

**UECS2354 SOFTWARE TESTING
ASSIGNMENT – 202201**

Project details:

DeliveryExpert is a courier service company that provides items pickup and delivery services from and to destinations within Selangor and Kuala Lumpur areas. You are to develop and test a system to be used by *DeliveryExpert* to calculate the delivery charges and to print the delivery note.

The delivery charges are as shown in the following table:

Item Type	Weight (grams)	Distance (km)	Charge (RM)
Document	< 300	–	3.00
	300 – 1000	< 10	4.00
		10 – 30	5.00
		> 30	6.00
	1001 – 3000	< 10	6.00
		10 – 30	8.00
		> 30	10.00
	3001 – 5000	< 10	12.00
		10 – 30	18.00
		> 30	25.00
	> 5000	< 10	25.00
		10 – 30	30.00
		> 30	35.00
Parcel	<1000	< 10	5.00
		10 – 30	8.00
		> 30	10.00
	1001 – 2000	< 10	15.00
		10 – 30	18.00
		> 30	25.00
	2001 – 3000	< 10	23.00
		10 – 30	28.00
		> 30	35.00
	3001 – 4000	< 10	35.00
		10 – 30	40.00
		> 30	50.00
	>5000	< 10	45.00
		10 – 30	50.00
		> 30	60.00

Additional charges:

- Same day delivery: additional RM10.00
- With insurance: additional RM15.00

Clients call up to determine whether the pickup and delivery addresses are within the delivery area that the company operates in based on postcode. Once the client accepts the charge, the client details are entered into the system unless it is an existing client, in which case, the details are retrieved from the system. Next the date is entered into the system. Once all details are entered, the service staff assigns one of the company's delivery staff members to pick up and deliver the items. The system will display the delivery note with the details of the delivery staff, item's weight, distance, and total charges. A copy of the delivery note will be emailed to user.

Assumption:

The distance is being calculated from a website and *DeliveryExpert* only enters the total distance into the system to calculate the total charges.

UECS2354 SOFTWARE TESTING
ASSIGNMENT - 202201

Assignment Deliverables:

Part A: Test Plan

Based on the project details and requirement, plan for the testing activities. Your test plan must contain scope, objective, test basis, features to be tested and not to be tested, test conditions, test entry and exit criteria. Refer to the test plan template.

Part B: Test Design

Based on the project details, create decision table, and design the test cases for the testing phase. There must be clear description in every test case. Refer to the test case template.

After analyse the requirement and clearly state the test objectives and test scope, create the decision table based on the condition stated in the test plan. You can use excel to create your decision table. After identify the rules for each conditions, design the test cases accordingly. Put the rule identifier as reference in your test cases.

For ex: if you have 2 rules for a condition #1. You can give any identifiers for the rules (*Rule #1* and *Rule #2*). In the test case template, design test case for each of this rules.

Rule#1:

Test Case#	Test Title	Test Summary	Test Steps	Test Data	Expected Result
Test case 001	To verify rule#1	if condition #1 is true, the action will be action#1	[list down the steps]	Ex: Item = document weight = 600 distance = 35	Action #1 & total charges = RMxx.xx
Test case 002	To verify rule#2	if condition #1 is false, the action will be action#1			Should return error.

****This is just an example, it's depends on how you design your test cases and it will be unique for each individual testers.**

Part C: Application code and Test Code

Jar files: Place all JAR files (jUnitParams & Mockito) in **C:\jars**

1. Application code:

You are responsible to complete and implement the classes in the application. It is depending on how you want to design your classes. However, the stated core functionalities should be working as per expected.

Note: The 'Print Delivery Note' and 'Email Notification' classes are not ready for testing. However, you can use stub and driver to test the integration within this module.

2. Test code:

Write test code using jUnit Framework and Mockito to test the application code.

**UECS2354 SOFTWARE TESTING
ASSIGNMENT – 202201**

Further Information:

The particular testing issues that you need to pay attention to and their associated marks are shown in the marking scheme (*UECS2354_Marking.docx*). Include comments in test code to clearly specify which aspect of these testing issues you are addressing. For example, when creating parameterised tests, include comments to state the approach that you are using (e.g., boundary value analysis, etc.).

The focus is NOT to write as many tests as possible, but rather to create the right number of tests necessary to verify the functionality of the methods. For example, if you are using equivalence partitioning, writing tests that use all the inputs within the same partition that produce identical output from a method is a waste of space.

You may give additional assumptions for your application, state them clearly in the report. To make your program more robust and avoid problems at run time, do as much status/error checking as you could in your program. And, good organization of the code and meaningful variable names would help readability, and liberal use of comments can help the marker understands what the program does and why. In addition, provide class diagram(s) to illustrate the design of your program.

Submission details:

Due date:

13 April 2022 (Wednesday – W12)

This is a group assignment. Form a group of 4 members in a group. Register your group member's name in the google sheet:

https://docs.google.com/spreadsheets/d/1UAXR_HKBFq_4DYKAFCI_BT2ovWbGsQ9smjDMGGmlI4U/edit?usp=sharing

Submit the following items through WBLE (link will be created in WBLE course page).

1. Java project folder:

Archive of Eclipse **project folder** that contains source code for both the application code and the test code. The application code and test code should be placed in two separate source directories.

Note: Do not include jar files as it may be blocked.

2. Report:

Your report should contain the following:

- Marking sheet (.doc) as the front page of the documentation
- Test plan (Part A)
- Test cases design (Part B)
- Assumptions (if any)
- Class diagram for the application code (*If you update, add, remove any details for the modules given above, reflect it in the class diagram*)
- The application code and jUnit test code, should be attached at the end of the documentation.

**UECS2354 SOFTWARE TESTING
ASSIGNMENT - 202201**

Total mark

The total mark of this practical assignment is 100. The 100 marks will contribute 20% of your final mark. It's your responsibility to understand the requirements of the tasks and prepare well for your submission. You might be asked questions about the works you submit to ensure that you understand them.

Late Submission

No late submission of assignment is allowed. Assignment received after the due date without valid reasons will be penalized using the following policy: 5 marks will be deducted for every day the assignment is overdue.

Plagiarism

It is important that your solutions to the practical assignment be your own work. It is perfectly acceptable to seek help and advice when completing the practical assignment, but this must not be taken to the point where what is submitted is in part someone else's work. Any group found to have committed plagiarism or cheating by copying program codes from other sources will be given a failing grade.