

**Software Engineering and Testing. BSC Year 2, 2020/2021**

**(Assignment 2 - 20%)**

**Assessment 2: Requirements Document**

**Submitted by: Names, Student numbers**

**Submission date**

**Declaration**

I herby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ordinary Degree in Computing in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated.

Author: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Dated: \_\_\_\_\_\_\_\_\_\_\_\_\_

Author: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Dated: \_\_\_\_\_\_\_\_\_\_\_\_\_

**Table of Contents**

**. . . . .**

# Title

# Client

# Project Overview (1 paragraph)

The project definition:

* what is the project,
* what the software will do.
* the main components of the software system,
* how will it be used

# Document Revision

Rev. 1.0 date – initial version

# Scope (max 1/2 page)

The functionality the project/software will **include** and **exclude –** required**,** desirable**,** optional**.**

Scope & Functionality of the project should be clear so that client and developer have same expectations…

1. **Walkthrough Scenarios**

Who is going to use/interact with the software /system and how will they use it.

e.g. for vending machine user and serviceman…

1. **Software Requirements Analysis:**

***Functional Requirements:***

These are statements of services the system should provide – how the system should react to particular inputs and how it should behave in particular situations. Explicitly state what the system should do. Every major scenario should be represented by a use case. Diagrams are encouraged. UML Use case diagrams, Use case specifications (as legible screen dumps, typed listings or activity diagrams)

Can approach them from a *user* and *system* point of view.

*User* – high level abstract requirements, readable by someone with no detailed technical knowledge.

*System* – detailed description of what the system should do. Targeted at technical staff and project managers…

e.g.

# 5.1 User Requirements

# 5.2 System Requirements

5.2.1 Use Cases

5.2.2 Use Case Specification

5.2.3 Activity Diagrams

# 5.3 Non-functional Requirements:

These are constraints on the service or functions offered by the system e.g. timing constraints

# Graphical User Interface Design

# Technical Requirements and Feasibility:

System models – UML

Development language – Java

Persistent storage – database?

Interface & Software / Hardware APIs

## Conclusion (1-2 paragraphs)

Your conclusions and recommendations (feasibility of the proposed project)

Additional sections: Table of Contents, executive summary, Index

Checklist: Is your document complete and correct?

*Content:*

* Do the requirements state the customers’ needs
* Are you satisfied with all parts of the document
* Do you believe all parts are possible to implement
* Is each part of the document in agreement with all other parts
* Do the requirements avoid specifying a solution
* Do the requirements avoid specifying a design

*Completeness*:

* Are all the necessary interfaces specified – this includes input and output
* Are the specifications precise enough
* Are all sections from the document template included – if changed, why?

*Clarity*:

* Are all requirements reasonable?
* Is the level of details for each requirements appropriate?
* Are the requirements written in a language appropriate to the reader?
* Are all items clear and unambiguous?