Assignment 2: A Room Booking System

Updated: 1 April 2018

Updated: 2 April 2018

Problem Specification:

With web services or distributed objects **(NB see notes below on allowed programming languages)** you should develop a client-server application implementing a room booking system with the following specification:

1. The client and server must be capable of running on 2 distinct machines connected by a network.
2. The server should manage a list of rooms for a one week period. The rooms are specified in a file (along with each room's capacity). For example

Room	Capacity
L221	50
XG14	35
T101	400
CG04	40

3. The server should provide the following functionality to a client:
    - Provide a list of the rooms it manages and their capacity.
    - Provide the timetable for a specified room for the full week.
    - Check the availability of a room given a specified day and time.
    - Book a room for a specified day and time.
4. The server should be able to process requests from many clients concurrently.
5. The client should be able to:
    - Display the timetable for a specified room for the full week.
    - Check the availability of a room given a specified day and time.
    - Allow a user to book a room for a specified day and time.
    - Be able to run in a test mode where many client requests are automatically generated with random delays between them.
    - Deal with an unreliable server or network.

Hint: you could use RMI to implement a basic room-booking service.

Instructions & Deadline:

1. The assignment is a **two** person project, worth 15% of the overall module mark. Any implimentation languages allowed. You can use your choice of support libraries or frameworks but you must pay attention to all application requirements defined above. (It's up to you if you want to do it individually, but marking will be the same).
2. A project demo will be required in our Monday lecture in Week 12, 15th April **(Demo will be graded - 5 marks)**
3. Please also write a short (5 page max) report with: **(Report will be graded - 5 marks)**
    - Design documentation which may get you some partial credit. Make sure to describe your concurrency strategies and client failure handling.
    - A plot/figure and a description of a simple concurrency or scalability study you did that shows how the system performs with many clients/requests (1 page only).
4. You have to submit all your source files and documentation as well as any peculiarities there may be with compiling or running it.**Source will be graded - 5 marks**
5. Please mail the above to me with a completed plagiarism declaration form which is signed by both of you by Friday 12th April 2019 **(Date confirmed)** .

A Word on Continuious Assessment

1. A resit is available for all components of this module.
2. Students who fail the module on the first sitting must resit each component (CA or exam) that they failed. In particular, students who fail the CA must resit the CA.
3. A student may not resit a component that has been passed and their marks for the component are carried forward.
4. Marks for failed components will be set to zero in ITS before the resit.

Good Luck!