Concordia University

**Department of Computer Science**

**and Software Engineering**

**Software Process**

**SOEN 341/4 S --- Winter 2016**

**Project Testing and Delivery Document**

|  |  |
| --- | --- |
| **Team information** | |
| **Team :** | |
| **Name** | **SID** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[List here all contributions/revisions to the document. List the name(s) of all contributors to all contributions.] **Grading Sheet**

|  |  |  |
| --- | --- | --- |
| Section | Evaluation criteria (see instructions in the template for details) | Grading |
| all | 10 marks are allocated for excellence, professionalism and quality of work above and beyond the correct meeting of specifications. | /10 |
| 1 | Presentation of the document | /5 |
| 2 | Introduction | /1 |
| 3.1 .  3.2  3.2.1  3.2.2  3.2.3  3.2.4 | Completeness of covered/uncovered items. Rationale of the importance of testing these items.  *Testing*  *Reproducibility of test cases. Exact description of test input data and expected results, and the procedure to convey all test cases. Description of the rationale for the derivation of each test case, e.g. equivalence partitioning analysis, branch coverage analysis, etc.*  *Unit testing*  Requirements testing:  Stress testing  *Security testing* | /3 .  /4  /8  /1  /1 |
| 4.1 .  4.2 | Clarity of instructions. Self-inclusion of the installation procedure, i.e. the installation does not necessitate the installation of external resources.  Clarity of instructions. Completeness of instructions, i.e. all system features’ usage instructions are provided. | /7 .  /7 |
| 5 | Completeness and clarity of cost to date in terms of person hours. | /3 |
| Total |  | /50 |

DO NOT REMOVE THIS PAGE WHEN SUBMITTING YOUR DOCUMENT

# Presentation

*Professionalism of the document in terms of look and feel including, but not limited to layout colour and binding.*

# Introduction

The instructions provided in blue are there to provide you indications describing the expected content of the respective sections. They are all to be deleted and replaced with appropriate content.

The introduction of the document provides an overview of the entire document, briefly introducing what are its goals, and what information is to be found in it.

# Testing Report

This section presents all the testing activities undertaken on the final product, as well as all the individual test cases used.

## Test Coverage

### Tested Items

List all tested items, along with the test cases that were applied on this item. For each test item, explain why it was necessary to test it. For instance, all features listed as requirements for each build is a mandatory test item. In addition, identify at least five units (i.e. classes/methods) and explain why they require unit testing due to their importance in the implementation through their frequency of use and/or the severity of the impact of their misbehavior. You can categorize your tested items, e.g. “Requirements”, “Units”, etc.

### Untested Items of Interest

List all untested items that you find would necessitate testing. Explain how it could be tested, and why it would be important to test.

## Test Cases

Description of all the test cases applied on the tested items using various techniques and testing different aspects of the system. The following sections are mandatory testing perspectives. Other sections can be added to provide appropriate additional testing perspectives. All test cases must be presented as to be reproducible, with the exact data and procedure to convey the test, as well as the expected result.

### Unit Testing

In your system, identify 2 substantive, mid-level testable units (classes, modules or subsystems)

For each of these two units, include a list of test cases and the code for the stubs and drivers used. Show the results of the testing. Where possible, a test suite should be used.. Explain what techniques were used to derive these tests.

### Requirements Testing

### For each tested requirement, include a list of test cases presented in the form of a concrete scenario of system usage and expected system reaction.

### Stress Testing

Describe potential extreme situations of system usage. Describe the design of tests that would verify system performance under these extreme conditions. Perform the tests and show the results.

### Security Testing

*Your project should be tested for security, Including resistance to SQL injection attacks, against at least two automated tools (e.g Nikko and Skipfish).*

# System Delivery

This section provides instructions as to how to install and use the software.

## Installation Manual

Exact description explaining how to install the system, from as self-contained compressed file or CD, to the actual execution of the software. The manual should be written such that any competent administrator should be able to install your system using your code and this manual.

## Users Manual

Exact description of how to use the system. All system features should be presented.

# Final cost estimate

This shall consist of a table listing all components of all phases of this project, including the person hours cost of each component of each phase. This should include documentation, design, implementation, testing, and acquisition of data.