Assignment 02
Name: Quoc Minh Vu
ID: 40040500

CORBA.IDL

```
module CORBA {
    interface DCMS {
        string createTRecord(in string managerID, in string firstName, in string
lastName, in string address, in string phone, in string specialization, in string
location);
        string createSRecord(in string managerID, in string firstName, in string
lastName, in string coursesRegistered, in string status);
        string getRecordCounts(in string managerID);
        boolean editRecord(in string managerID, in string recordID, in string
fieldName, in string newValue);
        boolean transferRecord(in string managerID, in string recordID, in string
remoteCenterServerName);
        string getRecordType(in string recordID);
        void startUDPServer();
        string printAllRecords();
        string printRecord(in string managerID, in string recordID);
    };
};
```

- All the functions from previous assignment are transformed from RMI to CORBA
- Function transferRecord() is added to transfer a specific record (given the recordID) to another server (given the server name)

CenterServer.java

```
public class CenterServer extends DCMSPOA {
```

- This class extends the class DCMSPOA so after it is registered to the ORB, it can be recognized and get messages dispatched by Object Adapter

```
private ORB orb;

public void setORB(ORB orb_val) {
    this.orb = orb_val;
}
```

- This class has an ORB object and a setORB() function. This function will be invoked when the servant is initiated.

XXXServer.java

```
// Initiate local ORB object
ORB orb = ORB.init(args, null);

// Get reference to RootPOA and get POAManager
POA rootPOA = POAHelper.narrow(orb.resolve_initial_references(Config.CORBA.ROOT_POA));
rootPOA.the_POAManager().activate();

// Create servant and register it with the ORB
CenterServer servant = new CenterServer(Config.Server_ID.MTL);
servant.setORB(orb);

// Get object reference from the servant
org.omg.CORBA.Object ref = rootPOA.servant_to_reference(servant);
DCMS dcmsServer = DCMSHelper.narrow(ref);

// Get the root Naming Context
org.omg.CORBA.Object objRef =
orb.resolve_initial_references(Config.CORBA.NAME_SERVICE);
```

```
NamingContextExt namingContextRef = NamingContextExtHelper.narrow(objRef);

// Bind the object reference to the Naming Context
NameComponent path[] = namingContextRef.to_name(Config.Server_ID.MTL.name());
namingContextRef.rebind(path, dcmsServer);

// Run the server
dcmsServer.startUDPServer();
System.out.println("Server " + Config.Server_ID.MTL.name() + " is running ...");
orb.run();
```

- This class has only one main() function which connect to the ORB core, get the reference to RootPOA, initiate the servant and register it with the ORB, get the object reference to the servant and bind it to naming context. Finally, it starts the server.

ManagerClient.java

```
public class ManagerClient
```

- This class is implemented to test the functionalities of the CORBA distributed system
- It has a GUI interactive interface that allows users to perform a bunch of actions supported by the system, either individually or concurrently.

Explaining multi-threading and synchronizing implementation

- In order to keep system's data's intergrity, the added function supporting transfer records from one server to another need to by synchronous.
- Whenever a manager request to transfer a record, it must wait if there is any other thread editing that particular record.
- When the record transfer is processing, no other thread is allowed to access or edit that record.
- If a thread edit a record which was already transferred, the system will notify users about the failure
- If a thread edit a record before it is transferred, the new information will also appear in the new server