CMPT 561 Project – Fall 215

Phase 1: Design a Rubric-based Evaluation System (RES)

Please post questions to Piazza about any ambiguities in this document then I will add further clarifications and post an updated document. **Note that the project is worth 30% of the overall grade** (10% for each phase). **Push your work to Github as you make progress.**

Project Phase 1 due date is by 5pm **Thursday 29^{h} October**.

1. Requirements

Design and build a system to share Evaluation Rubrics and allow using them to evaluate a list of items (e.g., evaluate a project implementation submitted by students). This system should focus on the use cases shown in Figure 1 and explained in Table 1.

Figure1. RES use cases

Note some use cases are omitted or simplified to keep the project scope reasonable. Also, note that the system should already have the …. Hence, no need to provide the ability to maintain these entities unless explicitly stated in the requirements.

Table 1. RES use cases description

Use Case	**Description**
Login	All users should first Login before using RES. To login, the user should enter their username and their password. The system should validate the entered credentials. If the login fails then an error message should be displayed and the user should be prompted to login again. If the login is successful, the system should display the menu. The users should only be granted access to the use cases shown in the use case diagram (e.g., …).
Add a Rubric	.

Update a Rubric	.
Update the list of items to evaluate and select the evaluation rubrics	.
Perform evaluation	
View evaluation results	.

2. Deliverables

- Identify and design the forms required for your Web interface and the flow between the forms. **Discuss your design with the instructor before the implementation.**
- Design and develop your model and document it using a class diagram.
- Design the Entity Repository methods and Design ERS controllers to meet the project requirements.
- **Design ERS Web UI:** the required HTML forms, navigation,
- Write a design and testing document to include the class diagram and screen shots of conducted tests.
- Demo your implementation and answer questions about the implementation. 20 minutes demo will be allocated to each team.

3. Grading

Your project will be graded based on the **completeness** and the **quality of the implementation**. In order to receive full credit in each area, it must be **1) complete, 2) done well, and 3) tested**. Below is the breakdown of the grading criteria and it will be further refined.

Grading Rubrics

Criteria	**%**	**Functionality**[*]	**Quality of the implementation**
Complete, correct, accurate and good quality implementation of the model	7		
Complete, correct and accurate implementation of Entity Repository methods	7		
Complete, correct and working implementation ERS use cases - Login and Home page with Menu	6		
356	10		
	8		
	8		
	8		
	8		
	16		
	12		
Design documentation Class Diagram showing **Entities**, **Repositories** and **Controllers**	5		
Testing documentation with evidence of correct execution using snapshots illustrating the results of testing to show that your implementation works and meets the requirements.	5		
Total	100		
Program does NOT compile	-50%		
No demo of the implementation	-50%		
Not submitting the design and testing documentation	-30%		
Not using the design and testing template	-10%		
Copying and/or plagiarism or not being able to explain or answer questions about the implementation	-100%		

[*] **Possible grading for functionality**: ***Working*** (get 70% of the assigned grade), ***Not working*** (lose 40% of assigned grade and ***Not done*** (get 0). The remaining grade is assigned to the quality of the implementation. In case your implementation is not working then 40% of the grade will be lost and the remaining 60% will be determined based on of the code quality and how close your solution to the working implementation. Code quality includes **correct usage of MVC**, applying OOP best practices particularly encapsulation, inheritance and polymorphism when relevant, meaningful naming of identifiers, no redundant code, simple and efficient implementation, clean code without unnecessary files/code, use of comments where necessary, proper white space and indentation.

Marks will be reduced for code duplication, poor/inefficient coding practices, poor naming of identifiers and unnecessary complex/poor user interface design.

4. Submission Guidelines

- Your design and testing Word Document **must follow the provided template**. It must be placed in a ***docs*** folder within your NeatBeans Project.
- Your implementation should be pushed to GitHub as you progress. Every team member should actively contribute to the project. I will assess each team member contribution based on the files they push to github. Potential free riders will be easy to spot.
- **You must submit a hardcopy of your design and testing document during the demo session. I will grade using the hardcopy.**

Important Notes:

- It is very critical to pay attention to the proper application of MVC.
- All assignments **must be your own original work.**
- Each team must schedule at least one office hour meeting with the instructor to review and discuss your implementation and get feedback before your submit your work.
- For any email you send me w.r.t. the project please CC all the team members also add CMPS356 to the email title.
- **No free ride is allowed!** All students must contribute to the best ability to the success of the projects. Team work skills are critical. Please help each other, learn from each other and keep a good team spirit. If the team complains about a student's poor contribution then he/she will be asked to submit his/her own solution individually.
- **All team members need to participate and be present during the demo of your solution.**
- **Office hours are your right**. Please use them and come and see me if you need any further clarifications and guidance (not solutions!).
- **Late submissions** will result in **severe point penalties**. If you submit one day late (or less), 10 points will be deducted from your grade, 2 days will deduct 30 points, and any submissions after that will receive an automatic zero.