

Assignment 2 (Team-based) ▲▼

Assignment 2: Team-based

Assignment name: Assignment 2 (Team-based)

Course code: ISYS3413/ISYS3475/ISYS1118

Weighting: 25%

Due date: Week 8 (21 September 2025, 11:59 pm)

1. Course Learning Outcomes Assessed

This assessment supports the following learning outcomes:

- CLO 2: evaluate requirements for a software system
- CLO 3: apply the process of analysis and design using the object-oriented approach.
- CLO 4: work effectively in a team to analyse the requirements of a complex software system, and solve problems by creating appropriate designs that satisfy these requirements
- CLO 5: communicate effectively with others, especially regarding the progress of the system development and the content of the design by means of reports and presentations. Use appropriate design, version control and collaboration tools to work effectively as a team.
- CLO 6: recognise and describe current trends in the area of software engineering.

2. Overview of Assessment

This assignment evaluates object-oriented structural and behavioural design in terms of use case diagrams, use case descriptions, and class diagrams. Based on the requirements collected for the **WorldWanderer** website in Assessment 1, you have to develop a use case diagram, use case descriptions, and a class diagram for the project. You also need to write some basic (skeleton) codes for the designed class diagrams.

3. Assessment Activities

General Description

Based on the requirements collected for the **WorldWanderer** website in Assessment 1, you need to do the following activities for the **WorldWanderer** platform.

IMPORTANT:

- Each team member must take responsibility for designing + implementing **one of the major functionalities** of the system - either Flight Comparison, Accommodation Comparison OR Vehicle Hire. The remaining functionality must be distributed **equally** amongst team members (Unless you are a group of 4, in which case the remaining functionality is allocated to your fourth member). Your diagrams must be **colour coded (and include a legend)** to clearly identify which section was completed by each member.
- You must complete work on the same major functionality throughout each of the activities below.

Activities in Assessment 2

Activity 1: Use Case Diagram (7 points): Each team is expected to create **one use case diagram** for the software project. The use case diagram should include a wide range of standard use case diagram notations to get all marks (**See the rubric for more information**). Please note:

- Child actors and child use cases are counted.

Activity 2: Use Case Descriptions (9 points): Each team member is expected to write down the use case description for **one of the important use cases** captured in Activity 1 of Assessment 2 using the template provided in the class. Each use case description has **3 points**. [Click here to download the template](#). (**See the rubric for more information**). Please note:

- It is totally up to you to decide which use case you want to describe - but it must relate to your designated system functionality.
- If **Use Case A** has an 'inclusion', 'exclusion', or 'generalization' relationship with other use cases and you decide to describe **Use Case A**, you **MUST** describe those use cases that have a relation with **Use Case A** (unless they relate to a Use Case being handled by another team member).
- If a use case has alternative flows, you must describe the alternative flows. The description of alternative flows will be counted.
- In Activity 2, the steps in IF-ELSE will be counted.

Activity 3: Class Diagram (7 points): Each team is expected to create **one class diagram** for the project. The class diagram should include a wide range of standard class diagram notations to receive all marks. (**See the rubric for more information**).
Note:

- Child classes (Subclasses) should have **ONE** specific attribute and **ONE** specific operation. As you know, they automatically inherit public/protected attributes and operations from their parent classes.
- Child classes are counted.
- The section of the Class diagram that each member completes, must relate to their designated system functionality from Activity 1 and 2. Other common classes should be split evenly amongst team members (Unless you are a group of 4, in which case the remaining functionality is allocated to your fourth member).

Activity 4: Code (2 points): Each team is expected to write the skeleton code of the class diagram designed in Activity 3 of Assessment 2 in Java. (**See the rubric for more information**). Note: There is no need to fully implement the methods. You must implement code related to your designated system functionality (from the class diagram)

4. Submission Instructions & Feedback

As this assignment is team-based, each team must submit a report that includes all activities described above. The team needs to upload the contribution form, along with other documents. The grade of each team member is determined based on their level of contribution. It is expected each team member equally contributes to this team-based assignment. [Click here to download the Statement of Contributions form.](#)


Submission Type: A ZIP file includes the following items

1. A Word/PDF document includes the diagrams for Activity 1 and Activity 3, and your answer to Activity 2.
 - **Note:** Once you create **diagrams** for Activity 1 and Activity 3 using UML tools, you need to copy and paste the **high-resolution images of the diagrams** into a Word/PDF document.
2. The code for Activity 4 can be simply copied into the Word/PDF document or can be submitted as a Java project along with other documents.
3. Completed contribution form
 - **Note:** It is acceptable to put your names in the signature part of the contribution statement form. There is **NO** need to do a physical or digital signature.

Report Length: The criteria are in the correctness and completeness of the artefacts in the report.

Late submission: Unless special consideration has been granted, the late penalty is 10% of the total mark ($25 \times 10/100 = 2.5$ marks) for the assessment per day late for up to 5 days (so the maximum late penalty is 50%). Submissions more than 5 days late are not accepted.

5. Required Software Tools

- For Activity 1 and Activity 3, you can use any **UML editors** such as Lucichart and Visual Paradigm. You **are not allowed** to use MS Word and MS PowerPoint to create UML diagrams.
- For Activity 2, you can use MS Word.
- For Activity 4, you can use any JAVA IDE such as Eclipse, IntelliJ, etc, (<https://www.eclipse.org/downloads/packages/> ). however, please make sure that your work can be opened in Eclipse.

6. Important Notes

- Please note that you can and should get ideas from the examples in lectures/tutorials/lectorials while you are working on your assignments. However, please note that you are not allowed to use (copy and paste) the same examples provided in lectures/tutorials/tutorials in the assignments, even if they can be applied to the **WorldWanderer** platform. For example, although some architectural decisions (e.g., selecting SQL Server or MySQL to store data) discussed in

lectures/tutorials/lectorials can be made for and applied to many systems, I strongly suggest you develop your own solutions for the assignments. Otherwise, you will lose some marks.

- You are NOT allowed to simply use (copy and paste) answers generated by AI-based tools such as ChatGPT for your assignments. I strongly suggest you develop your own solutions for the assignments. Otherwise, you will lose some marks.
- You need to work on Assignment 2 as part of a group. You are not allowed to submit Assignment 2 individually. The individual submission in Assignment 2 will **NOT be marked (Zero Mark)**.
- Extension requests should be made **one working day** before the deadline.
- **Extensions are for individuals, not per group.** In other words, if a member of a group gets an extension, the group should submit the assignment by the deadline and that member should submit the updated version of the assignment to me via email. I will check the differences between the original version and the updated version.

7. Assessment Criteria

Your report will be assessed on the following criteria:

Activity 1: Use Case Diagram (7 points): Each team is expected to create a use case diagram for the software project. The use case diagram should include a wide range of standard use case diagram notations to get all marks.

Rubric for use case diagram	Mark
The created use case diagram is meaningful, well-described, and in the scope of the project and uses a good number of each standard use case diagram notation (i.e., at least 6 diverse actors, 12 use cases, 4 diverse relationships including <<inclusion>>, <<extension>>, association and generalization relationships), without any mistakes.	7 points
The created use case diagram is to a large extent meaningful, well-described, and in the scope of the project and uses a reasonable number of each standard use case diagram notations (e.g., at least 5 diverse actors, 11 use cases, 3 diverse relationships including <<inclusion>>, <<extension>>, association and generalization relationships), with 1-3 minor mistakes.	4.6 to <7 points

The created use case diagram is to some extent meaningful, well-described, and in the scope of the project and uses a reasonable number of each standard use case diagram notation (e.g., at least 4 diverse actors, at least 8 use cases, 2 diverse relationships including <<inclusion>>, <<extension>>, association and generalization relationships), with 1-3 mistakes.	2.3 to <4.6 points
The created use case diagram is to some extent meaningful, well-described, and in the scope of the project and uses only a few numbers of each standard use case diagram notation (e.g., actors, use cases, diverse relationships including <<inclusion>>, <<extension>>, association and generalization relationships), with more than 3 mistakes.	>0 to <2.3 points
The student team does not submit any use case diagram or a student submits the assignment individually (not as part of a group).	0 point
The maximum mark for use case diagram	7 points
Total mark for Activity 1 (one use case diagram)	1 * 7 = 7 points

Activity 2: Use Case Descriptions (9 points): Each team is expected to write down the use case description of three of the most important use cases captured in Activity 1 of Assessment 2 using the template provided in the class. Each use case description has 3 points. [Click here to download the template](#)

Rubric for each Use Case Description	Mark
The created use case description is meaningful, well-described, and in the scope of the project, and fully follows the template, without any mistakes. The use case description should include at least six activities/steps in its main flow and one of the following items : one secondary actor, one precondition, one postcondition, or one alternative flow.	3 points
The created use case description is to a large extent meaningful, well-described, and in the scope of the project, and to a large extent follows the template, with 1-3 minor mistakes. The use case description should include at least five activities/steps in its main flow and one of the following items : one secondary actor, one precondition, one postcondition, or one alternative flow.	2 to <3 points

The created use case description is to some extent meaningful, well-described, and in the scope of the project, and to some extent follows the template with 1-3 mistakes. The use case description should include at least four activities/steps in its main flow.	1 to <2 points
The created use case description is to some extent meaningful, well-described, and in the scope of the project, and to some extent follows the template with more than 3 mistakes.	>0 to <1 points
The student does not submit any use case description or a student submits the assignment individually (not as part of a group).	0 points
The maximum mark for each use case description	3
Total mark for Activity 2 (three use case descriptions)	3 * 3 = 9 points

Activity 3: Class Diagram (7 points): Each team is expected to create a class diagram for the project. The class diagram should include a wide range of standard class diagram notations to receive all marks.

Rubric for Class Diagram	Mark
The created class diagram is meaningful, well-described, and in the scope of the project and uses a good number of each standard use class diagram notations. The class diagram should include at least 12 classes, each of the classes should have cardinality, and the diagram should include 4 types of relationships, including association, generalization, aggregation, and composition relationships, without any mistakes. Each class should include 2 attributes and two operations.	7 points
The created class diagram is to a large extent meaningful, well-described, and in the scope of the project and uses a reasonable number of each standard class diagram notations. The class diagram should include at least 11 classes, each of the classes should have cardinality, and the diagram should include 3 types of relationships, including association, generalization, aggregation, and composition relationships, with 3 minor mistakes. Each class should include 2 attributes and two operations	4.6 to <7 points
The created class diagram is to some extent meaningful, well-described, and in the scope of the project and uses a reasonable number of each standard	2.3 to <4.6 points

class diagram notation. The class diagram should include at least 8 classes, each of the classes should have cardinality, and the diagram should include 2 types of relationships, including association, generalization, aggregation, and composition relationships, with 1-3 mistakes. Each class should include 2 attributes and two operations.	
The created class diagram is to some extent meaningful, well-described, and in the scope of the project and uses only a few numbers each standard class diagram notation (e.g., classes, cardinality, diverse relationships including association, generalization, aggregation, and composition relationships), with more than 3 mistakes. Each class should include 2 attributes and two operations.	>0 to <2.3 points
The student team does not submit any class diagram or a student submits the assignment individually (not as part of a group).	0 point
The maximum mark for class diagram	7 points
Total mark for Activity 3 (one class diagram)	1 * 7 = 7 points

Activity 4: Code (2 points): Each team is expected to write the skeleton code of the class diagram designed in Activity 3 of Assessment 2 in Java.

Rubric for Code	Mark
The developed code well aligns with the created class diagram and reflects all the classes and their relationships in the class diagram, without any mistakes.	2 points
The developed code to a large extent aligns with the created class diagram and reflects all the classes and their relationships in the class diagram, with 1-3 minor mistakes.	1.5 to <2 points
The developed code to some extent aligns with the created class diagram and reflects the majority of the classes and their relationships in the class diagram, with few mistakes.	1 to <1.5 points
The developed code to some extent aligns with the created class diagram and reflects only some of the classes and their relationships in the class diagram, with some mistakes.	>0 to <1 points

The student team does not submit any code for the created class diagram or a student submits the assignment individually (not as part of a group).	0 point
The maximum mark for code.	2 points
Total mark for Activity 4 (one code skeleton)	1 * 2 = 2 points

Criteria	Ratings					Pts
Activity 1: Use Case Diagram	7 pts Full Marks The created use case diagram is meaningful, well-described, and in the scope of the project and uses a good number of each standard use case diagram notation (i.e., at least 6 diverse actors, 12 use cases, 4 diverse relationships including <<inclusion>>, <<extension>>, association and generalization relationships), without any mistakes.	7 to >4.6 pts Partial Mark The created use case diagram is to a large extent meaningful, well-described, and in the scope of the project and uses a reasonable number of each standard use case diagram notations (e.g., at least 5 diverse actors, 11 use cases, 3 diverse relationships including <<inclusion>>, <<extension>>, association and generalization relationships), with 1-3 minor mistakes.	4.6 to >2.3 pts Partial Mark The created use case diagram is to some extent meaningful, well-described, and in the scope of the project and uses a reasonable number of each standard use case diagram notation (e.g., at least 4 diverse actors, at least 8 use cases, 2 diverse relationships including <<inclusion>>, <<extension>>, association and generalization relationships), with 1-3 mistakes.	2.3 to >0.0 pts Partial Mark The created use case diagram is to some extent meaningful, well-described, and in the scope of the project and uses only a few numbers of each standard use case diagram notation (e.g., actors, use cases, diverse relationships including <<inclusion>>, <<extension>>, association and generalization relationships), with more than 3 mistakes.	0 pts No Marks The student team does not submit any use case diagram or a student submits the assignment individually (not as part of a group).	7 pts
Activity 2.1: Use Case Description 1	3 pts Full Marks The created use case description is meaningful, well-described, and in the scope of the project, and fully follows the template, without any	3 to >2.0 pts Partial Mark The created use case description is to a large extent meaningful, well-described, and in the scope of the project, and to a large extent	2 to >1.0 pts Partial Mark The created use case description is to some extent meaningful, well-described, and in the scope of the project, and to some extent follows the	1 to >0.0 pts Partial Mark The created use case description is to some extent meaningful, well-described, and in the scope of the project, and to some extent follows the		3 pts

Criteria	Ratings					Pts
	mistakes. The use case description should include at least six activities/steps in its main flow.	follows the template, with 1-3 minor mistakes. The use case description should include at least five activities/steps in its main flow.	template with 1-3 mistakes. The use case description should include at least four activities/steps in its main flow.	template with more than 3 mistakes.	0 pts No Marks The student does not submit any use case description or a student submits the assignment individually (not as part of	
Activity 2.2: Use Case Description 2	3 pts Full Marks The created use case description is meaningful, well-described, and in the scope of the project, and fully follows the template, without any mistakes. The use case description should include at least six activities/steps in its main flow.	3 to >2.0 pts Partial Mark The created use case description is to a large extent meaningful, well-described, and in the scope of the project, and to a large extent follows the template, with 1-3 minor mistakes. The use case description should include at least five activities/steps in its main flow.	2 to >1.0 pts Partial Mark The created use case description is to some extent meaningful, well-described, and in the scope of the project, and to some extent follows the template with 1-3 mistakes. The use case description should include at least four activities/steps in its main flow.	1 to >0.0 pts Partial Mark The created use case description is to some extent meaningful, well-described, and in the scope of the project, and to some extent follows the template with more than 3 mistakes.	0 pts No Marks The student does not submit any use case description or a student submits the assignment individually (not as part of a group).	3 pts
Activity 2.3: Use Case Description 3	3 pts Full Marks The created use case description is meaningful, well-described, and in the scope of the project, and fully follows	3 to >2.0 pts Partial Mark The created use case description is to a large extent meaningful, well-described, and in the scope of the	2 to >1.0 pts Partial Mark The created use case description is to some extent meaningful, well-described, and in the scope of the project, and to	1 to >0.0 pts Partial Mark The created use case description is to some extent meaningful, well-described, and in the scope of the project, and to	0 pts No Marks The student does not submit any use case description or a student submits the assignment individually	3 pts

Criteria	Ratings					Pts
	the template, without any mistakes. The use case description should include at least six activities/steps in its main flow.	project, and to a large extent follows the template, with 1-3 minor mistakes. The use case description should include at least five activities/steps in its main flow.	some extent follows the template with 1-3 mistakes. The use case description should include at least four activities/steps in its main flow.	some extent follows the template with more than 3 mistakes.	(not as part of a group).	
Activity 3: Class Diagram	7 pts Full Marks The created class diagram is meaningful, well-described, and in the scope of the project and uses a good number of each standard use class diagram notations. The class diagram should include at least 12 classes, each of the classes should have cardinality, and the diagram should include 4 types of relationships, including association, generalization, aggregation, and composition relationships, without any mistakes.	7 to >4.6 pts Partial Mark The created class diagram is to a large extent meaningful, well-described, and in the scope of the project and uses a reasonable number of each standard class diagram notations. The class diagram should include at least 11 classes, each of the classes should have cardinality, and the diagram should include 3 types of relationships, including association, generalization, aggregation, and composition relationships,	4.6 to >2.3 pts Partial Mark The created class diagram is to some extent meaningful, well-described, and in the scope of the project and uses a reasonable number of each standard class diagram notation. The class diagram should include at least 8 classes, each of the classes should have cardinality, and the diagram should include 2 types of relationships, including association, generalization, aggregation, and composition relationships,	2.3 to >0.0 pts Partial Mark The created class diagram is to some extent meaningful, well-described, and in the scope of the project and uses only a few numbers each standard class diagram notation (e.g., classes, cardinality, diverse relationships including association, generalization, aggregation, and composition relationships), with more than 3 mistakes. Each class should include 2 attributes and two operations.	0 pts No Marks The student team does not submit any class diagram or a student submits the assignment individually (not as part of a group).	7 pts

Criteria	Ratings					Pts
	Each class should include 2 attributes and two operations.	with 3 minor mistakes. Each class should include 2 attributes and two operations.	with 1-3 mistakes. Each class should include 2 attributes and two operations.			
Activity 4: Code	2 pts Full Marks The developed code well aligns with the created class diagram and reflects all the classes and their relationships in the class diagram, without any mistakes.	2 to >1.5 pts Partial Mark The developed code to a large extent aligns with the created class diagram and reflects all the classes and their relationships in the class diagram, with 1-3 minor mistakes.	1.5 to >1.0 pts Partial Mark The developed code to some extent aligns with the created class diagram and reflects the majority of the classes and their relationships in the class diagram, with few mistakes.	1 to >0.0 pts Partial Mark The developed code to some extent aligns with the created class diagram and reflects only some of the classes and their relationships in the class diagram, with some mistakes.	0 pts No Marks The student team does not submit any code for the created class diagram or a student submits the assignment individually (not as part of a group).	2 pts
Late Deduction	0 pts Deduction		0 pts No Marks			0 pts
Total Points: 25						