CSCI5410

Serverless Data Processing

Group Project – DALVacationHome
Sprint 1

Submitted By:-
Divyank Shah – B00966377
Vivek Sonani – B00953064
Zeel Ravalani – B00917373
Yuci Wang – B00965074

Table of Contents

Research

Service: Amazon Lex

Amazon Lex is an AWS service for building conversational interfaces for applications using voice and text.[1] Amazon Lex provides the same conversational model that is used in Amazon Alexa. Amazon Lex was launched in April 2017 for developer community.
Best Practices: To optimize Amazon Lex V2 interactions, ensure all API requests use Signature V4 for authentication and avoid sending confidential information in session IDs. Configure slot value recognition by setting the valueSelectionStrategy field appropriately, and validate slot values before use. Use clear, non-ambiguous training data to improve intent recognition, and utilize the TSTALIASID alias only for manual testing, as it points to the Draft version and may cause interruptions if updated. For production, publish and use a stable version with an alias. [2]
Use Case: Amazon Lex will be used to create the Virtual Assitant in the DalVacationHome project.

Service: Amazon DynamoDB

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. With DynamoDB, you can create database tables that can store and retrieve any amount of data and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation. [3]
Best Practices: To optimize DynamoDB usage, employ batch operations for efficiency and cost reduction, utilize conditional updates for data consistency, and implement pagination for large queries. Use exponential backoff for throttled requests and leverage DynamoDB Streams for replication and event-driven actions. Additionally, implement caching for frequently accessed data, use transactions for atomic operations, and employ DynamoDB Local for development and testing. Monitor usage and performance with CloudWatch, and follow security best practices to protect your data[4].
Use Case: In DalVacationHome Amazon DynamoDB will be used to store question and answers for MFA in login, logs for communication between registered user and agents and for storing other data.

Service: Amazon Cognito

Amazon Cognito is an identity platform for web and mobile apps. It’s a user directory, an authentication server, and an authorization service for OAuth 2.0 access tokens and AWS credentials[5]. It mainly consist of two components user pool and identity pool which can work separately or in tandem based on the access need of the user.
Best Practices: To use Amazon Cognito effectively, enable multi-factor authentication, enforce strong password policies, and implement account recovery mechanisms. Integrate with external identity providers and use fine-grained access control with user groups and IAM roles. Protect user data with encryption and regular audits, optimize user experience with Cognito-hosted UI and custom flows, and ensure scalability and performance by monitoring usage and optimizing Lambda triggers. Follow security best practices and maintain compliance through audit logging and regular policy reviews.
Use Case: In DalVacationHome Amazon Cognito will be used to handle primary user authentication using user ID and password, ensuring secure and scalable user management.

Service: AWS Lambda

AWS Lambda is a serverless computing service that allows to run the code without managing servers. It automatically scales and executes the code in response to events, charging only for the compute time consumed [6].

Best Practices: Optimize function code by keeping it concise and efficient, leveraging modular design and libraries where possible. Configure functions appropriately, including setting memory allocation and timeouts based on workload requirements. Ensure scalability by designing functions to handle varying loads and considering concurrency limits. Utilize metrics and alarms to monitor function performance and health, enabling timely responses to issues. Safeguard functions with security best practices, including IAM roles, encryption, and least privilege access [7].

Use Case: AWS Lambda will be utilized within DalVacationHome for executing the multifactor authentication process, encompassing question/answer verification, Caesar cipher, and virtual assistant interaction, ensuring secure approval for room bookings.

Service: AWS SQS

Amazon SQS provides a reliable hosted queue service for seamless integration and decoupling of distributed software components [12].

Best Practices: For both standard and FIFO queues, prioritize batching messages to optimize throughput and cost efficiency. For FIFO queues specifically, utilize message deduplication and sequence numbers to ensure message order and prevent duplicates [13].

Use case: In DalVacationHome, AWS SQS will be utilized between application components such as "Room Booking Request" and "Room Booking approval" processes.

Service: AWS SNS

Amazon SNS is a managed service facilitating message delivery from publishers to subscribers through topics, serving as communication channels. Subscribers receive messages via various endpoints like AWS Lambda, HTTP, email, and mobile notifications, enabling asynchronous communication [10].

Best practices: Verify that only appropriate subscribers are assigned to your AWS SNS topics, preventing unauthorized access. Secure SNS topics to disallow unknown cross-account access, enhancing data protection across environments. Restrict publishing and subscription permissions to prevent unrestricted access by 'Everyone' to your SNS topics. Implement Server-Side Encryption and utilize KMS Customer Master Keys to safeguard the content of SNS topics against unauthorized exposure [11].

Use case: AWS SNS will be utilized in the notification module of DalVacationHome.

Service: GCP Pub/Sub

To achieve message passing between customer and We will use pub/sub to ingest user interaction and server events [12]. This module will log messages in DynamoDB. When the user forwarded a message, we may use some kind of stream processing tool. And it will deliver the message to the DynamoDB.

Best practices: Use of batch processing of the messages. Messages will be batched into one request to publish. It trades latency for efficiency. We need to also set maximum outstanding messages to avoid too many messages accumulating in memories resulting in error [13].

Use cases: Message communication between customers and agents.

Service: GCP CloudRun
To host our application we will need to use CloudRun. We will also apply IAM to restrict the internet traffic [14]. CloudRun will create unique endpoints for every services which will facilitate use of spring boot. It can also scales fast according to the increasing requests.
Best practices: Configure CPU to be always-allocated to use background activities. This will make cloud run supports background activities. Delete temporary files to avoid out of memory error [15].
Use cases: Use cloud run to host the backend.

Service: AWS CloudFormation
All the infrastructures can be scripted in one yaml file. This will help the teammates have the same understanding of the infrastructures. It will also reduce human errors. And CloudFormation makes tracking changes of the infrastructures easier. We will use git to track different versions of yaml files. All changes can be seen which makes error tracking easier.
Best practices: Use parameter constraints when defining infrastructures [16].
Use cases: Management of all infrastructures.

Technology Stack and Application Design Considerations
Backend:
- **Node.js for Lambda functions:** Well-suited for AWS Lambda due to its event-driven nature and extensive runtime support [17]. Consider using the Serverless Framework or AWS Toolkit for simplified deployment and development [18].
- **Spring Boot for GCP Cloud Run:** Offers a rapid development framework for microservices. Google Cloud provides a Spring Boot adapter for Cloud Run deployment [19].

Frontend:
Bootstrap + Javascript (jQuery) + React:
- A popular and well-supported stack
- Bootstrap offers pre-built components for responsive design.
- Javascript (jQuery) adds interactivity.
- React facilitates building complex user interfaces with reusable components.
- Consider using Material-UI for pre-built React components that follow Google's Material Design guidelines [20].

Application Design Best Practices
- **Microservices Architecture:** Considering a microservices architecture to break down the application into smaller, independent services for better scalability and maintainability [21]. This aligns well with the serverless approach on AWS Lambda and Cloud Functions. Both Node.js and Spring Boot offer good support for microservices development.
- **API Design Best Practices:** Planning to follow RESTful API design principles for creating well-structured and predictable APIs [22]. This ensures consistency and ease of use for integrating different functionalities across the application.
- **Code Quality and Maintainability:** Enforceing code style guidelines (e.g., using linters), write unit tests for your code, and document your code clearly for better maintainability [23]. These practices ensure code quality, reduce errors, and improve collaboration within the development team.

- **Frontend Design Decisions:** Exploring Material Design principles for a consistent and user-friendly frontend experience [24]. Material Design offers a set of UI components and best practices for building visually appealing and intuitive user interfaces

Google Language API Research

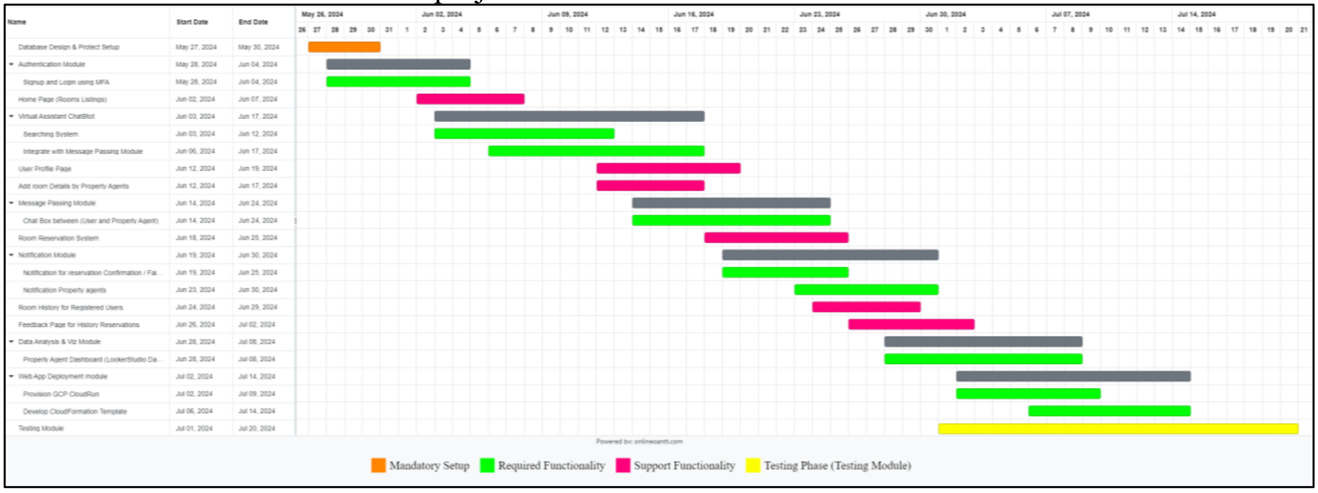
The Google Natural Language API provides functionalities for sentiment analysis, entity recognition, and other text processing tasks. It can be integrated with your backend functions (Node.js or Spring Boot) to analyze user feedback and present sentiment in the frontend [25].

- **Functionalities:** Sentiment Analysis: Analyze the emotional tone of user feedback (positive, negative, neutral).
- **Integration:** The API offers client libraries for Node.js and Java, allowing integration with your backend functions written in either language.
- **Pricing:** Google Cloud offers a free tier for the Natural Language API, with pay-as-you-go pricing for exceeding the free tier quota [25].

By following these research findings and best practices, we plan to build a robust, scalable, and user-friendly DALVacationHome application.

Project Planning

We have created a Gantt chart with the timeline of the project. Figure 1. displays the Gantt Chart with initial timeline for the project.



Name	Start Date	End Date
Database Design & Protect Setup	May 27, 2024	May 30, 2024
Authentication Module	May 28, 2024	Jun 04, 2024
Signup and Login using MFA	May 28, 2024	Jun 04, 2024
Home Page (Rooms Listings)	Jun 02, 2024	Jun 07, 2024
Virtual Assistant ChatBot	Jun 03, 2024	Jun 17, 2024
Searching System	Jun 03, 2024	Jun 12, 2024
Integrate with Message Passing Module	Jun 06, 2024	Jun 17, 2024
User Profile Page	Jun 12, 2024	Jun 19, 2024
Add room Details by Property Agents	Jun 12, 2024	Jun 17, 2024
Message Passing Module	Jun 14, 2024	Jun 24, 2024
Chat Box between (User and Property Agent)	Jun 14, 2024	Jun 24, 2024
Room Reservation System	Jun 18, 2024	Jun 25, 2024
Notification Module	Jun 19, 2024	Jun 30, 2024
Notification for reservation Confirmation / Fai...	Jun 19, 2024	Jun 25, 2024
Notification Property agents	Jun 23, 2024	Jun 30, 2024
Room History for Registered Users	Jun 24, 2024	Jun 29, 2024
Feedback Page for History Reservations	Jun 26, 2024	Jul 02, 2024
Data Analysis & Viz Module	Jun 28, 2024	Jul 08, 2024
Property Agent Dashboard (LookerStudio Da...	Jun 28, 2024	Jul 08, 2024
Web App Deployment module	Jul 02, 2024	Jul 14, 2024
Provision GCP CloudRun	Jul 02, 2024	Jul 09, 2024
Develop CloudFormation Template	Jul 06, 2024	Jul 14, 2024
Testing Module	Jul 01, 2024	Jul 20, 2024

Figure 1. Gantt Chart with initial project timeline [26].

We are going to use the Gitlab board for managing the project work for effective collaboration. Gitlab Board Link: https://git.cs.dal.ca/yuci/CSCI5410-S24-SDP-7/-/boards.

Individual Contribution

Table 1. Individual Contribution

Divyank Shah	Background Research on AWS services Amazon Lex, Amazon DynamoDB and Amazon Cognito.
Vivek Sonani	Background Research on AWS services AWS Lambda, AWS SQS and AWS SNS.
Yuci Wang	Background Research on services GCP Pub/Sub, GCP CloudRun and AWS CloudFormation.
Zeel Ravalani	Gantt Chart Preparation, background research on Tech Stack to be used for project, Application design best practices and research on Google Language API.

References

[1] "What is Amazon Lex?", *Amazon Web Services Documentation - Amazon Lex V1* [Online]. Available: https://docs.aws.amazon.com/lex/latest/dg/what-is.html. [Accessed May 25, 2024].

[2] "Guidelines and best practices - amazon lex," *Amazon Web Services Documentation - Amazon Lex* [Online]. Available: https://docs.aws.amazon.com/lexv2/latest/dg/guidelines.html [Accessed May 25, 2024].

[3] "What is Amazon dynamodb?," *Amazon Web Services Documentation - Amazon DynamoDB* [Online]. Available: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html. [Accessed May 25, 2024].

[4] M. Prajapati, "Best practices for dynamodb," *DEV Community* [Online]. Available: https://dev.to/monikaprajapati_70/best-practices-for-dynamodb-4jh6. [Accessed May 25, 2024].

[5] "What is Amazon Cognito?," *Amazon Web Services Documentation - Amazon Cognito* [Online]. Available: https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html. [Accessed May 25, 2024].

[6] "What is AWS Lambda? - AWS Lambda", *AWS Docs*. [Online]. Available: https://docs.aws.amazon.com/lambda/latest/dg/welcome.html. [Accessed: May 25, 2024].

[7] "Best practices for working with AWS Lambda functions", *AWS Docs*. [Online]. Available: https://docs.aws.amazon.com/lambda/latest/dg/best-practices.html. [Accessed: May 25, 2024].

[8] "What is Amazon Simple Queue Service", *AWS Docs*. [Online]. Available: https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/welcome.html. [Accessed: May 25, 2024].

[9] "Best practices for Amazon SQS", *AWS Docs*. [Online]. Available: https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-best-practices.html. [Accessed: May 25, 2024].

[10] "What is Amazon SNS? - Amazon Simple Notification Service", *AWS Docs*. [Online]. Available: https://docs.aws.amazon.com/sns/latest/dg/welcome.html. [Accessed: May 25, 2024].

[11] "AWS SNS Best Practices", *AWS Docs*. [Online]. Available: https://www.trendmicro.com/cloudoneconformity-staging/knowledge-base/aws/SNS/. [Accessed: May 25, 2024].

[12] Google Cloud, "What is Pub/Sub?", *Google* [Online]. Available: https://cloud.google.com/pubsub/docs/overview?_gl=1*1rtagtp*_up*MQ..&gclid=Cjw

KCAjw9cCyBhBzEiwAJTUWNafUSwk87q71DzLl0ithE9sRH8kmYEYHUKlruOOaeCE4p9xs0ohOARoCRWMQAvD_BwE&gclsrc=aw.ds. [Accessed: May 24, 2024].

[13] Google Cloud, "Best practices to publish to a Pub/Sub topic", *Google* [Online]. Available: https://cloud.google.com/pubsub/docs/publish-best-practices. [Accessed: May 27, 2024].

[14] Google Cloud, "What is Cloud Run", *Google* [Online]. Available: https://cloud.google.com/run/docs/overview/what-is-cloud-run. [Accessed: May 24, 2024].

[15] Google Cloud, "General development tips", *Google* [Online]. Available: https://cloud.google.com/run/docs/tips/general. [Accessed: May 27, 2024].

[16] AWS, "What is AWS CloudFormation?", *AWS* [Online]. Available: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/Welcome.html. [Accessed: May 24, 2024].

[17] "Aws Lambda adds support for node.js 16," *AWS Documentation* [Online]. Available: https://aws.amazon.com/about-aws/whats-new/2022/05/aws-lambda-adds-support-node-js-16/. [Accessed May 27, 2024].

[18] "AWS Serverless Application Model - Amazon Web Services," *AWS Documentation* [Online].Available: https://aws.amazon.com/serverless/sam/. [Accessed May 27, 2024].

[19] T. Nadkarni, "Step by step guide to deploying spring boot application on Google Cloud run," *Frontend Engineering For Modern Web* [Online]. Available: https://www.frontendeng.dev/blog/31-step-by-step-guide-to-deploy-spring-boot-app-on-google-cloud-run. [Accessed: May 27, 2024].

[20] "Material UI: REACT components that implement Material Design," *Material UI* [Online]. Available: https://mui.com/material-ui/. [Accessed May 27, 2024].

[21] "Microservices Pattern: Microservice architecture pattern," *microservices.io* [Online]. Available: https://microservices.io/patterns/microservices. [Accessed: May 27, 2024].

[22] "What is rest?," *REST API Tutorial* [Online]. Available: https://restfulapi.net/. [Accessed May 27, 2024].

[23] "Maintainability," *Wikipedia* [Online]. Available: https://en.wikipedia.org/wiki/Maintainability. [Accessed: May 27, 2024].

[24] "Material Design Guidance," *Material design* [Online]. Available: https://m2.material.io/design. [Accessed: May 27, 2024].

[25] "Cloud natural language | google cloud," *Google* [Online]. Available: https://cloud.google.com/natural-language?hl=en. [Accessed: May 27, 2024].

[26] "Free online gantt chart software," *Free Online Gantt Chart Software* [Online]. Available: https://www.onlinegantt.com/#/gantt. [Accessed May 29, 2024].

Serverless Data Processing Team - 7 Meeting Logs

Meeting Date	Meeting Agenda	Minutes of Meeting	Attendance	Meeting Recordings
21 May, 2024	Initial discussion and project overview	1. We discussed the arrange a meeting with Head TA. The link is not out yet. 2. We created the google doc for the team report due on 29th May. We haven't created the repository due to details haven't been released yet. 3. We also decided the features as well as the backend modules. 4. We will wait for 1 or 2 days for more details to be released.	All Members Present	https://dalu-my.sharepoint.com/:v:/g/personal/vv842962_dal_ca/EQHUqrb39VRPlZf84GIs3hYBe7yZA04qrFmrqZITrN5dYg?referrer=Teams.TEAMS-ELECTRON&referrerScenario=MeetingChicletGetLink.view.view
23 May, 2024	Further discussion and task distribution	1. Decided upon individual topics to research and contribute in the Sprint 1. 2. Document Discussed about the overall features and planning of the project. 3. Booked and appointment with TA for the mandatory meet	All Members Present	https://dalu-my.sharepoint.com/:v:/g/personal/yc457137_dal_ca/Efak15RbzaxMnxLK9l-WDXgBkz5qjW6eEKsuKgHNrfxP2A?referrer=Teams.TEAMS-ELECTRON&referrerScenario=MeetingChicletGetLink.view.view