COMP6231 /1 – Sections BB – Summer 2020

ASSIGNMENT #2

Due: June 28th

Important Note

- The work can be realized ***individually***.
- Your program should be compiled, executed and return the expected results; otherwise a mark 0 (zero) will be assigned.
- The delivery must be made no later using the Website submission as mentioned in the course outline.
- If you are having difficulties understanding sections of this assignment, feel free to email the Teaching Assistants. It is strongly recommended that you attend the tutorial sessions which will cover various aspects of the assignment.

Distributed Player Status System (DPSS) using Java IDL

In this assignment, you are going to implement the distributed player status system (DPSS) from Assignment #1 in CORBA using Java IDL. In addition to the 4 operations introduced in *Assignment #1* (namely, *createPlayerAccount*, *playerSignIn*, *playerSignOut*, and *getPlayerStatus*) the following operations also need to be implemented.

Additional Operation for Player

- *transferAccount* (*Username*, *Password*, *OldIPAddress*, *NewIPAddress*)

  When a player invokes this method from his/her geo-location, the server associated with this player (determined by the *OldIPAddress*) searches its hash table to check if the account with this *Username* exists. If it exists, the entire account is transferred to the geo-location server associated with the *NewIPAddress*. Note that the account should be removed from the hash table of the current server and should be added to the hash table of the remote server atomically (that is, both operations should succeed or none should succeed). The server informs the player whether the operation was successful or not and both the server and the player store this information in their logs.

Additional Operation for Administrator

- *suspendAccount* (*AdminUsername*, *AdminPassword*, *AdminIP*, *UsernameToSuspend*)

  When an admin invokes this method from his/her geo-location, the server associated with this admin (determined by the *AdminIP*), if the credentials are accepted, searches its hash table to check if the account with this *UsernameToSuspend* exists. If it exists, the entire account is removed from the hash table of this geo-location server. The server informs the admin whether the operation was successful or not and both the server and the admin store this information in their logs.

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

- Write the Java IDL interface definition for the modified DPSS with all the 6 specified operations
- Implement the modified DPSS. You should design a server that maximizes concurrency. In other words, use proper synchronization that allows multiple officers to perform operations for the same or different records at the same time
- Test your application by running multiple players with the 3 servers. Your test cases should check correct concurrent access of shared data, and the atomicity of *transferAccount* operations (e.g. what if an account being suspended needs to be transferred and both operations were initiated at the same time?).

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

Submitting Assignment #2

- Naming convention for zip file: Create one zip file, containing all source files (.java, .doc or .pdf or .txt, etc.) for your assignment using the following naming convention:

  The zip file should be called *a#_studentID*, where # is the number of the assignment *studentID* is your student ID number. For example, for the second assignment, student 123456 would submit a zip file named *a2_123456.zip*

- Submit your zip file at: https://fis.encs.concordia.ca/eas/ as **Programming Assignment** and submission **#2**. Assignments submitted to the wrong directory would be discarded and no replacement submission will be allowed.
- Submit only **ONE version** of an assignment. If more than one version is submitted the last one, before the deadline date, will be graded and all others will be disregarded.

Evaluation Criteria of Assignment #2 (100 points)

Activities	Points
Q1: Describe the techniques you use and your architecture, including the data structures: - Design of architecture: 5 pts. - Description of techniques (including data structures) used: 5 pts.	10 pts.
Q2: Design proper and sufficient test scenarios and explain what you want to: - Design of 10 scenarios: 5 pts. - Explanation of each scenario: 5 pts.	10 pts.
Q3: Describe the most important/difficult part in this assignment	10 pts.
Q4: You must provide the following: - UML design of your architecture (Web Service design, Server design): 5 pts. - Text description of your design: 4 pts. - Provide a document to 10 pages: 1 pt.	10 pts.
Q5: The correctness of code	60 pts.