Name: Sheetal Turaga

Email: [Turaga.s@northeastern.edu](mailto:Turaga.s@northeastern.edu)

**Assignment Overview:**

In project 2, the goal is to build a multi-threaded key-value store with client and server using Java RMI for Remote Procedure call communication. As the name suggests, that client taken the input from the user and makes a remote procedure call that passes on these arguments and returns the appropriate response from the procedure that implemented on a remote server.

Since we were making the server multi-threaded, the idea was to have multiple clients connect using the same port number to perform the operations. Using synchronized maps and UUID that were randomly generated, there was to be a record kept of all the operations that were called using RPC along with the record for all the acknowledgements and call backs that were communicated using the same the port number.

**Technical Impression:**

The plan to start with was to be able to have multiple clients to use one server instance. The server would be multithreaded to allow the inflow of multiple PUT, GET, DELETE calls from the network using the Server Interface to be providing the registry link to the client as well as server stub. All the operations information along with their key-value data, pre-populated or provided by the user is stored using the Properties class.

The Client logger is mostly a helper to traverse the command line arguments and process them while the Client.java file is where the PUT, DELETE, GET operations have been implemented or pre-populated. The Server.java and Key Map Manager together implement the procedure that are called by the client for and the callback for the client with acknowledgements, but it will need to compile and run the ServerManager.java file.

In retrospect to implementing this as well as the previous projects, I would like more practice on how to multi-threading and the way to test it along with learning how to configure dependencies/plugins, compile and run these projects.

\*\*\*\*\*\*\*\*\*\*\*