The aim of this document is to understand the implementation of Amazon Lex API and its usage in the project.

To be analyzed and discussed at a later stage,

**Amazon Lex:**

* Works with voice or text inputs
* Need to specify the conversational flow

**How does it work? Main Components:**

* **Bot:** A bot performs automated tasks such as ordering a pizza, booking a hotel, ordering flowers, and so on. An Amazon Lex bot is powered by Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) capabilities, the same technology that powers Amazon Alexa.

Amazon Lex bots can understand user input provided with text or speech and converse in natural language. You can create Lambda functions and add them as code hooks in your intent configuration to perform user data validation and fulfillment tasks.

* **Intent:** An intent represents an action that the user wants to perform. You create a bot to support one or more related intents. For example, you might create a bot that orders pizza and drinks. For each intent, you provide the following required information:
  + Intent name– A descriptive name for the intent. For example, OrderPizza.
  + Sample utterances – How a user might convey the intent. For example, a user might say "Can I order a pizza please" or "I want to order a pizza".
  + How to fulfill the intent – How you want to fulfill the intent after the user provides the necessary information (for example, place order with a local pizza shop). We recommend that you create a Lambda function to fulfill the intent.

You can optionally configure the intent so Amazon Lex simply returns the information back to the client application to do the necessary fulfillment.

In addition to custom intents such as ordering a pizza, Amazon Lex also provides built-in intents to quickly set up your bot.

* **Slot** – An intent can require zero or more slots or parameters. You add slots as part of the intent configuration. At runtime, Amazon Lex prompts the user for specific slot values. The user must provide values for all required slots before Amazon Lex can fulfill the intent.

For example, the OrderPizza intent requires slots such as pizza size, crust type, and number of pizzas. In the intent configuration, you add these slots. For each slot, you provide slot type and a prompt for Amazon Lex to send to the client to elicit data from the user. A user can reply with a slot value that includes additional words, such as "large pizza please" or "let's stick with small." Amazon Lex can still understand the intended slot value.

* **Slot type** – Each slot has a type. You can create your custom slot types or use built-in slot types. For example, you might create and use the following slot types for the OrderPizza intent:
  + Size – With enumeration values Small, Medium, and Large.
  + Crust – With enumeration values Thick and Thin.

**Model Building API Operations**

To programmatically create bots, intents, and slot types, use the model building API operations. You can also use the model building API to manage, update, and delete resources for your bot. The model building API operations include:

* PutBot, PutBotAlias, PutIntent, and PutSlotType to create and update bots, bot aliases, intents, and slot types, respectively.
* CreateBotVersion, CreateIntentVersion, and CreateSlotTypeVersion to create and publish versions of your bots, intents, and slot types, respectively.
* GetBot and GetBots to get a specific bot or a list of bots that you have created, respectively.
* GetIntent and GetIntents to get a specific intent or a list of intents that you have created, respectively.
* GetSlotType and GetSlotTypes to get a specific slot type or a list of slot types that you have created, respectively.
* GetBuiltinIntent, GetBuiltinIntents, and GetBuiltinSlotTypes to get an Amazon Lex built-in intent, a list of Amazon Lex built-in intents, or a list of built-in slot types that you can use in your bot, respectively.
* GetBotChannelAssociation and GetBotChannelAssociations to get an association between your bot and a messaging platform or a list of the associations between your bot and messaging platforms, respectively.
* DeleteBot, DeleteBotAlias, DeleteBotChannelAssociation, DeleteIntent, and DeleteSlotType to remove unneeded resources in your account.

You can use the model building API to create custom tools to manage your Amazon Lex resources. For example, there is a limit of 100 versions each for bots, intents, and slot types. You could use the model building API to build a tool that automatically deletes old versions when your bot nears the limit.