学校代码：12331

****

毕 业 设 计 说 明 书

基于移动设备的C++在线IDE的设计与实现

|  |  |
| --- | --- |
| **姓 名：** | 栾锟 |
| **学 号：** | 170507250117 |
| **指导教师：** | 张茂红（副教授） |
| **学 院：** | 数据科学与计算机学院 |
| **专 业：** | 计算机科学与技术 |
| **完成日期：** | 2021年4月15日 |

山东女子学院毕业设计作者声明

本人声明：本人完成的毕业设计及呈交的毕业设计说明书是本人在导师指导下取得的成果。对前人及其他人员对毕业设计的启发和贡献已在毕业设计说明书中做出了明确的声明，并表示了谢意。除了特别加以标注和致谢的地方外，本人的毕业设计方案中不包含其他人和其他机构已经实现、发表或者撰写过的成果。

本人同意学校根据《中华人民共和国学位条例暂行实施办法》等有关规定保留本人的毕业设计说明书并向国家有关部门或资料库送交本说明书或其电子版，允许本说明书被查阅或借阅；本人授权山东女子学院可以将本毕业设计说明书的全部或者部分内容编入有关数据库进行检索，可以采用影印、缩印或者其它复制手段汇编本说明书。

作者签名： 日期： 年 月 日

指导教师签名： 日期： 年 月 日

摘要

随着移动智能终端的广泛应用，移动终端正向功能增强化、多模化、定制化、平台开放化的方向发展，移动APP具有不可估量的市场，同时现代教育对编程的普及，越来越多的中学开始进行青少年编程相关课程，而高校方面，越来越多的理工科院校也开始将程序设计语言加入学生培养方案，促使越来越多的学生开始学习程序设计语言也随着云概念的提出，云操作系统，云编译环境这些名词慢慢提出云端编译环境也慢慢流行起来成为未来一大趋势。因此开发一款在线IDE有着无限的发展前景。

本系统采用目前项目开发过程中最流行最方便的浏览器/服务器（B/S）模式，以nginx+uwsgi框架作为本系统开发的Web服务器，前端方面主要使用基于VUE的uni-app框架具有便捷的开发特性，后端使用linux+python flask轻量级框架实现网络交互与信息处理，并通过HbuilderX 作为前端uni-app的开发与调试工具，后端方面使用pycharm的远程调试+ssh进行后端的python代码的编译与调试。在设计上，本系统应具有良好的实时性、交互性、可扩展性、易扩充性和开发性。

在线IDE的详细开发与具体设计，成功实现了云端的编译，本系统的设计与开发能够极大的减轻了开发者对于环境调试的难度，提高了开发的便捷性，实现从传统开发到不受地域设备限制的新开发模式

关键词：IDE; g++; uni-app; nginx; linux; python; flask;

Abstract

With the wide application of mobile intelligent terminals, mobile terminals are developing in the direction of enhanced functions, multi-modality, customization and open platform. Mobile APP has an immeasurable market. At the same time, with the popularization of programming by modern education, more and more middle schools begin to offer programming courses for teenagers. More and more science and engineering colleges also begin to add programming language to student training programs, prompting more and more students to begin to learn programming language. With the introduction of the concept of cloud, cloud operating system and cloud compiling environment, such terms are slowly put forward, and cloud compiling environment is slowly becoming a big trend in the future. Therefore, the development of an online IDE has unlimited development prospects.

This system adopts the most popular and convenient browser/server (B/S) mode in the current project development process, and uses NGINX + UWSGI framework as the Web server for the development of this system. The front end mainly uses the UNI-APP framework based on VUE, which has convenient development characteristics. The backend uses Linux + Python Flask lightweight framework to realize network interaction and information processing, and HBuilderX is used as the development and debugging tool of the front-end Uni-App. The backend uses PyCharm's remote debugging + SSH to compile and debug the Python code at the back end. In the design, the system should have good real-time, interactive, expansible, easy to expand and development;

Online IDE detailed development and specific design, successfully realized the compilation of the cloud, the design and development of the system can greatly reduce the difficulty of developers for environment debugging, improve the convenience of development, to achieve from the traditional development to the new development mode not limited by the regional equipment

**Key words:** IDE; g++; uni-app; nginx; linux; python; flask;

目 录

[1 绪论 2](#_Toc134)

[1.1 研究背景及意义 2](#_Toc29310)

[1.2 国内外研究现状 2](#_Toc12361)

[1.3 论文组织结构 2](#_Toc21057)

[2 相关开发技术及工具 2](#_Toc22273)

[2.1 FLASK 2](#_Toc21173)

[2.2 UNI-APP 2](#_Toc9199)

[2.3 LINUX-SHELL 2](#_Toc9199)

[2.4 开发环境 2](#_Toc31915)

[2.4.1 hbuilderX 2](#_Toc236)

[2.4.2 nginx服务器 2](#_Toc11707)

[2.4.3 pythcarm 2](#_Toc3015)

[3 系统分析 2](#_Toc20649)

[3.1 软件需求分析 2](#_Toc32671)

[3.2 可行性分析 2](#_Toc27100)

[3.2.1 经济可行性 2](#_Toc13818)

[3.2.2 技术可行性 2](#_Toc8337)

[3.3 系统功能分析 2](#_Toc26025)

[4 系统设计 2](#_Toc29367)

[4.1 总体设计 2](#_Toc13756)

[4.2 功能模块设计 2](#_Toc28819)

[4.2.1 用户功能模块 2](#_Toc11530)

[4.2.2 存储功能模块 2](#_Toc1695)

[4.2.3 前后端交互功能模块 2](#_Toc2851)

[5 系统实现 2](#_Toc5959)

[5.1 UNI-APP与Flask交互实现 2](#_Toc14314)

[5.2 注册页面实现 2](#_Toc1415)

[5.3 登录页面实现 2](#_Toc23120)

[5.4 目录页面实现 2](#_Toc32391)

[5.5 编辑页面实现 2](#_Toc32391)

[5.6 报错页面实现 2](#_Toc32391)

[5.7 控制台页面实现 2](#_Toc32391)

[6 结论 2](#_Toc24085)

[6.1 总结 2](#_Toc11426)

[6.2 展望 2](#_Toc30187)

[致谢 2](#_Toc6546)

1 绪论

1.1 研究背景及意义

随着网络发展，云服务的相比于传统PC端环境有着巨大优势：

（1）隔离PC复杂的环境配置；

（2）无需庞大的本地占用，例如小程序与网页端，无需下载就可以完成工作；

（3）随着网速发展，通信技术的进步，降低了网络的延时性，云服务能处理更庞大的数据，且延时性不会太高。

1.2 国内外研究现状

目前国内外云端编译环境已经初步成型，国外不乏一些优秀的在线IDE，Analyzing the learning process (in Programming) by using data collected from an online IDE[1]就对于课程引入在线IDE做了深刻的讨论，而国内，在线编译环境也是相当，特别是国内外的几大算法竞赛OnlineJudge，如Acwing，蓝桥云课，PTA, Leetcode等几大在线评测网站也开始有了在线调试功能，并且由于在线评测系统的特点本身一定程度就有代码的回显功能实现了一定程度上的云存储，可以看出云编译环境市场巨大，特别是Acwing网站 已经开发出了AC Editor这个在线IDE工具，并且有相当高的完成度，支持云存储，编译，调试，报错，智能提示等功能

另一方面讲，随着移动端的发展，移动端化本来就是互联网发展的一大趋势，而当前小程序市场还没有成型的在线编译小程序，因此市场巨大,有着无限发展前景。

1.3 论文组织结构

本文是从基于轻量级flask web框架[2]与uni-app技术和运用各种编程语言设计和开发的在线ide系统，本系统的开发与设计采用的是目前使用最流行的浏览器/服务器（B/S）模式开发，并且现以简单易用的Nginx服务器作为本系统开发和设计的Web服务器，同时使用易学易用，功能强大的flask技术实现，并通过集成开发环境Pycharm平台部署所有的具体项目，集代码运行，调试，测试于一体的所有开发功能。下面详细介绍本文的组织结构：

1. 通过对研究背景的分析以及对国内外的研究现状，得出本系统的设计与开发在当前市场有着一定的重要性和必要性。

第二章 运用曾经学过的所有专业知识和理论知识开发**在线IDE**，本章节主要介绍了**在线IDE**在详细设计与开发的整个过程中需要用到的高级开发语言，开发技术，开发工具和相应的集成开发环境等等，为**在线IDE**的实现提供了充足的软硬件条件和可能性；

第三章 主要从需求分析和可行性分析以及系统功能分析三个方面着重剖析，在详细了解了该系统的实现无论是从个人经济上，还是从个人专业技术上来讲都是可行的，可实施的；

第四章 本章通过第三章对**在线IDE**整体框架的详细需求分析后，根据这些分析得到的结果进行**在线IDE**的总体设计，并根据总体设计这条主线进行详细的功能模块设计，并运用相应的结构图，流程图来进行简单的说明和诠释；

第五章 本章详细介绍了详细设计的实现过程以及各个环节进行详细的设计及代码实现，并对成功开发完成的**在线IDE**的各个功能模块分别进行独立的操作和测试，然后在模块与模块之间进行整体的操作和测试，以期达到最终的系统需求规划。

第六章 本章通过成功完成对**在线IDE**的背景研究，系统分析，系统设计，系统实现等一系列具体步骤的设计与开发，最后进行总结开发经验，总结在本系统具体开发过程中遇到的各类问题，目前存在的缺陷和对以后**在线IDE**的美好展望等等。

2 相关开发技术及工具

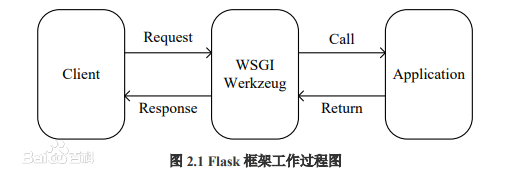
2.1 Flask[3]

Flask是一个轻量级的可定制框架，使用Python语言编写，较其他同类型框架更为灵活、轻便、安全且容易上手。它可以很好地结合MVC模式进行开发，开发人员分工合作，小型团队在短时间内就可以完成功能丰富的中小型网站或Web服务的实现。另外，Flask还有很强的定制性，用户可以根据自己的需求来添加相应的功能，在保持核心功能简单的同时实现功能的丰富与扩展，其强大的插件库可以让用户实现个性化的网站定制，开发出功能强大的网站。

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，其中Diango是Python的框架中使用度最高的，但是Django如同java的EJB（Enterprise JavaBeans JavaEE服务器端组件模型）多被用于大型网站的开发，但对于大多数的小型网站的开发，使用SSH（Struts+Spring+Hibernat的一个JavaEE集成框架）就可以满足，和其他的轻量级框架相比较，Flask框架有很好的扩展性，这是其他Web框架不可替代的。



**图1 flask运行机制**

2.2 linux shell[4]

Shell是系统的用户界面，提供了用户与内核进行交互操作的一种接口。它接收用户输入的命令并把它送入内核去执行。

实际上Shell是一个命令解释器，它解释由用户输入的命令并且把它们送到内核。不仅如此，Shell有自己的编程语言用于对命令的编辑，它允许用户编写由shell命令组成的程序。Shell编程语言具有普通编程语言的很多特点，比如它也有循环结构和分支控制结构等，用这种编程语言编写的Shell程序与其他应用程序具有同样的效果。

Linux提供了像MicrosoftWindows那样的可视的命令输入界面--X Window的图形用户界面（GUI）。它提供了很多桌面环境系统，其操作就像Windows一样，有窗口、图标和菜单，所有的管理都是通过鼠标控制。

每个Linux系统的用户可以拥有他自己的用户界面或Shell，用以满足他们自己专门的Shell需要。

同Linux本身一样，Shell也有多种不同的版本。主要有下列版本的Shell：

Bourne Shell：是贝尔实验室开发的。

BASH：是GNU的Bourne Again Shell，是GNU操作系统上默认的shell。

Korn Shell：是对Bourne SHell的发展，在大部分内容上与Bourne Shell兼容。

C Shell：是SUN公司Shell的BSD版本。

Z Shell：The last shell you’ll ever need! Z是最后一个字母，也就是终极Shell。它集成了bash、ksh的重要特性，同时又增加了自己独有的特性。

2.3 UNI-APP[5]

uni-app 是一个使用 Vue.js 开发跨平台应用的前端框架。开发者通过编写 Vue.js 代码，uni-app 将其编译到iOS、Android、微信小程序等多个平台，保证其正确运行并达到优秀体验。

uni-app 继承自 Vue.js，提供了完整的 Vue.js 开发体验。uni-app 组件规范和扩展api与微信小程序基本相同。有一定 Vue.js 和微信小程序开发经验的开发者可快速上手 uni-app ，开发出兼容多端的应用。

uni-app提供了条件编译优化，可以优雅的为某平台写个性化代码、调用专有能力而不影响其他平台。

uni-app打包到App时仍然使用了5+引擎，5+的所有能力都可以在uni-app中可以使用。在App端运行性能和微信小程序基本相同。

对于技术人员而言：不用学那么多的平台开发技术、研究那么多前端框架，学会基于vue的uni-app就够了。

对于公司而言：更低成本，覆盖更多用户，uni-app是高效利器。uni-app多端演示为方便开发者体验uni-app的组件、接口、模板，DCloud发布了Hello uni-app演示程序（代码已开源，详见Github），Hello uni-app实现了一套代码，同时发布到iOS、Android、微信小程序。

2.4 开发环境

2.4.1 HbuilderX[6]

HBuilderX，H是HTML的首字母，Builder是构造者，X是HBuilder的下一代版本。我们也简称HX。

HX是轻如编辑器、强如IDE的合体版本。

特点：

轻巧

仅10余M的绿色发行包(不含插件)

极速

不管是启动速度、大文档打开速度、编码提示，都极速响应

C++的架构性能远超Java或Electron架构

vue开发强化

HX对vue做了大量优化投入，开发体验远超其他开发工具

详见 按下Alt+鼠标左键可直接打开网址

小程序支持

国外开发工具没有对中国的小程序开发优化，HX可新建uni-app或 小程序、快应用等项目，为国人提供更高效工具

markdown利器

HX是唯一一个新建文件默认类型是markdown的编辑器，也是对 md支持最强的编辑器

HX为md强化了众多功能，请务必点击【菜单-帮助-markdown语 法示例】，快速掌握md及HX的强化技巧！

清爽护眼

HX的界面比其他工具更清爽简洁，绿柔主题经过科学的脑疲劳测 试，是最适合人眼长期观看的主题界面

详见强大的语法提示

HX是中国唯一一家拥有自主IDE语法分析引擎的公司，对前端语 言提供准确的代码提示和转到定义(Alt+鼠标左键)

高效极客工具

更强大的多光标、智能双击，···，让字处理的效率大幅提升

了解HX的极客技巧，详见，更强的json支持现代js开发中大量json 结构的写法，HX提供了比其他工具更高效的操作

详见扩展性

HX支持java插件、nodejs插件，并兼容了很多vscode的插件及代 码块。

还可以通过外部命令，方便的调用各种命令行功能，并设置快捷键。

如果习惯了其他工具(如vscode或sublime)的快捷键，在菜单工具- 快捷键方案中可以切换。

2.4.2 nginx服务器[7]

Nginx (engine x) 是一个高性能的HTTP和反向代理web服务器，同时也提供了IMAP/POP3/SMTP服务。Nginx是由伊戈尔·赛索耶夫为俄罗斯访问量第二的Rambler.ru站点（俄文：Рамблер）开发的，第一个公开版本0.1.0发布于2004年10月4日。其将源代码以类BSD许可证的形式发布，因它的稳定性、丰富的功能集、示例配置文件和低系统资源的消耗而闻名。2011年6月1日，nginx 1.0.4发布。

Nginx是一款轻量级的Web 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器，在BSD-like 协议下发行。其特点是占有内存少，并发能力强，事实上nginx的并发能力在同类型的网页服务器中表现较好，中国大陆使用nginx网站用户有：百度、京东、新浪、网易、腾讯、淘宝等。

2.3.3 pycharm[8]

PyCharm是一种Python IDE，其带有一整套可以帮助用户在使用Python语言开发时提高其效率的工具，比如， 调试、语法高亮、Project管理、代码跳转、智能提示、自动完成、单元测试、版本控制等等。此外，该IDE提供了一些高级功能，以用于支持Django框架下的专业Web开发。同时支持Google App Engine，更酷的是，PyCharm支持IronPython！这些功能在先进代码分析程序的支持下，使 PyCharm 成为 Python 专业开发人员和刚起步人员使用的有力工具。PyCharm的主要功能是：

编码协助

其提供了一个带编码补全，代码片段，支持代码折叠和分割窗口的智能、 可配置的编辑器，可帮助用户更快更轻松的完成编码工作。

项目代码导航

该IDE可帮助用户即时从一个文件导航至另一个，从一个方法至其申明 或者用法甚至可以穿过类的层次。若用户学会使用其提供的快捷键的话 甚至能更快。

代码分析

用户可使用其编码语法，错误高亮，智能检测以及一键式代码快速补全 建议，使得编码更优化。

Python重构

有了该功能，用户便能在项目范围内轻松进行重命名，提取方法/超类， 导入域/变量/常量，移动和前推/后退重构。

支持Django

有了它自带的HTML, CSS 和 JavaScript编辑器 ，用户可以更快速的通 过Djang框架进行Web开发。此外，其还能支持CoffeeScript, Mako 和 Jinja2。

支持Google App引擎

用户可选择使用Python 2.5或者2.7运行环境，为Google APp引擎进行 应用程序的开发，并执行例行程序部署工作。

集成版本控制

登入，录出，视图拆分与合并--所有这些功能都能在其统一的VCS用户 界面（可用于Mercurial, Subversion, Git, Perforce 和其他的 SCM）中得到。

图形页面调试器

用户可以用其自带的功能全面的调试器对Python或者Django应用程序 以及测试单元进行调整，该调试器带断点，步进，多画面视图，窗口以 及评估表达式。

集成的单元测试

用户可以在一个文件夹运行一个测试文件，单个测试类，一个方法或者 所有测试项目。

可自定义&可扩展

可绑定了 Textmate, NetBeans, Eclipse & Emacs 键盘主盘，以及 Vi/Vim 仿真插件。

3 系统分析

3.1 软件需求分析

软件需求分析是开发人员通过前期进行详细且准确的了解用户和项目的具体的功能，安全性，可靠性和性能等各种信息，然后进行认真的调研和准确的分析，其次将用户提出的这些非形式的需求通过简单易懂的软件需求分析语言表述出来并能够正确转化为开发人员能够理解的，完整的需求定义，从而确定该系统必须做什么，能够做什么的一个完整过程。软件需求从定义的层面上分为用户需求层次、功能需求层次和业务需求层次等三类需求，下面我们用一个完整的，具体的软件需求分析过程来了解一下，如图3-1所示。



**图3-1 软件需求分析过程**

3.2 可行性分析

**在线IDE**能够保证在程序编译环境，不受时间，不受空间的限制，在规定的时间内在不同的地点通过互联网就能进行在线编译。下面从开发和设计在线ide的技术可行性和经济可行性两个方面进行系统开发整个过程中的可行性分析。

3.2.1 经济可行性

本系统在详细设计与开发的全部过程中，并没有用到过多，功能过于强大的软硬件设备工具，本系统采用目前最流行也是最简单方便的浏览器/服务器（B/S）模式开发，以当前简单易用的nginx工具作为本系统开发的Web服务器，使用轻量级的flask技术进行后端开发和简便的UNI-APP框架进行前端逻辑设计，并通过开发工具Pycharm平台和ssh工具部署后端项目到服务器并成功运行和调试代码，微信开发者工具来进行具体系统调试。主要的成本在于服务器的维护。所以通过上面的分析得出该**在线IDE**在理论上和实际操作中都具有一定的经济可行性。

3.2.2 技术可行性

**该IDE系统**在详细设计和明确开发的全部过程中，具备一定的专业技术，包括所学到的软件开发语言的语法，软件开发工具的熟练使用和软件开发环境的基本配置等等，在软件开发过程中，本系统需要用到的开发语言诸如python以及typescript编程语言，flask开发技术和vue编程语言，开发环境诸如python3.6，g++,nginx服务器等等，开发工具诸如Pycharm集成开发平台，Hbuilder,以及ssh工具等等，满足**该IDE**在整个开发过程中最基本的技术要求，所以该系统在开发过程实际操作中具有一定的技术可行性。

3.3 系统功能分析

本系统主要面向用户使用。：

用户具有自己存储空间文件及文件夹的增删改查以及访问的权限，

**在线IDE**的设计与开发可以帮助用户能够实现程序语言源代码的编写并存储在云空间中，可以通过编译来获取报错信息，也可以提交输入来获取结果等工作。**在线IDE**也可以让用户从登录界面进行账号信息验证进入到各自相对应的模块界面内，修改个人信息操作，在设计上，本系统应具有良好的实时性、交互性、可扩展性、易扩充性和开发性。

4 系统设计

对给定系统的需求分析进行概述结束后，将对给定系统的开发与设计进行总体概述。在给定系统开发与设计的概述阶段中，将根据该系统各个具体模块功能实现的不同，对系统模块的所有功能进行必要的模块划分操作，并针对这些划分后的模块功能进行详细的设计操作[[[1]](#endnote-1)]。

4.1 总体设计

根据第三章**在线IDE**的系统分析，进行详细的总体设计。本系统的总体设计主要包含：

（1）用户的登录验证，注册以及用户信息的修改

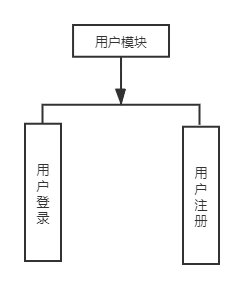
（2）用户存储空间的分配与管理可以实现文件与文件夹的增删改查与访问

（3）前后端交互部分，前端提交代码到服务器接收报错信息，以及前端输入与后端数据返回功能

4.2 功能模块设计

4.2.1 用户功能模块

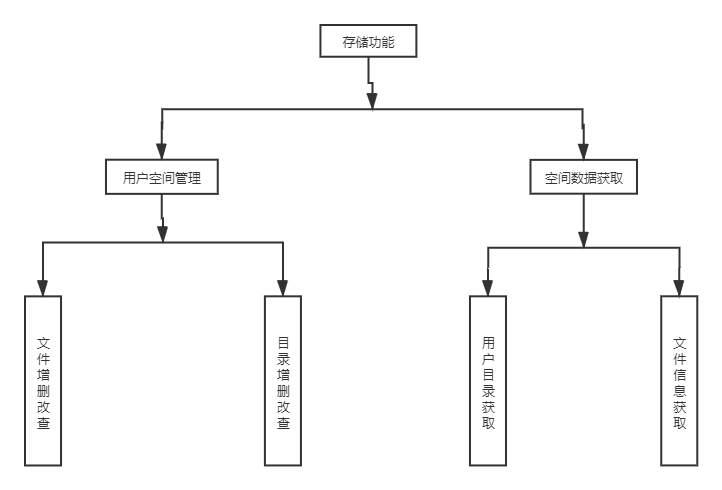
用户模块主要是用户信息的登录验证与注册，如图4-1所示。



**图4-1 用户功能模块**

4.2.2 存储功能模块

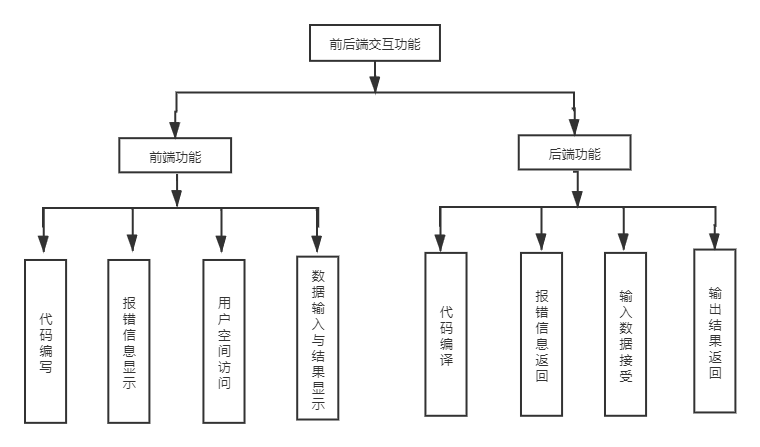
存储功能模块是针对为用户分配的用户空间进行管理，包括用户文件的增删改查，用户目录表获取，用户文件信息获取，如图4-2所示。

****

**图4-2 存储功能模块**

4.2.3 前后端交互功能模块

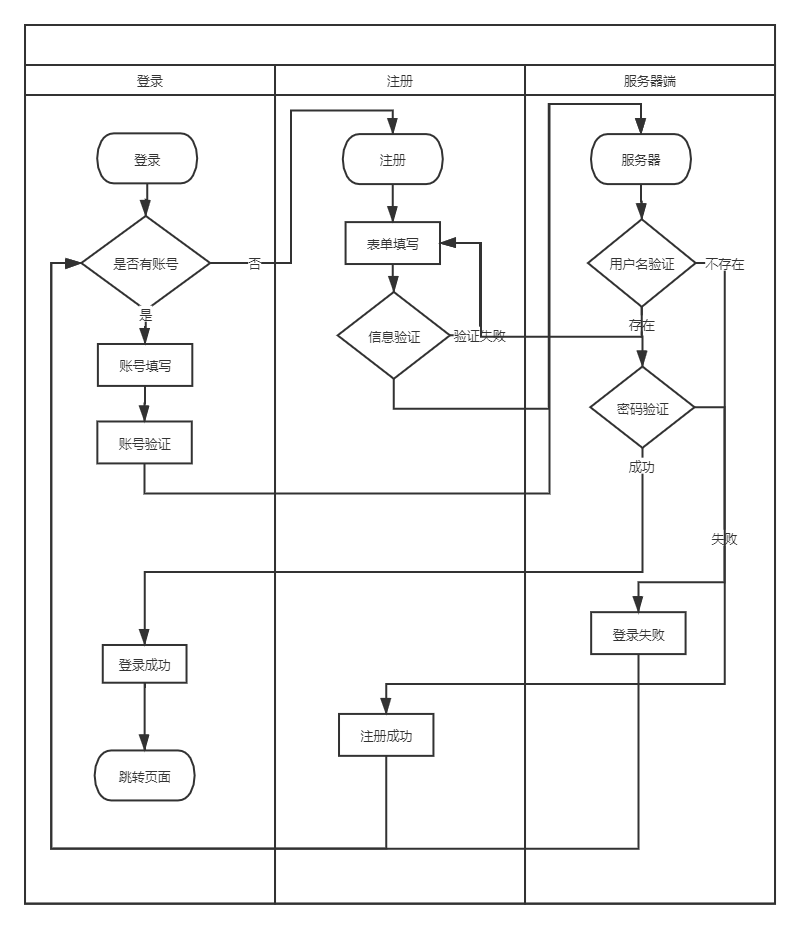
前后端交互功能模块主要分为两部分，前端部分负责编写代码，报错信息的显示，输入数据的编写，输出结果的显示；后端部分负责编译代码，输入数据，如图4-3所示。

****

**图4-3 前后端功能模块**

4.3 系统主要功能流程图

（1）用户登录系统功能流程图，如图4-4所示。用户从前台登录主界面进行学生用户名，密码等数据的验证和选择相应正确登录的权限，或注册之后进行信息的初始化获取后端数据。

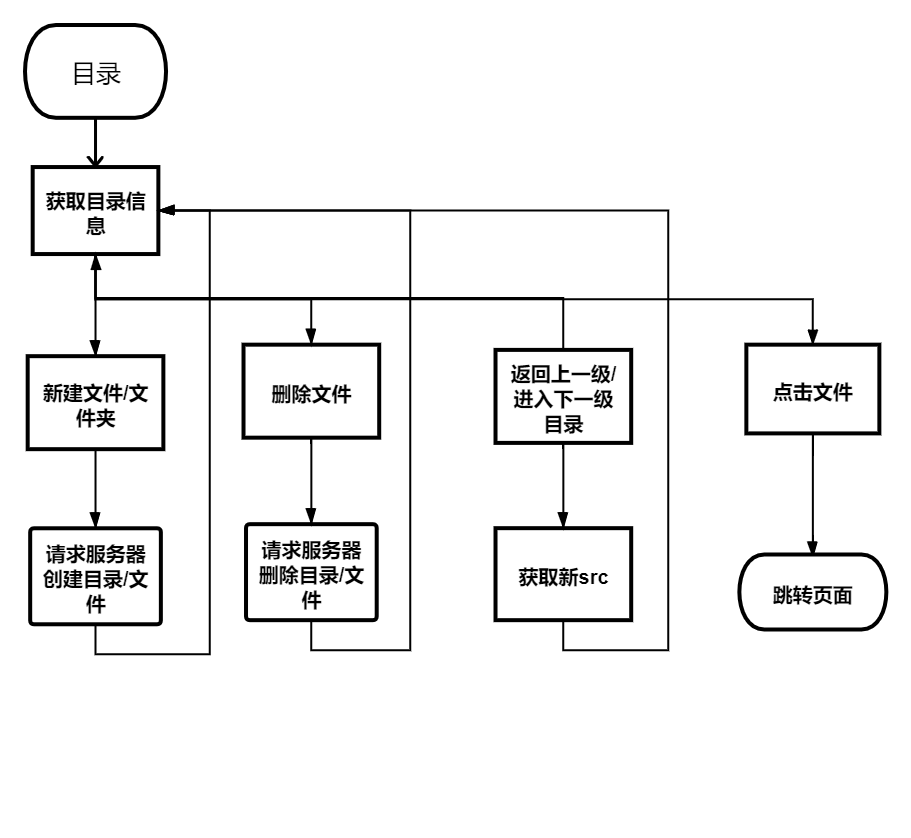


**图4-4 用户登录系统功能流程图**

（2）用户目录功能。

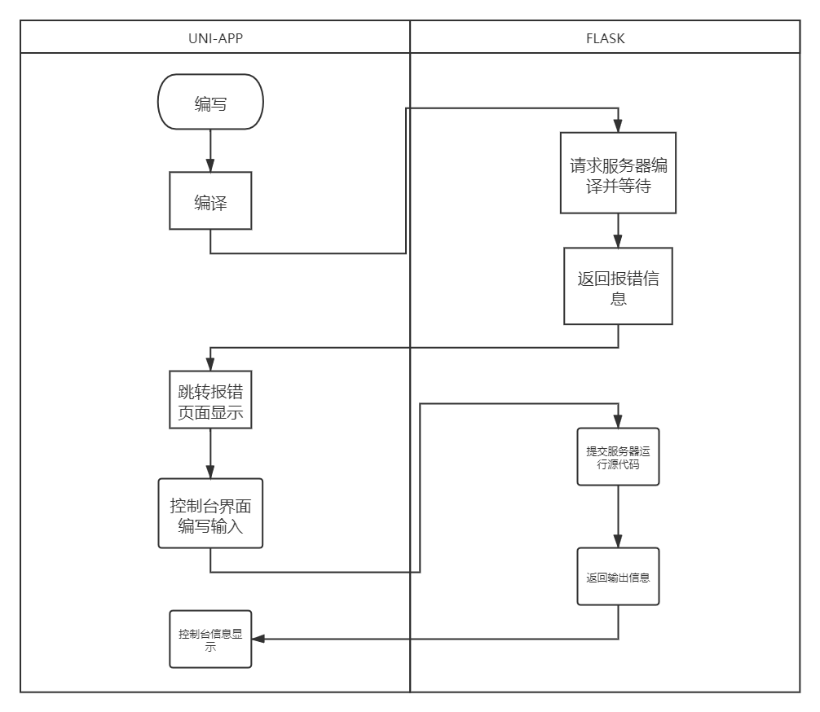
用户目录功能实现了云存储，为用户提供了存储和管理功能

该界面实现了目录信息的可视化的同时也可以让用户实时的管理存储空间，可以对实现目录的多级访问，文件的打开，目录和源文件的添加与删除。

****

**图4-5 目录功能流程图**

（3）前后端功能流程图（如图4-6所示）：考生用户通过前端界面编写代码通过前端请求将代码发送到服务器进行编译返回报错，输入输入数据传送到后台返回输出数据

****

**图4-6 前后端交互功能流程图**

5 系统实现

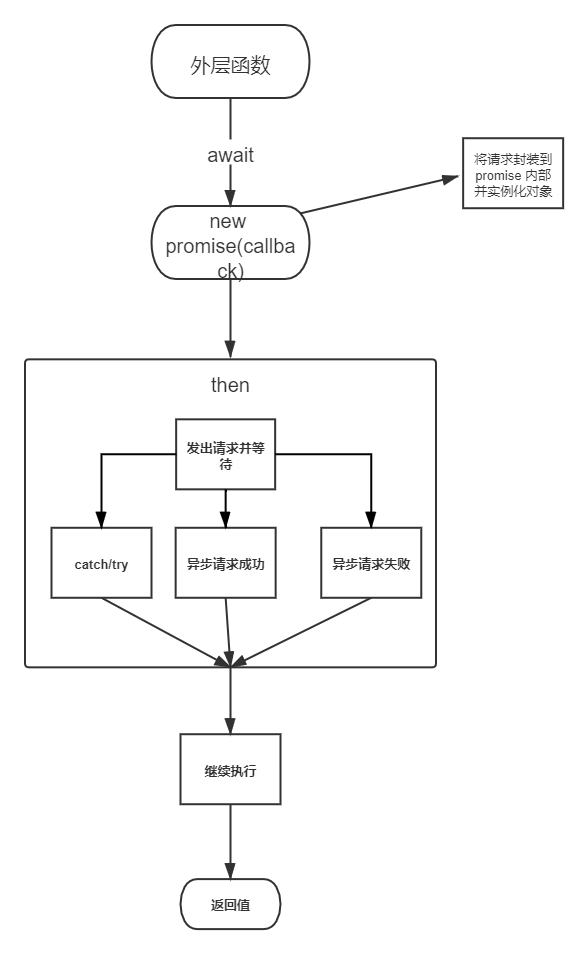
5.1 uni-app 与Flask交互实现

5.1.1 问题背景

在uni-app中，uni.request等许多接口都是异步[9]的，直接使用可能会导致页面渲染完毕时，数据还未成功获取的情况，无论是登录验证与注册申请，还是编译环节的代码提交与报错信息接收，都依赖于同步请求。因此，必须将原有的异步请求转为同步方式[10]。

5.1.2 解决方法

基于状态机原理通过使用Typescript的promise[11]类结合 async[12] 和 await来使异步函数同步化，用户登录系统功能流程图，如图5-1所示。



**图5-1 用户登录系统功能流程图**

实例代码：

public async register(username:string,Password:string):Promise<Boolean>{

···

return await this.userexist()!=true;

}

public async userexist():Promise<Boolean>{

···

return new Promise((resolve, reject) =>{

uni.request({

···

});

}).catch((e)=>{});

}

async function reg():void{

if(await register.register(tip.model.name,tip.model.password))

{

···

}

else{

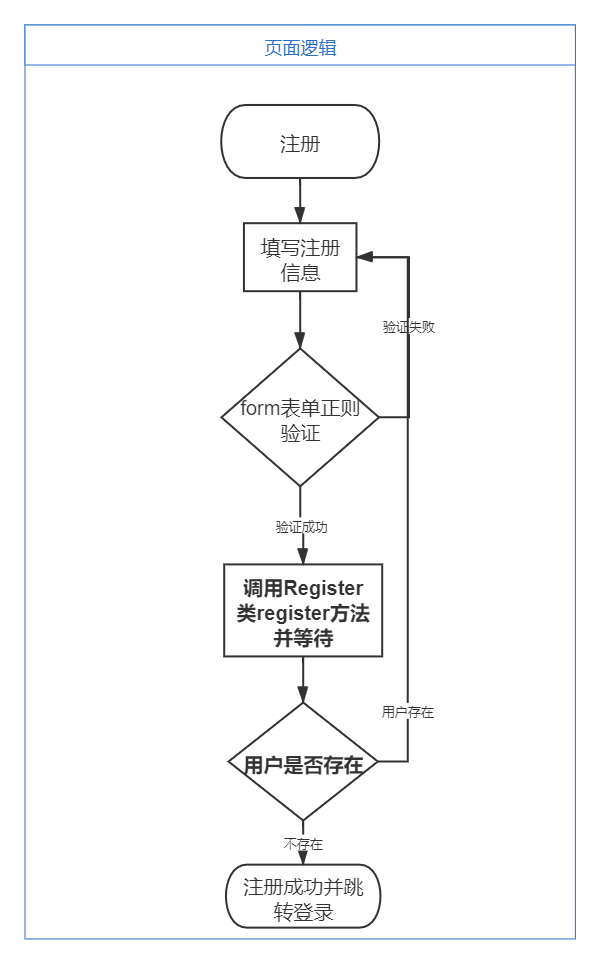
···

}

}

5.2 注册实现

注册页面负责注册信息的填写。通过form表单的正则规则，当失去焦点就对数据进行校验，校验失败会出现提示信息，当提交时submit函数会使用await调用Register类的register方法来进行注册，并根据返回值进行相应的页面跳转或者信息提示。用户注册页面逻辑流程图，如图5-2所示。



**图5-2 用户注册页面逻辑**

正则验证规则

rules: {

name: [{

required: true,

message: '请输入用户名',

trigger: ['change', 'blur'],

},

{

pattern: /^[a-zA-Z\_][0-9a-zA-Z\_]{5,14}$/,

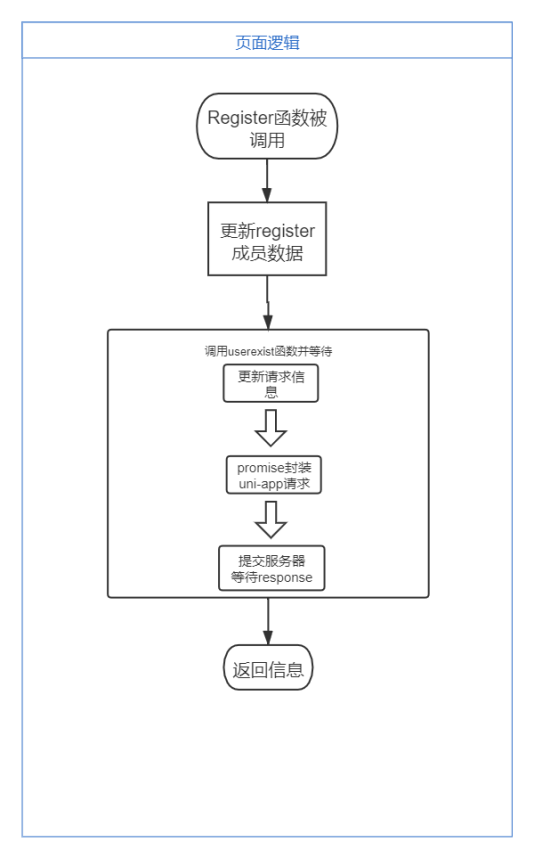
message: '需同时含有字母和数字，长度在6-12之间',

trigger: ['change', 'blur'],

}

],···

Typescipt层在user.ts中构造了Register类，将与注册相关的方法封装在类中，register方法通过使用promise构造封装http请求的方式来实现同步请求，调用userexist方法来获取当前注册用户名是否存在，完成注册并返回到页面相应信息。用户注册typescript逻辑，如图5-3所示。



**图5-3 用户注册typescript逻辑**

public async userexist():Promise<Boolean>{

this.src = getsrc(this.username);

this.root = this.src+"space/";

return new Promise((resolve, reject) =>{

uni.request({

url: 'http://www.cloudcd.top:8080/register',

data: {

user:this

},

method: 'POST',

header: {

'custom-type': 'application/json'

},

success: (res) => {

resolve(res.data=="exist");

},fail:(res) => {

reject(true);

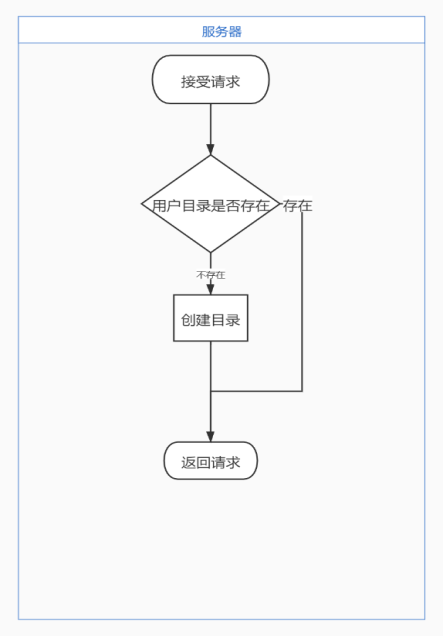
}

});

}).catch((e)=>{});

}

服务器接收请求，并判断用户目录是否存在？如果不存在，就创建用户目录。并且生成一个文件名为密码的文件夹，用户注册服务器逻辑图，如图5-4所示。



**图5-4 用户注册服务器逻辑**

5.3 登录页面实现

（1）登录按钮实现登录页面跳转。点击登录按钮，首先进行用户名和密码字段的正则验证，如果验证成功则提交到服务器进行用户验证，服务器端先查找用户空间是否存在？如果存在，再检查用户密码文件夹是否与密码一致并返回相应的数据，如图5-5所示。

**图5-5 登录页面**

登录按钮实现

async login() {

let flag = await this.user.login();

if (flag == false) {

showtip("用户不存在或密码错误", "error");

} else if (flag == true) {

showtip("登录成功", "success");

switchtotabar("/pages/home/home")

} else {

showtip("用户名或密码格式不正确", "warning");

}

}

Flask服务器login验证：

def userexist(user):#检测用户是否存在 用户空间是space 密码文件时password

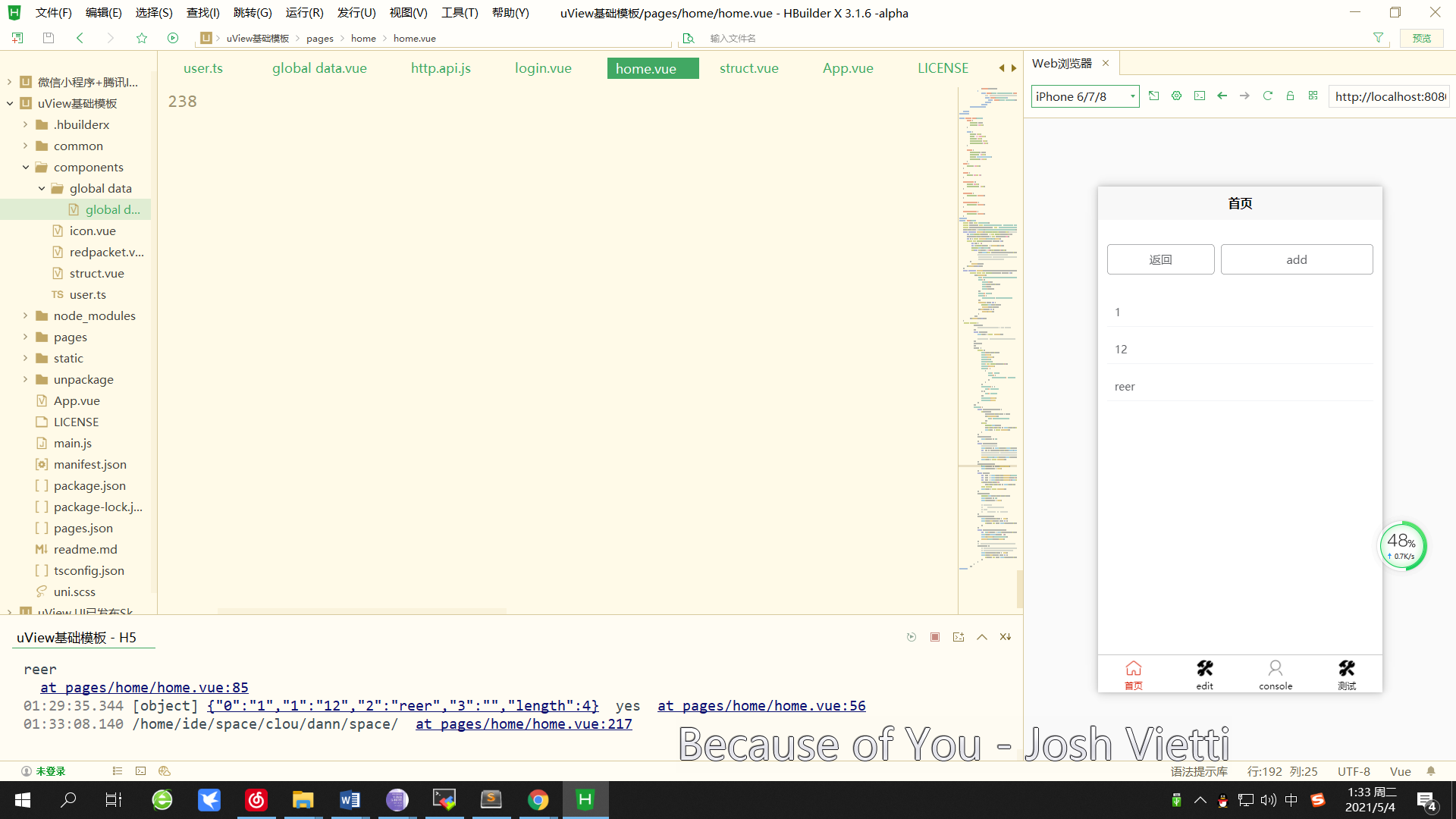
if file.fileexist(user["root"]) and file.fileexist(user["src"]+user["password"]+"/"):

return True

return False

5.4 目录页面实现

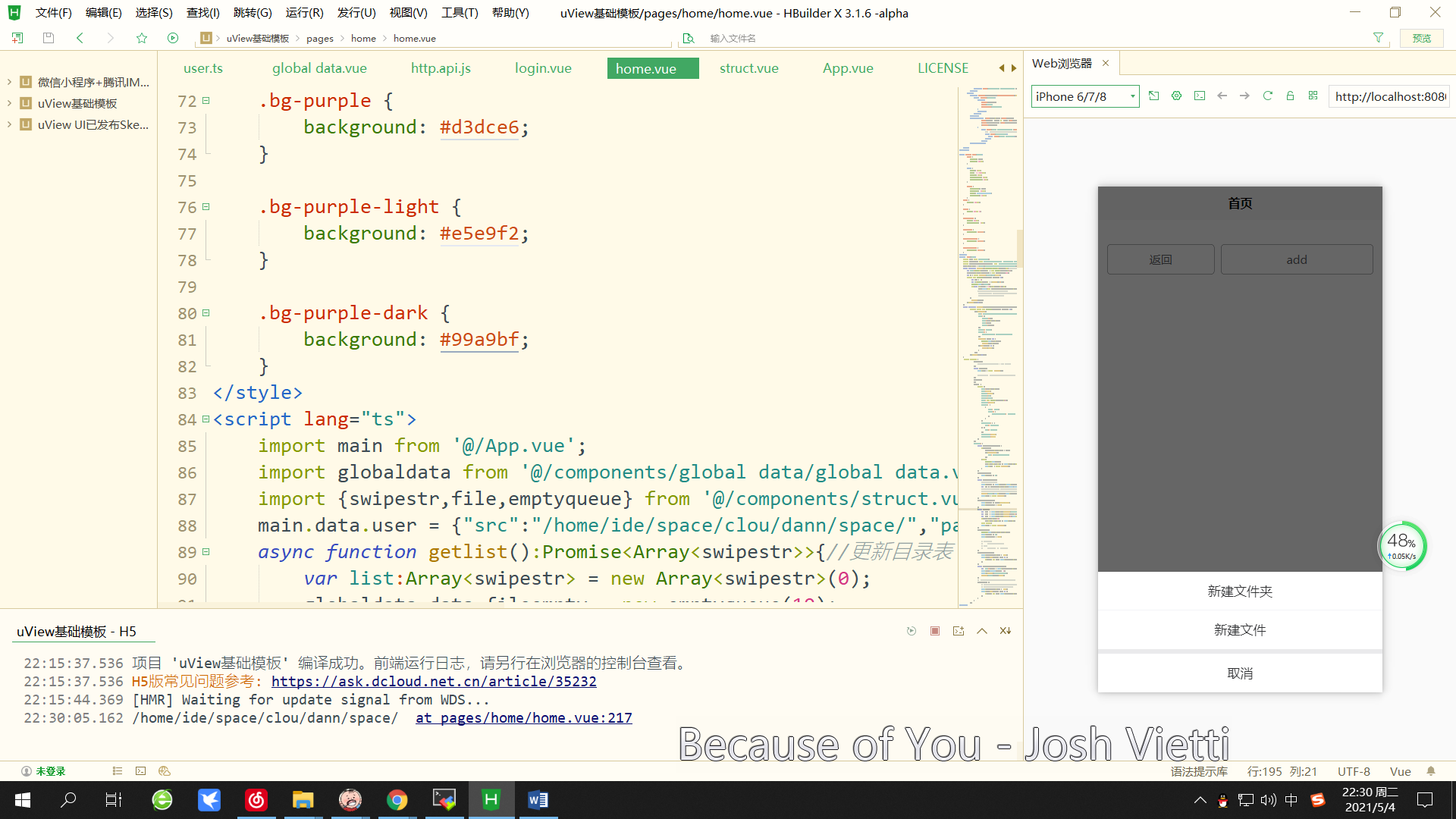
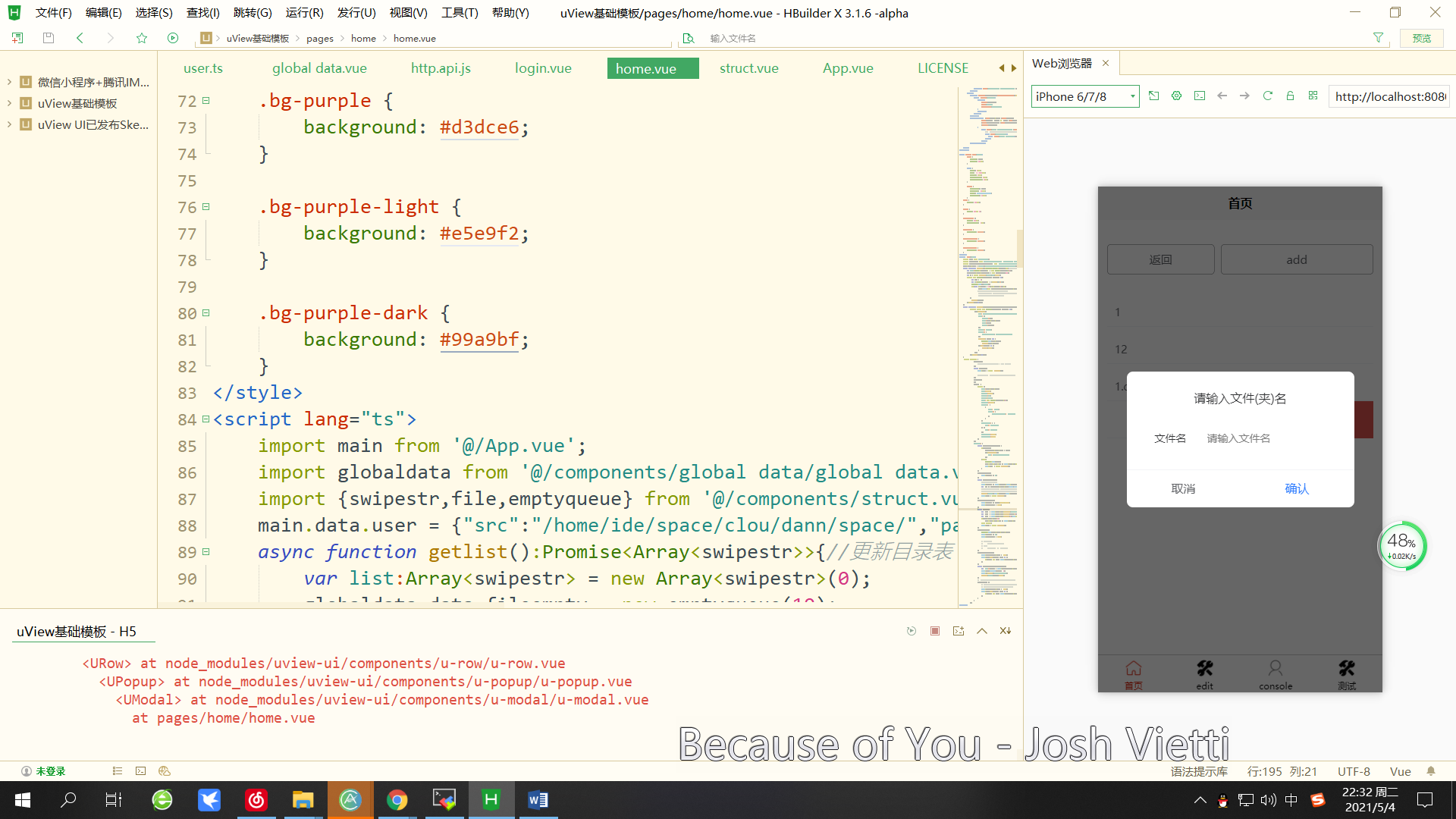
目录页面主要布局采用的是swipe窗口，方便用户通过滑动来进行目录的管理，包括目录与文件的添加和删除功能，多级目录与文件的访问功能如图5-6所示。



**图5-6目录页面**

（1）Add按钮：实现文件或者文件夹的新建

通过对u-action-sheet操作菜单和u-modal模态框的控制，当点击按钮时弹出操作菜单，之后根据选择的是文件还是文件夹来判断添加的是文件夹还是源文件，进而弹出模态框进行文件名的输入，当用户点击确认时，前端将当前的目录信息与文件名发送到服务器，通过flask app请求flask调用linuxshell创建相应的目录或是源文件

**图5-7新建页面**

Flask关键代码：

由于创建目录和新建文件的不同所以会通过isfile标记来判定当前新建的是目录还是文件

def addfile(src,isfile,name):

if isfile=='true':

subprocess.Popen("touch "+src+name,shell=True)

else:

subprocess.Popen("mkdir "+src+name,shell=True)

前端关键代码：

当菜单被点击时，传入index从而判断当前新建类型是文件还是文件夹，并且让模态框显示

sheet\_click(index){

this.addtype = index?true:false;

this.modal\_show = true;

},

当模态框被点击时，根据文件类型修改文件名，并向服务器发送文件添加请求且等待，请求完成后进行目录更新

async modal\_confirm(){

//addtype为真是文件夹

this.addname = this.addtype?this.addname+".cpp":this.addname;

await request("add",this.addname,this.addtype);

this.list = await getlist();

},

（2）返回按钮：返回到上级目录

通过判定user类中的src成员和root成员是否相等来判定当前用户是否在根目录上，如果在根目录则不进行返回上级操作,否则向服务器发出请求，通过linux的cd ..命令返回上级目录并返回目录表

async back(){

if(this.user.src==this.user.root)

return;

this.list = await back();

},

（2）点击：打开指定文件或者进入指定目录

目录表的数据结构成员由文件名和文件类型组成，因此每次从服务器获取目录信息后都会根据文件名判断当前文件属于文件夹还是源文件

目录获取代码：

l[i].indexOf(".cpp")>-1?true:false 对于当前字符串查找文件名后缀如果存在说明对当前文件属于文件，

async function getlist():Promise<Array<swipestr>>{//更新目录表

var list:Array<swipestr> = new Array<swipestr>(0);

globaldata.data.fileempty = new emptyqueue(10);

let s = await request("get","",false);

return new Promise((resolve, reject) =>{

var len = 0;

var l:Array<string> = s.split("\n");

for(var i:number = 0;i<l.length-1;i++){

list.push(new swipestr(l[i],globaldata.data.fileempty.dequeue(),l[i].indexOf(".cpp")>-1?true:false))；

}

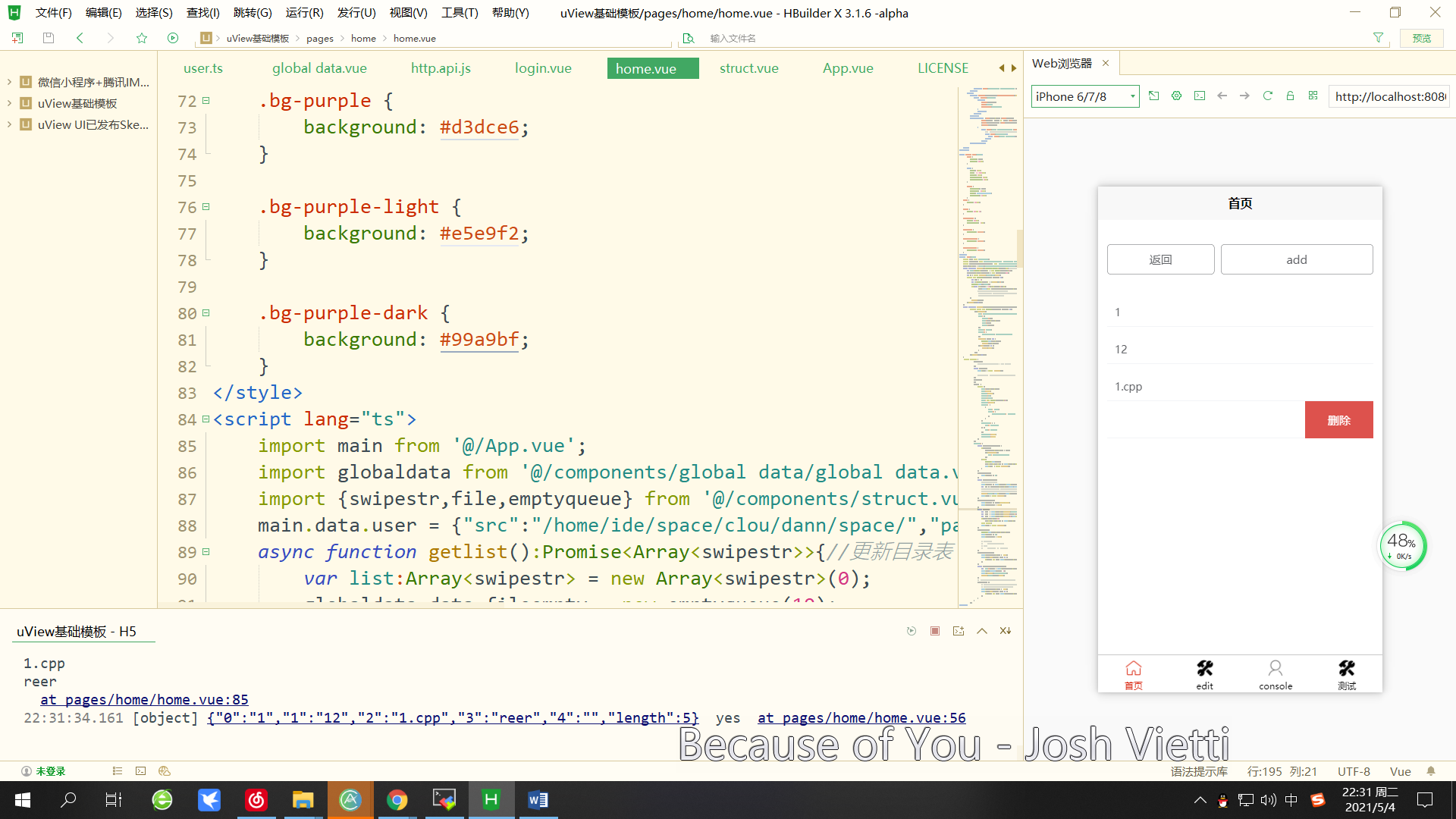
resolve(list);

}).catch((e)=>{});

};

（3）左滑删除按钮：删除选定的文件或文件夹

当用户左滑点击删除时，会先获取当前文件名并向服务器发送请求删除，当删除文件夹时由于当前删除文件夹可能内部包含多个子文件夹，因此需要进行递归删除



**图5-8文件删除**

Flask删除关键代码：

def delfile(src):

subprocess.Popen("rm -rf "+src,shell=True)

5.5 编辑页面实现

该页面通过输入框来实现代码的输入分别有返回、run、报错、控制台等四个按钮。

（1）返回：通过route路由跳转来实现返回到目录页面；

（2）Run:提交代码到服务器进行编译并生成源文件；

前端负责将当前界面的代码拼接成字符串，并将代码发送到服务器进行编译

服务器端负责将文件先写入到用户目录中，之后调用g++编译器进行编译并把源程序生成到文件名为用户密码的文件夹中，并返回报错信息到全局变量中存储。

function getcodelist(list:Array<codelist>):string{//代码拼接

var len = list.length;

var s:string="";

for(var i:number=0;i<len;i++){

s=(s+list[i].data+"\n");

}

return s;

};

编译命令：

uni.request({

url: 'http://www.cloudcd.top:8080/complier',

data: {

request:"complier",

user:main.data.user,

filename:this.filename,

code:getcodelist(this.list)

},

method: 'POST',

header: {

'custom-type': 'application/json'

},

success: (res) => {

globaldata.data.default = res.data.split("\n");

}

})

Flask端编译功能

def complier(user, code, filename):

file = open(user[“src”]+filename, "w")

file.write(code)

file.close()

default = open(user[“root”]+user[“password”]+"default","w")

result = subprocess.Popen("g++ " + user[“src”] + filename + " -w -Wall -O3 -o " + user[“root”]+ user[“password”]+"/app ", shell=True,

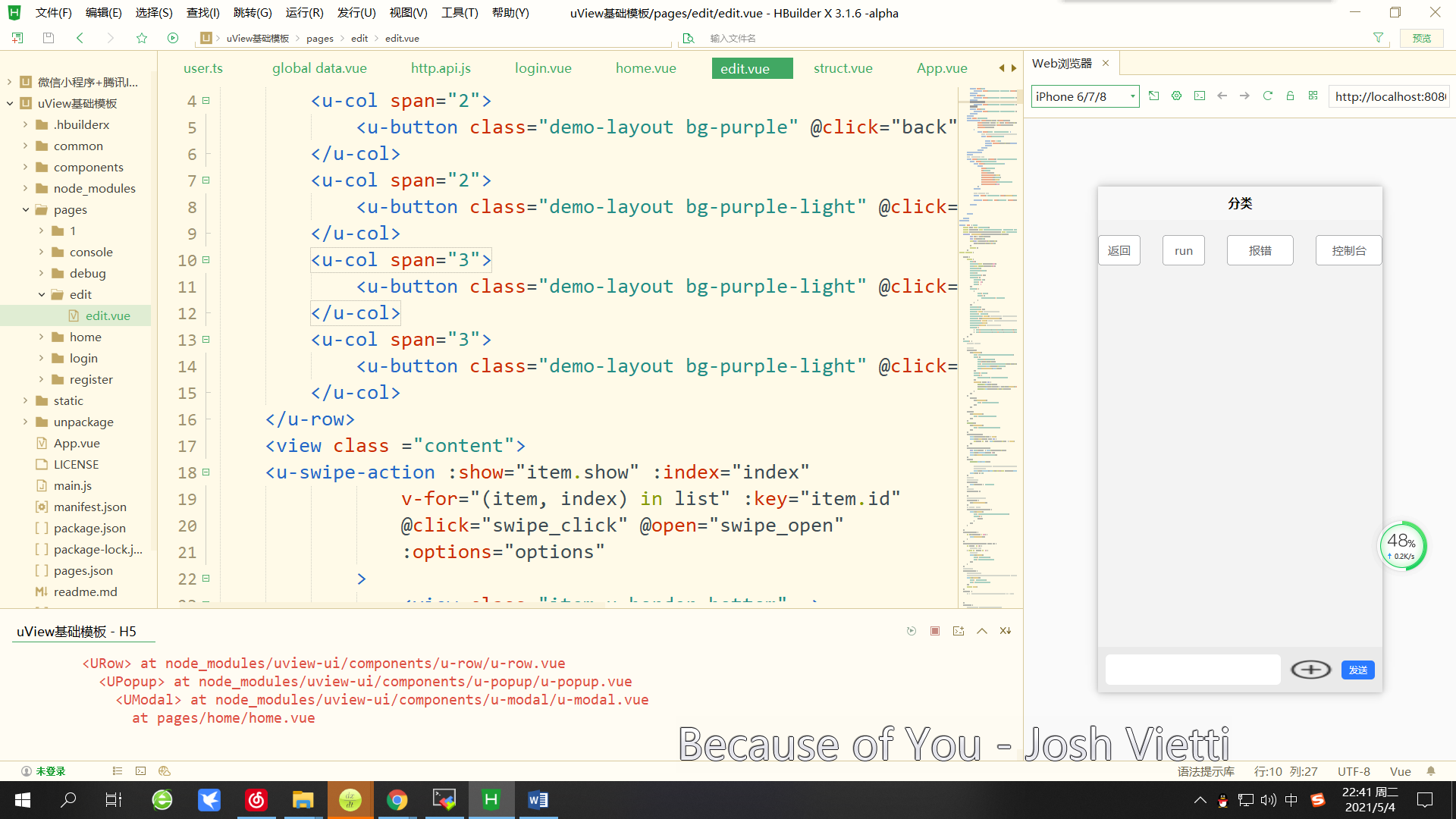
stderr=subprocess.PIPE)

s = result.communicate()[1]

default.close()

（3）报错：跳转报错页面检查报错信息；

（4）控制台:跳转控制台页面



**图5-9编辑页面**

5.6 报错页面实现

报错页面主要负责报错信息的显示，便于用户对程序进行debug

在编辑页面进行编译时小程序就从flask获取到了报错信息，并保存到全局变量中，因此每次打开页面就对全局变量中的报错信息进行加载，容器选用ucellgroup进行循环渲染。

其关键代码:

前端实现:

<view class="wrap">

<view flex-direction="columm">

<u-cell-group :show="item.show" :index="index"

v-for="(item, index) in list"

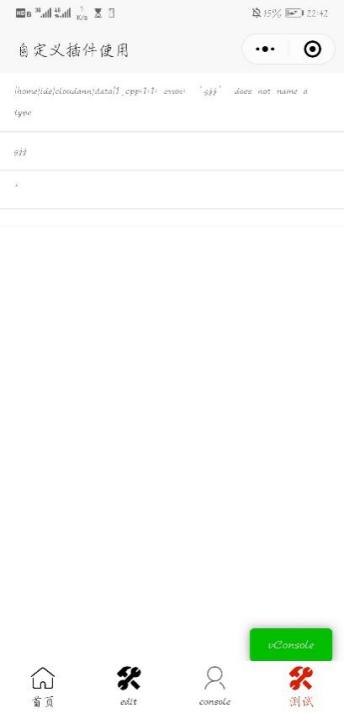
>

<u-cell-item :title=item :arrow="false"></u-cell-item>

</u-cell-group>

</view>

</view>

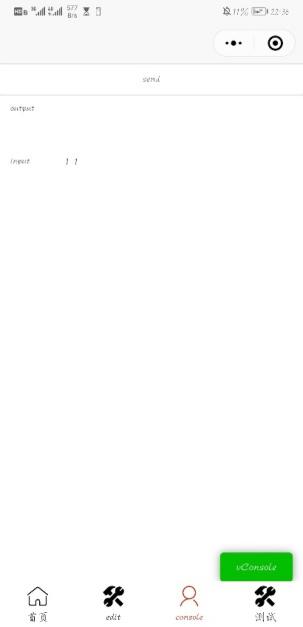


**图5-10 报错页面**

5.7 控制台页面实现

该页面就是处理数据的输入，与运行结果的输出，采用filed控件来获取多行的输入，当按下发送键时将数据发送到服务器获取输出信息。

前端界面如图5-1-1所示。通过field组件完成数据的输出与显示，当按下send按钮时，获取input组件的数据并向服务器flask发送数据并请求运行当前的源代码并返回相应的结果，从而显示在output上，



**图5-11 控制台界面****前端**

Flask端负责与linux shell的交互，flask app接受到前端的请求后，开始调用complie的run函数，先将前端接收到的输入数据写入到filein文件中，之后通过subprocess来命令linux shell来运行之前编译环节在用户空间中生成的源程序，并通过文件流的方式将filein文件中的输入导入当前执行的文件中，并获取输出并返回前端

Flask端关键代码：

def run(src,inpu):

filein = open(src+"filein","w")

filein.write(inpu)

filein.close()

filein = open(src+"filein","r")

result = subprocess.Popen(src+"app",shell=True,stdin=filein,stdout = subprocess.PIPE);

s = result.communicate()[0]

if s == None:

s = ""

filein.close()

return s

6 结论

## 6.1 总结

本文围绕在线ide的详细设计和开发，通过了解在线ide的研究背景，以及在线ide目前在国内外的研究现状，得出在线ide的设计与开发的可行性和必要性，首先，本文主要从在线ide的开发技术，开发工具和相应的开发环境分析着手，为在线ide的实现提供了充足的条件和可能性[[[2]](#endnote-2)]。其次，从在线ide的需求分析和可行性分析以及系统功能分析三个方面着重剖析了在线ide实现无论从经济上，还是技术上都是完全能实现的，为在线ide的详细设计与开发提供了可行性。最后，通过对在线ide的系统设计方面展开论述，通过对在线ide的三类用户的功能模块设计并通过编码和布局实现其主要功能。

同时，本系统采用目前最流行的，也是最简单方便的浏览器/服务器（B/S）模式开发，以当前简单易用的nginx工具作为本系统开发的Web服务器，使用易学易用，功能强大的flask[13]与uni-app框架并通过开发工具pycharm与微信开发者工具[14]以及mobaxterm[15]部署具体项目并成功运行和调试代码，使得系统具有较强的可移植性，较低的开发费用和一定的系统稳定性。

## 6.2 展望

经过测试，在线ide基本上已实现系统设计的要求，但是也存在一定的不足之处，比如界面不完美，安全性不高，并且现在开发出来的在线ide对还没有完全良好的移动端特性，希望随着移动设备的发展会有更多的可能。

参考文献

[1] Espana-Boquera S , D Guerrero-Lopez, Hermi Da -Perez A , et al. Analyzing the learning process (in Programming) by using data collected from an online IDE[C]// International Conference on Information Technology Based Higher Education & Training. IEEE, 2017.

[2] Grinberg M . Flask web development : developing web applications with Python[J]. O'Reilly Media, Inc. 2018.

[3] 叶锋. Python最新Web编程框架Flask研究[J]. 电脑编程技巧与维护, 2015(15):27-28.

[4] Wright C S . Starting to Write Your Own Linux Shellcode[J]. Social Science Electronic Publishing, 2018.

[5] 陈思, 冷雪. 微信小程序开发方式对比[J]. 电子制作, 2020,000(002):52-53,22.

[6] 许溜溜. 基于HBuilder快速开发移动端APP的设计与实现[J]. 电脑知识与技术, 2020, v.16(10):80-81.

[7] 戴华. 基于Nginx和Memcached的高并发WEB服务器设计[D]. 复旦大学, 2013.

[8] Chittibabu P . PyCharm IDE – A Handbook of Python Programming[M]. 2020.

[9]姚成昊. 基于Promise的链式异步请求处理方法,装置及电子设备:, CN109669720A[P]. 2019.

[10]熊俊漉. 基于React的Web前端组件化研究与应用[D]. 重庆邮电大学, 2018.

[11]周安辉.Node.js异步编程模式探讨[J].四川职业技术学院学报,2018,28(04):149-154. [1]熊俊漉. 基于React的Web前端组件化研究与应用[D]. 重庆邮电大学, 2018.

[12] Trevor Burnham. JavaScript异步编程[M]. 人民邮电出版社, 2013

[13]熊俊漉. 基于React的Web前端组件化研究与应用[D]. 重庆邮电大学, 2018.

[14]王婷婷. 微信小程序开发[J]. 信息技术与信息化, 2018, 225(12):67-68.

[15] Netze H . Nutze nur noch MobaXTerm | SSH-Client Putty: Fast vergessene Sicher. | heise Netze-Newsforen[J]. Heise Zeitschriften Verlag..

致谢

大学的时间一点一滴的流逝，即将迎来了人生中最后一次的毕业季。在这个美好的校园时光里，我们的毕业设计也进入了尾声。

在参加毕业设计的整个过程中，非常感谢帮助我的老师，同学还有关心我的朋友们，是他们在我遇到问题的时候给予我最大的帮助，让我克服困难。在我的整个毕业设计的过程中，给予我最大帮助的是我的授业导师张老师，从毕业设计的选题开始，任务书，开题报告，到中期检查以及答辩PPT的制作，每一步都认认真真的给我们作指导，提建议，张老师兢兢业业的工作态度，认真负责的工作作风时时刻刻熏陶着我，影响着我，同时也培养了我的动手能力和实践经验，使我的专业技术和理论知识都得到了进一步的提高。

在完成毕业设计的过程中，也要感谢母校给我们提供了宝贵的学习与实践的机会。最后，再次由衷的感谢我的导师张老师对我的悉心教导。

1. [↑](#endnote-ref-1)
2. [↑](#endnote-ref-2)