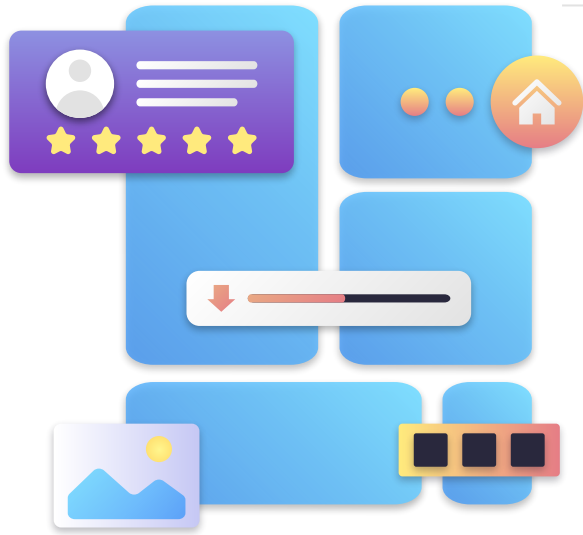


Communication Application

Group 7





Internal Chat System

Designed to handle messages amongst employees in a corporate environment

Messaging is group-based, logged and handled server side



Members

Marcos Barajas

Nadeem Mohammed



Emanuel Miranda-Mercado



Arshdeep Brar

Irtiza Khan



TABLE OF CONTENTS

/01 Basic Requirements



Functional Requirements
Other Requirements

/02 Schedule

/03 Application Demo

/03 Project Overview

Basic Requirements

- Two types of users: **system admins** and **employees**
- **System admins** can **view** chat logs; **create** new users; **delete** existing users
- **Server** will **receive data** from multiple **clients**; process; organize
- **Messages** be **saved** in a chat log accessible to system admin via server
- **Messages** will have **delivered** and **read** status
- **Username** and **password** created by **system admin**
- **Interface** will be **simple** and provide options according to the type of user

Project schedule

Messaging application project schedule																																																					
Group 7																																																					
1/24/2022 (Monday)					1																																																
					Week 9							Week 10							Week 11							Week 12							Week 13							Week 14							Week 15						
					21 Mar 2022							28 Mar 2022							4 Apr 2022							11 Apr 2022							18 Apr 2022							25 Apr 2022							2 May 2022						
					21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6		
					M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
TASK	START	END	DAYS	% DONE																																																	
Client meeting	Mon 2/28/22	Mon 2/28/22	0	100%																																																	
Meeting 5	Wed 3/02/22	Wed 3/02/22	0	100%																																																	
Client meeting	Wed 3/16/22	Wed 3/16/22	0	100%																																																	
Comdense classes	Wed 3/16/22	Wed 3/30/22	14	100%																																																	
Phase 3																																																					
Update requirements	Mon 4/04/22	Sun 4/10/22	6	100%																																																	
Update use cases	Mon 4/04/22	Sun 4/10/22	6	100%																																																	
Update classes	Mon 4/04/22	Mon 4/25/22	21	100%																																																	
Meeting 6	Tue 4/05/22	Tue 4/05/22	0	100%																																																	
Meeting 7	Wed 4/06/22	Wed 4/06/22	0	100%																																																	
Meeting 8	Tue 4/12/22	Tue 4/12/22	0	100%																																																	
Create draft GUIs	Tue 4/12/22	Tue 4/19/22	7	100%																																																	
Meeting 9	Wed 4/13/22	Wed 4/13/22	0	100%																																																	
Meeting 10	Thu 4/14/22	Thu 4/14/22	0	100%																																																	
Finalize GUIs	Tue 4/19/22	Wed 5/04/22	15	80%																																																	
Meeting 11	Fri 4/22/22	Fri 4/22/22	0	100%																																																	
Meeting 12	Sat 4/23/22	Sat 4/23/22	0	100%																																																	
Application launch	Sat 4/23/22	Mon 5/02/22	9	80%																																																	
Meeting 13	Sun 4/24/22	Sun 4/24/22	0	100%																																																	
Meeting 14	Wed 4/27/22	Wed 4/27/22	0	100%																																																	
Meeting 15	Sat 4/30/22	Sat 4/30/22	0	100%																																																	
Presentation prep	Sun 5/01/22	Wed 5/04/22	3	90%																																																	
Project complete																																																					



Application Demo

Employee

Employee
<ul style="list-style-type: none">- username : string- password : string- isOnline: bool- isSystemAdmin: bool
<ul style="list-style-type: none">+ Employee()+ Employee(username: string, password: string)
<ul style="list-style-type: none">+ setUsername(username: string) : void+ setPassword(password: string): void+ setIsSystemAdmin(isSystemAdmin: bool): void+ setIsOnline(isOnline: bool): void
<ul style="list-style-type: none">+ getUsername() : string+ getPassword(): string+ getIsSystemAdmin(): bool+ getIsOnline(): bool
<ul style="list-style-type: none">+ validate(password: string) : bool

Employee
LIST

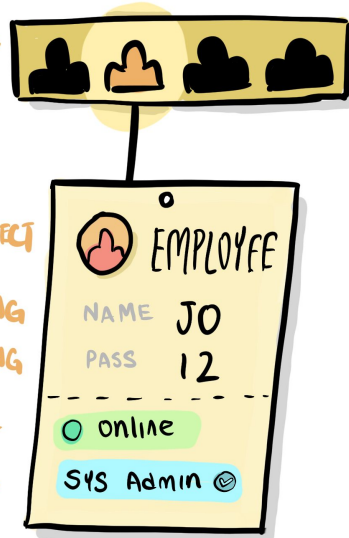
OBJECT

STRING

STRING

BOOL

BOOL

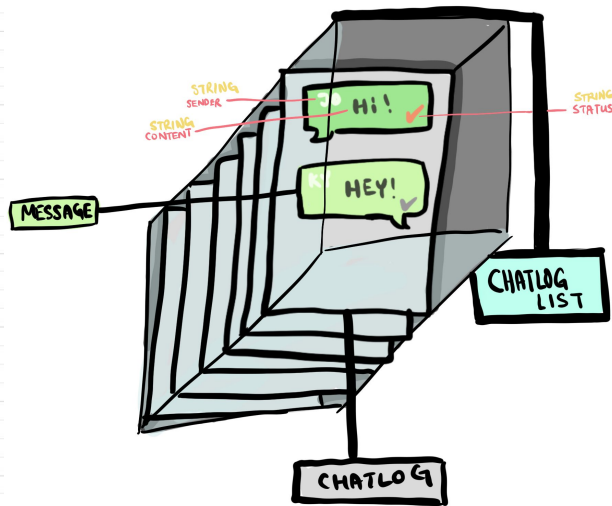


EmployeeList

EmployeeList
<ul style="list-style-type: none">- employees: Vector<Employee>
<ul style="list-style-type: none">+ EmployeeList()+ EmployeeList(employees: Vector<Employee>)
<ul style="list-style-type: none">+ setEmployees(Vector<Employee>) : void+ getEmployees() : Vector<Employee>
<ul style="list-style-type: none">+ remove(username: string) : void+ contains(username: string): bool+ get(username: string): Employee+ get(index: int): Employee

ChatLogList

ChatLog
<ul style="list-style-type: none">- name : string- messages : Vector<Message>- members: Vector<string>
<ul style="list-style-type: none">+ ChatLog()+ ChatLog(name: string, members: Vector<string>)+ ChatLog(name: string, members: Vector<string>, messages: Vector<Message>)+ setName(name: string) : void+ setMessages(messages: Vector<Message>): void+ setMembers(members: Vector<string>): void+ getName() : string+ getMessages(): Vector<Message>+ getMembers(): Vector<string>+ addMessage(message: Message): void+ toString(): string

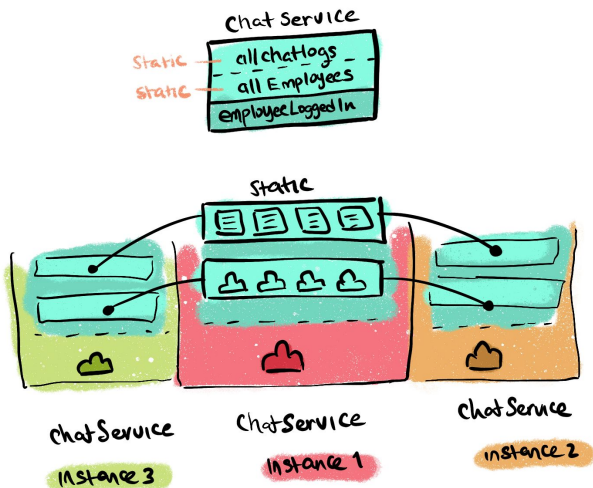


ChatLogList
<ul style="list-style-type: none">- chatLogs : Vector<ChatLog>
<ul style="list-style-type: none">+ ChatLogList()+ ChatLogList(chatLogs: Vector<ChatLog>)+ setChatLogs(chatLogs: Vector<ChatLog>) : void+ getChatLogs() : Vector<ChatLog>+ add(chatLog: ChatLog): void+ contains(name: string): bool+ remove(name: string): void+ get(name: string): ChatLog+ get(index: int): ChatLog+ getMemberChatLogs(username: string): ChatLogList

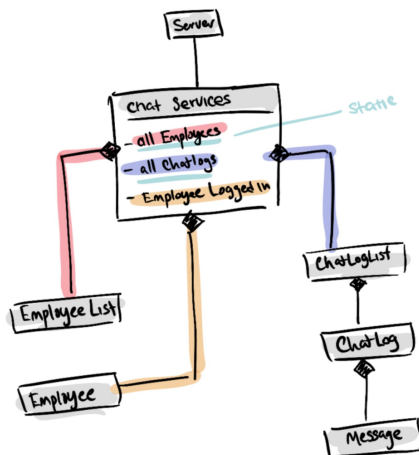
ChatLog

Chat Service

Static Member Variables



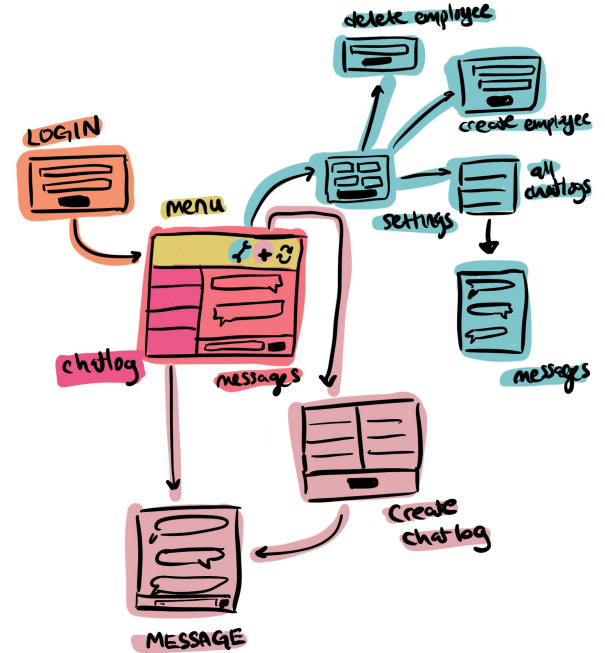
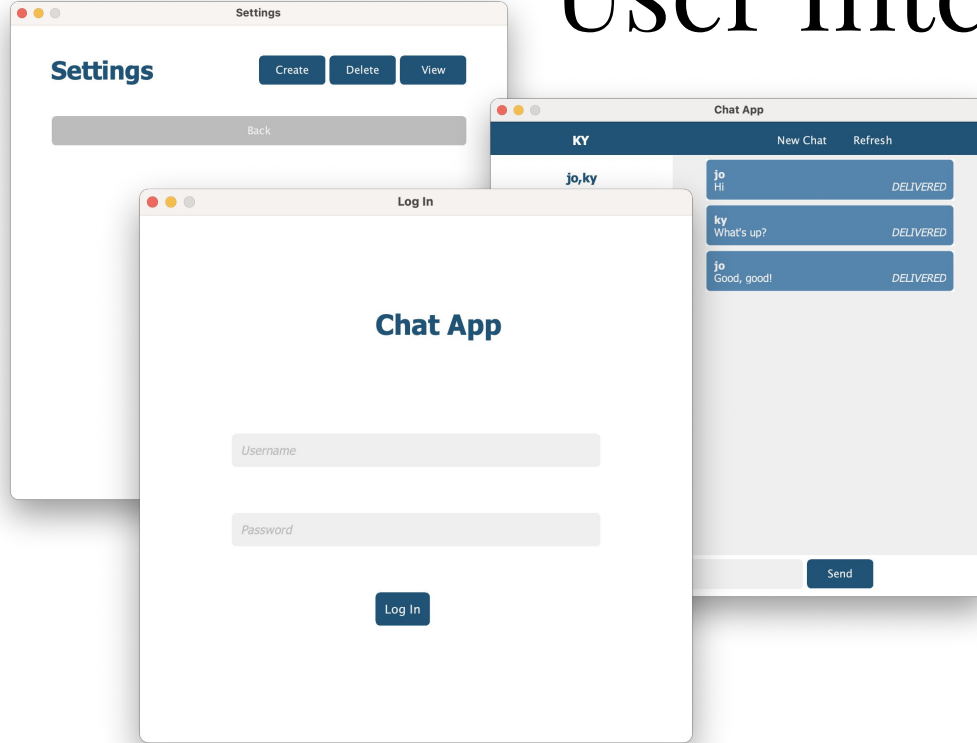
Rough Class Diagram



ChatService

- employeeLoggedIn: Employee
 - allChatLogs: ChatLogList
 - allEmployees: EmployeeList
-
- setEmployeeLoggedIn(employeeLoggedIn: Employee): void
 - setAllEmployees(allEmployees: EmployeeList): void
 - setAllChatLogs(allChatLogs: ChatLogList): void
 - getEmployeeLoggedIn(): Employee
 - getAllEmployees(): EmployeeList
 - getAllChatLogs(): ChatLogList
-
- getEmployeeChatLogList(): ChatLogList
 - getEmployeeChatLog(name: String): ChatLog
-
- getChatLog(name: String): ChatLog
 - getNewMessage(chatLogName: String): Message
 - getEmployee(username: String): Employee
 - login(username: String, password: String): boolean
 - logout(): void
 - createChatLog(name: String, members: Vector<String>): void
 - createEmployee(username: String, password: String, isSystemAdmin: boolean): void
 - deleteEmployee(username: String): void
 - sendMessage(chatLogName: String, message: Message): void
-
- loadChatService(): void
 - saveChatService(): void

User Interface



Client

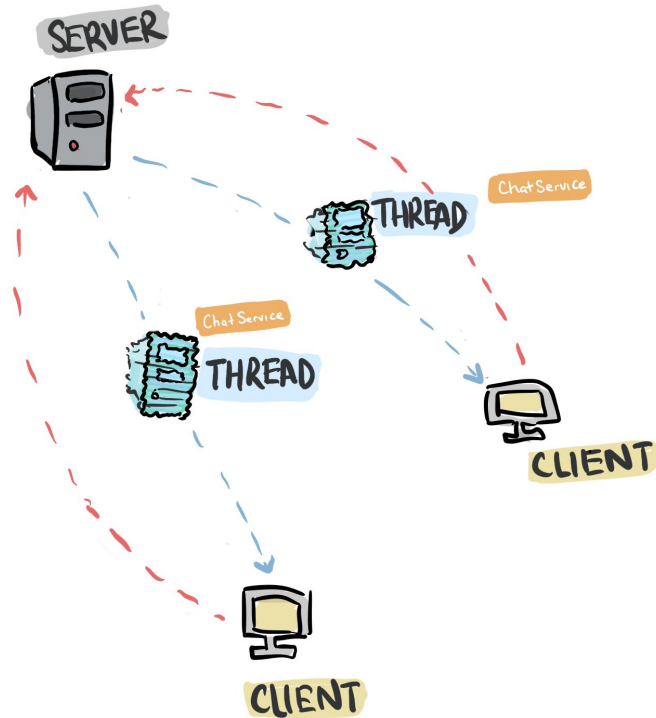
```
public Employee getEmployee(String username){  
    SocketMessage employeeMessage = new SocketMessage("getEmployee");  
    employeeMessage = setEmployeeLoggedInForSocketMessage(employeeMessage);  
    employeeMessage.getEmployee().setUsername(username);  
  
    send(employeeMessage);  
    SocketMessage returnEmployeeMessage = receive();  
  
    return returnEmployeeMessage.getEmployee();  
}
```

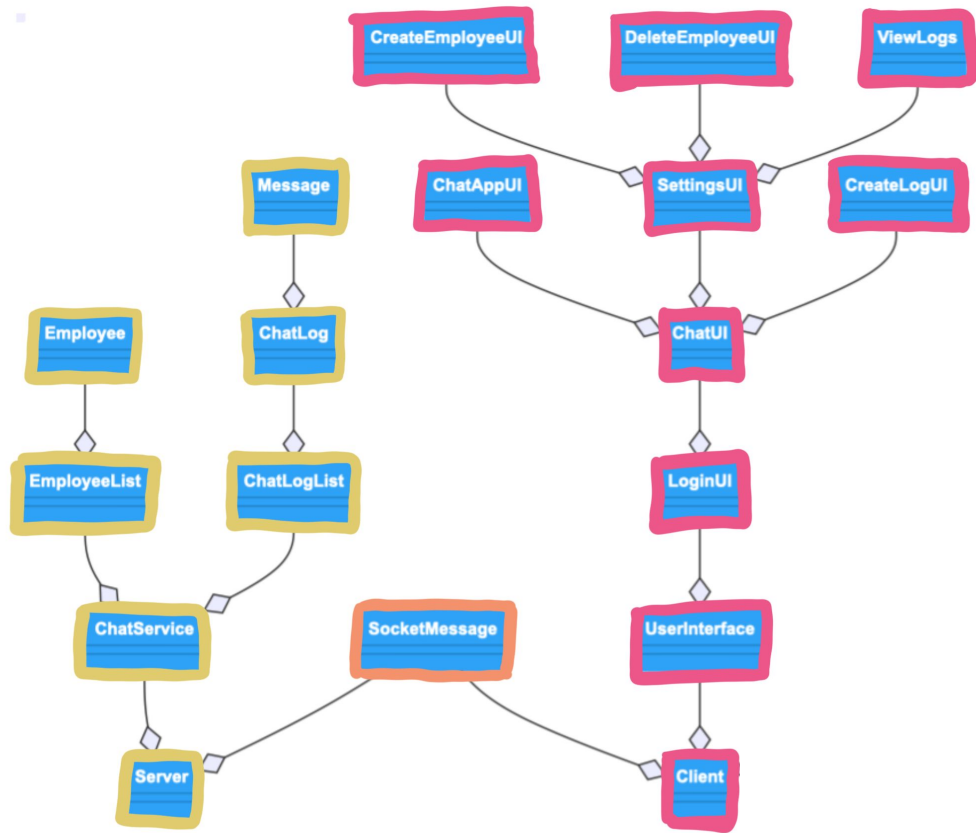
Server

```
private void getEmployee(SocketMessage socketMessage) {  
    SocketMessage returnEmployeeMessage = new SocketMessage("getEmployee");  
    setEmployeeLoggedIn(socketMessage);  
    String username = socketMessage.getEmployee().getUsername();  
  
    returnEmployeeMessage.setEmployee(chatService.getEmployee(username));  
    returnEmployeeMessage.setStatus("SUCCESS");  
  
    send(returnEmployeeMessage);  
}
```

ChatService

```
public Employee getEmployee(String username){  
    username = username.toLowerCase();  
    return ChatService.allEmployees.get(username);  
}
```





Frontend