

National College of Ireland

Object Oriented Software Engineering

(HDAIML_SEPBL_HDBC_SEP23OL_HDSDEV_SEP23, HDCSDEV_INT HDSDEV_JANBL_YR1, HDWD_SEPBL)

Project (100%)

Introduction

This project is designed to evaluate all the learning outcomes of the Object Oriented Software Engineering module as outlined below:

1. Demonstrate the conceptual, practical and technical skills of planning and monitoring a project plan using an appropriate CASE tool
2. Describe in detail the theory, concepts and methods pertaining to Software Engineering such as Agile and UML.
3. Create requirements using use case modelling concepts.
4. Demonstrate conceptual and technical skills in the analysis, design, implementation and testing of a software development solution individually or as part of a team.
5. Employ tools and techniques for Software Engineering.

Objectives

In teams of 2 or 3, your task is to design an application to address the requirements of a specific business domain. You should select the business domain from the table below based on the **last digit of the student number** of the member of the team that has the **first letter of their surname nearest to the start of the alphabet**.

Last digit of student number	Problem Domain
0 or 5	Pharmacy
1 or 6	Garden Centre
2 or 7	Restaurant
3 or 8	Motor Mechanic
4 or 9	Newsagent

Alternatively, you may choose a business domain familiar to one or more members of your team, provided that you discuss this with your lecturer and obtain their prior approval before starting work on the project.

The project is divided into three parts:

- Part A is worth 45% of the total mark and focusses on the analysis and design of the system.
- Part B is worth 45% of the total mark and concentrates on the strategy to be used for implementing the system.
- Part C is worth 10% of the total mark and comprises a reflective journal detailing each individual team member's learning experience.

Part A (45 marks)

In this first part of the assessment you are expected to project plan a project and apply UML diagrams to the resolution of the problem area identified above.

You should produce a report detailing your design. The report should include the following:

1. As a team, identify and describe the actors and construct a use case diagram. Describe what your use case diagram is showing (approx 100 words) (7 marks)
2. Each team member should select one distinct use case from the use case model and describe it in detail. The use case must contain an alternate flow or exceptional flow. (5 marks)
3. As a team, create a glossary in which all project-related terminology that requires clarification is both listed and fully defined. (3 marks)
4. As a team, create a conceptual class diagram modelling the architecture of the proposed system. The conceptual class diagram should demonstrate the use of **three or more** of the following:
 - attributes;
 - relationships;
 - navigability;
 - association;
 - multiplicity; and
 - composition.

Describe what the conceptual class diagram is showing (approx 100 words) (10 marks)

5. Each team member should select one of the system operations and should develop a contract for it. (5 marks)

6. Using appropriate design patterns, each team member should create a communication diagram based on the contract they developed in task 5, including a description of what the diagram is showing (approx 100 words). (7.5 marks)
7. Discuss how risk, quality and communication will be managed in your project. Provide justifications for your choices. (5 marks)
8. Describe and justify the development methodology you will follow. (2.5 marks)

Part B (45 marks)

In this part of the assessment you are expected to apply agile methods such as TDD and SCRUM to the resolution of the selected problem domain.

1. Using an appropriate object oriented language (e.g. Java), fully develop the classes required to implement one of the use cases described in part 1 of the assessment. The use case must contain at least one alternate flow. (15 marks)
2. Fully test the classes developed for Q1 above, naming and justifying the test methodology followed. Describe the tests carried out, detailing how they will ensure that the classes are free from errors and detail the results of the tests. A minimum of 3 tests must be created and run. (15 marks)
3. Provide detailed artefacts of the agile methodology followed, such as user stories, backlogs and burndown charts. (15 marks)

Part C (10 marks)

This consists of an individual weekly reflective journal detailing each team member's learning experience. There should be a journal entry of 250-300 words for each week from week 5 onwards. This should explore questions such as:

- How can I related what I have learned to my work on the team project?
- What difficulties have I encountered in applying my knowledge when contributing to the team project?
- What would be the causes for those difficulties and how could I overcome them?
- Are there issues in the way the team is working and how could they be addressed?

Software Tools

Visual Paradigm is recommended for the production of UML diagrams and Agile artefacts. NCI has a license for the professional version of Visual Paradigm. Ask your lecturer for details. For the development and testing of classes, NetBeans is the recommended tool.

Submission

Note: Each part has tasks that are either carried out as a team or as individual members of the team. Answers for individual tasks should all be included in the same report, with the team member who wrote the answer clearly identified.

For example, for task 5 in Part A, you should have a section as follows:

System Contracts

Student Number for Team Member A

Answer from team member A...

Student Number for Team Member B

Answer from team member B...

Part A of the report should be uploaded as a single PDF document to the TurnItIn link on Moodle **by 23:59 on Friday November 10th, 2023** at the very latest. There will also be a link for submitting supporting files, such as those created in Visual Paradigm. Submission of these files is optional but recommended. There should be **one upload per team**.

Part B of the report should be uploaded as a single PDF document to the TurnItIn link on Moodle **by 23:59 on Friday December 15th, 2023**. You will also be required to upload your code artefact. This should be a zip file containing the classes developed and the suite of test carried out. As before, there will also be a link for uploading supporting files, such as the Agile artefacts produced in Visual Paradigm. There should be **one upload per team**.

Part C of the report should be uploaded as a single PDF document to the TurnItIn link on Moodle **by 23:59 on Friday December 15th, 2023**. There should be **one upload per student**.

Late submissions will not be accepted unless an extension has been officially approved.

Academic Integrity

Your submission must be entirely the work of your team members. Any written work created by others must be properly cited and should be paraphrased or summarised where possible, otherwise it should be included in quotes. Figures not created by you should include an acknowledgment detailing the name(s) of the creator(s).

Code found on the internet should not be claimed as your own, but instead a comment should be included in the source code indicating where you obtained it.

The use of large language models such as ChatGPT is **strictly prohibited**.

Students are strongly advised to familiarise themselves with the Guide to Academic Integrity produced by the NCI Library¹.

Note: All submissions will be electronically screened for evidence of academic misconduct, e.g. plagiarism, collusion and misrepresentation. Any submission showing evidence of such misconduct will be referred to the college's academic misconduct committee for disciplinary action.

¹<https://libguides.ncirl.ie/academicintegrity>