**OlivApp - Olive Oil Mills Management App**





***CMPS356 Project Phase 1 – Web UI Design and Web API Implementation (15% of the course grade).***

|  |  |
| --- | --- |
| **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** |
| 1. **Application Design:**  * 3 Class Diagrams for Entities, Repositories and Services [5 pts each]. * Discussion of **design rationale** of 5 key design decisions[5 pts] | 20 |  |  |  |
| 2) Create test data JSON file for each of the entities. | 5 |  |  |  |
| **3) [30%] Design and implement the Web UI using HTML and CSS for:** |  |  |  |  |
| Main menu design | 4 |  |  |  |
| Login | 4 |  |  |  |
| Register Customer | 5 |  |  |  |
| Get Jobs | 20 |  |  |  |
| Add Job | 10 |  |  |  |
| Update Job |  |  |  |
| Cancel Job | 1 |  |  |  |
| Add Invoice | 8 |  |  |  |
| Update Invoice |  |  |  |
| View Invoice | 6 |  |  |  |
| Pay Invoice | 5 |  |  |  |
| Mill Calendar | 4 |  |  |  |
| Add Job Note | 5 |  |  |  |
| Get Job Notes | 5 |  |  |  |
| Get Products (i.e., Products Catalog use case) | 8 |  |  |  |
| Add / Update Product | 7 |  |  |  |
| Jobs Summary Report | 8 |  |  |  |
| **[40%] Implement the repositories and the services (i.e., Web API) using JavaScript for:** |  |  |  |  |
| Login | 5 |  |  |  |
| Add Customer | 6 |  |  |  |
| Get Jobs | 10 |  |  |  |
| Add Job | 6 |  |  |  |
| Update Job | 5 |  |  |  |
| Cancel Job | 5 |  |  |  |
| Add Invoice | 6 |  |  |  |
| Update Invoice | 5 |  |  |  |
| Get Invoice | 10 |  |  |  |
| Pay Invoice | 5 |  |  |  |
| Get Jobs for Mill Calendar | 8 |  |  |  |
| Add Job Note | 5 |  |  |  |
| Get Job Notes | 5 |  |  |  |
| Get Products (i.e., Products Catalog use case) | 8 |  |  |  |
| Add Product | 6 |  |  |  |
| Update Product | 5 |  |  |  |
| **5) Testing documentation** using screen shots illustrating the testing of Web UI and Web API. | 5 |  |  |  |
| - Discussion of the project contribution of each team member [-10pts if not done] |  |  |  |  |
| **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

# Class Diagrams: Entities, Repositories and Services

# Discussion of design rationale of 5 key design decisions

# User Interface Testing

# Login

# Register Customer

# Get Jobs

# Add Job

# Update Job

# Cancel Job

# Add Invoice

# Update Invoice

# View Invoice

# Pay Invoice

# Mill Calendar

# Add Job Note

# Get Job Notes

# Get Products (i.e., Products Catalog use case)

# Add / Update Product

# Jobs Summary Report

# Web API Testing

# Login

# Add Customer

# Get Jobs

# Add Job

# Update Job

# Cancel Job

# Add Invoice

# Update Invoice

# Get Invoice

# Pay Invoice

# Get Jobs for Mill Calendar

# Add Job Note

# Get Job Notes

# Get Products (i.e., Products Catalog use case)

# Add Product

# Update Product