**CST-361 CLC-Project Guide**

Contents

[Overview 2](#_Toc494358676)

[Milestone 1 3](#_Toc494358677)

[Milestone 2 5](#_Toc494358678)

[Milestone 3 6](#_Toc494358679)

[Milestone 4 8](#_Toc494358680)

[Milestone 5 10](#_Toc494358681)

[Milestone 6 12](#_Toc494358682)

[Milestone 7 13](#_Toc494358683)

# Overview

In this course, students will work in teams of two to design and build an Internet of Things (IoT) application as shown in Figure 1. The application will be designed to leverage eight common design patterns often found in an Enterprise Java application.

* MVC Design Pattern
* Façade Design Pattern
* DAO Design Pattern
* DTO Design Pattern
* Interceptor Design Pattern
* Dependency Injection Design Pattern
* Factory Design Pattern
* Singleton Design Pattern

Bus. API

REST API

HTTP

IoT Device

IoT Reporting App

DB

Figure 1 <insert title>

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

**Embedding**

The IoT application is required to generate data; therefore, students will be given the option to write an application to emulate the IoT embedded device functionality or if available, students can chose to use an actual embedded device, such as a Raspberry Pi with appropriate sensors. It recommended that students keep this simple, as the focus of this course is not on developing the IoT applications or working with embedded devices.

**Back End Service**

The IoT application is required to process and store the data; therefore, students will design a back-end service using Enterprise Java technologies for a REST-based API that will be consumed by the IoT embedded application. The REST API should not be anonymous and leverage at a minimum HTTP Basic Authentication for securing the REST API endpoint. The implementation of the REST API should simply be a façade over any business logic required to process the IoT data and store the data in a relational database. It is recommended that the students keep the data model design as simple as possible to meet the project requirements, as this course is not focused on database design. The JavaDB or MySQL database can easily suffice for the project.

**Front End Web Application**

The IoT application is required to tailor to the user; therefore, students will design a front-end web application using Enterprise Java technologies to implement a simple IoT Reporting application. The Reporting application will leverage a number of common design patterns and should provide a tabular data report as well as a visual chart based report. It will be important to research available open source charting libraries and/or JSF components that could be leveraged in the final solution.

**Project Management**

Each group will leverage and apply the Scrum methodology practiced in CST-247 to manage the delivery of the team project.

For a refresh on Scrum, review the following resources:

<http://www.scrumguides.org/>

<https://www.scrum.org/resources/what-is-scrum>

**Project Milestones**

The team project is designed and built using an iterative approach and delivered using the milestones outlined below. It should be noted that all milestones include a design report. However, the application code will be used in all milestones except for Milestones 1 and 2.

**Project Documentation**

Documentation of all technical decisions and technical designs will be via a formal design report that captures all appropriate UML diagrams, ER diagrams, UI designs, and other technical artifacts to support the design of the end-to-end solution and application. Refer to the "Design Report Template" located within the Course Materials for detailed instructions.

The Design Report at a minimum will contain the following technical elements:

1. Cover Page: Outline a summary of the project assignment objectives and team member tasks.
2. UML Diagrams: Use case diagrams for all requirements, component diagrams for solution, and class diagrams for all non-framework classes.
3. User Interface Designs: Use wireframe designs for all screens.
4. Database Design: Use an ER diagram.
5. Test Plan: Test cases for all functionality.

# Milestone 1

The focus of Milestone 1 is on the initial planning and upfront design work using UML diagrams. To complete this milestone refer to the guidelines below:

* Using an Agile methodology like Scrum, identify how the team will be organized, when team meetings will be held, when peer reviews will be held, project requirements, and what components of the project team members will work on. Ensure the creation of a milestone delivery schedule as well.
* Research existing IoT applications.
* Identify the IoT application (i.e., weather, motion, location) to be designed, the IoT embedded device (emulated or Raspberry Pi) to be used, and the deployment strategy that will be taken for the IoT embedded application.
* Draw a draft UML use case diagram for a potential IoT device and the end user.
* Draw a draft UML component diagram depicting a potential logical block diagram for an IoT solution.
* Draw a UML class diagram modeling the attributes for a potential IoT device that is capturing data such as weather, motion, location, etc.
* Draw a potential UML deployment diagram.
* Create the initial design report capturing the above design elements.

**Deliverable**

Initial Design Report (no code is expected to be delivered)

***Requirements***

**Cover Sheet** (to include the approach the team will take in completing the project)

**Technical Elements**

Requirements will be captured using UML use case diagrams.

Solution will be captured using UML class and component diagrams.

Deployment strategy will be captured using UML deployment diagram.

**Non-Technical Elements**

Planning and delivery methodology

Possible unknowns or risks identified for the project

IoT data that will be measured and reported (requires instructor approval)

IoT device that will be used in the solution (requires instructor approval)

*Performance Level Ratings*

|  |  |
| --- | --- |
| **Meets Expectations** | Performance consistently met expectations in all essential areas of project construction, at times possibly exceeding expectations, and the quality of work overall was very good. The most critical goals were met. |
| **Near Expectations** | Performance did not consistently meet expectations. Performance failed to meet expectations in one or more essential areas of project construction and/or recording, one or more of the most critical goals were not met. |
| **Below Expectations** | Performance was consistently below expectations in most essential areas of project construction and/or recording, reasonable progress toward critical goals was not made. Significant improvement is needed in one or more important areas. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Below Expectations** | **Near Expectations** | **Meets Expectations** | **Earned** |
| Identify how the team will be organized. | 0 pts – 6 pts | 7 pts – 9 pts | 10 pts |  |
| Research IoT Application. | 0 pts – 9 pts | 10 pts – 14 pts | 15 pts |  |
| The team develops the initial design report. The documentation is well presented and includes all technical and non-technical elements. | 0 pts – 9 pts | 10 pts – 14 pts | 15 pts |  |
| Writer is clearly in command of standard, written, academic English. Prose is largely free of mechanical errors. | 0 pts – 6 pts | 7 pts – 9 pts | 10 pts |  |
| **TOTAL** |  |  |  | **/50** |
| **Instructor Feedback** | | | | |

# Milestone 2

The focus of Milestone 2 is the continuation of the planning and upfront design work, as well as finalizing the IoT embedded application and back-end web application. To complete this milestone refer to the guidelines below:

* Update the project management goals, objectives, and tasks.
* Ensure instructor approval for the IoT application, IoT embedded device, and the deployment strategy.
* Update the UML diagrams for use cases, classes, and deployment.
* Update the UML component diagram of the complete end-to-end solution.
* Create the initial user interface wireframe designs for the IoT reporting application (draft of user interface web pages).
* Create the initial database design and object models to support the IoT data and reporting.
* Update the design report initially started.

**Deliverable**

Design Report (no code is expected to be delivered)

*Requirements*

**Cover Page**

**Technical Elements**

Requirements will be captured using UML use case diagrams.

Solution will be captured using UML class and component diagrams.

Object model will be captured using UML class diagram.

Deployment strategy will be captured using UML deployment diagram.

Initial user interface will be captured using a wireframe diagram.

Initial database design will be captured using an ER database diagram.

**Non-Technical Elements**

Planning and delivery methodology

Possible unknowns or risks identified for the project

*Performance Level Ratings*

|  |  |
| --- | --- |
| **Meets Expectations** | Performance consistently met expectations in all essential areas of project construction, at times possibly exceeding expectations, and the quality of work overall was very good. The most critical goals were met. |
| **Near Expectations** | Performance did not consistently meet expectations. Performance failed to meet expectations in one or more essential areas of project construction and/or recording, one or more of the most critical goals were not met. |
| **Below Expectations** | Performance was consistently below expectations in most essential areas of project construction and/or recording, reasonable progress toward critical goals was not made. Significant improvement is needed in one or more important areas. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Below Expectations** | **Near Expectations** | **Meets Expectations** | **Earned** |
| Approved IoT Application | 0 pts – 6 pts | 7 pts – 9 pts | 10 pts |  |
| The team updates the design report. The documentation is well presented and includes all technical and non-technical elements. | 0 pts – 20 pts | 21 pts – 29 pts | 30 pts |  |
| Writer is clearly in command of standard, written, academic English. Prose is largely free of mechanical errors. | 0 pts – 6 pts | 7 pts – 9 pts | 10 pts |  |
| **TOTAL** |  |  |  | **/50** |
| **Instructor Feedback** | | | | |

# Milestone 3

The focus of Milestone 3 is on applying the MVC and façade design patterns to develop an IoT transactional business application using Enterprise Java technologies. To complete this milestone refer to the guidelines below:

* Update the project management goals, objectives, and tasks.
* Design and implement iteration 1 of the IoT embedded (or emulated) application.
* Design and implement iteration 1 of the IoT reporting application.
* Update the UML diagrams for use cases, applicable classes, deployment, and component.
* Update the wireframe designs.
* Update the ER database design.
* Complete a test plan. The test plan is a sequence of steps the developer will perform to validate that application functionality is working properly to the business requirements and technical design. Refer to the "Test Plan Template," located in the course materials when completing this step.
* Update the design report from Milestone 2.

*Code Requirements:*

* IoT Embedded Application:
  + Application that obtains or generates IoT data identified in Milestone 1.
  + Application can emulate an embedded device and be written as a console based Java application. The IoT data generated should represent a real world IoT device.
  + Optional use of an actual embedded device, such as a Raspberry Pi with appropriate sensors, to generate the IoT data. The application can be written in Python or other programming language supported by the Raspberry Pi.
* Front end Web Reporting Application:
  + The reporting application will contain login and registration screens that leverages a relational database for authentication. All screens should be written JavaServer Faces demonstrating the use of the MVC design pattern and façade design pattern. The JavaDB or MySQL database can be used as a database for the project. Students can write their persistence code using either JDBC or JPA.
  + The reporting application should provide a tabular data report to display the captured IoT data.
  + The reporting application should provide a visual chart report to display the captured IoT data. The students should research available open source charting libraries and JSF components for use in their final solution.

**Deliverables**

Iteration 1 implementation of the IoT Embedded application

Iteration 1 implementation of the IoT Reporting application

Updated Design Report

*Performance Level Ratings*

|  |  |
| --- | --- |
| **Meets Expectations** | Performance consistently met expectations in all essential areas of project construction, at times possibly exceeding expectations, and the quality of work overall was very good. The most critical goals were met. |
| **Near Expectations** | Performance did not consistently meet expectations. Performance failed to meet expectations in one or more essential areas of project construction and/or recording, one or more of the most critical goals were not met. |
| **Below Expectations** | Performance was consistently below expectations in most essential areas of project construction and/or recording, reasonable progress toward critical goals was not made. Significant improvement is needed in one or more important areas. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Below Expectations** | **Near Expectations** | **Meets Expectations** | **Earned** |
| IoT Embedded Application | 0 pts – 13 pts | 14 pts – 19 pts | 20 pts |  |
| IoT Reporting Application | 0 pts – 13 pts | 14 pts – 19 pts | 20 pts |  |
| The team updates the design report. The documentation is well presented and includes all technical and non-technical elements. | 0 pts – 9 pts | 10 pts – 14 pts | 15 pts |  |
| Writer is clearly in command of standard, written, academic English. Prose is largely free of mechanical errors. | 0 pts – 9 pts | 10 pts – 14 pts | 15 pts |  |
| **TOTAL** |  |  |  | **/70** |
| **Instructor Feedback** | | | | |

# Milestone 4

The focus of Milestone 4 is on applying the DAO and DTO design patterns to design and develop an IoT transactional business application built using Enterprise Java technologies. To complete this milestone refer to the guidelines below:

* Update the project management goals, objectives, and tasks.
* Complete iteration 2 of the IoT embedded (or emulated) application
* Design and implement the IoT REST API to be consumed by the embedded application.
* Complete iteration 2 of the IoT reporting application.
* Update the UML diagrams for use cases, applicable classes, deployment and component.
* Update the wireframe designs.
* Update the ER database design.
* Complete the REST API design.
* Update the test plan.
* Update the design report from Milestone 3.

*Code Requirements:*

* IoT Embedded Application:
  + Finish integration with the REST API.
* Back-End Service REST API:
  + REST-based service that will be consumed by the IoT embedded application
  + REST-based service that uses JSON as a data transport format and demonstrate the use of the data transfer object design pattern. Students should leverage JAX-RS for the REST based service.
  + REST-based service that is not anonymous and leverages at a minimum HTTP Basic Authentication for securing the REST API endpoint.
  + REST-based service should demonstrate the application of the façade design pattern and the DTO design pattern as well as implement any business logic required to process the IoT data as well as interface to data access objects that store the data in a relational database.
  + REST based service should demonstrate the application of the data access design pattern to store the data in a relational database.
  + The JavaDB or MySQL database can be used as a database for the project. Students can write their persistence code using either JDBC or JPA.
* Front-End Web Reporting Application:
  + Re-factor the reporting application to use the DAO design pattern. The JavaDB or MySQL database can be used as a database for the project. Students can write their persistence code using either JDBC or JPA.
  + The reporting application should provide a tabular data report to display the captured IoT data.
  + The reporting application should provide a visual chart report to display the captured IoT data. The students should research available open source charting libraries and JSF components for use in their final solution.

**Deliverables**

Iteration 2 implementation of the IoT Embedded application

Iteration 1 implementation of the IoT REST API

Iteration 2 implementation of the IoT Reporting application

Update Design Report

*Performance Level Ratings*

|  |  |
| --- | --- |
| **Meets Expectations** | Performance consistently met expectations in all essential areas of project construction, at times possibly exceeding expectations, and the quality of work overall was very good. The most critical goals were met. |
| **Near Expectations** | Performance did not consistently meet expectations. Performance failed to meet expectations in one or more essential areas of project construction and/or recording, one or more of the most critical goals were not met. |
| **Below Expectations** | Performance was consistently below expectations in most essential areas of project construction and/or recording, reasonable progress toward critical goals was not made. Significant improvement is needed in one or more important areas. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Below Expectations** | **Near Expectations** | **Meets Expectations** | **Earned** |
| IoT Embedded Application | 0 pts – 6 pts | 7 pts – 9 pts | 10 pts |  |
| IoT REST API | 0 pts – 6 pts | 7 pts – 9 pts | 10 pts |  |
| IoT Reporting Application | 0 pts – 13 pts | 14 pts – 19 pts | 20 pts |  |
| The team updates the design report. The documentation is well presented and includes all technical and non-technical elements. | 0 pts – 13 pts | 14 pts – 19 pts | 20 pts |  |
| Writer is clearly in command of standard, written, academic English. Prose is largely free of mechanical errors. | 0 pts – 6 pts | 7 pts – 9 pts | 10 pts |  |
| **TOTAL** |  |  |  | **/70** |
| **Instructor Feedback** | | | | |

# Milestone 5

The focus of Milestone 5 is on applying the interceptor, dependency injection, factory, and singleton design patterns to solve a real business problem and thru design and development of an application built using Enterprise Java technologies. To complete this milestone refer to the guidelines below:

* Update the project management goals, objectives, and tasks
* Complete iteration 2 of the IoT embedded (or emulated) application
* Complete iteration 2 of the IoT REST API
* Enhance the IoT Reporting application using additional design patterns
* Complete iteration 2 of the IoT reporting application
* Update the UML diagrams for use cases, applicable classes, deployment, and component
* Update the wireframe designs
* Update the ER database design
* Update the test plan
* Update the design report from Milestone 4.

*Code Requirements:*

* Front end Web Reporting Application:
  + For monitoring purposes, add a method interceptor to demonstrate the use of the interceptor design pattern to log all method entry and exit in each EJB in the application. This should be done using the standard Enterprise Java CDI framework.
  + For making application loosely coupled, inject all instances of all EJB's into their respective dependent components, to demonstrate the use of the dependency injection design pattern. This should be done using the standard Enterprise Java CDI framework.
  + In support of the interceptor, which will log all method entry and exit points, a logging service should be designed that is implemented using a singleton design pattern.
  + The factory design pattern should be used to create and instantiate all object model and DTO classes used by the REST API and reporting application.

**Deliverables**

Iteration 2 implementation of the IoT Embedded application

Iteration 2 implementation of the IoT REST API

Iteration 2 implementation of the IoT Reporting application

Updated Design Report

*Performance Level Ratings*

|  |  |
| --- | --- |
| **Meets Expectations** | Performance consistently met expectations in all essential areas of project construction, at times possibly exceeding expectations, and the quality of work overall was very good. The most critical goals were met. |
| **Near Expectations** | Performance did not consistently meet expectations. Performance failed to meet expectations in one or more essential areas of project construction and/or recording, one or more of the most critical goals were not met. |
| **Below Expectations** | Performance was consistently below expectations in most essential areas of project construction and/or recording, reasonable progress toward critical goals was not made. Significant improvement is needed in one or more important areas. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Below Expectations** | **Near Expectations** | **Meets Expectations** | **Earned** |
| IoT Application & Services | 0 pts – 13 pts | 14 pts – 19 pts | 20 pts |  |
| IoT Reporting Application | 0 pts – 13 pts | 14 pts – 19 pts | 20 pts |  |
| The team updates the design report. The documentation is well presented and includes all technical and non-technical elements. | 0 pts – 13 pts | 14 pts – 19 pts | 20 pts |  |
| Writer is clearly in command of standard, written, academic English. Prose is largely free of mechanical errors. | 0 pts – 6 pts | 7 pts – 9 pts | 10 pts |  |
| **TOTAL** |  |  |  | **/70** |
| **Instructor Feedback** | | | | |

# Milestone 6

The focus of Milestone 6 is on code refactoring and cleanup. To complete this milestone refer to the guidelines below:

* Update the project management goals, objectives, and tasks
* By design there are no new functional requirements being added to the application
* The students should use this time to refactor and clean up their code based on peer reviews and instructor feedback
* The students should use this time to clean up and finalize their design report. Their design is meant to represent all technical elements of their finished project
* The students should also use this time to start preparing for the project presentation
* The students should prepare a strategy for how to test their complete application and write all the test cases to support the testing of their complete application

**Deliverables**

Updated Design Report

Test Plan and Test Cases

*Performance Level Ratings*

|  |  |
| --- | --- |
| **Meets Expectations** | Performance consistently met expectations in all essential areas of project construction, at times possibly exceeding expectations, and the quality of work overall was very good. The most critical goals were met. |
| **Near Expectations** | Performance did not consistently meet expectations. Performance failed to meet expectations in one or more essential areas of project construction and/or recording, one or more of the most critical goals were not met. |
| **Below Expectations** | Performance was consistently below expectations in most essential areas of project construction and/or recording, reasonable progress toward critical goals was not made. Significant improvement is needed in one or more important areas. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Below Expectations** | **Near Expectations** | **Meets Expectations** | **Earned** |
| Test Plan and Test Cases | 0 pts – 13 pts | 14 pts – 19 pts | 20 pts |  |
| Execute Test Plan | 0 pts – 13 pts | 14 pts – 19 pts | 20 pts |  |
| The team updates the design report. The documentation is well presented and includes all technical and non-technical elements. | 0 pts – 13 pts | 14 pts – 19 pts | 20 pts |  |
| Writer is clearly in command of standard, written, academic English. Prose is largely free of mechanical errors. | 0 pts – 6 pts | 7 pts – 9 pts | 10 pts |  |
| **TOTAL** |  |  |  | **/70** |
| **Instructor Feedback** | | | | |

# Milestone 7

The focus of Milestone 7 is on the final project presentation and final project submission. To complete this milestone refer to the guidelines below:

* By design there are no new functional requirements being added to any application.
* The project presentation should mimic that of a Technical Design review given in the industry.
* The project screencast should support the following requirements:
* There were 9 design patterns used in the project. For each design pattern:
  + Verbally describe WHAT the design pattern is and provide a description of the pattern
  + Verbally walkthrough HOW and WHERE the design pattern was implemented in the code
  + If necessary, create more than one screencast
* Final end to end demo of ALL application functionality:
  + Running IoT Embedded Application
    - Posting and logging of IoT Data to the Server
    - Error condition when Server database is down
  + Running IoT Reporting Application:
    - Register new user (also validating the same user cannot be registered twice)
    - Login new or existing user (validating good and bad credentials)
    - Logout
    - Tabular IoT Data Report
    - Chart IoT Data Report
    - Error condition when Server database is down
* All team members MUST present in the screencast

**Deliverables**

Final Project Presentation.

Final Project Screencast.

**Final Project Scoring Guide**

This assignment uses a scoring guide. Please review the scoring guide prior to beginning the assignment to become familiar with the expectations for successful completion.

**Final Project Presentation Rubric**

This assignment uses a rubric. Please review the rubric prior to beginning the assignment to become familiar with the expectations for successful completion.