

CONCORDIA UNIVERSITY

DEPARTMENT OF
COMPUTER SCIENCE AND SOFTWARE ENGINEERING

COMP 6231, Winter 2019

Instructor: R. Jayakumar

ASSIGNMENT 2

Issued: Feb. 15, 2019

Due: Mar. 8, 2019

Note: *The assignments must be done individually and submitted electronically.*

Distributed Library Management System (DLMS) using Java IDL (CORBA)

In this assignment, you are going to implement a Distributed Library Management System (DLMS) from Assignment 1 in CORBA using Java IDL. In addition to the 3 student operations (namely- *borrowItem*, *findItem*, and *returnItem*) introduced in Assignment 1, the following operation needs to be implemented.

- *exchangeItem (studentID, newItemID, oldItemID)*

A student might invoke this operation to change an item she/he has already borrowed with another item in any library. In this case, the *current_library_server* (which receives the request from the user) first checks whether the user has borrowed the old item, then checks with the *new_library_server* (from which the new item has to be borrowed) whether this new item is available, and if both checks are successful then atomically lends the new item to the user and returns the old item. That is, borrow and return operations should both be successful or none of them should be done. Note that all these checks, borrow and return operations should be done using UDP/IP messages as they are server-to-server communications.

In this assignment you are going to develop this modified application in CORBA using Java IDL. Specifically, do the following:

- Write the Java IDL interface definition for the modified DLMS with all the 7 specified operations.
- Implement the modified DLMS. You should design a server that maximizes concurrency. In other words, use proper synchronization that allows multiple users to correctly perform operations on the same or different records at the same time.
- Test your application by running multiple clients with the 3 servers. Your test cases should check correct concurrent access of shared data, and the atomicity of *exchangeItem* operation.

Your submission will be graded for correct and efficient implementation of all the operations in addition to correct use and implementation of mutual exclusion in accessing shared data and proper exploitation of concurrency to achieve high performance.

MARKING SCHEME

- [30%] *Design Documentation*: Describe the techniques you use and your architecture, including the data structures. Design proper and sufficient test scenarios and explain what you want to test. Describe the most important/difficult part in this assignment. You can use UML and text description, but limit the document to 10 pages. Submit the documentation and code electronically by the due date; print the documentation and bring it to your DEMO.
- [70%] *DEMO in the Lab*: You have to register for a 5–10 minutes demo. Please come to the lab session and choose your preferred demo time in advance. You cannot demo without registering, so if you did not register before the demo week, you will lose 40% of the marks. Your demo should focus on the following.
 - [50%] *The correctness of code*: Demo your designed test scenarios to illustrate the correctness of your design. If your test scenarios do not cover all possible issues, you will lose part of marks up to 40%.
 - [20%] *Questions*: You need to answer some simple questions (like what we have discussed during lab tutorials) during the demo. They can be theoretical related directly to your implementation of the assignment.

QUESTIONS

If you are having difficulties understanding any aspect of this assignment, feel free to contact your teaching assistant (Ms. Kritika Sharma at kritikasharma462@gmail.com, Mr. Pranav Bhatia at pb.comp6231@gmail.com or Mr. Kishan Bhimani at kishanbhimani9111@gmail.com). It is strongly recommended that you attend the tutorial sessions, as various aspects of the assignment will be covered there.