**概要设计说明书**

项目名称： SchoolSNS

项目组长： 刘芃杰

其他成员： 田易、李锦、叶水欢

日 期： 2019/12/20

南京师范大学 计算机科学与技术学院制

目 录

[1 引言 3](#_Toc25154486)

[1．1编写目的 3](#_Toc25154487)

[1．2背景 3](#_Toc25154488)

[1．3定义 3](#_Toc25154489)

[1．4参考资料 3](#_Toc25154490)

[2　总体设计 3](#_Toc25154491)

[2．1需求规定 3](#_Toc25154492)

[2．2运行环境 4](#_Toc25154493)

[2．3基本设计概念和处理流程 4](#_Toc25154494)

[2．4结构 5](#_Toc25154495)

[2．5功能器求与程序的关系 6](#_Toc25154496)

[2．6人工处理过程 6](#_Toc25154497)

[2．7尚未解决的问题 6](#_Toc25154498)

[3　接口设计 6](#_Toc25154499)

[3．1用户接口 6](#_Toc25154500)

[3.1.1 用户注册界面 6](#_Toc25154501)

[3.1.2 用户登录界面 6](#_Toc25154502)

[3.1.3 用户创建流程界面 7](#_Toc25154503)

[3.1.4 用户审核界面 7](#_Toc25154504)

[3.1.5 用户申请界面 7](#_Toc25154505)

[3．2外部接口 7](#_Toc25154506)

[3．3内部接口 7](#_Toc25154507)

[4　运行设计 8](#_Toc25154508)

[4．1运行模块组合 8](#_Toc25154509)

[4．2运行控制 8](#_Toc25154510)

[4．3运行时间 8](#_Toc25154511)

[5　系统数据结构设计 8](#_Toc25154512)

[5．1逻辑结构设计要点 8](#_Toc25154513)

[5．2物理结构设计要点 9](#_Toc25154514)

[5．3数据结构与程序的关系 9](#_Toc25154515)

[6　系统出错处理设计 10](#_Toc25154516)

[6．1出错信息 10](#_Toc25154517)

[6．2补救措施 10](#_Toc25154518)

[6．3系统维护设计 10](#_Toc25154519)

# **1** 引言

## 1．1编写目的

目的：让该软件开放的所有群体及个体对软件的背景、需求、设计理念、设计过程、软件实现所需要的硬件设备、软件设计及后续维护有一个基本了解。

预期读者：可能使用该软件的企业、公司、工作人员、私人；

## **1．2背景**

a．待开发软件系统的名称：SchoolSNS；

　　b．任务提出者：姜乃松

开发者：刘芃杰、田易、李锦、叶水欢

用户：可能使用该软件的企业、公司、工作人员、私人

运行该软件的计算站：腾讯云服务器。

## **1．3定义**

需求：用户解决问题或达到目标所需的条件和功能。

个人：拥有该账号的个体。

好友：添加的账号不同的其他用户。

## **1．4参考资料**

1.页面变量说明.doc

2.流程平台需求.doc

# **2　总体设计**

## **2．1需求规定**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能 | 输入 | 处理 | 输出 |
| 注册 | 昵称、账号、密码 | 添加到数据库 | 提示：注册成功 |
| 登录 | 账号、密码 | 数据库检索 | 进入系统 |
| 加好友 | 账号或昵称 | 显示符合条件的用户 | 所添加的用户出现在好友列表 |
| 审核好友申请 | 登录本人账号 | 通过或是拒绝申请 | 审核结果 |
| 设置头像 | 单击所中意的头像选项 | 更改头像 | 显示新头像 |
| 聊天 | 聊天内容 | 发送至好友 | 显示聊天内容 |

## 2．2运行环境

软件系统软件运行环境：windows7及以上系统，MySQL，Javascript，Python，PYQT。

服务器要求安装有PHP7.0以上环境和Apache服务器软件。

需要Chrome、FireFox、Edge等主流浏览器才可保证正确运行。

## **2．3基本设计概念和处理流程**

SchoolSNS

系统

注册

登录

选择操作类型

好友管理

选择好友

设置头像

加好友

聊天

选择头像

审核好友申请

## 2．4结构

概念结构

好友信息

注册信息

个人信息

头像

昵称

账号

消息

账号

昵称

初次密码

密码再确认

账号

密码

昵称

头像

状态

## 2．5功能器求与程序的关系

|  |  |  |
| --- | --- | --- |
|  | 用户 | 用户的好友 |
| 用户基本信息 | √ |  |
| 好友申请状态 | √ | √ |
| 聊天记录 | √ | √ |

## 2．6人工处理过程

用户注册和登录时，需要手动选择“注册”或“登录”按钮，输入账号和密码。

用户需要在好友列表中选择一个好友才能开始聊天。

用户需要手动选择通过或拒绝一个好友申请。

用户加好友时需要在输入框中输入账号或昵称搜索符合条件的用户。

## **2．7尚未解决的问题**

由于各个网络NAT模式不同，没能做到真正的P2P，改由服务器转接

# **3　接口设计**

## **3．1用户接口**

### 3.1.1 用户注册界面

用户选择注册按钮，输入昵称、账号、两次密码提交，若账号不重复且两次密码一致即可注册成功。

### 3.1.2 用户登录界面

用户选择登录按钮，输入账号和密码，软件审核通过后进入软件正式使用页面。

### 3.1.3 用户软件主界面

用户登录成功后可以得到好友信息及自己的信息：

界面左上角显示自己的头像和昵称；

界面右上角有“好友邀请”控件和“添加好友”控件；

界面底部有“账户设置”按钮，单击设置按钮可跳转到设置窗口；

界面中部有好友列表和状态，单击好友按钮可跳转到相应窗口；

### 3.1.4 用户加好友界面

用户在输入框，输入账号或昵称，系统显示符合条件的用户，用户通过加好友按钮添加好友。

### 3.1.5 用户审核好友申请界面

系统显示所有自己收到的好友申请列表，若未决定，就显示两个按钮（通过，不通过）；如果已经决定，就显示结果。

### 3.1.6 用户聊天界面

界面上半部分显示聊天记录，包括消息发送者昵称、消息发送时间以及聊天内容数据；

界面下半部分为文本输入区，在此发送新消息。

## **3．2外部接口**

无

## **3．3内部接口**

用户登录系统模块调用用户信息的数据；

用户聊天模块同步好友的聊天内容；

审核申请模块调用申请的数据。

# **4　运行设计**

## **4．1运行模块组合**

1、客户端发给服务器：登录、注册、发送消息、设置个人信息、搜索好友、申请加好友、审核好友申请、客户端下线通知服务器；

2、服务器发给客户端：登录结果、注册结果、在线时接收新消息、返回好友搜索结果、在线时接收好友申请结果、发送好友下线消息。

## 4．2运行控制

1、用户登录时，数据库会根据用户账号显示用户相关信息。

2、用户提出添加好友申请后，需要对方审核，系统将审核结果显示给用户。

3、用户登录后可以设置和修改个人信息，并发起聊天。

## **4．3运行时间**

每个操作（点击）反应时间不超过0.5秒。

# **5　系统数据结构设计**

## **1逻辑结构设计要点**

|  |  |  |  |
| --- | --- | --- | --- |
| 表格 | 字段 | 标记 | 状态 |
| friend | status | 0 | 待审核 |
| 1 | 同意 |
| 2 | 拒绝 |
| chathistory | readed | 0 | 未读 |
| 1 | 已读 |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | 标识符 | 描述信息 | 状态 |
| Userinfo(用户信息表) | id,nick,password,avatar，ip，port | 保存用户基本信息 | 使用 |
| Friend(好友申请表) | toid，fromid，status | 保存好友申请信息 | 使用 |
| Chathistory(上线提醒表) | fromid，toid，senttime datatime，content varchar，readed tinyint | 保存上线信息 | 使用 |
| Optjson() | ope，funcname |  | 使用 |

## 5．2物理结构设计要点

由客户端输入的信息存入服务端的数据库中，访问方式根据操作人员而定。

## **5．3数据结构与程序的关系**

1、系统的数据结构由标准数据库语言SQL生成

2、创建表格的程序：

create table userinfo

(

id int not null,

nick varchar(20) not null,

password varchar(20) not null,

avatar int not null,

ip varchar(50),

port int,

primary key(id)

);

create table friend

(

toid int not null,

fromid int not null,

status int not null,

primary key(toid,fromid)

);

create table chathistory

(

fromid int not null,

toid int not null,

sendtime datetime not null,

content varchar(200) not null,

readed tinyint not null,

primary key(fromid,toid,sendtime)

);

create table optjson

(

ope int not null,

funcname varchar(20) not null,

primary key (ope)

);

# **6　系统出错处理设计**

## **6．1出错信息**

|  |  |
| --- | --- |
| 错误类型 | 错误提示 |
| 输入信息错误 | 信息输入错误，请重新输入 |
| 没有选择条件，就进行查询 | 请选择要查询的信息 |
| 程序运行错误 | 程序运行错误 |

## 6．2补救措施

1. PHP操作数据库的代码，采用事务处理，即mysqli\_begin\_transaction函数。

即当SQL语句出错时，可以进行回溯。

2、当原始系统数据丢失时启动副本建立和启动技术，由于数据库中已经有备份，故在系统出错后可以依靠数据库的恢复功能。

3、恢复及再启动技术说明将使用的恢复再启动技术，使软件从故障点恢复执行或使软件从头开始。

## **6．3系统维护设计**

 由于本系统没有深入复杂设计，没有外加维护模块，所以维护工作比较简单，仅对数据库做一些基本维护即可。