摘 要

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx。

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx。

**关键词：xxx，xxx，xxx，xxx**

# 目 录

第一章 针对复杂工程问题的方案设计与实现 1

1.1 针对复杂工程问题的方案设计 1

1.2 针对复杂工程问题的推理分析 1

1.3 针对复杂工程问题的方案实现 1

第二章 模块分析

2.1 前端设计

2.2 后端设计

第三章 系统测试 2

第四章 知识技能学习情况 3

第五章 分工协作与交流情况 4

参考文献 5

致谢 6

**说明:**

1. **报告要求8000字以上。**
2. **本模板仅为基本参考，请各位同学根据个人情况进行目录结构扩展。**
3. **报告正文必须双面打印。**

# 第一章 针对复杂工程问题的方案设计与实现

* 1. 针对复杂工程问题的方案设计

（详述方案设计情况。体现针对综合设计课题需求，设计满足特定需求的总体设计和详细设计；体现能够集成单元过程进行软件系统流程设计，对流程设计方案进行优选；体现创新意识。2000字以上）

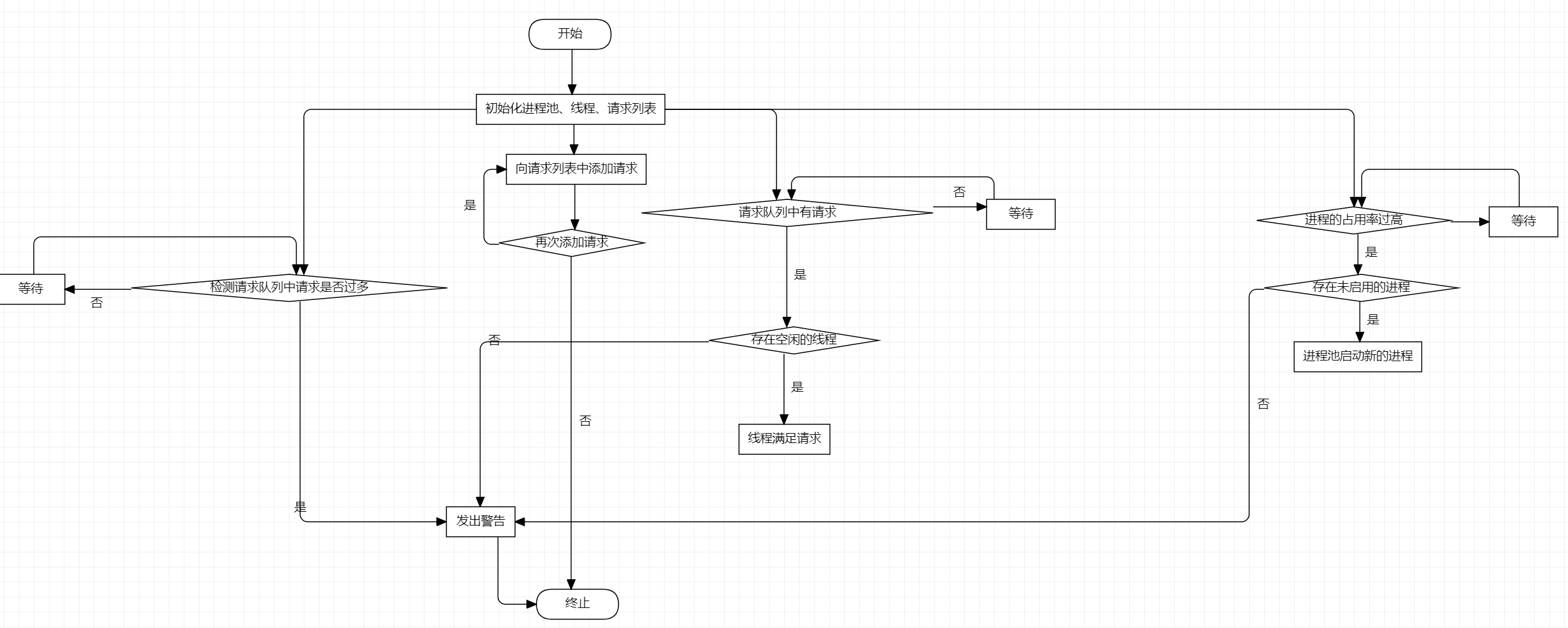
* 1. 针对复杂工程问题的推理分析

1. 确定开发功能

首先这里有两个对象，一个是Client端，一个是Server端。用户能够通过图形化界面设置进程个数；其次，每个服务进程的可服务的请求数量是确定的，当请求数量不断增多时，逐渐启动进程池中的服务进程；当请求数量减少时，则降低服务进程的数量。当请求数量超过进程池中所有服务进程能够处理的最大数量时，则系统给出告警信息；最后，用户能够通过前端界面看到进程的个数变化，以及动态检测进程池的变化。

1. 设计开发

通过前面功能的定义，小组成员绘制了流程图，完成了概要设计。



1. 代码开发

使用python在本地IDE上进行后端的实现，使用微信小程序开发工具实现前端界面的展示。

1. 产品测试

第一阶段，前端设计好后，用户能够通过微信小程序设置进程的数量，同时界面能够显示出当前进程数；第二阶段，当用户不断申请进程后，如果进程池满了，那么用户将在界面上看到警告信息，并且用户无法再申请新的进程。

* 1. 针对复杂工程问题的方案实现

本次课题主要是多进程的平衡调度问题，同时，我们想的还有更多。要求能够自行设置进程的数量以及进程池的动态变化，因此我们打算设计一个图形用户界面，形成一个窗口，而不仅仅是控制台显示的字符界面。在实现的时候，会遇到两个问题：一个是后端实现进程调度平衡的编程语言，一个是实现前端展示的编程语言，同时这两种语言间有很好的接口进行关联。因此，我们最后选择编程工具是Python和微信小程序前端设计语言WXML、WXSS。

Python编程语言最大的优点之一，是其具有伪代码的特质，它可以让我们在开发 Python 程序时，专注于解决问题，而不是搞明白语言本身。针对于本次课题，python中提供了大量的开发模块能够迅速的解决进程的创建和销毁，并且由于 Python 是开源的，它已经被移植到许多平台上。如果能够避免使用依赖系统的特性，那就意味着，所有 Python 程序都无需修改就可以在好多平台上运行，包括 Linux 、Windows、FreeBSD、Solaris 等等，甚至还有 PocketPC、Symbian 以及 Google 基于 Linux 开发的 Android 平台。解释型语言几乎天生就是跨平台的。Python 作为一门解释型的语言，它天生具有跨平台的特征，只要为平台提供了相应的 Python 解释器，Python 就可以在该平台上运行。

因此，鉴于python的优势，我们选择微信小程序这样的高级前端实现。WXML和WXSS的编程方式和HTML于CSS基本一致，甚至可以说是就是从他们演变过来的。同时，使用此开发平台能够让最终的UI界面更方便的展示出来。仅仅需要通过手机微信扫描二维码就能够进入小程序，完成进程的创建、进程数量的变化过程以及进程池的运行状态。

最后，对于技术标准，就是根据python的手册执行的代码撰写。

# 第二章 模块分析

* 1. 前端设计
     1. **前端界面设计**

本项目使用微信小程序 MINA 架构进行编写，整体 UI 使用 Vant 组件库，该 UI 库适合移动端开发，平均组件体积小（平均体积不到1 kb），稳定性高，其单元测试覆盖率高达 90% ，并且支持按需引入和主题自定义设置，且上手容易，故使用该 UI 组件库。

由于本作品是学术类作品，因此我们需要让界面尽可能清爽简洁，满足使用时需求。

小程序的易用性有如下表现：

1.屏幕小：

与Web系统相比，APP安装在手机端，展示屏幕只有几英寸，能够展示的信息就显得非常有限和珍贵，我们确保每个页面功能独立，单个页面上的组件要隔离充足的控件，防止误触，并将有价值的信息放大，放在显眼的位置。

2.页面跳转逻辑：

由于要保证单个功能的独立，我们不同的页面将尽量保证功能的独立，为此我们需要在在跳转不同的功能场景时，做好完善的页面跳转逻辑。混乱的跳转逻辑，会使用户难以操作微信小程序。

3.输入困难化：

需要考虑到在小小的手机屏幕上打字是非常辛苦的一件事，按钮小不好按、又容易出错，因此应该适当减少用户的输入操作；比如可以设置搜索的自动匹配；展示出热门搜索词等，多使用可供使用的预设值。

4.体量压缩：

考虑到移动网络的不稳定性，以及加载速度，微信小程序安装包大小也影响着用户体验，应该尽量做到精简，我们使用分包加载和懒加载技术，极大降低应用的首屏加载所需要的时间，尽量每次仅加载少量的数据包。

* + 1. **前端界面展示**

1.调整参数界面：

图1：调整参数界面

1. 可以通过手动输入进程数和线程的参数达成参数的手工更改；
2. 参数实时更改便于实时观察线程分配实现的效果；
3. 底部导航系统简洁地提供了小程序各个功能的模块，便于快速切换查看；
4. 设置，历史，统计，其他设置四大功能模块，分别负责不同的功能；

2.操作逻辑视图：

图2：操作逻辑图

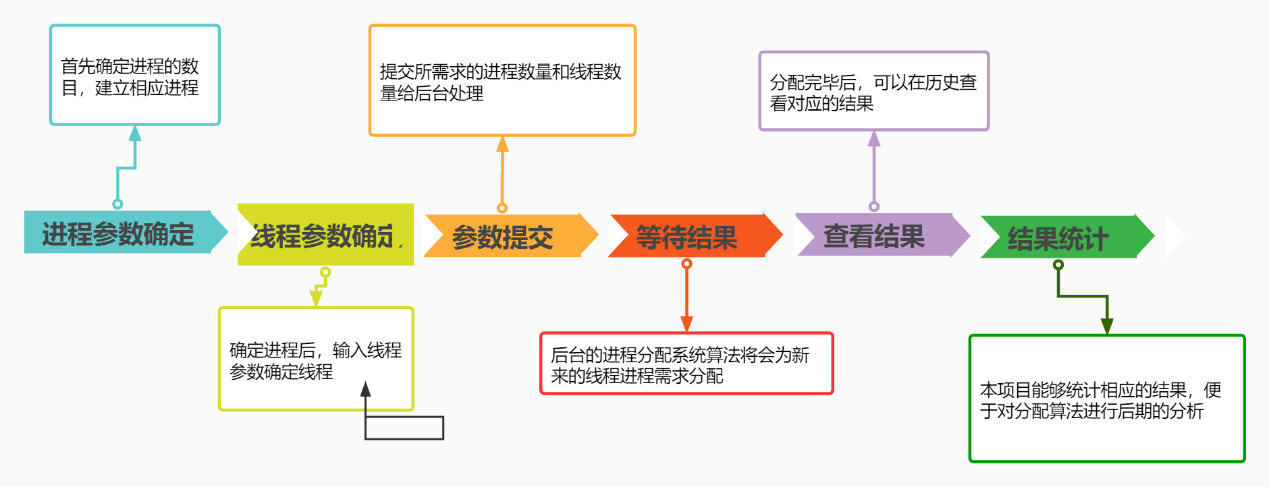
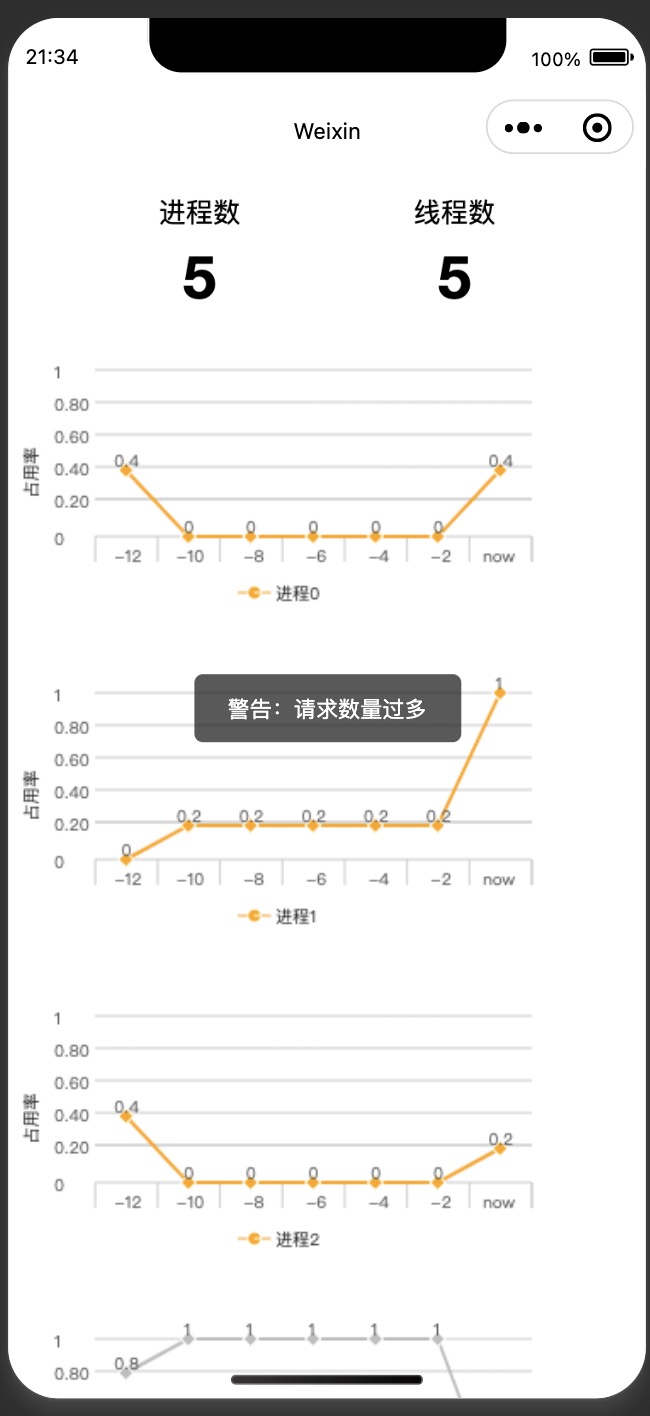


图3: 进程线程参数修改配置流程图

相较于普通参数代码修改的方式，前端修改参数的方法避免了每次测试参数不一致带来的繁琐，提高线程进程分配测试的效率。

3.进程池动态变化图：



通过提示框弹出的形式提示用户当前进程的分配情况，能够使用户快速察觉问题

通过折线图的方式展示进程的状况与分配的情况，直观，清晰，便于观察分析

图4：进程池变化图

在用户操作每一个交互按钮时，不仅要提示操作反馈，也要防止过度反馈。我们的项目使用 “ 轻提示 ” 的方式，使用户在提交表单、填写表单 、 切换页面时，做到相应的提示信息。

4.历史数据查询



可以看到操作的日志：实践，需求的进程数目，线程数目等数据

图5：历史数据查询界面

* + 1. **前端代码实现**

在设计前端页面时，我们将我们的小程序一共分为了5个页面，分别是：

* display页面：实时展示当前运行程序的总体情况：一共有多少线程、一共有多少进程；
* history页面：用于展示我们对于线程分配修改的简单历史记录；
* historyDetail 页面：用于展示我们对于线程分配修改的详细历史记录；
* index页面：主界面，用于设置你想设置的线程数以及进程数。并且你也以通过这个页面跳转到其他别的页面之中去；
* record页面：点击更改进入调整参数的界面。

A picture containing table

Description automatically generated

图6 前端页面代码目录

1. common代码实现

common目录下代码：

Graphical user interface, text

Description automatically generated

图7 common代码目录详情

在common目录中只有一个文件`weui.wxss`这个目录中主要存放的是一些常用的wxss的类。这样使用单独的一个包来管理便于其他包的引用，以及管理各个类的指标。

1. display代码实现

display目录下代码：

Text

Description automatically generated

图8 display代码目录详情

首先每隔X秒主动向<http://127.0.0.1:5000/processStat> 地址发送请求，请求格式如下：



代码1 display代码

其中利用wx.request发送POST请求，将返回得到的数据放入变量message中。并且将整个wx.request包装在Promise对象中，并在结尾返回resolve()，以此来实现小程序中代码的异步执行。

接收到的信息举例如下：

{0: [0, 0, 0, 0, 0, 0], 1: [0, 0, 0, 0, 1, 1], 2: [0, 0, 0, 0, 0, 0], 3: [1, 1, 1, 1, 1, 1],4: [0, 1, 0, 1, 0, 1]}

其中键值对的键是进程的编号，列表中的各个值（除最后一个）代表当前进程对应的线程的工作状态，0代表线程休息，1代表线程工作；列表的最后一个值代表当前进程是不是处于启用状态（0代表未启用，1代表已启用）

在display.js的onshow代码如下：



代码2 display代码

代码的具体工作流程为：收到后台传递的进程状态参数后，将其进行处理（提取单独进程的状况、计算占用率、更新进程对应的占用率列表），针对每一个进程，单独创建一个wxCharts对象，以展示在前台，折线图在wxml文件中对应代码如下：



代码3 display代码

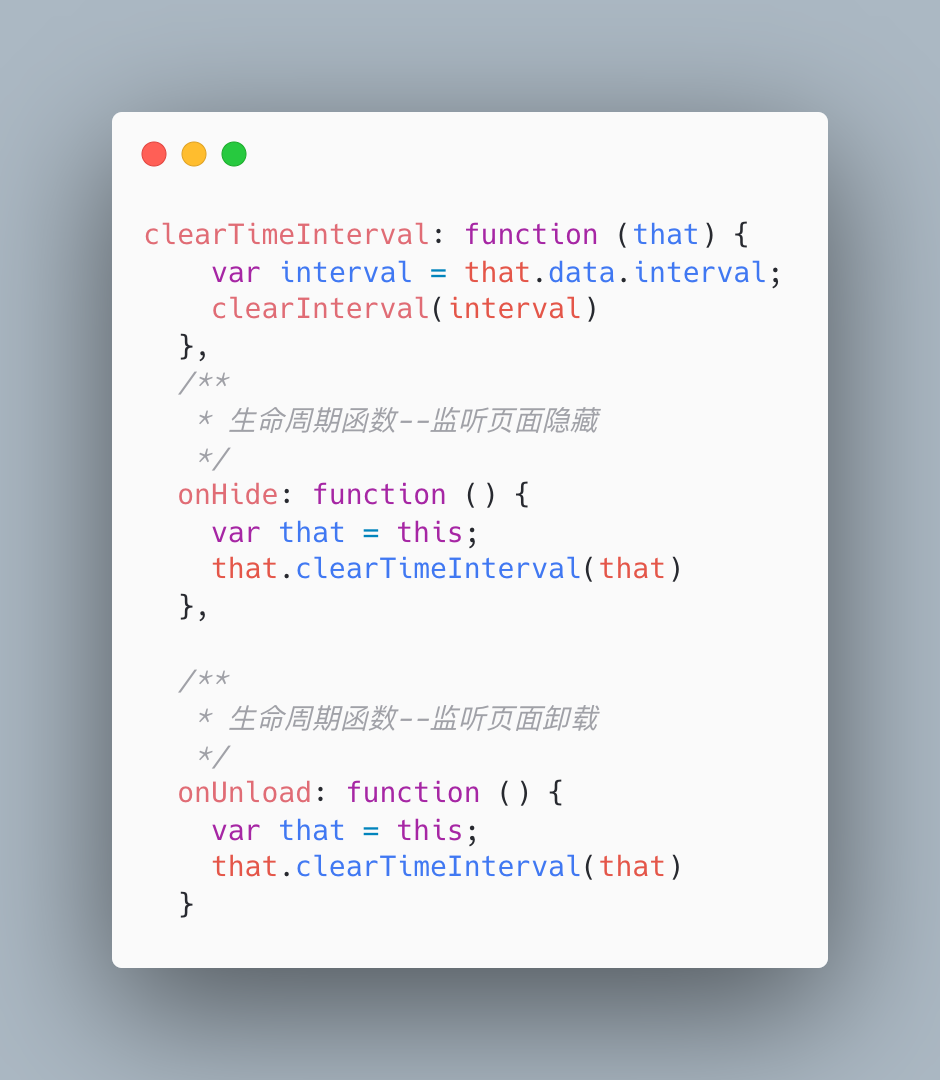
利用wx:for进行循环创建、同时利用canvas-id指定折线图的序号

针对于请求数目过多场景下的提醒需求，同样使用wx.request请求http://127.0.0.1:5000/isSafe，得到是否请求数目过多的信号量，然后利用wx.showToast短暂的弹出弹窗进行提醒：



代码4 display代码

最后将所有的Promise对象装入Timer容器中，并在退出、刷新页面的时候利用clearTimeInterval清空记录和停止发送：



代码5 display代码

1. history代码实现

history目录下代码：

Graphical user interface, text, application, chat or text message

Description automatically generated

图9 history代码目录详情

由于我们已经在前面的章节详细解释了我们对于小程序页面的设计，所以我们这里抛开`history.wxml`以及`history.wxss`两个页面设计的文件。重点介绍该页面逻辑交互部分的代码`history.js`。

`history.js`文件中主要实现了几个函数：

* onLoad: 页面初始化
* onShow: 展示我们所得到的历史数据
* historyToggle：触发是否展示我们的历史记录
* recordStart：准备开始记录我们的修改记录
* recordMove：随着我们记录的增加修改、移动我们的显示

1. historyDetail代码实现

historyDetail目录下代码：

Text

Description automatically generated  
图10 historyDetail代码目录详情

由于我们已经在前面的章节详细解释了我们对于小程序页面的设计，所以我们这里抛开`historyDetail.wxml`以及`historyDetail.wxss`两个页面设计的文件。重点介绍该页面逻辑交互部分`historyDetail.js`的代码：

* onLoad：页面初始化
* recordStart：准备开始记录我们的修改记录
* recordMove：随着我们记录的增加修改、移动我们的显示
* recordEnd：如果记录数量显示到一定数目，则不显示最早加进来的数据
* deleteRecordEvent：删除我们想要删除的记录事件

1. index代码实现

index目录下代码：

Graphical user interface, application

Description automatically generated

图11 index代码目录详情

由于我们已经在前面的章节详细解释了我们对于小程序页面的设计，所以我们这里抛开`index.wxml`以及`index.wxss`两个页面设计的文件。重点介绍该页面逻辑交互部分`index.js`的代码：

* onLoad：页面初始化
* onShow：这里我们使用了POST方法，来获取我们想要的信息。并集中用于展示我们的相关信息。

1. record代码实现

record目录下代码：

Graphical user interface, text, application

Description automatically generated with medium confidence

图12 record目录代码详情

由于我们已经在前面的章节详细解释了我们对于小程序页面的设计，所以我们这里抛开`record.wxml`以及`record.wxss`两个页面设计的文件。重点介绍该页面逻辑交互部分`record.js`的代码：

大部分和前面代码功能类似，这里不再赘述。

* + 1. **整体前后端设计**

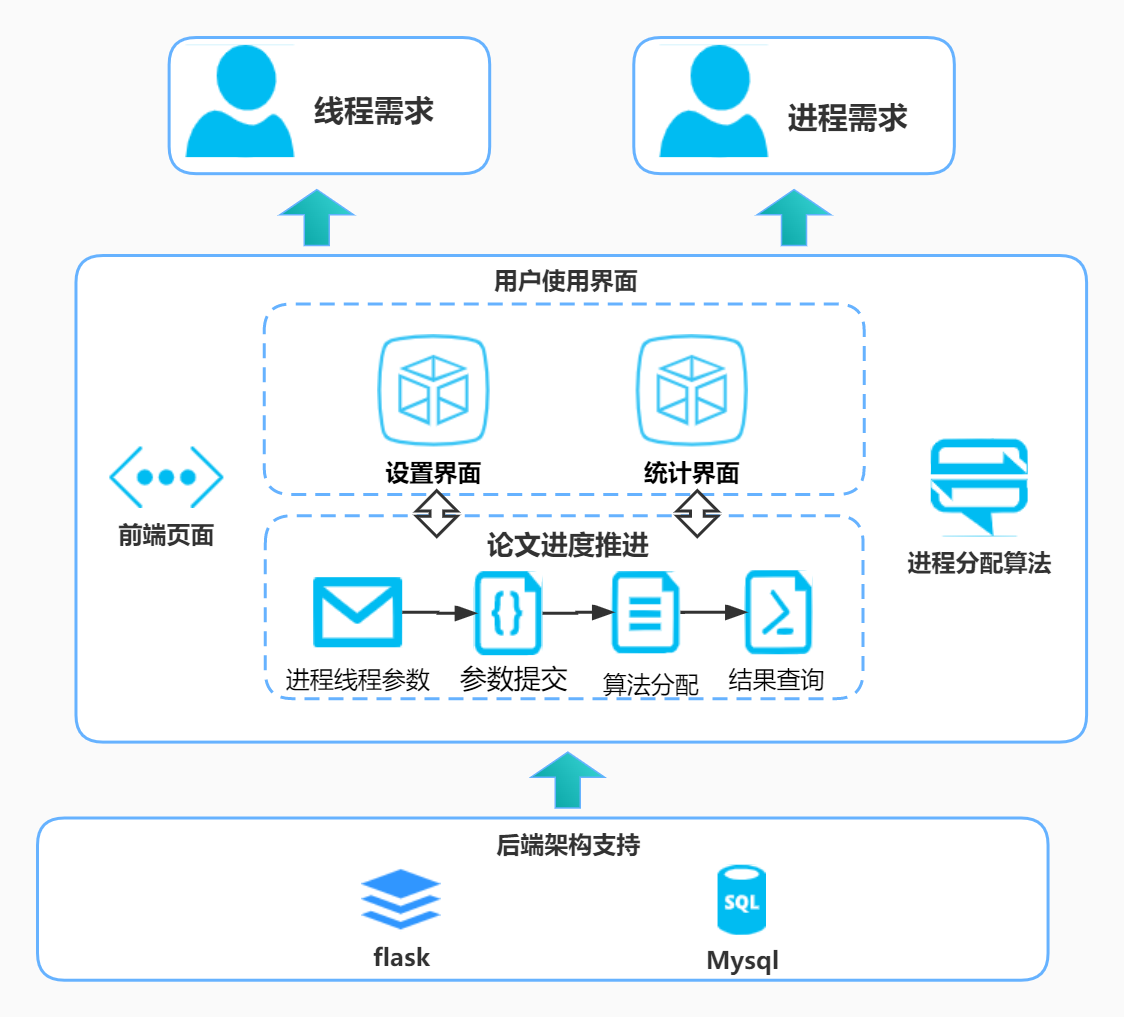


图13 整体设计

* 程序结构采取前后端分离的架构：

1、前端技术：微信小程序。

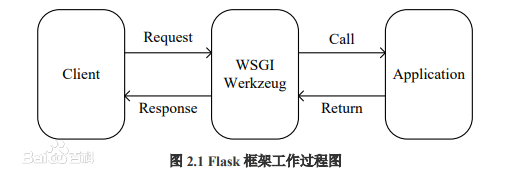
（1）使用公众号网页调试，开发者可以调试微信网页授权和微信JS-SDK 详情

（2）使用小程序调试，开发者可以完成小程序的 API 和页面的开发调试、代码查看和编辑、小程序预览和发布等功能。

2、后端技术：小程序采用了python flask框架， Flask是目前十分流行的web框架，采用Python编程语言来实现相关功能。它被称为微框架(microframework)，“微”并不是意味着把整个Web应用放入到一个Python文件，微框架中的“微”是指Flask旨在保持代码简洁且易于扩展，Flask框架的主要特征是核心构成比较简单，但具有很强的扩展性和兼容性，程序员可以使用Python语言快速实现一个网站或Web服务。一般情况下，它不会指定数据库和模板引擎等对象，用户可以根据需要自己选择各种数据库。Flask自身不会提供表单验证功能，在项目实施过程中可以自由配置，从而为应用程序开发提供数据库抽象层基础组件，支持进行表单数据合法性验证、文件上传处理、用户身份认证和数据库集成等功能。Flask主要包括Werkzeug和Jinja2两个核心函数库，它们分别负责业务处理和安全方面的功能，这些基础函数为web项目开发过程提供了丰富的基础组件。Werkzeug库十分强大，功能比较完善，支持URL路由请求集成，一次可以响应多个用户的访问请求；支持Cookie和会话管理，通过身份缓存数据建立长久连接关系，并提高用户访问速度；支持交互式Javascript调试，提高用户体验；可以处理HTTP基本事务，快速响应客户端推送过来的访问请求。Jinja2库支持自动HTML转移功能，能够很好控制外部黑客的脚本攻击。系统运行速度很快，页面加载过程会将源码进行编译形成python字节码，从而实现模板的高效运行；模板继承机制可以对模板内容进行修改和维护，为不同需求的用户提供相应的模板。目前Python的web框架有很多。除了Flask，还有django、Web2py等等。

* **小程序结构采取前后端交互：**

Flask的基本模式为在程序里将一个视图函数分配给一个URL，每当用户访问这个URL时，系统就会执行给该URL分配好的视图函数，获取函数的返回值并将其显示到浏览器上，其工作过程见图。

****

Flask与前端交互的过程是通过Json字段来完成的，可以完成对前端发送的请求的处理，并将对应的json字符串返回给前端。

# 第二章 系统测试

（详细介绍测试环境搭建情况、测试案例设计情况、测试数据构造情况以及测试结果等。1000字以上）

# 第三章 知识技能学习情况

使用Python作为该课题的主要编程语言，编程工具为 Pycharm ， docker 作为容器为前端提供服务。同时，代码存放在 Github 上，能够让小组中每一个人都能在上面实时更新代码和数据。

1. Python学习

首先由于部分小组成员没有接触过 Python ，就阅读《Python基础教程》大致学习了 Python 的基础语法，同时由于已经了解进程和线程的概念，因此又学习了 Python 的进程和线程调度机制以及如何使用 Python 创建 thread 。

1. Docker

Docker 技术使用 Linux 和内核功能（例如 Cgroups 和 namespaces来分隔进程，以便各进程相互独立运行。这种独立性正是采用容器的目的所在；它可以独立运行多种进程、多个应用，更加充分地发挥基础设施的作用，同时保持各个独立系统的[安全性](https://www.redhat.com/zh/topics/security)。容器工具（包括 Docker ）可提供基于镜像的部署模式。这使得它能够轻松跨多种环境，与其依赖程序共享应用或服务组。Docker 还可在这一容器环境中自动部署应用（或者合并多种流程，以构建单个应用）。此外，由于这些工具基于 Linux 容器构建，使得 Docker 既易于使用，又别具一格——它可为用户提供前所未有的高度应用程访问权限、快速部署以及版本控制和分发能力。

1. Github 学习

GitHub是最大的开源代码托管平台，旨在促进在一个共同项目上工作的个人之间的代码托管、版本控制和协作。 通过该平台，无论何时何地，都可以对项目进行操作（托管和审查代码，管理项目和与世界各地的其他开发者共同开发软件），GitHub 平台为开源项目和私人项目都提供了项目处理功能。

1. logging模块

logging模块是Python内置的标准模块，主要用于输出运行日志，可以设置输出日志的等级、日志保存路径、日志文件回滚等；相比print，具备如下优点：

1. 可以通过设置不同的日志等级，在release版本中只输出重要信息，而不必显示大量的调试信息；
2. print将所有信息都输出到标准输出中，严重影响开发者从标准输出中查看其它数据； logging 则可以由开发者决定将信息输出到什么地方，以及怎么输出。

因此我们采用 logging 模块而不是单纯进行 print 操作。

1. 微信小程序编程语言WXML和WXSS学习

# 第四章 分工协作与交流情况

（对课题的分工协作及团队交流情况进行详细描述。需要体现每位团队成员完成团队分配的工作的情况，在团队中的角色及承担的责任）

4.1 分工情况

1. 前端：前端技术一般分为前端设计和前端开发，前端设计一般可以理解为网站的视觉设计，前端开发则是网站的前台代码实现，包括基本的HTML和CSS以及JavaScript/ajax，最新的高级版本HTML5、CSS3，以及SVG等。前端的发展是互联网自身发展的一个缩影。前端技术指通过浏览器到用户端计算机的统称，存贮于服务器端的统称为后端技术。前端开发主要职能就是把网站的界面更好地呈现给用户。

主要负责人：卞加勉、彭冲

2. 后端：也称为“服务器端开发”，也就是在系统“后面”所发生的事情。同样，在后端服务器和浏览器或应用程序之间存储网站、应用数据和中间媒介的服务器都属于后端。简单来说，在应用程序或网站的屏幕上看不到的所有东西都是前端的后端。它更多的是与数据库进行交互以处理相应的业务逻辑。需要考虑的是如何实现功能、数据的存取、平台的稳定性与性能等。

主要负责人：石佳一、张运宸

3. 文档：主要负责跟进项目，及时将每次会议内容以及工作进程更新到文档中。并且参与了相关软件测试代码和工具的开发。

主要负责人：夏野、钟镇阳

4.2 合作交流

接收到软件工程综合设计并组建团队了解任务后，我们团队坚持每周一小会、每月一大会的合作交流模式。定期跟进计划、更新目标、检查工作运行情况并及时做出调整。

截止到期中软件工程综合设计校内阶段性检查，张运宸和石佳一已经基本完成了课题任务书中Server端程序的功能，并做了简单单元测试。Server端程序能够正常响应多个Client端的请求。与Server端并行的还有报警线程和监控线程。当服务进程不足且进程池中有阻塞的进程时，Server端的监控线程能够线性的启动进程池中的进程，子进程能够自动打开服务线程，从而能够服务更多的进程；当服务进程不足且进程池中没有阻塞的进程时，报警线程将会给出告警信息。后续将进行后端程序的API封装，为前端提供可靠的API接口。除此之外，完善和改进后端程序的，提高程序的鲁棒性。

彭冲和卞加勉已经完成了前端界面的设计，菜单的跳转逻辑，后续将不断完善代码，并与后端同学制定接口设计规则，完成整个软件系统。夏野和钟镇阳同学完成了期中报告的编写，并在不断的跟进项目，编写了用于软件测试的工具，节省了单元测试和后端压力测试的时间。

截止到期末软件工程综合设计校内阶段性检查阶段，我们小组的项目已经基本完成。实现了想要达到的目标，并且通过了软件测试。并且完成了期末最终报告和答辩的准备。

在未来进一步开发中，我们希望我们的小程序可以增加更多的功能并且可以在更多的平台上展示。在此次软件工程综合训练后，我们会继续完善我们的项目。在前端方面更加核心用户的需求，并针对化 的做出功能性优化。后端方向优化我们的代码质量，更加增强鲁邦想，可以让软件更加稳定。

# 参考文献

1. 李振春,刁瑞,韩文功,等. 线性时频分析方法综述[J]. 勘探地球物理进展, 2010, 33(4): 239-246
2. S.G.Mallat. 信号处理的小波导引:稀疏方法[M]. 戴道清,杨力华. 北京,2012,15-16

# 致谢

本报告的工作是在我的指导教师XXX老师的悉心指导下完成的，………….