

## Problem Description

The objective of this lab assignment is to gain practical experience in building a distributed software system using Java Remote Method Invocation (RMI). A distributed software system is a model where components on networked computers interact by exchanging messages to achieve a common goal. In this lab, you will enhance the Student Registration System from Lab 2 by transforming it into a distributed system using Java RMI.

You can use the RMI code sample on Canvas that was demoed during lecture or the following sample project to learn more about Java RMI

[Java RMI Demo](#)

## Task

Your task is to turn the existing student registration system from Lab 2, including the required modifications, which are the logging, overbooking, and course-conflict checking components, into a distributed system. You will be provided with the initial framework to help you develop a functional system.

## Architecture of the Distributed System

Refer to Figure 1 for the architecture diagram of the current framework.

The system includes the following components:

- Student.java: Defines the Student class within the system.
- Course.java: Defines the Course class within the system.
- Courses.txt: A text file containing a list of courses.
- Students.txt: A text file containing a list of students.
- Database.java: This interface is designed for data access operations, specifically reading data from Courses.txt and Students.txt. Classes implementing the IActivity interface will utilize the implementations of Database.java at runtime to fetch necessary data.
- Client.java: This client-side component is responsible for interfacing with remote objects based on user input. It incorporates a command line interface (CLI) that allows users to invoke specific remote methods.
- IActivity.java: An interface that extends java.rmi.Remote, which includes the execute() method for remote execution. Implementations of this interface, such as ListStudents, encapsulate their specific logic — for instance, listing all students — within the execute() method to be invoked remotely via RMI.

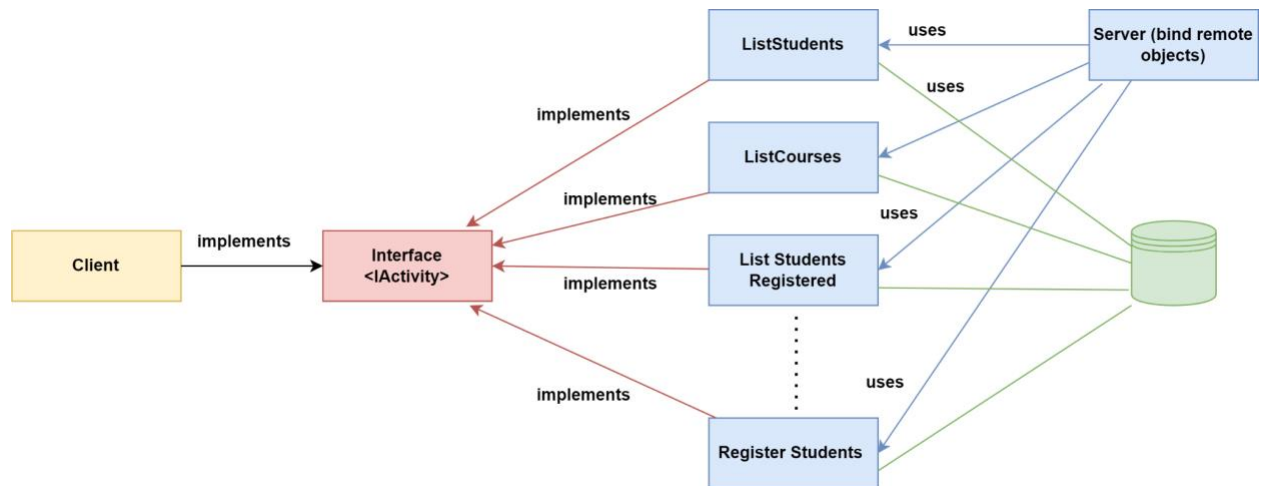


Figure 1. The architectural diagram of the distributed system

### Grading Criteria

- The correct implementation and operation of the solutions - we will test your solutions with our test data.
- The degree to which your solutions adhere to the distributed software system where possible to do so.