实 验 报 告

实验人：梁允楷 学号：18342055 日期：2019、5、4

李赞辉 18342053

院（系）：数据科学与计算机学院

专业（班级）：软件工程教务一班

实验题目： 校园电动巴士项目

1. **实验目的**

本实验面向C++语言的初学者。

主要让实验者熟悉面向对象的编程思想以及类的使用。

**二. 实验环境**

本实验基于Atom和Code block开发，利用GitHub进行合作，参考主流的编码规范，如Google C++Style Guide（中文版）

2.1 编程语言和开发工具

编程语言： C++

开发工具： Atom和Code block

2.2 编码规范

要求遵循良好的程序设计风格来设计和编写程序。基本编码规范：

1. 标识符的命名要到达顾名思义的程度；

2. 关键代码提供清晰、准确的注释；

3. 程序版面要求：

a) 不同功能块用空行分隔；

b) 一般一个语句一行；

c) 语句缩进整齐、层次分明。

**三．实验内容**

0. 为了提高编程效率，实现更好的团队协作，我们采用Github平台进行代码同步。

1. 先构建SchoolBus、BusStation两个类，BusStation使用静态链表存储，SchoolBus用数组存储，完善类中的相关的数据成员与接口

2. 从main函数开始，从使用者角度思考程序的编写，写开始界面、初始化等，将用户引向乘客模式或调度员模式，乘客模式拥有查询功能，调度员拥有查询功能和设置功能

3. 进行分层与封装，将数据初始化、合法性输出检测、登录功能等的实现封装进init文件中，乘客与调度员模式封装进passenger和controller文件中

4. 完善用户界面，逐步实现查询与设置功能，开头简单介绍校园巴士的特点便利、实用，给用户良好的使用交互。

5. debug，完善返回键功能和用户界面

注：最开始我们尝试用链表存储BusStation，但注意到多文件调用时方便性，且有感于当时vmatrix上的一道有关静态链表的题目，故选择了静态链表，既能满足我们存储线路的需求，也便于查询功能的设计

**四.分析与设计**

简要描述程序设计的过程，包括设计思路，设计要点及特色；程序的不足与改进等。要求画出程序的简单流程图。

4.1、需求分析：

分为乘客与调度员两个部分，含有明了的提示和引导界面，输入的合法性检测。

乘客可以通过车站名字了解车站停靠的小白、正驶向该车站的小白和候车人数，为方便顾客，顾客可从程序提供的线路中查询车站，实现车站选择界面的条理化、可视化

乘客可以通过小白编号查询小白状态，包括是否在停靠、空位数量、停靠或正开往的车站

调度员设置小白的状态等相关信息、设置车站的相关信息，同时，还应能判断调度员的设置是否合理，并在不合理时作出提醒

调度员也具备乘客所具备的查询功能

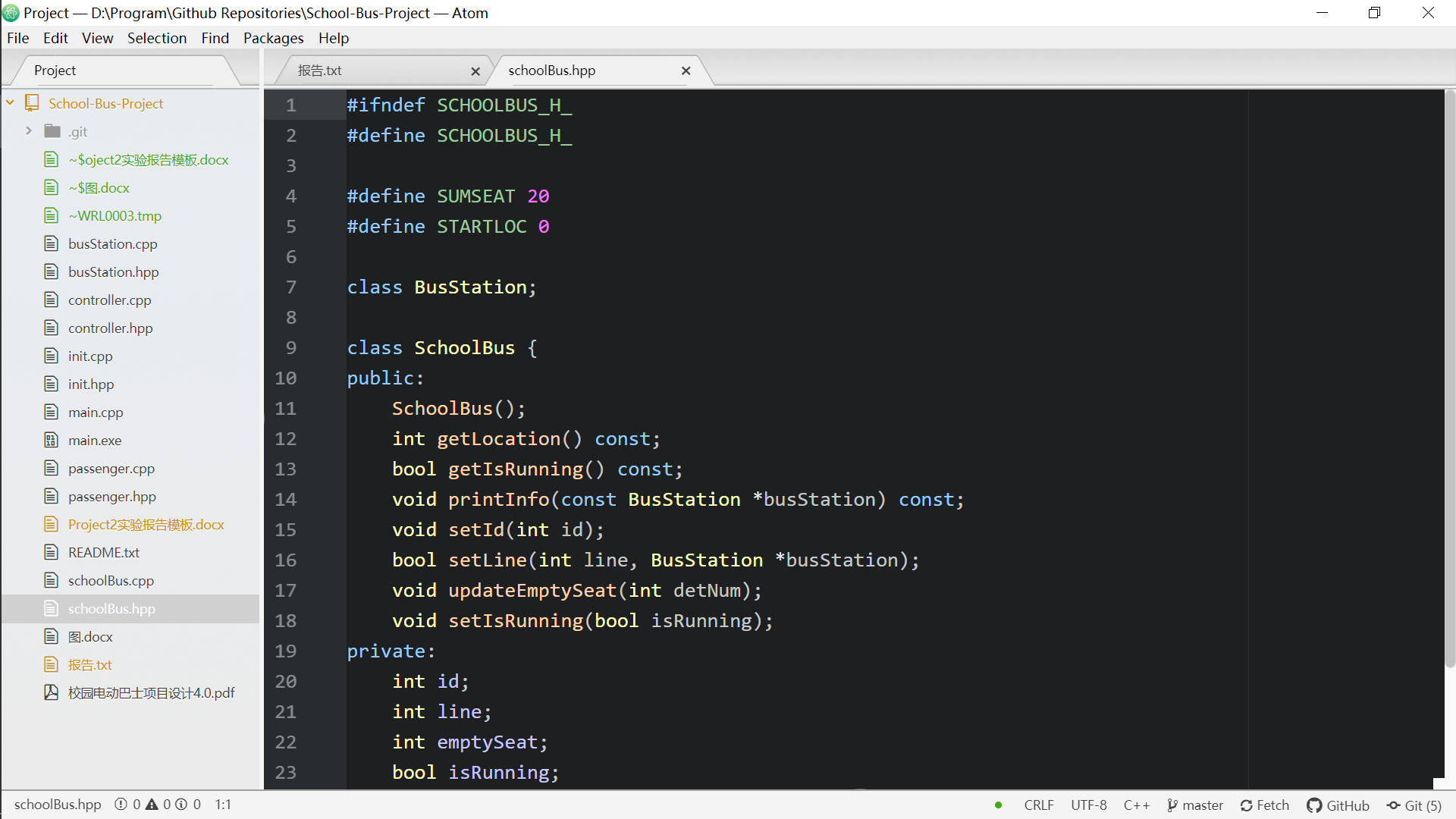
值得注意的是，我们在车站基础上添加了线路，因为实际需求中，小白是有固定的路线的。而且，校园中可能存在多条路线，每条路线的使用的高峰期可能不一样，所以我们设计小白能存储它当前走的线路，并能被调度员调到其它线路，而乘客查询的信息也会相应作出调整，这样小白的运行就十分灵活了。

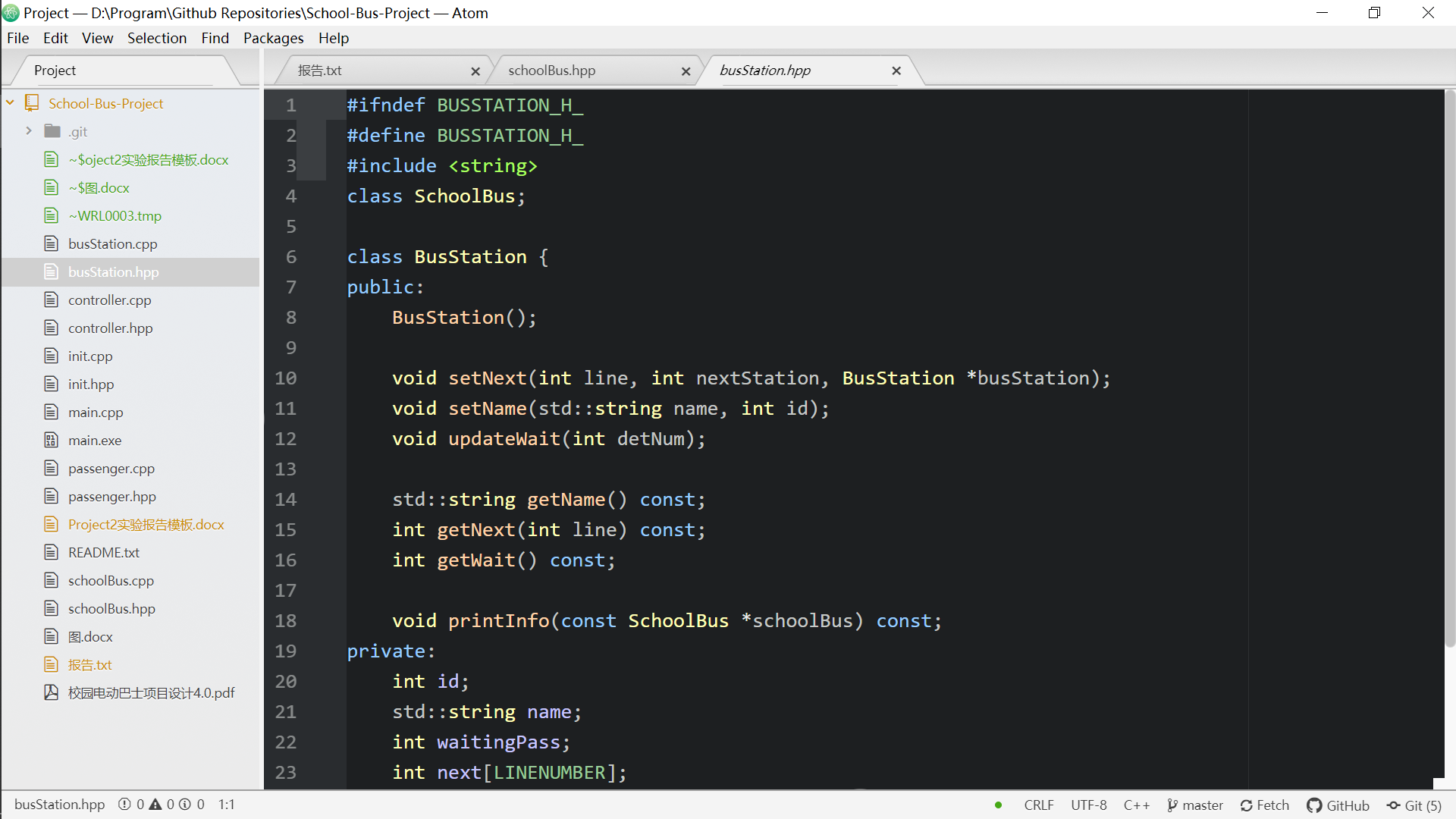
系统功能图

4.2、类结构设计

包含：SchoolBus, BusStation

在设计程序的过程中，我们发现两个类之间**互相调用**，很容易会出现重复调用的问题。后来我们上网查询，可以在头文件中先声明另一个类，然后通过指针进行访问，就可以实现两个类之间的相互调用。





类关系图：

4.3、细节设计

接口设计：

登录调度员账号和进入调度员模式采用两个完全不同的接口，使用设置许可权的方式，避免简单粗暴地将进入调度员模式直接嵌套进登录的接口。这样的实现，登录的接口无需传入schoolBus和busStation的参数，使得登录函数仅仅负责登录一项功能，使用松耦合的方式构建程序，增加了程序的可维护性。

每个接口实现的逻辑过程都比较短，一个接口的代码长度都不超过20行。

数据成员设计：

按照题目要求，实现了最基本的数据成员设计。

其中，比较特别的地方就是设计车站的时候，使用了链表用于连接同一条线路的相邻站点。

成员函数：

除了实现基本的get, set函数，我们加入了一个输出整合完成的数据的函数，方便各个接口之间的调用。

**五、实验结果**

设计有代表性的输入数据，并分析测试结果以及针对测试中发现的问题所进行的修正与改进。

一个好的程序离不开不断的调试，最能减少bug出现的调试方式就是写完一层调试一层 ，所谓测试驱动开发，尽可能做到完成一个接口就能确保接口功能的正确性。

说明：由于采取测试驱动开发的方法，实验过程中遇到的bug比较少，目前发现只有一个bug：用户交互中访问路线会出现访问无效内存的问题，这是由于一开始设计类的时候，内存分配是从0开始编号，而交互界面是从1开始编号；这是设计类的时候，没有想清楚而遗留下来的问题。解决的方案自然是用户每次输入路线以后都进行一次减1的操作。

**六、设计心得**

一个好的程序离不开好的初始结构，通过一个良好的架构，实现层层分层，能有效减少一个函数的行数，即使函数功能专一化，提高代码可读性和可重复利用性，如我们类的划分，对一些功能如合法性输入的封装等，都贯彻了这一理念，使我们单一函数的代码普遍低于20行，便于我们测试、更改。

设计过程也激发我们对问题的创新性思考，不似过去做编程题，围绕算法或知识点进行思考，实现项目的原理不复杂，但如何实现的更好是个难题，无论是架构、数据成员设计还是输入的提示信息，无不是我们过去未思考过的，而这些，都需要我们的创新性思考。

**七、项目详情可参见：**

**https://github.com/warpmatrix/School-Bus-Project**