Project Specification – Version 1.0

Release Date: May 26, 2020

Title:

DALServerlessLMS - A Multi-Cloud based Serverless Learning Management 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 large-scale 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 at the last phase of the development will not be addressed.

Hypothetical Scenario:

DALSoft5410 is building a serverless Learning Management System (LMS) using multi-cloud deployment model, and backend-as-a-service (BaaS). The LMS - "DALServerlessLMS", should provide customization feature, and additional services for instructors, and limited services to students. The LMS should provide an online virtual assistance, which can quickly answer the queries of users, and in addition, it should provide a chat functionality between the registered users.

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 components of the project whenever there is a change in the requirements. One important aspect of this project is building a serverless portable application, which can be used by multiple clients without making

changes to the system. E.g. University "X" uses the application to perform file management, searching, chat, and analysis. Another university "Y" can use the same application by creating their accounts in the system and start using the system without making any changes. The setup for university "X", and "Y" should not take more than few minutes. In addition, the application should provide certain options to perform user level configuration change. The main module of the application includes – user management, file management, data processing, analytics, instant messaging, virtual assistance etc.

Important Dates:

Date Given: May 26, 2020

Midterm Review: Week of June 15, 2020 {During Lectures, and Lab}

Final Demonstration: Week of July 13, July 20, and July 27 - For your team's time slot, a survey will be conducted on the 1st week of July.

Report Submission Deadline:

- Feasibility Report - June 9, 2020 at 11:59 pm
- Design Document - June 26, 2020 at 11:59 pm
- Final Report - July 31, 2020 at 11:59 pm

Note: For late submission of any report, there will be a 10% deduction per day. (exception – Student Declaration of Absence for medical reasons).

DALServerlessLMS 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.

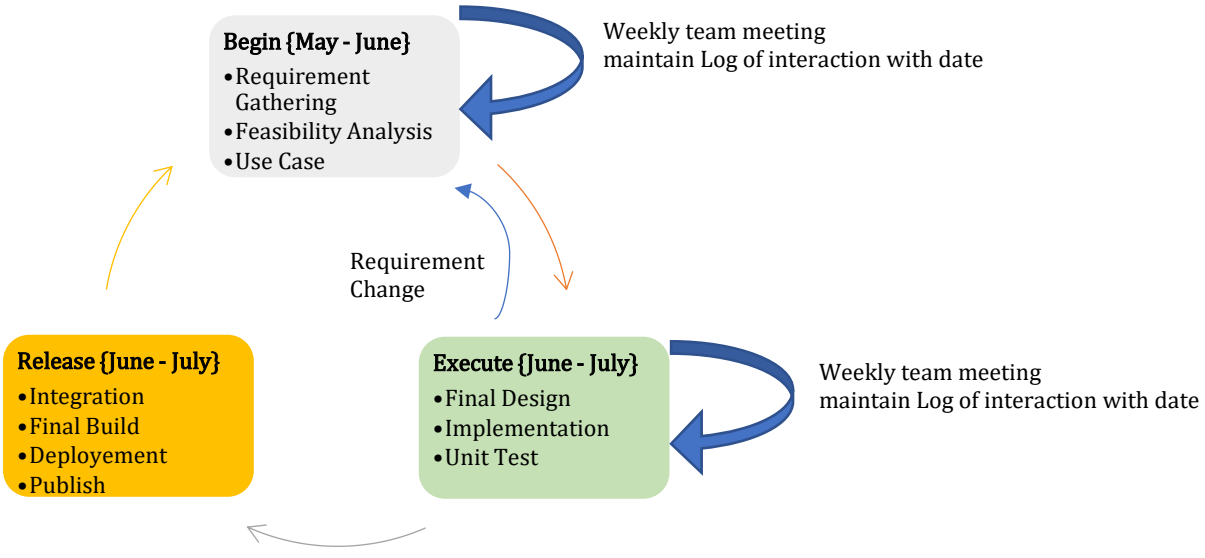
You must use Severless architecture to build the following

1. User Management Module:
    a. Sign up validation (GCP) – Instructor, and Student signup and validation
    b. Managing and storing User details (AWS)
2. Authentication Module:
    a. User authentication Logic (GCP+AWS) – Multifactor {ID-password, and Question Answer}
3. Online Support Module:
    a. Bots should respond to queries (AWS) – Online virtual assistance for navigation, searching, processing etc.
4. Chat Module:
    a. Group members should be able to communicate (GCP) – Instant Messaging Engine
5. Data Processing:
    a. A container-based application to extract named entities (uppercase entities) from uploaded files and build a word cloud (GCP). Use cloud run to run your container.

6. Machine Leaning:
    a. To identify the classification of the file and add appropriate tags (AWS) – classification of files based on size, metadata, title, type etc. SageMaker and choice of algorithms could be used. E.g. if files on computer science, life science, history etc. are uploaded, then it should classify the files based on similarity score. Depending on the type of job, or type of events, the system should call appropriate algorithm and perform the analysis.
7. 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 (AWS)
8. Other Essential Modules:
    a. Testing Module (GCP + AWS) – Test cases must be validated
    b. Report Generation Module, e.g. User Statistics (GCP and/or AWS)
    c. Visualization Module (GCP)
9. Documentation:
    a. This project requires extensive and systematic documentation.
    b. Every team meeting must be logged with dates, and added as part of design document, and final report.

Note: This is initial requirements of the project. You can start the feasibility analysis and project planning. In version 2.0 few more details will be added.

Project Flow:



Short Term Expectations and Deliverables:

1. Working on Feasibility Analysis Document
    a. Identifying core components of project
    b. Find Challenges
    c. Knowing team members, and identify strength
    d. Knowledge Transfer {Please use Team's channel or email "you can Cc me and TAs"}
    e. Regular meetings with team members
    f. Document your questions, and concerns

2. Midterm Review
    a. In midterm review, I will check your progress and provide you feedback on your approach
    b. Your meeting logs will be reviewed.
        - Your meeting logs should be submitted as part of feasibility, design, and final documentation

Final Deliverables:

1. Documents:
    a. Feasibility Report – 2%
    b. Design Document – 8%
    c. Final Report - 10%

2. Presentation:
    a. Pre-recorded presentation (maximum 30 min) on background study, team contribution, and test case. {You can use zoom to record your Team's presentation and upload the audio/video, and the presentation scripts (if any) to specific Brightspace submission folder}
    b. Pre-recorded presentation (maximum 30 min) on demonstrating the working model, test results, and analysis {You can use zoom to record your Team's presentation and upload the audio/video, and the presentation scripts (if any) to specific Brightspace submission folder}
    c. Synchronous session with Instructor, Tas (maximum 30 min) for Q&A

Communication:

1. Office 365 Teams Channel for Project: Project Discussion Channel
    a. Use @mention to communicate with your group/ TA/ Instructor
2. For email communication
    a. Cc your team members, and TAs

Note: The requirements might change, and your team should welcome any changes in the requirements.