AWS Chatbot Challenge

Amazon Lex and AWS Lambda

Benjamin Towner

Cloud Solutions Architect / Engineer

[AWS Chatbot Challenge 2](#_Toc488037971)

[ChatBot Design Concept 2](#_Toc488037972)

[Overview 2](#_Toc488037973)

[Uniqueness 3](#_Toc488037974)

[Design Components 3](#_Toc488037975)

[Intents 3](#_Toc488037976)

[Slots 3](#_Toc488037977)

[Lambda 3](#_Toc488037978)

[Webhooks 3](#_Toc488037979)

[Entry Requirements 4](#_Toc488037980)

[Demo Video 4](#_Toc488037981)

[Github 4](#_Toc488037982)

[Access to Working Bot 4](#_Toc488037983)

[Testing Instructions 4](#_Toc488037984)

[Text Description 4](#_Toc488037985)

[Devpost Submission 4](#_Toc488037986)

[Appendix 4](#_Toc488037987)

[Eligibility 4](#_Toc488037988)

[Requirements 4](#_Toc488037989)

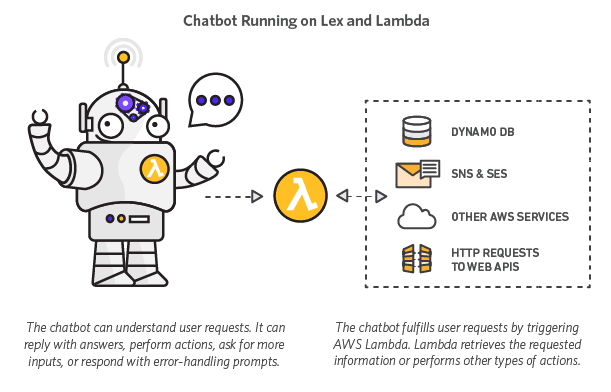
[How to enter 5](#_Toc488037990)

[Judging Criteria 5](#_Toc488037991)

# AWS Chatbot Challenge

Chatbots are changing how companies interface with their customers. With chatbots, you can easily fulfill the needs of your customers in an automated way using natural, human-like chat interfaces. Chatbots serve a variety of use cases, such as customer support, transaction fulfillment, data retrieval, or even DevOps functions (ChatOps).

However, building and running chatbots is a difficult task. First, most developers lack the deep learning expertise necessary to create bots that can intelligently interpret and respond to text. In addition, developers must also provision, manage, and scale the compute resources necessary to run the bot’s code.



What if you could build chatbots with sophisticated natural language processing and almost no operational overhead? [Amazon Lex](https://aws.amazon.com/lex/) is a fully managed service for building conversational interfaces into any application using voice and text. Lex is powered by the same deep learning technologies that power Amazon Alexa and lets you build natural language chatbots. Lex is integrated with [AWS Lambda](https://aws.amazon.com/lambda/), a service that lets you run code without provisioning or managing servers. Lambda enables you to write and run logic for your chatbot using serverless compute. Getting started with Amazon Lex and AWS Lambda is quick and easy.

# ChatBot Design Concept

## Overview

**MotherBot** helps by Sharing Calendars, Approved Contacts and provides a self-service administration platform for the *C.E.O. of the Home* to effectively schedule and organize. The AWS Chat Bot allows various processes to be made available to these household with the primary purpose of verifying the identity of the individual connections, formalizing an approval process and managing the ‘feasibility’ on the calendar.

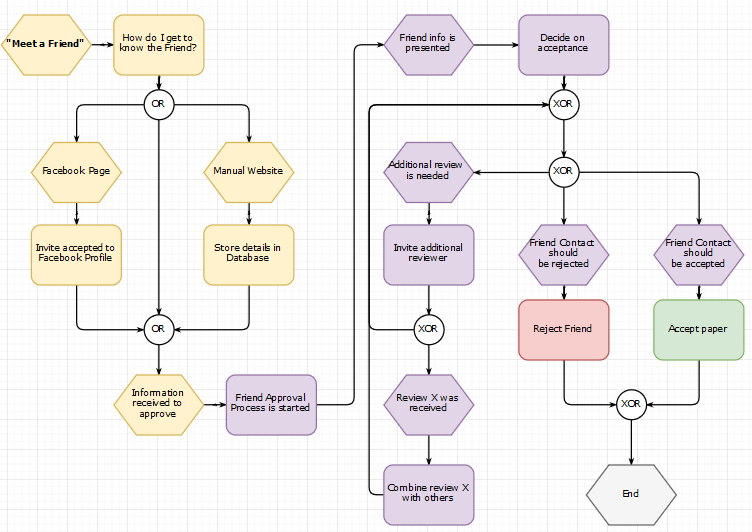
## Uniqueness

Households are like mini-organizations, except their members or ‘***little users***’ often find themselves in a place of constant challenge and response. Seeking approval and then finding out how to facilitate getting it done is the world that ‘***Tweens’*** live in. The coordination required among Parents often requires effective communication practices regardless of your Work-life or Marital status. Technology can enable Parents or Parental guardians to manage the household workloads while providing an appropriate level of privacy and respect.

# Design Components

There are 3 main functional aspects to **MotherBot**.

1. Identity and Access Management (Contact Management)
2. Activity Approval
3. Calendar Feasibility



## Intents

* Meet A Friend
* Can I Make A Call
* Can I GoTo

## Slots

## Lambda

## Webhooks

# Entry Requirements

1. A **demo video** that clearly shows your bot functioning on its intended platform.
2. **Share your code** repository publicly or privately through GitHub or BitBucket with testing@devpost.com.
3. **Access to a working bot**.
4. **Testing instructions** with anything we need to know to test your bot.
5. **Text description** - include a brief explanation of what the bot Application does and what makes it unique.
6. A **completed submission form** on Devpost.

## Demo Video

## Github

## Access to Working Bot

## Testing Instructions

## Text Description

This document will serve as the Text Description of MotherBot, the Challenge and the Submission.

## Devpost Submission

# Appendix

## **Eligibility**

This competition is open to the following:

* Individuals, and teams of individuals, who have reached the age of majority in their eligible jurisdiction of residence at the time of entry
* Organizations (employ 50 or fewer people) - must have been duly organized or incorporated and validly exist in an eligible area at the time of entry
* Large Organizations (employ more than 50 people) in eligible areas will be eligible for a non-cash recognition prize

The competition welcomes submitters from most countries around the globe. However, individuals or organizations may be disqualified if they are based in a nation, state, province, or territory where U.S. or local law prohibits participating in the competition or receiving a prize. This includes individuals, who are residents of, and organizations domiciled in Brazil, Quebec, Cuba, Sudan, Iran, North Korea, Syria and any other country designated by the United States Treasury's Office of Foreign Assets Control.

## Requirements

Main Requirement:

Build a conversational, natural language chatbot using Amazon Lex. Use Lex’s integration with AWS Lambda to execute logic on the backend, such as for fulfilling user intent or performing user data validation.

Your submission must be a new or existing bot (If existing, submitters must have updated their bot to run on Amazon Lex and AWS Lambda during the submission period).

Submission Recommendations:

Solutions can, but are not required to

* Be deployed to Slack, Facebook Messenger, or Twilio
* Use other AWS services
* Record or retrieve data from sources like Salesforce, HubSpot, Marketo, Microsoft Dynamics, Zendesk, and QuickBooks using Lex’s pre-built enterprise connectors.
* Incorporate speech capabilities using a service like Amazon Polly
* Leverage third-party APIs, SDKs, and services

## How to enter

1. Register for the AWS Chatbot Challenge.
2. Create an account on [AWS](https://portal.aws.amazon.com/gp/aws/developer/registration/index.html).
3. Visit the [Resources page](https://awschatbot2017.devpost.com/details/resources) for links to documentation and resources.
4. Shoot your demo video that demonstrates your bot in action. Prepare a written summary of your bot and what it does.
5. Provide a way to access your bot for judging and testing, including a link to your repo hosting the bot code and all deployment files and testing instructions needed for testing your bot. (The Github or BitBucket code repository may be public or private. If the repository is private, share access with [testing@devpost.com](mailto:testing@devpost.com)).
6. Submit your bot on AWSChatbot2017.Devpost.com before July 18, 2017 at 5pm PDT and share access to your bot, its repo and its deployment files.

## Judging Criteria

* **Customer Value**   
  The extent to which the bot provides value to your users. Does your bot help solve a problem or painpoint for your users?
* **Bot Quality**  
  Creativity and originality of the bot. Is your bot differentiated and does it solve your users' problems in a unique way?
* **Implementation of the Bot**  
  How well the bot was built and executed by the developer. Does the bot function as intended and does it recognize and respond to the most common phrases asked of it?