Coursework Specification

Software Engineering of A Modern Computer Application

1 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 software lifecycle and be able to critically analyse a problem to decide on a relevant process model.
2	Demonstrate an understanding of the role of risk in a project and be able to identify and manage both risk and the impact of change on a project.
3	Demonstrate an understanding of the role of requirements analysis and elicitation and the relationship between requirements and design and use this knowledge to be able to create functional and non-functional requirements for a project
4	Be able to critically evaluate and utilise design paradigms of object-oriented analysis and design, component-based design, and service-oriented design.
5	Describe and employ software techniques, in the implementation of designs to achieve desired software quality including reliability, efficiency and robustness
6	Understand and be able to apply a range of testing methods and techniques.
7	Use refactoring in the process of modifying a software component
8	Understand how software reliability contributes to system reliability and be able to apply multiple methods to develop reliability estimates for a software system.

2 The problem to be solved

In this coursework, you are required to work in a group of 4 students as a software engineering team to develop a given modern computer application by applying appropriate software engineering methodology and technologies. Each member of the team will be in charge of one subsystem and collaborate with the other members of the team to ensure the interactions between subsystems work, and each subsystem performs as an integral part of the whole system. A brief description of the specific application is given in the document "CHC6173-2023 Case Study," which is available on the student website. You are required to perform a series of software engineering tasks and to produce a set of software engineering documents in order to demonstrate:

(a) your understanding of the principles of software engineering methodologies

(b) your capability to select and apply appropriate software engineering techniques, and

(c) your skills in using advanced software engineering tools.

3 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 scheme is given in the file 'CHC6173-2023 Detailed Marking Scheme', which is available on the student website.

a. Task 0: Organisation of the Team.

Each member of the coursework team must be responsible to the development of one the following subsystems:

- HealthGuardian Patient: to deliver functions for patients in the form of mobile app
- HealthGuardian doctor: to deliver functions for doctors in the form of mobile and desktop/web-based app
- HealthGuardian pharmacy: to deliver functions for pharmacist in the form of desktop/web-based app
- HealthGuardian Lab: to deliver functions for laboratory technician staff in the form of desktop/web-based app

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.

b. Task 1: Software Modelling and Specification (40 Marks)

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

(a) *Use Case Model* (10 Marks, Team effort): The team should work together to develop one Use Case Diagram to cover all types of users and all the use cases of the whole system (including all subsystems) to specify the scope of the software engineering project.

(b) *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 system.

(c) *Structural Model* (10 Marks, Individual effort with collaboration to the team): Each member of the team should develop one Class Diagram to define the conceptual structure of the subsystem by defining the classes in the subsystem and their relationships. *Note*: (a) Classes must contain attributes and methods. (b) Overlap classes (i.e. those in more than one subsystem) must be consistent between team members' models.

(d) *Behaviour Model* (10 Marks, Individual effort): Each member of the team should produce one Sequence Diagram to define the internal process of the interactions between the objects inside your subsystem in the use case that you selected in task 1(b).

c. Task 2: Software Architectural Design (30 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 abstraction levels as follows.

(a) *Overall architecture of the whole system* (10 Marks, Team effort): At this level of abstraction, the team should produce an architectural design of the whole system. You should specify the architecture in the UML *component diagram* with a set of components (i.e. microservices plus databases) and the connectors between them. The components and connectors should be described in a *textual documentation* of their functionalities. *Note*: In this part of the design, you are not required to give the internal structure of the components and connectors.

(b) *Architecture of the subsystem* (20 Marks, Individual effort): At this level of abstraction, you should produce an architectural design of your subsystem. You should specify the internal structures of the components and the connectors in the overall architecture of Task 2(a). That is, to give the classes in the components and connectors. You are required to use the *UML component diagram* to present the design of your subsystem.

d. Task 3: Software Testing (30 Marks)

In this task, you will work as a software quality engineer to develop a ***Software Test Plan***. Your test plan should consist of two parts:

(a) *Unit test plan* (15 Marks, Individual effort): One class of objects of your subsystem should be selected to produce a detailed unit test plan. It should include a set of test cases that adequately covers all the methods of the class.

(b) *System test plan* (15 Marks, Individual effort): You should select one use case of your subsystem. System Test Plan should be developed for testing the application on the selected use case. The test plan should contain test cases that cover all different scenarios of the use case.

4 Submission of Coursework

4.1 When to submit

The submission deadline is at **23:59pm on 20th December 2023**.

4.2 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 convention given in the table below.

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

Table 1. Files to be included in submission

File name	**Format**	**Example**	**Content**
CHC6173_CW_ Gxx.zip	Zip file	CHC6173_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	TeamPart_G01.pdf	The use case diagram and architectural design of the whole system.
Member_Gxx_StdNum1.pdf	PDF	Member_G01_1900001.pdf	The work of student 1900001, including one activity diagram, one class diagram, one sequence diagram, one component diagram of the subsystem, one unit test plan and one system test plan.
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.*

Please note that 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.

4.3 Where to submit

You must upload **one** zip file containing all components to the student website.

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

4.4 Feedback on your coursework

- 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.