COMP 504: Graduate Object-Oriented Programming and Design

Lecture 25: Chat App API Design

Mack Joyner (mjoyner@rice.edu)

https://www.clear.rice.edu/comp504

Announcements & Reminders

Hw #4 due today at 11:59pm

Chat App API Design

Chat App API Design

- Create all the tasks that are needed to meet the API design
  - Start with use cases
  - Assign tasks with deadlines for when those tasks should be complete
  - Everyone should have tasks assigned to them.

- Determine how you will keep track of task status
  - What is complete, in progress (ETA), not started
  - Are there status update meetings outside of class?

- Determine how to most efficiently work as a group
  - How do you get code/documents from one team member to the next
  - Do you need to have group work meetings outside of class?

Use Cases

- A use case is an example of how a user might interact with an application

- Design and develop an application by collecting as many use cases as possible

- Use case set informs developer what interfaces are needed to build system

UML Diagrams

- Reflect possible use cases in the system

- Use cases drive interface
  - Interface does not constrain use cases
  - Construct use cases before designing system

- Explain how your system works
  - Customer uses UMLs to understand interfaces within system

Chat Rooms

- Chat rooms allow people with shared interests to communicate
  - send and receive messages
  - restrictions on who can join the chat room (imposed by owner)
  - users can create chatrooms (owner)
- Each user is simultaneously running their own instance of a chat room
- User can chat with multiple people in a room (joined room)
- User can chat with multiple people in different rooms (joined multiple rooms)

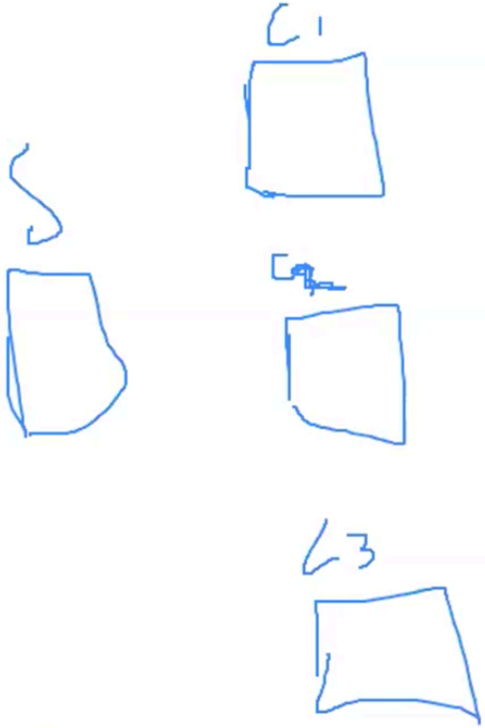
Use Cases: Creating a Chat Room

Admin:
Determines who joins the room
Removes someone from the room

Mute, Ban

Approval process for Admin
Delete Room   Delete Posts
Rights

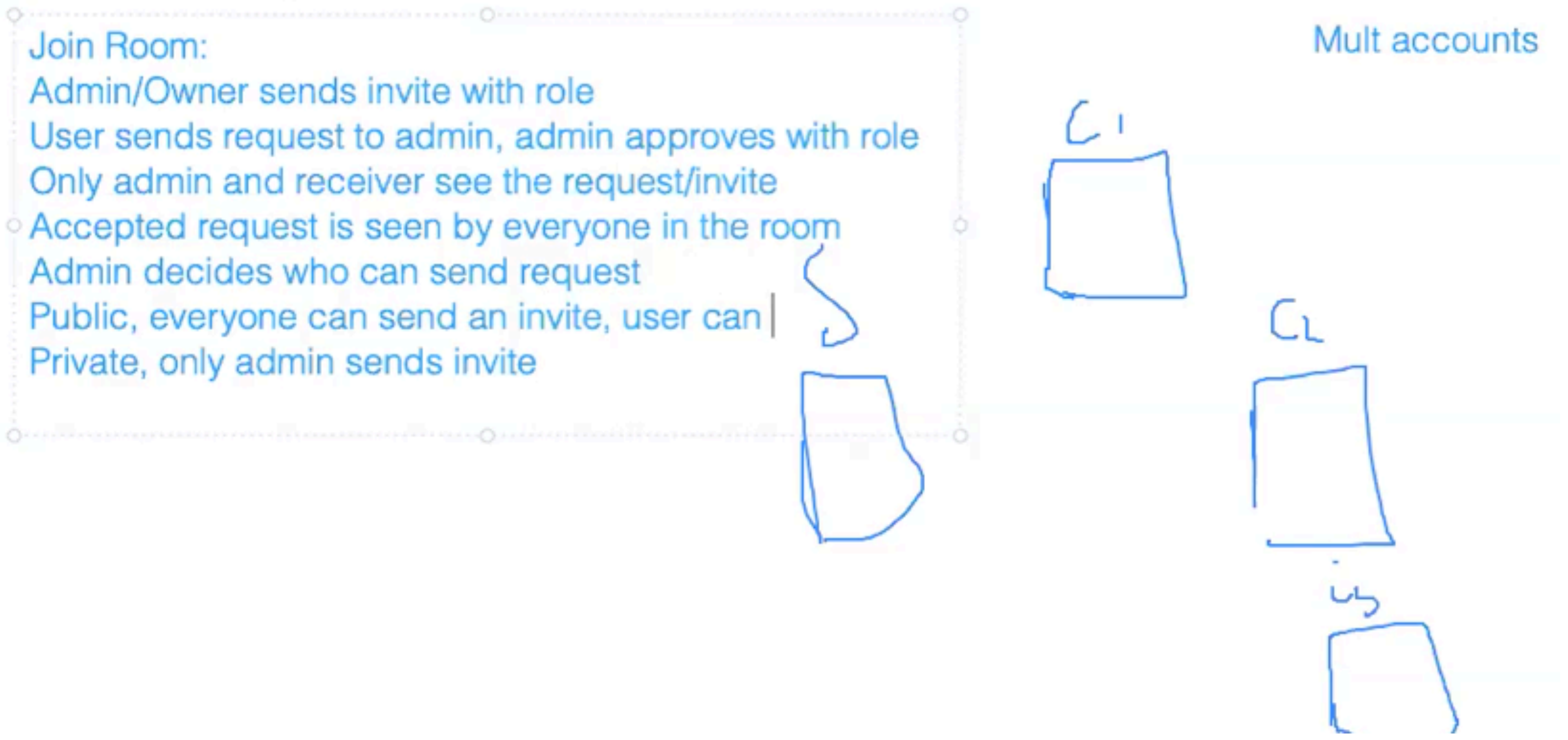
Mult accounts



Create:
Everyone (admin is creator)
Quota for Rooms
Owner initially added

Private, public
Contacts, General Chat (Lobby) - all rooms/public, Name room, topics - games (rooms, users)

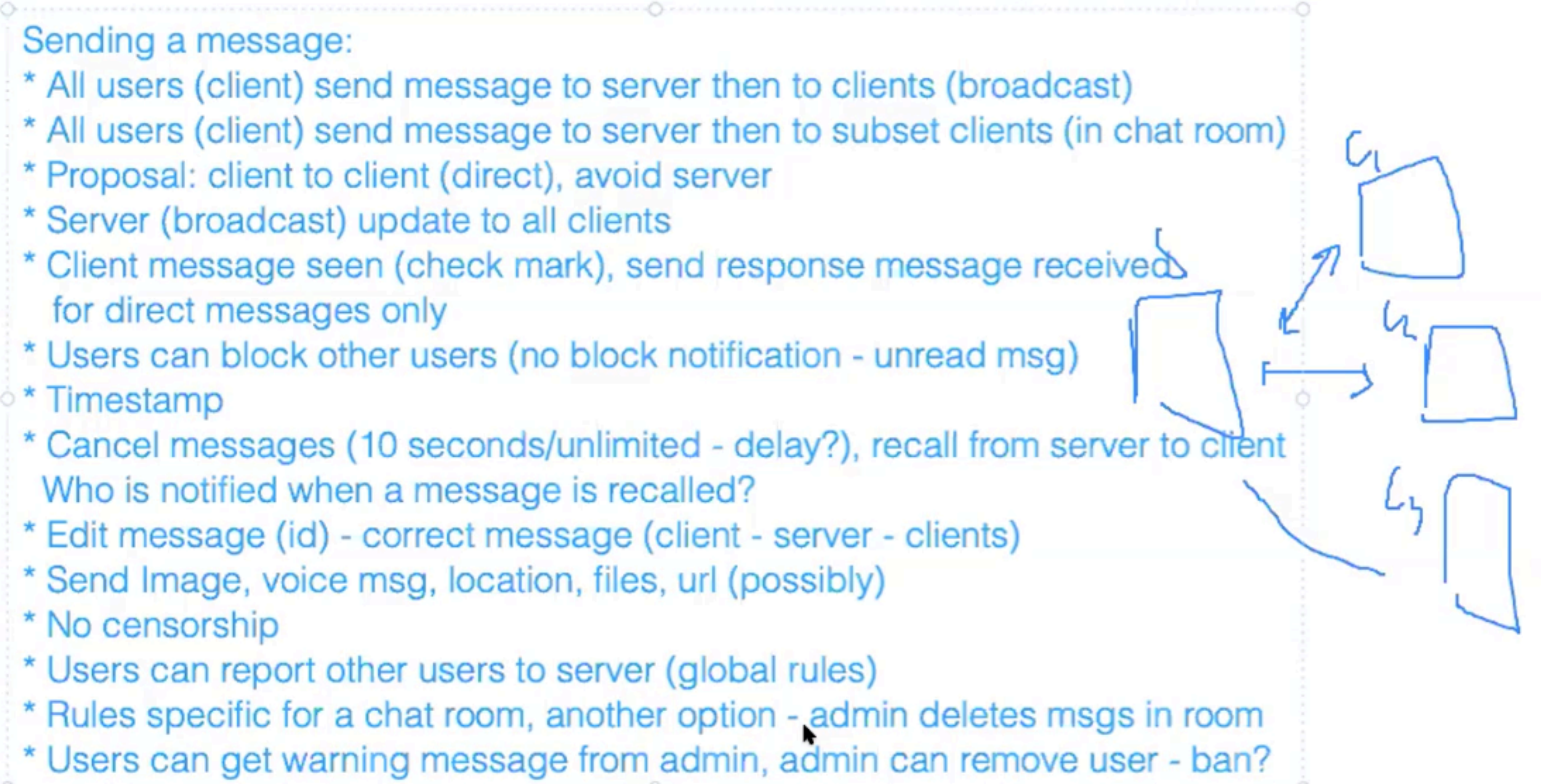
Use Cases: Joining a Chat Room



Join Room:
Admin/Owner sends invite with role
User sends request to admin, admin approves with role
Only admin and receiver see the request/invite
Accepted request is seen by everyone in the room
Admin decides who can send request
Public, everyone can send an invite, user can
Private, only admin sends invite

Mult accounts

Use Cases: Sending a Chat Room Message



Sending a message:

* All users (client) send message to server then to clients (broadcast)
* All users (client) send message to server then to subset clients (in chat room)
* Proposal: client to client (direct), avoid server
* Server (broadcast) update to all clients
* Client message seen (check mark), send response message received
  for direct messages only
* Users can block other users (no block notification - unread msg)
* Timestamp
* Cancel messages (10 seconds/unlimited - delay?), recall from server to client
  Who is notified when a message is recalled?
* Edit message (id) - correct message (client - server - clients)
* Send Image, voice msg, location, files, url (possibly)
* No censorship
* Users can report other users to server (global rules)
* Rules specific for a chat room, another option - admin deletes msgs in room
* Users can get warning message from admin, admin can remove user - ban?

Use Cases: Leaving a Chat Room

Chap App API Design

- Submit team name and agreed upon role for each team member (team lead, tech lead, doc lead, developer) by today at 10:40am.

- Project team lead will create GitHub classroom repo with team name.

  Other team members should join their team repo.

  Github classroom (Chat App): https://classroom.github.com/g/0_Xu4mWj

- Commit design API document(s) with design decisions, interfaces (can be picture), use cases discussed during group meeting