CS 5800 - Course Project – Version 1
Maximum Points: 100 pts (25% of the course grade*).

[1 point] Cover page (see example below)

Project Title
Team Members

Cover Page Example

Development and Management of the Project: [Name of the Project]

Team Members

	Student 1	Student 2	Student 3	Student 4	Student 5
Name	[Name of Student]	[Name of Student]	[Name of Student]	[Name of Student]	[Name of Student]

[2 point] List of Content (new page)
[1 point] List of Figures (new page)
[1 point] List of Tables (new page)

List of Content Example

List OF FIGURES 1

LIST OF TABLES 2

ABSTRACT 3

PROJECT PLAN 3

1 INTRODUCTION 6

1.1 SYSTEM-AS-IS 6

1.2 SYSTEM-TO-BE.. 7

2 PROCESS MODEL 9

3 EFFORT AND SCHEDULE.. 6

3.1 EFFORT.. 6

3.2 SCHEDULE 6

3.3 ALLOCATION OF RESOURCES

APPENDIX 32

* The other 5% of the course grade will be assigned to the project presentation.

[15 points] Project Plan (new page)

Introduction

System-As-Is

System-To-Be

Process Model

Development model and its activities to be undertaken (*an iterative model should be chosen*)

Organization of the Project

Specification of the roles (project manager, tester, programmer, analyst, etc.) and their responsibilities

Methods and techniques

Description of the methods and techniques to be used during requirements engineering, design, implementation, and testing, and how this documentation will be taken care of

Effort and Schedule

The size (measure in terms of number of unadjusted function points – IFPUG's method) estimate for the project. The schedule of the tasks using a PERT chart (*GanttProject* must be used - https://www.ganttproject.biz/). The tasks to generate this report and its milestones must be presented.

Allocation of resources to the tasks (*GanttProject* must be used)

Delivery

The procedures to be followed in handing over the system to the customer must be stated

[25 points] Requirements Specification (new page)

High-Level Goals (Unique Identifiers)

Enumerate, describe, and name each high-level goal of the system-to-be with a unique identifier

Primary and Secondary Actors

List all actors of the system and their corresponding roles

Use Case diagram (s)

Elaborate the use case diagram (s) of the proposed system-to-be. Recommended tool: *Astah* - https://astah.net/.

Use Case Specification (s)

Specify each base use case included in the use case diagram (s) before. Include at least three Alternative Courses and three Exception Courses.

Paper-based Prototype

Visual description of the system including menus (navigability), screens, functions, reports, messages. Recommended tool: Balsamic Wireframes (drawing flavor) - https://balsamiq.com/wireframes/.

Assumptions

List all assumptions and their corresponding rationale. Assumptions must be grouped by categories if more than one is specified.

Domain Properties

List all domain properties and their corresponding rationale.

Functional Requirements

List all functional requirements (FR) and their corresponding rationale. FRs should be grouped by categories and must have a unique identifier for traceability.

Non-Functional Requirements

List all non-functional requirements (NFR) and their corresponding rationale. NFRs should be grouped by categories and must have a unique identifier for traceability.

Traceability Matrix

Include requirements (IDs), design (classes), implementation (screens), and designed test cases.

[25 points] Design (new page)

Entity Relationship Diagram (data conceptual model)

Recommended tool: *Draw.io* – https://www.diagrams.net/.

Data Logical Model

Recommended tool: *SQL Power Architect* - http://www.bestofbi.com/page/architect.

System Architecture View and Style/Pattern

Draw the implementation viewpoint of your system detailing the architectural pattern used.

Class Diagram

Recommended tool: *Astah* - https://astah.net/.

[15 points] Implementation (new page)

GitHub repository with the software project including:

Data Physical Model

PostgreSQL must be used - https://www.postgresql.org/.

Graphical User Interface:

Java Swing or *Java FX* must be used.

Source-Code and Configuration Files

Specify all layers, packages, classes, libraries, configuration/initialization (readme files). *Java* must be used for the implementation - https://www.oracle.com/java/technologies/downloads/.

Persistence:

ORM implementation. *Hibernate* must be used - https://hibernate.org/. However, JDBC is allowed for some cases as well.

Web Development:

JSP - *Java Server Page* and *Java Servlet* must be used (Java EE - Java Enterprise Edition).

Web (Java Servlet) container – *Apache Tomcat* - http://tomcat.apache.org/ must be used.

A minimum 5 and maximum 10-minute video containing a project demonstration

[10 points] Testing Documentation (new page)

Unit Tests

Scenario and Test Cases defined (minimum 6). *JUnit* must be used (summary report) - https://junit.org/junit4/.

Integration Tests

Scenario and Test Cases defined (minimum 3) (paper-based or *JUnit*).

System Tests

Scenario and Test Cases defined (minimum 1) (paper-based).

[4 points] Discussion (new page)

Analysis of the results obtained, and lessons learned.

[1 point] References

Appendix (optional)