**CST-361 CLC-Project Guide**

Contents

[Overview 2](#_Toc72484197)

[Milestone 1 3](#_Toc72484198)

# 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** | | | | |