

# Exam 3b – Database Web App (JSP)

---

MSCI:3300 – Software Design & Development

Due **Monday, December 12, 2016 at 8:00am**

This take home, programming exam is worth **60 points**.

You are tasked with developing a database driven web application using Java & JSP utilizing the MVC design pattern for keeping customer contact details. The database diagram can be found on the last page. From your web application you should be able to read the database; add, delete, and update records in the database; and search for records in your database.

1. Draw a wireframe diagram for the application as well as an HTML5 & CSS3 prototype.
2. Design and draw a storyboard for your application.
3. Make screen sketches for each page of your application.
4. Create a class diagram showing a constructor, a default constructor, setters and getters, and `.ToString()` for your application.
5. Draw data flow for Read, Add, Delete, Update and Search functionality.
6. Please use Github for source code management. Place your web application on your web server, the same way you posted homework.
7. This is an individual effort. You may use any resource at your disposal, except the “cut & paste” work of another person. Collaboration is fine, but make sure you are creating your own work from the collaboration.
8. All development must stop on **Monday, December 12, 2016 at 8:00am**.

**Late submissions will not be accepted.**

## System Requirements:

1. The search text box should appear in a horizontal DIV on every page.
2. A single search box should search both the first and last name fields, be case insensitive and find the pattern in the word.  
(ie. a search for "ryan" should return  
"Paul **Ryan**" and "**RYAN** GOSLING" and "david **ryan**air" and "JORYAN JOREA")
3. The application should have a menu, driven from an "includes" file.
4. The application should make use of external CSS and "includes" files to make your application modular and easily maintainable. At minimum you should have a header, footer, and menu.
5. Your input box "types" should match the input data.
6. Your input box types should contain required field validators where appropriate.
7. Required fields should be denoted in your application and provide user-friendly error messages.
8. Your application should use "Post" instead of "Get" for form submission.
  - a. Get passes information in the request header field (think URL). It is size limited.
  - b. Post passes information in the request body field. It is NOT size limited and the information would be encrypted if we were using SSL.
9. Your UI should be pleasing to the eye (text and colors), easy to use and learn, and incorporate the norms and best practices we discussed in class.
10. When your page is viewed as a normal user, they should be able to list all records and search. They should not be able to update, add, or delete.
11. When your page is viewed as an administrator, they should be able to search, list all records, add, update and delete records as well as logout.

## Database Diagram:

