**Team Number: 1 Date: 4/1/2016**

# Lab Exercise: JSP & MVC

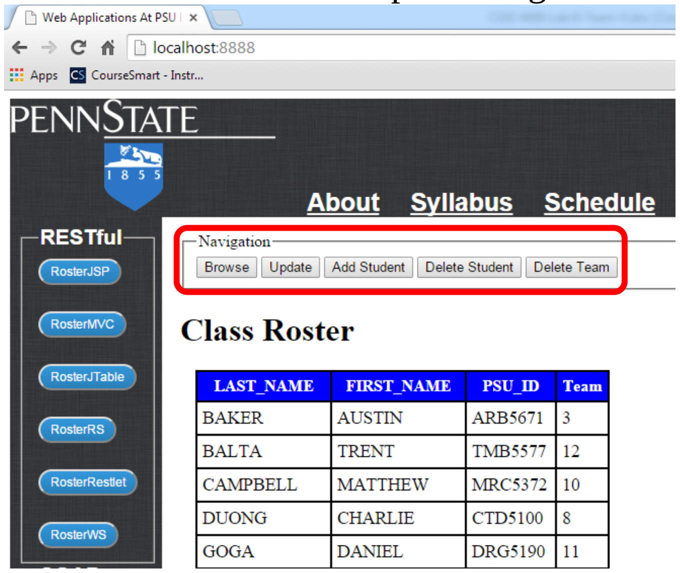
## Lab Objectives

1. In Lecture 21, we discussed the project BankMVC, which has a good architecture but can only handle doGET. In Lecture 22, we discussed the project predictionsREST, which can handle CRUD requests but has no pages for a client to submit complex request (we used curl instead). The project you build in this lab should combine the benefits of both the projects we have covered.
2. MVC pattern
    - HttpServlet as controller: CRUD operations
    - JSP pages as views
    - JavaBeans as models
3. GlassFish as service container.

## Description of the Problem

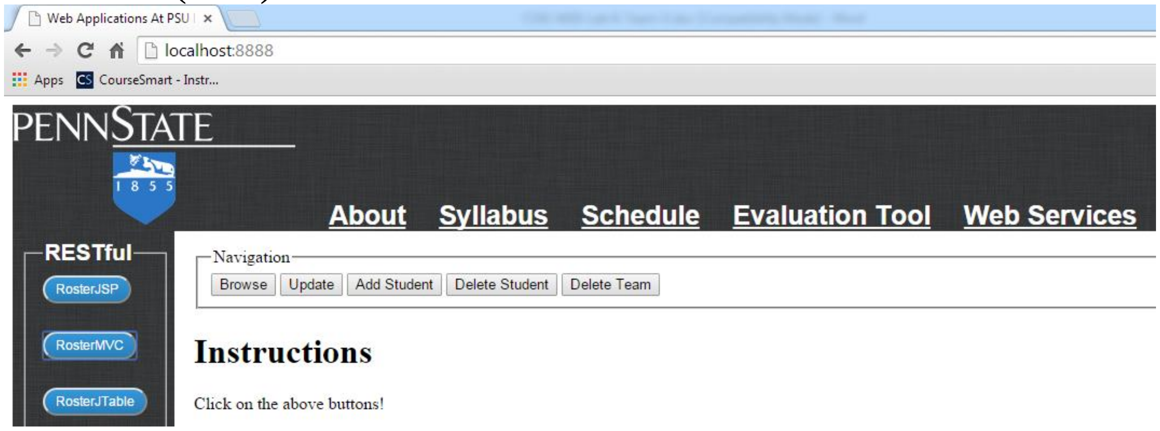
1. [**5 points**] [Refactoring] From the last lab, you should have already created a folder “Services”. Now, refactor the server so that the resources related to RosterJSP are within a folder named “RosterJSP”.
2. [**5 points**]Under “Services” folder, create another folder named “RosterMVC”, which is the place where you put all the resources/code generated from this Lab.
3. [**30 points**] In NetBean create a web application named “**WebRosterMVC**” (it should be within the folder **RosterMVC**)
    a. You should have a folder “Controller” that holds RosterCRUDController.java, which inherits HttpServlet.
    b. You should have a folder “Models” to hold application data and logic (Java Beans). Your beans should be able to read information from roster.txt and manage roster information.
    c. You should have a folder “Views” to hold all the JSP pages.
    d. Check the project “BankMVC” (learned in lecture 21) to see the locations of the MVC folders.
    e. Use GlassFish to host roster.jsp (GlassFish uses port **8080**)

4. [**15 points**]All your JSP pages should contain links for the clients to take full advantage of the CRUD services offered by RosterCRUDController.java. You may use JSP "include directive" or JSP "include action" to include those links common to all the pages. The figure below shows one display where the available actions are grouped together and placed at the top of a page. However, you are encouraged to be more creative for presenting those links (actions).



| LAST_NAME | FIRST_NAME | PSU_ID | Team |
|---|---|---|---|
| BAKER | AUSTIN | ARB5671 | 3 |
| BALTA | TRENT | TMB5577 | 12 |
| CAMPBELL | MATTHEW | MRC5372 | 10 |
| DUONG | CHARLIE | CTD5100 | 8 |
| GOGA | DANIEL | DRG5190 | 11 |

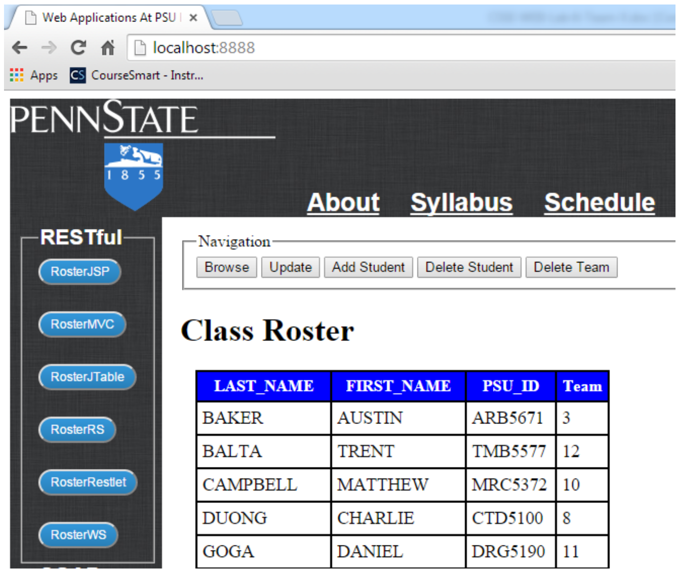
5. [**10 points**] A click on the "Web Services" link should lead to the following display, where the button "RosterJSP" should still function well (lab 9) after this lab.



6. [**5 points**] Client-side javascript: when a user clicks the "RosterMVC" button in the side region, the request is first sent to the Node server. The node server should forward the request to the RosterCRUDController servlet hosted in GlassFish: use **res.redirect**("http://localhost:**8080**/WebRosterMVC/") [Note that if

your web browser and web servers are hosted on different machines, "localhost" should be replaced by the IP address or host name of the machine where WebRoster is running.].

7. [**5 points**]The default response should be displayed similar to the figure below:



Class Roster

| LAST_NAME | FIRST_NAME | PSU_ID | Team |
|---|---|---|---|
| BAKER | AUSTIN | ARB5671 | 3 |
| BALTA | TRENT | TMB5577 | 12 |
| CAMPBELL | MATTHEW | MRC5372 | 10 |
| DUONG | CHARLIE | CTD5100 | 8 |
| GOGA | DANIEL | DRG5190 | 11 |

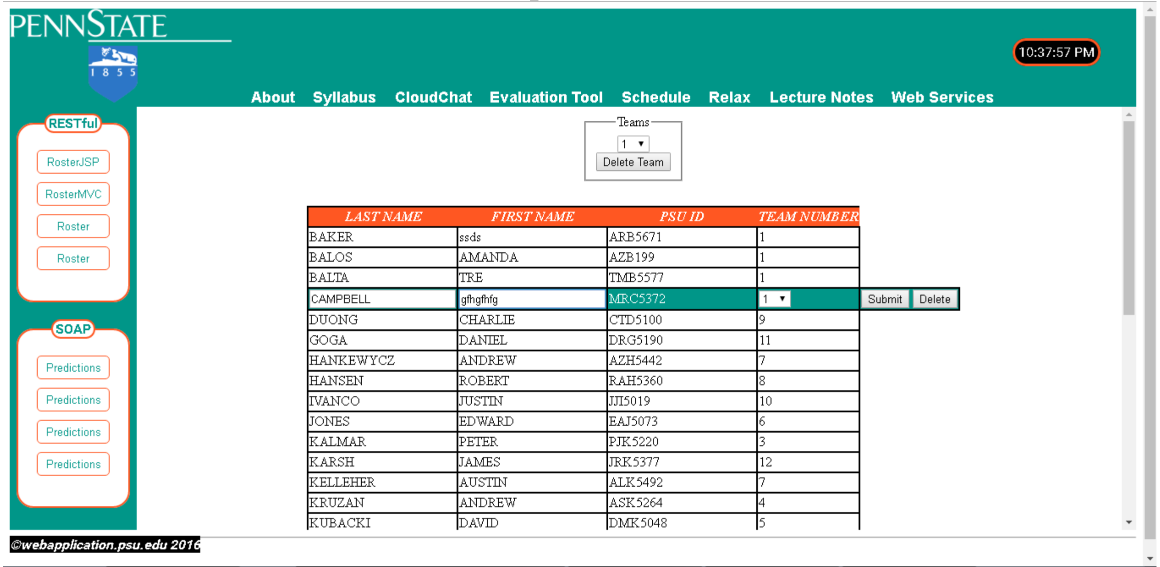
8. [**5 points**]The table should be re-ordered whenever a user clicks one of the table headers (maybe toggle between increasing and decreasing orders). You can reuse the techniques implemented in Lab9.
9. [**20 points**]Each client action (say, a button press) should bring the client to an appropriate page to perform the corresponding action. Also, each page should have links for the client to take any other actions.

## Lab Submission:

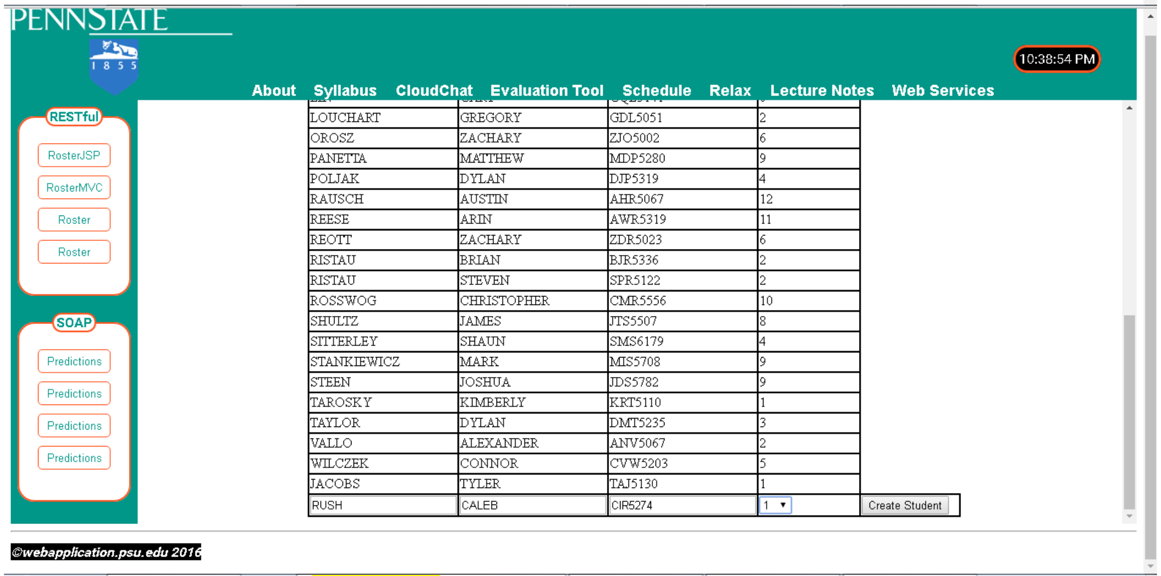
1. **Provide individual performance in the table below (performance factor is a real number between 0.0 and 1.0, individual grade is lab grade times his/her performance factor). [Check the "ACM code of ethics, if you do not know it. If you do not honestly record teammates' performance factor, you are actually encouraging them to be lazy. In doing this, you are NOT helping them, but preparing them to fail in their career.]**

| Student Name | Performance factor |
|---|---|
| Caleb Rush | 0.33 |
| Matt Downey | 0.33 |
| Nick Totolos | 0.33 |

2. Paste a screenshot inside this lab report to demonstrate how it works.



| LAST NAME | FIRST NAME | PSU ID | TEAM NUMBER |
|---|---|---|---|
| BAKER | ssds | ARB5671 | 1 |
| BALOS | AMANDA | AZB199 | 1 |
| BALTA | TRE | TMB5577 | 1 |
| CAMPBELL | gfhgfhfg | MRC5372 | 1 |
| DUONG | CHARLIE | CTD5100 | 9 |
| GOGA | DANIEL | DRG5190 | 11 |
| HANKEWYCZ | ANDREW | AZH5442 | 7 |
| HANSEN | ROBERT | RAH5360 | 8 |
| IVANCO | JUSTIN | JJI5019 | 10 |
| JONES | EDWARD | EAJ5073 | 6 |
| KALMAR | PETER | PJK5220 | 3 |
| KARSH | JAMES | JRK5377 | 12 |
| KELLEHER | AUSTIN | ALK5492 | 7 |
| KRUZAN | ANDREW | ASK5264 | 4 |
| KUBACKI | DAVID | DMK5048 | 5 |



| | | | |
|---|---|---|---|
| LOUCHART | GREGORY | GDL5051 | 2 |
| OROSZ | ZACHARY | ZJO5002 | 6 |
| PANETTA | MATTHEW | MDP5280 | 9 |
| POLJAK | DYLAN | DJP5319 | 4 |
| RAUSCH | AUSTIN | AHR5067 | 12 |
| REESE | ARIN | AWR5319 | 11 |
| REOTT | ZACHARY | ZDR5023 | 6 |
| RISTAU | BRIAN | BJR5336 | 2 |
| RISTAU | STEVEN | SPR5122 | 2 |
| ROSSWOG | CHRISTOPHER | CMR5556 | 10 |
| SHULTZ | JAMES | JTS5507 | 8 |
| SITTERLEY | SHAUN | SMS6179 | 4 |
| STANKIEWICZ | MARK | MIS5708 | 9 |
| STEEN | JOSHUA | JDS5782 | 9 |
| TAROSKY | KIMBERLY | KRT5110 | 1 |
| TAYLOR | DYLAN | DMT5235 | 3 |
| VALLO | ALEXANDER | ANV5067 | 2 |
| WILCZEK | CONNOR | CVW5203 | 5 |
| JACOBS | TYLER | TAJ5130 | 1 |
| RUSH | CALEB | CIR5274 | 1 |

3. Save this report to a PDF file with the name **CSSE-WEB-Lab-10-Team-X.pdf**, **where X is your team number**;
4. Submit to StepStone. Your submission should be one zip file with the name **CSSE-WEB-Lab-10-Team-X.zip** (where **X** is your team number) that includes:
   a. **CSSE-WEB-Lab-10-Team-X.pdf**;
   b. The Website folder, excluding the sub-folder node_modules generated by npm;