LEVEL 4

Software Engineering

Student Guide

Modification History

Version	Date	Revision Description
V1.0	March 2024	For release

Published by: NCC Education Limited, Adamson House, Towers Business Park, Wilmslow Road, Didsbury, Manchester M20 2EZ, UK.

Tel: +44 (0) 161 438 6200 Fax: +44 (0) 161 438 6240 Email: info@nccedu.com
http://www.nccedu.com

CONTENTS

1. Unit Overview and Objectives

This unit aims to provide the learner with a broad introduction to the software engineering methods and models commonly employed in a software development industry, along with an understanding of the underlying theoretical frameworks and of associated issues such as project management and security of software systems development.

At the end of the unit, students should be able to:

1. Explain the need for an engineering approach to software development
2. Explain software engineering key practices and principles
3. Create models of software data and processes using object oriented modelling approaches
4. Describe and evaluate tools and techniques to enhance productivity and quality of software

2. Learning Outcomes and Assessment Criteria

Learning Outcomes; The Learner will:	**Assessment Criteria;** The Learner can:
1. Explain the need for an engineering approach to software development	1.1 Discuss the nature of Software 1.2 Explain the motivation for the development of software engineering 1.3 Describe the cost of software maintenance 1.4 Explain software quality (code, quality, security) 1.5 Explain Software Engineering 1.6 Discuss software engineering ethics
2. Explain software engineering key practices and principles	2.1 Describe the software product lifecycle and SDLC models 2.2 Explain why alternative software development lifecycles have been developed. 2.3 Discuss the role of requirements engineering, system modelling, testing, maintenance, reuse, project management, cyber security in software engineering. 2.4 Explain software engineering design principles 2.5 Discuss approaches to secure software development models
3. Create models of software data and processes using object oriented modelling approaches	3.1 Explain the purpose and range of modelling to produce quality software. 3.2 Create appropriate UML models for a well-defined scenario of limited scope. 3.3 Explain how the models embody the principles of good practice in software engineering.
4. Describe and evaluate tools and techniques to enhance productivity and quality of software	4.1 Explain Object Oriented programming 4.2 Describe project management tools and methods for ensuring software quality (Planning, monitoring, control, risk, quality, organisation) 4.3 Explain risk assessment and control 4.4 Explain methods and techniques for software testing and change management (quality, monitoring & control) 4.5 Discuss the role of documentation in software engineering (Risk register, Gantt chart etc)

3. Syllabus

Syllabus			
Topic No	Title	Proportion	Content
1	Introduction to Software Engineering Module	1/12 2 hours of lectures 2 hours of tutorials	• Defining Software and its unique characteristics. • Software application domains • The Software crisis and issues in software development • Software maintenance: types, costs ISO/IEC 14764 • Definition of software quality • Factors contributing to measures of software quality • Software engineering ethics ***Learning Outcomes: 1***
2	Introduction to Software Engineering Key Practices and Principles	1/12 2 hours of lectures 2 hours of tutorials	• Software product lifecycle and SDLC models • Teamwork • Project management, productivity • Requirements engineering • System Design, modelling, reuse • Testing • Configuration management • Maintenance • Software reuse • Object-oriented programming • Discuss approaches to secure software development models ***Learning Outcomes: 2***
3	Software Development Life Cycle Models	1/12 2 hours of lectures 2 hours of tutorials	• Software process models • Waterfall model • Incremental model • Agile development • Criteria to select SDLC model appropriate to each type of project. • Coping with change • Process improvement ***Learning Outcomes: 2,3***

4	Requirements Engineering	1/12 2 hours of lectures 2 hours of tutorials	• Functional and Non-Functional Requirements (different types of requirements) • Requirements engineering processes • Requirements elicitation & Analysis • Requirements specification • Requirements validation &Management • Requirements change ***Learning Outcomes: 2***
5	System Modelling and Design	1/12 2 hours of lectures 2 hours of tutorials	• Design principles: Separation of concerns, Encapsulation & Abstraction • Component based software engineering • UML modelling: interaction, structural and behavioural models • Use Cases, Class diagrams, object diagrams, activity diagrams, sequence diagrams. • Patterns and architecture. ***Learning Outcomes: 2,3***
6	Software Implementation & Testing	1/12 2 hours of lectures 2 hours of tutorials	• Implementation issues: reuse, configuration management, open-source development • Validation, Verification and Testing • Development testing • Test driven development • Test Techniques • Release testing • User Testing • Traceability • Software evolution ***Learning Outcomes: 2,4***
7	System Dependability and Security	1/12 2 hours of lectures 2 hours of tutorials	• Security defined: dependable systems • Secure software development models • Threat modelling • Code security • Software quality assurance • Software security testing • Reliability engineering: availability and reliability, fault tolerance • Safety Engineering: safety critical systems, formal specifications ***Learning Outcomes:2, 3, 4***

8	Project Management	1/12 2 hours of lectures 2 hours of tutorials	• Project Planning: scheduling, estimating, Gantt charts. • Risk assessment and management • Teamwork • Quality management ***Learning Outcomes: 4***
9	Mini project 1	1/12 1 hour of lectures 3 hours of seminars	• Analysis and Planning • Requirements Gathering ***Learning Outcomes: 2***
10	Mini Project 2	1/12 1 hour of lectures 3 hours of seminars	• Design and Prototyping • Software Development ***Learning Outcomes: 3***
11	Mini Project 3	1/12 1 hour of lectures 3 hours of seminars	• Testing and Deployment ***Learning Outcomes: 4***
12	Mini Project 4	1/12 1 hour of lectures 3 hours of seminars	• Project Presentation and Demonstration ***Learning Outcomes: All***

4. Related National Occupational Standards

The UK National Occupational Standards describe the skills that professionals are expected to demonstrate in their jobs in order to carry them out effectively. They are developed by employers and this information can be helpful in explaining the practical skills that students have covered in this unit.

Related National Occupational Standards (NOS)
Sector Subject Area: IT and Telecoms **Related NOS:** ESKITU020, ESKITU021, ESKITU041

5. Resources

Lecturer Guide: This guide contains notes for lecturers on the organisation of each topic, and suggested use of the resources. It also contains all of the suggested exercises and model answers.

PowerPoint Slides: These are presented for each topic for use in the lectures. They contain many examples which can be used to explain the key concepts. Handout versions of the slides are also available; it is recommended that these are distributed to students for revision purposes as it is important that students learn to take their own notes during lectures.

Student Guide: This contains the topic overviews and all of the suggested exercises. Each student will need access to this and should bring it to all of the taught hours for the module.

5.1 Additional Hardware and Software Requirements

Hardware: Desktops with Windows OS

Software: UML

6. Pedagogic Approach

Suggested Learning Hours						
Guided Learning Hours				Assessment	Private Study	Total
Lecture	Tutorial	Seminar	Laboratory			
20	16	12	-	30	72	150

The teacher-led time for this module is comprised of lectures, laboratory sessions and tutorials. The breakdown of the hours is also given at the start of each topic, with 5 hours of contact time per topic.

6.1 Lectures

Lectures are designed to introduce students to each topic; PowerPoint slides are presented for use during these sessions. Students should also be encouraged to be active during this time and to discuss and/or practice the concepts covered. Lecturers should encourage active participation and field questions wherever possible.

6.2 Tutorials

Tutorials provide tasks to involve group work, investigation and independent learning for certain topics. The details of these tasks are provided in this guide and also in the Student Guide. They are also designed to deal with the questions arising from the lectures, laboratory sessions and private study sessions.

6.3 Laboratory Sessions

During these sessions, students are required to work through practical tutorials and various exercises. The details of these are provided in this guide and also in the Student Guide. Some sessions will require more support than others as well as IT resources. More detail is given in this guide.

6.4 Private Study

In addition to the taught portion of the module, students will also be expected to undertake private study. Exercises are provided in the Student Guide for students to complete during this time. Teachers will need to set deadlines for the completion of this work. These should ideally be before the tutorial session for each topic, when Private Study Exercises are usually reviewed.

7. Assessment

This module will be assessed by means of a mini project + report worth 100% of the total mark. These assessments will cover the learning outcomes and assessment criteria given above. Sample assessments are available through the NCC Education Virtual Learning Environment (http://vle.nccedu.com/login/index.php) for your reference.

8. Further Reading List

A selection of sources of further reading around the content of this module must be available in your Accredited Partner Centre's library. The following list provides suggestions of some suitable sources:

- Sommerville, I. (2016). *Software Engineering Edition 10th,*
  ISBN- 0133943038
  ISBN- 978-0133943030

- Mohapatra, Pratap K.J. Software Engineering: A Lifecycle Approach
  ISBN-9788122427219
  ISBN-9788122428469

Topic 1: Introduction to the Software Engineering Unit

1.1 Learning Objectives

This topic provides an overview of the unit and an introduction to software engineering. It aims to help students develop an understanding of the software and software development concepts and the importance of software engineering as a systematic method to resolve software development challenges.

On completion of the topic, students will be able to:

- Discuss the nature of software
- Explain the motivation for the development of software engineering
- Describe the cost of software maintenance
- Explain software quality
- Discuss software engineering ethics

1.2 Pedagogic Approach

Information will be transmitted to the students during the lectures. They will then investigate the topic further during the laboratory sessions. Private study exercises will be provided to further enhance the students' understanding of the topic.

1.3 Timings

Lectures: 2 hours

Tutorial Sessions: 2 hours

Private Study: 5 hours

1.4 Lecture Notes

The following is an outline of the material to be covered during the lecture time and should be read in conjunction with the slides provided.

The structure of this topic is as follows:

- Unit Overview
- Session Overview
- Software definitions
  - Software characteristics
  - Software application domains
- Software crisis and issues in software development
- Software Engineering
- Software Quality Measurement & Factors

- Software Engineering Ethics

1.5 Tutorial Sessions

The time allocation for this topic is 2 hours.

Exercise 1:

Give a definition for “software” and explain different types of software by examples.

Exercise 2:

What are software characteristics and how they may dispose challenges in software development?

Exercise 3:

What is the definition of quality and name some methods for software quality checking.

Exercise 4:

Explain “Product” and “Public” as 2 principles of ACM Code of Practice for software engineers.

Exercise 5:

What is the definition of “Software Engineering”?

Exercise 6:

What are elements of software process?

Exercise 7:
List 3 challenges that software development projects may face with examples:

1.6 Private Study

The time allocation for private study in this topic is expected to be 5 hours.

Exercise 1:

Take your time to read ACM "Software Engineering Code of Practice" from below link.

- *https://ethics.acm.org/code-of-ethics/software-engineering-code/*

Then discuss how you would apply the practices that are relevant to below cases:

- You are working in a game development software company and recently attended a conference on gender discriminately in game applications. You wonder if your developed games have similar issues.
- You are developing a mobile application for disabled people which they can interact with to send their health records and issues to their GP and receive medical advice and prescriptions.
- A client has contacted you and ordered a website to be developed. In the following meeting with the client you discover that they requested user interface pages are exactly same or very similar to their competitor's web.

Topic 2: Introduction to Software Engineering Key Practices and Principles

2.1 Learning Objectives

In this session students will get deep knowledge of SDLC and its stages. They will also get into cyber security principles and how and by which methods these principles may apply and implement in SDLC. Finally they will learn about skills in project management.

On completion of the topic, students will be able to:

- Describe the software product lifecycle
- Explain why alternative software development lifecycles have been developed
- Discuss the role of requirements engineering, system modelling, testing, maintenance, reuse, project management, cyber security in software engineering
- Explain software engineering design principles

2.2 Pedagogic Approach

Information and theory of the topic will be presented to the students during lectures. They will then practise the skills during the tutorial sessions. Students are expected to undertake their own private study to understand the theory fully and put the lectures in context.

2.3 Timings

Lectures: 2 hours

Private Study: 5 hours

Tutorials: 2 hours

2.4 Lecture Notes

The following is an outline of the material to be covered during the lecture time and should be read in conjunction with the slides provided.

The structure of this topic is as follows:

- Software product lifecycle (SDLC)
- Teamwork
- Project management, productivity
- Requirements engineering,
- System Design, modelling, reuse
- Testing
- Configuration management
- Maintenance
- Cyber security

2.5 Tutorial Sessions

The time allocation for this topic is 2 hours.

Exercise 1:

List 7 stages of SDLC and describe each.

Exercise 2:

Compare 3 models of secure software development, MS SDL, Touchpoint and SecureCode.

Exercise 3:

What is CIA in cyber security?

Exercise 4:

What are the 6 elements of project management and how they must be balanced by changes (give examples)?

Exercise 5:

How team structure differs in agile and traditional software development life cycle can be controlled?

Exercise 6:

What are the approaches to estimation in project management?

2.6 Private Study

The time allocation for private study in this topic is expected to be 5 hours.

Exercise 1:

Read the detailed practices of Microsoft SDL from below link.

- https://www.microsoft.com/en-us/securityengineering/sdl/practices#practice1

Taking into account what you read about MS SDL practise and below case scenario, explain how you would implement the security practices and when in SDLC

- You are a project manager for a company who has recently been ordered for development of a client website. The business is to observe different local restaurants to customers, take their orders and deliver foods to their doors.

Topic 3: Software Development Lifecycle

3.1 Learning Objectives

In this lecture on software development lifecycle, students will delve into the systematic process of designing, creating, testing, and maintaining software systems. They'll gain insights into the phases of SDLC, from requirements analysis to deployment, understanding how each stage contributes to producing high-quality software. Additionally, students will explore methodologies including Agile or Waterfall, learning to adapt these frameworks to specific project needs.

On completion of the topic, students will be able:

- Discuss different models for SDLC.
- Explain why alternative software development lifecycles have been developed.
- Distinguish between different SDLC models and select the most appropriate for a project.
- Explain how change is controlled in agile methods.

3.2 Pedagogic Approach

Information and theory of the topic will be presented to the students during lectures. They will then practise the skills during the tutorial sessions. Students are expected to undertake their own private study to understand the theory fully and put the lectures in context.

3.3 Timings

Lectures:	2 hours
Private Study:	5 hours
Tutorials:	2 hours

3.4 Lecture Notes

The following is an outline of the material to be covered during the lecture time and should be read in conjunction with the slides provided.

The structure of this topic is as follows:

- Software process models
- Waterfall model
- Incremental model
- Agile development
- Coping with change
- Process improvement

3.5 Tutorial Sessions

The time allocation for this topic is 2 hours.

Exercise 1:

Explain 3 characteristics of the original waterfall model.

Exercise 2:

What are the main differences between incremental and iterative software development models?

Exercise 3:

What are the main values of agile software development?

Exercise 4:

Explain 4 metrics to select type of software development model mainly agile or waterfall for a project.

3.6 Private Study

The time allocation for private study in this topic is expected to be 5 hours.

Exercise 1:

Research and find a list of factors which impact the types of SDLC used for different types of software projects.

Topic 4: Requirements Engineering

4.1 Learning Objectives

In this lecture on Requirement Engineering, students will delve into the systematic process of eliciting and documenting system requirements. They'll gain insights into the requirements analysis methods and system specification report writing.

On completion of the topic, students will be able to:

- Discuss the role of requirements engineering.

4.2 Pedagogic Approach

Information and theory of the topic will be presented to the students during lectures. They will then practise the skills during the tutorial sessions. Students are expected to undertake their own private study to understand the theory fully and put the lectures in context.

4.3 Timings

Lectures:	2 hours
Private Study:	5 hours
Tutorials:	2 hours

4.4 Lecture Notes

The following is an outline of the material to be covered during the lecture time and should be read in conjunction with the slides provided.

The structure of this topic is as follows:

- Functional and Non-Functional Requirements
- Requirements engineering processes: requirements elicitation, requirements analysis, requirements specification, requirements verification and validation, and requirements management.
- Requirements change management.

4.5 Tutorial Sessions

The time allocation for this topic is 2 hours.

Exercise 1:

Explain difference between a user requirement and system requirement in requirements engineering. Provide examples of each.

Exercise 2:

Explain the 5 stages of requirements engineering.

Exercise 3:

Explain the process of change management:

4.6 Private Study

The time allocation for private study in this topic is expected to be 5 hours.

Exercise 1:

Research different methods of requirement elicitation and identify the advantages and disadvantages of each method.

Topic 5: Software Modelling and Design

5.1 Learning Objectives

This lecture is designed to explore fundamental principles of software design, to introduce various modelling techniques, including UML (Unified Modelling Language), and to visually represent software architecture, system components, and their interrelationships.

On completion of the topic, students will be able to:

- Explain software engineering design principles
- Explain the purpose and range of modelling to produce quality software.
- Create appropriate UML models for a well-defined scenario of limited scope.
- Explain how the models embody the principles of good practice in software engineering

5.2 Pedagogic Approach

Information and theory of the topic will be presented to the students during lectures. They will then practise the skills during the tutorial sessions. Students are expected to undertake their own private study to understand the theory fully and put the lectures in context.

5.3 Timings

Lectures: 2 hours

Private Study (Lab): 5 hours

Tutorial: 2 hours

5.4 Lecture Notes

The following is an outline of the material to be covered during the lecture time and should be read in conjunction with the slides provided.

The structure of this topic is as follows:

- System modelling
- Design principles
- UML models
- Design pattern

5.5. Tutorial Sessions

The time allocation for this topic is 2 hours.

Exercise 1:

What are the stages of the design process and which UML diagrams are used for each stage?

Exercise 2:

What are the main elements of a design pattern and what are the objectives of design patterns?

5.6 Private Study

The time allocation for private study in this topic is expected to be 5 hours.

Exercise 1:

Download UML software: https://www.umldesigner.org/download/

Find and explain the main elements of the following diagrams:

a) use case diagram

b) activity diagram

c) class diagram

Draw use-case, activity and class diagrams for an online food delivery software system.

Topic 6: Software Implementation & Testing

6.1 Learning Objectives

On completion of the topic, students will be able to:

- Discuss the role of system implementation , testing and maintenance in software engineering.
- Explain methods and techniques for software testing and change management

6.2 Pedagogic Approach

Information and theory of the topic will be presented to the students during lectures. They will then practise the skills during the tutorial sessions. Students are expected to undertake their own private study to understand the theory fully and put the lectures in context.

6.3 Timings

Lectures: 2 hours

Private Study: 5 hours

Tutorials: 2 hours

6.4 Lecture Notes

The following is an outline of the material to be covered during the lecture time and should be read in conjunction with the slides provided.

The structure of this topic is as follows:

- Software Implementation Issues
    - Software Reuse
    - Configuration Management
    - Open-source Development
- Validation, Verification and Testing
    - Development testing
    - Test driven development
    - Test Techniques
    - Release testing
    - User Testing

6.5 Tutorial Sessions

The time allocation for this topic is 2 hours.

Exercise 1:

List an explain 3 major software implementation issues.

Exercise 2:

What are levels of software reuse (with examples)**.**

Exercise 3:

Explain configuration management and its activities.

Exercise 4

Explain the stages of testing with example testing types of each stage.

6.6 Private Study

The time allocation for private study in this topic is expected to be 5 hours.

Exercise 1:

Do research on software automated testing, list them and explain which testing abstract level they are working in.

Topic 7: System Dependency & Security

7.1 Learning Objectives

On completion of the topic, students will be able to:

- Describe methods for ensuring software quality.
- Describe methods for Secure Software Engineering

7.2 Pedagogic Approach

Information and theory of the topic will be presented to the students during lectures. They will then practise the skills during the tutorial sessions. Students are expected to undertake their own private study to understand the theory fully and put the lectures in context.

7.3 Timings

Lectures: 2 hours

Private Study: 5 hours

Tutorials: 2 hours

7.4 Lecture Notes

The following is an outline of the material to be covered during the lecture time and should be read in conjunction with the slides provided.

The structure of this topic is as follows:

- Security defined: dependable systems
- Reliability engineering: availability and reliability, fault tolerance
- Safety Engineering: safety critical systems, formal specifications

7.5 Tutorial Sessions

The time allocation for this topic is 2 hours.

Exercise 1:

What are the properties of system dependability?

Exercise 2:

List 4 numbers of good practices for software dependable process:

Exercise 3:

What are formal methods in software engineering and name 4 formal methods.

Exercise 4:

Name and explain 2 key practices for secure software development.

7.6 Private Study

The time allocation for private study in this topic is expected to be 5 hours.

Exerise1:

- Identify differences between dynamic and static code security analysis
- find tools for each of these two security testing

Topic 8: Project Management

8.1 Learning Objectives

This topic provides an overview of essentials of “Software Project Management". This lecture will explore key concepts in software project management, covering project planning, scheduling, risk management, team collaboration, and effective communication, emphasizing practical strategies for successful project delivery.

On completion of the topic, students will be able to:

- Describe project management tools and methods for ensuring software quality (Planning, monitoring, control, risk, quality, organisation)
- Explain risk assessment and control
- Explain methods and techniques for software testing and change management (quality, monitoring & control)
- Discuss the role of documentation in software engineering (Risk register, Gantt chart etc

8.2 Pedagogic Approach

Information and theory of the topic will be presented to the students during lectures. They will then practise the skills during the tutorial sessions. Students are expected to undertake their own private study to understand the theory fully and put the lectures in context.

8.3 Timings

Lectures: 2 hours

Private Study: 5 hours

Tutorials: 2 hours

8.4 Lecture Notes

The following is an outline of the material to be covered during the lecture time. Please also refer to the slides.

The structure of this topic is as follows:

- Project Planning: Scheduling, Estimating, Gantt charts.
- Risk assessment and management
- Teamwork
- Quality management

8.5 Tutorial Notes

The time allowance for tutorials in this topic is 2 hours.

Exercise 1:

What is CASE tool and what are their application in SDLC?

Exercise 2:

How is project management defined and which factors influence it?

Exercise 3:

What is risk management process and bring 3 examples of project risks.

Exercise 4: what is the human motivation hierarchy and what is its role in people management:

Exercise 5:

What is the difference between team management in agile and traditional software development?

8.6 Private Study

The time allocation for private study in this topic is expected to be 5 hours.

Exercise 1:

Considering the following case study:

Alice is a software project manager working in a company that develops alarm systems. This company wishes to enter the growing market of assistive technology to help elderly and disabled people live independently. Alice has been asked to lead a team of 6 developers that can develop new products based around the company's alarm technology.

Alice's assistive technology project starts well. Good working relationships develop within the team and creative new ideas are developed. The team decides to develop a peer-to- peer messaging system using digital televisions linked to the alarm network for communications. However, some months into the project, Alice notices that Dorothy, a hardware design expert, starts coming into work late, the quality of her work deteriorates and, increasingly, that she does not appear to be communicating with other members of the team.

Alice talks about the problem informally with other team members to try to find out if Dorothy's personal circumstances have changed, and if this might be affecting her work. They don't know of anything, so Alice decides to talk with Dorothy to try to understand the problem.

After some initial denials that there is a problem, Dorothy admits that she has lost interest in the job. She expected that she would be able to develop and use her hardware interfacing skills. However, because of the product direction that has been chosen, she has little opportunity for this. Basically, she is working as a C programmer with other team members.

Although she admits that the work is challenging, she is concerned that she is not developing her interfacing skills. She is worried that finding a job that involves hardware interfacing will be difficult after this project. Because she does not want to upset the team by revealing that she is thinking about the next project, she has decided that it is best to minimise conversation with them.

Adapted from Sommerville, 2016

What actions should Alice as the project manager take to resolve the issue?

Topic 9: Mini Project 1

9.1 Learning Objectives

This topic introduces the first part of deliverable for the final coursework – Min Project.

- The objective of the mini project is to develop a software prototype by applying the software engineering knowledge acquired from this unit.

9.2 Pedagogic Approach

Information will be transmitted to the students during the lecture. They will then practise the skills during the laboratory sessions and extend their understanding during private study time.

9.3 Timings

Lectures: 1 hour

Laboratory Sessions: 3 hours

Private Study: 8 hours

9.4 Lecture Notes

The following is an outline of the material to be covered during the lecture time. Please also refer to the slides.

The structure of this topic is as follows:

- Introduction to your coursework
- Introduction to mini project
- Task 1, 2 of mini project

9.5 Laboratory Sessions

The laboratory time allocation for this topic is 3 hours.

Task 1: Project Description: the problems, aims and objective of the project, scope of the project, project plan. (15 marks)

Task 2: Analysis: requirements specification, use case diagrams (20 marks)

Requirements Specifications: Students should gather and document all the requirements for their project, including functional and non-functional requirements. This involves conducting interviews, surveys, and research to understand the needs of stakeholders. Students should document identified.

9.6 Private Study

The time allocation for private study in this topic is expected to be 8 hours.

Exercise 1:

Students should continue the Mini Project Task during their private study time which is Task 1 and Task 2 of the Mini Project brief.

Topic 10: Mini Project 2

10.1 Learning Objectives

On completion of the topic, students will be able to:

- Create appropriate UML models for a well-define scenario of limited scope.

10.2 Pedagogic Approach

Information will be transmitted to the students during the lecture. They will then practise the skills during the laboratory sessions and extend their understanding during private study time.

10.3 Timings

Lecture: 2 hours

Laboratory Sessions: 3 hours

Private Study: 8 hours

10.4 Lecture Notes

The following is an outline of the material to be covered during the lecture time. Please also refer to the slides.

The structure of this topic is as follows:

- Introduction to Mini project 2
- Task 3 of Mini project 2

10.5 Laboratory Sessions

The laboratory time allocation for this topic is 3 hours.

Task 3: Design: Class diagrams and Use case diagram (30 marks)

10.6 Private Study

The time allocation for private study in this topic is expected to be 8 hours.

Exercise 1:

Students should continue the Mini Project Task during their private study time which is Task 3 of the Mini Project brief.

Topic 11: Mini Project 3

11.1 Learning Objectives

On completion of the topic, students will be able to:

- Explain the role of implementation in software development SDLC.

11.2 Pedagogic Approach

Information will be transmitted to the students during the lectures. The tutorial will then provide an opportunity to review the key ideas and obtain further guidance and support.

11.3 Timings

Lectures: 1 hour

Private Study: 8 hours

Laboratory session: 3 hours

11.4 Lecture Notes

The following is an outline of the material to be covered during the lecture time. Please also refer to the slides.

The structure of this topic is as follows:

- Introduction to Mini project 3
- Tasks 4 of the mini project 3

11.5 Laboratory Sessions

The laboratory time allocation for this topic is 3 hours.

Task 4: Implementation: Prototype (10 marks)
In Task 4, students should focus on building a prototype of their project. This involves translating the design specifications into a functional prototype that demonstrates key features and functionalities of the system. The prototype should serve as a proof of concept and allow for testing and refinement before full-scale development.

In Task 4, students are not required to code the prototype. They can utilise simulation software like Sketch or Adobe XD to design interactive prototypes without writing code. Alternatively, for web-based projects, students can use website builders such as Wix to create functional prototypes without coding. Students could only focus on front-end development.

11.6 Private Study

The time allocation for private study in this topic is expected to be 8 hours.

Exercise 1:

Students should continue the Mini Project Task during their private study time which is Task 4 of the Mini Project brief.

Topic 12: Mini Project 4

12.1 Learning Objectives

On completion of the topic, students will be able to:

- Discuss the role of testing in SDLC
- Discuss the role of project closure and conclusion

12.2 Pedagogic Approach

Information will be transmitted to the students during the lectures. They will then practise the skills during the laboratory sessions and extend their understanding during private study time. The tutorial will then provide an opportunity to review the key ideas and obtain further guidance and support.

12.3 Timings

Lectures: 1 hours

Laboratory Sessions: 3 hours

Private Study: 8 hours

12.4 Lecture Notes

The following is an outline of the material to be covered during the lecture time. Please also refer to the slides.

The structure of this topic is as follows:

- Introduction to Mini project 4
- Introduction to Task 5 and 6 of mini project 4

12.5 Laboratory Sessions

The laboratory time allocation for this topic is 3 hours.

Task 5: Testing: Test Cases (10 marks)

In Task 5, students will develop test cases to evaluate the functionality and performance of their project. Test cases should cover various scenarios to ensure the system behaves as expected and meets the defined requirements.

Task 6: Conclusion (5 marks)

Task 6 involves summarising the project's key findings, lessons learned, and recommendations for future improvements. Students should reflect on the project's successes and challenges, drawing conclusions about its overall effectiveness and significance.

Task 7: Project Presentation and Demonstration (10 marks)

12.6 Private Study

The time allocation for private study in this topic is expected to be 8 hours.

Exercise 1:

Students should continue the Mini Project Task during their private study time, which is Task 5, Task 6 and Task 7 of the Mini Project brief.