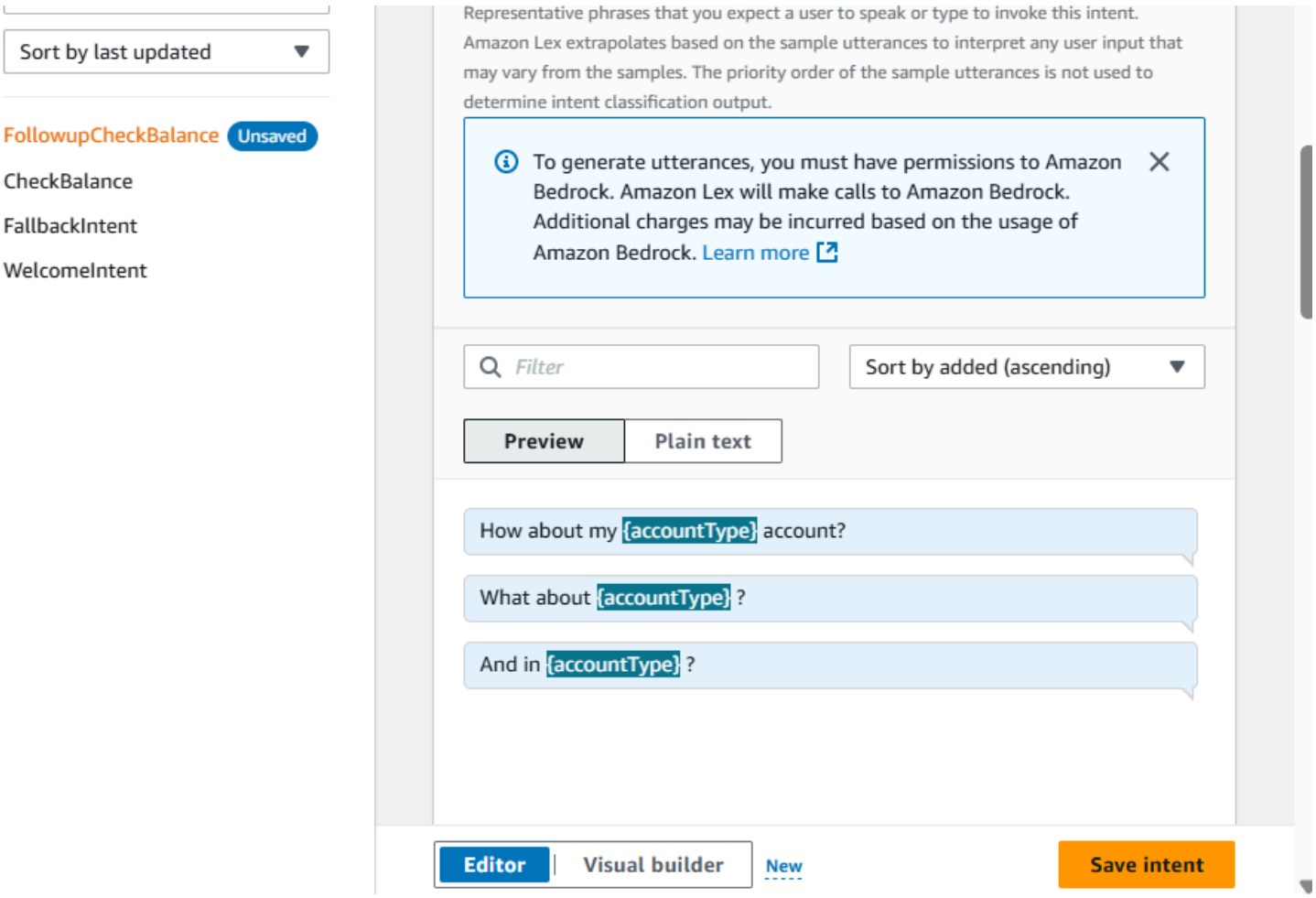


Slots

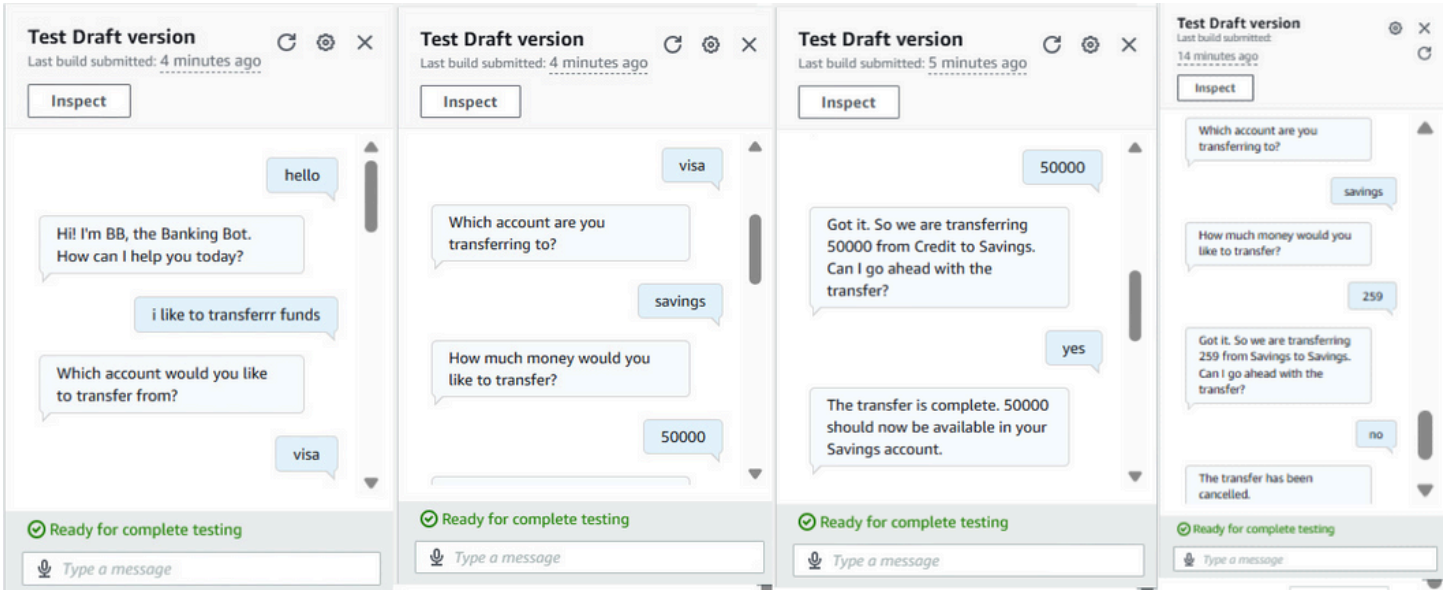

Slot type values
Modify the list of values used to train the machine learning model to recognize values for a slot.
Search slot type values
Checking
Tab or ; or enter return for new value
Savings
Tab or ; or enter return for new value
Credit
Tab or ; or enter return for new value
Add value
credit card
visa
mastercard
amex
american express
Maximum 140 characters. Valid characters: A-Z, a-z, 0-9, @, #, $
Use slot values as custom vocabulary Info


Slots (2) - optional Info
Information that a bot needs to fulfill the intent. The bot prompts for slots required for intent fulfillment, in priority order below.
Add slot
Filter
Prompt for slot: accountType
Message: For which account would you like your balan...
Slot type
accountType
Prompt for slot: dateOfBirth
Message: For verification purposes, what is your date ...
Slot type
AMAZON.Date
Required for this intent
The bot will prompt for this slot during the conversation if a value is not provided by the user.
Name
dateOfBirth
Slot type
AMAZON.Date
Prompts
For verification purposes, what is your date of birth?

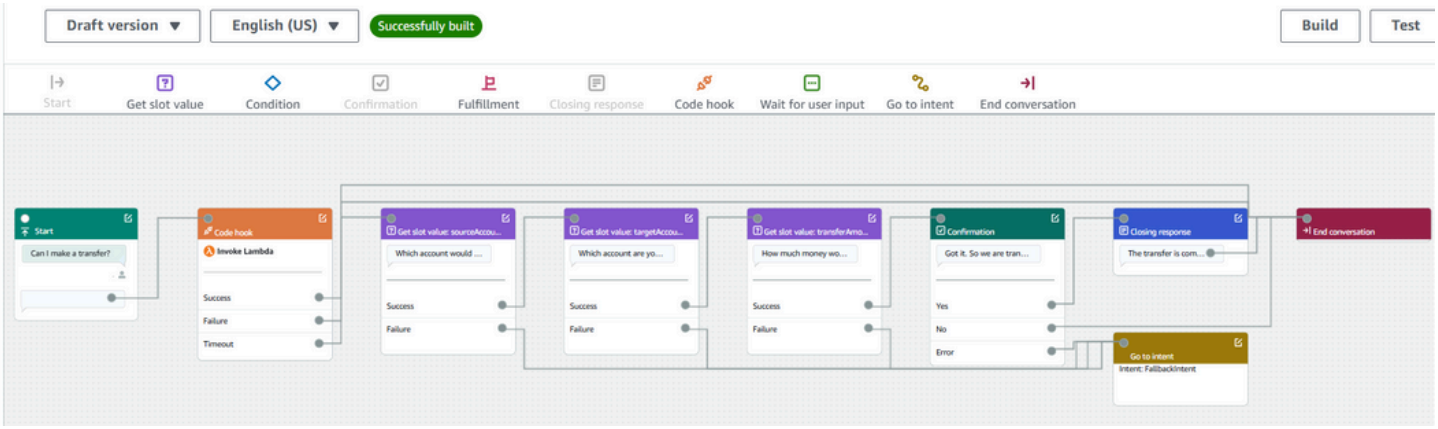
Intents



Banking Bot



Visual Representation flow of chatbot



Key Takeaways

- Building a chatbot was easy with Amazon Lex with help of built-in tools.
- Slot type and context tags are important for gathering inputs and helps in managing conversation flow of user with correct information.
- Managing the chatbot to handle the fallback intent and generating responses human - friendly.
- These chatbots can be used in industries like banking, customer support which automates common tasks and improves the user experience.

Conclusion

This project provides hands-on experience of creating a Banking Bot that can greet, check bank balance, Transfer funds and handle fallback which it doesn't understand. With Amazon Lex and Lambda creating serverless chatbots for interaction with users.

These tasks of creating intents, Slots, Utterances, tags can be automated with the help of Cloud Formation, where it is a Infrastructure as a Code(Iaac) where single file can be uploaded and can update, modify or delete the chatbot configurations as needed.