

PART – C

CREATING a RideRequest Bot

In the taxi and rental service chatbot, I decided to create a bot with the novel approach of creating only a single intent Ride Request which handles the slots based on the response of the user. Instead of creating two separate intents as Taxi and Self-Drive, I merged them into one and handled the responses with the help of conditional branching of AWS Lex.

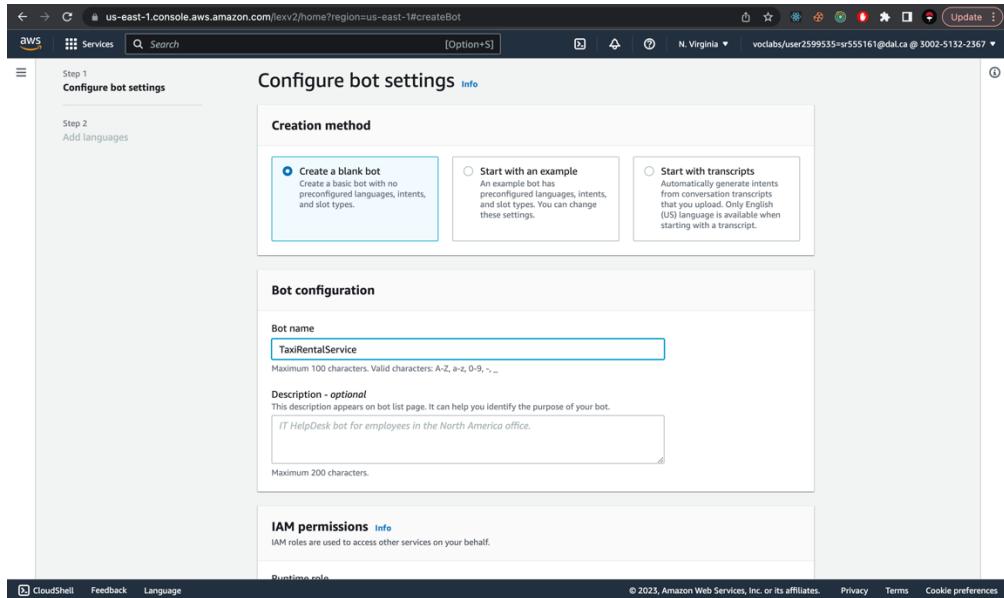


Fig 1: Creating a blank bot, TaxiRentalService

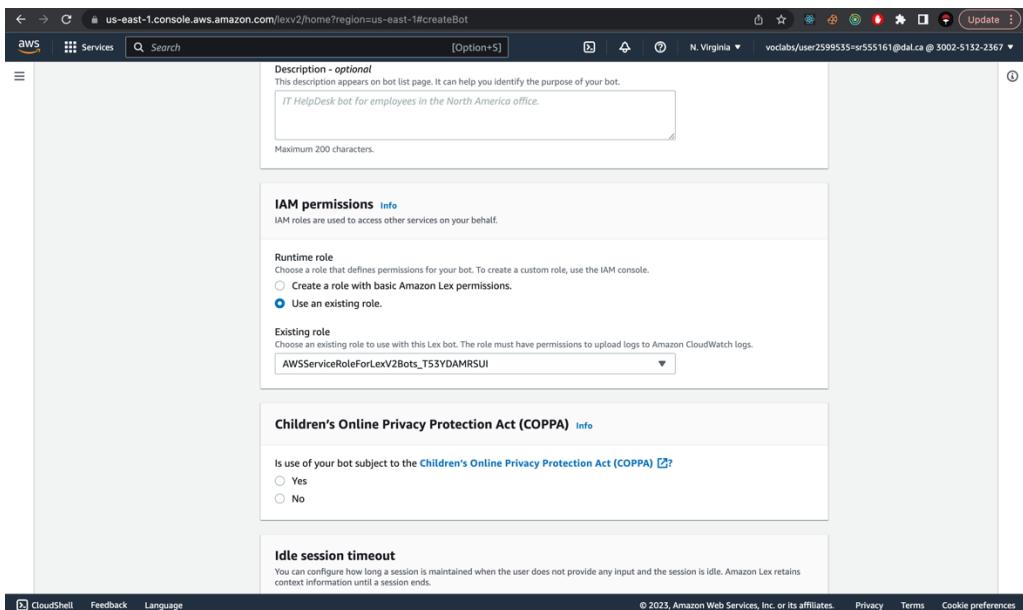


Fig 2: Use the existing role

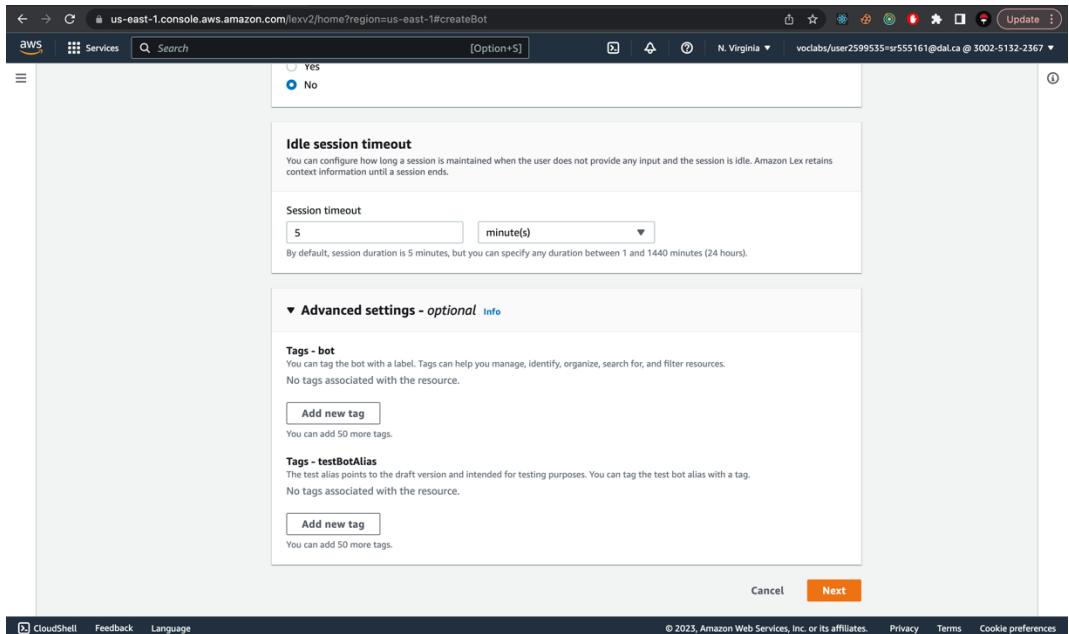


Fig 3: Go to next page

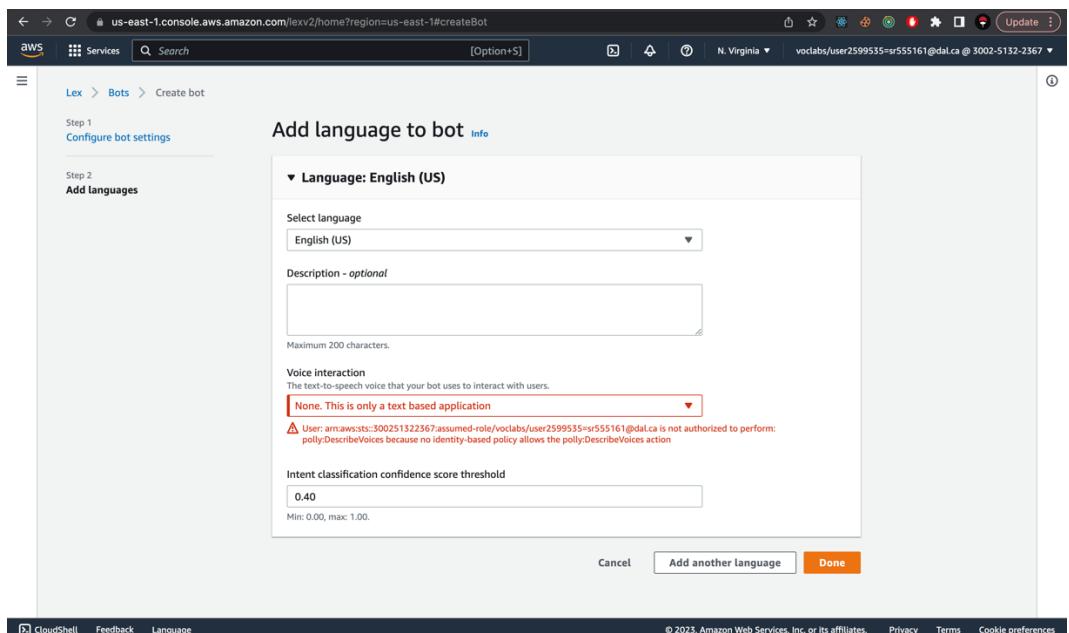


Fig 4: Select English as the language

The screenshot shows the AWS Lambda console interface. At the top, there's a search bar and a navigation bar with 'Services' selected. Below the navigation is a 'Create New Function' button. The main area is titled 'Lambda Functions' and shows a table with two rows: 'HelloWorld' and 'HelloWorld (1 version)'. The 'HelloWorld' row has a 'Edit' button. On the right side, there are tabs for 'Overview', 'Code', 'Logs', 'Metrics', and 'Actions'.

Fig 5: Creating an Intent

The screenshot shows the AWS Lex console interface. At the top, there's a search bar and a navigation bar with 'Services' selected. Below the navigation is a 'Create New Intent' button. The main area is titled 'Intents' and shows a table with two rows: 'RideRequest' and 'FallbackIntent'. The 'RideRequest' row has a 'Edit' button. On the right side, there are tabs for 'Overview', 'Details', and 'Test'.

Fig 6: Providing Sample Utterances

Fig 7: Providing initial response when user types utterance

Fig 8: Providing Slots for the user to gather the requirements

Now for the slot Service Type which takes either Taxi or Self-Drive, I created a condition branching by going into the advanced options. Condition is that if the ServiceType slot is chosen as Self-Drive by the user then it will skip the other slots and directly elicits a slot PickupTime.

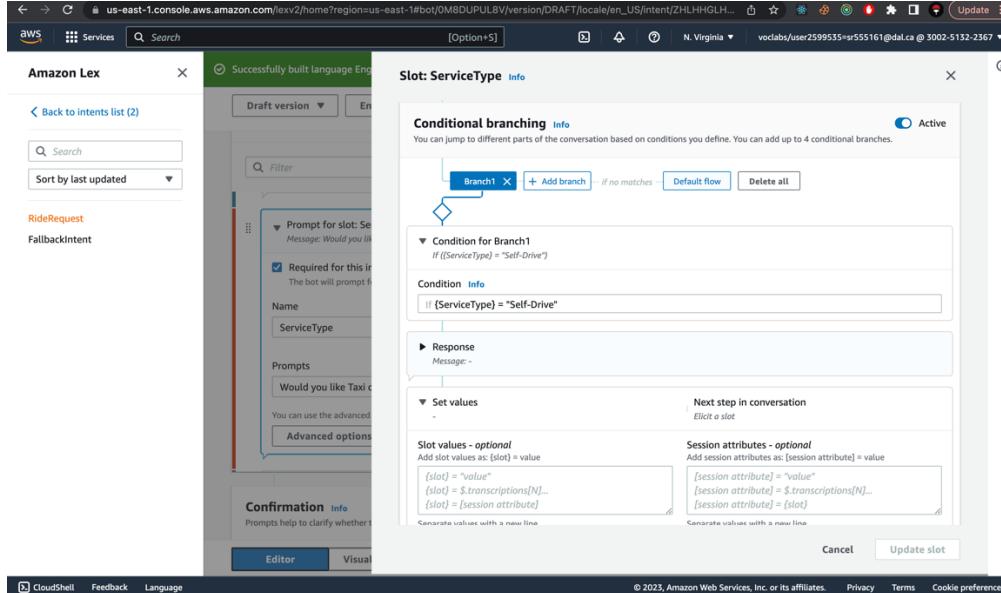


Fig 9: Providing a conditional branching of ServiceType

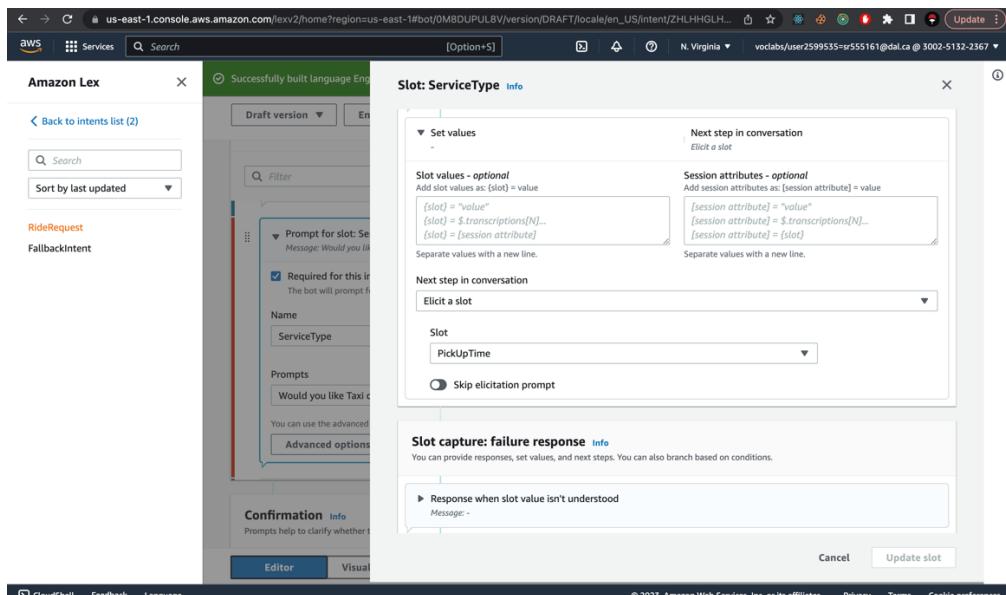


Fig 10: Specifying the next step in conversation to elicit a PickUpTime Slot

Slots (6) - optional

- Prompt for slot: Address Slot type AMAZON.AlphaNumeric
- Prompt for slot: PickUpDate Slot type AMAZON.Date
- Prompt for slot: PickUpTime Slot type AMAZON.Time
- Prompt for slot: ArrivalTime Slot type AMAZON.Time

Confirmation

Editor Visual builder New

Fig 11: Remaining Slots

Confirmation

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

Prompts to confirm the intent Message: Can I go ahead with your request?	Responses sent when the user declines the intent Message: Okay. Your request will not be submitted.
---	--

Confirmation prompt
What will the bot say to prompt the user to confirm this intent?
Can I go ahead with your request?

Decline response
What will the bot say if the user says NO to the confirmation prompt?
Okay. Your request will not be submitted.

Fulfillment

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

Editor Visual builder New

Fig 12: Confirming the intent before booking a ride.

Fulfillment message has been handled with the help of conditional branching for the self-drive and taxi.

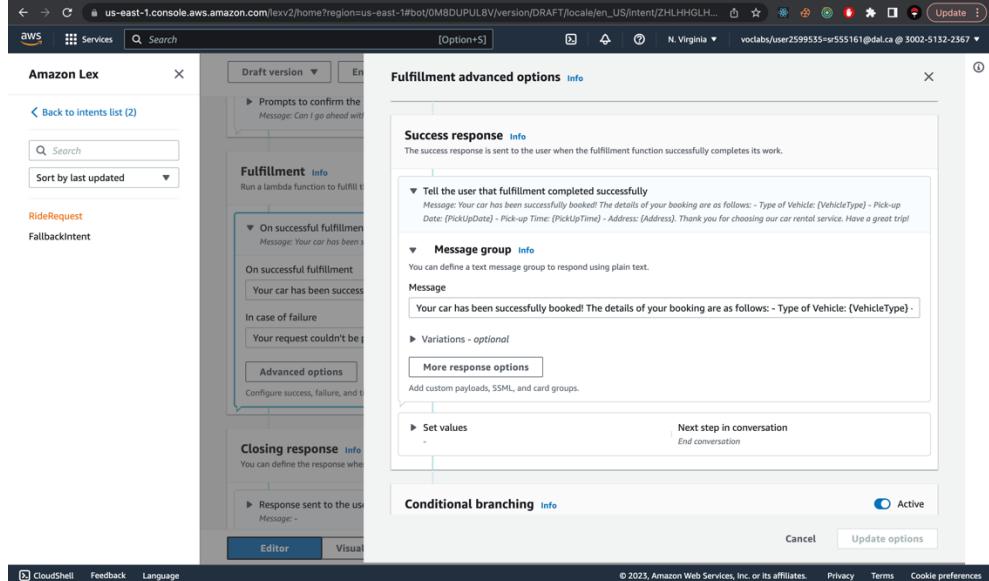


Fig 13: Fulfillment message when user books a taxi.

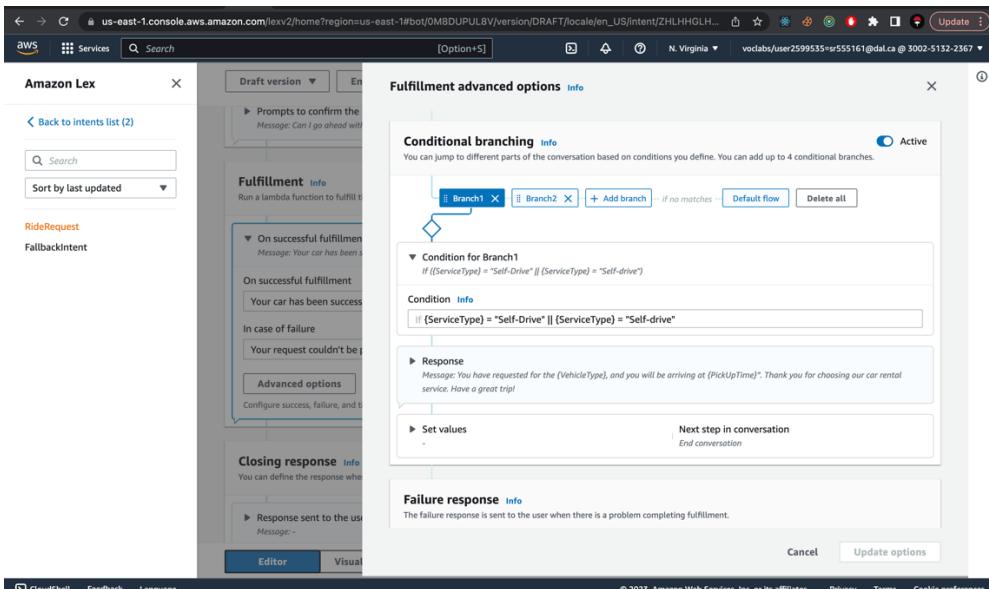


Fig 14: Fulfillment response set up for self-drive with the help of conditional branching.

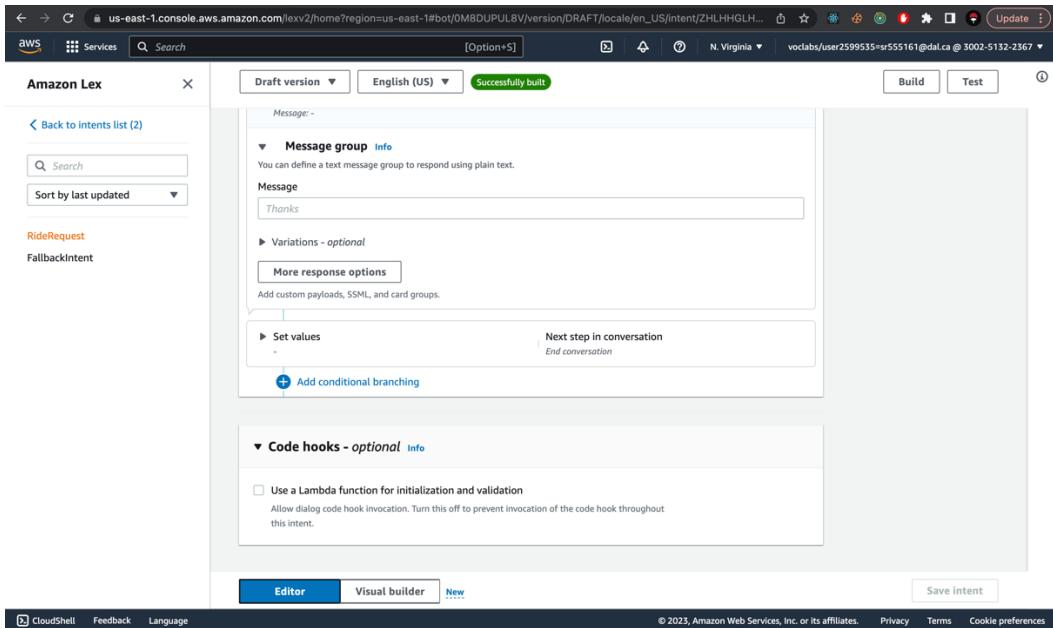


Fig 15: Empty Closing response.

CASE – 1: Book a Taxi

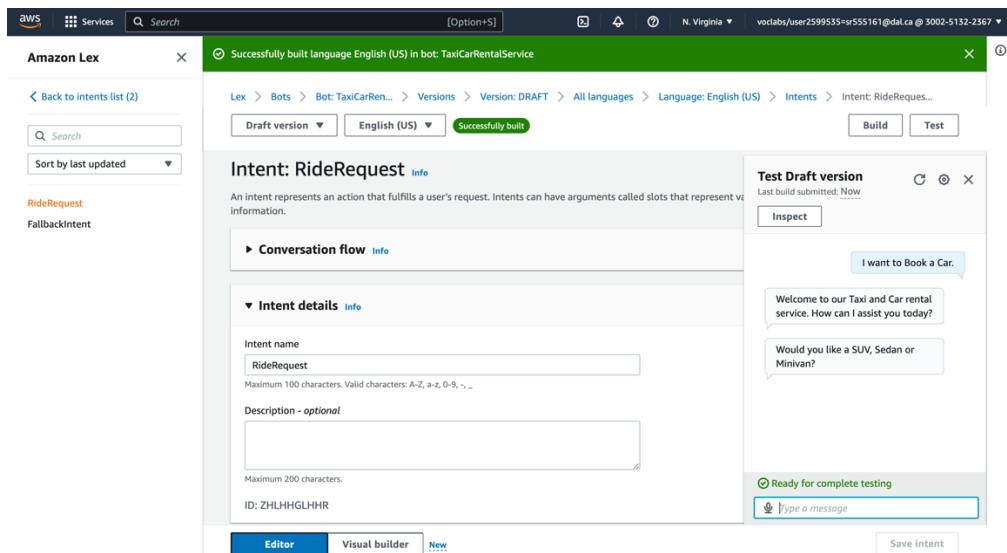


Fig 16: Sample Utterance

Intent: RideRequest

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

Conversation flow

Intent details

Intent name: RideRequest

Description - optional:

ID: ZHLHHGLHHR

Test Draft version

Last build submitted: 1 minute ago

Inspect

SUV

Would you like Taxi or Self-drive?

Taxi

What's your address?

Ready for complete testing

Type a message

Fig 17: Slot of type Address specific to Taxi.

Intent: RideRequest

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

Conversation flow

Intent details

Intent name: RideRequest

Description - optional:

ID: ZHLHHGLHHR

Test Draft version

Last build submitted: 1 minute ago

Inspect

What's your address?

505, Goldberg Building

At what date you will pick up the taxi?

today

At what time you will pick up the taxi?

Ready for complete testing

Type a message

Fig 18: Slot of type Date and Time specific to Taxi.

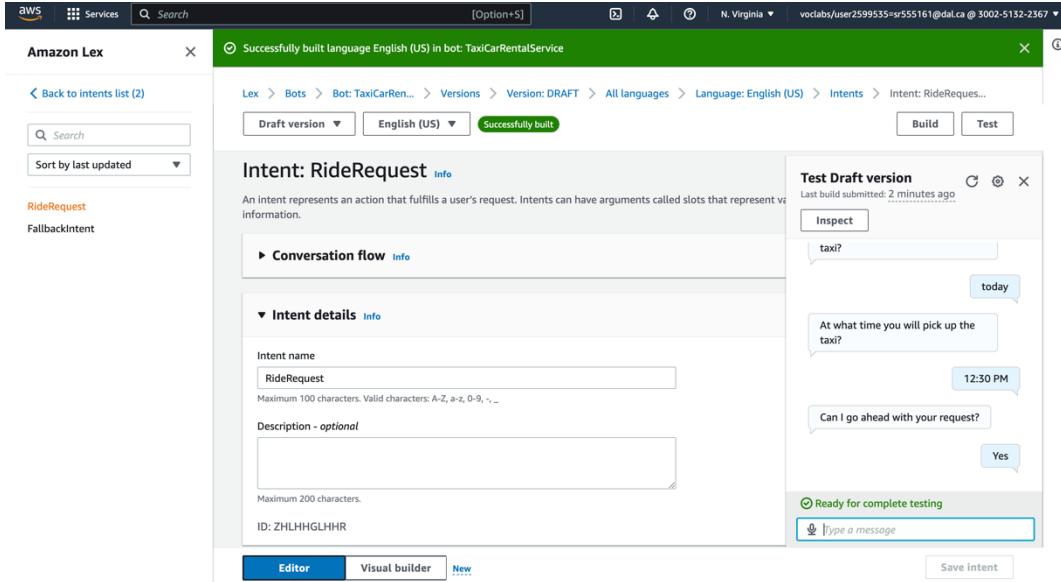


Fig 19: Confirmation Response before booking a ride.

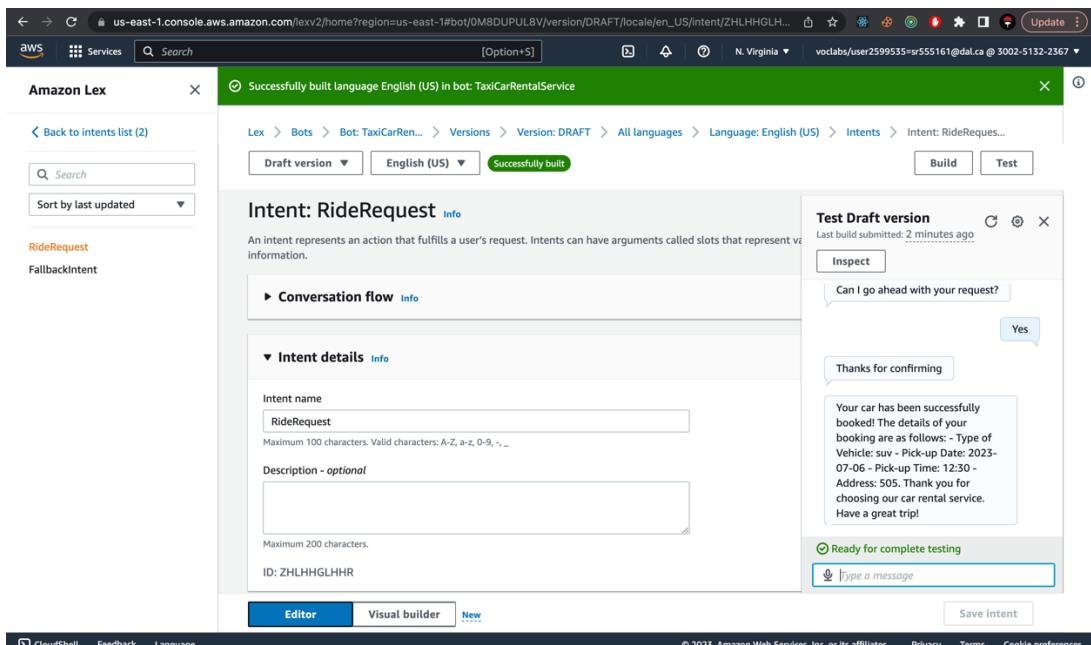


Fig 20: Success and closing response directly generated from the fulfillment

CASE – 2: Book a Self-Drive

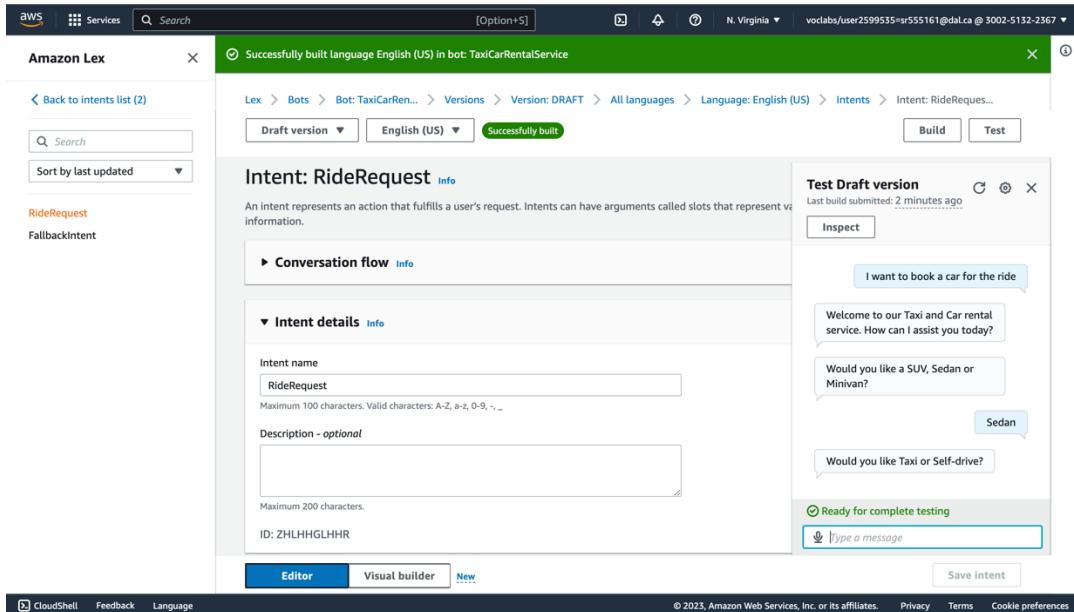


Fig 21: Utterance and Slot for Self-Drive

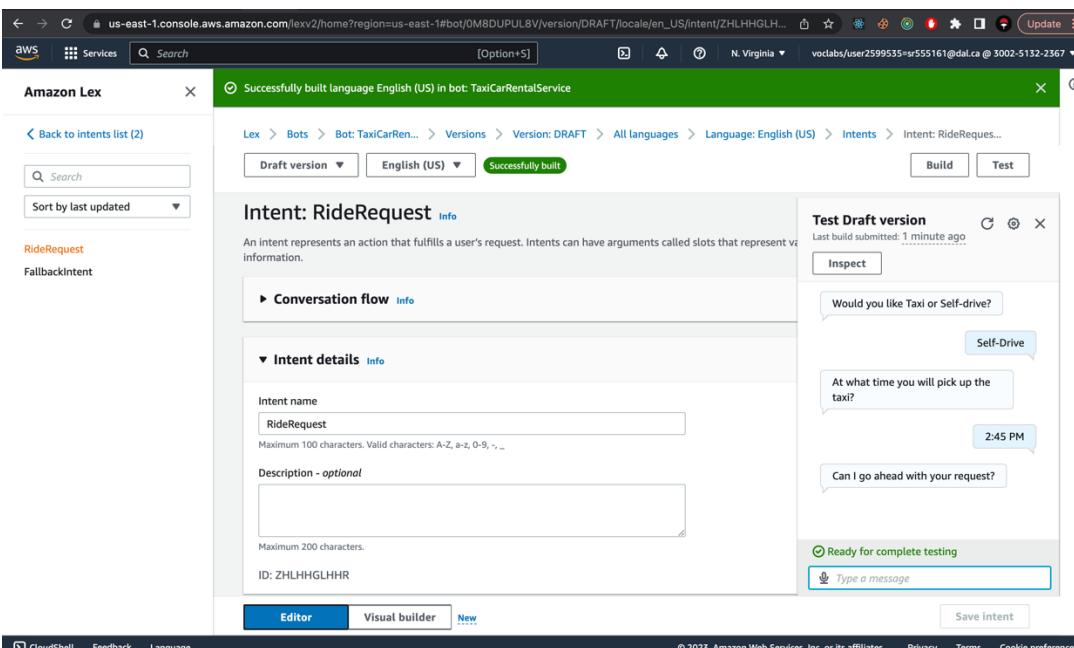


Fig 22: Confirmation for booking a self-drive car

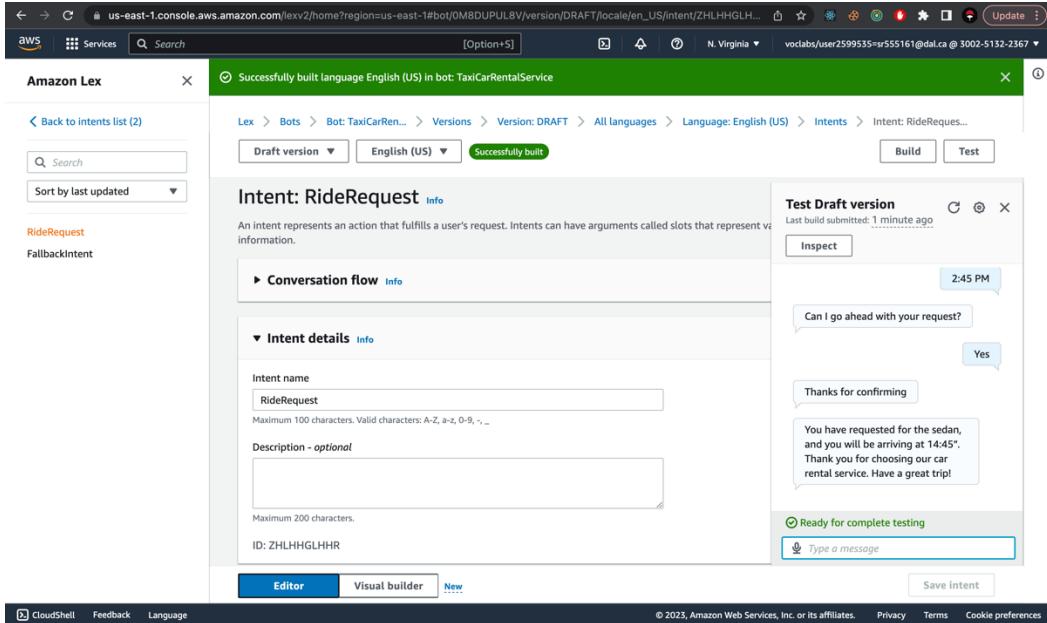


Fig 23: Success and closing response directly generated from the fulfillment

CASE – 3: Ride Cancelled

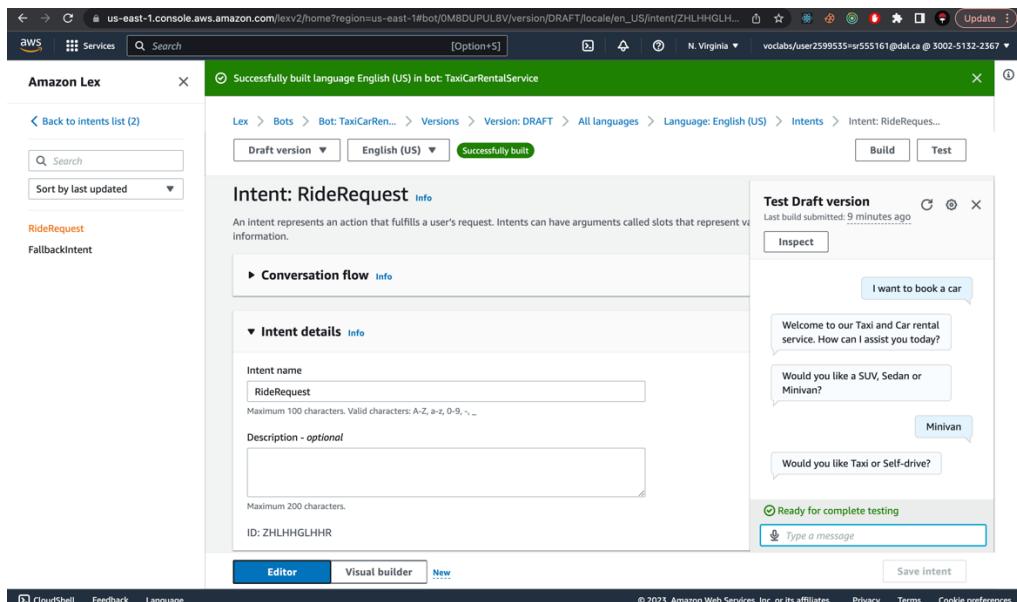


Fig 24: Utterance and Slots by the user.

The screenshot shows the Amazon Lex console interface. The top navigation bar includes 'Lex' > 'Bots' > 'Bot: TaxiCarRentalService' > 'Versions' > 'Version: DRAFT' > 'All languages' > 'Language: English (US)' > 'Intents' > 'Intent: RideRequest'. A green banner at the top right says 'Successfully built language English (US) in bot: TaxiCarRentalService'. Below this, the 'Intent: RideRequest' details are shown, including its name, description, and ID. To the right, a 'Test Draft version' window displays a conversation transcript:

- User: Would you like Taxi or Self-drive?
- Bot: Self-Drive
- User: At what time you will pick up the taxi?
- Bot: 9:00 AM
- User: Can I go ahead with your request?
- Bot: No

A green status bar at the bottom right indicates 'Ready for complete testing'.

Fig 25: Saying no to go ahead with the request to book a minivan.

This screenshot is nearly identical to Fig 25, showing the same intent details and test transcript. The main difference is the final message in the test transcript:

- User: Can I go ahead with your request?
- Bot: No
- User: Okay. Your request will not be submitted.

Fig 26: Fulfillment response by the lex.

