**CMPT606 – Advanced Database (Fall 2019) – Project Phase 1**

**<ToDo: Add a project title >**

|  |  |
| --- | --- |
| **Group Id:** | G? |
| **Group Members:** | Student1 full name (StudentId)  Student2 full name (StudentId)  Student3 full name (StudentId) |

**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. **Requirements specification**   Clear definition of the project requirements documented as use cases. | 10 |  |  |  |
| 1. **Relational Database Model**   - Sound and complete database model to meet the requirements. Meaningful naming of entities and attributes.  - Justified usage of appropriate indexes. | 30 |  |  |  |
| 1. **Use cases implementation**  * Class Diagrams for Entities, Repositories and Services. * Complete and correct implementation to deliver use cases. | 45 |  |  |  |
| 1. **Testing documentation**   Evidence illustrating the testing of a working solution. | 10 |  |  |  |
| 1. **Writing quality and organization of the project report**   - Good report organization with logical order and transitions.  - All figures are accurate, clear, labeled and of good quality. | 5 |  |  |  |
| **Total** | 100 |  |  |  |

**\*** Quality includes efficient design, clean code without unnecessary files/code, meaningful naming of identifiers, no redundant code, proper white space and indentation. **Marks will be reduced** forcode duplication, poor/inefficient coding practices, poor naming of identifiers and unnecessary complex/poor design.

# Requirements specification

<ToDo: present an overview od the project. Add the use cases diagram and use cases summary table. />

# Relational Database Model

# ER Diagram

# Indexes

<ToDo: Include a summary table of indexes created to improve query performance and discuss the rational for your choices. />

# Use cases implementation

# Entities Diagram

# Repositories and Services diagram

# Testing

<ToDo: Add a subsection for each use case and include evidence of a working solution (e.g., screenshot of Web API output, query results, etc. />

# …

# …