# CST-323 CLC Project Guide

**Note:** This project will be developed within Collaborative Learning Communities (CLC).

## Contents

© 2023. Grand Canyon University. All Rights Reserved.

## Project Overview

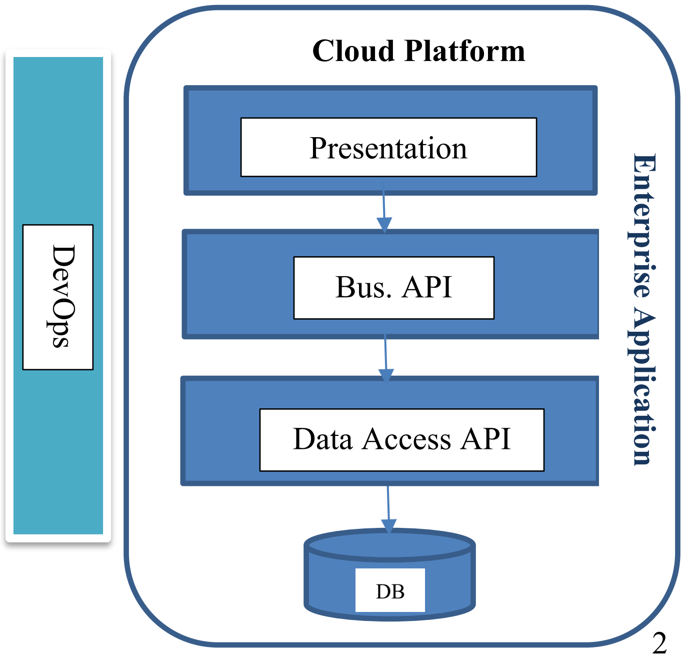
In this course, students will design and build a complete enterprise class N-Layer web application using a desired web stack that can include any of the following: Spring Framework, Laravel Framework, Spring Boot, Express/Angular, Express/React, or .NET MVC Framework (using .NET Core). Keep in mind that the web stack chosen may or may not be supported by all Cloud Providers used in this course. It is highly recommended to simply use the Spring Framework or PHP Laravel Framework (if you are in the BSCP program) or the Spring Boot Framework (if you in the BSSD program) since the activities used in the course will be based on one of these web frameworks and both of these frameworks are widely supported by all the Cloud Platforms being used in the course.

The enterprise application being designed and built must adhere to a specified set of high-level functional and technical requirements, as shown in the figure below. Students will have the freedom to pick the application they want to design and develop. They will work in teams of no more than two unless there is an odd number of students, in which case a team of three will be acceptable. Example applications could include, but are not limited to, an eCommerce application, Customer Management application, Order Management application, Blog Site application, DVD/Book/Music Management application, IoT application, and so forth.

Remember, this course is about learning Cloud Computing Concepts and various Cloud Platforms so you should pick an application that is relatively easy to design and build. This is not a course to learn how to design and build a complex web application. The application must be demonstrable on your local laptop after Topic 2 and be able to be deployable to various Cloud Platforms during Topics 3, 4, and 5.

Each Milestone delivery will include a Design Report that captures the technical approach, design decisions, UI wireframe designs, Sitemap designs, ER database designs, and project risks/issues. All code developed during the project will be maintained in a GIT-based Version.

Students will need approval for the project chosen from the instructor prior to moving onto their detailed design and development of the application.



- Presentation Services (MVC Framework)
- Business Services (SpringBeans, WCF, PHP)
- Persistence Services (Spring Data, PDO, ADO)
- Database (MySQL, PostgreSQL, SQL Server)
- Design Patterns:
  - N-Layer
  - MVC
  - DAO
  - DI
  - Interceptor
- Cloud Platform (one of the following):
  - Azure, AWS, Google, Heroku
- DevOps:
  - Logging
  - Monitoring (Availability, Log Alerts)
  - Build Pipeline (for application code)

The design and code must support the following high-level requirements:

- The application must be designed using an N-Layer architecture with distinct and separate presentation components, business services, and persistence services.
- The application must adhere to industry best practices, exception handling, data validation, and error handling as discussed either in text book readings, prior lectures/courses in one's program, or provided as peer code review feedback.
- The web application must be written using one of the following web frameworks: Spring Framework, Laravel Framework, or .NET MVC Framework if you are in the BSCP program or the Spring Boot, Express/Angular, Express/React, or .NET Core MVC Framework if you are in the BSSD program. Any exception to the before mentioned web frameworks MUST be approved by the instructor. Keep in mind that the web stack chosen may or may not be supported by all Cloud Providers. As an option, students have the choice to integrate IoT Devices, such as a Raspberry Pi, into their design. However, this must be approved by the instructor.
- The application must support persisting business data in a relational database that includes either MySQL, PostgreSQL, or SQL Server or a non-relational database that includes MongoDB. Any exception to the before mentioned relational databases MUST be approved by the instructor.
- The application must use an appropriate persistence framework, such as Spring Data, Spring JDBC, JPA/Hibernate, PDO, ADO.NET, or .NET Entity Framework, to access the database. Any exception to the before mentioned persistence frameworks MUST be approved by the instructor.
- The application must be designed and leverage the following design patterns if supported by the chosen framework: N-Layer, MVC, DAO, Dependency Injection, and Interceptor.
- The application must be deployed on one of the following Cloud Platforms: Azure, AWS, Google, or Heroku. There will be no exceptions to the before mentioned Cloud Providers and traditional application hosting providers, such as Hostable or GoDaddy, will not be permitted to be used for this project.
- The application must demonstrate a responsive design using the Bootstrap framework.
- The application must support the following minimal set of DevOps Principles including the following:
    - Use of a logging framework, such as SL4J, Log4j, Log4n, or jsLog that is appropriate for your application framework.
    - Use of log statements for entry to and exit from all major controller, business service, and data service methods.
    - Use of a cloud-based logging provider, such as Loggly.
    - Use of application availability provider, such as Uptime Robot.
    - Monitoring of log files and application availability.

  - Setup of an automated build pipeline for your application code.
- The application code must be fully documented use an appropriate programming language commenting format specification, such as JavaDoc, NDoc, PhpDoc, JSDoc.

**Project Milestones:**

The CLC team project will be designed and built using an iterative approach and delivered using the following five milestones. It should be noted that all milestones will include a design report and except for Milestone 1, an application code will be delivered to support the appropriate milestone requirements. It is expected that the code will be refactored during each iteration based on peer code review or instructor feedback.

**Milestone 1**

- Project proposal: Proposed application concept, web framework, cloud provider, high-level solution design, and draft division of work across team. **Note:** No application or database code is expected for this milestone.

**Milestone 2**

- Iteration 1 of application running locally on a laptop
- Design report

**Milestone 3**

- Iteration 2 of application running in the cloud
- Updated design report

**Milestone 4**

- Iteration 3 of application running in the cloud with DevOps
- Updated design report

**Milestone 5**

- Final application code (with JavaDoc and necessary code refactoring)
- Final project presentation
- Final design report

## Milestone 1: Project Proposal

**Overview**

In this milestone, students will complete the project proposal.

**Execution**

Execute this assignment according to the following guidelines:

1. The team must provide a detailed write-up:
    a. Describing what domain and products will be managed by the application. The use of an IoT Device must be approved by the instructor.
    b. Describing the high-level features and functionality that will be supported by the application.
    c. Outlining the frameworks and technologies, along with the versions of the frameworks and technologies, to be used by the application.
2. The team should identify one or more desired cloud providers that will be used in the project. The team will be given a choice to finalize or change their choice of cloud providers after Milestone 2 if desired and within reason.
3. The team must provide an initial sitemap illustrating all the logical pages of the application and how they will interact with each other. The team can also optionally submit any user interface wireframe concepts or initial designs.
4. The team must provide a high-level breakdown of how the work will be planned, managed, and divided out among the team members. It is recommended to use a formal agile project delivery methodology for this project.
5. The team will document the risks (technical and functional) that need to be managed moving forward with the project.
6. It is expected that the team will meet with the instructor if a project cannot be identified.
7. The project must be approved by the instructor.

**Submission**

Submit the following to the digital classroom:

1. A Word document that provides a complete description of what domain and products will be managed by the application, the versions of the frameworks, and technologies to be used by the application, as well as the high-level features and functionality that will be supported by the application.
2. Initial design report (using the provided "Design Report Template," located within Class Resources) with the following sections completed:
    a. Cover page with list of tasks completed
    b. Planning documentation outlining how the project will be managed
    c. Design documentation
        1) General technical approach

2) Key technical decisions (with choice of cloud providers, frameworks, and technologies)
3) Risks
4) Draft sitemap diagram
5) Draft user interface diagrams
6) Draft ER database diagram

## Milestone 2: Cloud Test Application on Laptop

**Overview**

In this milestone, students will deliver a second build of the application to be deployed locally.

**Execution**

Execute this assignment according to the following guidelines:

1. Update the project design specifications (using the provided Design Report Template):
    a. Cover sheet
    b. Planning documentation
    c. General technical approach
    d. Key technical decisions (final cloud provider, frameworks, and technologies)
    e. Risks
    f. Sitemap diagram
    g. User interface diagrams
    h. ER database diagram and DDL script
    i. UML class diagrams (all controllers, models, business services, data services)
    j. URL of GIT repository
    k. URL of screencast
2. Deliver a second build of the application making sure:
    a. It is fully functional per the objectives outlined in project proposal.
    b. It is fully functional per the frameworks and technologies outlined in project proposal.
    c. It is fully functional running on local laptop.
    d. There is a screencast (5–8 minutes) demonstrating a code walk-through and all features.
3. Any major technical changes or a change in cloud providers must be approved by the instructor.

**Submission**

Submit the following to the digital classroom:

1. Design report (with URL to GIT repository and URL to screencast)
2. Zip file with all application code

## Milestone 3: Cloud Test Application on Cloud

**Overview**

In this milestone, students will deliver a third build of the application to be deployed to a cloud provider.

**Execution**

Execute this assignment according to the following guidelines:

1. Update the project design specifications within the design report:
   a. Cover sheet
   b. Planning documentation
   c. General technical approach
   d. Key technical decisions (final cloud provider, frameworks, and technologies)
   e. Risks
   f. Sitemap diagram
   g. User interface diagrams
   h. ER database diagram and DDL script
   i. UML class diagrams (all controllers, models, business services, and data services)
   j. URL of GIT repository
   k. URL of screencast
2. Deliver a third build of the application making sure:
   a. It is fully functional per the objectives outlined in project proposal.
   b. It is fully functional per the frameworks and technologies outlined in project proposal.
   c. It is fully functional running on the cloud.
   d. There is a screencast (5–8 minutes) demonstrating a code walkthrough and all features.

**Submission**

Submit the following to the digital classroom:

1. Design report (with URL to GIT Repository and URL to screencast)
2. Zip file with all application code

## Milestone 4: DevOps Integration

**Overview**

In this milestone, students will ensure their project has DevOps functionality.

**Execution**

Execute this assignment according to the following guidelines:

1. Update the design report (as needed):
    a. Cover sheet
    b. Planning documentation
    c. General technical approach
    d. Cloud Install instructions
    e. Key technical decisions (updated with logging, monitoring, and build pipeline)
    f. Risks
    g. Sitemap diagram
    h. User Interface Diagrams
    i. ER database diagram and DDL script
    j. UML class diagrams (all controllers, models, business services, data services)
    k. URL of GIT repository
    l. URL of screencast
2. Deliver the application making sure there is:
    a. Integration of a logging framework, such as SL4J, Log4j, Log4n, or MonoLog, with log statements for entry to and exit from all major controller, business service, and data service methods
    b. Demonstration of application availability monitoring using an availability provider, such as Uptime Robot
    c. Demonstration of an automated build pipeline for the application code
    d. Full functionality running on the cloud
    e. A screencast (5–8 minutes) demonstrating a code walk through and all features

**Submission**

Submit the following to the digital classroom:

1. Design report (with URL to GIT repository and URL to screencast)
2. Zip file with all application code

## Milestone 5: Final Project

**Overview**

In this milestone, students deliver their final design specifications.

**Execution**

Execute this assignment according to the following guidelines:

1. Finalize the design report:
    a. Cover sheet
    b. Planning documentation
    c. General technical approach
    d. Key technical decisions (with all frameworks, technologies, and DevOps supported)
    e. Risks
    f. Sitemap diagram
    g. User interface diagrams
    h. ER database diagram and DDL script
    i. UML class diagrams (all controllers, models, business services, data services)
    j. URL of GIT repository
    k. URL of screencast (5–8 minutes) demonstrating final application
2. Finalize the application making sure:
    a. Code comments in all classes and methods.
    b. The code is refactored as necessary.

**Submission**

Submit the following to the digital classroom:

1. Final design report (with URL to GIT Repository and URL to screencast)
2. Zip file of generated code documentation
3. Zip file with all final application code