UECS2354 SOFTWARE TESTING
ASSIGNMENT
JAN 2021

MARKING REPORT

Group number: 20

Name	Student ID
1. Gervin Fung Da Xuen	1801655
2. Tan Ching Erl	1805252
3. Tan Kai Fung	1806915
4. Tieu Zi Yee	1806886

Marks breakdown

Part A: Test Plan (10 marks)

Component	Max Mark	Marks Obtained	Remark/Comment
Test objective, scope, deliverable	5		
Test basis, condition, entry, exit criteria	5		

Part B: Test Design (20 marks)

Component	Max Mark	Marks Obtained	Remark/Comment
Decision table	5		
Appropriateness of test cases	15		

Part C: Java Program (application code and test code) (70 marks)

Component	**Max Mark**	**Marks Obtained**	**Remark/Comment**
Setup jar file location to C:\jars	2		
Source directories	3		
Appropriate used of assertsXXX methods.	10		
Using parameterised tests correctly	10		
Invalid values are checked for in implemented code, and tests for invalid values are performed.	10		
Use of mocks or stubs for testing.	10		
Combining test cases into test suites	5		
Setting up some tests so that test values are read from a text file instead of hardcoding into test coe	10		
Perform integration testing after unit tests have been completed	10		

A:	**B:**	**C:**	**Total:** **/100**	**/20**

Part A: Test Plan

1.0 Introduction

1.1 Scope

1.1.1 Item Module

In this module, each item will be tested to ***ensure*** that **it's an unique object** and ***will not*** result in a **hash collision** with another item object. The function is also tested to ensure that **promotional items have indications** that they are on discount. The function will also test to ensure that **items with item type cake have indications that they are cake**. Lastly, the function will also test to make sure that all item's member price is less than non-member price

1.1.2 Customer Module

In this module, **Area** and **District** input will also be tested to ensure that **Area** members entered are ***among the 20 areas*** of Melaka, and **District** is ***among the 3 districts*** of Melaka.

For input that only accepts numbers, testing will be done to ensure that **only numeric input** is allowed. For input that requires alphabets and space only, testing will be done to ensure that input with **alphabets with no numbers or white space or any symbols** is accepted.

1.1.3 Member Module

In this module, **member login** will be tested. **Contact** input will also be tested to ensure that it obeys the ***format of contact number*** dictated by Malaysia

1.1.4 Order Module

In this module, tests will be conducted to ensure that there are only 20 delivery areas and that 20 delivery rates are greater than 0. Also ***various calculation*** functions will be tested. The **calculation for delivery charge function** is tested to ensure that correct delivery rates are charged **according to delivery areas**, otherwise, *NullPointerException* will be thrown if the said delivery area does not exist. Testing will be carried out to ensure that if the same item has been ordered twice, the previous quantity will be added to the latest quantity. Also, testing will be done to ensure that total price of ordered items are calculated based on the item's price (*Member price - member/Non-member price - non-member*) multiplied by the quantity ordered. The **calculation of total price of the order** is tested to ensure it is correct based on ***correct delivery rate*** and ***total price of ordered items*** and that it meets the business requirements provided in the guideline. Moreover, testing will be done to ensure if the ***total price of ordered items*** is less than RM25, there will be an *additional charge* of RM3. Tests will also be carried out to ensure that **empty order or null order or quantity equal or less than 0** will throw *IllegalArgumentException*.

1.1.5 Payment Module

Testing will be done to verify whether the payment is correct or not. Testing is done to verify that an will be thrown when **Payment is made** and the **order status is pending**. Next, testing will also be done to verify that Payment only **accepts *Online Banking*** or ***Credit Card***, anything other than that will result in an *exception* being thrown. Third, testing will also be done to ensure that Payment will return ***messages*** that correspond to either **failure** or **success** of payment. Lastly, testing will be done to ensure that calling an ***under development*** method will result in throwing an *exception*.

1.1.6 File Handler Module

In this module, testing will be done to make sure each file needed by application code **exists**. In doing so, we can make sure that if IOException is thrown, it will be ***due to file content***, not because of missing files. Lastly,**writing** to and **reading** from file content will be tested to ensure that the function ***writes*** and ***reads*** the way we intend it to be, hence, if error arise, we know it would be due to the arrangement of attributes of object in String, not due to ***write/read malfunction***

Objective

The objective of this test plan is to **find bugs and defects** within the scope of the program mentioned above. By finding bugs and defects, the *errors made and other relevant information on the errors made* while developing the program can be found, and therefore, **preventing the failures** that may be caused by human mistakes. Programmers would be able to ***fix the bug***, ***improve the quality*** of the program and ***overall experience*** of users.

The second objective of this test plan is to test if the **functionalities provided by the program** fulfill the **functional requirements specified by the client**. This can reduce the *business risk* and *gain confidence*. Although the concept of ***impossibility of exhaustive testing holds true***, we can do our best to ***find as many bugs as possible*** and build a higher level of confidence in the system by referring to other general testing principles.

<u>Test Basis</u>

In the homemade cakes and pastries ordering system, the test basis of the system included:

- The system should allow the member to log in to the system.
- The system should allow the user to register as the member or guest.
- The system should allow the user to make orders for cakes and pastries.
- The system should allow the user to add the item in the same order.
- The system should added previous item ordered quantity with the new ones if user happened to order same item twice
- The system should calculate total order price as the following
    - The price of all the items ordered will be calculated based on user type(Eg. Member price for member, Non-member price for guest).
    - If items ordered are promotional, 5% discount on the price of item will be calculated
    - Addition charge of RM3 will be added if total price is less than RM25
    - Finally, total order price will be total price + delivery area charge
    - Each item is checked to ensure the amount of promotional and non-promotional items is exactly the same as stated in business requirements.
- The system should calculate the total price of the order including the delivery charges.
- The system should only allow member to have the option to choose whether to pay now or later
- The system should allow the user to make payment through credit card or online banking.
- The system should save registered member and the orders they made into 2 different files

<u>Test Conditions</u>

1.2.1 Item Module

- Each item's hashCode is checked to ensure its uniqueness.
- Each item's item type is checked to ensure the amount of item with item type cake and pastry is exactly the same as stated in business requirement.
- Each item is checked to ensure the amount of promotional and non-promotional items is exactly the same as stated in business requirements.
- Each item is checked to ensure that each item’s member price is lower than non-member price

1.2.2 Customer Module

- The area and district for registration will be checked if it’s among the listed area and district
- If input requires number only, number validation will be carried out to ensure the string input is parsable as number and its within a certain range
- If input requires string only, string validation will be carried out to ensure the string input start with alphabet and only contains alphabet and space

1.2.3 Member Module

- The ID and password of member will be checked during login
- The contact for registration will be checked against the proper format of contact setted by Malaysia

1.2.4 Order Module

NOTE: ONLY non-promotional item has item with price below RM 25, therefore, promotional item price will ALWAYS be larger than RM25

- The user is a member, and the item chosen is non-promotional item, total price of the order will be calculated by multiplying the member price of item with the quantity ordered
- The user is a guest, and the item chosen is non-promotional item, total price of the order will be calculated by multiplying the non-member price of item with the quantity ordered
- The total order price is less than RM25, an additional charge of RM3 will be added
- The total order price is equal or more than RM25, an additional charge of RM3 will not be added
- The delivery rate charged on the order is checked based on the area of the user
- The user ordered the same item twice, previous ordered item quantity will add to the latest quantity.
- The total price that users need to pay is checked after the total price of ordered items and delivery rate is computed.

1.2.5 Payment Module

- The payment will accept Order and Payment Type as well as account number, if order status is Pending, or Payment type is neither Online Banking or Credit Card, an IllegalArgumentException will be thrown

1.2.6 File Handler Module

- The file handler will be able to add/remove new data from its list
- The writer will write content to a file
- The reader will be able to read the exact content from file

Unit Test

Test Entry

- Planning phase has been completed.
- System design is properly reviewed.
- Business and functional requirements are defined.
- Testable codes or units are available.

Exit Criteria

- Successful execution of the unit tests.
- All the identified bugs have been fixed and closed.
- Project code is complete.

Integration Test

Test Entry

- Completion of Unit Testing

Exit Criteria

- Successful execution of the integration tests.
- Correct performance of the software according to the requirements specified.
- Priority bugs have been fixed and closed.

<u>Part B</u>

2. Test Design (*Decision Table*)

2.1. Customer Registration

2.1.1 District Validation

Conditions	1	2	3	4	5	6
String contains number	Y	-	-	-	-	-
String contains symbols	-	Y	-	-	-	-
String contains alphabets and space but not within 3 districts listed	-	-	Y	-	-	-
String contains white space	-	-	-	Y	-	-
String contains empty space	-	-	-	-	Y	-
String entered is among 3 districts listed	-	-	-	-	-	Y
Action						
Repeat input	Y	Y	Y	Y	Y	N
Ask Area	N	N	N	N	N	Y

2.1.2 Area Validation

Conditions	1	2	3	4	5	6
String contains number	Y	-	-	-	-	-
String contains symbols	-	Y	-	-	-	-
String contains alphabets and space but not within 3 districts listed	-	-	Y	-	-	-
String contains white space	-	-	-	Y	-	-
String contains empty space	-	-	-	-	Y	-
String entered is among 20 areas listed	-	-	-	-	-	Y
Action						
Repeat input	Y	Y	Y	Y	Y	N
Ask Street Name	N	N	N	N	N	Y

2.2. Member Login and Registration

Member module extends Customer module

2.2.1 Login as member

Conditions	1	2	3	4
Valid ID	N	N	Y	Y
Valid password	N	Y	N	Y
Action				
Repeat input	Y	Y	Y	N
Login successful	N	N	N	Y

2.2.2 Valid Contact(For registration)

Conditions	1	2	3	4	5	6
String contains random number	Y	-	-	-	-	-
String contains alphabets	-	Y	-	-	-	-
String contains symbols	-	-	Y	-	-	-
String contains white space	-	-	-	Y	-	-
String contains empty space	-	-	-	-	Y	-
String follow the format of contact of Malaysia	-	-	-	-	-	Y
Action						
Repeat input	Y	Y	Y	Y	Y	N
Create member	N	N	N	N	N	Y

2.3. Make Order

2.3.1 Make Valid Order

DC - Delivery Charge(RM 3)

M - Member Price

NM - Non-member Price

Q - quantity

	Member				Non Member			
Conditions	1	2	3	4	5	6	7	8
Promotional item	N	Y	Y	N	N	Y	Y	N
Non-Promotional item	Y	Y	N	Y	Y	Y	N	Y
Total more than 25	N	Y	Y	Y	N	Y	Y	Y
Action								
Total Price	{M*Q} + DC	{M*0.95*Q} + {M*Q}	{M*0.95*Q}	{M*Q}	{NM*Q} + DC	{NM*0.95*Q} + {M*Q}	{NM*0.95*Q}	{NM*Q}

2.3.2 Make Invalid Order

Conditions	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Customer’s area input contains empty space	Y	-	-	-	-	-	-	-	-	-	-	-	-	-
Customer’s area input contains numbers	-	Y	-	-	-	-	-	-	-	-	-	-	-	-
Customer’s area input contains symbols	-	-	Y	-	-	-	-	-	-	-	-	-	-	-
Member’s area input contains alphabets	-	-	-	Y	-	-	-	-	-	-	-	-	-	-
Member’s invalid area input contains symbols	-	-	-	-	Y	-	-	-	-	-	-	-	-	-
Items ID out of range between 1-20	-	-	-	-	-	Y	Y	-	-	-	-	-	-	-
Customer make empty order	-	-	-	-	-	-	-	Y	-	-	-	-	-	-
Customer make null order	-	-	-	-	-	-	-	-	Y	-	-	-	-	-
Member make empty order	-	-	-	-	-	-	-	-	-	Y	-	-	-	-
Member make null order	-	-	-	-	-	-	-	-	-	-	Y	-	-	-
Order with 0 quantity	-	-	-	-	-	-	-	-	-	-	-	Y	-	-

Order status success	-	-	-	-	-	-	-	-	-	-	-	-	Y	-
Order status pending	-	-	-	-	-	-	-	-	-	-	-	-	-	Y
Action														
Throw Exception	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N

2.4. Make Payment

Conditions	1	2	3	4	5	6	7	8	9
String is O	Y	-	-	-	-	-	-	-	-
String is o	-	Y	-	-	-	-	-	-	-
String is C	-	-	Y	-	-	-	-	-	-
String is c	-	-	-	Y	-	-	-	-	-
String contains empty space	-	-	-	-	Y	-	-	-	-
String contains white space	-	-	-	-	-	Y	-	-	-
String contains alphabets	-	-	-	-	-	-	Y	-	-
String contains numbers	-	-	-	-	-	-	-	Y	-
String contains symbols	-	-	-	-	-	-	-	-	Y
Action									
Payment type	Online Banking	Online Banking	Credit card	Credit card	-	-	-	-	-

2.5. Item

Item Validation

Conditions	1	2	3	4
Search for number of items	Y	-	-	-
Search for number of promotional Item and discount is 5%	-	Y	-	-
Search for number of item’s item type is cakes	-	-	Y	-
Each item’s member price is lower than non-member price	-	-	-	Y
Action				
Number of items	20	5	13	20

Test Case

Please refer to Excel File (Test_Case_Group20)

Part C

1.0 Assumptions

1. After a customer register as a member, an ID number will be assigned and received via SMS.
2. Postal code of Melaka is not restricted as long as it is 5 digit numbers.
3. Credit card numbers contain 16 digits.
4. Online banking accounts contain 10 digits.
5. Orders with the same items and quantity are acceptable by adding into the latest quantity, and will not be overwritten or replaced.
6. Only members can choose to pay later.
7. Since order only available in Melaka, therefore, this app is only for Melaka residence, indicating that their State is hardcoded

2.0 Application Code

Please refer to folder src/main

3.0 Test Code

Please refer to folder src/test

4.0 Class Diagram

Please refer to image file (classDiagram.png)