Graduation Project Guide

Introduction

This guide is meant to help the project team produce a good final year project report. A good report is one that presents your project work *concisely* and effectively. It should contain various materials relevant to the work you have undertaken in respect of your project; it should be organised into a logical framework; and it should be supported by written material that follows well-established academic conventions in a consistent fashion.

Purpose and Method

The purpose of the project is, in the context of the degree you are studying, to integrate various aspects of the taught material and to demonstrate your (academic) research skills and your (professional) analysis, design and implementation skills. It gives you the opportunity to conduct in-depth work on a substantial problem to show individual creativity and originality, to apply where appropriate knowledge, skills and techniques taught throughout the degree programme to further oral and written communication skills, and to practise investigative, problem-solving, management and other transferable skills.

Teamwork and Project Management

Each team is made up of a number of students working on one project. *Teamwork is* ***required*** *since team work is an integral part of the project.*

All team members are expected to participate in all project activities and none of these activities are to be performed exclusively by a student. Each team member or pair should be responsible for all aspects of development required for the functionality they have.

Report Format

A report is submitted by the whole team and will be graded as a whole, and points will be split among the team members according to the declared contributions

A report must have a *cover page* containing:
. the course title,
. group number,
. project title,
. . submission date, and
. all team-member names.
Negative points will be assigned to reports missing- or having an incomplete cover page.

The **second page** of each report must detail the **breakdown of individual contributions** to the project (use more pages if necessary)—. **Each student** should provide an **itemized list** of his or her contributions to components of the report, such as: requirements specification (use cases and non-functional requirements) etc and other: any other relevant contribution.

Report chapters

Abstract: A summary of the objectives and accomplishments. Typically 1 page long.

Introduction: Describe the background of the project work. Establish the context. Discuss why this problem is important. Briefly describe the development process you will follow.

Literature Review: Provide a survey and a critical review of related prior work.

Analysis and Requirements:
Describe the problem analysis, enhanced with an analysis model in UML. Specify the resulting set of system level and software level requirements.
Describe how you did requirements elicitation, conducted the analysis, and arrived at the specified requirements. Provide analysis models, not just words. Some suggested model elements are: use cases, sequence diagrams, and domain models. The analysis models should express the system architecture and the top level behavioral requirements. Don't provide a superficial model with just one or two context level use case diagrams.

1. System Requirements

a. Enumerated Functional Requirements
Extract the requirements from the customer's narrative and list them in a table, one row per requirement. The first column shows a unique label "REQ-*x*". The second column shows briefly describes the requirement.

b. Enumerated Nonfunctional Requirements
List, prioritize, and describe the Nonfunctional Requirements. The non-functional requirements numbering should continue the functional requirements list.

3. Functional Requirements Specification
Derive the use cases based on the requirements from Section 2 and Section 2 above..

   a. Stakeholders
   Identify anyone and everyone who has interest in this system (users, managers, sponsors, etc.). Stakeholders should be humans or human organizations.

   b. Actors and Goals
   Identify the *roles* of people or devices that will directly interact with the system, their *types* (initiating vs. participating) and the *goals* of the initiating actors.

   c. Use Cases

      i. Use case Description
      For **all** use cases that you can think of (based on your System Requirements), write a *brief* or *casual* text description. List explicitly the requirements that each use case responds to.

      ii. Use Case Diagram
      Draw the use case diagram with all the use cases. Indicate the relationships, such as <<include>> and <<extend>>.

d. System Sequence Diagrams
   Draw the system sequence diagrams for the **few most important** use cases selected above.
e. Class Diagram and Interface Specification

Show all classes and their associations. Only indicate attributes and operations.

If you cannot fit the class diagram on one page, or it looks too cluttered, create one "overview" class diagram showing all classes and their relationships, but for each class show only a single compartment with the class name (leave out attributes and operations).

Chapter on Design

Describe the architectural and detailed design models in a disciplined manner using both text and comprehensive design models, ideally expressed in UML. Use of UML is highly recommended over using ad hoc or older modeling notations. Suggested UML design model elements are: class diagrams, interaction diagrams, structured classes, components, subsystems, and deployment models.

a. System Sequence Diagrams
   Draw the system sequence diagrams for the **few most important** use cases selected above.
b. Class Diagram and Interface Specification

Show all classes and their associations. Only indicate attributes and operations.

If you cannot fit the class diagram on one page, or it looks too cluttered, create one "overview" class diagram showing all classes and their relationships, but for each class show only a single compartment with the class name (leave out attributes and operations).

Chapter on Testing and Validation

Describe how testing and validation tasks were performed. Describe the plans and strategies used in unit testing, integration testing and system testing. Describe the test plans and provide test procedures for testing the critical functions.
Describe the test tools used.

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

Test Cases

Description of all the test cases applied on the tested items using various techniques and testing different aspects of the system.

Unit Testing

For each tested unit,. Explain what techniques were used to derive these tests, e.g. black box/equivalence partitioning, white box/basis path, etc.

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.

Chapter on Tools and technologies: Describe the tools and technologies used in accomplishing the project in the context of the project activities

The first seminar will be evaluated and must include the following:

1. Problem statement
2. Related work
3. Requirements analysis
4. Architecture design (Software and Hardware)
5. Gantt Chart
6. Implantation of at lest 30% of the final work