Group Term Project

Background Narrative

Oconee Fire Rescue is a mostly volunteer organization in Oconee County, GA. Its fire fighters and rescue personnel have to maintain multiple certifications, many of which expire. The department would like an online system that tracks its members, their contact information, and the certification status of each of its current and former members.

Required Use Cases

Login. An administrative user can log in using an email and password. A logged in user can view and update all data, and view reports. A user not logged in sees only a login form.

Data Entry. A logged in user can see a list of members, including basic information such as radio number and station.

Delete and Update. A logged in user can add, edit, or delete member records, certifications, and other entities required for the application to function. At a minimum, when an member or certification entity record is displayed, provide links to delete or update that record.

Reports. Several reports are available to logged in users, including:
- A report that lists people with expired certifications (which may be filtered by certification),
- A report that lists members by station and radio number, and shows email addresses. This report shall be able to be exported to a CSV file.

A logged in user shall have the option of showing all members in a report or filtering by station.

Use custom HTTP 400 and HTTP 500 error pages. In the event of a 500 error, the server should email a system administrator with necessary debug information (minimally, error information and stack trace), via an email address set in a separate configuration file.

Required Components:

Your project must include the following required components:
- **Database:** Use a database to store relevant data. (Either MySQL or SQLite.) Make sure that your database is robust and follows the database design lessons you have learned during your MIST courses. Data attributes that must be included in the database are listed on the last page of this document. Be sure to apply good database design techniques that you learned in your Data Management course to create an appropriate data model for this data.
- **HTML & CSS:** Style and design your pages professionally.
- Proper demonstration and use of MVC components and separation of concerns.
- **Data validation.** Validate your data inputs on both the client and server. (For example, first name and last name might be required fields.) Make sure that your application “degrades gracefully.”
- Relevant state management. (Maintain a logged in session appropriately.)

General Requirements

- Use standard Java conventions.
- Use Java classes in the model as much as possible. Separate your concerns. (Do not put "view" methods in your "models", for example.)
- All page views should have a consistent and pleasing look and feel. Use a common CSS file to style the pages. (Bootstrap is fine.)
- 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. Three is a hard ceiling.
- `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 member's 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 Tuesday, March 22.
- **Deliverable 2 — Design documents.** Submit via ELC by 10:00pm Tuesday, April 5 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 — Design discussion.** Schedule a 30-minute meeting with your professor (during office hours is ideal) that occurs within approximately one week following submitting Deliverable #2 (by Wednesday, April 13)
- **Deliverable 4 – Application files.** Submit via ELC by 10:00pm Thursday, April 28 a zipped folder containing all components (Eclipse 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!

Entities

Build a web application with database that includes *at least* the following entities. You may add other entities and relations as you see fit, but the following are minimally required. As described above, your web application shall permit an authenticated user to view, create, edit, and delete the following entities. The included class diagrams are a guide. You will need to flesh them out further as you develop your project.

Users

Provide separate log-ins for each user. Encrypt or use a one-way hash for passwords. All users have the same administrative privileges. Users should be able to change their passwords and email addresses. Although not required, you may choose to include a password recovery process, whereby a temporary password is sent to a user's email address.

User
`email : String` `password : String`
`static authenticate(email, password) : User`

People

Each member of the department should have his/her personal information stored, including address, email, phone number(s), date of birth, gender, and position within the department. Each member may have a radio number, and may be active or inactive in the department.

Person

`firstName : String`
`lastName : String`
`radioNumber : String`
`stationNumber: integer`
`isActive : bool`
`…`
`// and other attributes as described/needed`

`bool isActive()`

Certifications

Each certification has a certifying agency, a certification name, and a default expiration period (in years).

Certificaton
`?` `// attributes as described/needed`
`?`

PersonCertifications

People and Certifications have a many–many relationship (A person may have many certifications. A certification may be earned by many people.) A person's certification may (or may not) have an expiration date, and an earned/renewed date.

PersonCertification
? // attributes as described/needed
?

Sample Data

Sample people:

Kathryn Pryde (Chief)
Female
1123 Xavier School Drive
Watkinsville, GA 30677
Work Phone: 707-555-1234
Mobile Phone: 707-555-2345
Radio#: A-1
Station: (all)
Certifications: Firefighter II (renewed Aug/2015, exp Aug/2017); CPR (renewed Jul 2014, exp Jul/2016), HAZMAT (exp Feb/2017), Extrication (no expiry)

Piotr Rasputin
Male
A31 Mother Russia Road
Seattle, WA 98133
Mobile Phone: 206-555-9876
Radio#: 841
Station#: 8
Certifications: EMT-Adv (exp Sep/2017); CPR (renewed Jul/2014, exp Jul/2016), Due Regard (renewed Oct/2015, exp Oct/2017)

Warren Worthington III
Male
1140 Experiment Station Rd
Watkinsville, GA
Work Phone: (706) 555-3945
Radio#: 122
Station#: 1
Certifications: Paramedic (exp Sep/2017); CPR (exp Jul/2016), Due Regard (exp Oct/2017), Firefighter I (exp Aug 2016)

Sample Certifications:
CPR (CPR for Healthcare Providers/American Heart Association), standard expiry: 2 years
CPR (CPR for the Professional Rescuer/American Red Cross), standard expiry: 2 years
Firefighter I (Athens Technical College), standard expiry: 3 years
Firefighter I (Gwinnett Technical College), standard expiry: 3 years
POST (Georgia POST Academy), standard expiry 5 years