Project Specification – Version 1.0 (Living Document)

Release Date: May 17, 2022

Title:

Serverless B&B *- A Multi-Cloud based Serverless bed and breakfast system*

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). This is like building a water plumbing system, which is built once with plenty of interconnected pieces, and used many times. It does not require any specialist for the operation. 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 system.

Explanation:

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 will follow an agile model, where each team should welcome change in the requirements. However, considering the time and resource restriction, requirement changes will be limited.

Hypothetical Scenario:

DALSoft5410 is building ***Serverless B&B*** using multi-cloud deployment model, and backend-as-a-service (BaaS). The application should provide customization feature, and additional services for authorized users, and limited services to guests. The ***Serverless B&B*** 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 some authorized systems.

This is a cloud-based hotel reservation system, where customers are allowed to register and book available rooms, and opt for services, such as Food. The Hotel Management component of the application allow registrations, and create invoices considering the stay, food ordered etc. A virtual assistance provides two different types of services, such as basic website navigation, and room availability check by guest users, and managing booking, ordering food by authorized hotel customers. There is an additional service, which is tour operator, and the service will be activated only if a tour is requested. The tour related operations are not part of the application, but is a component, which needs some integration with the actual application. Fig 1 shows an overview of the application.

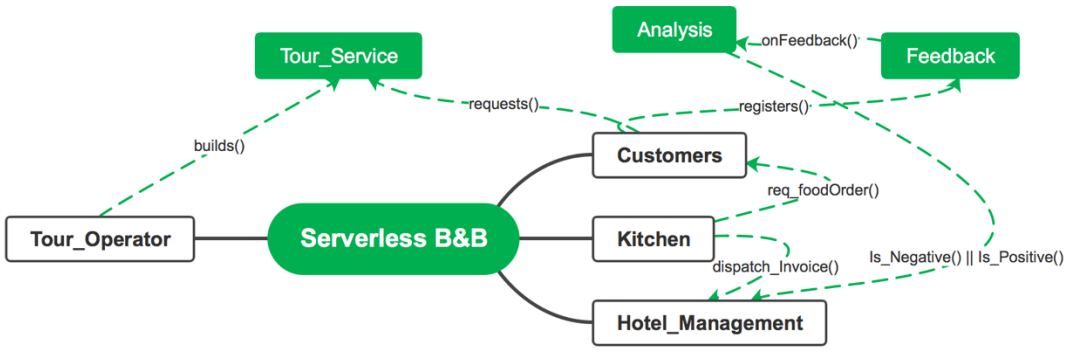

Fig 1: Serverless B&B Application

The primary services of the application are:

- **Customers**: This service is responsible for registering customers and allow them to login to the "Serverless B&B" application. In addition, the service is responsible to communicate with other services to register feedback, book tours, and order food etc.
- **Hotel_Management**: This service is responsible for registering customers and allow them to login to the "Serverless B&B" application. It can book rooms or beds and interact with Kitchen to acknowledge its service. It can receive customer feedback analysis report and may change a service.
- **Kitchen**: This service is responsible to receive order request from customers, send invoices to hotel management, and prepares meal (only breakfast). The meal preparation requires inventory access, cooking, scheduling, order management etc.
- **Tour_Operator**: This is a service, which will always listen to requirements from Customers, and builds tailored tour services. It generates tickets/pass and email it to customers.

DALSoft5410 has selected serverless application to minimize the development and project running cost. The company has identified two cloud platforms - AWS, and GCP to build, test, and deploy their application. They have decided to follow the official documentations of AWS and GCP to build the different pieces.

If they select server-oriented architecture, then they need to manage and configure the backend service, which they cannot do due to their resource limitations. Therefore, serverless is the only solution they found at this point. They have obtained two types of accounts from AWS, and one account from GCP, which they can use for building, testing, and deploying their application.

Since they are going to follow agile method, they can build, test, and change each component of the project whenever there is a change in the requirements.

Serverless B&B Application Overview:

In this application, you are required to build a front-end application, which will interact with Backend-as-a-Service (BaaS). Following are the broad categories of requirements that are identified at the beginning of the project. The name of cloud platform written in parenthesis indicates that you need to use that cloud platform for the task.

You must use Severless architecture to build the following

1. User Management Module:
    a. Sign up validation (GCP or AWS) – Registering customers for new room booking. A returning customer does not need to re-register.
    b. Once registered the user will get a customer number dynamically assigned with a valid logic based on the room booking.
    c. Managing and storing User details (GCP or AWS)
2. Authentication Module:
    a. User authentication Logic - Multi-cloud model (GCP+AWS) –
        - ID-password – Cognito OR FirebaseAuthentication
        - Question Answer: Firestore+CloudFunction OR DynamoDb+Lambda OR similar approach
        - caesar cipher – CloudFunction OR Lambda
3. Online Support Module:
    a. Bots should respond to queries (AWS) – Online virtual assistance for navigation, searching available rooms, booking rooms, ordering food etc.
4. Message Passing Module:
    a. Authorized customers should be able to communicate with Hotel Management and Tour operators (GCP) – Pub/Sub
5. Machine Leaning:
    a. To identify the similarity of stay duration of customers and propose a tour package
    b. To identify the polarity of customer feedback and to add appropriate score (GCP) – You can use GCP built-in algorithms.
6. Web Application Building and hosting:
    a. Building a front-end application using suitable framework, and calling backend services – Use of React, Nodejs, Lamda etc.
    b. Hosting of entire application and user/client facing interface (GCP CloudRun or AWS)
7. Other Essential Modules:
    a. Testing Module (GCP + AWS) – Test cases must be validated. Check for error and border line cases.
    b. Report Generation Module, e.g. User login or access Statistics (GCP and/or AWS)
    c. Visualization Module (GCP) – Customer booking graph, customer food orders, profit or income charts etc.
8. Documentation:
    a. This project requires extensive and systematic documentation.
    b. Every team meeting must be logged with dates, and added as part of conception document, design document, and final report.