|  |  |
| --- | --- |
| D:\Users\ae\AppData\Local\Temp\SNAGHTMLd44051a.PNG | **Web App** |

***CMPS356 Project Phase 1 – WebApp Design***

|  |  |
| --- | --- |
| **Group Id:** | G? |
| **Group Members:** | Student1 full name (StudentId)  Student2 full name (StudentId)  Student3 full name (StudentId)  **Emails:** student1@student.qu.edu.qa; student2@student.qu.edu.qa; student3@student.qu.edu.qa; |

**Grading Rubric - In the *Functionality* column please specify either: *Working (completed x%)*, *Not Working (completed x%)* or *Not done*.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **%** | **Functionality\*** | **Quality of the implementation** | **Score** |
| **Application Design**  Architecture Design, Classes Design and the UI Design to deliver RIFQ use cases.  The design documentation should include at least the following:  • Application Architecture Diagram  • Class Diagram showing Entities, Repositories and Services and Controllers.  • UI Design and navigation.  • Discussion of design rationale of key design decisions. | 20 |  |  |  |
| **Complete and correct implementation of the requirements:** |  |  |  |  |
| **Image result for add task icon Add Course Task** | 22 |  |  |  |
| **https://cdn1.iconfinder.com/data/icons/MetroStation-PNG/128/MB__tasks.png Update Course Task** | 16 |  |  |  |
| **Image result for delete task icon Delete Course Task** | 2 |  |  |  |
| **Related image Get Course Tasks** | 12 |  |  |  |
| **Image result for tasks calendar icon Student Tasks Calendar** | 10 |  |  |  |
| **Image result for google calendar logo vector Sync Tasks to Google Calendar** | 0 |  |  |  |
| **Related image Add Course Comment** | 6 |  |  |  |
| **Related imageGet Course Comments** | 6 |  |  |  |
| **Testing documentation** with evidence of correct implementation using snapshots illustrating the results of testing. | 6 |  |  |  |
| **Total** | 100 |  |  |  |
| Copying and/or plagiarism or not being able to explain or answer questions about the implementation | -100% |  |  |  |

**\* Possible grading for functionality**: ***Complete and*** ***Working*** (get 70% of the assigned grade), ***Complete and*** ***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. Quality **correct application of MVC**, includes meaningful naming of identifiers, no redundant code, simple and efficient design, clean code without unnecessary files/code, use of comments where necessary, proper white space and indentation. **Marks will be reduced** forcode duplication, poor/inefficient coding practices, poor naming of identifiers and unnecessary complex/poor user interface design.

# Design Documentation

# Application Architecture Diagram

# Class Diagrams: Entities, Repositories and Services and Controllers

# UI Design and navigation

(You may use the screenshots of implemented UI).

# Discussion of design rationale of key design decisions

# User Interface Testing

# Add Course Task

# Update Course Task

# Delete Course Task

# Get Course Tasks

# Student Tasks Calendar

# Add Course Comment

# Get Course Comments