# Online Tutoring System

This system manages online tutoring by connecting students with teachers based on their subject of interest and availability. Clients in this system can be either students or teachers, and they register with a central server that helps maintain a directory of active users. The server computes matches between students and teachers and facilitates the initial connection over TCP. Additionally, the server periodically broadcasts system statistics to all clients via UDP, keeping them updated about the number of active students and teachers.

## Client-Side

1. Registration:  
- Both students and teachers connect to the central server via TCP to register.  
- At registration, clients provide: their **role**: “s” (student) or “t”(teacher) – you can generate this randomly; the **subject** they are teaching or learning (a string with a maximum of 32 characters); their **availability**: a list of integers representing the hours (0 to 23) during which they are available for tutoring.  
The server computes matches (if any) upon a new client's arrival.

2. Receiving Matches: if a match is found (a student and a teacher with the same subject and overlapping availability), the server sends their contact information to each other, and terminates the TCP connection.

3. Listening for Updates: Clients listen on UDP for periodic updates from the server about the number of active students and teachers in the system. These updates are displayed in real-time to keep users informed about system activity.

## Server-Side

1. Managing Clients: The server maintains a list of active students and teachers, including their roles, subjects, and availability. Upon a new client’s registration, the server checks for matches immediately. If a match is found, it facilitates the connection over TCP and removes both clients from the active list.

2. Broadcasting Updates: The server sends periodic updates (every 2 seconds) via UDP to all clients, providing the current count of active students and teachers.

## Examples

### Compatible Client Example:

Teacher A:  
- Role: teacher, Subject: Mathematics, Availability: [9, 10, 14, 15, 18].  
Student B:  
- Role: student, Subject: Mathematics, Availability: [10, 11, 14, 19].  
Teacher A and Student B are compatible because they share the same subject (Mathematics) and have overlapping availability at hour 10 and 14.

### Incompatible Client Example:

Teacher C:  
- Role: teacher, Subject: Physics, Availability: [8, 12, 16, 18].  
Student D:  
- Role: student, Subject: Mathematics, Availability: [9, 13, 17, 19].  
Teacher C and Student D are incompatible because their subjects do not match (Physics vs. Mathematics). No match is made, and the server keeps both clients in the active list for future matches.

## Grade 5 Requirement

For the grade 5:  
1. A client sends their subject to the server over TCP and a list of hours they are available (as a list of integers! don’t use strings for this).