MIST 4630 – Fall 2014
Group Project

Scenario: A Document Management System (DMS) allows a set of users to store and manipulate various types of documents. For this project, you will create a DMS that is focused on resumes. This résumé system will have three types of users - an administrator, members who can post a résumé, and the employers who can browse or search for résumés.

Features: (This is a minimum list of features. You can add more, but be sure to manage the scope in order to complete by end of semester)

Entry page: This view should provide a description of the system and the ability for employers to start browsing or searching for résumés. The page should also include a link that allows members and administrators to login.

Administrator Features: The system should be able to detect when the user with administrator access logs in. An administrator should be able to:

- add, modify and delete members from the system.
- view and delete posted résumés

Member Features: A member should be able to:

- fill out or edit a profile page
- upload a résumé document (pdf format)
- remove their résumé

Employer (Public) Features: An employer or otherwise public viewer should be able to:

- Browse through all posted profiles and résumés
- Filter the list of résumés based on graduation date or keywords.

Your project must include the following components:

- **Database:** You will need to design and create a database to store relevant data. Make sure that your database is robust and follows the database design lessons you have learned during your MIST courses.
- Complete set of MVC components for all requests/responses.
- Relevant State Management objects.

General Requirements: Your application should also conform to the following general requirements:

- Validate data on both the client and server side. Make sure that your application "degrades gracefully." You may choose to add additional error pages to the list of application elements above.
- Use standard Java conventions.
- Use Java classes in the model as much as possible.
- All page views should have a consistent and pleasing look and feel - **using CSS to style the pages**.
- The application should be thoroughly tested and debugged.
- The work submitted should be entirely from the efforts of team members with no outside help. Keep in mind that professors who are devious enough to develop projects like this one may have other more devious methods for detecting this type of activity.

Teams: Your team should fit the following class description:

- *numTeamMembers* – 1 to 3 per team.
- *getAlong*() – your team should implement this method by default. If an error occurs it should throw a *notifyProfessor* exception ASAP.
- *numMembersCoding* – this value should equal *numTeamMembers*. While this would be difficult to check during the professor's test/deployment phase, it is in each team members interest to be familiar with all aspects of this projects so that they can avoid the dreaded *failFinal exception*.

Deliverables: Your deliverables and due dates follow:

- **Deliverable 1 – Team list –** send by email a list of team members on your team via email to craig.piercy@gmail.com by *10:00pm Monday, Oct. 20*.
- **Deliverable 2 – Design documents –** deliver via ELC by *10:00pm Monday, November 3* design diagrams for your project. These should include at least a database data model, a UML class diagram, a site map diagram showing the pages that a user will see as they move through the site, and a non-functional mock-up of Web pages and navigation.
- **Deliverable 3 – Application files –** deliver via ELC by *10:00pm* Monday, *December 8* a zipped folder containing all components (Netbeans project folder, database file, image files, etc.) of your project. Warning: as this is the last official day of this class for the semester, there will be no extensions to this due date.

Good luck and have fun!