Spirent-pep-logo

**2018 International Internship Program**

**2nd round screening process**

**Project Design Documentation**

**Dongjin Park**

# Table of Contents

[Table of Contents 2](#_Toc511353391)

[I. Introduction 3](#_Toc511353392)

[I.1 Purpose 3](#_Toc511353393)

[I.2 Tasks 3](#_Toc511353394)

[II. Software Design 3](#_Toc511353395)

[II.1 Overview 3](#_Toc511353396)

[II.2 Class Relationships 9](#_Toc511353397)

[II.3 Database Schema 9](#_Toc511353398)

[III. Functionality 9](#_Toc511353399)

# Introduction

This project is to develop a chat-messaging system. It is based on server and client using multithreading, sockets, and database. There is one chat room and clients(users) can chat in chat room. Server and client are written in Java(jdk10) and database is written in MariaDB(Centos7). My development environment is Windows7 64bit, and I use Intellij as a develop tool.

## I.1 Purpose

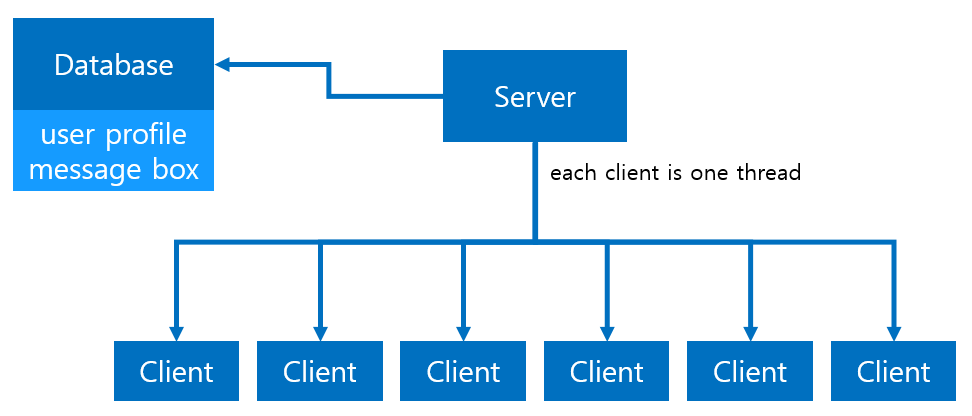
The purpose of this document is to describe the design of Chat Messaging System and how it works.

## I.2 Tasks

* implement server which spawns one thread per client(multithreading) and users TCP socket to communicate with client and also manages all message flow
* implement database which is store user profile and chat history
* implement client which is able to chat with another client and sign in/up

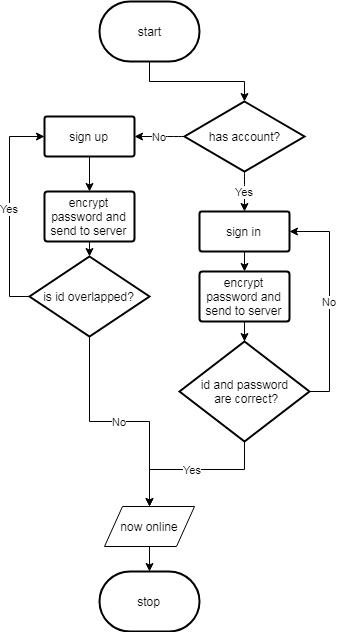
# Software Design

## II.1 Overview

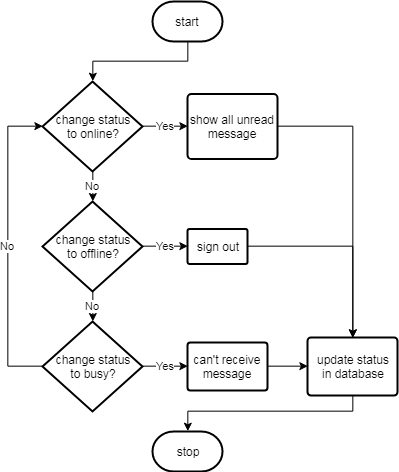


- Figure 1. overview -

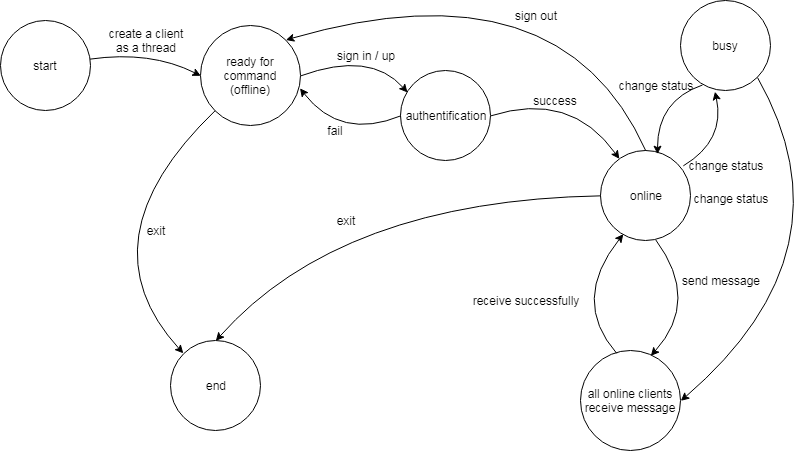
## II.2 Flow Charts and Diagrams



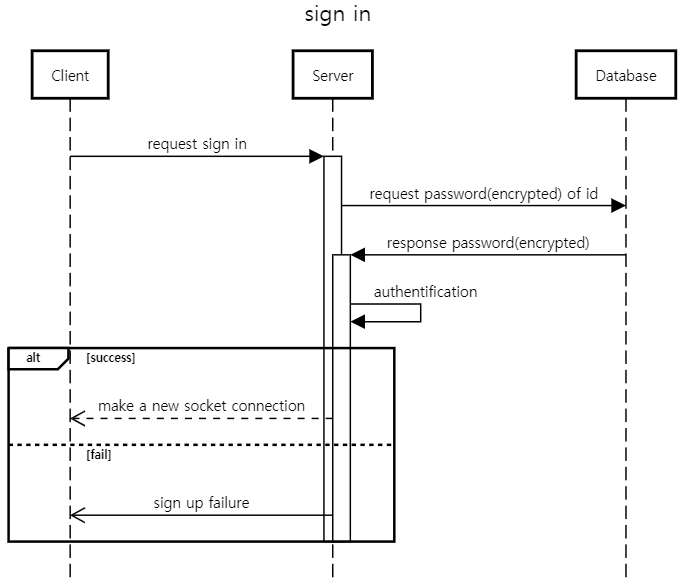
- Figure 2. sign in/up flow chart -



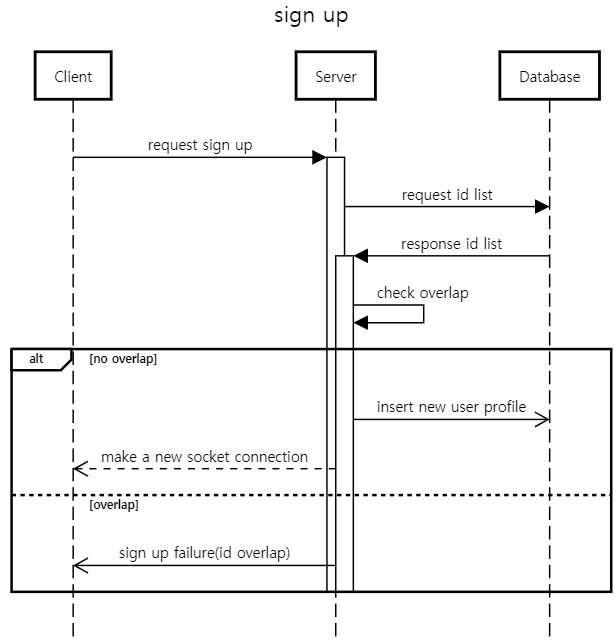
- Figure 3. change status flow chart -



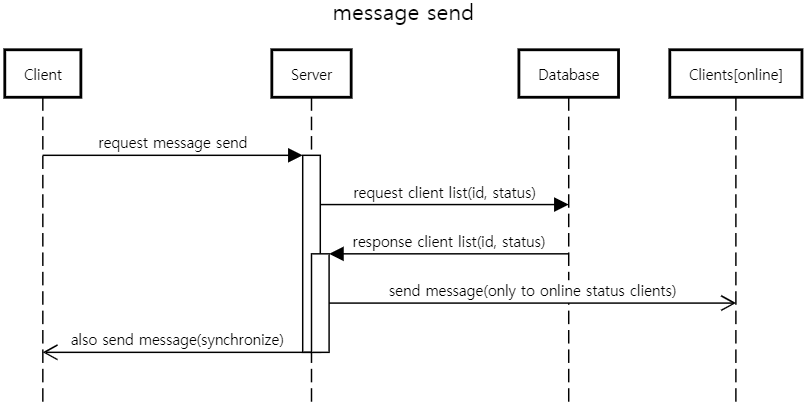
- Figure 4. state transition diagram -



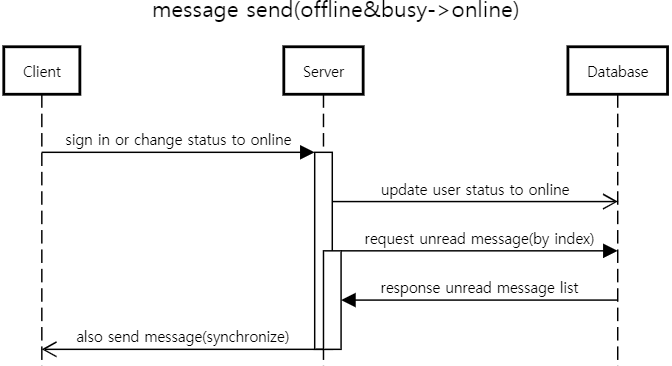
- Figure 5. sign in sequence diagram -



- Figure 6. sign up sequence diagram -

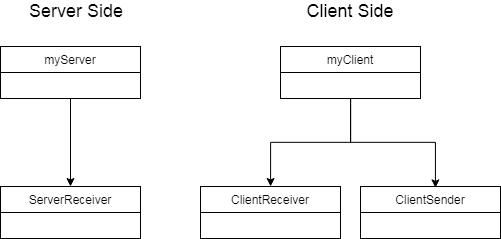


- Figure 7. message send sequence diagram -



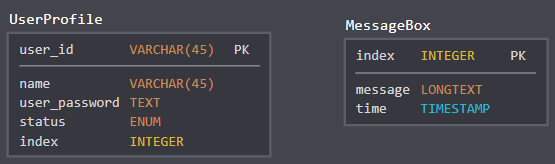
- Figure 8. message send(offline&busy->online) -

## II.3 Class Relationships



- Figure 9. Class Relationships -

## II.4 Database Schema



- Figure 10. Database Schema -

In UserProfile, user id is primary key. So user’s id must not be overlapped. Also there is name which is used in chat room. User password is stored as encrypted plain text(by sha-256) for information security issue. User status is also in database. If user changes status, database must be updated. The last one, index indicates how much this user has read the message.

MessageBox has all messages which clients send. All messages have unique index(in order) and it is a yardstick of read or unread. And also all messages have to include the time stamp when it is sent.

# Functionality

1. Server
   1. manage all clients as a thread(multithreading)
   2. use TCP/IP socket connection with client
   3. use database when receives request from client
   4. check id and password when client requests sign in
   5. check if id is overlapped when client requests sign up
   6. manage all messages
      1. check all user is read or not
      2. after receive message, insert it to all user’s database
      3. load message history
      4. check time stamp when receive message from client
2. Client
   1. sign in
   2. sign up
   3. send message
   4. change status(online, offline(sign out), busy)
      1. in online status, user can send message and receive message
      2. in offline status, user can’t to anything except sign in
      3. in busy status, user can send message but can’t receive message
   5. receive unread message when status is changed to online
   6. load message history
   7. sign out(make status offline)