COMP 6231 Distributed System Design

Summer 2017

Instructor: Sukhjinder K.Narula

Assignment 2

Chong Li 27813724
Mengying Ding 40035587

In this assignment, we designed a distributed system by using CORBA. We have three servers(MTL, LVL and DDO) and one Client(ManagerClient) class. What is different from last assignment is that, this time we use CORBA instead of the language specific RMI.

Main Points for System Design

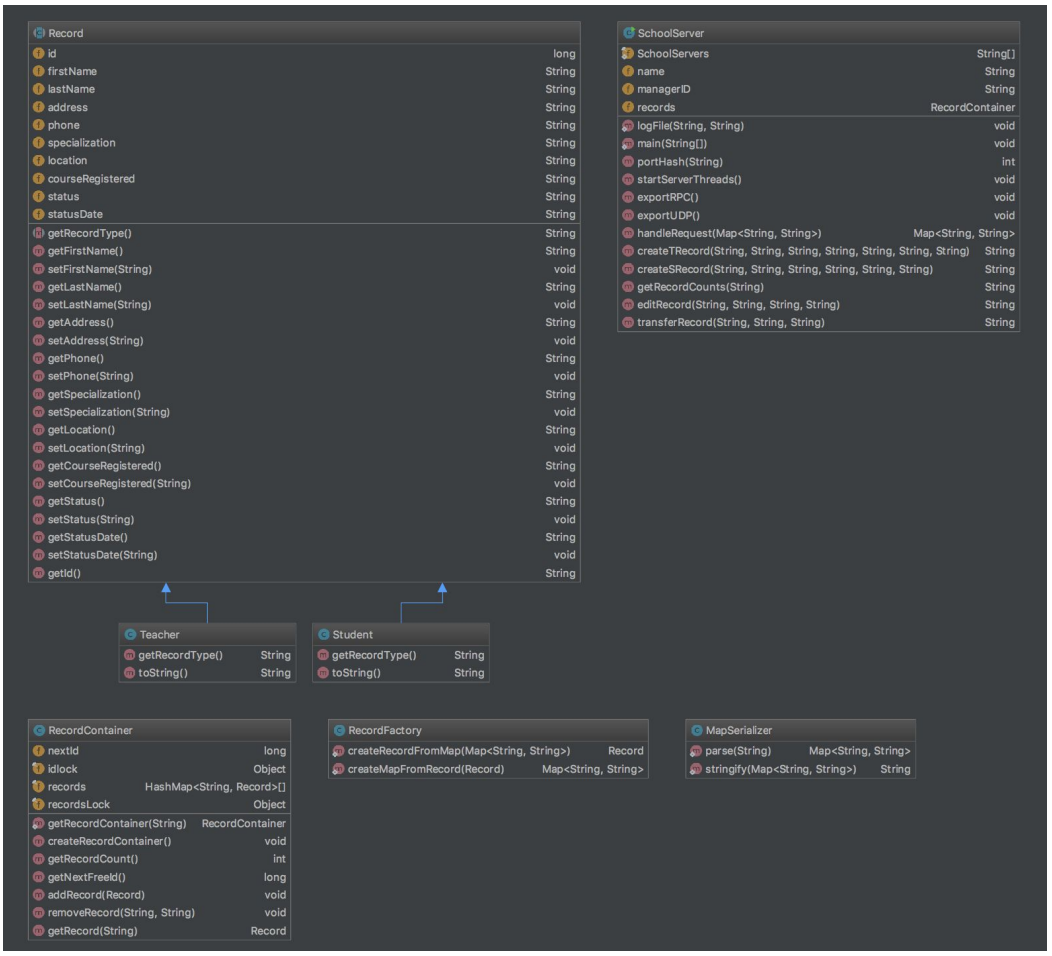
Data Structures

- We designed Record / Student / Teacher classes to hold record information (please refer to the last part of this section)
- A hashmap to hold records on each server. The key is the first character of the last name, the value is a list stores record ID and other information.

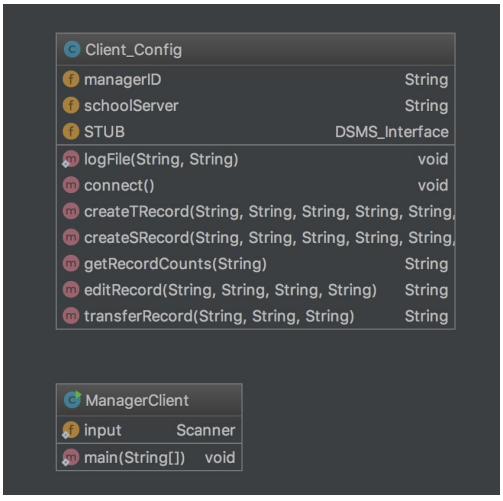
Request Format

We define the format of requests, which includes 4 digits indicating the type of the operation first, and the rest is the detailed information of the request.

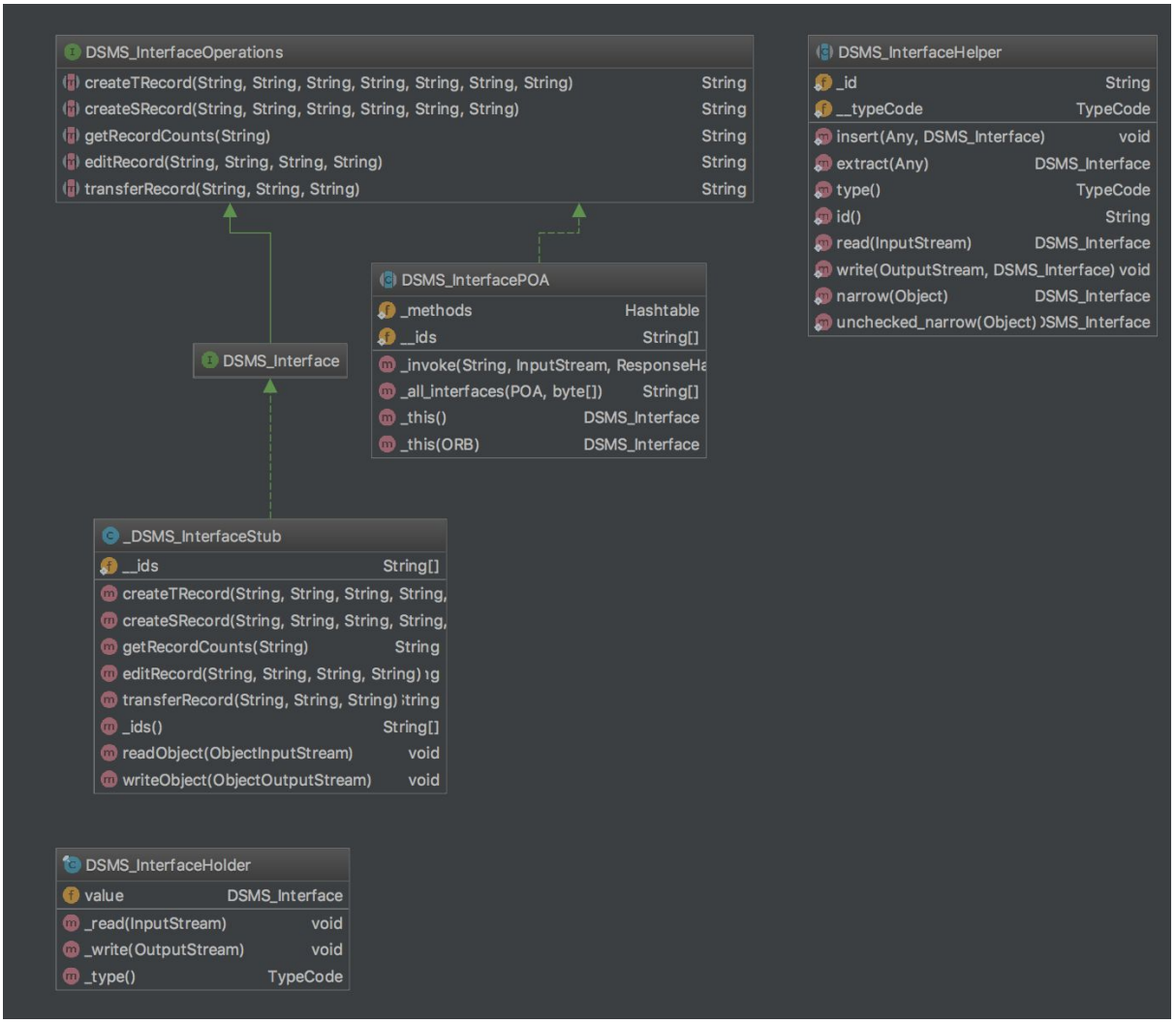
Server



Client



CORBA



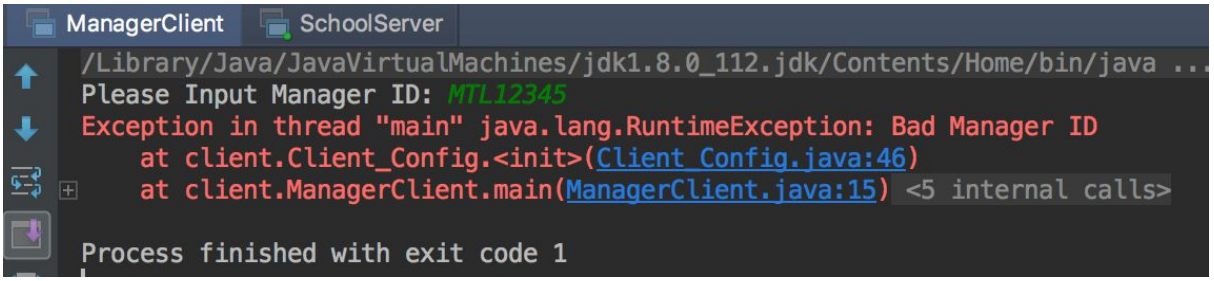
Multithreading / Synchronization

We synchronize blocks that deal with file I/O, read/write and operations related to the hashmap.

Test Cases

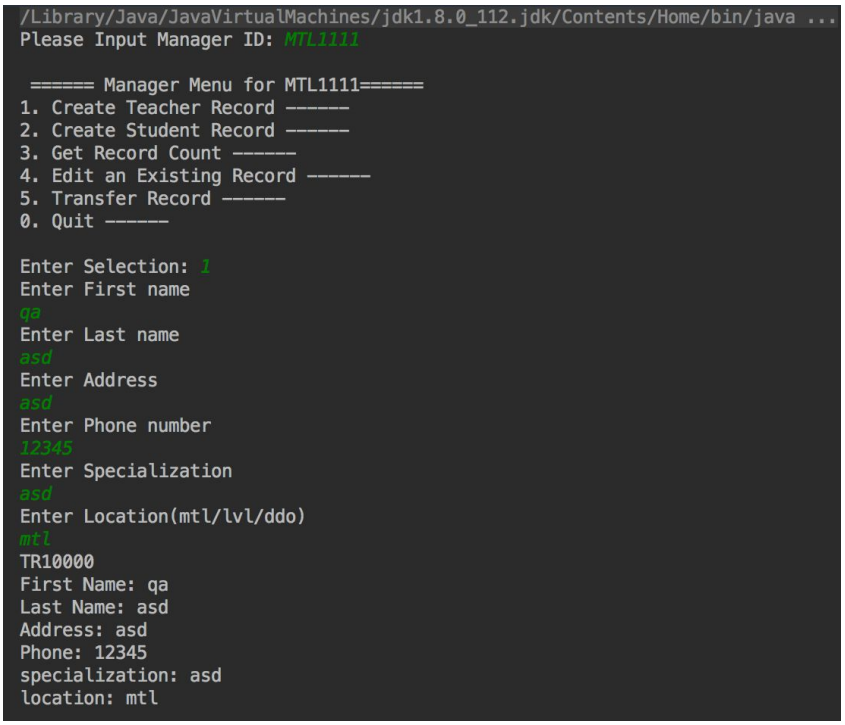
Scenario 1.1

Manager enters invalid ID (the ID isn't stored in any of the 3 servers, wrong format included)



Scenario 1.2

A manager(MTL1111) successfully logs in and choose option 1 to create a teacher record (almost the same for option 2 create a student record)



Scenario 2.1

A manager(MTL1111) chooses option 3 to get counts of records on all 3 servers

```
 ====== Manager Menu for MTL1111======
1. Create Teacher Record ------
2. Create Student Record ------
3. Get Record Count ------
4. Edit an Existing Record ------
5. Transfer Record ------
0. Quit ------

Enter Selection: 3
MTL : 1, LVL : 0, DDO : 0,
```

Scenario 2.2

Using MTL manager(MTL111) to check record count after creating a record on MTL

```
 ====== Manager Menu for MTL1111======
1. Create Teacher Record ------
2. Create Student Record ------
3. Get Record Count ------
4. Edit an Existing Record ------
5. Transfer Record ------
0. Quit ------

Enter Selection: 3
MTL : 0, LVL : 1, DDO : 0,
```

Scenario 3.1

A manager(LVL1111) chooses to edit a student record

```
 ====== Manager Menu for LVL1111======
1. Create Teacher Record ------
2. Create Student Record ------
3. Get Record Count ------
4. Edit an Existing Record ------
5. Transfer Record ------
6. Change to another manager ------
0. Quit ------

Enter Selection: 4
Enter RecordID
SR10000
Enter Field Name
status
Enter New Value
not_active
Manger has edit the status of SR10000 to new value: not_active
```

Scenario 4.1

As a manager(MTL1111), she transfers a record to LVL server

```
 ====== Manager Menu for MTL1111======
1. Create Teacher Record ------
2. Create Student Record ------
3. Get Record Count ------
4. Edit an Existing Record ------
5. Transfer Record ------
0. Quit ------

Enter Selection: 5
Enter Record ID:
TR10000
Enter Remote Server:
LVL
TR10000 has been transferred to LVL
```

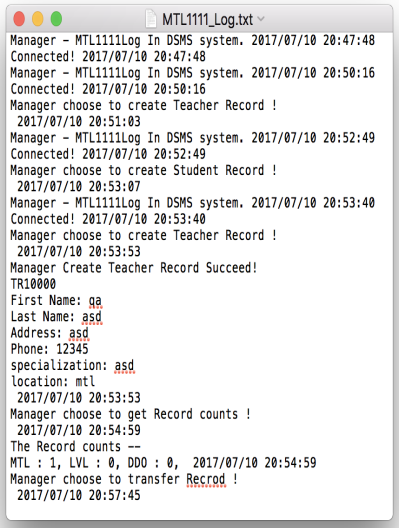
Scenario 4.2

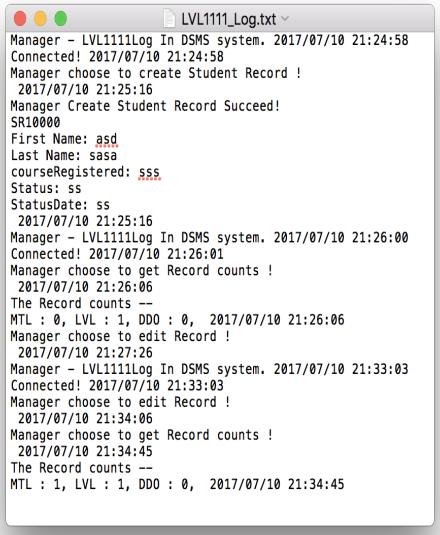
Manager MTL1111 tries to transfer LVL record to MTL, but failed since it's an invalid operation.

```
 ====== Manager Menu for MTL1111======
1. Create Teacher Record ------
2. Create Student Record ------
3. Get Record Count ------
4. Edit an Existing Record ------
5. Transfer Record ------
0. Quit ------

Enter Selection: 5
Enter Record ID:
TR10000
Enter Remote Server:
mtl
_FAIL_
```

Screenshots for Logs （ related to parts of those scenarios above ）





The Most Import Problems We Met

- CORBA standard is more complex than RMI, it costs time to understand the functions of each generated .java file and know how to use them.
- Although we have the tutorial of setting up CORBA, we spent a lot of time to make sure that it is runnable on our own machine.
- CORBA is extremely old-fashioned. When there are problems, sometimes we cannot get accurate solutions.

Conclusion

- Start the project as soon as possible.
- CORBA is too old, nobody is gonna use it.