



# CHATAPP

GHOSTBUSTERS



# Team Members



**Kexi Dang**  
**Project Lead**



**Samantha Gilmore**  
**Tech Lead**



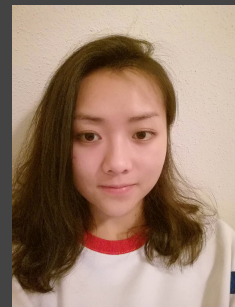
**Siri Manjunath**  
**Doc Lead**



**Po-Kai Chang**  
**Developer**



**Calvin Liu**  
**Developer**



**Serena Yang**  
**Developer**



# Outline

- Use Cases
- Design Patterns
- UML Diagrams
- Backend Overview
- Design FAQs
- Demo

# Welcome to Chatapp

You can have a lot of fun here, please register first.

## Register

register

# Welcome to the chat room

## Your rooms

test into room owner  
into room

test

## Rooms you can join

[Your Profile](#)

[Create room](#)

[Logout](#)

\* means you own this room

# Key Use Cases



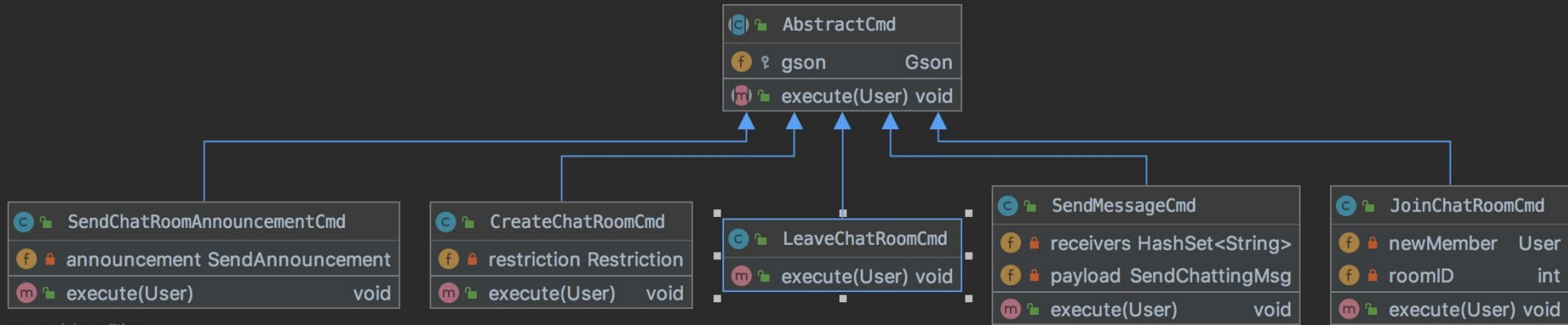
- Owner leaves the chatroom when there are no other users present.
  - If there are still members, second user to join becomes the owner
  - Else, the chatroom is deleted in this case.
- Do not store user info. When a user leaves the web page or the websocket times out, the user is deleted and chatroom information is not maintained.
- Announcements:
  - Owner Message
  - User Leaves
  - New Owner



# Design Patterns

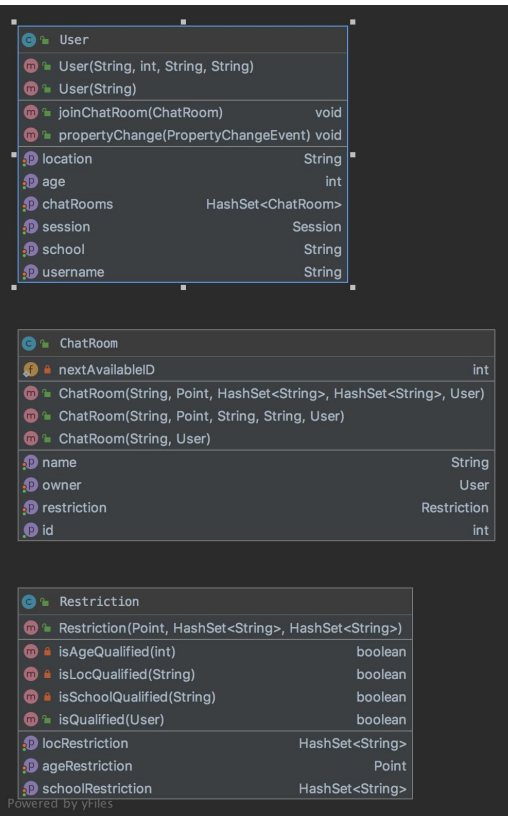
- Command
- Union
- Observer
- Singleton

# UML Diagrams: Cmd

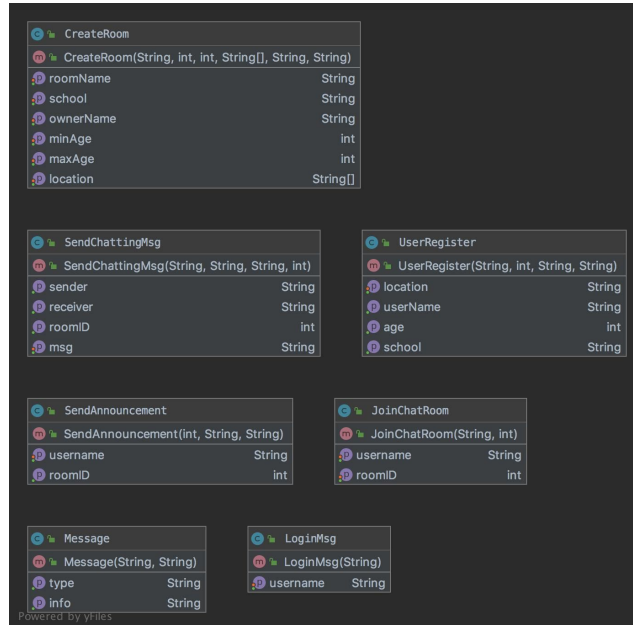




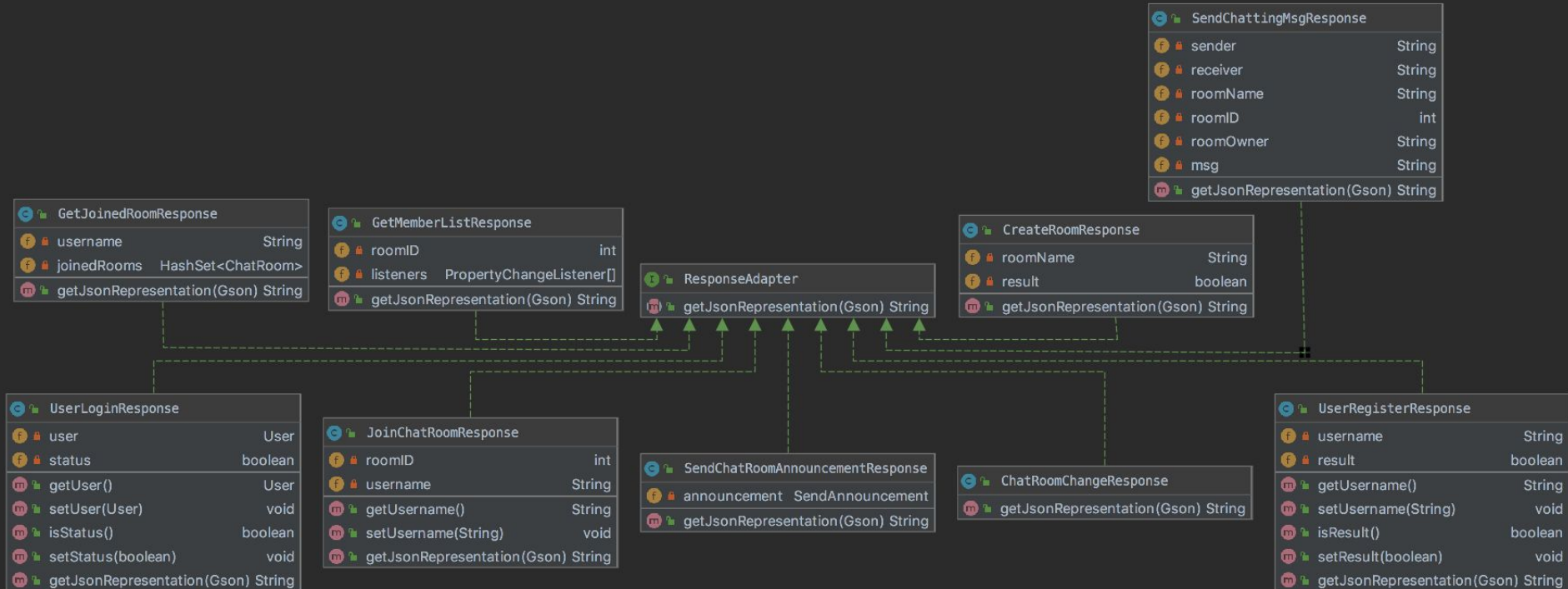
# UML Diagrams: Model



# UML Diagrams: Request



# UML Diagrams: Response





# Backend

- Dispatcher
  - Singleton
  - All users listen to the “user” property
  - All members in a chat room listen to a property named as its roomId
- Command
  - Pass commands to all listeners through the Dispatcher



# Backend

## HTTP

- Register
- Get the member list of a chat room
- Get chat room list

## WebSocket

- Login
- Send and receive chatting message
- Send and receive chat room announcement
- Notify clients when room/member changes



# FAQ

1. How do we distinguish between WS msg type? @Calvin
2. How do we use the dispatcher in the backend? @Calvin
3. How do you maintain WS connections across pages? @kexi
4. Why do we use both HTTP and WS? @kexi
5. How does the backend identify users and Chat rooms? @kexi
6. How to keep the chat history/announcement history? @pokai



# How do we distinguish between WS msg type?

## General form of WS msg

```
{type: Msg Type,  
  info: Payload}
```

## Send Chat Room Announcement

```
{type: "announcement",  
  info: { username: "XX",  
          announcement: "YY",  
          roomID: 1}}
```

## Send Chat Msg

```
{ type: "chat",  
  info: { sender:xxx, receiver:xx,  
          roomID:xx, msg:xx}}
```



**How do we use the dispatcher in the backend?**





## How do you maintain WS connections across pages?

- `<script type="text/html"><div>hello</div></script>` will not render in the page, but same coding as normal HTML.
- Websocket will be created in a separate file which will be inserted to the DOM when users register successfully and won't be replaced if url changes.

Done! You have your own front-end router.



## Why do we use both HTTP and WS?

- Websocket is used in full-duplex communication to replace with polling.
- Websocket is expensive than normal HTTP. (HTTP 101, handshake, header...)
- Getting data like room list uses HTTP request. Chat/Announcements uses WS.
- Receive WS messages first, then have a HTTP request.



# How does the backend identify users and Chat rooms?

Users are distinguished by the username; Chat rooms are distinguished by the id.

- When the user login to our app, front-end will send a message of the username to back-end , which let the session of websocket maps to a username. Username is the unique, duplicate username cannot be registered successfully.
- `GET /joined_rooms/:username` or `{type: "enter_room", info: {username: 'xxx', roomId: 123}}`
- Chat rooms have their unique id, back-end send the room data with id.
- `GET /member_list/:roomId` or `{type: "enter_room", info: {username: 'xxx', roomId: 123}}`



## How to keep the chat history/announcement history?

Information saved in frontend



**What happens if the owner leaves a chatroom?**



**Why is this the optimal design?**

# Demo

## Welcome to the chat room

### Your rooms

test into room owner  
into room

test

### Rooms you can join

[Your Profile](#)

[Create room](#)

[Logout](#)

\* means you own this room

<https://chatapp-team-ghostbusters.herokuapp.com/>