Assignment 4 (Individual)

Assignment 4: Individual

Assignment name: Assignment 4 (Individual)

Course code: ISYS3413/ISYS3475/ISYS1118

Weighting: 25%

Due date: (19 October 2025, 11:59 pm)

1. Course Learning Outcomes Assessed

This assessment supports the following learning outcomes:

- CLO 1: explain how iterative software engineering processes can facilitate software development
- CLO 2: evaluate requirements for a software system
- 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.
- CLO 7: Design and carry out tests using various testing techniques and tools.

2. Overview of Assessment

This assignment will assess your ability in terms of writing quality unit tests, test cases, user stories, acceptance tests, and working with Git/GitHub. You will be asked to implement and test a small function in the **WorldWanderer** website. Then you need to move your source codes to GitHub. You also need to write **five user stories** for the

WorldWanderer website, along with their acceptance criteria, following the templates and structures taught in the lectures. Finally, you need to make a video of your Java program.

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** website.

Activities in Assessment 4

Activity 1: Git and Unit Test (15 points) + a Video of your program: In the **WorldWanderer** website, there is a class called **FlightSearch**. The **FlightSearch** class has a function called **"runFlightSearch"**. You are asked to test and implement this function. The **FlightSearch** class has the following methods and attributes. A copy can be downloaded **here** .

```
package flight;

public class FlightSearch {
   private String  departureDate;
   private String  departureAirportCode;
   private boolean emergencyRowSeating;
   private String  returnDate;
   private String  destinationAirportCode;
   private String  seatingClass;
   private int     adultPassengerCount;
   private int     childPassengerCount;
   private int     infantPassengerCount;

   public boolean runFlightSearch(String departureDate,    String departureAirportCode,   boolean emergencyRowSeating,
                                  String returnDate,       String destinationAirportCode, String seatingClass,
                                  int adultPassengerCount, int childPassengerCount,       int infantPassengerCount) {
      boolean valid = true;

      //TODO: Validate all the provided parameters.
      //if the search parameters meets the given conditions,
      //   the function should initialise all the class attributes and return true.
      //else
      //   the function should return false

      return valid;
   }
}
```

1. **runFlightSearch function**. This function will validate the flight search criteria to determine if the search conditions are valid. The following conditions should be considered when validating the search parameters. If the search information meets the following conditions, the **runFlightSearch** function should initialise the attributes in the class and return true. If the search information does not meet the following conditions,

the function should not initialise the values and should instead return false. Note that all string values are lowercase - when executing test cases all string input parameters should be lowercase (hence variations in case are not to be considered as part of your test cases).

Condition 1. The total number of passenger per search must be at least 1 and cannot exceed 9.

Condition 2. Children cannot be seated in emergency row seating or first class.

Condition 3. Infants cannot be seated in emergency row seating or business class.

Condition 4. All children (aged 2-11 years old) must be seated immediately next to at least one adult passenger (ie up to 2 children per adult). eg if the adult passenger count is 2, then up to 4 child passengers are allowed.

Condition 5. Each infant (<2 years old) must be seated on an accompanying adults lap (only one infant is allowed per adult)

Condition 6. The departure date cannot be in the past (this is based on the current date when runFlightSearch method is called).

Condition 7. All dates must be formatted in the format DD/MM/YYYY eg "23/11/2025" and must be validated to ensure that the combination is correct (eg "29/02/2026" would be invalid as 2026 is not a leap year). Ensure that **STRICT** date validation is applied.

Condition 8. All flights are two way only (ie include return flights) and the return date cannot be before departure date.

Condition 9. The seating class must be one of ("economy', "premium economy", "business", "first").

Condition 10. Only economy class flights have emergency row seating.

Condition 11. Only the following airports are available: "syd" (Sydney), "mel" (Melbourne), "lax" (Los Angeles), "cdg" (Paris), "del" (Delhi), "pvg" (Shanghai) and "doh" (Doha). eg for a flight from Melbourne to Shanghai, the departure airport code would be "mel" and the destination airport code would be "pvg". Furthermore, the departure airport and destination airport cannot be the same.

In this activity, each student is requested to do the following activities :

Activity 1.1 Develop and write down a reasonable number of test cases. The function should have at least 12 test cases - one for each condition, and one case where all

input is valid (**use this template Download**). Each test case should have two test data (except the case where all input is valid where you will need at least 4 test data to properly test different combinations). Your test cases should be testing for **boundary values**. You need to implement the unit tests for this function using JUnit (7 points).

Activity 1.2 Implement this function in Java using any Java IDEs (7 points).

Activity 1.3 Move the implemented function and the unit tests to a private GitHub repository (1 point). In the video, you need to show that you have submitted the code to GitHub. There is no need to show the process.

Activity 1.4 **Record your screen while running and testing your program**. In the recorded video, you need to explain the implementation of the function (30 seconds), the implementation of unit tests for the function (30 seconds, and run the unit tests (1 minute). The recorded video should also show that your program is in GitHub (30 seconds). The total recording should not be more than **4 minutes** and **100 MB**. The recorded video should have a **720p resolution**. You need to upload the recorded video to Canvas as part of your submission. **Alternatively**, once you created the video with SharePoint or MS Teams, you can insert **the link to the recorded video in a WORD/PDF file** and give **access to your tutor** to watch the recorded video. The WORD/PDF file should be uploaded as part of your submission to Canvas. The recorded video will be used by markers to mark the student. Recorded videos have **no mark**, but you will **lose significant marks** if you do not upload a recorded video.

Note 1. Your program should initialise the values in the class (if the search details are valid). There is no need to store the search details or actually execute them.

Note 2. You MUST write enough meaningful comments to explain your code in JUnit 5 and Java.

Note 3. You MUST develop your code (the implementation of functions and JUnit test code) in any Java IDEs and use JUnit 5 for tests.

Note 4. You just need to develop a Console-based program (application). There is no need to add Graphic User Interfaces or menu to your program (to interact with end-users).

Note 5. The code written above is just a placeholder. You are allowed to edit it and add new functions (e.g., validator methods for various fields etc.) and new attributes. You are completely FREE on how you want to implement and test these functions. The key point is that you need to write enough test cases to test the conditions.

Note 6. You only need to write test cases to test the conditions that are explicitly mentioned above. There is no need to test other conditions.

Activity 2: User Story and Acceptance Criteria (10 points): Based on the requirements collected for the **WorldWanderer** website, each student is asked to write **five** user stories and **three** acceptance criteria for each user story. Each user story and its acceptance criteria have **2 points**. The user stories and acceptance criteria should be written based on the template and principles taught in class.

4. Submission Instructions & Feedback

As this assignment is individual, each student needs to submit a report that includes all activities described above.

Submission Type: A Zip file includes the following items:

1. A written report that contains (a) Test Cases developed for testing the function and (b) User Stories and Acceptance Criteria of the User Stories.

2. The whole Java project that includes the implementation of the specified function and Unit Tests

3. A recorded video of your Java project (4 minutes, 720p resolution, 100 MB)

4. A WORD/PDF file that includes a link to the recorded video (give access to ONLY your tutor to watch this video) and a link to your GitHub Repository (ensure that you add your timetabled tutor to your GitHub repository by going to Settings > Collaborators under the Access section in the left-hand menu in Github).

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 x 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, You can develop your code (the implementation of functions and JUnit test code) in any Java IDEs such as **Eclipse** and **IntelliJ** and use **JUnit for tests.** However, projects must be able to be opened in Eclipse for testing/review by markers.
- For Activity 2, you can use MS Word.

6. Important Notes

- Please note that you can and should get ideas from the examples in lectures/tutorials/lectorials/assignment description (e.g., conditions) 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/assignment description in the assignments, even if they can be applied to the **WorldWanderer** website. 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.
- Extension requests should be made **one working day** before the deadline.

7. Assessment Criteria

Your report will be assessed on the following criteria:

Activity 1: Git and Unit Test (15 points): In this activity, each student is requested to implement and test the **runFlightSearch** functions in the **FlightSearch** class in the **WorldWanderer** website. More specifically, in this activity, each student is requested to do the following activities:

Activity 1.1 Develop and write down a reasonable number of test cases. The function should have at least 12 test cases. Each test case should have two boundary test data - except the all valid case which has four. In other words, the function should have at least 28 test data. You need to implement the unit tests for this functions using JUnit (7 points).

Rubric for Unit Test and Test Cases for each function	**Mark**

The Unit tests developed for the given function are correct, and a reasonable number of test cases (at least 12 test cases and each test case should have two boundary test data - except the all valid case which has four) are correctly written, without any mistakes. The written codes have enough meaningful comments.	7 points
The Unit tests developed for the given function are to a large extent correct, and the majority of test cases (at least 10 test cases and each test case should have two data) are correctly written, with 1-3 minor mistakes. The written codes have to a large extent meaningful comments.	5 to <7 points
The Unit tests developed for the given function are to some extent correct, and at least half of the test cases are correctly written, with 1-3 mistakes. The written codes have to some extent meaningful comments.	3 to <5 points
The Unit tests developed for the given function are to some extent correct, and less than 50% of the test cases are correctly written, with more than three mistakes. The written codes have to some extent meaningful comments.	>0 to <3 points
The student does not submit any test code and test cases for the given function.	0 point
The maximum mark for Unit Test and Test Cases for the function	7 points

Activity 1.2 Implement the runFlightSearch function in Java using any Java IDEs (7 points)

Rubric for each function implementation	**Mark**
The codes developed for the given function are correct without any mistakes. The written codes have enough meaningful comments.	7 points
The codes developed for the given function are to a large extent correct with 1-3 minor mistakes. The written codes have enough to a large extent meaningful comments.	5 to <7 points
The codes developed for the given function are to some extent correct with 1-3 mistakes. The written codes have enough to some extent meaningful comments.	3 to <5 points

The codes developed for the given function are to some extent correct with more than 3 mistakes. The written codes have enough to some extent meaningful comments.	>0 to <3 points
The student does not submit any code for the given function.	0 point
The maximum mark for the given function implementation.	3.5 points

Activity 1.3 Create a private GitHub account, move the implemented function and it's unit tests to the private GitHub repository, and explain it in the recorded video **(1 point)**.

Rubric for GitHub	**Mark**
The codes of the functions and unit tests are successfully moved to the GitHub repository, without any mistakes.	1 point
The codes of the functions and unit tests are successfully moved to the GitHub repository, with 1-3 minor mistakes.	0.75 to <1 point
The codes of the functions and unit tests are successfully moved to the GitHub repository, with 1-3 mistakes.	.5 to <.75 points
The codes of the functions and unit tests are successfully moved to the GitHub repository, with more than 3 mistakes.	>0 to <.5 points
The student does not submit any code to GitHub.	0 point
The maximum mark for Unit Test, Test Cases, GitHub.	1 point

Activity 1.4: A recorded video that explains your Java program, Unit Tests, and the private repository: The recorded video will be used by markers for marking the student. Recorded videos have no mark, but you will lose significant marks if you do not provide a recorded video.

Activity 2: User Story and Acceptance Criteria (10 points): Based on the requirements collected for the **WorldWanderer** website, each student is asked to write **five** user stories and **three** acceptance criteria for each user story. Each user story and

its acceptance criteria have **2 points**. The user stories and acceptance criteria should be written based on the template and principles taught in class.

Rubric for each user story and its acceptance criteria	**Mark**
The developed user story and its acceptance criteria are meaningful, well-described, and in the scope of the project and fully follow the template and principles, without any mistakes.	2 points
The developed user story and its acceptance criteria are to a large extent meaningful, well-described, and in the scope of the project and to a large extent follow the template and principles, with 1-3 minor mistakes.	1.5 to <2 points
The developed user story and its acceptance criteria are to some extent meaningful, well-described, and in the scope of the project and to some extent follow the template and principles, with 1-3 mistakes.	1 to <1.5 points
The developed use story and its acceptance criteria are to some extent meaningful, well-described, and in the scope of the project and to some extent follow the template and principles, with more than 3 mistakes.	>0 to <1 points
The student does not submit any user story and its acceptance criteria.	0 point
The maximum mark for each user story and its acceptance criteria.	2 points
Total mark for **Activity 2** (five user stories and their acceptance criteria)	5 * 2 = 10 points

Criteria	Ratings					Pts
Activity 1.1.1 (Unit Test and Test Cases for runflightSearch function)	**7 pts** **Full Marks** The Unit tests developed for the given function are correct, and a reasonable number of test cases (at least 12 test cases and each test case should have two boundary test data - except the all valid case which has four) are correctly written, without any mistakes. The written codes have enough meaningful comments.	**7 to >5.0 pts** **Partial Mark** The Unit tests developed for the given function are to a large extent correct, and the majority of test cases (at least 10 test cases and each test case should have two data) are correctly written, with 1-3 minor mistakes. The written codes have to a large extent meaningful comments.	**5 to >3.0 pts** **Partial Mark** The Unit tests developed for the given function are to some extent correct, and at least half of the test cases are correctly written, with 1-3 mistakes. The written codes have to some extent meaningful comments.	**3 to >0.0 pts** **Partial Mark** The Unit tests developed for the given function are to some extent correct, and less than 50% of the test cases are correctly written, with more than three mistakes. The written codes have to some extent meaningful comments.	**0 pts** **No Marks** The student does not submit any test code and test cases for the given function.	7 pts
Activity 1.2.1 (runflightSearch function implementation in Java)	**7 pts** **Full Marks** The codes developed for the given function are correct without any mistakes. The written codes have enough meaningful comments.		**5 to >3.0 pts** **Partial Mark** The codes developed for the given function are to some extent correct with 1-3 mistakes. The written codes have enough to some extent meaningful comments.		**0 pts** **No Marks** The student does not submit any code for the given function.	7 pts

Criteria	Ratings					Pts
		7 to >5.0 pts **Partial Mark** The codes developed for the given function are to a large extent correct with 1-3 minor mistakes. The written codes have enough to a large extent meaningful comments.		**3 to >0.0 pts** **Partial Mark** The codes developed for the given function are to some extent correct with more than 3 mistakes. The written codes have enough to some extent meaningful comments.		
Activity 1.3	**1 pts** **Full Marks** The codes of the function and unit tests are successfully moved to the GitHub repository, without any mistakes.	**1 to >0.75 pts** **Partial Mark** The codes of the function and unit tests are successfully moved to the GitHub repository, with 1-3 minor mistakes.	**0.75 to >0.5 pts** **Partial Mark** The codes of the function and unit tests are successfully moved to the GitHub repository, with 1-3 mistakes.	**0.5 to >0.0 pts** **Partial Mark** The codes of the function and unit tests are successfully moved to the GitHub repository, with more than 3 mistakes.	**0 pts** **No Marks** The student does not submit any code to GitHub.	1 pts
Activity 2.1 (User Story 1 and its three acceptance criteria)	**2 pts** **Full Marks** The developed user story and its acceptance criteria are meaningful, well-described, and in the scope of the project and fully follow the template and principles, without any mistakes.	**2 to >1.5 pts** **Partial Mark** The developed user story and its acceptance criteria are to a large extent meaningful, well-described, and in the scope of the project and to a large extent follow the template and principles, with 1-3 minor mistakes.	**1.5 to >1.0 pts** **Partial Mark** The developed user story and its acceptance criteria are to some extent meaningful, well-described, and in the scope of the project and to some extent follow the template and principles, with 1-3 mistakes.	**1 to >0.0 pts** **Partial Mark** The developed use story and its acceptance criteria are to some extent meaningful, well-described, and in the scope of the project and to some extent follow the template and principles, with more than 3 mistakes.	**0 pts** **No Marks** The student does not submit any user story and its acceptance criteria.	2 pts
Activity 2.2 (User Story 2 and its three acceptance criteria)	**2 pts** **Full Marks** The developed user story and its acceptance criteria are meaningful,	**2 to >1.5 pts** **Partial Mark** The developed user story and its acceptance criteria are to a large extent	**1.5 to >1.0 pts** **Partial Mark** The developed user story and its acceptance criteria are to some extent meaningful, well-	**1 to >0.0 pts** **Partial Mark** The developed use story and its acceptance criteria are to some extent	**0 pts** **No Marks** The student does not submit any user story and its	2 pts

Criteria	Ratings					Pts
	well-described, and in the scope of the project and fully follow the template and principles, without any mistakes.	meaningful, well-described, and in the scope of the project and to a large extent follow the template and principles, with 1-3 minor mistakes.	described, and in the scope of the project and to some extent follow the template and principles, with 1-3 mistakes.	meaningful, well-described, and in the scope of the project and to some extent follow the template and principles, with more than 3 mistakes.	acceptance criteria.	
Activity 2.3 (User Story 3 and its three acceptance criteria)	**2 pts Full Marks** The developed user story and its acceptance criteria are meaningful, well-described, and in the scope of the project and fully follow the template and principles, without any mistakes.	**2 to >1.5 pts Partial Mark** The developed user story and its acceptance criteria are to a large extent meaningful, well-described, and in the scope of the project and to a large extent follow the template and principles, with 1-3 minor mistakes.	**1.5 to >1.0 pts Partial Mark** The developed user story and its acceptance criteria are to some extent meaningful, well-described, and in the scope of the project and to some extent follow the template and principles, with 1-3 mistakes.	**1 to >0.0 pts Partial Mark** The developed use story and its acceptance criteria are to some extent meaningful, well-described, and in the scope of the project and to some extent follow the template and principles, with more than 3 mistakes.	**0 pts No Marks** The student does not submit any user story and its acceptance criteria.	2 pts
Activity 2.4 (User Story 4 and its three acceptance criteria)	**2 pts Full Marks** The developed user story and its acceptance criteria are meaningful, well-described,	**2 to >1.5 pts Partial Mark** The developed user story and its acceptance criteria are to a large extent meaningful, well-	**1.5 to >1.0 pts Partial Mark** The developed user story and its acceptance criteria are to some extent meaningful, well-described, and in the scope of	**1 to >0.0 pts Partial Mark** The developed use story and its acceptance criteria are to some extent meaningful, well-	**0 pts No Marks** The student does not submit any user story and its acceptance criteria.	2 pts

Criteria	Ratings					Pts
	and in the scope of the project and fully follow the template and principles, without any mistakes.	described, and in the scope of the project and to a large extent follow the template and principles, with 1-3 minor mistakes.	the project and to some extent follow the template and principles, with 1-3 mistakes.	described, and in the scope of the project and to some extent follow the template and principles, with more than 3 mistakes.		
Activity 2.5 (User Story 5 and its three acceptance criteria)	**2 pts** **Full Marks** The developed user story and its acceptance criteria are meaningful, well-described, and in the scope of the project and fully follow the template and principles, without any mistakes.	**2 to >1.5 pts** **Partial Mark** The developed user story and its acceptance criteria are to a large extent meaningful, well-described, and in the scope of the project and to a large extent follow the template and principles, with 1-3 minor mistakes.	**1.5 to >1.0 pts** **Partial Mark** The developed user story and its acceptance criteria are to some extent meaningful, well-described, and in the scope of the project and to some extent follow the template and principles, with 1-3 mistakes.	**1 to >0.0 pts** **Partial Mark** The developed use story and its acceptance criteria are to some extent meaningful, well-described, and in the scope of the project and to some extent follow the template and principles, with more than 3 mistakes.	**0 pts** **No Marks** The student does not submit any user story and its acceptance criteria.	2 pts
Deduction	**0 pts** **Deduction**		**0 pts** **No Marks**			0 pts
						Total Points: 25