聊天室系统概要设计

# 1.引言

## 1.1编写目的

在需求分析阶段中已经对本系统的功能需求做了详细的阐述。本阶段将会在需求分析阶段的基础上进行进一步的概要设计，完成系统的的大致设计并明确系统的数据结构与软件结构、确定设计方案。

在下一阶段的详细设计中，本阶段的概要设计将作为参考，以方便完成整个系统的设计工作。

## 1.2定义

## 1.3参考资料

《软件工程》 陆慧恩编著 上海交通大学出版社 2016-8

《概要设计说明书》 csdn网络资料库 2016-11

《概要设计》 博客园网络资料 2009-6

# 2.总体设计

## 2.1设计原则和要求

1.用户界面设计原则：由于本聊天系统是面向普通用户的，因此在界面设计方面需要拥有良好的人机交互界面，并且需要一定的文字标注用以提醒用户，布局也需考虑用户操作习惯，做到整齐明显、统一规范。

2.数据库设计原则：本系统涉及到不同用户之间的聊天，在数据库设计方面需要考虑数据的准确性和部分数据的唯一性，同时需要考虑数据操作的简易性，尽量避免数据的浪费。

3.系统操作性原则：本系统面向普通用户，基于用户人群对计算机的了解程度，要求本系统在操作性方面做到简洁易懂、快捷，并且尽可能缩短用户理解系统操作的时间。

4.数据安全性原则：由于用户信息、用户聊天记录具有一定的隐私性，因此要求系统对数据一定的防黑处理。

## 2.2需求规定

客户端需满足图中规格：



1.在登录首页输入账号密码，点击登录按钮，若账号或密码错误则提示“登陆失败，账号或密码错误或者用户不存在”；若账号及密码都正确则进入聊天室，显示所有在线用户、群聊内容及私聊内容。如果用户是第一次登录的话则必须要进行注册。

2.进入聊天室后，可以选择公聊、私聊，同时可以查找用户、查看用户信息，还可更改个人资料或退出聊天室。

## 2.3运行环境

操作系统：Windows 10

数据库系统：MySQL8.0

开发工具：eclipse

开发平台：Java

## 2.4基本设计和处理流程

下面采用图表的方式对聊天室系统软件的操作流程及基本设计概念进行说明分析。

### 2.4.1系统框架图



### 2.4.2操作流程图



# 3.接口设计

## 3.1用户接口

系统以鼠标和键盘为用户接口方便用户对系统数据的操作。在用户界面部分为了便于实现人机交互、易于上手，本系统采用了图形用户接口，鉴于整个系统采用Java开发，因此使用Java GUI进行界面开发，由awt、swing库中可视化组件进行界面布局设计。

## 3.2外部接口

外部接口为鼠标、键盘等交互工具，出客户端外还需配置一服务端，两者调用同一数据库以保障数据一致性。

## 3.3内部接口

系统内部程序依据不同子系统设计不同的public类，同时为了保障数据的安全性，大部分类都会进行封装，各模块间采用函数调用，参数传递、返回值的方式，同时会设计单独的用户类以分别表示用户信息，在有关模块调用时直接传递用户对象。

# 4.运行设计

## 4.1客户端

### 4.1.1登录模块

通过数据库储存用户注册时的数据，之后将用户登陆数据与之校验来有效快捷地实现用户相应的数据操作和管理，主要包括：

1.用户的注册、信息提示

2.用户的登录、信息提示

### 4.1.2在线用户模块

用户通过该模块获取聊天室状态基本信息，主要包括：

1.显示聊天室在线用户

2.显示聊天室用户上下线信息

### 4.1.3聊天模块

用户通过该模块实现一定的聊天功能，主要包括：

1.在聊天室中进行公聊

2.对指定用户进行私聊

3.储存一定量的聊天信息

4.查看聊天信息

### 4.1.4退出模块

1.实现用户的退出功能

## 4.2后台服务端

### 4.2.1数据库操作管理

后端通过单一类封装了系统同数据库交互用到的各个方法，来简化该交互过程，主要包括：

1.创建新用户

2.用户信息的更改与添加

3.通过给定用户数据项信息查找用户

4.创建新的聊天记录数据记录

5.通过给定数据项信息查找聊天记录

### 4.2.2用户操作管理

后台服务端通过对用户发送的数据进行分类处理，并提示用户有关连接信息，在系统整体上作为控制器存在，主要包括：

1.获取用户操作类型信息

2.依据不同用户操作调取不同数据库操作管理方法

3.在服务端提示用户上下线信息

4.在服务端显示在线用户

### 4.2.3客户端连接管理

1.控制连接端口开关

# 5.数据结构设计

## 5.1数据库结构设计

本系统采用MySQL数据库对系统数据进行储存、维护管理，所有数据设计2张数据表：

1.用户信息表 accounts\_information

记录用户的信息，主键为用户账号名。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名称 | 数据类型 | 大小 | 约束 | 描述 |
| name | varchar | 20 | 主键、非空 | 用户账号名 |
| password | varchar | 15 | 非空 | 用户密码 |
| sex | varchar | 5 |  | 用户性别 |
| phone | Varchar | 17 |  | 用户手机号 |
| email | varchar | 20 |  | 用户邮箱 |
| age | int |  | 无符号 | 用户年龄 |
| create\_time | datetime |  |  | 用户创建时间 |
| id | int |  | 非空、自增、唯一 | 用户id |
| brithday | varchar | 25 |  | 用户生日 |
| image\_src | varchar | 50 |  | 用户头像路径 |

2.聊天记录表 chatrecord

记录用户聊天记录，主键为用户账号名。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名称 | 数据类型 | 大小 | 约束 | 描述 |
| name | varchar | 20 | 主键、非空 | 用户账号 |
| chats | Varchar | 50 |  | 聊天信息 |
| send\_time | datetime |  |  | 信息发送时间 |

## 5.2数据结构与程序的关系

用户在操作过程中对个人信息进行的增改查操作及聊天信息发送操作，都是通过程序中的MySQL语句来实现的，用户可以对自身信息进行修改，并且只能用户本身能对自身信息进行修改以确保数据的安全性。

# 6.系统出错处理设计

## 6.1出错信息

1.用户登录信息错误：用户登录时输入的信息与数据库中保存信息不符

2.用户查找错误：根据用户给定查找信息未在数据库中找到与之匹配的用户

3.数据库错误：在用jdbc连接、操作数据库时，发生MySQL语句错误或数据库连接错误

4.端口冲突错误：由于服务端开启端口已经被其他应用占用，导致端口冲突错误

## 6.2处理措施

说明错误出现后系统可能采取的处理措施，包括：

1、对于用户信息有关的一般错误，采取由对话框对应提醒用户并让用户重新输入或退出的处理方式。

2.对于数据库连接、端口号冲突等影响系统整体的错误，返回记录错误信息的日志文件，并强制退出系统。

# 7.系统维护设计

由于系统项目比较小，因此维护方面并未设置过多的设计，仅可通过对各个模块的修改来进行单独维护。