需求规格说明书

目录

[1．引言 3](#_Toc506358922)

[1.1编写目的 3](#_Toc506358923)

[1.2项目背景 3](#_Toc506358924)

[1.3定义 3](#_Toc506358925)

[1.4参考资料 4](#_Toc506358926)

[2．任务概述 4](#_Toc506358927)

[2.1目标 4](#_Toc506358928)

[2.2运行环境 4](#_Toc506358929)

[3．功能需求 5](#_Toc506358937)

[4.1功能划分 5](#_Toc506358938)

[4.2功能描述 5](#_Toc506358939)

[4．性能需求 5](#_Toc506358940)

[4.1数据精确度 5](#_Toc506358941)

[4.2时间特性 6](#_Toc506358942)

[5．运行需求 6](#_Toc506358944)

[5.1用户界面 6](#_Toc506358945)

[6．其它需求 6](#_Toc506358949)

[7．用例设计 6](#_Toc506358949)

# 1．引言

## 1.1编写目的

《同学帮帮忙》是一个类似于知乎的应用的校园版，用于解决一些校内的同学之间的困难，可用于一些学习上的、生活上的问题提问及解决。用其作为软工PK项目，不仅可以练习项目的一些设计思维及文档书写，深入理解软件工程；同时还可以将其转化为一款校内的小型应用。

## 1.2项目背景

虽然当前有很多已经很完善的线上提问平台，如知乎，百度知道等，但是由于这些应用都是针对整个社会群体的，故一些定向于校内的问题，在这些已经成型的平台上不能得到很好的解决。并且有些帮助，是需要通过线上线下结合完成的，显然，这些已经成型的平台很难为用户提供这一点（网上用户大多使用虚拟信息，来自各地，线下帮助难以得到实现）。

故《同学帮帮忙》正好很好的解决了这点，这是一款面向校内这个群体用户的应用，地理位置的允许，使得线下帮助很容易进行。

在考虑到当今大学生的习惯，实现移动端的应用将会更加实用，手机是一般学生都会随身携带的，故基于移动端的应用将会使用起来更加方便。

结合以上各点，故决定开发基于Android端的《同学帮帮忙》校内互助平台。

## 1.3定义

客户端：学生群体使用的Android端应用，用于提出问题和帮助其他同学解决问题

服务器端：将学生群体联系起来，对问题进行显示、解决控制等

数据库：用于存储个人信息、问题、解决情况等

## 1.4参考资料

# 2．任务概述

## 2.1目标

实现用户通过客户端发送待解决的问题、服务器端将问题显示给所有用户；实现客户端对得到的问题列表进行解决，服务器端记录，显示给问题提出者。

## 2.2运行环境

服务器：

操作系统：Microsoft Windows XP

数 据 库：MySQl

客户端：

操作系统：android

# 3．功能需求

## 3.1功能划分

服务器：移动端的连接注册功能，修改密码功能，验证登录功能，存储客户端提出的问题功能，记录问题解决状态功能。

客户端：接收来自于服务器的待解决问题列表并显示，向服务器进行注册，向服务器发送修改密码，通过服务器验证登陆，解决指定问题功能。

## 3.2功能描述

1．移动端的连接注册功能：客户端在使用该应用之前，需要进行注册，务器需要进行注册信息接收、验证并返回特定指示。

2．修改密码功能：接收客户端发送的修改密码请求、验证并返回特定指示。

3．验证登录功能：接收客户端发送的登陆请求，验证登录是否成功并返回标识。

4．存储客户端提出的问题功能：客户端提出问题时，上交给服务器，服务器需要对其进行存储

5．记录问题解决状态功能：记录问题的解决状态（已解决，待解决，解决中）

6．接收来自于服务器的待解决问题列表并显示：接收来自于服务器的所有其他用户的问题的列表，将其显示在界面，供用户解决

7．向服务器进行注册：用户在使用之前，需要先进行注册，将注册信息交由服务器端进行处理

8．向服务器发送修改密码：修改当前用户密码，将修改信息发送给服务器进行处理

9．通过服务器验证登陆：使用前，需要进行登录，将账号密码信息发送给服务器端进行处理判断是否登陆成功

10．解决指定问题功能：可以对从服务器得到的问题列表中的某个问题进行解决，并通知服务器

# 4．性能需求

## 4.1数据精确度

需要保证客户端及服务器端交互的的信息准确无误，没有缺损。

## 4.2时间特性

在网络良好的状态下，客户端从服务器获取服务及信息，应在3秒之内

# 5．运行需求

## 5.1用户界面

加强用户体验，进行必须的Android的控件的自定义

# 6．其它需求

对应于校内应用的要求，应根据学生心理进行用户体验设计

# 7．用例分析

用例图：





客户端：

|  |  |
| --- | --- |
| 用例名称： | 注册 |
| 用例ID： | HS-001 |
| 角色： | 客户端 |
| 用例说明： | 注册账号 |
| 前置条件： | 服务器处于运行状态 |
| 基本事件流： | 1. 填写注册信息 2. 提交给服务器，服务器验证通过 3. 注册成功 |
| 其它事件流： |  |
| 异常事件流： | 1. 填写注册信息 2. 提交服务器，账号已存在 3. 注册失败 |
| 后置条件： |  |

|  |  |
| --- | --- |
| 用例名称： | 修改密码 |
| 用例ID： | HS-002 |
| 角色： | 客户端 |
| 用例说明： | 客户端修改密码 |
| 前置条件： | 服务器处于运行状态 |
| 基本事件流： | 1. 客户端填写修改信息 2. 发送到服务器端，进行处理 3. 修改密码成功 |
| 其它事件流： |  |
| 异常事件流： | 1. 客户端填写修改信息 2. 发送到服务器端，进行处理 3. 账号不存在、账号或密码错误，修改失败 |
| 后置条件： |  |

|  |  |
| --- | --- |
| 用例名称： | 登录 |
| 用例ID： | HS-003 |
| 角色： | 客户端 |
| 用例说明： | 客户端进行登录 |
| 前置条件： | 服务处于运行状态 |
| 基本事件流： | 1. 填写账号、密码 2. 发送到服务器端、进行验证 3. 登录成功 |
| 其它事件流： |  |
| 异常事件流： | 1. 填写账号、密码 2. 发送到服务器端、进行验证 3. 3.账号或密码错误，登录失败 |
| 后置条件： |  |

|  |  |
| --- | --- |
| 用例名称： | 获取、显示问题列表 |
| 用例ID： | HS-004 |
| 角色： | 客户端 |
| 用例说明： | 客户端从服务器获取问题列表并进行显示 |
| 前置条件： | 服务处于运行状态 |
| 基本事件流： | 1. 发送请求到服务器端 2. 服务器端返回问题列表 3. 显示问题列表 |
| 其它事件流： |  |
| 异常事件流： |  |
| 后置条件： |  |

|  |  |
| --- | --- |
| 用例名称： | 解决问题 |
| 用例ID： | HS-005 |
| 角色： | 客户端 |
| 用例说明： | 客户端选定问题进行解决 |
| 前置条件： | 服务处于运行状态 |
| 基本事件流： | 1. 选定要解决的问题 2. 通知服务器该问题已被该客户端解决 |
| 其它事件流： |  |
| 异常事件流： | 1. 选定要解决的问题 2. 通知服务器该问题已被该客户端解决 3. 服务器端查询到该问题已被其他用户解决，返回提示 |
| 后置条件： |  |

|  |  |
| --- | --- |
| 用例名称： | 获取、显示个人信息 |
| 用例ID： | HS-006 |
| 角色： | 客户端 |
| 用例说明： | 客户端获取用户个人信息、显示 |
| 前置条件： | 服务处于运行状态 |
| 基本事件流： | 1. 请求服务器 2. 服务器返回信息、进行显示 |
| 其它事件流： |  |
| 异常事件流： |  |
| 后置条件： |  |

服务器端：

|  |  |
| --- | --- |
| 用例名称： | 验证注册 |
| 用例ID： | HS-007 |
| 角色： | 服务器端 |
| 用例说明： | 验证客户端的注册 |
| 前置条件： | 客户端请求 |
| 基本事件流： | 1. 处理客户端发送的注册信息 2. 返回是否注册成功 |
| 其它事件流： |  |
| 异常事件流： |  |
| 后置条件： |  |

|  |  |
| --- | --- |
| 用例名称： | 验证修改密码 |
| 用例ID： | HS-008 |
| 角色： | 服务器端 |
| 用例说明： | 验证客户端修改密码是否成功 |
| 前置条件： | 客户端请求 |
| 基本事件流： | 1. 验证修改信息是否正确 2. 返回是否修改成功 |
| 其它事件流： |  |
| 异常事件流： |  |
| 后置条件： |  |

|  |  |
| --- | --- |
| 用例名称： | 验证登录 |
| 用例ID： | HS-009 |
| 角色： | 服务器端 |
| 用例说明： | 验证客户端是否登录成功 |
| 前置条件： | 客户端请求 |
| 基本事件流： | 1. 验证客户端的登录信息 2. 返回是否登录成功 |
| 其它事件流： |  |
| 异常事件流： |  |
| 后置条件： |  |

|  |  |
| --- | --- |
| 用例名称： | 返回问题列表 |
| 用例ID： | HS-010 |
| 角色： | 服务器端 |
| 用例说明： | 返回问题的列表给客户端 |
| 前置条件： | 客户端请求 |
| 基本事件流： | 1. 从数据库取出所有问题 2. 将问题列表返回给客户端 |
| 其它事件流： | 1. 从数据库取问题，当前无待解决问题 2. 返回标识通知客户端当前无待解决问题 |
| 异常事件流： |  |
| 后置条件： |  |

|  |  |
| --- | --- |
| 用例名称： | 存储问题 |
| 用例ID： | HS-011 |
| 角色： | 服务器端 |
| 用例说明： | 将客户端发送的问题进行存储 |
| 前置条件： | 客户端请求 |
| 基本事件流： | 1. 获取问题 2. 将问题存储到数据库 3. 通知客户端问题已存储 |
| 其它事件流： |  |
| 异常事件流： |  |
| 后置条件： |  |

|  |  |
| --- | --- |
| 用例名称： | 存储问题状态 |
| 用例ID： | HS-012 |
| 角色： | 服务器端 |
| 用例说明： | 当问题状态改变时，更新数据库表里信息 |
| 前置条件： | 客户端发送状态 |
| 基本事件流： | 1. 在数据库里对新的问题状态进行更新 2. 返回标识通知客户端状态已更新 |
| 其它事件流： |  |
| 异常事件流： |  |
| 后置条件： |  |