SPL 191 Assignment 2 Grading

1. Automatic Tests

Provided to you a package which contains the Jsons files used for testing, as well as the testing script.
The following is description for the tests used.

Test number	Description	Weight in pts.	Error code
1	Simple test. The same test in the website.	10	**T1**
2	Simple test. Only one instance of each service. 2 books (the first one price is 80, the second is 20) with 10 copies each.. 10 customers with balance 100 in there credit card. Each customer orders both books (in tick 5 and 8). Output: all customers should have receipt for both books. Their balance should be 0. The final amount of each book should be 0. The earnings of the store should be 1000.	4	**T2**
3	Simple test. Same test in 2 - the orders now are on tick 1, and multiple instances of all services. The output is the same like in 2.	3	**T3**
4	Simple test. Same test in 2 - only one instance of resourceService and multiple instances of the other services. The output is the same like in 2.	3	**T4**
5	Stress test. Few customers and many number of orders. 15 books on inventory, each with initial amount 10. The price of each book is 20. Each customer order each book twice.	3	**T5**

	Output: Each customer should have at most 5 receipts.. The earnings of the store should be 1000.		
6	Stress test. Same like in 5 but with one instance of selling service. The output is the same.	3	**T6**
7	Stress test. Only one book of each type of book on inventory. Each book should be taken by exactly one customer. There must be 15 receipts total in money register. The earnings of the store are 300. The number of receipts of each customer is at most 5.	4	**T7**
8	Stress test. few customers, many number of orders - orders in the same tick. All customers order two same books - where they have money only for one book. 10 receipts total, one receipt per customer. Earnings 200.	4	**T8**
9	Terminating gracefully test - Test that thread terminate gracefully: speed = 1. Only checks that there is no deadlock, and the invariants hold (number of books isn't negative, balance in credit card is not negative).	6	**T9**

Common mistakes that cause one or more tests to fail:

Mistake number	Problem
1	Deadlocks.
2	Money register is not thread safe/ Did not handle the problem of multiple threads trying to update the total earnings simultaneously.
3	Inventory is not thread safe/ Did not handle the problem of multiple selling service trying to charge the credit card of the same customer simultaneously.
4	The sequence of the three operations: checking the availability of the book, charging credit card and taking the book, is not

	atomic.
5	Did not make sure that all services do not miss the first tick (that is, starting sending orders only when all services have initialized).
6	Sleeping an arbitrary amount of time waiting for orders to complete (using Future.get(TimeUnit) instead of Future.get()).
7	Sleeping an arbitrary amount of time waiting for all threads to terminate. The problem when this time is much longer that the timeout we used.
8	Logical errors. Such as: not checking the amount of books is greater than zero, not checking the balance in credit card is enough, etc.

2. Frontal Check and Code Review

Note: In part 1 you were required to write a general framework implementation. The implementation should be independent from part 1. In particular, you cannot assume that all micro-services unregister at once, or before completing all events. This was clarified clearly in the assignment page.

Error	**Description**	**Points**
F1	Did not identify the race condition between unregister and sendEvent in MessageBus.	6
F2	Synchronizing the complete method. The race-condition between sendEvent and complete can be solved easily without synchronization.	**0**
F3	Full synchronizing the awaitMessage method. This causes that only one micro-service fetches a message each time. Very bad.	3
F4	MessageBus is synchronized. Too much blocking in MessageBus. You needed to block micro-services as less as possible.	6

F5	Did not identify the race-condtion of two customers attempting to take the same book simultaneously.	6
F6	One or more of Inventory, MoneyRegister, or ResourcesHolder is not threadsafe.	6
F7	Full synchronization of Inventory	4
F8	deliver() method is called by another thread than the thread of LogisticsService. It is mentioned in the assignment that the LogisticsService should handle the delivery event.	**0**
F9	The method acquireVehicle is blocking. This is bad implementation (and can cause deadlocks in our scenario). The return type of this method should have implied the better implementation.	3
F10	Did not identify the race condition between acquireVehicle and releaseVehicle	3
F11	Partially implementation	60
F12	Did not submit unit tests	10

3. The grade

A = automatic tests grade (40%).
B = Unit tests, frontal check and code review grade (60%) - (60-total points reduced).

GRADE = A + B