YOOBEE

COLLEGE OF CREATIVE INNOVATION

Programme

Master of Software Engineering

Course

MSE800 Professional Software Engineering
(Level 8, 30 Credits)

Assignment 1

Object-Oriented Programming Assignment

(Classification - Individual)

Weighting within course:

30%

Assessment Tasks to Learning Outcome and GPOs mapping

Tasks	Learning Outcomes	GPOs
Task 1	LO1	GPO1
Task 2, Task 3	LO2	GPO4

Objective:

The objective of this assignment is to apply software engineering principles and practices to develop a functional Car Rental System. You will utilize object-oriented programming concepts, design patterns, and appropriate software development methodologies to create a robust and user-friendly system.

Assessment instructions

- Ask your tutor if you need further explanation or if the instructions are not clear.
- The purpose of this assessment is to assess your knowledge. As part of your academic and professional integrity, you must work alone on this assessment. In the event Yoobee suspects collusion, this will be addressed. For more information on plagiarism, please refer to the Student Handbook.
- Submit your completed assessment online (Blackboard) in the correct space provided.
- Percentage and feedback will be returned within 15 days of the submission date.

Learning Outcomes

LO1: Apply advanced software engineering skills in the context of architectural styles, testing procedures, and the software development lifecycle.

LO2: Analyse how diverse software engineering applications can produce innovative solutions to meet specific industry requirements.

Graduate Profile Outcomes (GPOs) covered

GPO1: Develop advanced software engineering knowledge and skills and apply these to solve emerging or existing problems.

GPO4: Critically analyse, assess and solve software-related problems using project management tools and techniques, creative thinking and enterprise skills.

Success Criteria:

You need to meet all the requirements of each of the learning outcomes and receive 50% or more to pass this assessment. You are allowed a maximum of three attempts. To meet all the requirements of each of the learning outcomes, you must achieve PASS results for each task item.

Grading:

The final grade will be determined by the score achieved in this assessment based on the following table. Should a second or third attempt be required the maximum contribution toward the overall mark for the tasks that required a second or third assessment attempt is 50%. A late submission is considered a second attempt, so the contribution will be capped at 50%.

Grade	Mark Band Range
A+	Meet all course requirements, mark range (90-100)
A	Meet all course requirements, mark range (85-89)
A-	Meet all course requirements, mark range (80-84)
B+	Meet all course requirements, mark range (75-79)
B	Meet all course requirements, mark range (70-74)
B-	Meet all course requirements, mark range (65-69)
C+	Meet all course requirements, mark range (60-64)
C	Meet all course requirements, mark range (55-59)
C-	Meet all course requirements, mark range (50-54)
D	Did not meet all course requirements, mark range (40-49)
E	Did not meet all course requirements, mark range (0-39)

Submission requirements:

Upload the following to Blackboard in the correct space provided.

- **Source Code Folder:** The folder containing source code along with other necessary project files or a text file containing link to your source code along with other necessary project files.
- **Release Build Zip:** Zip the release build executable.
- **ReadMe:** A text/pdf file containing details of User Documentation.
- **Design and Architecture:** A pdf file containing UML diagrams.
- **Maintenance and Support:** A pdf file containing strategies for managing maintenance, versioning, and backward compatibility.

Background:

You have been hired as a software engineer by a car rental company that wants to automate its rental process. The company currently relies on manual paperwork, which is time-consuming and prone to errors. Your task is to design and implement a Car Rental System that will streamline the rental process, improve efficiency, and enhance customer satisfaction. Requirements of the Rental System are the following:

Requirements:

User Management:

a. Implement user registration and login functionality.

b. Differentiate between customer and admin roles, each with specific privileges.

Car Management:

c. Create a database of available cars, including their details (ID, make, model, year, mileage, available now, minimum rent period, maximum rent period.)

d. Allow admins to add, update, and delete car records.

Rental Booking:

e. Enable customers to view available cars and their details.

f. Implement a booking feature that allows customers to select a car, specify rental dates, and provide necessary details.

g. Calculate the rental fees based on the selected car, rental duration, and any additional charges.

Rental Management:

h. Allow admins to manage rental bookings, including approving or rejecting requests.

Assignment Tasks:

Task 1: Design and Architecture (LO1)

Design an object-oriented architecture for the car rental system. Consider using appropriate design patterns (such as Factory Method, Singleton, Observer) to enhance modularity and maintainability. Provide a high-level architectural diagram and a description of how each component interacts.

Task 2: Innovative Solutions (LO2)

Propose an innovative feature or enhancement that sets your car rental system apart from traditional systems. This could involve integration with modern technologies (such as mobile apps, IoT devices, or cloud services) or introducing a unique user experience. Explain how this innovation addresses a specific industry requirement and provides a competitive advantage.

Task 3: Software Evolution (LO2)

Outline a plan for evolving the car rental system over time. Describe how you would handle updates, bug fixes, and new feature additions. Discuss strategies for managing software maintenance, versioning, and backward compatibility. Highlight the importance of software engineering principles in ensuring the system's long-term success.

Marking Rubric:

Your assignment will be evaluated based on the following criteria:

Criterion & Weighting	A (80-100) %	B (65-79) %	C (50-64) %	D (40-49) %	E (0-39) %
Task 1: Design and Architecture Weighting: 30%					
User Documentation (10%)	Provide clear and well-structured ReadMe file that describe the Car Rental System to Users and Programmers. ReadMe file includes **ALL** of the following: • A step-by-step guidance on how to configure, install and operate the Car Rental System • All relevant files are included. Also explain the purpose of each file in the context of the system • The ReadMe file clearly state the licensing terms under which the Car Rental System is released. • Any known bugs or issues with the system should be documented. • Credit section should include your details as developer of the Car Rental System	Provide clear and well-structured ReadMe file that describe the Car Rental System to Users and Programmers. ReadMe file include the following: • A step-by-step guidance on how to configure, install and operate the Car Rental System • All relevant files are included. Also explain the purpose of each file in the context of the system • The ReadMe file clearly state the licensing terms under which the Car Rental System is released.	Provide clear and well-structured ReadMe file that describe the Car Rental System to Users and Programmers. ReadMe file include the following: • A step-by-step guidance on how to configure, install and operate the Car Rental System • All relevant files are included. Also explain the purpose of each file in the context of the system	Provide a ReadMe file that does not clearly describe the Car Rental System to Users and Programmers.	The provided ReadMe file does not clearly describe the Car Rental System to users and programmers, lacking essential details and clarity necessary for understanding and utilization.

System Documentation (20%)	To explain the design and architecture of the system, **ALL** of the following UML diagrams are provided: • Class Diagram • Use Case Diagram • Sequence Diagram	To explain the design and architecture of the system, **any 2** of the following UML diagrams are provided: • Class Diagram • Use Case Diagram • Sequence Diagram	To explain the design and architecture of the system, any 1 of the following UML diagrams are provided: • Class Diagram • Use Case Diagram • Sequence Diagram	Only one of the following UML diagrams is provided to explain the design and architecture of the system: Class Diagram, Use Case Diagram, or Sequence Diagram, but it is incomplete or unclear, lacking the necessary detail to effectively communicate the system's design and architecture.	No UML diagram is provided to explain the design and architecture of the system, indicating a lack of visual representation to aid in understanding the system's structure and functionality.
Task 2: Innovative Solutions Weighting: 40%					
Car Rental System (30%)	• The system meets **ALL 4** specified requirements and provide a seamless car rental experience for customers **AND** admins. • **Include** an innovative feature or enhancement that sets your car rental system apart from traditional systems.	• The system meets **at least any 3 of the 4** specified requirements and provide a good car rental experience for customers **AND** admins. • **Does not include** any innovative feature or enhancement.	The system meets **at least any 2 out of the 4** specified requirements and provide a good car rental experience for **EITHER** customers **OR** admins.	The system meets **any 1** of the specified requirements and provide a poor car rental experience for EITHER customers OR admins.	The system fails to meet any of the specified requirements, providing a poor car rental experience for either customers or admins, or both.
Coding standard (10%)	Demonstrate **All** of the following guidelines and conventions: • Modularity and Encapsulation • Performance considerations • Commenting and Documentation • Indentation and Formatting • Naming Conventions	Demonstrate **any 3 to 4** of the following guidelines and conventions: • Modularity and Encapsulation • Performance considerations • Commenting and Documentation • Indentation and Formatting • Naming Conventions	Demonstrate **any 1 to 2** of the following guidelines and conventions: • Modularity and Encapsulation • Performance considerations • Commenting and Documentation • Indentation and Formatting • Naming Conventions	Demonstrate **NONE** of the following guidelines and conventions: • Modularity and Encapsulation • Performance considerations • Commenting and Documentation • Indentation and Formatting • Naming Conventions	The system demonstrates none of the following guidelines and conventions: modularity and encapsulation, performance considerations, commenting and documentation, indentation and formatting, and naming conventions. This indicates a lack of adherence to fundamental principles of software development and professionalism.

Task 3: Software Evolution Weighting: 30%					
Maintenance and Support	A plan is provided covering **ALL** of the following strategies for: • Managing software maintenance • Versioning • Backward compatibility	A plan is provided covering **any 2** of the following 3 strategies: • Managing software maintenance • Versioning • Backward compatibility	A plan is provided covering **any 1** of the following 3 strategies: • Managing software maintenance • Versioning • Backward compatibility	A plan is provided covering only one of the following strategies: managing software maintenance, versioning, or backward compatibility, but the plan is incomplete or lacks detail, indicating a limited understanding of software management principles.	No plan is provided covering any of the specified strategies: managing software maintenance, versioning, or backward compatibility, indicating a lack of consideration for essential aspects of software development and maintenance.

Note: *The ranges for each grade level encompass the full 11-point grading system as outlined in the accompanying table. Please refer to the table for detailed percentage ranges associated with each letter grade.*