

# CSCI 5410 -Project Feasibility Report

Team: **SApp\_19**

(1) Hashik Donthineni - B00868314

(2) Fenil Nikeshkumar Shah - B00871894

(3) Pathik Kumar Patel – B00869765

(4) Dharaben Thakorbhai Gohil -B00884960

## Overview

We will be creating a Food Delivery System using serverless components by using the concept of a “Cloud Data plumbing system” to process the data. The application will have three user roles: Admin, Customer, Restaurant management. The customers can place the order, track the orders, and provide ratings & feedbacks to the restaurants, and can use the discount coupons offered by the restaurants. Also, they can interact with the chatbot to navigate through the application and, they can chat in real-time with the restaurant representatives and even the managers in case of escalation.

For the restaurants’ management team, they can add/modify the menu and add discount coupons, check the similarity score for their recipes, recommend food items to the customers and keep track of their revenue using the reporting and visualization module.

The admin can keep the track of the cloud resource usages and user statistics of the application.

## List of Features

- User sign-up and sign-in
  - Users can sign up as a restaurant or as a customer.
  - Sign-up, sign-in validation is done using Google SSO.
  - **AWS Cognito** is used for sign-up, sign-in, and user management.
  - **Technologies used:** React(front-end), AWS Cognito, Google SSO.
- Multifactor authentication
  - Users can use app based TOTP for multifactor authentication.
  - MFA of a user is done with TOTP from **AWS Cognito**
  - **Technologies used:** React(front-end), AWS Cognito
- Discounts and coupons
  - While checking out users will be able to apply coupon codes to save money.
  - Coupons will be stored in a database and the codes will be validated by AWS Lambda.
  - **Technologies used:** React(front-end), AWS RDS database, NodeJS on Lambda.
- Browse the restaurants and place the food order.
  - The customer will browse the restaurants and food items from the menu and after that, they can place the food order simply by selecting the items of their choice and the quantity of the items.
  - **Technologies used:** The **React** framework will render the restaurant and food details. The order details of the customer will be stored in the **AWS RDS database**.
- Chatbot for application navigation
  - The User can ask any queries about application navigation and tracking the other features. Users can communicate with chatbots.

- **Technologies used:** **AWS Lex** is useful to create and build conversational chatbots quickly. For that one must specify the conversational flow to create a bot.
- Customer Support using instant messaging.
  - The customers would be able to contact the restaurant customer service representative in case of any query related to the service or inquiry.
  - **Technologies used:** The **Firebase Cloud Messaging** service will be used in the backend to provide the instant messaging service.
- Escalation to the restaurant management.
  - In case the customer query is not solved by the restaurant customer representative and the customer needs additional support then the customer query will be forwarded to restaurant management and the customer can have a chat with restaurant management.
  - **Technologies used:** The **Firebase Cloud Messaging service** will be used in the backend to provide the instant messaging service.
- Recipe Similarity score.
  - Each restaurant can upload its famous recipe on the app. The app scans the recipe, gives a tag, and shows the similarity score with other recipes in the app.
  - **Technology Used:** An NLP-based machine learning model trained using **GCP AutoML** service will be used to check the similarity score of the recipe uploaded.
- Tracking the order delivery:
  - Customers will be provided to track their order delivery status right from placing the order till it is delivered.
  - **Technologies used:** For this, the delivery status will be updated through API and the status of the delivery will be maintained into the **AWS RDS MySQL** database.
- Provide Feedback & Ratings
  - Customers will be allowed to provide ratings and feedback on the food items.
  - **Technologies used:** The ratings and customer feedback would be stored and managed into the AWS RDS database and will be fed to the data processing module which uses **Amazon QuickSight** service to create a word cloud based on the data.
- Customer recommendations:
  - In this feature, the User will receive recommendations from the application about the new features, new coupons, new recipes included in the menu, popular recipes of the restaurants, and so on.
  - **Technologies used:** GCP AutoML is used to implement this feature.
- Menu:
  - Users can go through the menu of the restaurants. In the menu, one can see the dish's name and its price.

- **Technologies used:** For Menu, the AWS RDS SQL instance will be used.
- Visualizations
  - Customers can see visualizations of their spending based on the food item categories and restaurants.
  - **Technologies used:** For the visualizations, we'll be using **Google Data Studio** and the data will be fed from the **AWS RDS SQL** instance.
- Customer Feedback Polarity:
  - Based on the feedback on the food items provided by the customers, the application will calculate the polarity of the feedback and the same would be displayed to the restaurant managers.
  - **Technologies used:** For calculating the polarity, we'll use **AWS Comprehend**.
- Food-Cloud
  - Based on the ratings provided by the customers on food items, the data will be fed to a service that will create a word cloud based on the ratings, and from it, the restaurant owners can get the cumulative ratings provided by the customers on various food items.
  - **Technologies used:** **Amazon QuickSight** will be used for this, and the service will be deployed and managed on Amazon ECS.
- Report Generation Module
  - System Admin will be provided with the functionality to create the reports related to the usage of the services, user statics, and billing information.
  - For the above factors, the visualizations will be provided to the admin.
  - **Technologies used:** For this feature, services such as **Google data studio** and **AWS Cost Explorer** will be used.

## Flowchart

### Customer Module

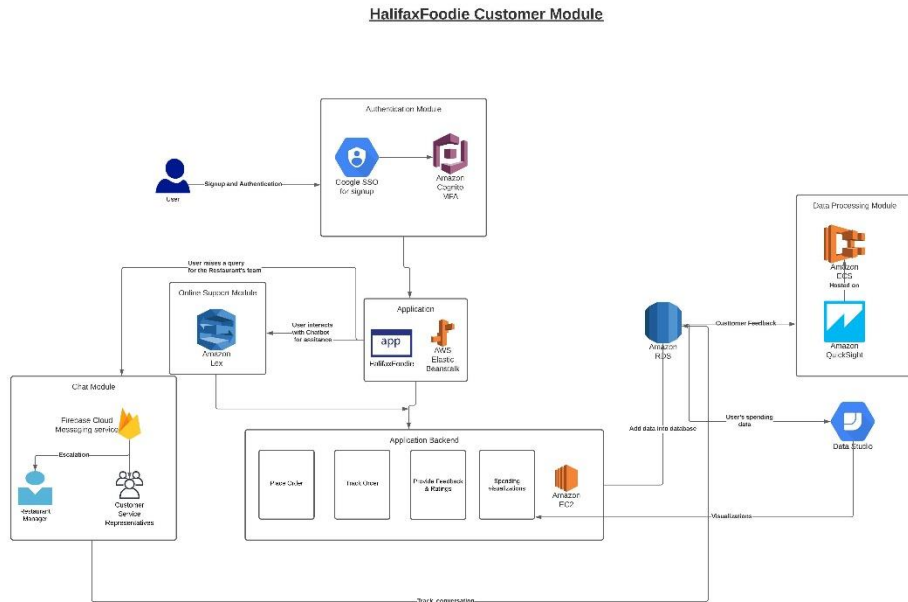


Figure 1: Flowchart for Customer module

### Restaurant Module

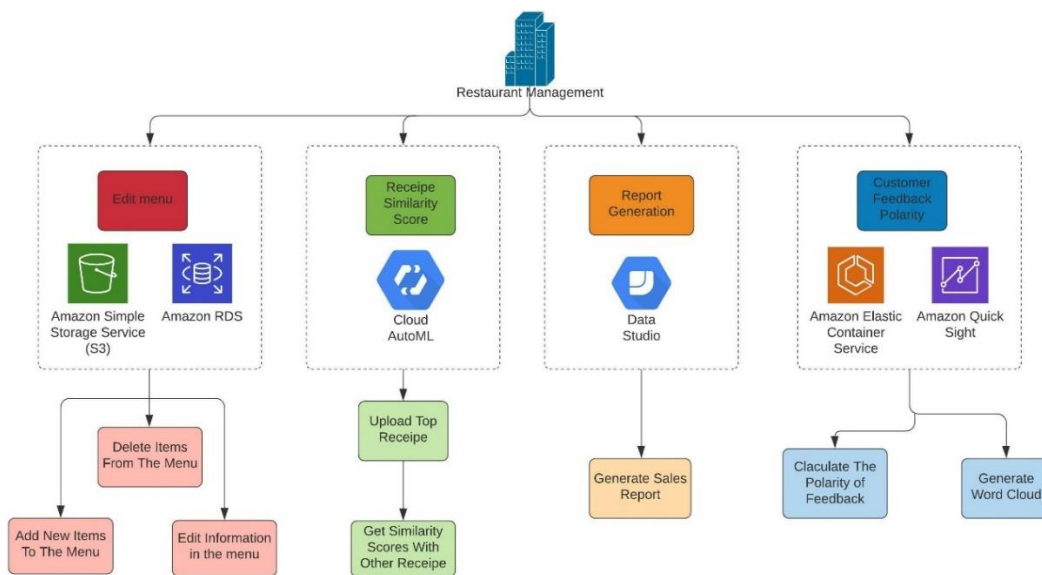


Figure 2: Flowchart for Restaurant module

## Services and their strengths, shortcomings

### AWS Cognito

AWS Cognito provides robust user authentication and management mechanism with many third-party authentications like Google, Facebook, Apple, etc. Additionally, it provides multi-factor authentication (MFA) to secure users' accounts [1].

**Shortcomings:** It doesn't seem to be relatively well documented. Can be harder to build and very few well-known companies use this.

### AWS Lex

AWS Lex is an excellent tool for building chatbots and a top contender for top chat-bots providers besides Dialogflow. SDK support and integration are great with Lex.

**Shortcomings:** Utterances and entity mapping is critical and it's hard to create data sets.[2]

### GCP AutoML

AutoML allows us to worry less about the inner working of machine learning and simply concentrate on the data we are providing essentially shortening the length of development time for us.

### Amazon QuickSight

Amazon Quicksight is a business intelligence tool that is quite useful to visualize data in different forms it is easy to set up and point to a data source and it accepts data from both AWS sources as well as other non-AWS sources. And it has attractive and cheap cost to pay. We are using it for the Wordcloud feature it offers.

**Shortcomings:** It's not properly mobile optimized. The customizations or the way of visualizing the data is limited as compared to the competitors.

### AWS RDS MySQL

AWS RDS isn't particularly a standing product among competitors but that doesn't mean it's bad it just means they are all on the same level, they offer serverless SQL. It's secured, fast, reliable, and periodic back-ups can be taken and, it's elastic.

### Google Data Studio

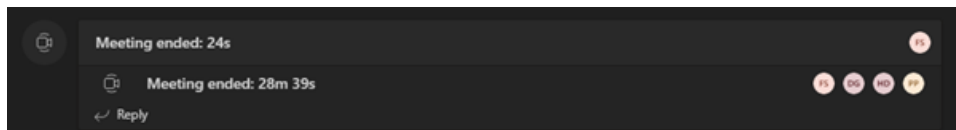
Another one of the best tools for BI it's used in our application to visualize data for our administrators. Visualize and if needed query the data. We can use **AWS Cost Explorer** API to visualize and understand the cost of maintenance.

## Project anticipated timeline:

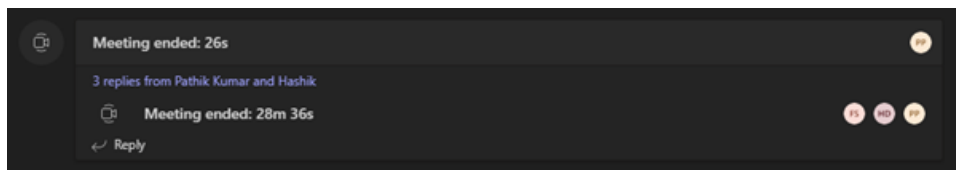
Timeline	features	Project completion
13 June – 20 June	Project setup	5%
21 <sup>st</sup> June – 26 <sup>th</sup> June	User Management	10%
27 <sup>th</sup> June – 7 <sup>th</sup> July	Online support, chat module and Escalation to the restaurant management	25%
8 <sup>th</sup> July – 20 <sup>th</sup> July	Browse the restaurants, place the food order, Tracking the order delivery, Visualizations, Menu	60%
21 <sup>st</sup> July – 27 <sup>th</sup> July	Report Generation Module, Provide Feedback & Ratings, Customer Feedback Polarity	90%
28 <sup>th</sup> July – 3 <sup>rd</sup> Aug	Testing the project	95%
4 <sup>th</sup> Aug – 6 <sup>th</sup> Aug	Deploying the Project and Submit the final project	100%

## Team meeting logs

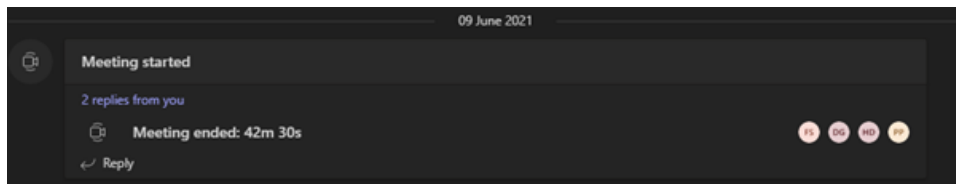
31<sup>st</sup> May:



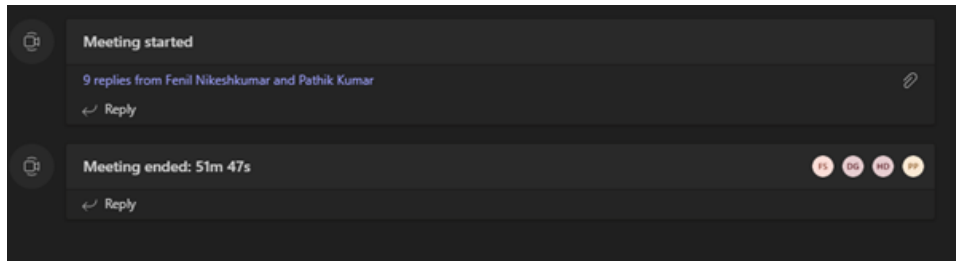
6<sup>th</sup> June:



9<sup>th</sup> June:



13<sup>th</sup> June:



### The budget of the project

Overall, the services we are going to use are involving the cost. We tried to make the project cost-efficiently. An estimate of our project is \$150 approximately. For example, Amazon QuickSight costs \$24/month for authors. AWS RDS MySQL costs \$0.017 per hour for db.t3.micro to store the database. In Google Cloud AutoML, \$3.15 per node hour for AutoML Vision Image Classification model training means it costs \$94.5/month. AWS Cost Explorer API costs \$0.01/request. Like these other services have some cost also. Though, we are going to use cloud-based features which take a charge according to the use. So, it might be minimal at the end of the project. [4][5][6]

### What do we know and how quickly we can learn?

From the brainstorming session, we are all well versed with React and NodeJS. We have an understanding of EC2, Lex, few GCP services, etc. from our lectures, assignment activities.

The **challenges** we are having are as a group we don't seem to have a lot of hands-on experience on the cloud services we are tackling this by learning the services as soon as possible to avoid delivery delays.

The **extras** we are planning is the customer feedback polarity for the restaurants to view and analyze.

### Why choose our team for this project?

We are using cloud serverless technologies which are pay per use. We are optimal with the choice of services such that they fall into our budget and get the job done without the loss of reliability, availability, or quality.



## References

- [1] "Amazon Cognito - Simple and Secure User Sign Up & Sign In | Amazon Web Services (AWS)", *Amazon Web Services, Inc.*, 2021. [Online]. Available: <https://aws.amazon.com/cognito/>. [Accessed: 13- Jun- 2021].
- [2] "Amazon Lex | chatbot with Amazon Lex | Amazon AWS", *Intellectyx*, 2021. [Online]. Available: <https://www.intellectyx.com/techstack/amazon-lex/>. [Accessed: 13- Jun- 2021].
- [3] "Online Diagram Software," *Lucidchart*. [Online]. Available: <https://www.lucidchart.com/>. [Accessed: 13-Jun-2021].
- [4] "Firebase Pricing. (n.d.). Retrieved June 13, 2021, from Google.com website: <https://firebase.google.com/pricing>".
- [5] "(N.d.). Retrieved June 13, 2021, from Amazon.com website: <https://aws.amazon.com/quicksight/pricing/>".
- [6] "Hill, C. (2020, October 31). Google Data Studio pricing, usage cost and limits. Retrieved June 13, 2021, from Analyticshelp.io website: <https://analyticshelp.io/blog/google-data-studio-pricing-cost-limits/>".