Capstone Project Weekly Progress Report

Project Title	MEALBUDDY
Group Name	GROUP_G
Student names/Student IDs	ELVIN IYPE MATHEW C0769974 ELDA VARGHESE C0769741 TOM JOSEPH C0760915 JEENA HELEN FRANCIS C0764493 CHINJU BABY C0769912
Reporting Week	19 JULY 2020 - 25 JULY 2020
Faculty Supervisor	WILLIAM POURMAJIDI

1. Tasks Outlined in Previous Weekly Progress Report

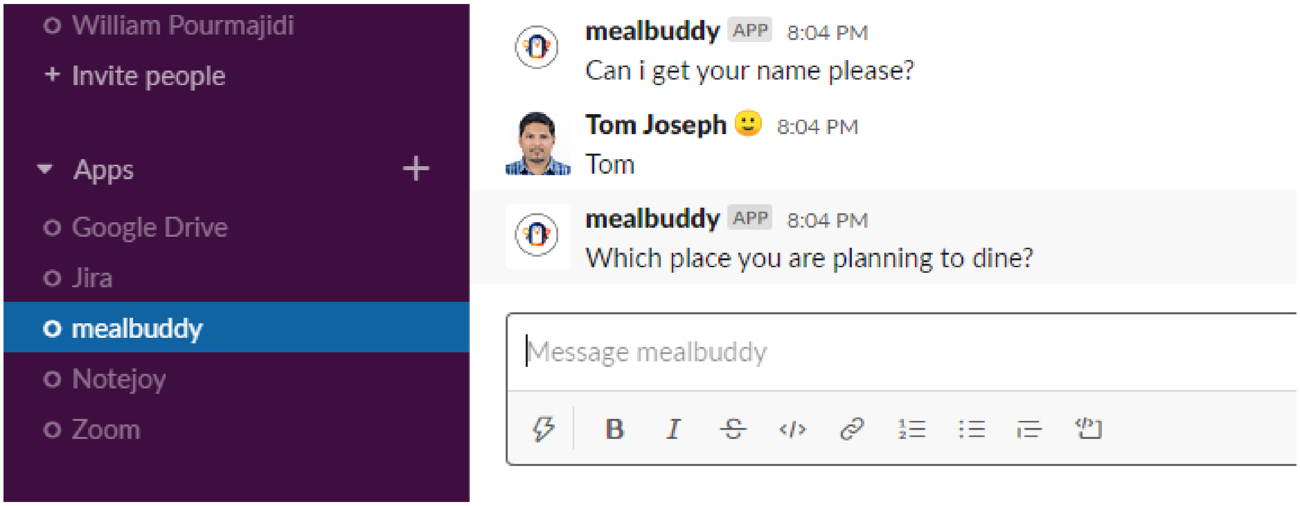
- **Slack integration study**
- **Slack integration**
- **Whole system Test Scenarios**
- **Machine Learning study**

2. Progress Made in Reporting Week

- **Slack is a proprietary business communication platform developed by American software company Slack Technologies. Slack offers many IRC-style features, including persistent chat rooms organized by topic, private groups, and direct messaging.**
- **Step 1: Create an Amazon Lex Bot**
- **Step 2: Sign Up for Slack and Create a Slack Team - Sign up for a Slack account and create a Slack team.**
- **Step 3: Create a Slack Application**
    a. **Create a Slack application on the Slack API Console**
    b. **Configure the application to add interactive messaging to your bot: you will get application credentials (Client Id, Client Secret, and Verification Token).**
- **Step 4: Integrate the Slack Application with the Amazon Lex Bot**

a. Sign in to the AWS Management Console, and open the Amazon Lex console at https://console.aws.amazon.com/lex/.
b. Choose the Amazon Lex bot that you created in Step 1.
c. Choose the Channels tab.
d. In the left menu, choose Slack.
e. On the Slack page, provide the following:
f. Type a name. For example, BotSlackIntegration.
g. Choose "aws/lex" from the KMS key drop-down.
h. For Alias, choose the bot alias.
i. Type the Client Id, Client secret, and Verification Token, which you recorded in the preceding step. These are the credentials of the Slack application.
j. Choose Activate. The console creates the bot channel association and returns two URLs (Postback URL and OAuth URL). Record them. In the next section, you update your Slack application configuration to use these endpoints as follows:
k. The Postback URL is the Amazon Lex bot's endpoint that listens to Slack events. You use this URL:
    - As the request URL in the Event Subscriptions feature of the Slack application.
    - To replace the placeholder value for the request URL in the Interactive Messages feature of the Slack application.
l. The OAuth URL is your Amazon Lex bot's endpoint for an OAuth handshake with Slack.

- Step 5: Complete Slack Integration
    a. Update the OAuth & Permissions feature as follows:
        - In the left menu, choose OAuth & Permissions.
        - In the Redirect URLs section, add the OAuth URL that Amazon Lex provided in the preceding step. Choose Add a new Redirect URL, and then choose Save URLs.
        - In the Bot Token Scopes section, add two permissions with the Add an OAuth Scope button. Filter the list with the following text:
            - chat:write
            - team:read
    b. Update the Interactivity & Shortcuts feature by updating the Request URL value to the Postback URL that Amazon Lex provided in the preceding step. Enter the postback URL that you saved in step 4, and then choose Save Changes.
    c. Subscribe to the Event Subscriptions feature as follows:
        - Enable events by choosing the On option.
        - Set the Request URL value to the Postback URL that Amazon Lex provided in the preceding step.
        - In the Subscribe to Bot Events section, subscribe to the message.im bot event to enable direct messaging between the end user and the Slack bot.
        - Save the changes.
- Step 6: Test the Integration
    a. Choose Manage Distribution under Settings. Choose Add to Slack to install the application. Authorize the bot to respond to messages.
    b. You are redirected to your Slack team. In the left menu, in the Direct Messages section, choose your bot. If you don't see your bot, choose the plus icon (+) next to Direct Messages to search for it.
    c. Engage in a chat with your Slack application, which is linked to the Amazon Lex bot. Your bot now responds to messages.

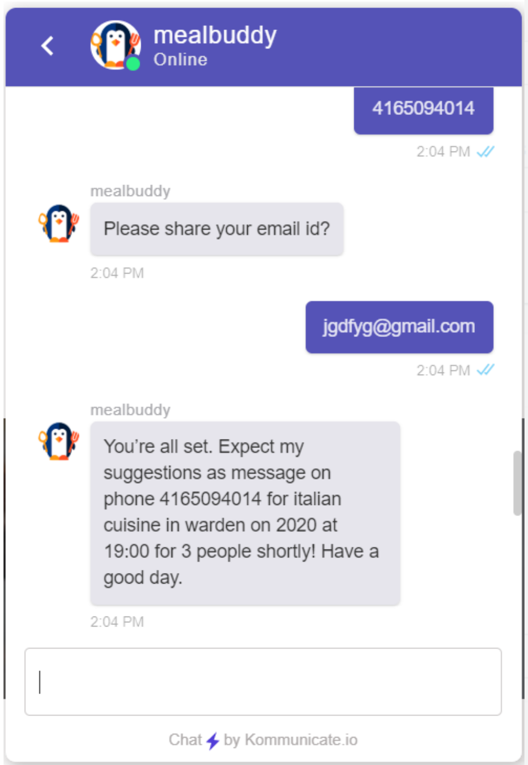


mealbuddy APP 8:04 PM
Can i get your name please?

Tom Joseph 🙂 8:04 PM
Tom

mealbuddy APP 8:04 PM
Which place you are planning to dine?

- whole system tested from **UI - LEX - LF1 - SQS- LF2**

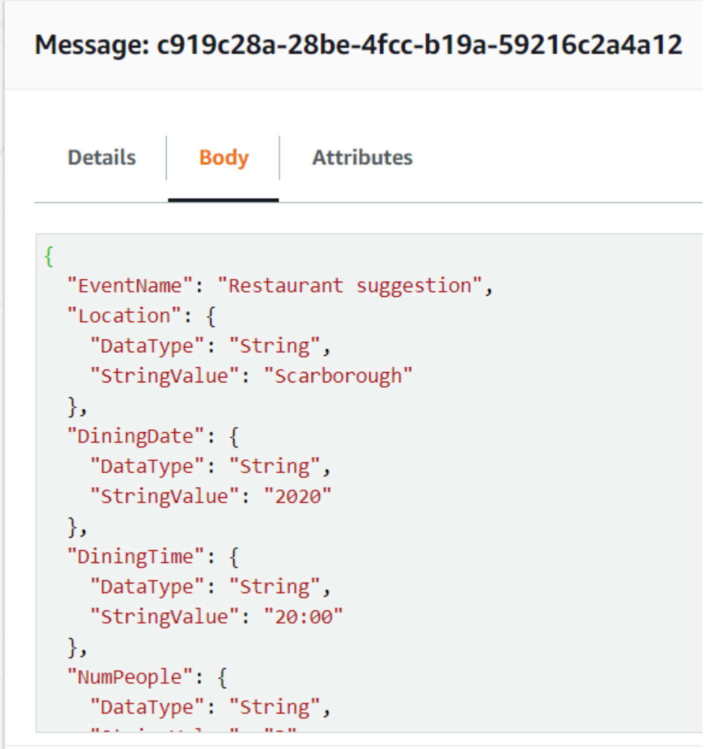


4165094014
2:04 PM

mealbuddy
Please share your email id?
2:04 PM

jgdfyg@gmail.com
2:04 PM

mealbuddy
You're all set. Expect my suggestions as message on phone 4165094014 for italian cuisine in warden on 2020 at 19:00 for 3 people shortly! Have a good day.
2:04 PM

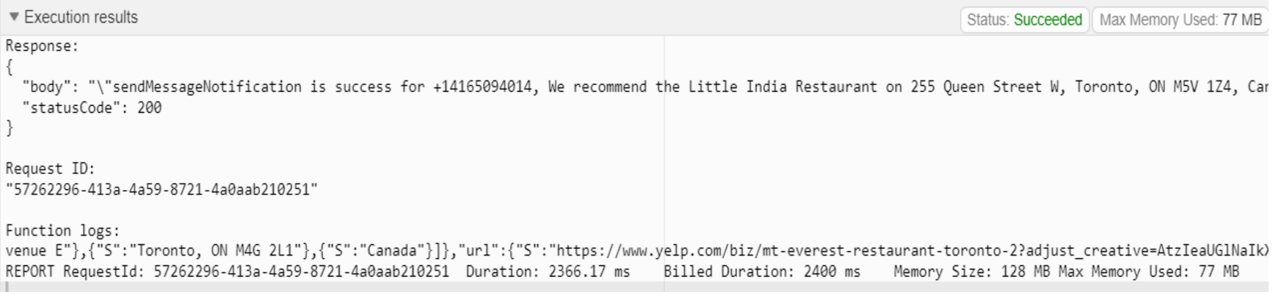


Message: c919c28a-28be-4fcc-b19a-59216c2a4a12

```
{
  "EventName": "Restaurant suggestion",
  "Location": {
    "DataType": "String",
    "StringValue": "Scarborough"
  },
  "DiningDate": {
    "DataType": "String",
    "StringValue": "2020"
  },
  "DiningTime": {
    "DataType": "String",
    "StringValue": "20:00"
  },
  "NumPeople": {
    "DataType": "String",
```

FRONTEND UI **SQS**

LF2

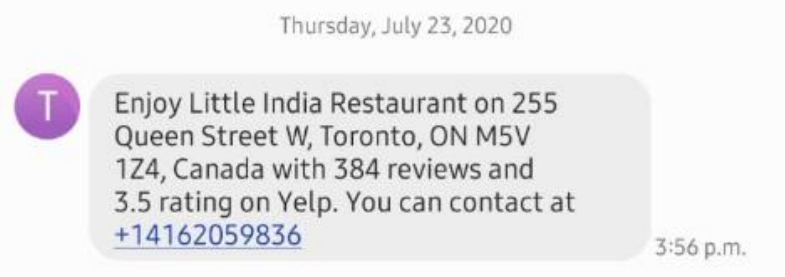


```
Response:
{
  "body": "\"sendMessageNotification is success for +14165094014, We recommend the Little India Restaurant on 255 Queen Street W, Toronto, ON M5V 1Z4, Car
  "statusCode": 200
}

Request ID:
"57262296-413a-4a59-8721-4a0aab210251"

Function logs:
venue E"},{"S":"Toronto, ON M4G 2L1"},{"S":"Canada"}]},"url":{"S":"https://www.yelp.com/biz/mt-everest-restaurant-toronto-2?adjust_creative=AtzIeaUGlNaIk
REPORT RequestId: 57262296-413a-4a59-8721-4a0aab210251  Duration: 2366.17 ms    Billed Duration: 2400 ms    Memory Size: 128 MB Max Memory Used: 77 MB
```

User received message from SNS



Enjoy Little India Restaurant on 255 Queen Street W, Toronto, ON M5V 1Z4, Canada with 384 reviews and 3.5 rating on Yelp. You can contact at +14162059836

- **Amazon SageMaker is a fully managed service that provides every developer and data scientist with the ability to build, train, and deploy machine learning (ML) models quickly. SageMaker removes the heavy lifting from each step of the machine learning process to make it easier to develop high quality models.**

- **We studied how to make restaurant recommendation systems based on semi-supervised learning.**
- **For that , we fetched the details of restaurants [id, cuisine, rating and review] from yelp fusion API.**
- **Then we are storing the data into a csv file for further use. From there by using machine learning techniques we created different models to train and test them. The different machine learning models will be evaluated to find the best model which will give the best restaurant with its rating and reviews**
- **Now we are going through different ML algorithms (Decision tree, Random forest , xgboost algorithm) to understand which is better to choose.**

3. Difficulties Encountered in Reporting Week

- **Luckily there weren't any challenges while integrating with slack. Amazon Cloud Services makes fairly good documentation for integration and makes it seem less work.**
- **Amazon sage maker is fairly new technology and its study is a bit of a task. Took time to understand the notebook instances to prepare and process data and to train and deploy machine learning models.**

4. Tasks to Be Completed in Next Week

- **Final Report work start**
- **Machine Learning based recommendation functionality**
- **Presentation work start**
- **Testing scenarios**