CSSE 477 – Milestone 4 – Application Server

You have built a web application server and improved several of its quality attributes in the previous milestones. In this milestone, you will develop an application that will run on your application server.

Requirements

To encourage creativity, I am not constraining your application with a lot of requirements. Here are the base requirements:

Server Side Requirements

Your application must be implemented as a plugin and have a minimum of **five** servlets. There must be at a bare minimum **one servlet each** for GET, POST, PUT, and DELETE requests and **at least one extra servlet** that may handle any of the four methods (your choice).

Client Side Requirements

The client side must be implemented using HTML and Java Script (particularly, Ajax). Here is an Ajax tutorial that you can refer to: http://www.w3schools.com/ajax/. You may also use other Javascript frameworks that provide higher-level API for doing Ajax, such as jQuery, Bootstrap, and Angular.

Use of CSS is encouraged. The data transport within HTTP can be done using JSON, XML, YAML, or GPB. In general, it is easier to work with JSON when using Javascript.

Deliverables

Report (pdf)
Your report must have the following sections:

1. Title Page

2. Change History

    You will **update the MS3 report**. Please append the changes to the **Change History** section to reflect changes in this version.

3. Architecture and Design

    3.1 Updated architecture diagram.
    3.2 Updated detailed design and a brief explanation.

4. Tactics/Feature Listing

    Show the cumulative work assignment so far in the project including the work for this milestone (who did what)

5. Architectural Evaluation and Improvements

    Show any improvements you made to the product along these dimensions (usability, in this milestone.

6. Sample Application: Database of Secret Service Agents (an example)

6.1 **About the Application**

Provide a brief summary of what the application does.

6.2 **API Design** (Read http://codeplanet.io/principles-good-restful-api-design/)

6.2.1 F1 - Retrieving a list of agents

Method: GET

URI: /agentdb/v1/agents

Request Body:

<none>

Response Body:

```
{
        “code”: 200,                    [NOTE: This could be app specific also]
        “message”: “Ok”,
        “payload”:
        [
                { “id”: 7, “name”: “James Bond”},
                { “id”: 8, “name”: “Jason Bourne”}
        ]
}
```

Development Status: One of TODO, DOING, DONE.

6.2.2 F2 - Retrieving a user

Method: GET

URI: /agentdb/v1/ agents/7

Request Body:

<none>

Response Body:

```
{
        “code”: 200,
        “message”: “Ok”,
        “payload”: { “id”: 7, “name”: “James Bond”}
}
```

OR

```
{
        “code”: 404,
        “message”: “Not Found”
}
```

Development Status: One of TODO, DOING, DONE.

6.2.3 F3 - Creating a new user

Method: POST

URI: /agentdb/v1/users/

Request Body:

```
{ “name”: “Ethan Hunt”}
```

Response Body:

```
{
        “code”: 201,
        “message”: “Created”,
        “payload”: { “id”: 9, “name”: “Ethan Hunt”}
}
```

Development Status: One of TODO, DOING, DONE.

You may use one of the following libraries to save time parsing XML, JSON, and other formatted strings to Java objects:

From XML or JSON to Java: XStream (http://xstream.codehaus.org/)
From JSON to Java: Gson (https://github.com/google/gson)
From YAML to Java: YamlBeans (http://yamlbeans.sourceforge.net/)
From GPB to Java: https://developers.google.com/protocol-buffers/docs/javatutorial

7. Future Improvements

Identify places where your project can be improved further and how to go about doing that.

Code (zip)

Bundle the project and turn-it in on Moodle.

Presentation (slides in **pdf)**

You will be given **five minutes each** (**firm**) to present your overall design of the project and to demo your application to the class (Monday-Tuesday, May 18-19, 2015). The quality of your overall design (architecture + class diagram), the sample application, and presentation will be evaluated by the entire class including your instructor. The presentation will contribute 15% to your MS4 grade. Turn in the pdf of your presentation on Moodle by Monday, May 18, 2015, 8:00 am.

Grading

You will be peer-reviewed by another team in class on Monday the 18th. You will swap your code and use the following rubrics to evaluate each other's work: 0 – No Try, 1 – Not Enough Thought or Implementation, 2 – Some Thought or Some Implementation, 3 – Developed Idea or Near Completion Implementation, 4 – Well-Developed Idea or Completely Implemented, 5 – Elegant or Creative Work. You must have your test machine/laptop ready for the live demo.

S.No	**Description**	**Ratings**
1	Application provides GET	0-5
2	Application provides POST	0-5
3	Application provides PUT	0-5
4	Application provides DELETE	0-5
5	Application provides at least one additional servlets	0-5
6	Application uses HTML and Ajax effectively (or other JS frameworks)	0-5
7	The design of API is well documented and matches implementation	0-5
8	Code quality	0-5
9	Overall quality of the project	0-5

Timeline

On or before Friday, May 15, 2015, 5:10 pm – Please verify your plan for application with the instructor. You must bring your report with all of the details completed for **Section #6 (API Design)** in the meeting. You must also show **a simple**

application that makes AJAX call to your servlet. It could be as simple as your servlet returning the “Hello World!” string as a result of Ajax call that gets displayed on the browser.

Monday, May 18, 2015, 8:00 am – Please turn in your report (pdf), your presentation slides (pdf), and the source code (zip) files on Moodle. We will conduct project evaluations during the class on Monday.

Tuesday, May 19, 2015, During Class – We will start the presentations on Monday. We will continue with the presentations during the Tuesday’s class. **Attendance is mandatory for both Monday and Tuesday classes.**