

北京邮电大学



《学生游学系统》项目开发文档

——总体设计方案报告

学院：计算机学院（国家示范性软件学院）

专业：计算机科学与技术

班级：2022211305

小组：第 09 小组

成员：张晨阳 2022211683

廖轩毅 2022211637

徐路 2022211644

2024 年 6 月 12 号

目录

- 1. 技术基础与开发环境 1
 - 1.1. 编程语言选择..... 1
 - 1.2. 开发环境与工具..... 2
 - 1.3. 数据库管理..... 2
- 2. 系统架构与设计 3
 - 2.1. 总系统架构..... 3
- 3. 功能模块详细设计 4
 - 3.1. 登录模块..... 4
 - 3.2. 游学推荐模块..... 4
 - 3.3. 路线规划模块..... 5
 - 3.4. 场所查询模块..... 6
 - 3.5. 游学日记模块..... 7

1.技术基础与开发环境

1.1. 编程语言选择

1.1.1. C++17

C++ 作为本项目后端开发的主要语言，用于实现各模块的主要功能、算法、数据结构，以及与数据库的连接。

本项目使用了 C++17 的 `template`、`<auto>`、模板参数等新特性。

1.1.2. JavaScript

JavaScript 在本项目中扮演了服务器端编程语言的角色，通过 Node.js 平台实现。

在本项目中，负责为前端应用程序提供数据接口和逻辑处理能力；与后端程序交互，调用编译好的 C++ 程序，从而实现前后端的解耦和功能的专业分工。

1.1.3. Vue 3.0

Vue 3.0 在本项目中扮演的角色是构建用户界面（UI）的前端 JavaScript 框架。

在本项目中，Vue 3.0 提供了一个强大、灵活且高效的前端开发解决方案，用于构建交互性强、用户友好的 Web 应用。

1.2. 开发环境与工具

1.2.1. 操作系统

Windows 11.

1.2.2. 集成开发工具

Visual Studio Code 1.90

在前后端调试过程中，前后端的服务需要分开运行，而 VSCode 作为一个文本编辑器，可以使多种不同文件出现在一起管理，且可以多开终端进入不同文件夹启动服务，作为本项目的开发工具十分合适。

1.2.3. Node.js 环境

使用 npm（Node Package Manager）来管理项目依赖和自动化工作流是现代前端和 Node.js 应用开发的标准做法。

1.2.4. Express.js 框架

Express.js 是一个基于 Node.js 平台的极简且灵活的 Web 应用开发框架，它使得 Web 服务器的搭建和 API 路由的设计变得快速而简单。

1.3. 数据库管理

1.3.1. MySQL8.0

本项目选择 MySQL8.0 作为数据库系统，因其安装配置简单，性能优越。

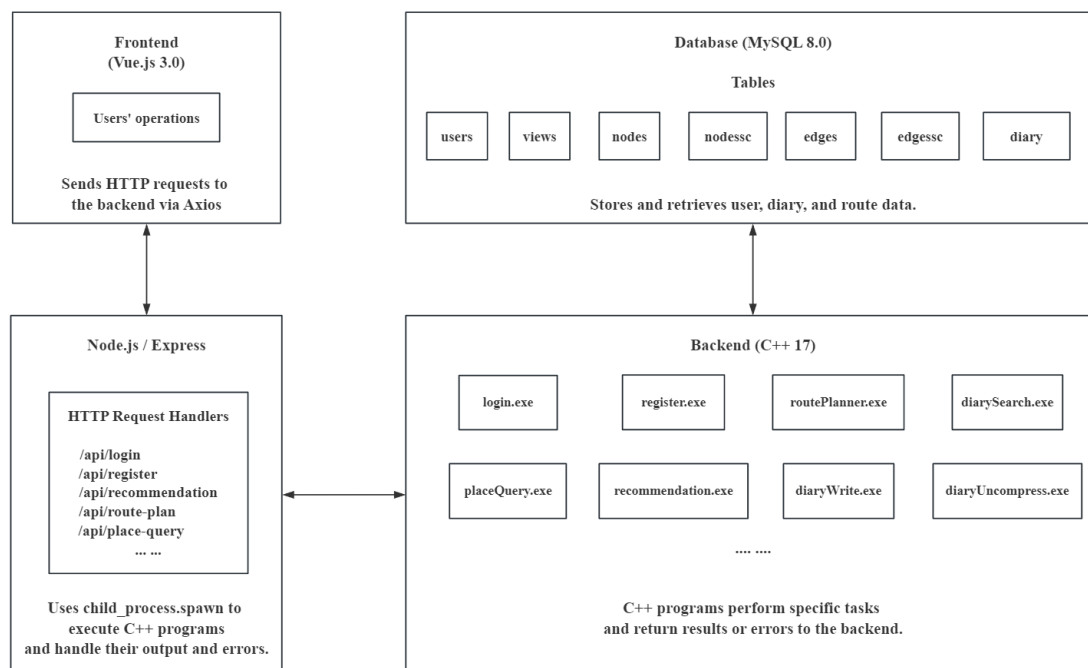
1.3.2. Navicat Premium 16

本项目使用 Navicat Premium 16 作为数据库的图形化工具，帮助处理数据库的各种操作、存储。

2. 系统架构与设计

2.1. 总系统架构

系统的完整架构图如下：



在本系统中，前端采用 Vue.js 3.0 技术开发，为用户提供直观的操作界面。

用户在前端界面上执行操作，例如登录、注册、搜索游学日记等，这些操作会触发前端通过 Axios 库发送 HTTP 请求到后端服务器。

后端服务器基于 Node.js 和 Express.js 构建，专门负责处理这些 HTTP 请求。

后端服务器接收到前端的请求后，会根据请求的类型调用相应的 API 端点，如 `/api/login`、`/api/recommendation`、`/api/register` 等。

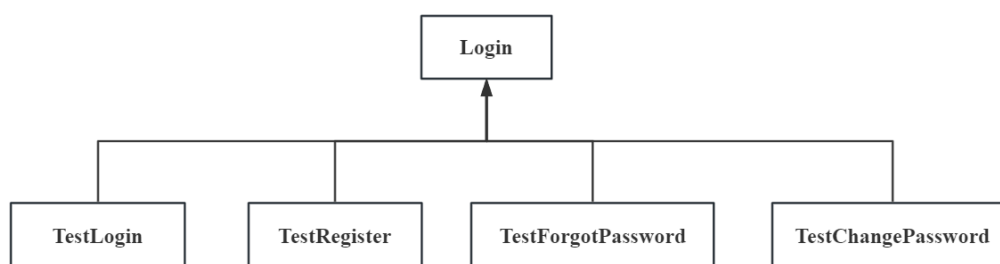
这些 API 端点实际上是 Node.js 使用 `child_process.spawn` 方法调用的 C++ 编写的后端程序。这些程序直接与 MySQL 8.0 数据库交互，存储和检索用户信息、游学日记和路线规划数据。

任务执行完毕后，C++ 程序将结果或错误信息返回给 Node.js 服务器，服务器再将这些信息封装成 HTTP 响应发送回前端。前端接收到响应后，根据内容更新用户界面，完成用户操作的闭环。

3.功能模块详细设计

3.1. 登录模块

登录模块设计如图：



各子模块主要功能如下：

Login： 实现各种登录有关的操作；

TestLogin： 响应前端的登录请求；

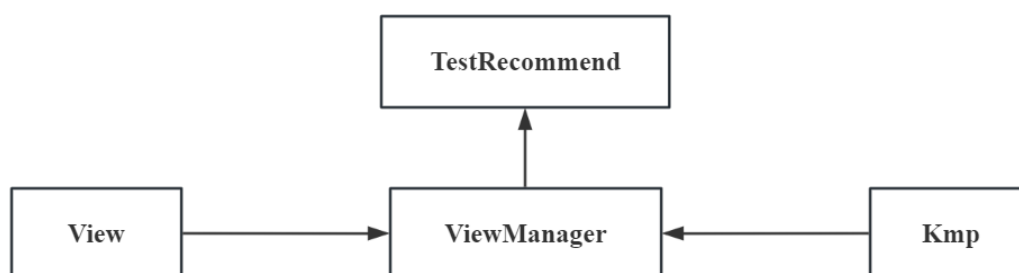
TestRegister： 响应前端的注册请求；

TestForgotPassword： 响应前端的找回密码请求；

TestChangePassword： 响应前端的修改密码请求。

3.2. 游学推荐模块

游学推荐模块设计如图：



各子模块主要功能如下：

TestRecommend： 响应前端的各种游学推荐请求；

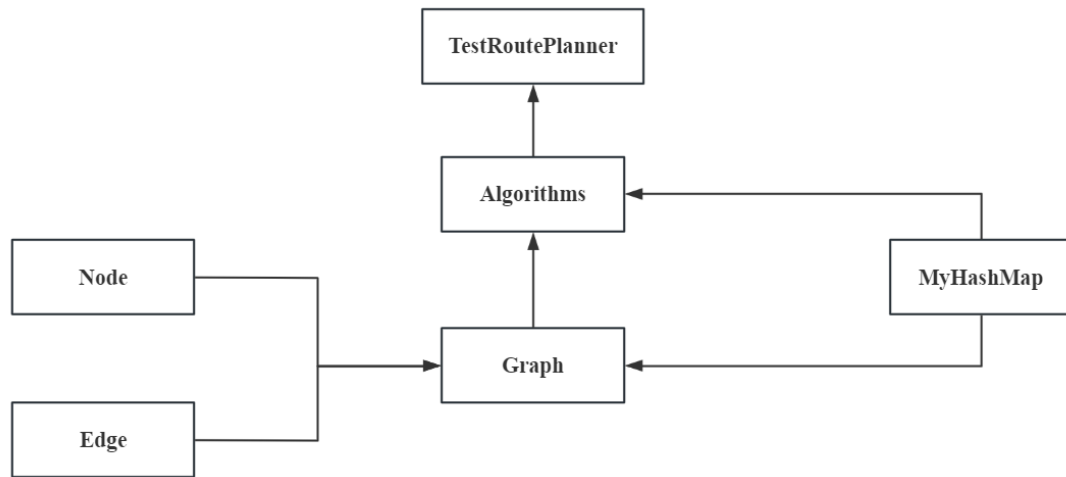
ViewManager： 实现游学推荐的主要功能：排序、搜索等；

View： 存储各景区、校园的数据结构；

Kmp： 实现 Kmp 字符串匹配算法；

3.3. 路线规划模块

路线规划模块设计如图：



各子模块主要功能如下：

TestRoutePlanner： 响应前端的各种路线规划请求；

Algorithms： 实现多种路线规划的核心算法；

Graph： 存储当前地图的数据结构；

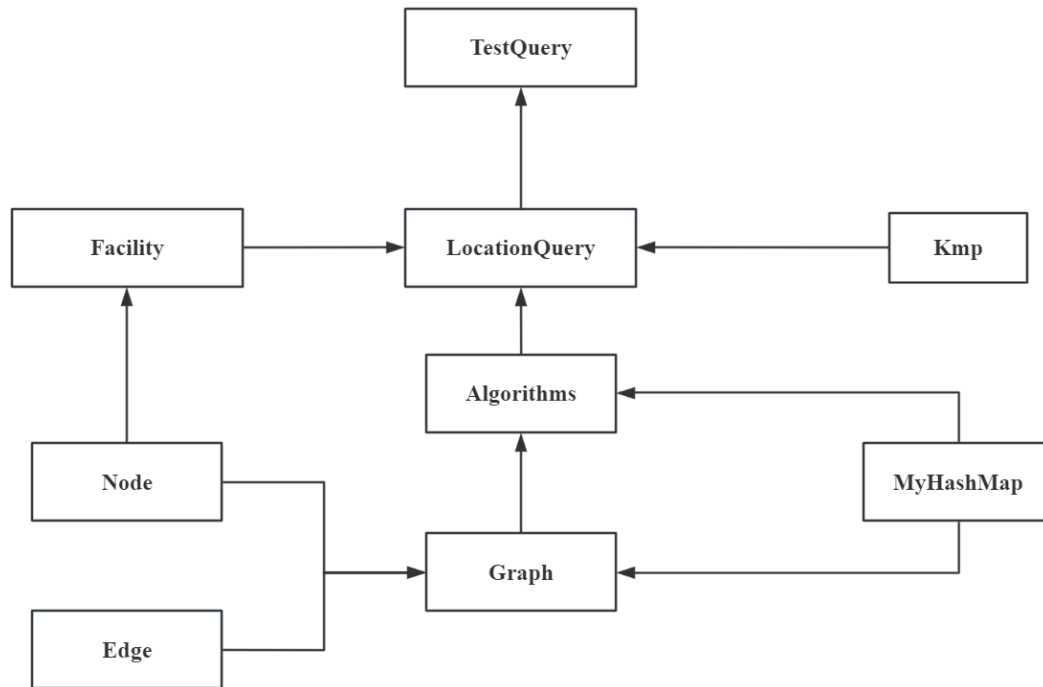
MyHashMap： 自己实现的哈希表数据结构；

Node： 用于存储不同建筑、设施的数据结构；

Edge： 用于存储不同道路的数据结构。

3.4. 场所查询模块

场所查询模块设计如图：



各子模块主要功能如下：

TestQuery： 响应前端的各种场所查询请求；

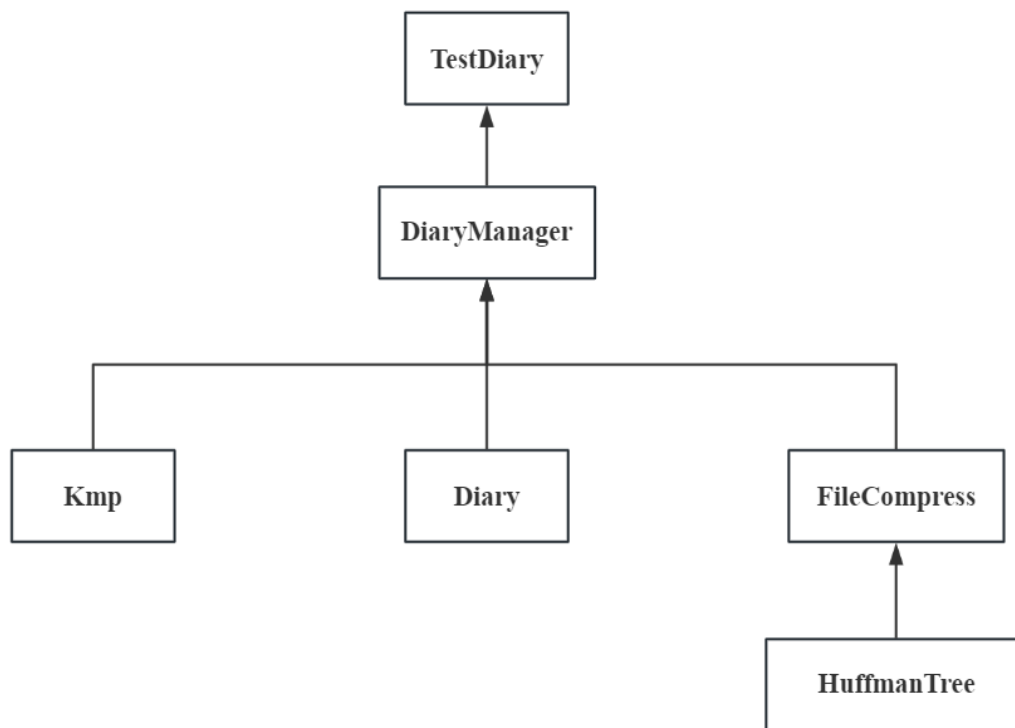
Facility： 用于存储设施的数据结构；

LocationQuery： 实现查询设施、计算距离并排序等核心功能；

剩余子模块与前文相同。

3.5. 游学日记模块

游学日记模块设计如图：



各子模块主要功能如下：

TestDiary： 响应前端的各种游学日记请求；

DiaryManager： 实现游学日记模块的主要功能、算法；

Diary： 用于存储日记的数据结构；

FileCompress： 实现日记压缩、解压的核心算法；

HuffmanTree： 实现压缩功能所需要的哈夫曼树；

剩余子模块与前文相同。