




Individual Sprint Work Summary – to be completed for each method you implemented and tested.

Student Name: Meet Mehta	Sprint Number:2
Student Number: A00258745	Group Number:
Method User Story:	
<div><div><div><div><div>ID: BOO3</div><div>Points: 5</div><div><div><div> Accepted</div></div></div><div>more </div></div><div><div>Theme: Book Keeping</div><div>As a store manager</div><div>I want to add an order in orderbook</div><div>So I can manage customer order details</div></div></div><div><div>1. Verify that customer ID exists in Customer Book</div><div>2. Verify that there is only one entry per customer ID</div><div>3. Verify frequency is daily, weekly or once a month</div><div>4. Verify that publication name is valid</div><div>5. Verify that starting date is valid</div></div></div></div> <div><div><div><div><div>ID: BOO4</div><div>Points: 3</div><div><div><div> Accepted</div></div></div></div><div><div>Theme: Book Keeping</div><div>As a store manager</div><div>I want to update an order in orderbook</div><div>So I can update customer requirements</div></div></div><div><div>1. Verify that customer ID is valid</div><div>2. Verify that any non delivery dates are updated</div><div>3. Verify new customer requirements are updated</div></div></div></div> <div><div><div><div><div>ID: BOO7</div><div>Points: 2</div><div><div><div> Accepted</div></div></div></div><div><div>Theme: Book Keeping</div><div>As a store manager</div><div>I want to update product</div><div>So I can assign orders</div></div></div><div><div>1. Verify Product Name is valid</div><div>2. Verify Product Price is valid</div><div>3. Verify Product is Added for valid entries</div></div></div></div>	

ID: BOO7

Points: 2

Accepted

Theme: Book Keeping

As a store manager

I want to update product

So I can assign orders

1. Verify Product Name is valid

2. Verify Product Price is valid

3. Verify Product is Added for valid entries

Class Name: OrderRegistration	
Method Name: addOrder	
If applicable, for your method, identify all Equivalence Partitions: Partition 1 : All valid parameters Partition 2 : Invalid Customer Partition 3 : Invalid Product Partition 4 : Invalid Frequency Partition 5 : Invalid StartDate Partition 6 : Invalid SuspensionDate	
Number of EP Test Cases: 6	% EP Coverage Achieved: 100

Class Name: OrderRegistration	
Method Name: updateOrder	
If applicable, for your method, identify all Equivalence Partitions: Partition 1 : All valid parameters Partition 2 : Invalid Customer Partition 3 : Invalid Product Partition 4 : Invalid Frequency Partition 5 : Invalid StartDate Partition 6 : Invalid SuspensionDate	
Number of EP Test Cases: 6	% EP Coverage Achieved: 100
Class Name: Product Registration	

Individual Sprint Work Summary – to be completed for each method you implemented and tested.

Method Name: updateProduct	
If applicable, for your method, identify all Equivalence Partitions: Partition 1 : All valid parameters Partition 2 : Invalid Name Partition 3 : Invalid Price	
Number of EP Test Cases: 3	% EP Coverage Achieved: 100

Class Name: DBConnection	
Method Name: getCustomers	
If applicable, for your method, identify all Equivalence Partitions: Returns the ArrayList of all customers	
Number of EP Test Cases:	% EP Coverage Achieved:

Class Name: DBConnection	
Method Name: getProducts	
If applicable, for your method, identify all Equivalence Partitions: Returns the ArrayList of all products	
Number of EP Test Cases:	% EP Coverage Achieved:

Class Name: DBConnection	
Method Name: insertProduct & updateProduct	
If applicable, for your method, identify all Equivalence Partitions: Partition 1 : Always takes valid Order as input parameter	
Number of EP Test Cases:	% EP Coverage Achieved:

Class Name: DBConnection	
Method Name: insertOrder & updateOrder	
If applicable, for your method, identify all Equivalence Partitions: Partition 1 : Always takes valid Product as input parameter	
Number of EP Test Cases:	% EP Coverage Achieved: