

学长帮帮忙

软件系统设计说明书

目 录

0. 文档介绍.....	1
0.1 文档目的与范围.....	3
0.2 读者对象.....	3
0.3 参考文献.....	3
1. 系统概述.....	错误! 未定义书签。
2. 设计约束.....	4
3. 开发、测试与运行环境.....	5
4. 软件系统结构图.....	错误! 未定义书签。
5. 功能模块设计概述.....	错误! 未定义书签。
5.1 模块汇总.....	错误! 未定义书签。
5.2 模块之间的关系.....	错误! 未定义书签。
5.3 命名规则（可选）	错误! 未定义书签。
6. 数据库设计概述.....	11
6.1 数据库环境说明.....	11
6.2 数据库命名规则.....	11
6.3 安全性设计说明.....	11
6.4 表汇总和表设计.....	12
7. 用户界面设计概述.....	错误! 未定义书签。
8. 综合考虑（可选）	18
8.1 稳定性和可扩展性.....	18
8.2 性能分析.....	18
8.3 复用和移植.....	错误! 未定义书签。
8.4 防错与出错处理.....	18
8.5 其它.....	错误! 未定义书签。

0. 文档介绍

本系统说明书是针对软件工程的小组课程作业——学长帮帮忙而编写。

0.1 文档目的与范围

本文档将会从系统概述、设计约束、开发、测试与运行环境、软件系统结构图、功能模块设计概述、数据库设计概述、用户界面设计概述、综合考虑这八大部分的内容对项目展开描述，希望能让读者了解这个项目的大致内容。目的是对该项目进行详细设计，在概要设计的基础上进一步明确系统结构，详细地介绍系统的各个模块，为进行后面的实现和测试作准备。

0.2 读者对象

本文档面向多种读者对象：

- (1) 项目经理：项目经理可通过此文档了解系统的相关功能和描述，也可以利用该文档把握好软件系统开发的方向以及预测进度，还有进行项目管理。
 - (2) 设计员：对需求进行分析，并设计出系统，包括数据库的设计。
 - (3) 程序员：充分认清系统的功能及用户的需求，方便编写系统使用手册。
 - (4) 测试员：根据该文档整理出系统的有关的合理的预期指标，以此为依据创建出测试系统的具体实例，以及对系统进行测试。
 - (5) 拟开发人员：了解开发组的系统需求是否能满足自身需求，及时交流意见。
 - (5) 用户：了解预期产品的功能和性能，并与分析人员一起对整个需求进行讨论和协商。
- 本详细设计说明书的预期读者为本项目小组的成员以及对该系统感兴趣，在以后想对系统进行扩展和维护的人员。

0.3 参考文献

《软件工程》(第二版) 张海藩 编著 人民邮电出版社出版

1. 系统概述

(1) 系统说明：

本系统是一个主要承办学校的帮扶活动的 APP，以触摸屏为主要交互工具，帮助学校发布两种形式的帮扶活动，一种是“一对多辅导”，另一种是“一对一辅导”。同时，我们还附加了私下结对辅导的活动，并增加了学习笔记模块。

(2) 主要功能：

① “一对多辅导”活动旨在寻找相关科目的学习成绩优秀的学长，主要面对的是大一新生人群，大一新生刚步入校园，可能还未习惯大学的学习生活，而大一新生也是学校里唯

一被强制参加晚自习的人群。这个时候，学长可以通过与学生们一起晚自习，回答大一新生的疑难问题，从而一定程度上可以改善大一新生不适应大学课程的问题。

② “一对一辅导”则是主要面向学习有困难的同学，尤其是挂科的同学，通过学长带领着一起相约学习，学长也可以在一定程度上进行问题辅导，培养同学的学习习惯，从而提高学生的学习成绩，减少挂科人数。

③ 私下结对辅导是一种学生自主发起辅导课程，目的是为了增加学生勤工俭学的机会，通过这个机会，可以改善部分学习成绩优秀并且需要课外兼职同学的处境。

④ 新增加的学习笔记模块是一个学长学姐们分享自己学习经验的平台，在这个平台上学长不但可以分享自己的学习经验，甚至还能上传自己的学习资料供需要的同学下载。

2. 设计约束

2.1 需求约束

- 本系统应当遵循的标准

数据命名的规则遵循《码出高效_阿里巴巴 Java 开发手册》、《Python PEP8》中相关的规定

- 软、硬件环境

基于 C/S 结构，开发阶段采用 windows 操作系统，轻量级数据库 Sqlite 进行开发，生产阶段采用 MySQL，用 orm 语言开发

- 接口/协议

服务端：win10

flask 接口：http 协议、tcp/ip 协议

- 用户界面

用户界面采用 Android 图形化界面，后台采用 windows 图形化界面，保证用户可以无障碍操作。

- 软件质量

正确性：系统必须交易能够被正确处理；

健壮性：系统应能够 7*24 小时无故障运行；

效率：系统可以支持 100 个终端同时发起交易，处理交易的时间不超过 10 秒钟；

易用性：APP 界面应采用图形化操作方式，便于用户操作；

安全性：报文中的关键数据域以密文方式传输；

可扩展性：应该充分考虑到将来功能的修改或增加，避免需求变更时大规模修改程序。

2.2 隐含约束

使用后台界面的是学校的业务人员，他们具有较强的业务技能和基本的电脑知识，对他们提供的操作界面应保证他们经过简单培训后无障碍的操作。

技术证明文件应该可以在多种操作系统上浏览。

应该把有可能变动的参数存放到配置文件或数据库中，保证修改参数的灵活性。

本 APP 的主要输入项目的数据类型：

输入数据类型	说明
文本型	主要用于学生的基本信息填写、辅导课程信息填写和学习笔记分享，以及学生填写评论
数字型	主要用于学生的学号填写认证
文件型	主要用于学生技术认证信息以及学习资料

图片	主要用于学生完善个人资料（头像）
----	------------------

本 APP 的主要输出项目的数据类型:

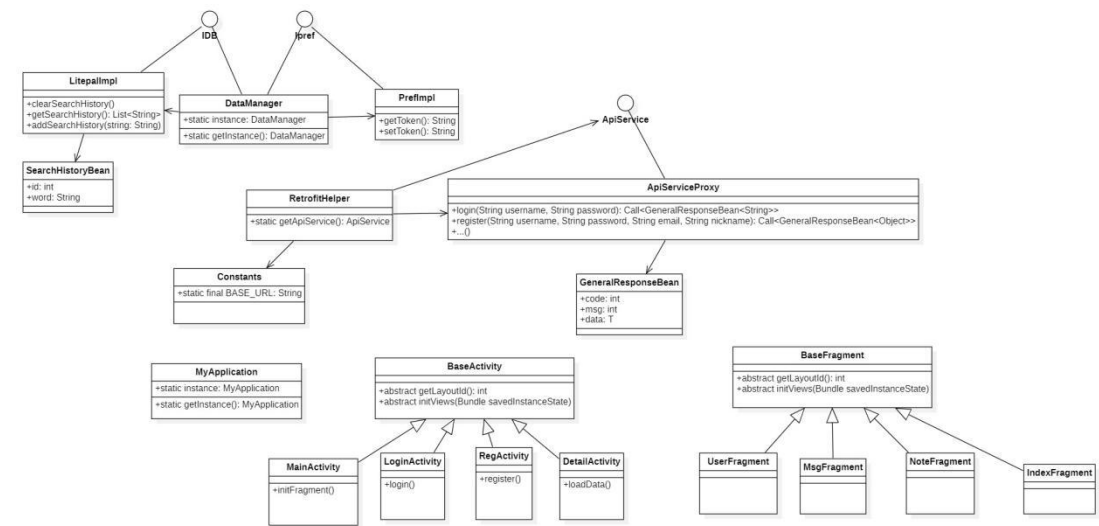
输出数据类型	说明
文本型	主要输出学业状况的相关文字信息（科目）、学习笔记、评论
数字型	各类数字（如绩点）

3. 开发、测试与运行环境

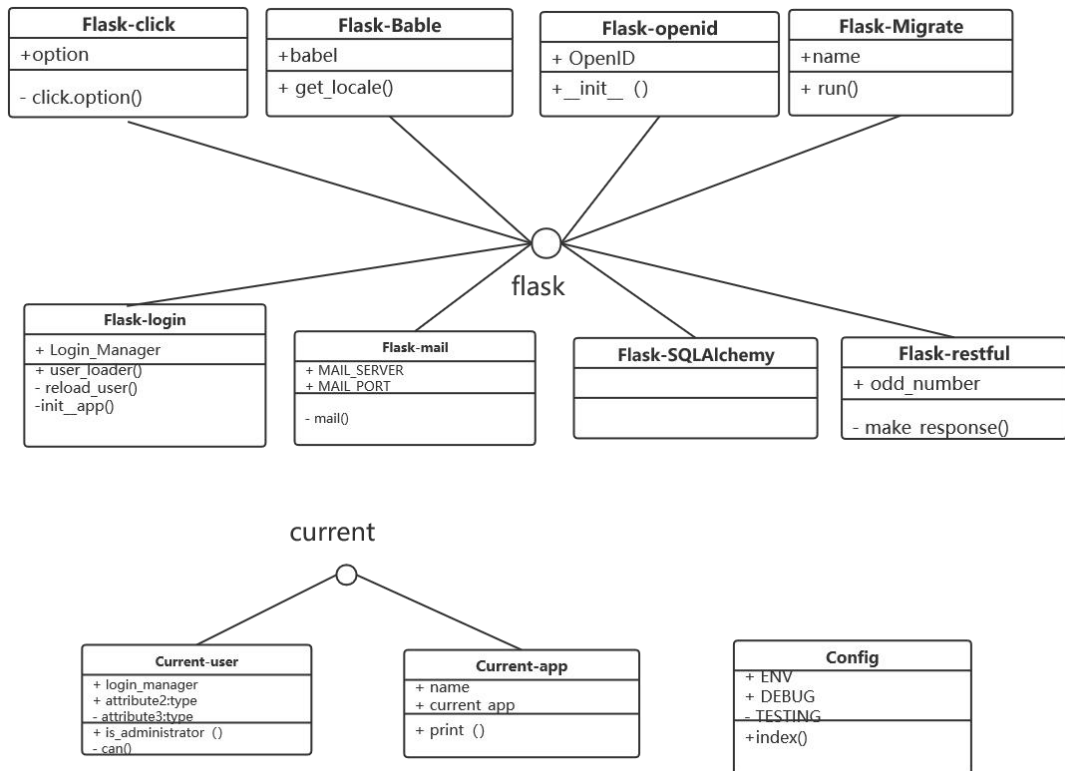
类别	标准配置
开发环境	windows10
	Phyon3.7
测试环境	Android10.0
	centos
运行环境	Android10.0
	centos

4. 软件系统结构图概述

4.1 前端类图



4.2 后端类图



4.3 类图说明

前端	
类名	功能概述
<i>LitepalImpl</i>	实现 Android 上的数据持久化方法 1
<i>SearchHistoryBean</i>	实体类
<i>DataManage</i>	数据管理类 使用单例模式 通过持有 LitePalImpl 以及 PrefImpl 的对象，使用委托的方式将数据访问与保存请求委托给 LitepalImpl 以及 PrefImpl 这两个类
<i>PrefImpl</i>	实现 Android 上的数据持久化方法 2
<i>ApiServiceProxy</i>	接口方法
<i>RetrofitHelper</i>	RetrofitHelper 是网络请求帮助类 使用单例模式 通过动态代理模式实现 ApiService 接口中的方法 借此实现网络请求
<i>Constants</i>	Constants 提供全局常量
<i>GeneralResponseBean</i>	实体类
<i>MyApplication</i>	Android 开发的组件

<i>BaseActivity</i>	Android 开发的组件通过一定的封装简化了 Android 的 Activity 中的一些样板代码
<i>MainActivity</i>	继承了 BaseActivity 为 APP 的主界面
<i>LogInActivity</i>	继承了 BaseActivity 用于实现登录视图以及相关操作
<i>RegActivity</i>	继承了 BaseActivity 实现注册
<i>DetailActivity</i>	继承了 BaseActivity 实现辅导详情页
<i>BaseFragment</i>	Android 开发的组件功能与 BaseActivity 类似 a
<i>UserFragment</i>	继承了 BaseFragment, 实现用户中心界面
<i>MsgFragment</i>	继承了 BaseFragment, 实现消息界面
<i>NoteFragment</i>	继承了 BaseFragment, 实现笔记界面
<i>IndexFragment</i>	继承了 BaseFragment, 实现首页部分
后端	
类名	功能概述
Flask-login	实现用户登录认证
Flask-mail	实现服务器发送邮件
Flask-SQLAlchemy	实现 orm 魔法
Config	提供全局变量
Current-app	提供工程函数注册
Current-user	提供当前登录用户操作
Flask-click	提供自定义命令
Flask-openid	认证
Flask-restful	Api 工具
Flask-Babel	提供国际化和本地话支持
Flask-Migrate	管理迁移数据库

5. 功能模块设计概述

5.1 模块汇总

前端	
模块名称	功能概述
Constants 模块	提供全局常量
Utils 模块	提供全局工具方法
Model 模块	提供数据存取以及网络请求支持

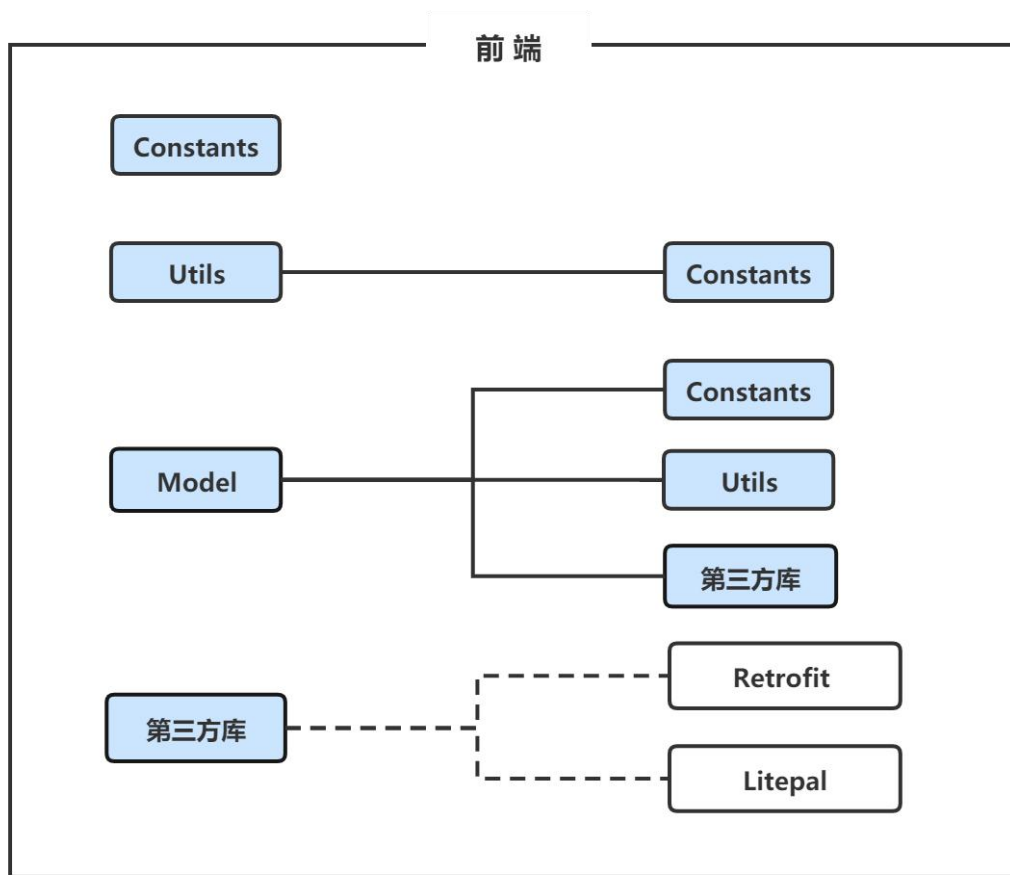
第三方库	涉及 Retrofit Litepal 等
后端	
模块名称	功能概述
Utils 模块	提供全局工具方法
config 模块	配置全局变量
models 模块	数据库表相关
第三方库	涉及 requests beautifulsoup 等

5.2 模块之间的关系

前端部分包含 4 个模块：

1. Constants 模块：提供全局常量
2. Utils 模块：提供全局工具方法
3. Model 模块：提供数据存取以及网络请求支持
4. 第三方库：Retrofit、Litepal 等

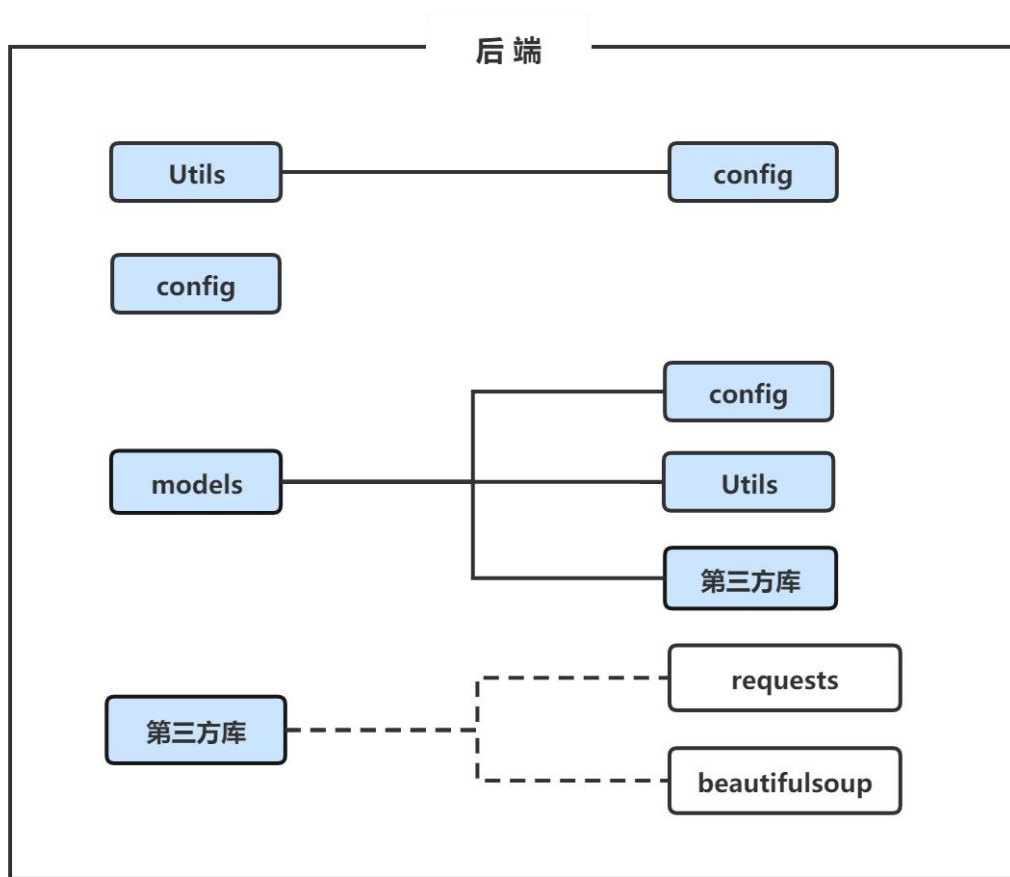
模块之间的依赖关系：



后端部分包含 4 个模块：

1. Utils 模块：提供全局工具方法
2. config 模块：配置全局变量
3. models 模块：数据库表相关
4. 第三方库：requests、beautifulsoup 等

模块之间的依赖关系：



5.3 命名规则

1. 模块的命名规则

在系统设计阶段应该为每个模块进行命名。命名的方法是，将模块英文名称的各个单词首字母组合起来，形成 3 到 5 个字符的缩写。若模块的英文名只有一个单词，可取该单词的前 3 个字母。各模块的命名以 3 个字母为宜。例如：

Arithmetic Logical Unit 模块，命名为 ALU。

Data Memory Interface 模块，命名为 DMI。

Decoder 模块，命名为 DEC。

2. 模块之间接口信号的命名

所有变量命名分为两个部分，第一部分表明数据方向，其中数据发出方在前，数据接收方在后，第二部分为数据名称。两部分之间用下划线隔离开。第一部分全部大写，第二部分所有具有明确意义的英文名全部拼写或缩写的第一个字母大写，其余部分小写。举例：

wire CPUMMU_WrReq;

下划线左边是第一部分，代表数据方向是从 CPU 模块发向存储器管理单元模块 (MMU)。下划线右边 Wr 为 Write 的缩写，Req 是 Request 的缩写。两个缩写的第一个字母都大写，便于理解。整个变量连起来的意思就是 CPU 发送给 MMU 的写请求信号。模块上下层次间信号的命名也遵循本规定。若某个信号从一个模块传递到多个模块，其命名应视信号的主要路径而定。

3. 模块内部信号:

模块内部的信号由几个单词连接而成, 缩写要求能基本表明本单词的含义; 单词除常用的缩写方法外 (如: Clock->Clk, Write->Wr, Read->Rd 等), 一律取该单词的前几个字母 (如: Frequency->Freq, Variable->Var 等); 每个缩写单词的第一个字母大写; 若遇两个大写字母相邻, 中间添加一个下划线 (如 DivN_Cntr); 举例:

```
SdramWrEn_n;  
FlashAddrLatchEn;
```

4. 文件夹的命名:

一般来说文件夹的命名要简单而易懂, 如 SPMS.App 和 SPMS.UI。如果一个文件夹名字太长, 则不太容易被快速识别。

6. 数据库设计概述

6.1 数据库环境说明

数据库系统: MySQL

设计工具: PowerDesigner

编程工具: PyCharm

有关配置:

Flask==1.1.1

Flask-SQLAlchemy==2.4.0

SQLAlchemy==1.3.7

6.2 数据库命名规则

表命名: 采用有意义的单词命名, 每个单词首字母大写。

字段命名: 采用有意义的单词命名, 使用下划线将每个单词隔开。

6.3 安全性设计说明

访问权限: 用户只能通过账号密码登录应用, 通过应用访问数据库, 而没有其他途径操作数据库。

用户密码：对用户密码进行加密处理，确保在任何地方都不会出现密码明文。

6.4 表汇总和表设计

表汇总	用途
User	存储用户信息
Help	存储发布的辅导记录
Note	存储学习笔记相关内容
Tutor	存储建立辅导关系后的信息

User:

	Name	Code	Data Type	Length	Precision	P	F	M
1	user_id	user_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	name	name	varchar			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	nickname	nickname	varchar			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	password	password	varchar			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5	sex	sex	varchar			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6	age	age	int			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
7	email	email	varchar			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Help:

	Name	Code	Data Type	Length	Precision	P	F	M
1	help_id	help_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	user_id	user_id	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	major	major	varchar			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	describe	describe	varchar			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	price	price	float			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
→	grade	grade	float			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Note:

	Name	Code	Data Type	Length	Precision	P	F	M
1	user_id	user_id	int			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	content	content	varchar			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
→	message	message	varchar			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Tutor:

	Name	Code	Data Type	Length	Precision	P	F	M
1	辅导编号	help_id	int			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	辅导人学号	Counselor_id	int			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	被辅导人学号	Mentee_id	int			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	辅导科目	major	varchar			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5	辅导人评价	C_evaluation	varchar			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
→	被辅导人评价	M_evaluation	varchar			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

7. 用户界面概述

7.1 官方帮扶

对接学院的各类帮扶，在首页的帮扶活动入口即可进入，比如我们进入了单人帮扶，看到某个求助同学的某科目要求，我们就可以申请帮助，求助同学在进入这个单人帮扶的页面的时候就可以看到有多少人愿意帮扶，经过自己的对比和星级判断，评论判断即可联系沟通，然后确定结对，后台就生成了这个帮扶的记录，要求双方在结对过程中打卡，最后看求助同学的进步与否来进行综测或者其他奖励

7.2 个人帮扶

在首页的不同科目（或者是兴趣技术类别）我们可以选择进入自己想要了解的版块，这里首先我们用户可以自己发布自己能够辅导他人的一个项目，点击加号填写信息完毕即可，然后在这个版块的列表里我们能看到不同人发布的辅导意向，对于感兴趣的人我们可以进行联系，然后预约，辅导者可以再自己的消息列表看到待处理的请求，这边就有别人的预约意向，同意后即可自行结对，后台为这个过程生成相应的记录。

登录：



认证：



主页：



详情：



帮扶活动:



发布辅导：

取消辅导信息发布

请选择要辅导的课程

心理学 绩点4.0

(若选择其他，请输入要辅导的技术，并上传证明材料)

例如C语言，spring框架等上传文件

支持扩展名：.rar .zip .doc .docx .pdf .jpg...

辅导宣言

请输入自己的辅导宣言

消息：

消息

系统通知

你好！欢迎注册学霸帮忙APP，在这里你可以提升自己的成绩，希望在未来的道路上，学长.....

09:41

待处理请求

及时处理预约信息

李子维

好啊，周末一起长跑吧

09:41

黄雨萱

明天我都有空

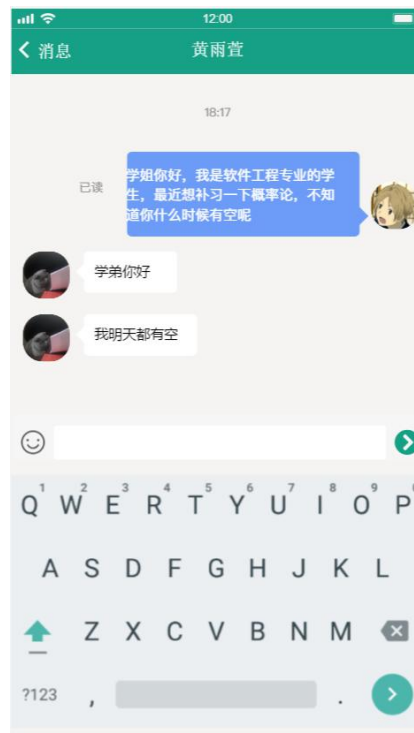
09:41

★ 首页

💬 消息

👤 我的

聊天：



预约:



8. 综合考虑

8.1 可扩展性

为了追求可扩展性，我们会引入一些设计模式，如状态模式、策略模式、模板方法、装饰器模式等，但这些模式不一定是性能友好的。所以，为了性能，我们可能写出一些反模式的、定制化的、不那么优雅的代码，这些代码其实是脆弱的，需求的一点点变动，对代码逻辑可能有至关重要的影响，所以还是不要过早优化，不要过度优化。

8.2 性能分析

本系统应当具备响应时间，吞吐量，并发用户数这些性能指标才能满足用户需求。为了实现性能指标我们可以考虑多加一级缓存，多进程多线程的并发处理，合并批量操作等方法

8.3 防错与出错处理

- 1、程序编写时充分考虑到用户的错误或非法输入，用户的错误操作等情况，并针对相应情况分别进行错误处理。
- 2、程序编写时会充分考虑到系统可能会遇到的各种异常情况，并针对这些情况做出相应的错误处理。
- 3、为了防止数据丢失，系统中应当提供相关接口进行数据库内容的备份、还原和导出；
- 4、为了尽量减少程序的漏洞，对接收的数据要进行适当的处理，比如过滤掉一些特殊的字符，如单撇、双撇、空格等等，必要时在注册时就禁止一些特殊字符。