

# Team Chicago

...

# Introduction

- Client-Server messenger system, using a database to store user information and message records
- Why did we choose this?
- In this presentation...
  - Design
  - Our process
  - Evaluations

# Functionality

## Base:

**Create** a new account. - High

**Login** to the server using a username and password. - High

**View** list of online user. - High

**Common** Chat Room - High

**Private** Chat Rooms - High

Chat History - High

## Extra:

Anonymous Login

Personalized UI

Translation

In the future: Chat encryption, Spell-checking

# Project Management

Regular Meetings for milestones.

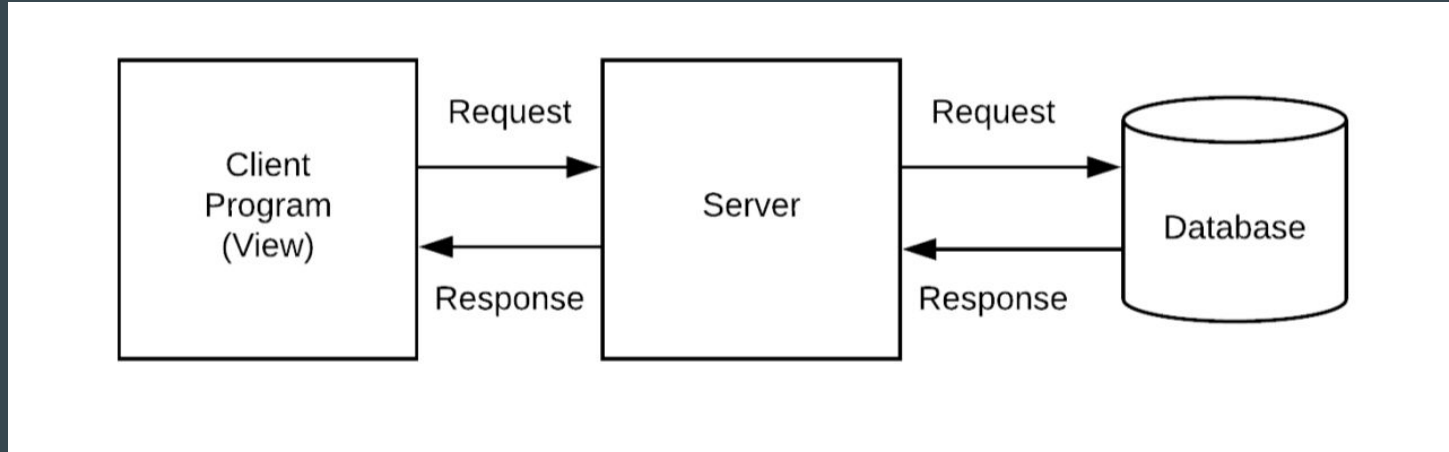
Daily Team Coding Sessions

Flexible Rotational Roles

Keep in touch: Messenger Group

Keep on track: Gantt Chart

# Architecture



Classes:

**Server:** Server, ServerThread, User, Message, Chat, GUI

**Client:** Client, GUI classes

# Architecture

## Protocol

Action	Request from Client	Response from Server	description
"sign-up"	(type, username, password)	(type, boolean_string:result, message)	a request from the client to create a new user profile
"sign-in"	(type, username, password, boolean_string:annon)	(type, boolean_string:result, message/anon username, server history...)	a request from the client to sign in to the server and begin using it.
"get-message"	(type, offset)	(type, message1, message2, ...)	a request from the client to receive the most recent list of public messages.
"send-message"	(type, content)	(type, result, message)	a request to send a message to the public chat.
"update-list"	none	(type, distinction, username)*	a message received whenever a user signs in or disconnects, telling the client to update its list of online users.
"create-group"	(type, Chat name, username1, username2, ...)	(type, Chat name, username1, username2, ...)	a request from the client to create a private chat group with the specified other users.
"send-private-message"	(type, chat id, content)	(type, chat id, content)	a request to send a message to a private group chat
"add-to-chat"	(type, chat id, username1, username2, ...)	(type, chat id, username1, username2, ...)	a request to add one or more members to an existing private group chat
"disconnect"	(type, username)	(type, distinction, username)*	a request to disconnect this user completely, response is actually an "update-list" message
"leave-chat"	(type, chat id, username)	(type, chat_id, username)*	a request to disconnect from a private group chat

Table 1: Protocol Description, response from server elements with a \*, indicate the action messages multiple users, and the user who initiated the action treats the message differently, as they receive their own name.

# Database

- Storage
- Client details
- Messages
- Database-Server-Client.

General ideology: client type something - send to server then transferred to the database for storage.

# GUI

- Server
- Sign up
- Sign in
- Chat room(private, public)
- Make group



# Testing and Evaluation

- Testing:
  - JUnit
  - Manual Testing
  - User Testing
- Evaluation
  - Final Product
    - Achieved core aims
    - Implemented a number of extra features
    - Room to improve ease of use in the future
  - Team
    - Met frequently
    - Everyone was kept up to date
    - Improvements

# Product Demo

- Connect to the Server
- Create a new account
- Login successfully
- Chat!
- Discover