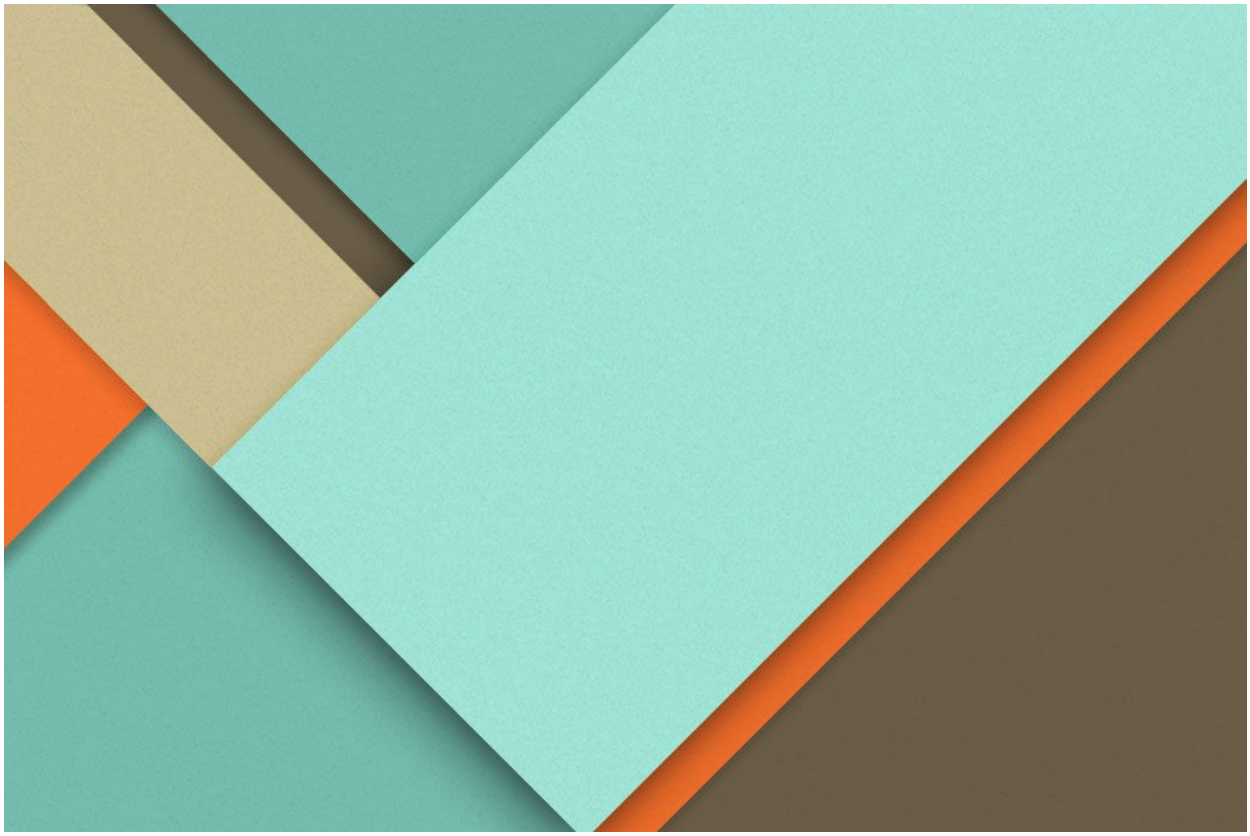

# Bulletin Board SystemV2.0


Names

Sherif Hamdy 34

Sherif Hassan Waly 33

Tarek Mohamed Nawara 38

## Problem Statement

In this assignment you will implement news bulletin board system, like we did in Assignment 1. But this time we will use RPC/RMI instead of sockets. So, we will be developing another distributed client/server version of the solution using a service-oriented request/reply scheme. Typically, we will use the same code we have written for Assignment 1 but using RMI for all necessary communication instead of the Sockets. Also, while RMI will take care of the client threads (we don't need to handle it like sockets), we still need to provide the necessary synchronization. In sum, the major difference from Assignment 1 is that Read and Write requests of clients will be remote method invocations. Unless otherwise specified in this document, Assignment 1 specification is valid for this assignment

## Design

The project structure is simple we have the following classes:

### IStore

Interface extending remote . It is used for communicating with the implementation throw java RMI. it holds two simple methods. readNews(), writeNews()

### Store

Implementation of the IStore interface, this class implements the two methods in the IStore interface, handling all the synchronization problems and logging the requests in the appropriate log files.

### RmiServer

This class is simply responsible for creating a registry , exporting and binding the store instance so that clients can invoke methods remotely over it.

### SSHManager

This class responsible for communicating with the client hosts throw SSH and executing commands. We use it for transfering the client's code and compiling then executing it.

### LOGHandler

This class is responsible for logging mechanism. All the requests coming to the store are being logged in a synchronous way and flushed periodically to the appropriate files.

### ReadResult

This is a representation of the result returned to the reader client from the store, it holds the rSeq, sSeq and news value of this client.

### WriteResult

This is a representation of the result returned to the writer client from the store, it holds the rSeq, sSeq.

### CommandsBuilder

This is an important class for building all the transfering, compiling and executing the client code on the client hosts.

## Simple workflow

First we create a RmiServer to bind the Store instance, then read all the necessary client code and transfer it to the clients hosts using SSHManager after that we build all the necessary commands for building and executing the code on the client host, when the clients are being executed they will begin to call the store via Java RMI mechanism, this will eventually execute the store code which handles all synchronization problems and will handle the client requests and log them in the correct order.