Online Advising Simulation using RMI & Message Queuing

Contents

[1. System Requirements 1](#_Toc514070683)

[2. Project Files 2](#_Toc514070684)

[3. Steps to run the application 2](#_Toc514070685)

[3.1 Compile all java files 2](#_Toc514070686)

[3.2 Running Message queue server 2](#_Toc514070687)

[3.3 Running Client processes: Student/Advisor/Notification 3](#_Toc514070688)

[3.4 Stopping client processes 4](#_Toc514070689)

[3.5 Stopping server 4](#_Toc514070690)

[4. Limitations 4](#_Toc514070691)

[5. Assumptions 4](#_Toc514070692)

[6. References 4](#_Toc514070693)

### System Requirements

Following are the system requirements of the project:

* Language: java
* Environment: Windows
* JDK version: 1.8 and above

## Project Files

* Message.java: *structure to hold the information of the student request for course clearance from advisor*
* Msg\_Node\_t.java: *class to hold the messages in the queue fashion implemented as a linked list*
* MQService.java: *remote interface for rmi application*
* MQServiceImp.java: *provides implementation of the remote interface*
* MsgQueueServer.java: *creates remote object and bind it to the RMIregistry*
* Advisor.java: *Advisor class searches the message queue server for any requests from student*
* Student.java: *Student class allows student to request for enrollment*
* Notification.java: *Notification class searches for advisor decision and notifies it to student*
* Msg\_File\_Persist.txt: *Stores the pending request*

## Steps to run the application

Open command prompt and go to directory where project files are located

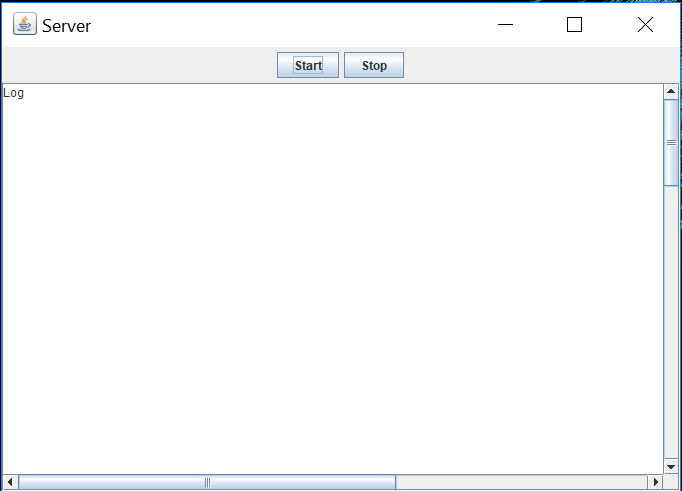
### Compile all java files

* Compile all the .java files
* Start RMI registry: Port number used for connection is 1099 with localhost.

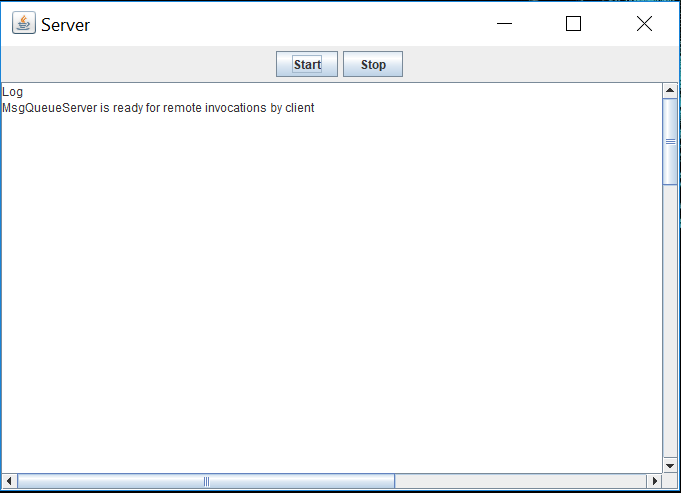
start rmiregistry

### Running Message queue server

* Start Message queue server using following command
  + java MsgQueueServer
* Server application will start and will look like follows:

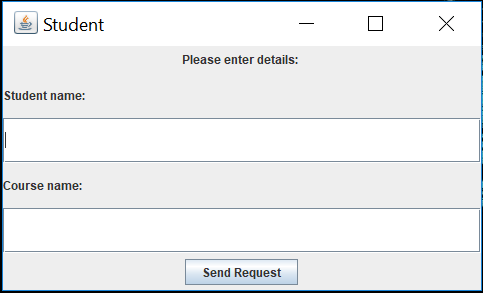


* Click on start button to start the server. Following message will appear after server starts



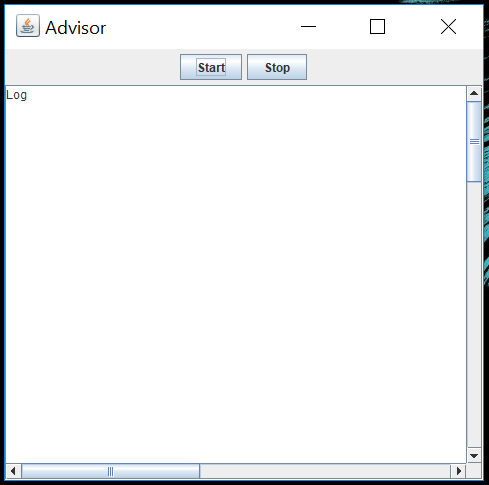
### Running Client processes: Student/Advisor/Notification

* Client process can be started in any order using following commands
  + java Student

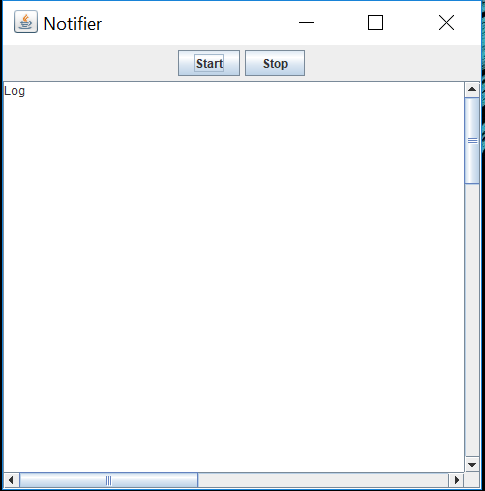


Enter student name and the requested course name and click enter

* + java Advisor



* + java Notification



### 3.4 Stopping client processes

* Advisor process tries to search the students requests every 3 sec until it is killed. The process can be stopped by clicking the stop button on GUI
* Notification process tries to search the advisor decision every 7 sec until it is killed. The process can be stopped by clicking the stop button on GUI

### Stopping server

* Server can be stopped after all three client processes are stopped by clicking the stop button on GUI
* If there are any pending requests the server will store them in text file. This file will be loaded when the server starts next time.

## Limitations

* The testing for application has been done using limited number of entries. The persistent file that stores the messages will look as below

4

advisor,jane ,DS,-1

student,doe ,DB,-1

advisor,smith ,DS,-1

student,john ,DB,-1

* RMI application runs on localhost and remote host.

## Assumptions

* Message queue server should be started first followed by clients in any order.
* After all the processes are stopped Message queue server will stop.

## References

* Starting and stopping thread: <https://stackoverflow.com/questions/11917714/stopping-a-thread-by-a-swing-button>
* Student advisor queue: https://code.google.com/archive/p/student-advisor-mq/source/default/source?page=2
* RMI application: https://www.javatpoint.com/RMI
* Serialize object: https://stackoverflow.com/questions/2374436/when-should-i-implement-java-io-serializable-in-rmi