Design Doc: Collaborative Online Judge System

# Overview

Collaborative Online Judge System (COJ) is a full-stack system supporting code room, instant communication, and collaborative code editing, compiling, execution, and result judgment. This document covers the details of the implementation of code room and instant communication from an engineering perspective.

# Major Use Cases

1. User can pick a problem and join its code room. They can join a specific code room, join a random code room, or start a new code room. The maximum number of users in a room is five.
2. When entering code room, if the user hasn’t logged in, then the user is supposed to enter his or her nickname or log in. If the user has logged in, then the web app will use his or her username.
3. After entering a code room, each user will be assigned with a specific color for coding identification.
4. User can chat with each other through chat box, the chat box allows user to send texts or images.

# Modules

**Navbar**

- Before login: Login/signup

- After login: Username/logout

**New Problem (only for admin)**

- Problem name

- Problem description (text editor)

- Difficulty (dropdown list)

**Problem List**

For each problem:

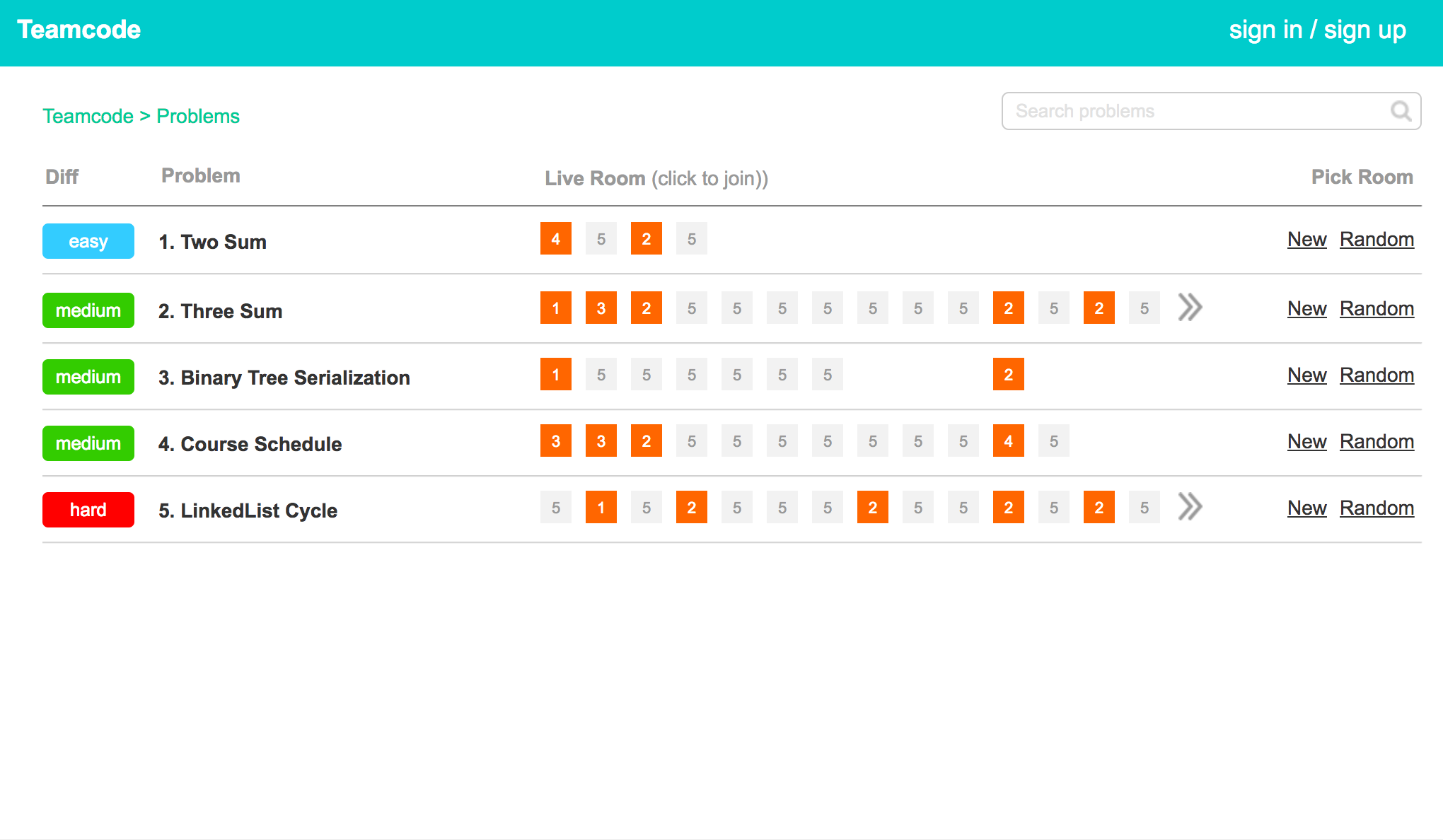
* Problem difficulty
* Problem subject
* Room list with number of users in each room
* New room link
* Random room link

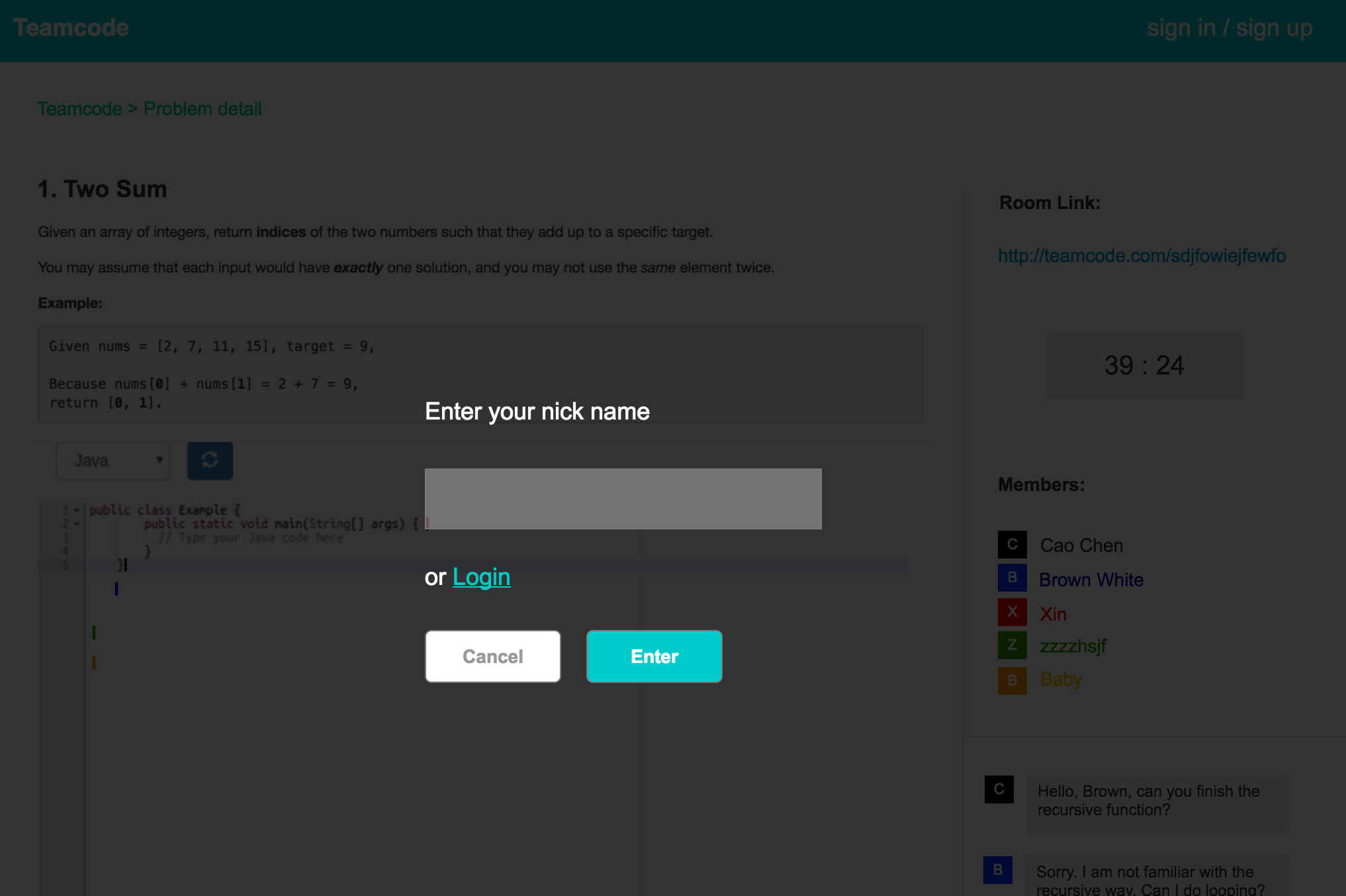
**Problem Detail**

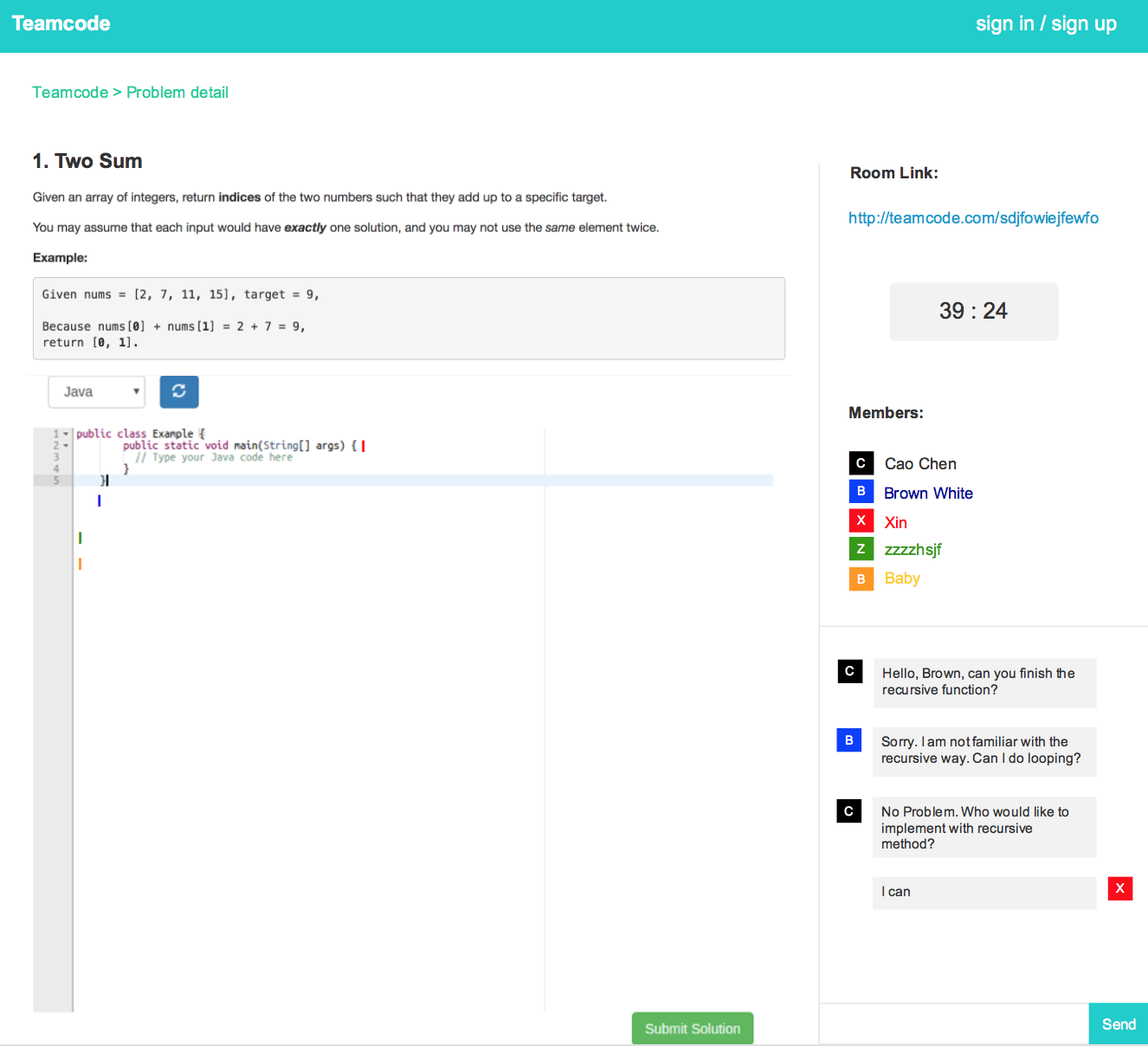
* Problem subject and description
* Code editor
* Code room link
* Timer
* Chat box

# UI Design

<http://qdnr0b.axshare.com/#g=1&p=problems>







# High-level Stack Program

|  |  |
| --- | --- |
| **Stack** | **Technologies** |
| Frontend – client | Angular.js, Socket.io |
| Frontend – server | Node.js, Socket.io, Redis, MongoDB, Nginx |
| Backend - executor | Nginx, Flask, Docker |

# Detailed Design

## Angular components

**Room entry**

* Component: room-entry
* Enter nickname or login when user hasn’t been logged in

**Room detail**

* Component: room-detail
* Including room link, timer, and team members

**Chat box**

* Component: chat-box

|  |  |
| --- | --- |
| **Functions at client side** | **Description** |
| initMessgeBox() | Initialize the message box, handshake socket.io, sessionid, input box |
| enterCodeRoom() | Bind to the click event of nickname submitting button, use socket to emit the nickname and socket.id to the server side |
| postNessage() | Check if the message box is empty, if not, emit the text, username, color, to the server side |
| displayMessage() | Display the new messages to the message box after clicking on the submit button |
| sessionid |  |

|  |  |
| --- | --- |
| **Functions at server side** | **Description** |
| roomid | Identify a code room |
| participant | List of users who can have access this session. |
| useridToRoomid | Map user id to to room id |
| collaboration | Sessionid, participant(socketid, color) |
| Listen to postMessage socket event | Loop through the participant and emit new messages to each user |

|  |
| --- |
|  |