

CSCI 5409 (Adv. Topics in Cloud Computing)
Project Title: Secured Lightweight Tourism Application for Canada

Date Given: Monday, Jan 20, 2020

Due: Timeline of deliverables given on page #2

**** Late submissions are not accepted and will result in a '0' on the deliverables**

Grading Scheme:

- Project Feasibility Report: 10%
Document will be evaluated based on the following:
 - Completeness of the feasibility report
 - Citations
 - Evidence of Group work. i.e. record of team members' contributions
- Project Design Report: 20%
Document will be evaluated based on the following:
 - Completeness of the design report
 - Citations
 - Evidence of Group work. i.e. record of team members' contributions
 - Proposed design and implementation techniques with justifications
 - Design pattern
 - Security of the system
 - Any specific novelty (such as, new authentication logic, optimization techniques etc.)
- Final Project Report: 20%
Document will be evaluated based on the following:
 - Completeness of the feasibility report
 - Citations
 - Evidence of Group work. i.e. record of team members' contributions
 - Implementation details
 - Evidence of testing (such as unit test, integration test, and validation test)
- Final Project Presentation: 50%
Each oral presentation will be evaluated based on the following:
 - Contributions of each member of a team during the presentation
 - Content of the presentation
 - Demonstration of working prototype
 - Presentation style, and verbal communication
 - Time management
 - Question/Answer session

Academic Integrity:

- This project requires group work. Therefore, each group is expected to complete their work by themselves. Collaboration with other groups is a violation of the academic integrity policy and will be reported to the AIO.
- Do not copy texts verbatim from online or printed materials
- Do not copy texts from other's work
- Do not submit other's work
- If you obtain help from Tutor(s), please acknowledge
- Provide citation for texts, images, tables, data etc.
- 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

1. Project Specification:

Hypothetical Scenario: *DalCloud5409* a hypothetical software company that provides cloud based software solutions for various organizations. In addition, *DalCloud5409* has recently received a \$60,000 fund to develop a tourism app for Canada. The app is accessible through mobile devices, and computers. Some of the primary requirements of this app are: security (2 factor authentication), accuracy, efficiency, and responsiveness.

The mobile app should allow the user to perform search of tourist destinations, and book a ticket. At initial stage the apps will allow information search on provincial/national parks, major beaches, and important cities. The app should highlight key features of the searched location(s). If a user wants to book a bus ticket, the app asks the user to create an account and places the order once the payment is done. A sample ticket is generated.

Key Components:

1. A simple thin client mobile apps (All functionalities must be performed at Cloud end)
2. Cloud-based Server Application with business logic
3. A backend cloud server with cloud instance(s), load balancing etc.
4. Database (if required) – SQL and/or NoSQL based architecture
5. Any additional tool(s)/ software that is required for communication and analysis

Functional Requirements:

1. Thin mobile client, which does not perform any major tasks in the mobile device. It should allow communication to the server, user query submission, and display of reports (e.g. ticket, analytics etc.)
2. All validations (e.g. user authentication etc.) must be performed at the server side
3. Optimum resource utilization (e.g. instance creation when needed, scaling containers etc.) must be performed. Otherwise, points will be deducted
4. Each group needs to design the system using containerization
5. The project must be divided into multiple loosely coupled modules
6. Each module must be tested before integration
7. All data must be transmitted and stored securely
8. Add a dummy card 1111-1111-1111-1111 for transactions, and perform validation

Information about Groups and AWS Account:

1. Each group will have 4 members.
2. All groups are formed by the instructor by randomly allocating group members
3. All members will receive the AWS account with \$40 credit. The account joining link and descriptions will be shared during the lecture and in email
4. The group members should share the account details within the group and develop each module using a different account.
5. Since the account credit is fixed, the cloud resources must be used in an efficient and optimized manner
6. Do not perform the same computation(s) or task(s) from multiple accounts. If you or your group member(s) run out of credit due to poor utilization of resources, unwanted services, or keeping unused open instances etc., then your points will be deducted

2. Project Timeline:

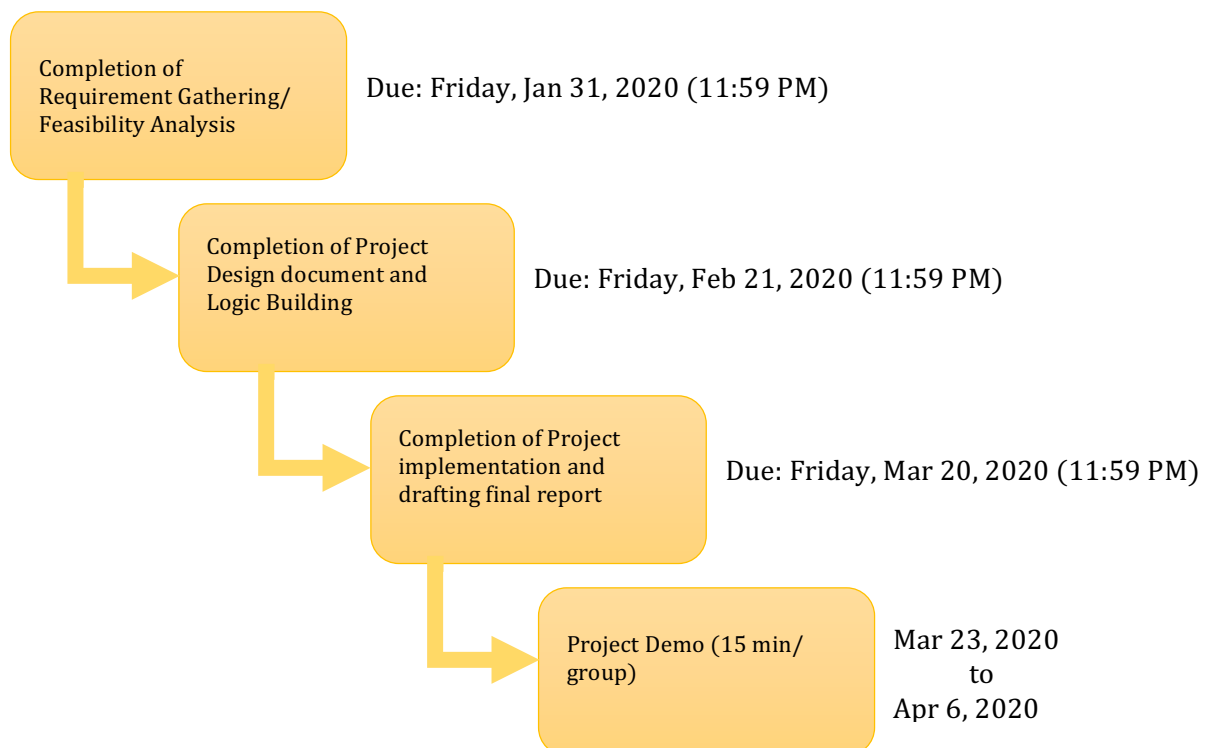


Figure 1: Timeline of deliverables

3. Required Documents:

1. Project Feasibility Study Report:

The project feasibility is used to check your understanding of the project, and cloud environment. The feasibility study report should be 4 - 5 pages in length (11-point font) and be divided into the following sections:

- **Title Page:** This includes the title of your project and the group members.
- **Project Requirement:** A brief paragraph, which highlights the key requirements of the project. A group needs to prepare this part based on the project specification given by the instructor.
- **Feasibility Study:** In this part, each group needs to add an architecture of the system. In addition, you need to identify the cloud services, methods, programming environment, computing resources etc. required for the project. Furthermore, you need to identify challenges, and possible solutions.

2. Project Design Document:

The design document should contain the problem statement, and details of the modules related to the project. The design document should be 10 - 12 pages (11-point font) and be divided into the following sections:

- **Title Page:** This includes the title of your project and the group members.
- **Project Requirement:** A brief paragraph, which highlights the key requirements of the project. A group needs to prepare this part based on the project specification given by the instructor.
- **Design:** In this part, each group needs to add the design specification of their project. Each group needs to come up with a solution using the concepts taught in the lecture, and other standard approaches. The project must be decomposed into multiple smaller meaningful modules. The design document should capture the details of module design, and justification. Apart from modules, the design document should contain flow charts, system/data models, detailed architecture, activity diagram etc. Furthermore, the document should provide information on how services will be created or invoked in the cloud platform, and what will be the approach to achieve optimization, and robustness.

3. Final Project Document:

The project report should be written like a technical manual for the system. It should contain a brief section of feasibility, and final design (2 pages). The rest of the report should contain implementation details, testing, analysis, limitations, and future scope. The project report should be of 10 - 12 pages (11-point font) and be divided into the following sections:

- **Title Page:** This includes the title of your project and the group members.
- **Project Requirement:** A brief paragraph, which highlights the key requirements of the project. A group needs to prepare this part based on the project specification given by the instructor.
- **Job Role:** There should be a section in the report, which highlights the detailed job roles of each member or contribution of each member in the project group.
- **Knowledge Sharing:** This section should contain information on how the team members have shared knowledge, and how they have leaned or worked on another module.
- **Completion Report:** A discussion of what was accomplished and what was not. Discuss milestones that were not accomplished and explain why they were not achieved. Identify and discuss any other interesting challenges (and possible solutions) that you encountered in the implementation. This section can be written as a technical report, where a group can add implementation details, such as how a cloud function is written, and how robustness is achieved etc.
- **Future Work:** A discussion of how the implementation can be extended or improved if you had more time and inclination to do so.