



DALHOUSIE UNIVERSITY

FACULTY OF
COMPUTER SCIENCE

CSCI 5410 Serverless Data Processing

Project Design Report

Group 4

Submitted by:

Adarsh Kannan Iyengar (B00900913)

Fenil Parmar (B00895684)

Jenish Patel (B00897765)

Kunj Patel (B00894376)

Meet Patel (B00899517)

Ruhi Rajnish Tyagi (B00872269)

Gitlab: https://git.cs.dal.ca/kvpatel/csci5410_group4

Contents

1. Project Architecture	3
2. Core Modules	4
2.1 User Management Module.....	4
2.2 Authentication Module.....	5
2.3 Online Support Module	5
2.4 Message Passing Module	7
2.5 Machine Learning Module	7
2.6 Web Application Building and Hosting.....	8
2.7 Other Essential Modules: Testing/Report Generation/Visualization	8
3. Roadmap	9
4. Module Distribution	11
5. Meeting Logs	12
Meeting 1.....	12
Meeting 2.....	12
Meeting 3.....	12
Meeting 4.....	13
Meeting 5.....	13
Meeting 6.....	14
Meeting 7.....	14
6. References	15

Table of Figures

Figure 1: Cloud Architecture	3
Figure 2: Unregistered Customer Roadmap	9
Figure 3: Registered Customer Roadmap	9
Figure 4: Admin Roadmap.....	10
Figure 5: Meeting 2 Log	12
Figure 6: Meeting 3 Log	12
Figure 7: Meeting 4 Log	13
Figure 8: Meeting 5 Log	13
Figure 9: Meeting 6 Log	14
Figure 10: Meeting 7 Log	14

1. Project Architecture

Figure 1 depicts the cloud architecture of the application. The diagram is designed using draw.io [3] by gathering knowledge from sources Google Cloud Documentation [4], and AWS Documentation [5]. As we can see in the figure, the users will interact with the front-end of the application developed in ReactJS. The front-end of the application is hosted on GCP Cloud Run via the docker image which is stored in the Container Registry of GCP. AWS API gateway is used to navigate the requests from various services to AWS Lambda and GCP API Gateway is used to direct the requests to the GCP Cloud Functions. Amazon DynamoDB is used for the storage of data for the application [4][5].

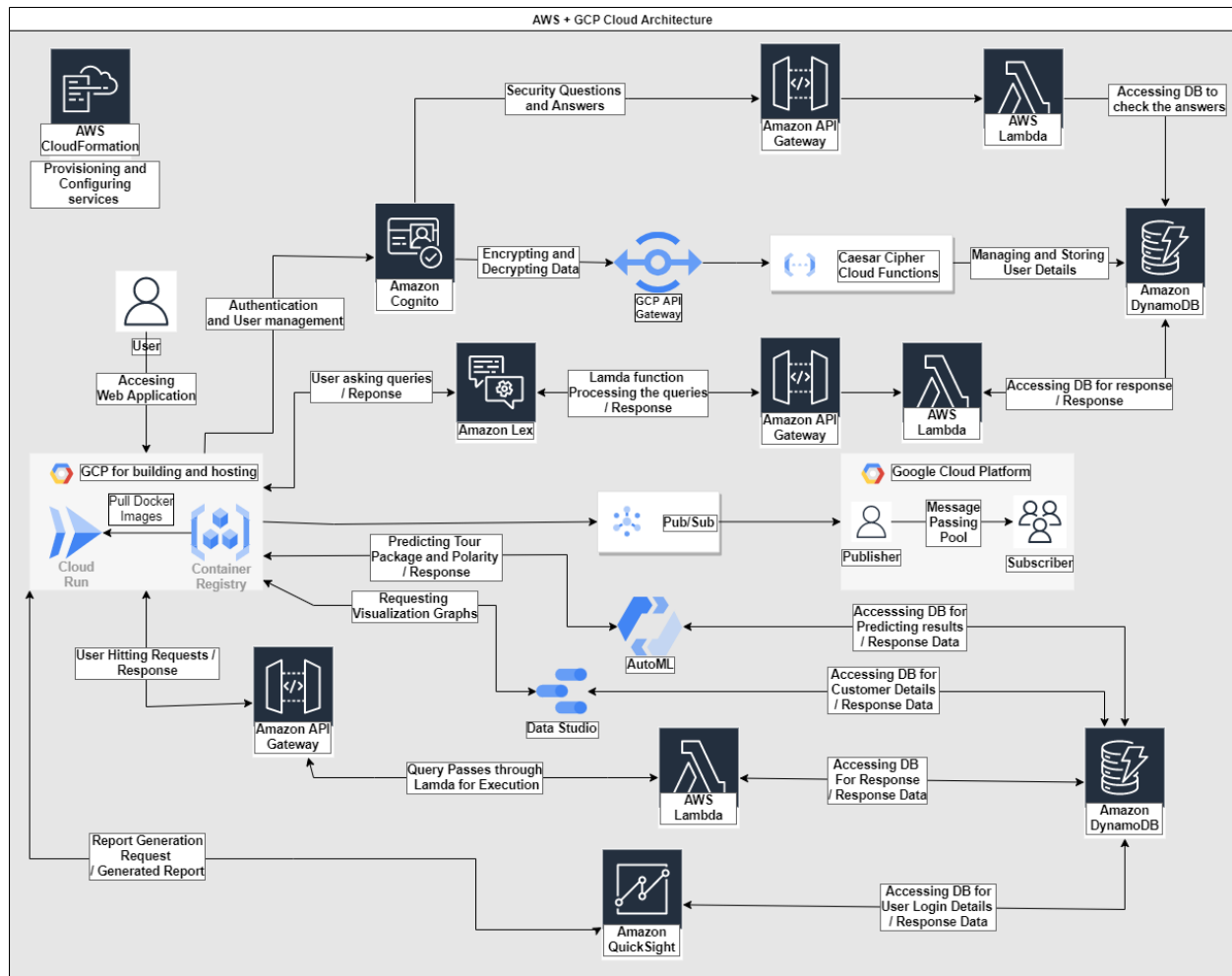


Figure 1: Cloud Architecture [3][4][5]

2. Core Modules

2.1 User Management Module:

- The main functionality of this module is to handle the user registration process along with dynamically assigning the User to the specific room booked by the user. The AWS Cognito [7] will be used for authentication, authorization, and user management. The users will be able to sign in using their username and password. The user pool and Identity pool will be used together to provide the essential services for the application. Furthermore, we will be also securely storing and maintaining the user details. AWS DynamoDB [8] service will be utilized to store the information of these users. The application initially will display the home page where the user will either log in to the application or can register themselves.
- If the user chooses to register themselves then the application will ask the user for details like Full Name, Email ID (It will be a unique User ID) and Password. There will be on-the-go validations done for the details entered by the user like for a valid email or for a strong password.
- Room Booking logic for Bread & Breakfast application:
 - At first, when there are no users registered within the application and there are no rooms booked by the users. When the first user registers with the application, the system checks if there are any rooms already booked. If no, then the user is assigned a room of their choice, and the user gets added to the logs of the customers registered till now in the bread & breakfast application data.
 - According to the requirements, each user detail can contain up to 3 users, so for the next two users that register with the application, their list will be added to the list of customers that were registered previously.
 - Now, if a fourth User registers with the application, then the application checks if there are any existing user details available to add a new User, if not then a new user detail gets created and respectively an entry is made in the AWS DynamoDB database.
 - To achieve this, we will be using in-code checks using AWS DynamoDB and AWS Lambda [9] function to dynamically add users to the respective customer's list.
- Now, all the information entered by the User, such as the full name, email and password and the information regarding the room bookings in the bread & breakfast is stored securely in the AWS DynamoDB [8].

2.2 Authentication Module:

This module is responsible for authentication.

1. We are planning to use AWS Cognito [7]. The module will utilize three ways such as ID-Password, Security Questions & Answers, and Caesar Cipher for the authentication.
2. The security questions & answers are stored in the DynamoDB [8] for verification.
3. Additionally, the Cloud Function service of the Google cloud platform will be used to cipher and decipher the plaintext with the Caesar Cipher algorithm.

2.3 Online Support Module:

A web-based method of providing customer care is online support. It is now a more common way to contact businesses than calling or visiting them. A component of customer relationship management is online support. The 24-hour customer care offered by online help is its finest benefit. When an issue arises or when a consumer has any queries about the service or product, they can quickly contact the agent.

The module's job is to deliver online support to both authorized and unauthenticated users employing Amazon Lex [11]. Lex will help create conversational answers that address the most common queries. The service will provide a variety of user contact scenarios so it can respond accurately and help users in the best way possible.

It will assist in the following areas:

1. Search for available rooms: The user will be able to see the available rooms.
2. Room bookings: The user can choose the kind of room they want and enter details to book the room.
3. Food ordering: It will give out the options like vegetarian, gluten-free, etc., match with the available menu, and suggest the options.
4. Navigation: It will redirect to the tour package
5. Contact us: General contact details of the B&B.

A demonstration of booking a room via Lex [4][5]:

Chatbot: Hi! Please choose one of the options:

1. Search for available rooms
2. Room bookings
3. Food ordering
4. Navigation
5. Contact us

Customer: 2

Chatbot: Excellent! Please choose to continue:

1. Double Room
2. Queen Suite

3. Deluxe Suite

Customer: 1

Chatbot:



Details: This will contain detailed information about the room, its benefits of it, and why is it special? and what do customers love about this?

Suite size: 300 sq ft

Beds: 1 Double(s)

Amenities: A/C, Wi-Fi, TV, Shower, Bath

Price: From \$100 per night

Chatbot: Provide your phone number to start your booking.

Customer: +1 234567890

Chatbot: Thank you. Number saved. Now type the name you want your reservation under.

Customer: Roma C.

Chatbot: Give the following details: Dates, No. of people, Number of nights, and special requests.

Customer: July 28, 2022, 3, 4, No scented candles

Chatbot: Great! Now lastly enter your email address.

Customer: roma@gmail.com

Chatbot: Thank you! A booking has been created under the name Roma C for a Double room on July 28, 2022, for 3 people for 4 nights. Your total amount is \$460.00 including taxes. Your booking reference is B00123456. Please provide this reference number on the day of your visit. Have a great stay!

Note: This is a conceptual demonstration, the application may or may not differ a bit from this.

2.4 Message Passing Module:

In the Message Passing Module, the main service that is used is the GCP Pub/Sub service as it is an easy-to-use service that fulfills the real-time notification requirement [10]. The authorized users will be notified of the situational messages from Hotel Management and Tour Operators modules via the GCP Pub/Sub service. These messages are stored in the AWS DynamoDB Database [8].

2.5 Machine Learning Module:

To identify the similarity of stay duration of customers and propose a tour package.

Google cloud platform offers various Machine Learning services which club into Vertex AI service [6]. To implement this module, we will use Vertex AI's Auto ML services [6]. Auto ML provides various methods to train, test, and evaluate the model. From the choice of many, we will be using the Tabular form of data to identify the suitable tour package and recommend it to the customers [6]. We will be using a classification algorithm to train the model. The process of developing this module is described below:

- We will create the dataset that will contain information on the stay duration of customers, and we will also set the pipeline that will continue updating the database with new entries about customers' stay duration.
- Clean that database and modify it to best fit the model.
- Apply classification algorithm on Vertex AI's Auto ML service [6].
- Train the model with the training dataset.
- Test the model with the test dataset and evaluate the performance by considering the Confusion matrix and Feature importance matrix [6].
- Generate the Endpoint to access the trained model and do the prediction.
- By entering the data into the model, we will receive a prediction from the model that will tell us what package is suitable for the customer based on their stay duration [6].

To identify the polarity of customer feedback and to add appropriate scores.

We will be using Vertex AI's Auto ML service to build this module [6]. Here, we must calculate the polarity of customer feedback with its score of it [6]. Therefore, we need to use the Text format of the data. We can use Text format with Auto ML and can build the machine learning model which takes text data and can give the polarity of customer feedback [6]. The implementation of this module is described below:

- We will create a Text dataset by adding data from various users to our application's database and collecting those data into one directory [6].
- After collecting the data, we have to label all the text files and divide them based on the polarity of the feedback [6].
- Cleaning of the data by removing stopwords, and other less important details from the data files.
- Build the model by using Sentiment Analysis on customer feedback data [6].
- Train the model using the training dataset.

- Test the model by using a test dataset with various matrix that defines the accuracy of the model.
- Generate the endpoint from which we can predict the result of the input data [6].

2.6 Web Application Building and Hosting:

We are going to use ReactJS for the front-end development and NodeJS with Express framework for the API connection as a back-end environment. Users will interact with the web application via the front-end module [4][5]. A good choice of front-end and back-end technologies is necessary when developing dynamic web applications [4][5]. Therefore, to achieve application flexibility and scalability, we have chosen to design the front-end using React JS. To access the functionality of GCP Cloud function services on the back end, we will be using Express JS, Node JS, and Lambda services [4][5]. This web application will be hosted by AWS Elastic Beanstalk service as it provides real-time load balancing by auto scaling [4][5].

2.7 Other Essential Modules: Testing/Report Generation/Visualization:

We are going to perform the validations and function tests for our application. Various validations and tests such as username-password validations, operations that can only be performed by the authorized users, whether a guest user can perform operations that are not restricted to them, and whether the chatbot can provide the correct information that the users are looking for, whether the users are notified for the relevant situations, whether the user can get the unique booking ids and order ids while booking the room and ordering the food, and many other tests according to the functionalities. The testing module will include all the possible test cases that are essential for the application to perform efficiently.

For the report generations module, we are going to use AWS QuickSight which is equipped with Business Intelligence dashboards with the applications [12]. This module generates reports on user login and access statistics. The authorized users of the application can generate the report. We are going to use the AWS CloudWatch service to log each of the user activities. The user activity information is provided to the QuickSight service to generate the reports [12].

To visualize the statistical data reports, we are going to use GCP Data Studio which can be integrated into various third-party services such as AWS services to create data stores and utilize them to generate different kinds of reports such as customer booking graphs, food ordering charts, profit charts, hotel's income charts etc. The data for visualizing and report generation will be stored in AWS DynamoDB [8].

3. Roadmap

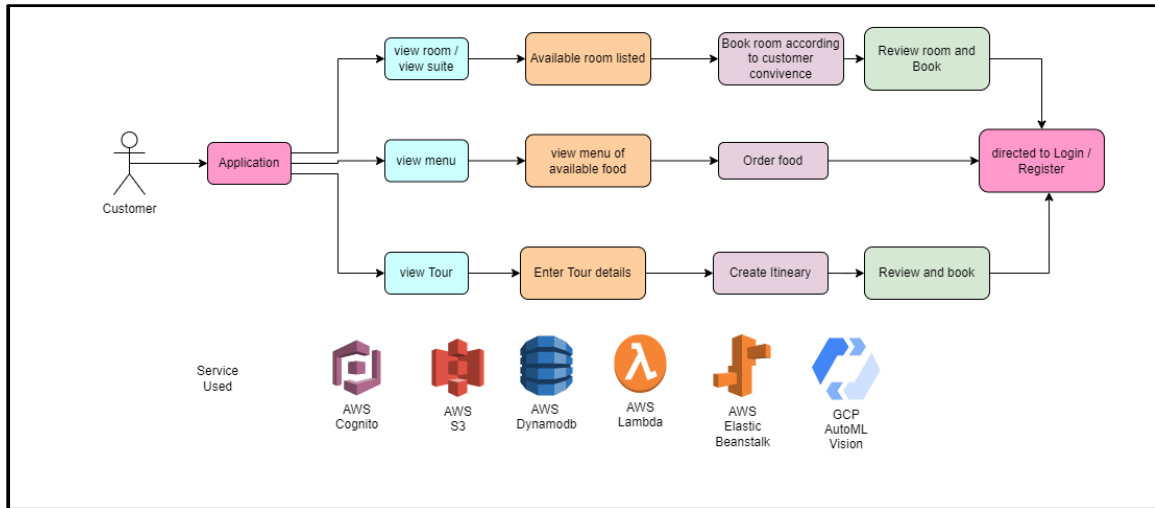


Figure 2: Unregistered Customer Roadmap [3][4][5]

- **Non-registered users**

Users who are not signed up will be able to browse rooms, menus, and tours. The user will see all the available rooms and their specifications when they access the rooms. The user will be sent to the login/registration page to proceed when attempting to select and book the room [4][5]. The menu can also be viewed by the user. When a user examines the menu, all the available breakfast options are displayed along with information like pricing. Additionally, the user can view recommended tours by selecting the view tours option. It will request the user's stay time and then display results based on how closely that duration matches the user's input [4][5].

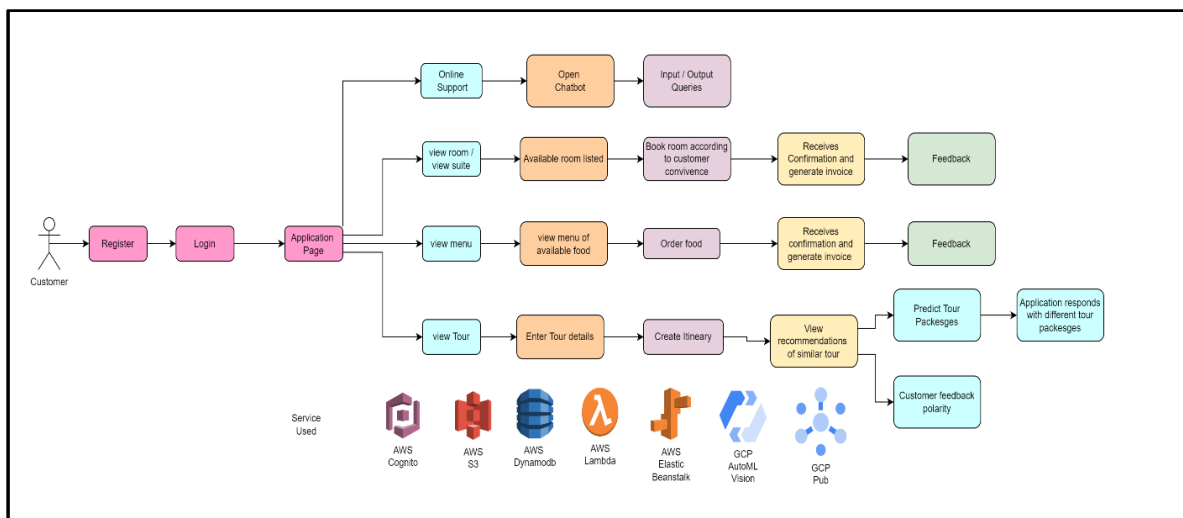


Figure 3: Registered Customer Roadmap [3] [4][5]

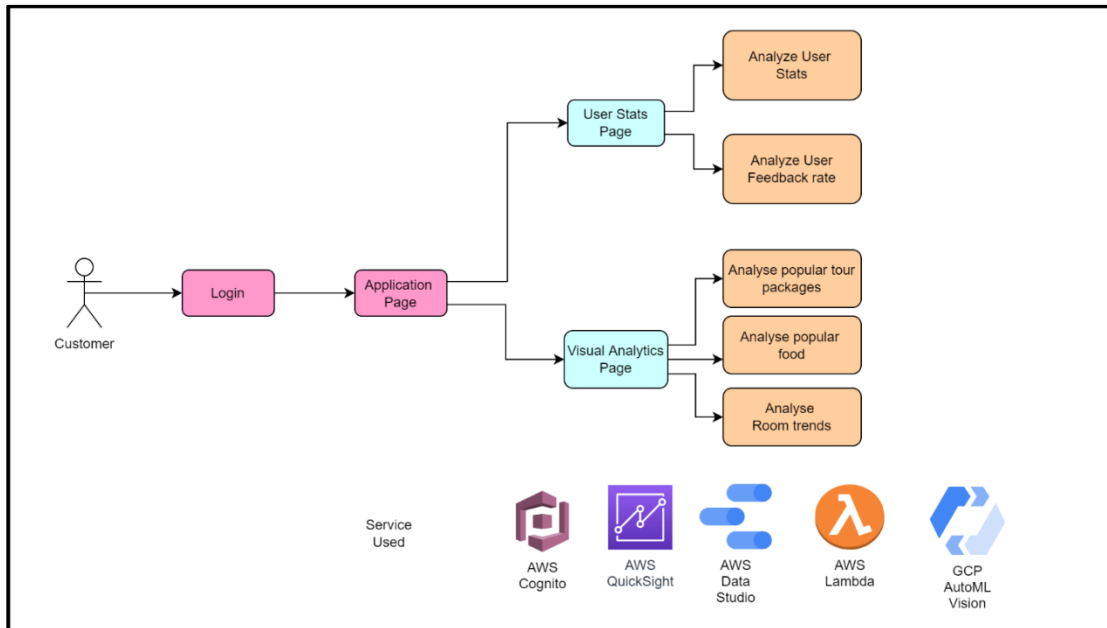


Figure 4: Admin Roadmap [3]

- **Registered users**

The register is chosen by the user from the navigation bar. They answer security questions and input their login and password. After submitting the form, Cognito will be used by a Lambda function through an AWS gateway to register the user [4][5]. The responses to the security question will also be kept in DynamoDB by Lambda. Additionally, upon successful registration, they will receive a random number that will be saved in DynamoDB [4][5].

Once the user is registered then he/she does not have to sign up again as data is already present in the DynamoDB so the user will sign in the next time he /she is using the application [4][5]. Now we will illustrate different scenarios of the interaction of users with the services available on our application [4][5].

I. User interaction with a chatbot:

The user will be able to communicate with a chatbot after logging in. Users can use this bot to search for available rooms and browse the website [4][5]. Additionally, it will give them suggestions and useful guidance for booking reservations and exclusive placing meal orders [4][5].

II. User booking a room:

The chatbot will let the user book a room or a bed as they interact with it. Once the user selects the option they want to use, this will happen automatically. Once the confirmation is received, the information about the reservation will be updated in DynamoDB [4][5]. After the reservation, the consumer will be asked for feedback. Sentiment analysis will be done on the feedback, which will be stored in DynamoDB [4][5].

III. User ordering food(breakfast):

Users will be able to examine a pre-defined menu that includes information on the breakfast alternatives that are offered, as well as their specifics and cost. Users will be able to choose from the menu and place meal orders. After a successful order, DynamoDB will save the order data, and the user will get an order confirmation notification [4][5]. They will also be able to see the order's current status, including whether it is being placed, prepared, ready, or delivered. Invoices are produced and saved to S3 buckets after orders are delivered [4][5].

IV. User viewing suggested tours:

Following the completion of the order, the customer will be requested to provide feedback [4][5]. This feedback will be kept in DynamoDB and based on the data, sentiment analysis will be performed to determine if the feedback was good or negative [4][5].

4. Module Distribution

Table 1: Module Distribution

No.	Module Name	Contributors Name
1	User Management Module	Adarsh Kannan Iyengar
2	Authentication Module	Jenish Patel
3	Online Support Module	Ruhi Rajnish Tyagi
4	Message Passing Module	Kunj Patel
5	Machine Learning Module	Fenil Parmar
6	Web Application Building & Hosting	Jenish Patel & Meet Patel
7	Other Essential Modules: Testing/Report Generation/Visualization	Kunj Patel & Fenil Parmar
8	Documentation	All members

5. Meeting Logs

Meeting 1

Date: 17th May 2022

Agenda: Team Introduction

Meeting 2

Date: 25th May 2022

Agenda: Discussion about the project specifications and getting to know the requirements.



Figure 5: Meeting 2 Log

Meeting 3

Date: 29th May 2022

Agenda: Task distribution and discussion regarding the cloud services that are to be used in the project.

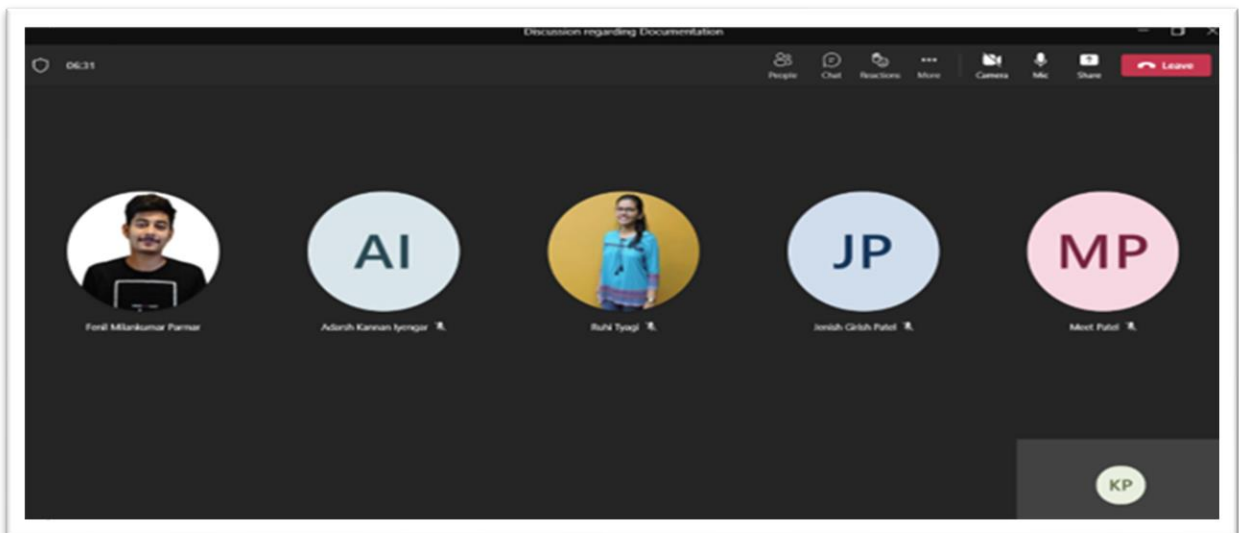


Figure 6: Meeting 3 Log

Meeting 4

Date: 1st June 2022

Agenda: Reviewing the conceptual design report

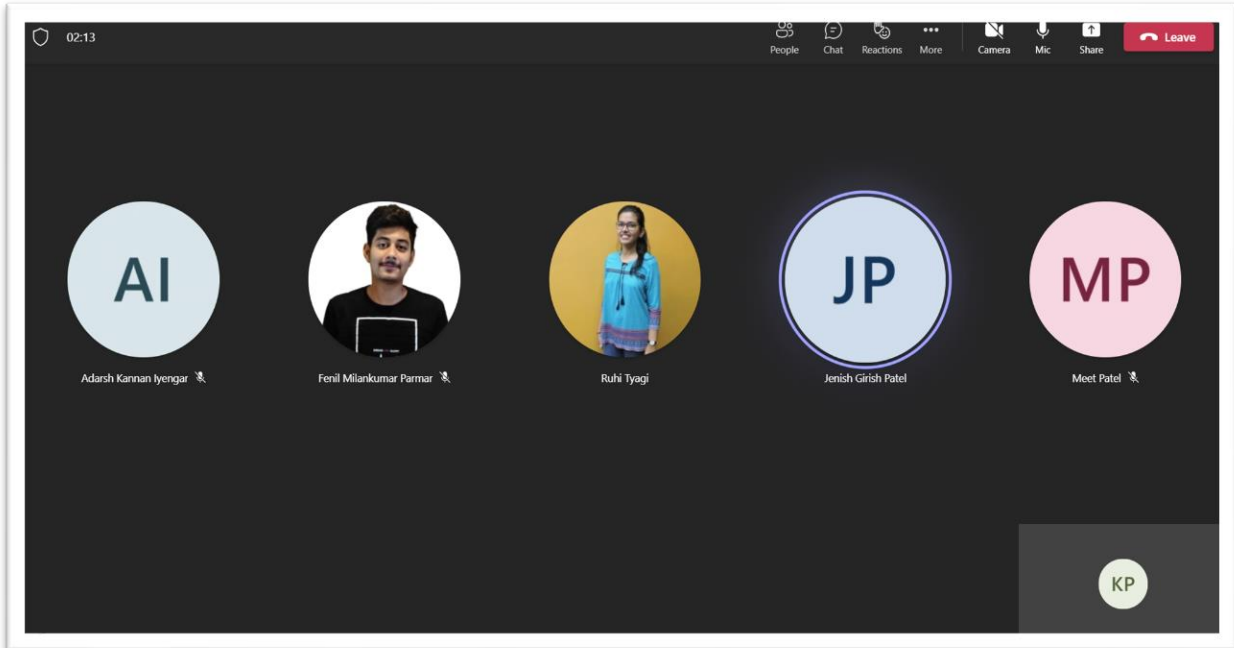


Figure 7: Meeting 4 Log

Meeting 5

Date: 28th June 2022

Agenda: Discussion regarding conception report feedback and dividing the modules amongst the team members.

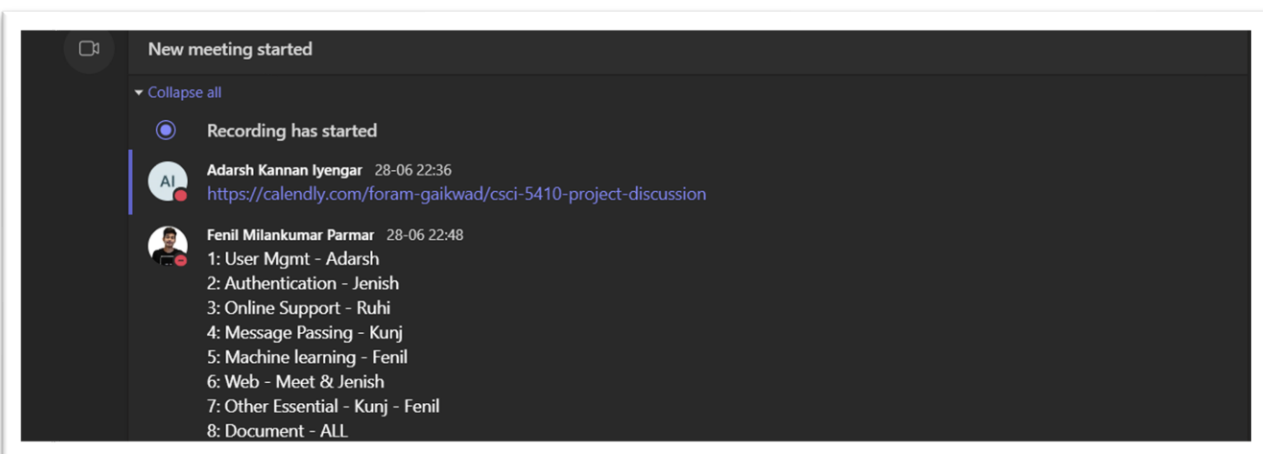


Figure 8: Meeting 5 Log

Meeting 6

Date: 4th July 2022

Agenda: Discussion regarding the design document and dividing the documentation tasks amongst the team members.

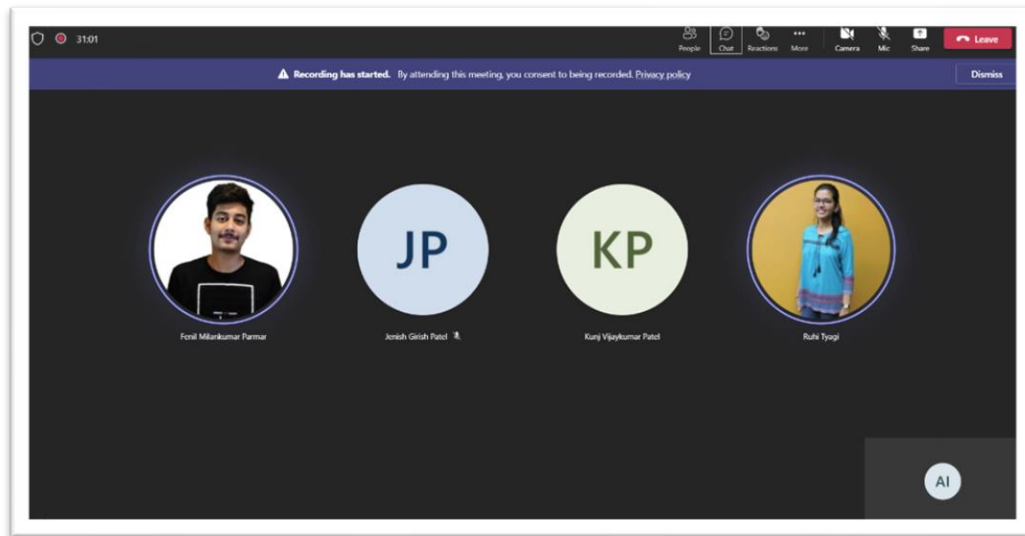


Figure 9: Meeting 6 Log

Meeting 7

Date: 5th July 2022

Agenda: Final meeting regarding the design document formatting and submission

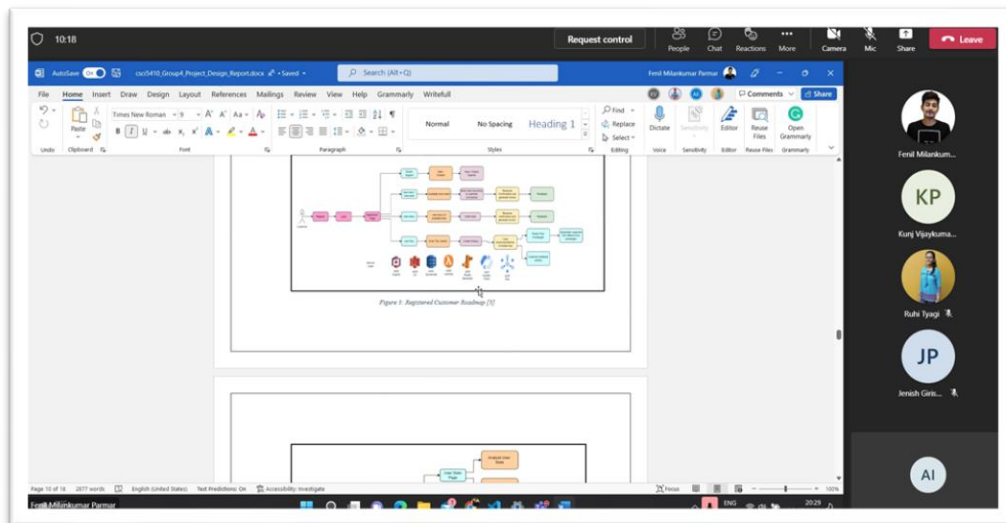


Figure 10: Meeting 7 Log

6. References

- [1] "Why Is Serverless the Future of Cloud Computing?", *Alibaba Cloud Community*, 2022. [Online]. Available: https://www.alibabacloud.com/blog/why-is-serverless-the-future-of-cloud-computing_597191. [Accessed: 05-Jul-2022].
- [2] M. Roberts and J. Chapin, "What Is Serverless?", *O'Reilly Online Learning*, 2022. [Online]. Available: <https://www.oreilly.com/library/view/what-is-serverless/9781491984178/ch04.html>. [Accessed: 05-Jul-2022].
- [3] "Flowchart Maker & Online Diagram Software", *App.diagrams.net*, 2022. [Online]. Available: <https://app.diagrams.net/>. [Accessed: 05-Jul-2022].
- [4] "Google Cloud documentation | Documentation", *Google Cloud*, 2022. [Online]. Available: <https://cloud.google.com/docs>. [Accessed: 05-Jul-2022].
- [5] "AWS Documentation", *AWS Docs*, 2022. [Online]. Available: <https://docs.aws.amazon.com/>. [Accessed: 05-Jul-2022].
- [6] "Vertex AI," *Google Cloud*. [Online]. Available: <https://cloud.google.com/vertex-ai>. [Accessed: 05-Jul-2022].
- [7] "AWS Cognito," Amazon.com. [Online]. Available: <https://aws.amazon.com/cognito/>. [Accessed: 05-Jul-2022].
- [8] "AWS DynamoDB," Amazon.com. [Online]. Available: <https://aws.amazon.com/dynamodb/>. [Accessed: 05-Jul-2022].
- [9] "AWS Lambda," Amazon.com. [Online]. Available: <https://aws.amazon.com/lambda/>. [Accessed: 05-Jul-2022].
- [10] "What is Pub/Sub?," *Google Cloud*. [Online]. Available: <https://cloud.google.com/pubsub/docs/overview>. [Accessed: 05-Jul-2022].
- [11] "What is Amazon Lex V2?", Amazon Lex, 2022. [Online]. Available: <https://docs.aws.amazon.com/lexv2/latest/dg/what-is.html>. [Accessed: 05-Jul-2022]
- [12] Amazon.com. [Online]. Available: <https://aws.amazon.com/quicksight/>. [Accessed: 05-Jul-2022].