

# Course project

Kim Hammar

7 dec 2015

## Revision History

Revision	Date	Author(s)	Description
0.1	07.12.15	KH	First draft

## Contents

<b>1</b>	<b>Abstract</b>	<b>3</b>
<b>2</b>	<b>Task specification</b>	<b>4</b>
2.1	Sub-assignment 1. A Single-User Information System for NOG	4
2.2	Sub-assignment 2. A Multi-User Information System for NOG	4
2.3	Sub-assignment 3. Chat Rooms for NOG . . . . .	4
<b>3</b>	<b>Platform</b>	<b>5</b>
<b>4</b>	<b>Software and technologies used</b>	<b>5</b>
<b>5</b>	<b>The application</b>	<b>6</b>
5.1	Functionality . . . . .	6
5.2	Protocols used . . . . .	6
5.3	GUI . . . . .	6
5.4	Architecture . . . . .	6
<b>6</b>	<b>Benchmarks</b>	<b>6</b>
<b>7</b>	<b>Documentation</b>	<b>6</b>

## **1 Abstract**

Course project in a course on Network programming in Java [1], carried out at Royal Institute of Technology, Stockholm.

## 2 Task specification

*You are “hired” by JEM inc (Java Enterprise Microsystems Inc.) to design and develop the distributed application software (clients and servers) for the NOG (Nordic Olympic Games) event.*

The NOG information system should allow storing, retrieving and updating personal information about NOG participants. The system should also be able to provide statistical information about participants. The system is to be developed in two version(1) a single-user version; (2) a multi-user version. You should also develop a NOG virtual meeting place. The NOG virtual meeting place is Internet based software which offers remotely located users to communicate and share information represented as textual, image or audio files.

### 2.1 Sub-assignment 1. A Single-User Information System for NOG

Develop a distributed application in Java that allows storing, retrieving and updating information about participants of NOG. The application should consist of a client with a user interface and a server. In this assignment we assume a **single user** semantics for the application, i.e It's not required to support coherency of multiple copies of participant records which may be cached by multiple client at the same time.

### 2.2 Sub-assignment 2. A Multi-User Information System for NOG

Develop a multi-user application, similar to the solution developed in sub-assignment 1. In this version a multi-user semantic is required. Many users can fetch the participants-data at the same time and when one user updates his local-cache of the data, the change need to be replicated among all clients connected in order to prevent them from using stale data.

### 2.3 Sub-assignment 3. Chat Rooms for NOG

Develop a distributed “building of chat rooms”.

### 3 Platform

The platform used for the development-process, benchmarks and tests is a computer running Xubuntu 14.04 LTS.

Java version: 1.7.0\_79 (OpenJDK version 7 update 79).

### 4 Software and technologies used

- Java Remote Method Invocation (java.\* package)
- Java Persistence API (java.\* package)
- JDBC (java.\* package)
- Java Swing (java.\* package)
- Java Socket (java.net package)
- PostgreSQL 9.3.9

No. threads	No. requests	Throughput	(KB/sec)
1Thread	100	4.946087644673063	87.81720651152439
2Threads	100	9.870693909781858	175.25301364623434
4Threads	100	19.68503937007874	349.50556717519686
10Threads	100	49.18839153959666	873.3341275209051
TOTAL	400	10.672358591248667	189.48647612059767

Table 2: My caption

## 5 The application

### 5.1 Functionality

### 5.2 Protocols used

### 5.3 GUI

### 5.4 Architecture

## 6 Benchmarks

## 7 Documentation

### References

- [1] Royal Institute of Technology. Network programming in java. <https://www.kth.se/social/course/ID2212/>, 2015. [Online; accessed 7-Dec-2015].