TeaTalk

Team Team
Jiaxin Wu, Lee-Hsun Hsieh, Shuai He,
Soham Pajwani, Xiangnan Chen, Yi Zhang

Meet our Team Team Team Members

Team Lead

- *Lee Hsun Hsieh*

Tech Lead

- *Xiangnan Chen*

Documentation Lead

- *Yi Zhang*

Developers

- *Soham Pajwani*
- *Shuai He*
- *Jiaxin Wu*

Design Choices

Factory design pattern

- **Create chat room**
- **Create message**

Observer design pattern

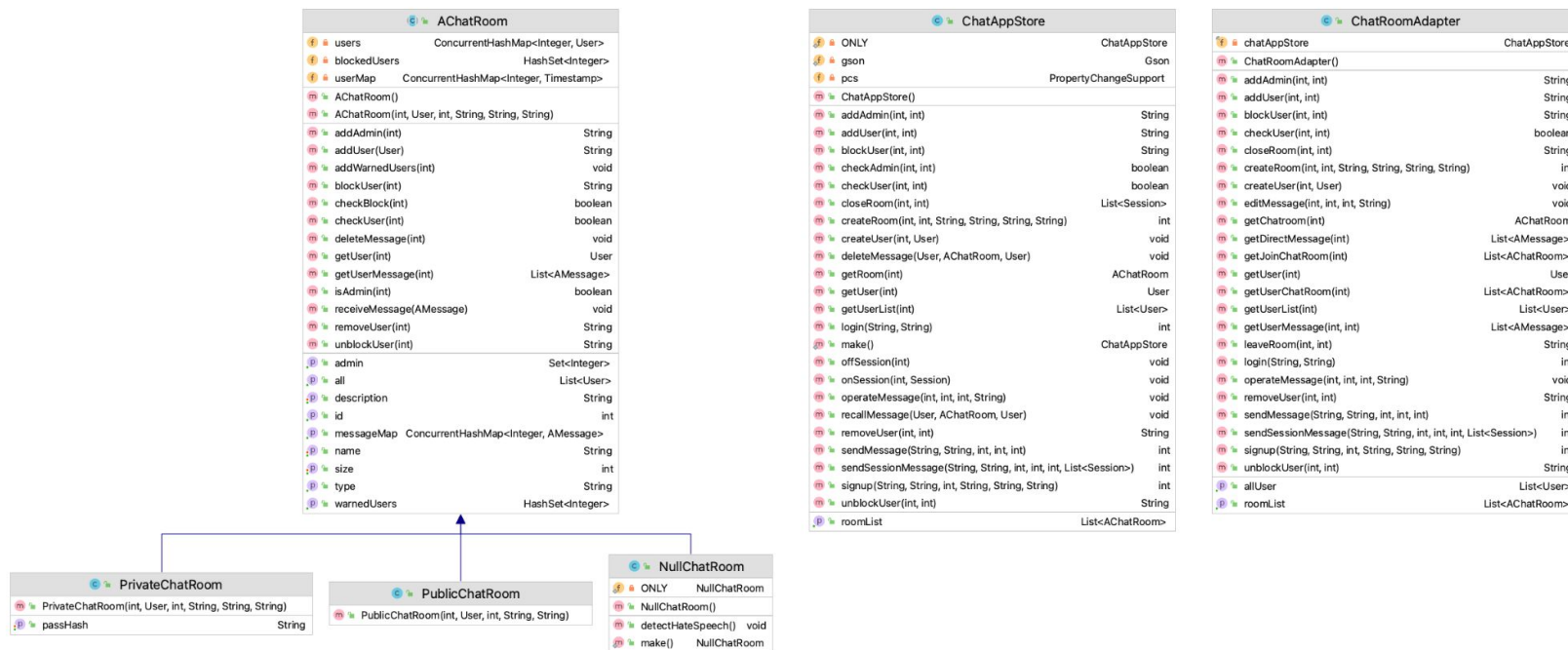- **Chat room update information as observers**

Singleton design pattern
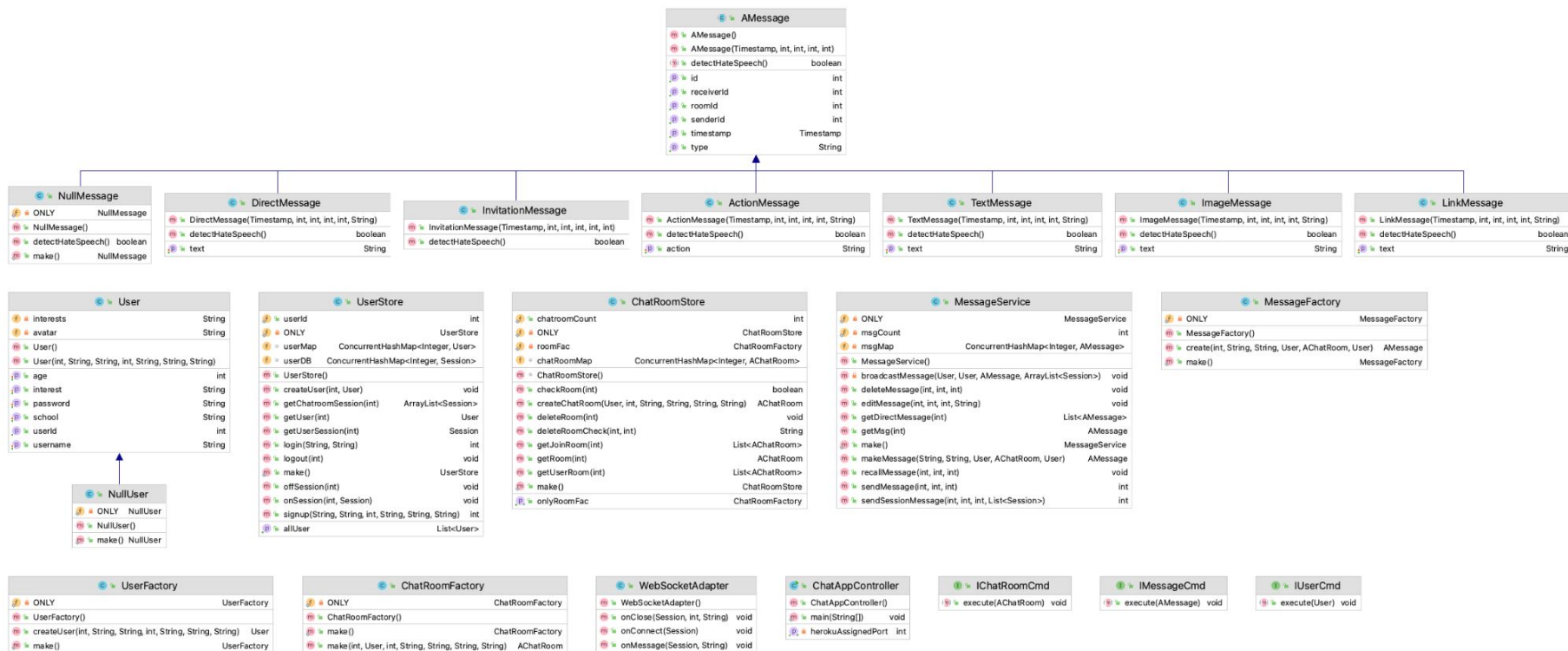
- **Factory and Store Objects**

Command design pattern

- **Update user/chat room/message info**

UML Diagram

UML Diagram Cont.

AMessage
- AMessage()
- AMessage(Timestamp, int, int, int, int)
- detectHateSpeech() — boolean
- id — int
- receiverId — int
- roomId — int
- senderId — int
- timestamp — Timestamp
- type — String

NullMessage
- ONLY — NullMessage
- NullMessage()
- detectHateSpeech() boolean
- make() — NullMessage

DirectMessage
- DirectMessage(Timestamp, int, int, int, int, String)
- detectHateSpeech() — boolean
- text — String

InvitationMessage
- InvitationMessage(Timestamp, int, int, int, int, int)
- detectHateSpeech() — boolean

ActionMessage
- ActionMessage(Timestamp, int, int, int, int, String)
- detectHateSpeech() — boolean
- action — String

TextMessage
- TextMessage(Timestamp, int, int, int, int, String)
- detectHateSpeech() — boolean
- text — String

ImageMessage
- ImageMessage(Timestamp, int, int, int, int, String)
- detectHateSpeech() — boolean
- text — String

LinkMessage
- LinkMessage(Timestamp, int, int, int, int, String)
- detectHateSpeech() — boolean
- text — String

User
- interests — String
- avatar — String
- User()
- User(int, String, String, int, String, String, String)
- age — int
- interest — String
- password — String
- school — String
- userId — int
- username — String

NullUser
- ONLY — NullUser
- NullUser()
- make() — NullUser

UserStore
- userId — int
- ONLY — UserStore
- userMap — ConcurrentHashMap<Integer, User>
- userDB — ConcurrentHashMap<Integer, Session>
- UserStore()
- createUser(int, User) — void
- getChatroomSession(int) — ArrayList<Session>
- getUser(int) — User
- getUserSession(int) — Session
- login(String, String) — int
- logout(int) — void
- make() — UserStore
- offSession(int) — void
- onSession(int, Session) — void
- signup(String, String, int, String, String, String) — int
- allUser — List<User>

ChatRoomStore
- chatroomCount — int
- ONLY — ChatRoomStore
- roomFac — ChatRoomFactory
- chatRoomMap — ConcurrentHashMap<Integer, AChatRoom>
- ChatRoomStore()
- checkRoom(int) — boolean
- createChatRoom(User, int, String, String, String, String) — AChatRoom
- deleteRoom(int) — void
- deleteRoomCheck(int, int) — String
- getJoinRoom(int) — List<AChatRoom>
- getRoom(int) — AChatRoom
- getUserRoom(int) — List<AChatRoom>
- make() — ChatRoomStore
- onlyRoomFac — ChatRoomFactory

MessageService
- ONLY — MessageService
- msgCount — int
- msgMap — ConcurrentHashMap<Integer, AMessage>
- MessageService()
- broadcastMessage(User, User, AMessage, ArrayList<Session>) — void
- deleteMessage(int, int, int) — void
- editMessage(int, int, int, String) — void
- getDirectMessage(int) — List<AMessage>
- getMsg(int) — AMessage
- make() — MessageService
- makeMessage(String, String, User, AChatRoom, User) — AMessage
- recallMessage(int, int, int) — void
- sendMessage(int, int, int) — int
- sendSessionMessage(int, int, int, List<Session>) — int

MessageFactory
- ONLY — MessageFactory
- MessageFactory()
- create(int, String, String, User, AChatRoom, User) — AMessage
- make() — MessageFactory

UserFactory
- ONLY — UserFactory
- UserFactory()
- createUser(int, String, String, int, String, String, String) — User
- make() — UserFactory

ChatRoomFactory
- ONLY — ChatRoomFactory
- ChatRoomFactory()
- make() — ChatRoomFactory
- make(int, User, int, String, String, String, String) — AChatRoom

WebSocketAdapter
- WebSocketAdapter()
- onClose(Session, int, String) — void
- onConnect(Session) — void
- onMessage(Session, String) — void

ChatAppController
- ChatAppController()
- main(String[]) — void
- herokuAssignedPort — int

IChatRoomCmd
- execute(AChatRoom) — void

IMessageCmd
- execute(AMessage) — void

IUserCmd
- execute(User) — void

REST API Design

GET Method:
- /downloadInfo/userlist
- /downloadInfo/roomlist
- /loadUserChatRoom

POST Method:
- /login
- /signup
- /leaveChatroom
- /joinChatroom
- /uploadInfo
- /operationToChatroom
- /operationToUser
- /operationToMsg

TeaTalk Demo

https://chatapp-api-team-team.herokuapp.com/login.html

Thank you!
Questions?