PROJECT REQUIREMENT DOCUMENT – CSCI 5410/S24

Title: DALVacationHome
Team Size: 3 to 4 Members
Project Length: Approx. 2.5 months
Team Leader: Bharat Shankaranarayanan

Objective:

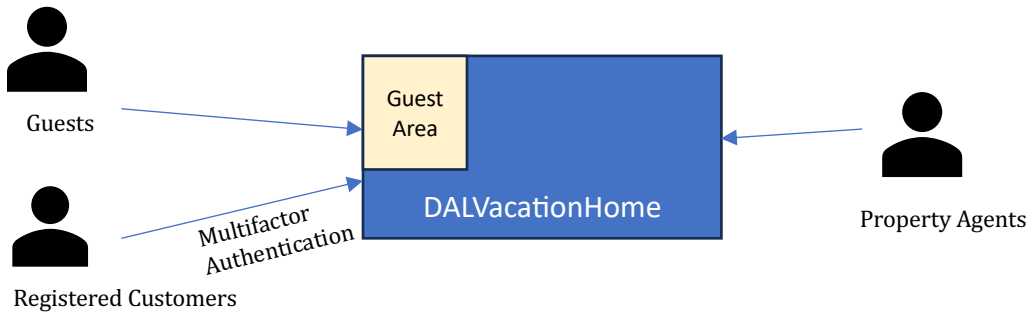
The primary objective of this project is to build cloud plumbing system, where an application will be designed using serverless technologies to process data (more specifically on-demand data). It should not require any specialist for using the implemented system. In this project, you will be building a "cloud data plumbing system", which could be used by many clients to process their data. You will use different backend services, and simple front-end application to build the application. This project is introduced in the Serverless Data Processing Course (CSCI 5410) to fulfill the course requirement. This is a group project (weightage 40%), and each group is required to perform specific tasks within a given time frame. There are project constraints, and scope, which must be followed by each team. The project has multiple deliverables, which must be completed at the end of each sprint.

Plagiarism Policy:

- This project is a group task. Collaboration of any type (outside the assigned group) amounts to a violation of the academic integrity policy and will be reported to the AIO.
- Content cannot be copied verbatim from any source(s). Please understand the concept and write in your own words. In addition, cite the actual source. Failing to do so will be considered as plagiarism and/or cheating.
- AI (Artificial Intelligence) tools cannot be used for any part of the project component. AI policy mentioned in syllabus must be followed.
- The Dalhousie Academic Integrity policy applies to all material submitted as part of this course. Please understand the policy, which is available at: https://www.dal.ca/dept/university_secretariat/academic-integrity.html

Hypothetical Scenario:

DALSoft5410 is building a serverless DALVacationHome application using multi-cloud deployment model, and backend-as-a-service (BaaS). The "DALVacationHome", should provide customization feature, and additional services for authorized users, and limited services to guests. The DALVacationHome should provide an online virtual assistance, which can quickly answer the queries of users, and in addition, the application should provide a message passing functionality between the authorized users and agents.



Use Case:

There will be three different types of users in the DALVacationHome system.
(1) **Guests**, (2) **Registered Customers**, and (3) **Property Agents** (They are super users or admin users).

Guest Users

- The guest users should be able to visit the application site, and check availability and tariffs of different types of rooms (rooms or recreation room).
- The guest users will be able to use virtual assistant or the chatbot to get basic navigation information.
- The guest users should be able to view the feedback and polarity of the overall feedback received for a particular room (rooms or recreation room)

Registered Customers

- Registered Customers will be able to perform all the operations that a guest user can perform. In addition, the following points will be applicable:
- A registered customer will get a notification once the registration process is completed.
- The registered customers will access the system using multi-factor authentication. 1st factor is userID/password, 2nd factor is Question-Answer, and 3rd factor is solving clue using Caesar cipher. Each factor must be validated in a sequential manner.
- The registered user will also get a notification after a successful sign-in process.
- Once a registered customer enters the system, they should be able to reserve a room or recreation room for a specific period (from a start date to an end date. Time can be set to standard, such as 12 pm).
- Once logged-in, a customer can also use the virtual assistant to
    - navigate within the site.
    - obtain the room number and duration of stay based on booking reference code.
    - Support customer to agent communication
- Registered customer should be able to provide feedback using a standard form about their stay in a room or usage of recreation room.

Property Agents

- Property Agents will be able to perform all the operations that a guest user can perform. In addition, property managers should be able to do the following:
- A property agent will get a notification once their registration process is completed.
- The property agents are admin users, but they will also access the system using multi-factor authentication. 1st factor is userID/password, 2nd factor is Question-Answer, and 3rd factor is solving clue using Caesar cipher. Each factor must be validated in a sequential manner.
- The property agents will also get a notification after a successful sign-in process.
- Once a property agent enters the system, they should be able to add/update a room or recreation room, such as adding a new room with a room number, adding more features {mini fridge, air purifier etc.), updating price, adding a discount code etc.
- Once logged-in, an property agents can also use the virtual assistant to
    - navigate within the site.
    - obtain the room number and duration of stay based on booking reference code.
- Property Agents should be able to communicate with the customer once a ticket is received. This is an asynchronous communication. It can also be designed as a synchronous communication as an additional feature.

TASK DETAILS

Module Details:

You must use Severless architecture to build the following modules.

1. **User Management & Authentication Module:**

Sign up with validation and multi-factor authentication using AWS – Perform the user registration through your light-weight front end application, and use the following services:

- User details should be stored in DynamoDB.
- User ID/Password (1st factor authentication) – AWS Cognito
- Question/Answer (2nd factor authentication) – DynamoDB + AWS Lambda
- Caesar cipher (3rd factor authentication) – AWS Lambda + DynamoDB

2. **Virtual Assistant Module:**

Bots should respond to queries (AWS Lex + DynamoDB + AWS Lambda) –

- Online virtual assistance for navigation within the application site. Such as, answering “how to register?”
- searching room numbers and usage or stay duration based on booking reference code.
- Accepts customer concerns and forwards to Property Agents.

3. Message Passing Module:

Message passing module (GCP pub/sub) will be used for message passing between parties.

- Customers will submit the concerns based on booking reference code, which will be published using GCP-Pub, and a function subscribed to message will forward it to a random property agent for answering.
- This entire communication should be logged in a NoSQL database like, DynamoDB or Firestore.

4. Notifications:

Notification module will use AWS SNS and AWS SQS

- The service will be used for sending successful registration, and successful login notifications to the registered user's email.
- The notification service will also be used for sending booking confirmation and booking failure. **Note**: Payment processing is not within the scope of the project. We assume payment will be done in-person after a successful booking, therefore, ignore building payment module.
- SNS – SQS should be used between application components, such as - "Room Booking Request" and "Room Booking Approval" process. Booking request will enter queue (Standard SQS), which will trigger Lambda for performing the approval task.

5. Data Analysis & Visualization:

Product of this module is used by all types of users.

- Property Agents can check total users, and login statistics on the admin page within frontend (LookerStudio dashboard – maybe embedded within a frame)
- Guests, Customers, and Property Agents should be able to see the customers' feedbacks in a tabular format within the frontend retrieved from DynamoDB or S3 bucket or Firestore or GCP storage.
- The feedback will automatically be analyzed in the backend to understand the sentiment and present in the frontend for all types of users. Use Google Natural Language API

6. Web Application Building and Deployment:

Build a front-end application using suitable framework such as, React and use it to call all the backend services –.

- Hosting of entire application and user/client facing interface should be done using GCP CloudRun
- Use CloudFormation or GCP Cloud Deployment Manager for automation of service deployment.

General Tasks:

- Testing Module (GCP + AWS) – Test cases must be validated. Use a table to capture the test cases and results of the testing.
- Check for errors and border line cases.
- In addition to test cases your team decides, you also need to perform the following tests:
    - Registered users' successful authentication and guests' users failed login,
    - Data accuracy or integrity test in upload-retrieval process to/from database,
    - Chatbot response test for valid and invalid utterances
    - API testing for reliability and functionality check,
    - Lambda or Cloud Function testing using test events and inspecting logs,

- Provide evidence of testing in the form of screenshots
- To maintain code quality – focus on code readability, code reusability, code cleaning, commenting.
- Perform regular git commits. Expected mainly during Sprint 2 and Sprint 3.
- Use any standard project management tool for better management of resources, and progress tracking.

Deliverables

***** Fill the Team Feedback Forms (Each member must submit each form once)**
Peer Feedback for Sprint 1: https://forms.office.com/r/xnF5T5txGw (May 15 - May 29)
Peer Feedback for Sprint 2: https://forms.office.com/r/g6FmhEihZf (May 29 - Jul 5)
Peer Feedback for Sprint 3: https://forms.office.com/r/0RYQsUH97q (Jul 5 - Jul 31)

Sprint 1 (10% of Course Total, which is 25% of Project Grade):

1. Deliver a 4-page Team report on **May 29, 2024 (20%)**
   [Report evaluation will be based on project rubric given in syllabus]
    - Document details of research,
    - Learning through background research must be documented.
    - Include individual contribution.
    - Project planning details with Gantt chart and highlighting the usage of project management tool.
    - **Additional Submission:**
        - Meeting logs in the form of spreadsheet highlighting agenda, minutes etc.
        - Teams Meeting recordings – links for Microsoft Streams of your meeting videos
2. Attend Q&A session with HeadTA between **May 20** and **May 28** and explain individual contribution and challenges. **(5%)**

70% Points assigned to group contribution	30% for individual contribution

saurabh.dey@dal.ca

Sprint 2 (10% of Course Total, which is 25% of Project Grade):

1. Deliver a 4-page Team report on **Jul 5, 2024 (20%)**
   [Report evaluation will be based on project rubric given in syllabus]
   - Provide details of research,
   - Individual contribution,
   - gitlab code repository link,
   - System architecture,
   - Pseudocode/ algorithm for important modules' logic
   - test cases,
   - evidence of testing for completed modules.
   - **Additional Submission:**
     - Meeting logs in the form of spreadsheet highlighting agenda, minutes etc.
     - Teams Meeting recordings – links for Microsoft Streams of your meeting videos
2. Attend Q&A session with HeadTA between **Jun 24** and **Jul 4** and explain individual contribution and challenges. **(5%)**

50% Points assigned to group contribution			50% for individual contribution		
Meetings	Document	Project Management	Code Quality	Git commits	Task status

Sprint 3 (20% of Course Total, which is 50% of Project Grade):

1. Deliver a 4-page Team report on **Jul 23, 2024 (20%)**
   [Report evaluation will be based on project rubric given in syllabus]
   - with details of research,
   - individual contribution,
   - gitlab code repository link,
   - updated final architecture,
   - Pseudocode/ algorithm for important modules' logic
   - test cases,
   - evidence of testing for completed modules.
   - **Additional Submission:**
     - Meeting logs in the form of spreadsheet highlighting agenda, minutes etc.
     - Teams Meeting recordings – links for Microsoft Streams of your meeting videos
2. Final DEMO 15-min presentation to the HeadTA and/or Instructor between **Jul 23** and **Jul 31**. **(25%)**
3. Overall Project Completion Status **(5%)**
4.

50% Points assigned to group contribution			50% for individual contribution		
Meetings	Document	Project Management	Code Quality	Git commits	Task status

saurabh.dey@dal.ca

Project Expectations:

- Please respect every individual and follow the “Culture of Respect” section provided in the course syllabus.
- All members should contribute equally to the project.
- One or two member(s) of a team completing the project cannot be considered as a successful group project, and the entire group may receive lower score.
- All members of a team may not receive similar score. Each sprint will have individual contribution grade as well.
- Try to use some form of project management software to track your project status and timeline. E.g.