**School of Engineering, Computing and Mathematics**

**Oxford Brookes University**

**COMPT5047: Applied Software Engineering**

**(Semester 1, 2025-2026)**

Coursework Specification

Software Engineering of a Modern Computer Application

# Assessed Learning Outcomes

This coursework counts as **100%** of the total assessment of this module. It is designed to assess your attainment of the following learning outcomes.

|  |  |
| --- | --- |
| 1 | Demonstrate an understanding of the role of requirements analysis and specification in software engineering and to be able to use this knowledge to create use case models and functional models of computer applications. |
| 2 | Demonstrate an understanding of the relationship between requirements and design and to be able to apply the knowledge to create structural and behavioural models of computer applications. |
| 3 | Critically evaluate and utilise design paradigms of object-oriented analysis and design, component-based design, and service-oriented design. |
| 4 | Use software modelling language such as UML and modelling tools in the context of model-driven software engineering. |
| 5 | Work in a group to apply the knowledge and skills developed in this module |

|  |  |
| --- | --- |
| **Engineering Council AHEP4 LOs assessed** | |
| C3 | Select and apply appropriate computational and analytical techniques to model complex problems, recognising the limitations of the techniques employed |
| C5 | Design solutions for complex problems that meet a combination of societal, user, business and customer needs as appropriate. This will involve consideration of applicable health & safety, diversity, inclusion, cultural, societal, environmental and commercial matters, codes of practice and industry standards |
| C6 | Apply an integrated or systems approach to the solution of complex problems |
| C14 | Discuss the role of quality management systems and continuous improvement in the context of complex problems |
| C16 | Function effectively as an individual, and as a member or leader of a team |

# Problems to Be Solved

In this coursework, you are required to work **in a group of 4 students** as a software engineering team to develop[[1]](#footnote-1) a given modern computer application by applying appropriate software engineering methodology and technologies. Each member of the team will conduct software engineering activities for developing one subsystem and collaborate with the other members of the team to ensure that these subsystems are integrated and interact with each other to form a consistent system. A brief description of the specific application is given in the document COMP5047-CW-Case Study, which is available on the module’s Moodle website.

You are required to perform a series of software engineering tasks specified in Section 3 and to produce a set of software engineering documents in order to demonstrate:

1. you understand the principles of software engineering methodologies,
2. you are capable of selecting and applying appropriate software engineering techniques, and
3. you have the skills of using advanced software engineering tools.

# Tasks to Do

The following specifies the tasks to be completed in this coursework and the distribution of marks in the assessment. A more detailed marking schemeis given in the file COMP5047-CW-Detailed Marking Scheme, which is also available on the module’s Moodle website.

## **Task 1: Project Management (20 Marks)**

1. **Team Structure**. Each student will be assigned at random to a team of four students from the class. The team must establish a project management scheme (e.g. elect a team leader, or take turns to be the team leader), and a collaboration mechanism (e.g. WhatsApp group, Project Wiki, etc.). The group must work closely throughout the whole semester as a well organised software engineering project team. No change of group is allowed.
2. **Teamwork Facility**. Each team must set up a GitHub repository for the coursework project to host the documents of the project.
3. **Documentation of Project Management**. The team must meet regularly (at least once a week) at a fixed time and day of the week. Each meeting should have an agenda before the meeting and a minute to record the meeting contents, which should cover at least the following aspects:
4. The attendances to the meeting,
5. Progress since the previous meeting, especially the state of the completion of actions.
6. A list of actions to take place after the meeting, and
7. Issues to be solved.

The meeting agendas and minutes must be uploaded to GitHub immediately after it is created.

1. **Allocation of Responsibilities**. Each member of the coursework group must be responsible to the development of one the following subsystems.
2. *USU Student App*: a mobile app to run on smart phones for students to participate in the student union’s activities.
3. *Student Union Management System*: a web-based application to run on desktop or tablet computer for officers of university-specific student union to manage and operate the student union.
4. *USU Operation System*: a web-based application to run on desktop or tablet computer for USU officers to manage the USU federation of student unions and realise its functions.
5. *Society Leader App*: a mobile app to run on mobile phones and on desktop or tablet computers for society leader to organise and operator societies.

***Note:***

(1) The members of the team should work on different subsystems, while the whole team should develop a coherent system where the subsystems interact with each other.

(2) Each subsystem will inevitably have two parts: one part on the user’s computer or mobile device and another part running on servers on the cloud. The parts on the Cloud that belong to different subsystems should be properly integrated so that they interact with each other to enable functions to deliver to various types of users. Moreover, there may well be components running on the Cloud common to more than one subsystem and/or do not belong to any specific subsystem. The team should also work together to perform the software engineering activities in order to develop such common components.

1. **Reports on Teamwork Performance**. By the end of the module, two reports on project management and teamwork performance should be written and included in coursework submission.
2. *Team Report on Project Management* (10 Marks, Team effort)*.* Each team must write a report on the project management work done by the group.
3. *Individual Report on Teamwork* (10 Marks, Individual effort)*.* Each member of the group must also write a report on your performance in project management with focus on your contribution to the project management, your technical contributions to the common parts of the system, and your collaboration with the other members of the group.

## **Task 2: Analysis and Specify Software Quality Requirements (20 Marks)**

In this task, each member of the team will work as a requirements analyst to produce a document that defines the quality requirements on your subsystem. You are required to submit a document as a part of the coursework submission that defines the quality requirements on **ONE** functional requirement given in the case study document. The definition of quality requirements should clearly specify the requirements on the following quality attributes.

* Security and Privacy protection
* Performance
* Reliability
* Scalability

## **Task 3: Specification and Modelling Software Functional Requirements (20 Marks)**

In this task, you will work as a requirements analyst to produce a UML model of the software system to be developed using the software modelling tool Papyrus. The UML model should contain the following types of models.

1. *Use Case Model* (10 Marks, Individual effort): Each member of the team should develop one Use Case Diagram to define the use cases of the subsystem to specify the scope of the software engineering project.
2. *Activity Model* (10 Marks, Individual effort): Each team member should **select one use case** of your subsystem to produce one Activity Diagram for the selected use case to specify the interactions between a user and the subsystem.

## **Task 4: Software Architectural Design (20 Marks)**

In this task, you will work as a software architect to produce an ***Architectural Design*** of the system in the *microservices architectural style*. The design should be at two different levels as follows.

1. *Architecture of the subsystem* (10 Marks, Individual effort): Each member of the team should produce an architectural design of your subsystem with focus on the microservices your subsystem provides and the microservices that your subsystem requests and other subsystems provides.
2. *Architecture of the whole system* (10 Marks, Team effort): The team should produce an architectural design of the whole system through integrating the subsystems together.

***Note****:*

1. *You should specify the architectural designs in the UML component diagrams with a set of component nodes that represent microservices and a set of interfaces to represent the APIs that are the connectors between them.*
2. *The components and connectors in architectural designs, including their methods and parameters, should be specified in a textual documentation to define their functionalities and meanings.*
3. *In this part of the design, you are not required to give the internal structure of the components, which will be the work of the task on detailed design.*

## **Task 5: Software Detailed Design (20 Marks)**

In this task, you will work as a software designer to produce a ***Detailed Design*** of the subsystem for object-oriented implementation of the components (i.e. the *microservices*) in your *architectural design* of the subsystem. The design should specify the components from both structural and behavioural aspects in UML.

1. *Structural Model* (10 Marks, Individual effort): Each student should **select one component** in the architectural design of your subsystem and develop a Class Diagram to define the structure of the component.

***Note***:

1. *The selected component must be in the architectural model of your subsystem, but not in any other subsystems.*
2. *The UML class diagram must contain the classes and relationships between them. You must give the attributes and methods of the classes.*
3. *The class model must be consistent with the architectural design that you made in Task 4.*
4. *Behaviour Model* (10 Marks, Individual effort): Each student should produce a UML Sequence Diagram for the same component selected in Task 5(a) to define the dynamic behaviour of the component.

***Note****:*

1. *The sequence diagram should specify the internal process of the interactions between the objects inside the component. It must be consistent with the structural model that you produced in Task 5(a).*
2. *The behaviour model must cover all possible scenarios of the operations of the component using advanced modelling facilities such as frames.*

# Submission of Coursework

## When to submit

The submission deadline is at **23:00pm on 5th Dec. 2025 (Friday of Week 11)**.

## What to be submitted

Each group should submit one compressed (zip) file that contains a set of files listed in the table below. The file names and their contents to be included in the coursework submission must follow the naming convention given in the table below.

***Note****: The text in red ink below should be replaced by your coursework group number.*

Table 1. Files to be included in submission

|  |  |  |  |
| --- | --- | --- | --- |
| **File name** | **Format** | **Example** | **Content** |
| COMP5047\_CW\_ Gxx.zip | Zip file | COMP5047\_CW\_G01.zip | The zip file should contain all the files of the coursework submission, where Gxx is your coursework group number. |
| Cover\_Gxx.pdf | PDF | Cover\_G01.pdf | The standard group coursework cover page. |
| TeamPart\_Gxx.pdf | PDF | Team\_G01.pdf | It should contain (a) team report on project management. (b) the work on Task 4(b), i.e. the architectural design of the whole system, including the UML component model and the textual documentation of the design. |
| Member\_Gxx\_StdNum1.pdf | PDF | Member\_G01\_1900001.pdf | The work of student 1900001, including (a) specification of quality requirements, (b) one UML use case diagram, (c) one UML activity diagram, (d) one architectural design of the subsystem in UML component diagram, (e) one UML class diagram for one component of the subsystem, (f) one UML sequence diagram for the same component of the subsystem, (g) a personal report on teamwork. |
| Member\_Gxx\_StdNum2.pdf | PDF | Member\_G01\_1900002.pdf | The work of student 1900002; *See above.* |
| Member\_Gxx\_StdNum3.pdf | PDF | Member\_G01\_1900003.pdf | The work of student 1900003; *See above.* |
| Member\_Gxx\_StdNum4.pdf | PDF | Member\_G01\_1900004.pdf | The work of student 1900004; *See above.* |

* Your files must be formatted as follows:
* It must be word-processed in 11-point Arial font and single-spaced.
* All pages must be numbered.
* Margins must be 1 inch on top, bottom, and both left and right sides of the paper.
* All pages must in A4 size.
* Student numbers must be entered on the cover page and state which subsystem is developed by which student.
* The files should not contain your name(s).

*Please Note:*

* *You are required to cite the work of others used in your solution and include a list of references using the university recommended referencing style.*
* *Please refer to university’s regulation on academic conduct and plagiarism.*
* *If you have used AI empowered tools, you must declare how you have used it in the coursework.*

## Where to submit

You must upload **one** zip file containing all components to the module's Moodle website.

The original editable files must be uploaded to GitHub with all the history of updating the artefacts.

## Feedback on your coursework

* Formative feedback will be given at the Week 3 Practical Class on your setup of project management facility.
* The formal summative feedback on your coursework will be provided within 2 weeks after your submission of the coursework.
* Informal verbal feedback can be obtained from the lecturer during the delivery of the module during practical classes. The students are encouraged to show their work to the lecturer and seek comments and advice.

1. No coding is required in this coursework. [↑](#footnote-ref-1)