

---

## Individual Assignment: Chat

---

Svetlana Ostrovskaya

### ARCHITECTURAL DIAGRAM

Architectural Diagram is illustrated on Figure 1.

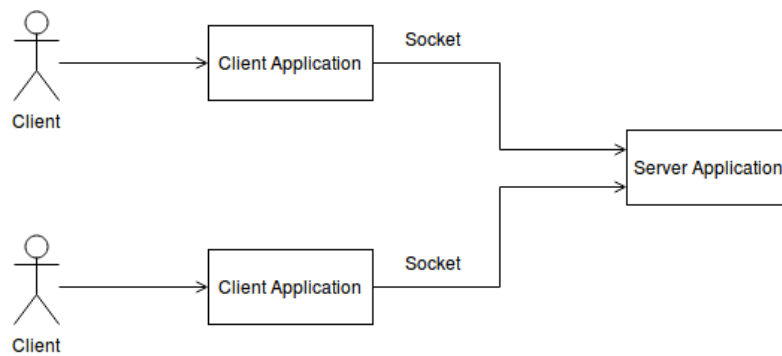


Figure 1 - Architectural Diagram

Actions, allowed for users, are represented as Use-Case Diagram, that is shown on Figure 2.

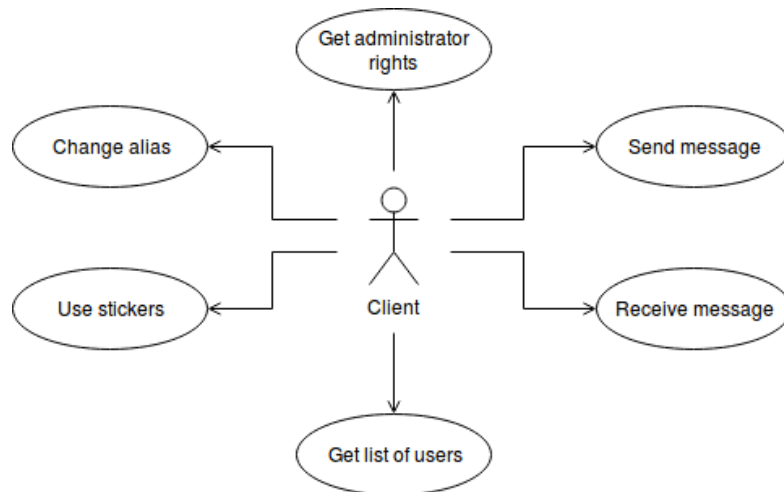


Figure 2 - Actions, allowed for common users

Additional actions, allowed for administrators, are illustrated on Figure 3.

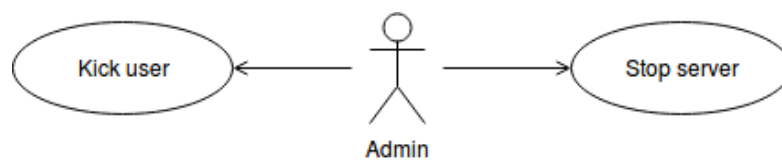


Figure 3 - Actions, allowed for administrator

## DESIGN DECISIONS

- Client - Server Architecture
  - Separated code for clients and server
  - Minimized system requirements for client application (all computations performed by server)
- Python Language
  - Simple for fast development
  - Cross-platform language
  - Python 2.7 is built-in on all Linux distributives
- Asynchronous Communication

- Several queries can be processed in parallel (at the same time)
- Non-blocking operations
- System resources are not idle

## HOW TO LAUNCH

- Docker version: 17.09.0-ce
- Unzip archive and select project folder
- Perform command "make build" in terminal

### Server

- Run command "make server"
- Message "Server is running..." means that launch was successfully performed

### Client

- Run command "make client"
- Put server's IP address in corresponded field "Host" (Server's IP can be found by using the following command: "docker inspect <Container\_ID>")
- Message "Connection established" means that launch was successfully performed

## COMMANDS

- /users - shows all connected users
- /alias - changes current alias (e.g. /alias <new alias>, alias should be longer than 3 characters)
- /sticker - sends stickers (e.g. /sticker lol)
- /exit - closes application
- /pass - allows to login as administrator (e.g. /pass qwerty)
- /kick - kicks user (e.g. /kick I.M)
- /stop - stops server