

Computing Final Year Project – Design Stage : 2017/2018

Submissions and Dates Version 3

The Final Year Project enables students to take responsibility for a complete development project throughout the design and planning aspects of the project. This allows for the integration of knowledge and experience. An incremental approach will be employed in producing an implementation to meet agreed acceptance criteria.

There will be a number of submissions/milestones throughout the semester. At each of these, the student is required to submit a piece of work. The supervisors will decide if progress to date is acceptable.

In relation to module co-ordination:

Helen Fitzgerald, helen.fitzgerald@ittralee.ie, is the Project Co-ordinator.

The role of the Project Co-ordinator is to :

- Assign supervisors to projects.
- Establish submission dates and convey these details to students and their supervisors.
- Collate final submissions.

Submission 1: Project Vision Statement

Thursday 14th September 2017

One paragraph on the vision of your project.

This is used to assign supervisors.

Submission 2: Project Proposal Document

Week of Monday 9 October 2017

Proposed Project Title and Objectives

Research areas defined and research commenced

Initial key citations and references identified

Initial research question Identified.

Defend your research question to the pool on **Thursday 12th October.**

Milestone Project Plan

Week of Monday 23rd October 2017

Main research chapter in Literature Review completed

Table Of Contents of remaining Literature Review.

Risk analysis including platform issues.(Table format)

Vision Document (**Determined in association with SQA Module**)

VISION DOCUMENT TEMPLATE

Entry criteria

Project Vision Statement.
Project Proposal Document
(Signed off or agreed with your supervisor)

Implementation

Complete Research write it up. Complete main chapter.
Vision Document.

A vision document defines the high-level scope and purpose of a project. A clear statement of the problem, proposed solution, and the high-level features of a product helps establish expectations and reduce risks.

The template should include the following:

- Purpose / Introduction
- Scope / Outline
- Stakeholder and user descriptions
- Product overview / features
- User requirements, functional and non-functional, Look at reqs models. Priorities list (MOSCOW method). Each requirement is uniquely identified. A 3 digit number starting with 000.
- Solutions – technology etc
- References

Class template

https://www.ibm.com/support/knowledgecenter/en/SSYMRC_4.0.3/com.ibm.rational.rrm.help.doc/topics/r_vision_doc.html

Agree and signoff with supervisor.

Functional Specification: **(Determined in association with SQA Module)**

FUNCTIONAL SPECIFICATION TEMPLATE

A functional specification is a formal document used to describe in detail for software developers a product's intended capabilities, appearance, and interactions with users. The functional specification is a guideline and continuing reference point as the developers write the programming code.

Should include the follow

Add User stories for all **Must have** and **Should have** requirements. Format for story as follows

A unique number – Req no – Story no.

Title

Description : As a <type of user> I want <some goal> so that <some reason>.

Prioritise using Moscow

Estimate using hrs – 6-8 hrs max

Fully defined Acceptance criteria. A test for each criteria.

Check that each story complies with the INVEST model.

Can be done in word or excel or online tool. Tools such as Trello, sonicagile, TFS.

Proof read, spell check, use Wiegers checklist.

Signoff by supervisor.

Project Plan.

Table with a weekly break down. Use hrs for task times.

ID any risks to the project. If possible id and preventative action or corrective action.

Exit criteria

Vision Document

Functional Spec

Project Plan , Risk Analysis

Main research chapter complete.

Milestone: Formal Design

Monday 29th November 2017

Design Phase documents **(Determined in association with SQA Module)**

Sprints (First draft)

Key Class Diagrams

Design Phase Template

Input Criteria

Vision Document

Functional Spec

Signed off by your supervisor

Implementation

Detailed system design document: A documented description of a product to provide high-level guidance for the software development team. It is a single reference point that outlines the architecture of the software project and how each part needs to work. May include some or all of the following as agreed with your supervisor.

- Use cases – for all must have and should user stories. Use template on x drive
- Key Class Diagrams , VOPC– Use Visio, Creately.
- Sequence Diagram
- Relational schema/ Entity relationship diagram
- Interface – Story board, prototype, Screen layouts.

Exit Criteria

Detailed system design document agreed and signed off by supervisor

Submission 3: Prototype & Design Presentation

Wednesday 13th December 2017

FYP Document: (Work in progress)

Abstract

Research Chapters

Methodology Chapter & Design Chapter

Some Increments (agreed with supervisor) implemented/tested

Electronic submissions onto the X drive.

FYP Design Presentations 13th December 2017.

See Appendix A for Implementation Phase in Semester 8

Implementation phase

Entry criteria.

Detailed design document
Backlog of prioritized user stories (requirements)

Implementation

Sprint = 24hrs (1 week – 12hrs (5-6 hrs lab + 5-6 hrs))

Activities

- Decide on items for the sprint from backlog
- Extra research may be needed
- May need extra design details.
- May need to identify test. (Agree with supervisor the main classes to be tested.) Follow unit test plan on x drive.
- Coding. Adhere to coding standards.
 - o Java <http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>
 - Python <https://www.python.org/dev/peps/pep-0008/>
 - o C# <http://se.inf.ethz.ch/old/teaching/ss2007/251-0290-00/project/CSharpCodingStandards.pdf>

Use a static analyser tool.

- o Java - PMD
- o Python Pylint: <https://github.com/pycqa/pylint>
- o C# Gendarme - <http://www.mono-project.com/docs/tools+libraries/tools/gendarme/>
- o

Unit test if needed

Run Acceptance tests

Monitor progress (log time per task) Review at end of sprint (estimate Vs Actual) Adjust as need be.

Use the college referencing system.

Review of sprint plan next sprint.

Exit Criteria

User stories acceptance tests passed
Clean code
Unit tests passed
Plan for next sprint
Update Backlog.