

## Project requirements

### Introduction

The goal of the project is to create a server application that will handle an instant messaging server

### Applications description:

The server and client applications should be coded in java based on the libraries discovered during the module lessons. Keep it simple. The system should work with users registered to the service and the server will deliver messages to the clients.

### Work to do:

#### Minimal specifications of the **server**:

- × The server must be able to:
  - × Register new users
  - × Maintain a list of registered user (add remove users)
  - × Deliver text messages to a given user
  - × Maintain messages (add remove messages)
  - × Accept **multiple** clients simultaneously (use threads).
  - × Deliver messages safely to the proper users
- × The server must be able to write logs:
  - × On a file
  - × The history must be kept (one file per month)
  - × 3 levels of log (info, warning, severe) should be handled all exceptions should have a severe level, all operations concerning files should have a log
- × We suggest to use text file to record messages,
- × There is the possibility to use HashMaps to handle the messages
- × We suggest to use command words to discuss between the client and the server

#### Minimal specifications of the **client**:

- × The client will be able to connect to the server through socket connections
- × The client should be able to register a user and remember the last registered user
- × The client will connect to the server and retrieve the messages of a given user
- × The client will be able to send a message to an already registered user
- × The client must have access to the full list of registered users
- × The client must be able to write logs:
  - × On a file
  - × The history must be kept (one file per execution of the application)
  - × 3 levels of log (info, warning, severe) should be handled

The features described hereafter are examples of possible **add-on** that could be developed according to the progress of your project:

- × Possibility to send messages to more than one registered user at the same time
- × Possibility to send pictures
- × Possibility to have an admin page to setup the parameters (and save them in the config file)
- × Your imagination

***This project will be shared with the lesson of David Russo and will be one of the bases of the evaluation at the end of the module.***

## Deliverables:

- × **2 running applications** in production state.
  - × Commented code
  - × The 2 applications must behave properly without bugs
  - × **No useless code or libraries should remain**
  - × **Copy paste from the web is not an option**
- × All the features described under « work to do » must be implemented.
  - × If all the functionalities are developed properly and coherently the project will be evaluated with a 5.
  - × Adding new functionalities will increase the evaluation to reach a 6.
  - × Use Java with Eclipse, provide a .jar for both the client and the server.
- × Provide technical documentation of your work.
- × Provide a user manual for the client application (short but clear, snapshots welcome).
- × Provide an administrator manual for the server (including the installation of the server)
- × **1 log book** per person that will help us to understand your project sequence events, the difficulties you had, with a half day resolution.
- × **How to prepare the presentation of your project**
  - × Documentation (powerpoint or/and schemes or/and mindmanager documents or/and doc defined as equivalent by the professors)
  - × Presentation of the project by the 2 students. During this presentation the professors should be able to clearly determine the contribution of each student.

## Organisation

### Details

---

- 2 Java programs
- Beware the copy paste! a check of the code against plagia will be performed, plagia gets an evaluation of 1.
- The code should be developed by 2 people at maximum.

### Technical aspects

---

#### Implementation / Development

- Programming language : Java with Eclipse
- The programs must run with on windows 8 with the JVM 1.70\_25 for the **server application** and for the **client application**

### Tips

---

Start directly with an excellent code writing (comments, exceptions, naming, etc...) instead of trying to repair things at the end of the project.

Respect the development standards and the concepts seen in the lessons

We do prefer clean code and functions without bugs than buggy half backed solutions. The server **must** be coded to insure a 24/7 exploitation.

### Organization:

---

The project will be delivered for June 6th 2014. The presentation will be scheduled on January 6th.

As feed back of your work we would like a zip file containing:

- The complete source code with a description on how to recompile it on an Eclipse environment using Java 1.7
- The **log book** that will allow us to understand the tasks you've carried out, the problems you encountered, with a half day resolution.