LetsMeet: Group Meeting Scheduling and Transportation Suggestions

Jordan Park
Department of Computer Science
Columbia University
jjp2181@columbia.com

Yida Lin
Department of Computer Science
Columbia University
yl3842@columbia.com

Zhi Ji
Department of Electrical Engineering
Columbia University
zj2242@columbia.com

Sin-Yi Huang
Department of Computer Science
Columbia University
sh3907@columbia.com

Abstract— **LetsMeet is a web application designed for conveniently scheduling group meetings using cloud computing technologies. This application leverages AWS modules and multiple APIs including Yelp Fusion, Open Weather Map, The Movie DB, and Google Map to provide integrated information and services such as chatbot, cuisine lookup, movie suggestion, and weather forecast in one interface. In addition, it can make transportation route suggestion and calculate Estimated Time of Arrival (ETA) for various travel modes to facilitate the meeting process.**

Keywords-Amazon Web Service, Order Chatbot, Transportation Suggestion, Restaurant Recommendation, Movie Recommendation, Weather Forecast

I. INTRODUCTION

Choosing a meeting place may not be as easy as it seems to be. Multiple factors need to be taken into account such as geographical location, type of entertainment, travel mode, route selection, and weather. It becomes especially intricate when involving a group of people, as the meeting may be severely delayed when unexpected situations such as traffic jam or communication disruption happen to one or more group members.

LetsMeet aims for addressing these issues and providing smooth experience for scheduling group meetings. Registered users can create meeting groups of various purposes and send invitations to friends via email. Within a group, users can specify their departure addresses which are used by the app to automatically estimate a midpoint that is relatively convenient for all members in the group. The benefit for choosing the midpoint is that it is relatively more convenient geographically for all group members. A chatbot is built-in to get user preference on cuisine type and travel mode. It can search the Yelp Fusion API [1] near the intended destination and let the user choose from a list of places that match the preference. In addition, the app provides weather forecast information, and shows the current trending movies to give users more entertainment options in one integrated interface.

Once the destination is set, the app leverages Google Map API [2] to make route suggestion and provide Estimated Time of Arrival (ETA) according to the preferred travel mode. The app can also send reminders via SMS messages to users who are behind according to ETA.

II. ARCHITECTURE

LetsMeet adopts the React framework [3] for the frontend. Multiple APIs and AWS modules were utilized for the backend. Figure 1 visualizes the overall architecture.

Below is a brief description of how *LetsMeet* works from the user perspectives:

1) The user registers for an account in the signup page.
2) Upon completion of registration, the user is redirected to the home page that is connected to the DynamoDB.
3) The home page displays sets of panels containing various information:
    a) The weather panel shows the current weather information.
    b) The movie panel shows the top trending movies of the day.
    c) The group panel shows a list of groups available to the user.
    d) The group creation panel let the user create new groups.
    e) The invitation panel let the user invite friends to join a group.
4) The user can use the chatbot service after joining a group for cuisine suggestions. Upon getting the preferred cuisine type and travel mode, the chatbot looks for suitable places near a location that is convenient for all group members and gives the user a few choices that closely match the preference.
5) Once the destination is set, the user can go to the map interface powered by Google Map API for travel suggestions. The departure places of group members and their associated ETAs are rendered on the map panel. The optimal route to the destination for the current user is suggested based on the preferred travel mode acquired by the chatbot. Optionally, there is an interface where the user can send reminders to other group members who are currently behind according to the ETA.

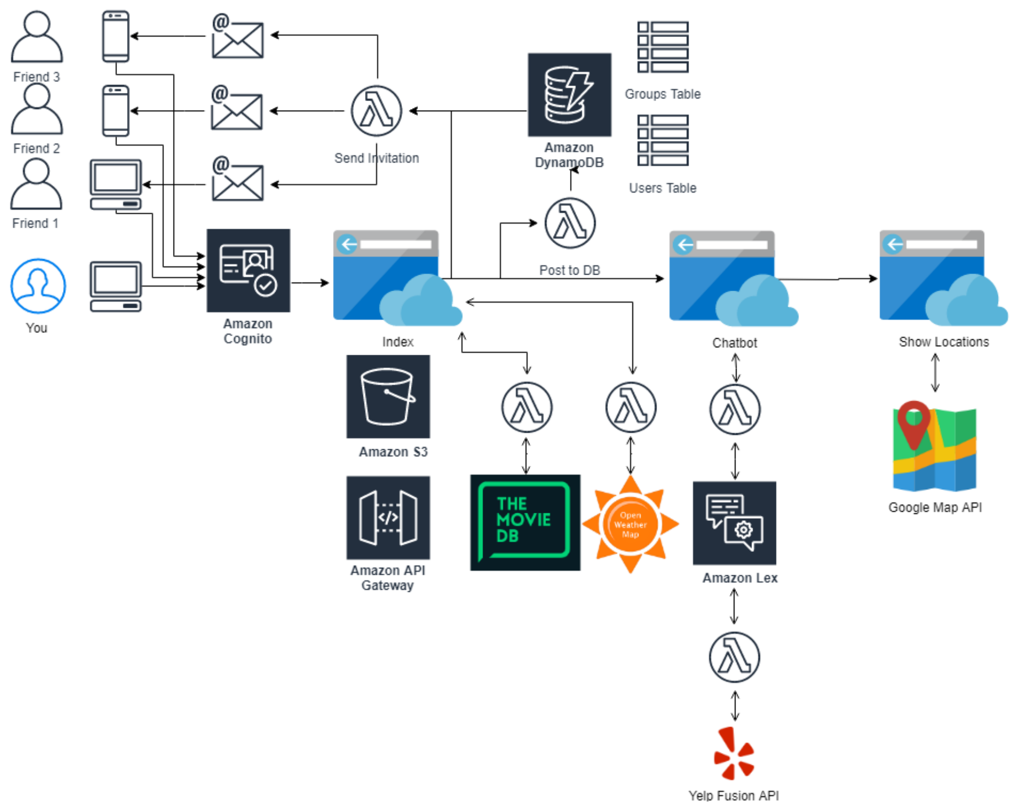


Fig. 1. The overall architecture of *LetsMeet*.

III. Implementation

A. *Frontend*

The frontend was written with the React Framework. It consists of the following webpages: The login page, sign up page, group control page, and map page. The frontend was deployed and hosted on AWS S3. Figure 2 visualizes the architecture of frontend.

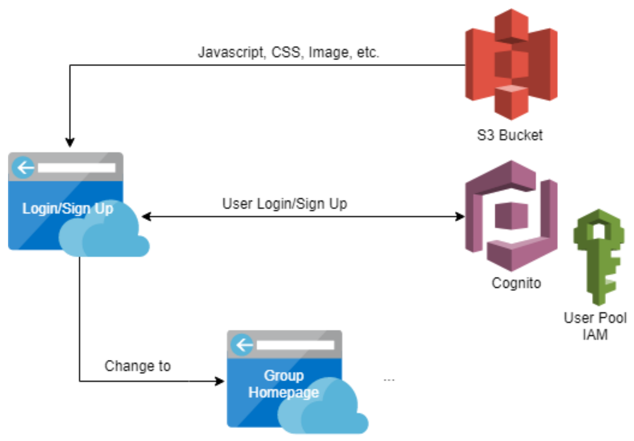
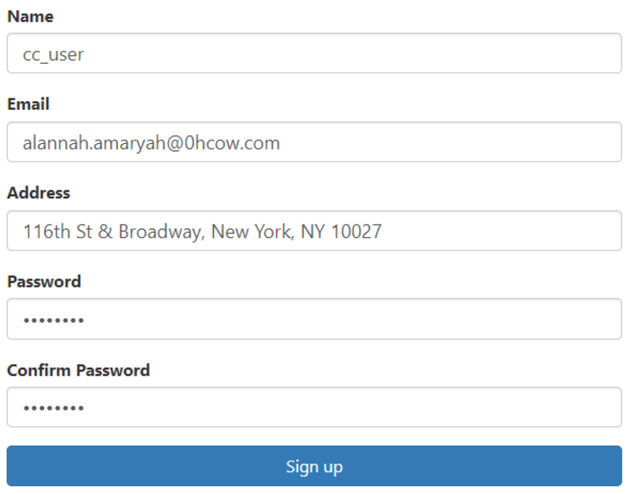


Fig. 2. The fronted architecture.

B. *Signup and Login*

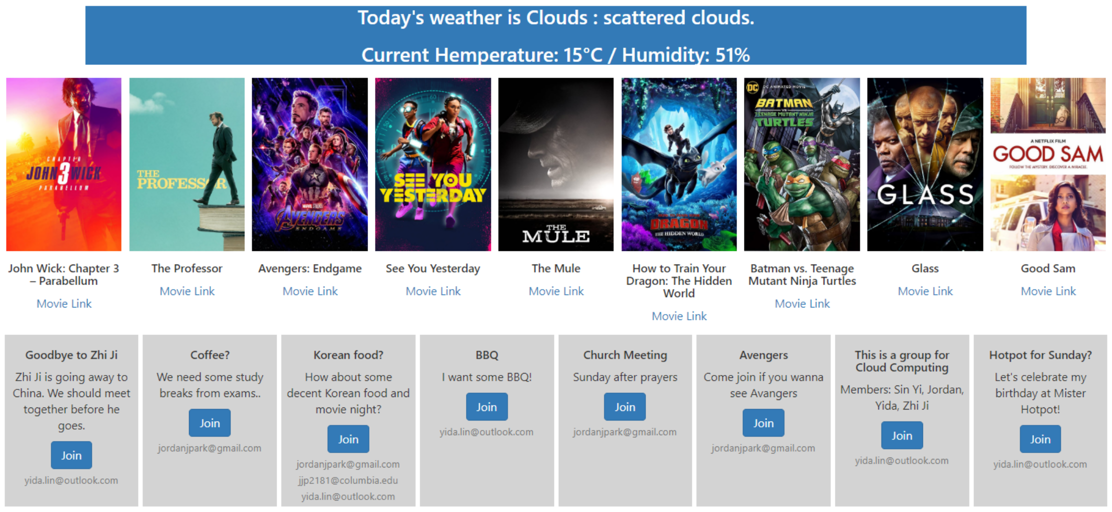
We use AWS Cognito to manage user identities. At the stage of signing up, the user needs to provide their name, email, location, and password. These information were also stored in the user table in AWS DynamoDB in which the email was used as the key. Figure 3 shows the information the user needs to provide in the sign up stage.

After signing up, the user logs in via email and password.



Name
cc_user
Email
alannah.amaryah@0hcow.com
Address
116th St & Broadway, New York, NY 10027
Password
••••••••
Confirm Password
••••••••

Fig. 3. The sign-up page.


Today's weather is Clouds : scattered clouds.

Current Hemperature: 15°C / Humidity: 51%

John Wick: Chapter 3 – Parabellum	The Professor	Avengers: Endgame	See You Yesterday	The Mule	How to Train Your Dragon: The Hidden World	Batman vs. Teenage Mutant Ninja Turtles	Glass	Good Sam
Movie Link	Movie Link	Movie Link	Movie Link	Movie Link	Movie Link	Movie Link	Movie Link	Movie Link

Goodbye to Zhi Ji	Coffee?	Korean food?	BBQ	Church Meeting	Avengers	This is a group for Cloud Computing	Hotpot for Sunday?
Zhi Ji is going away to China. We should meet together before he goes.	We need some study breaks from exams..	How about some decent Korean food and movie night?	I want some BBQ!	Sunday after prayers	Come join if you wanna see Avangers	Members: Sin Yi, Jordan, Yida, Zhi Ji	Let's celebrate my birthday at Mister Hotpot!
Join	Join	Join	Join	Join	Join	Join	Join
yida.lin@outlook.com	jordanjpark@gmail.com	jordanjpark@gmail.com jjp2181@columbia.edu yida.lin@outlook.com	yida.lin@outlook.com	jordanjpark@gmail.com	jordanjpark@gmail.com	yida.lin@outlook.com	yida.lin@outlook.com

Fig. 4. The group homepage with weather information and entertainment options.

C. *Group Homepage*

In the group control webpage, the user can create new groups by specifying the group name and description. An entry associated with the group will be sent to AWS DynamoDB and inserted into the group table. The user can invite friends to a group via email. This feature is implemented by letting the frontend send a POST request with the email address of the invited user to the AWS API Gateway, which then triggers an AWS Lambda function that sends emails via AWS Simple Email Service (SES). Figure 4 presents the entertainment options and weather information. Figure 5 visualizes the architecture design of the homepage. Figure 6 presents the interface for managing groups.

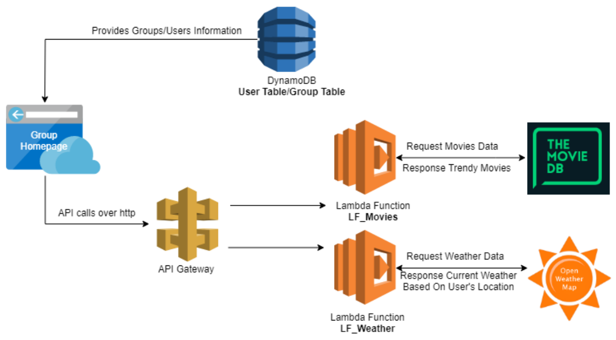


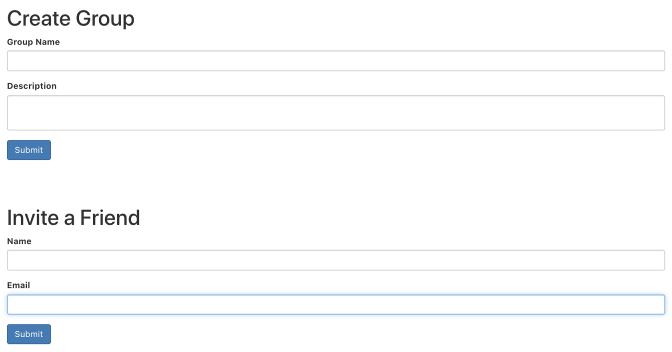
Fig. 5. The group homepage architecture.

D. *Weather*

The group homepage shows the current weather information in the weather panel. This is implemented by sending a POST request with the current geographic location (latitude, longitude) as the request body to AWS API Gateway, which then invokes the AWS Lambda function LF_Weather that queries the Open Weather Map API [5]. The frontend then extracts the returned results and displays the weather condition, temperature and humidity in the weather panel.

E. *Trending Movies*

The group homepage exhibits a list of popular movies in the movie panel. Similar to the weather information, this is implemented by sending a GET request from the frontend to AWS API Gateway, which then invokes the AWS Lambda function LF_Movies that queries The Movie DB API [4]. The frontend lists the posters of the top ten most trendy movies with associated links, which would redirect the user to the corresponding page in The Movie DB upon clicking.


Create Group

Group Name

Description

Submit

Invite a Friend

Name

Email

Submit

Fig. 6. The group management interface.

F. *Chatbot*

A chatbot is integrated to acquire user preference on cuisine type and travel mode. The chatbot was implemented in AWS Lex with an AWS Lambda function LF_Lex_Handler upon fulfillment of the chatting intent. The chatbot searches the Yelp Fusion API for cuisines near the midpoint of all group members and let the user pick from a list of top choices with detailed information. It then forwards the coordinates of the selected restaurant to the map webpage. Figure 7 and 8 visualize the chatbot design.

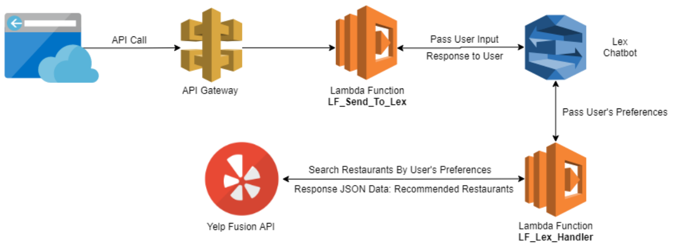
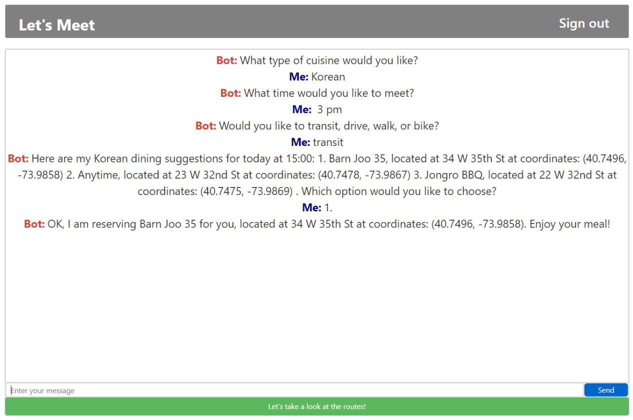

API Call
API Gateway
Lambda Function LF_Send_To_Lex
Pass User Input
Response to User
Lex Chatbot
Pass User's Preferences
Search Restaurants By User's Preferences
Response JSON Data: Recommended Restaurants
Yelp Fusion API
Lambda Function LF_Lex_Handler

Fig. 7. The chatbot architecture.



Bot: What type of cuisine would you like?
Me: Korean
Bot: What time would you like to meet?
Me: 3 pm
Bot: Would you like to transit, drive, walk, or bike?
Me: transit
Bot: Here are my Korean dining suggestions for today at 15:00: 1. Barn Joo 35, located at 34 W 35th St at coordinates: (40.7496, -73.9858) 2. Anytime, located at 23 W 32nd St at coordinates: (40.7478, -73.9867) 3. Jongro BBQ, located at 22 W 32nd St at coordinates: (40.7475, -73.9869) . Which option would you like to choose?
Me: 1.
Bot: OK, I am reserving Barn Joo 35 for you, located at 34 W 35th St at coordinates: (40.7496, -73.9858). Enjoy your meal!

Fig. 8. The chatbot designed for acquiring user preference on cuisine type and travel mode.

G. Map

The map webpage was implemented by leveraging a React package react-google-maps. The map panel renders all users departure places and the target destination with markers. It calculates the ETAs of all group members and shows the optimal route for the current user based on the prefered travel mode, which is acquired from the user via the chatbot.

The map webpage also consists of an interface from which the user can send reminders via SMS to other group members who are late according to the ETA. This interface was implemented by sending a POST request from the frontend to the AWS API Gateway, which then triggers the AWS Lambda function LF_Send_SNS that send messages via AWS Simple Notification Service (SNS). Figure 9 shows the architecture of the map service and Figure 10 visualizes the interface.

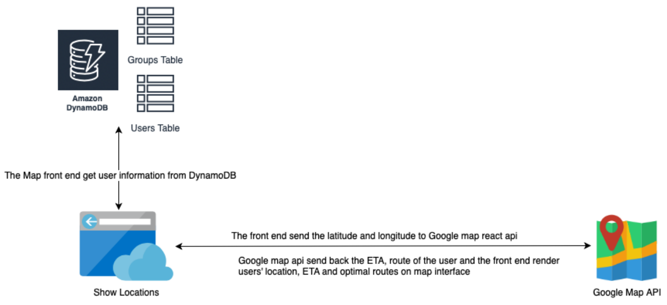
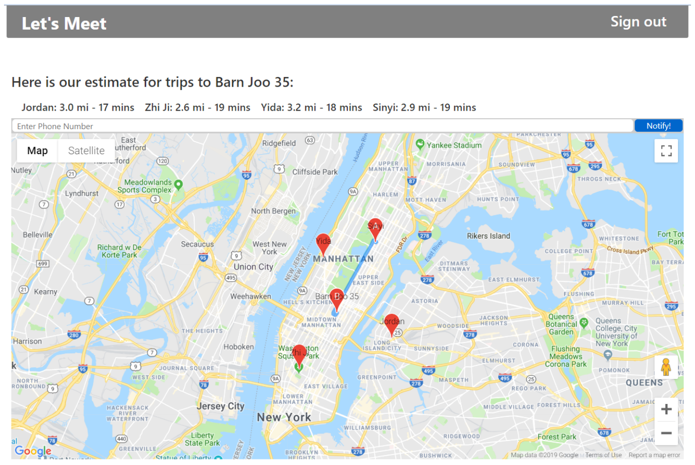

Amazon DynamoDB
Groups Table
Users Table
The Map front end get user information from DynamoDB
The front end send the latitude and longitude to Google map react api
Google map api send back the ETA, route of the user and the front end render users' location, ETA and optimal routes on map interface
Show Locations
Google Map API

Fig. 9. The map architecture.

IV. Results

Upon deployment, users can register account, create groups, and send invitations in LetsMeet. The app can suggest restaurants based on user preference and locations acquired via a chatbot, and suggest optimal routes and ETA according to the preferred travel mode. It also displays the current weather and trending movie information in an integrated interface to provide more entertainment options.



Here is our estimate for trips to Barn Joo 35:

Jordan: 3.0 mi - 17 mins Zhi Ji: 2.6 mi - 19 mins Yida: 3.2 mi - 18 mins Sinyi: 2.9 mi - 19 mins

Fig. 10. The map interface showing the user locations and the target destination. The optimal route and estimated time of arrival (ETA) were rendered by utilizing React package *react-google-maps*.

V. Conclusion

Leveraging AWS modules and multiple APIs including Google Map, Yelp Fusion, Open Weather Map, and The Movie DB, *LetsMeet* provides a convenient interface for scheduling group meetings and integrates related services such as movie suggestion, cuisine lookup, weather forecast, route suggestion, and ETA calculation to facilitate the process.

VI. Related Links

1) Deployed Version: http://restaurant-meetup.s3-website-us-east-1.amazonaws.com/
2) Presentation Decks: https://tinyurl.com/yyvsqczl
3) Github Repo: https://github.com/zhiji95/LetsMeet
4) Youtube demo: https://youtu.be/M0toieYQfDw

Acknowledgment

We would like to thank Prof. Sambit Sahu for his insightful lectures throughout the semester. We are also grateful to teaching assistants Aditya Das, Robby S. Costales, and Pankil T. Daru for their tremendous help and guidance.

References

[1] Yelp Fusion API: https://www.yelp.com/developers/documentation/v3/get_started
[2] Google Map API: https://developers.google.com/maps/documentation/
[3] React, A JavaScript library for building user interfaces: https://reactjs.org/
[4] The Movide DB API: https://developers.themoviedb.org/3/getting-started/introduction
[5] OpenWeatherMap API: https://openweathermap.org/current