

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

**(Assignment 3 - 20%)**

**Assessment 3: Design and Draft Implementation**

**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:

# *Abstract / Executive Summary (200 Words max)*

# Project Definitions (Steven)

* Purpose of document
* What is the project?
* Functional Specifications
* Main components of the software system,

# Document Revision

Rev. 1.0 date – initial version

# Methodology

System models – UML (Habiba)

**Use of, and necessity of OOAD (Piotr)**

Object-Oriented Analysis and Design (OOAD) is crucial in software engineering for developing complex systems by breaking them down into small pieces. It enables the understanding of system requirements and defines objects, facilitating robust, scalable, and maintainable software.

What is a class diagram? (Rochelle)

A class diagram is a visual representation of structure of the system you are creating showing the classes as well as the attributes and methods of these classes. It also shows the relationships between classes. Class diagrams can also be used to help develop ERD’s needed for the database that the systems will need. Class diagrams can be used in many of the stages of developing software such as helping you find out what are the main components of your software and understand the requirements needed for the system.

Static Versus Dynamic Case Diagrams? (Rochelle)

Static case diagrams are only show which entities interact with each. They don’t show what occurs when this interaction happens. Dynamic diagrams do show what happens when they interact.

**What is an ERD? (Piotr)**

An Entity-Relationship Diagram (ERD) is a tool in database design, that shows the structure of data and how different entities relate to each other. Typically, we would use symbols such as rectangles to represent entities, diamonds to represent relationships, and lines to connect entities and relationships.

Purpose of using classes? (Habiba)

Volatile versus Persistent storage – Object Instances / Database? (Steven)

User Interface template chosen and how it can aid in executing the functional specification of the project. (Steven)

1. **Requirements (Habiba)**

4.1 Use Cases

4.2 Use Case Specifications

(Specifically – how use case specifications have been used as a means to develop the ***classes/attributes/methods*** and database ***tables***)

1. **Case Diagrams (Rochelle)**

**Class Diagram** – Show all relationships, multiplicities, associations, generalisations (inheritance), aggregations (compositions) - See lecture 4.

Paragraph to explain ALL design decisions.

**Entity Relationship Diagram** – Show all relationships, multiplicities,

**Conceptual Diagram**:A diagram of a company

Description automatically generated

**Physical ERD of DATABASE**:

A diagram of a computer

Description automatically generated

Paragraph to explain ALL design decisions.

# Conclusions (Piotr)

Your conclusions and recommendations as to how far the project has progressed.

Your changes to the original proposal that the design has revealed and necessitated.

Additional sections: Table of Contents, Table of Figures, References, Index

Checklist: Is your document complete and correct?

*Content:*

* Does the design include all requirements from the customers’ needs
* Are you satisfied with all parts of the document?
* Do you believe all parts have been implemented?
* Have you explained your methodology and design choices?
* Have you clearly articulated your understanding of the purpose of all diagrams created ?
* What are these diagrams? Why you need them? How were they developed?
* Is each part of the document in agreement with all other parts?
* Does the design create a solution for the initial proposal?

*Completeness*:

* Are all the necessary components specified?
* Are the design specifications precise enough?
* Are all sections from the document template included – if changed, why?

*Clarity*:

* Is the design reasonable?
* Is the level of details for each design section appropriate?
* Is the design written in a language appropriate to the intended audience of software engineering teams?
* Are all items clear and unambiguous?