

## Project 2: Control IoT Simulator via Web Pages

### I Overview

In this project, we will work on features that allow us to control the IoT simulator via web pages using the HTTP protocol. For actual smart plugs, these features make it possible to turn a light on or off just by using a browser on your smartphone without the need to install any apps. You may access the web page following *Section III.C of the Project Instructions* but please keep in mind the web page may or may not work depending on your progress of this project.

User story is a widely used tool for requirements analysis in Agile software development. Each user story uses a short and informal piece of natural language to describe one or more desired features for both end users and developers. Usually, one organizes a user story using the template “As a ... (who), I want ... (what), so that ... (why).” Using such a template will help to meet the so-called SMART criteria for the user story to be specific, measurable, achievable, relevant, and time-bound. End users prefer to communicate using natural language but too much informal description makes developers difficult to evaluate what should be done – user stories achieve a balance between the two.

You will find the user stories describing the requirements in the next section. You will implement the features in the `HTTPCommands` class so please first read its source code. Pay attention to the method `report` as you could call this method to generate web pages. Locate the comment starting with `// P2:` and your code should be implemented there.

Once you have some rough ideas on how to implement the features, you should start to create unit tests in a file like `src/test/java/ece448/iot_sim/HTTPCommandsTests.java`. Follow the red-green cycle to add unit tests and to implement desired features.

### II.

#### User Stories

1.

#### Plug Report

As an end-user, I want to access the web page containing a report of the plug with the name “plugName” at the path `/plugName` without any query string, so that I can view the report using a browser. The report should include whether the switch is on and its power reading.

2.

#### Toggle or Switch a Plug On/Off

As an end-user, I want to control the plug with the name “plugName” at the path `/plugName` with a query string, so that I can control the plug using a browser. To toggle a plug, the query string is `action=toggle`. To switch a plug to be on, the query string is `action=on`. To switch a plug to be off, the query string is `action=off`.

3.

#### Control Feedback

As an end-user, I want to receive the up-to-date report as the response to the path /plugName with a query string, so that I can verify that the plug acts properly.

III.

#### Testing Procedures

The testing procedures are implemented in `ece448.grading.GradeP2`. It should be straightforward to verify all user stories are covered.