

Cynergy

Cynergy is a real-time chat application that allows multiple users to communicate with each other. It also supports cross-platform application.

Supported Platforms

- Web
- Windows (7/8/8.1/10)
- Linux (Debian-Based)

Technologies Used

Cynergy is built with the following technologies:

Front-end	Back-end
Material Design	NodeJS
Bootstrap	ExpressJS
Jquery	SockJS

For the desktop version, **Cynergy** is built with **Electron**

Objectives

The following are the objectives of **Cynergy**.

- To provide a platform for users to communicate with different users.

Features

The following are the features of **Cynergy**.

- Real-time messaging
- Private messaging
- User mentions
- Message deletion (*administrator*)
- Kick/Ban users (*administrator*)
- See user IP address (*administrator*)
- Emoji support
- Dark mode

User Levels

- User

- VIP (can delete messages)
- Moderator (can kick/ban users)
- Administrator (all of the above)

Commands

User Commands

<code>/pm <user> <message></code>	Sends private message to specific user.
<code>/me <message> or /em <message></code>	Send message in italic.
<code>/shrug <message></code>	Send message with <code>'~_(_)_/'</code>
<code>/name <message></code>	Change username.
<code>/user</code>	Show connected users.
<code>/help</code>	Show help dialog.
<code>/clear</code>	Clear message history.
<code>/reconnect</code>	Reconnects user to the server.

Administrator Commands

<code>/alert <message></code>	Alerts everyone in chat.
<code>/kick <user></code>	Kicks user from the server.
<code>/ban <user> <minutes></code>	Bans user from the server in number of minutes.
<code>/role <user> <0-3></code>	Change user role.

Console Commands

Note: Execute role command first in the console in order for administrators to assign roles to other users.

<code>/role <user></code>	Change user role.
---------------------------------	-------------------

Use Case Analysis

1. Send Message

Brief Description:

- This allows the user to send a message.

Actors:

- Users

Scenarios:

- User sends a message.

Pre-conditions:

- User is connected to the server.

Post-conditions:

- User is communicating with users connected to the server.

2. View Message

Brief Description:

- This allows the user to view a message.

Actors:

- Users

Scenarios:

- User views a message.

Pre-conditions:

- User is connected to the server.

Post-conditions:

- User is receiving messages from users connected to the server.

3. Assign Role

Brief Description:

- This allows the user to assign a role to a user.

Actors:

- Administrator

Scenarios:

- Admin assigns a role to a user.

Pre-conditions:

- User is connected to the server.
- User has administrative privileges.

Post-conditions:

- User is assigning users with roles.

UML diagrams

Use Case Diagram  use-case-diagram

Class Diagram  class-diagram

Usage

Install package dependencies using **npm**.

```
npm i or npm install
```

Start server

```
npm start
```

Go to <http://localhost:3000>

Note: The port number can be change in `config.json`.

Deploying using **ngrok**.

Download **ngrok**.

```
https://ngrok.com/download
```

Start server (**Cynergy**).

```
npm start
```

Start **ngrok** (Windows).

```
ngrok.exe http <port-of-server>
```

Note: The port is probably 3000.

Open link generated by **ngrok**.

Contributions

Groups

- Group 7
- Group 4