

COMP6231 Project 1

Current Version: 0.1

Date: Oct. 12, 2012

Author: Yuan Tao

Instructor: Dr. R. Jayakumar

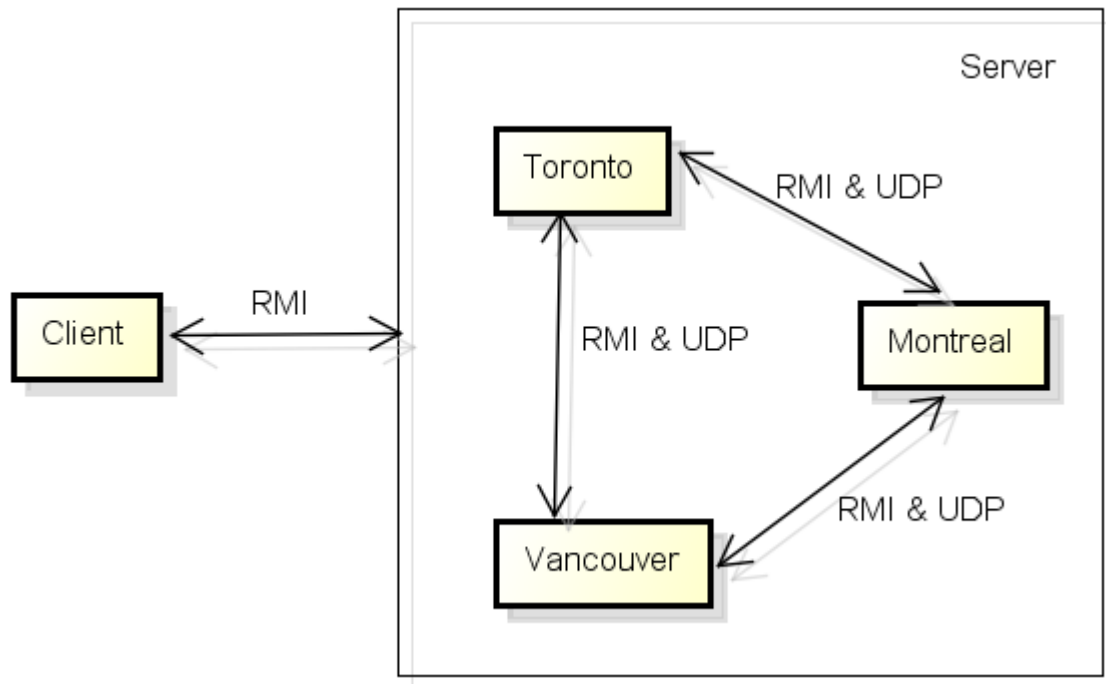
Concordia University

Table of Contents

1.	Design-----	2
1.1.	System Architecture -----	2
1.2.	Class Diagram-----	3
1.2.1.	drs-common -----	3
1.2.2.	drs-server -----	4
1.2.3.	drs-client -----	5
2.	Test scenarios-----	6
	Appendix A: Revision History-----	7

1. Design

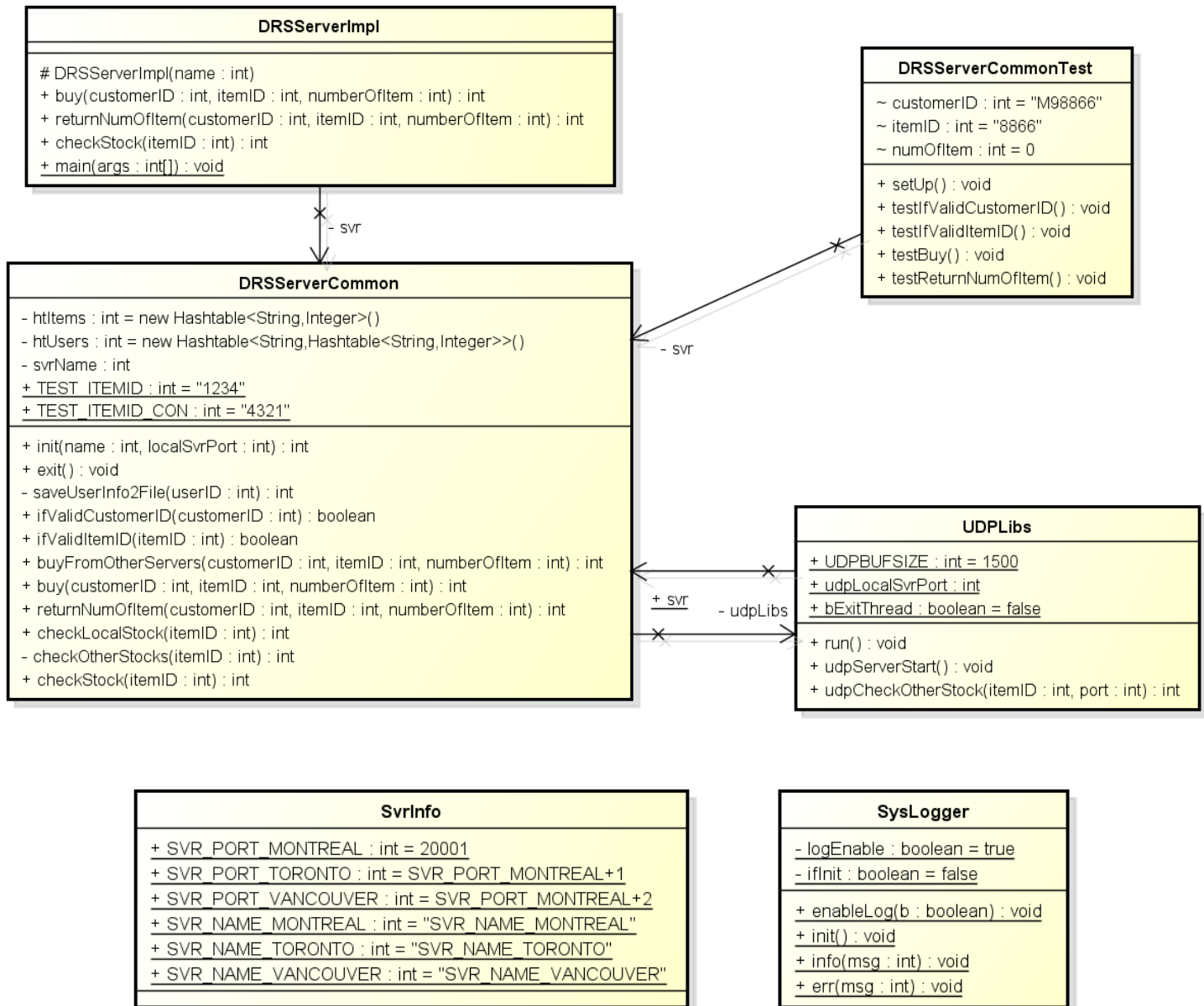
1.1. System Architecture



For the communication between clients and servers, there is only one way which is RMI. But for the communication among servers, there are two ways which are both RMI and UDP.

1.2. Class Diagram

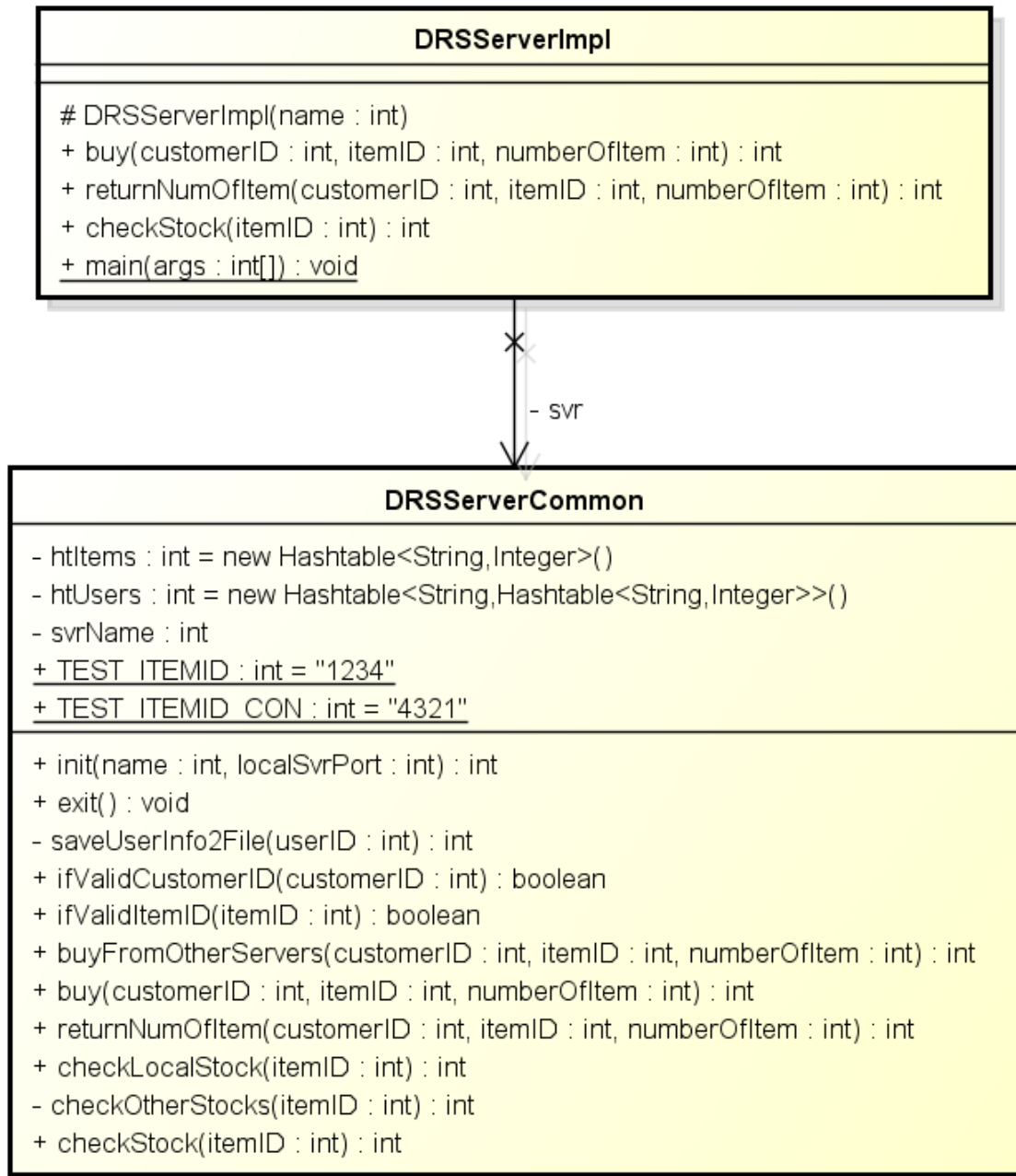
1.2.1. drs-common



From the figure, there is a Unit Test Class whose name is DRSServerCommon which is used to test some of the interfaces of the class DRSServerCommon, a class that used by all the servers.

1.2.2. drs-server

The project has three server objects, which are identified by their different names that defined in drs-common.SvrInfo.



1.2.3. drs-client

DRSClient
<u>+ main(args : int[]) : void</u>

ConcurrentTest
- svrName : int = SvrInfo.SVR_NAME_MONTREAL - customerID : int = "M10001" - itemID : int = DRSServerCommon.TEST_ITEMID - numberOfItem : int = 10
+ ConcurrentTest(svrName_ : int, customerID_ : int, itemID_ : int, numberOfItem_ : int) - simpleTestBuy(svr : DRSCCommon, svrName : int, customerID : int, itemID : int, numberOfItem : int) : void - simpleTestReturn(svr : DRSCCommon, svrName : int, customerID : int, itemID : int, numberOfItem : int) : void + simpleTest(svrName : int, customerID : int, itemID : int, numberOfItem : int) : void + run() : void <u>+ concurrentTest(svrName : int, customerID : int, itemID : int, numberOfItem : int) : void</u>

2. Test scenarios

1. Unit test cases for DRSServerCommon class.
2. For the client program, it provides two interfaces to do the test:
 - a) A customer buys one item many times and checks the item after each buying, and try to buy the item when it is unavailable. Then the customer returns some of the items he has bought and then checks the item and buys the items again until it is unavailable. Then the client tries to buy the item again.
 - b) Three customers, one for each server, concurrently (each one is running in a thread) buying the same item for random times and check the item after every time buying it. Then they return the item for random times and check the item once they return the item. At last, they buy the item again for random times.
3. If we want to test the scenarios that many customers buy different items concurrently just like scenarios 2.b, we need to modify the following code and then run these clients:

DRSServerCommon.java, line 45:

```
public static final String TEST_ITEMID_CON = "4321";  
// Just to make sure that different clients have different itemIDs.
```


Appendix A: Revision History

Version	Date	Author	Remark
V0.1	Oct. 12, 2012	Yuan Tao	Draft