



**《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](#_Toc169106648)

[1.1. 编程语言选择 1](#_Toc169106649)

[1.1.1. C++17 1](#_Toc169106650)

[1.1.2. JavaScript 1](#_Toc169106651)

[1.1.3. Vue 3.0 1](#_Toc169106652)

[1.2. 开发环境与工具 1](#_Toc169106653)

[1.2.1. 操作系统 1](#_Toc169106654)

[1.2.2. 集成开发工具 2](#_Toc169106655)

[1.2.3. Node.js环境 2](#_Toc169106656)

[1.2.4. Express.js框架 2](#_Toc169106657)

[1.3. 数据库管理 2](#_Toc169106658)

[1.3.1. MySQL8.0 2](#_Toc169106659)

[1.3.2. Navicat Premium 16 2](#_Toc169106660)

[2. 系统架构与设计 3](#_Toc169106661)

[2.1. 总系统架构 3](#_Toc169106662)

[2.2. 后端各模块结构设计 3](#_Toc169106663)

[3. 功能模块详细设计 3](#_Toc169106664)

# 技术基础与开发环境

## 编程语言选择

### 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。

### 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。

### 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。

## 开发环境与工具

### 操作系统

Windows 11.

### 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。

### 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。

### 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。

## 数据库管理

### 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。

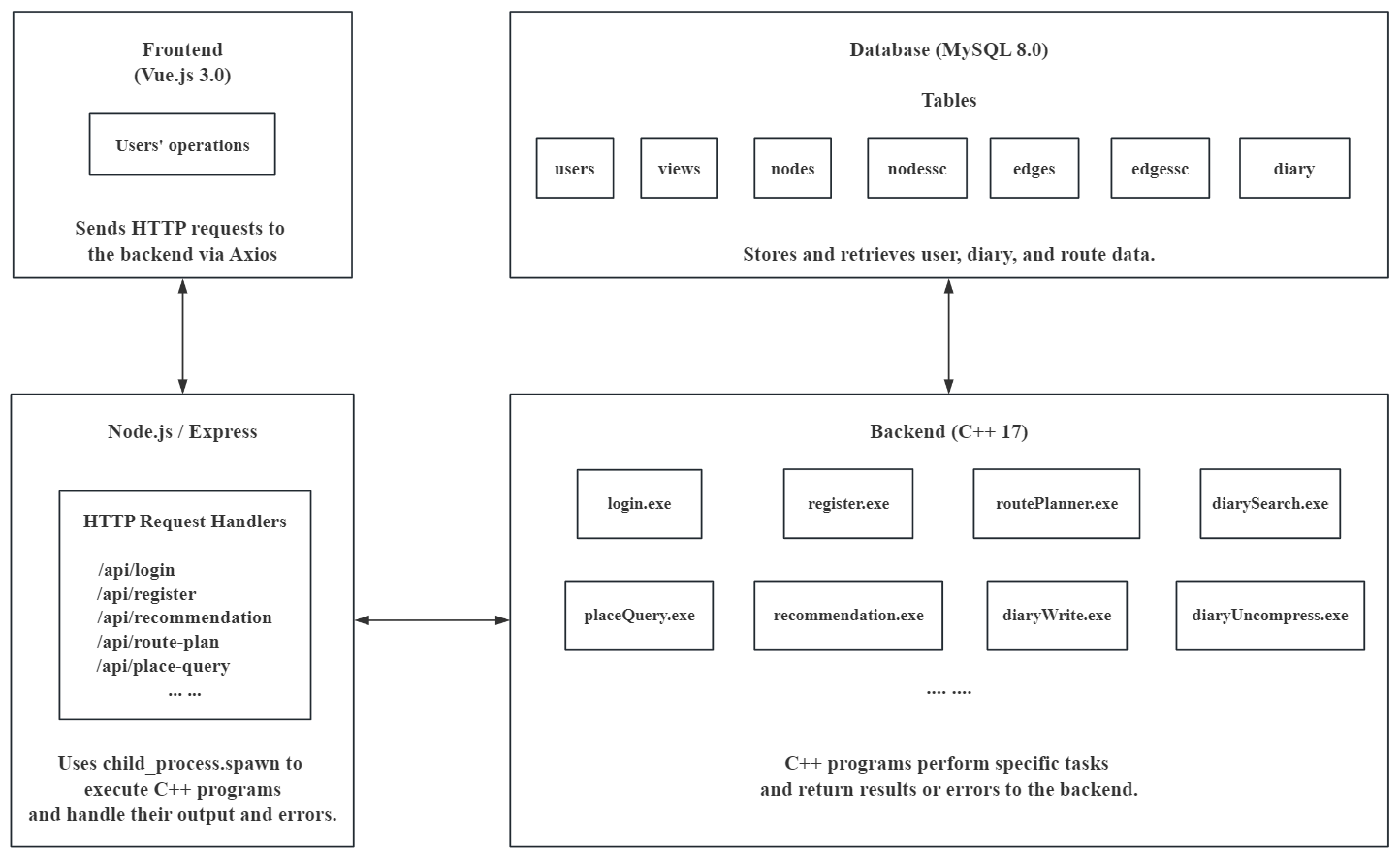
### Navicat Premium 16

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。

# 系统架构与设计

## 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。

## 后端各模块结构设计

各功能模块的结构如下：

# 功能模块详细设计