**北京邮电大学课程设计报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **课 程 设**  **计 名 称** | **数 据 结 构** | | **学 院** | **计算机学院** | **指导教师** | **杨 震** |
| **班 级** | **班内序号** | **学 号** | | **学生姓名** | **分 工** | |
| **2020211306** | **17** | **2020211437** | | **陈禹璋** | **前端设计、用户交互** | |
| **2020211306** | **27** | **2020211447** | | **吴建军** | **文件操作、地图数据采集** | |
| **2020211306** | **28** | **2020211448** | | **吴镇均** | **后端算法、数据结构设计** | |
| **课**  **程**  **设**  **计**  **内**  **容** | 大学中每位同学每学期会有多门不同的课程，课程分布在不同的教学楼甚至不同的校区；每门课程都会有一些课程资料、作业、考试和课程群等内容；此外每位同学在课外还会有一些个人或者集体的活动安排。  线下课程辅助系统可以帮助学生管理自己的课程和课外活动，具备课程导航功能、课程信息管理和查询功能，以及课外信息管理和查询功能等。每天晚上系统会提醒学生第二天上的课，每门课需要交的作业和需要带的资料，以及考试的信息；快要上课时系统根据该课程的上课地点设计一条最佳线路并输出；学生可以通过系统管理每门课的学习资料、作业和考试信息；在课外，学生可以管理自己的个人活动和集体活动信息，可以进行活动时间的冲突检测和闹钟提醒。 | | | | | |
| **学 生**  **课程设计**  **报 告**  （附 页） | 见附页 | | | | | |
| **课**  **程**  **设**  **计**  **成**  **绩**  **评**  **定** | 遵照实践教学大纲并根据以下四方面综合评定成绩：  1、课程设计目的任务明确，选题符合教学要求，份量及难易程度  2、团队分工是否恰当与合理  3、综合运用所学知识，提高分析问题、解决问题及实践动手能力的效果  4、是否认真、独立完成属于自己的课程设计内容，课程设计报告是否思路清晰、文字通顺、书写规范  **评语**:  **成绩:**  指导教师签名：  2022年 月 日 | | | | | |

注：评语要体现每个学生的工作情况，可以加页。



**课 程 报 告**

|  |  |
| --- | --- |
| 课程名称 | 数据结构 |
| 课程题目 | 线下课程辅助系统的设计与实现 |
| 小组成员 | 陈禹璋 2020211437 |
|  | 吴建军 2020211447 |
|  | 吴镇均 2020211448 |
| 班级 | 2020211306 |
| 指导教师 | 杨 震 |

二〇二二年六月

**目录**

[1 概览 1](#_Toc10071)

[1.1 任务描述 1](#_Toc32187)

[1.2 开发环境 1](#_Toc3609)

[1.3 项目创新点 1](#_Toc32403)

[1.4 配置文件说明 2](#_Toc29654)

[1.5 后端数据结构说明 4](#_Toc32162)

[2 功能需求 8](#_Toc10864)

[2.1 课程信息管理和查询 8](#_Toc28836)

[2.2 课外信息管理和查询 8](#_Toc30368)

[2.3 课程导航 8](#_Toc24465)

[2.4 模拟系统时间 9](#_Toc26115)

[2.5 建立日志文件 9](#_Toc14607)

[2.6 选做及拓展 9](#_Toc20020)

[3 模块介绍 11](#_Toc24469)

[前端 11](#_Toc1527)

[后端 11](#_Toc23667)

[3.2 前端 11](#_Toc12423)

[3.2.1 数据保存功能 11](#_Toc20578)

[3.2.2 计时与倍速功能 11](#_Toc29823)

[3.2.3 管理员功能 12](#_Toc13639)

[3.2.4 导航功能 12](#_Toc771)

[3.2.5 课程信息功能 13](#_Toc25416)

[3.2.6 活动信息功能 14](#_Toc14593)

[3.2.7 闹钟功能 14](#_Toc7022)

[3.2.8 搜索功能 14](#_Toc13263)

[3.3 后端 15](#_Toc11362)

[3.3.1 文件管理功能 15](#_Toc15220)

[3.3.2 导航功能 16](#_Toc10763)

[3.3.3 日志功能 16](#_Toc14417)

[3.3.4 搜索功能 17](#_Toc14097)

[3.3.5 排序功能 17](#_Toc22285)

[3.3.6 配置文件读取 18](#_Toc6038)

[3.3.7 HTTP服务 18](#_Toc21745)

[4 用户指南 19](#_Toc16040)

[4.1 安装依赖 19](#_Toc26)

[4.2 编译项目 19](#_Toc21552)

[4.3 使用说明 19](#_Toc4640)

[4.3.1 运行系统 19](#_Toc24425)

[4.3.2 导航规划 21](#_Toc302)

[4.3.3 课程表 22](#_Toc25139)

[4.3.4 所有课程 23](#_Toc665)

[4.3.5 课程信息 24](#_Toc21425)

[4.3.6 今日课程 27](#_Toc4990)

[4.3.7 我的活动 27](#_Toc7438)

[4.3.8 设置闹钟 28](#_Toc16103)

[4.3.9 今日活动 30](#_Toc4463)

[4.3.10 时间控制 30](#_Toc22155)

[4.3.11 管理员模式 30](#_Toc25168)

[5 测试评估 32](#_Toc14888)

[5.1 范例执行结果及说明 32](#_Toc23514)

[测试样例一 32](#_Toc27393)

[测试样例二 32](#_Toc20738)

[5.2 评价和改进 33](#_Toc21520)

[6 实验总结 35](#_Toc7524)

[6.1 实现的难点、遇到的问题和解决方案 35](#_Toc13535)

[6.2 成员心得 36](#_Toc4922)

[7 致谢 37](#_Toc19505)

[8 参考资料 38](#_Toc7657)

**摘要**

大学中每位同学每学期会有多门不同的课程，课程分布在不同的教学楼甚至不同的校区；每门课程都会有一些课程资料、作业、考试和课程群等内容；此外每位同学在课外还会有一些个人或者集体的活动安排。

线下课程辅助系统可以帮助学生管理自己的课程和课外活动，具备课程导航功能、课程信息管理和查询功能，以及课外信息管理和查询功能等。每天晚上系统会提醒学生第二天上的课，每门课需要交的作业和需要带的资料，以及考试的信息；快要上课时系统根据该课程的上课地点设计一条最佳线路并输出；学生可以通过系统管理每门课的学习资料、作业和考试信息；在课外，学生可以管理自己的个人活动和集体活动信息，可以进行活动时间的冲突检测和闹钟提醒。

*关键词：路径规划，课程管理，文件操作，算法优化，冲突检测，路径导航*

**Abstract**

Each student in the university will have several different courses each semester, and the courses are distributed in different buildings or even different campuses; each course will have some course materials, assignments, exams and course groups; in addition, each student will have some individual or group activities scheduled outside of class.

The offline course assistance system can help students manage their courses and extracurricular activities, with course navigation, course information management and query functions, as well as extracurricular information management and query functions. Every night the system reminds students of the next day's classes, the assignments and materials they need to turn in for each class, and information about exams; when classes are about to start the system designs an optimal route according to the class location of the course and outputs it; students can manage study materials, assignments and exam information for each class through the system; outside of class, students can manage their individual activities and group activity information, and can perform activities time conflict detection and alarm clock reminders.

*Keywords: path planning, course management, file operations, algorithm optimization, conflict detection*

**1** 概览

**1.1** 任务描述

不限编程语言，进行线下课程辅助系统的设计与实现。完成所有基本功能，并完成所有拓展功能，在能力范围内进行了自带算法优化。

**1.2** 开发环境

后端

• Go 1.18

• Windows 10 21H1

• Visual Studio Code 1.66.2

前端

• Microsoft Edge

• HTML5 + CSS3 + JavaScript(ES5)

• 第三方库引用情况及用途说明：

• jQuery v3.5.1（操作 DOM 树）

• 百度地图API（绘制路线，标示建筑）

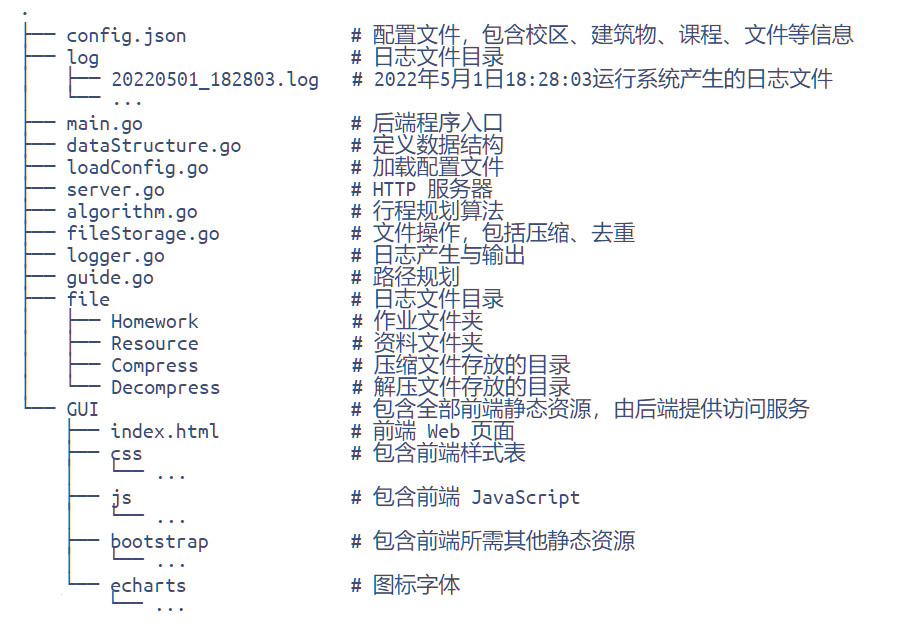
**1.3** 项目创新点

首次大规模使用Go语言[[[1]](#endnote-0)]，采用BS架构构建前后端服务结构，具备对规划和模拟的进程进行展示与记录，即图形化与日志的功能；还调用了百度地图API进行路线绘制，采样上百个点位和路线，更接近实际生活，亦可考虑交付实用。此外，在文件处理方面具备强悍的压缩和MD5去重功能，并能够对整个目录进行压缩管理，在导航算法部分采用了堆优化的迪杰斯特拉算法与全排列相结合的模式，在搜索部分采用了优化后的RK算法进行模糊匹配，以及Trie前缀树的高效查找前缀匹配的算法。

本系统采用前后端分离的设计模式、浏览器/服务器（B/S）的架构模式。后端（服务端）负责全部计算任务；前端（客户端）仅承担信息展示与交互的任务，不参与任何计算。

服务端程序启动后，完成初始化并建立 HTTP 服务，监听前端的请求。用户访问前端页面后，以 AJAX 轮询的方式与系统后端进行内容同步，当用户需要进行规划、查询、模拟等操作时，向后端发起 GET 请求，后端计算完成后将结果返回。

工程包含配置文件、日志文件、后端程序、前端资源四大部分，文件结构及简要说明如下：



**1.4** 配置文件说明

配置文件中保存的是需要在服务器开启时进行初始化的、数据量较大的数据集。

总体结构

{

    "Port": 9090,   //服务器端口

    "schedule": [{…},{…},...],   //课程表中的所有课程

    "scheduleTimeSlot": [{…},{…},...],   //课程表中的课程时间信息

    "curriculums": [{…},{…},...],   //所有课程信息

    "Location": [{…},{…},...],   //所有点的信息，用于导航

    "Road": [{…},{…},...],   //所有边的信息，用于导航

    "ActivityType": [{…},{…},...],   //活动类型以及活动类型的具体内容

    "Activity": [{…},{…},...],   //初始化的活动，即一开始就有的活动

    "BusSchedule": [{…},{…},...],   //巴士发车时间以及途径时间

    "District": [{…},{…},...]   //所有标志地点的信息，

// 即前端可以选中使用的地点

}

课程表的课程信息

"schedule": [

        {

            "name": "计算机网络",   //课程名称

            "info": [   //课程信息

                {

                    "week": 0,   //这节课在星期几

                    "order": 0,   //这节课在当天第几节

                    "isExam": **false**,   //这节课是不是考试

                    "district": "沙河校区",   //这节课所在校区

                    "spot": "沙河综合教学楼",  //这节课所在地点

                    "classroom": "N216"   //这节课所在教室

                },

                ...

            ]

        },

        ...

]

课程表的课程时间信息

"scheduleTimeSlot": [

        {

            "startHour": 8,   //开始的时间-小时

            "startMinute": 0,   //开始的时间-分钟

            "endHour": 8,   //结束的时间-小时

            "endMinute": 45   //结束的时间-分钟

        },

        ...

]

所有课程信息

"curriculums": [

        {

            "name": "计算机网络",   //课程的名称

            "instructor": "高小明",   //课程的教师

            "group": "12121212",   //课程的QQ群

            "homework": [],   //课程的已布置作业

            "resource": []   //课程的已上传资料

        },

        ...

]

所有点的信息

"Location": [

        {

            "Lname": "沙河大门",   //地点的名称

            "LID": 1,   //地点的ID

            "x": 116.29036,   //地点的经度

            "y": 40.163069   //地点的纬度

        },

        ...

]

所有边的信息

"Road": [

        {

            "RID": 1,   //边的ID

            "FromL": 1,   //边的起点的ID

            "ToL": 2,   //边的终点的ID

            "IsBycle": **true**,  //这条边是否是自行车道

            "Crowd": 0.3,   //这条边的拥挤度

            "Length": 300   //这条边的长度

        },

        ...

]

活动类型信息

"ActivityType": [

        {

            "Type": "个人活动",   //活动类型的名称

            "Content":["自习","锻炼","外出","游戏","其他"]  //活动类型的具体内容

        },

        ...

]

初始活动信息

"Activity": [

        {

            "Week": 5,   //活动的星期

            "Hour": 9,   //活动的开始时间-小时

            "Minute": 0,   //活动的开始时间-分钟

            "District": "沙河校区",   //活动所在校区

            "Spot": "沙河操场",   //活动所在地点

            "Classroom": "操场",   //活动所在教室

            "Activity": "晨跑",   //活动内容

            "ActivityType": "个人活动",   //活动类型

            "ActivityContent": "锻炼"   //活动类型的具体内容

        },

        ...

]

巴士时间信息

"BusSchedule": [

        {

            "StartHour": 9,    //巴士发车时间-小时

            "StartMinute": 50,    //巴士发车时间-分钟

            "Duration": 60   //坐巴士所需时间

        },

        ...

]

所有标志地点信息

"District": [

        {

            "name": "沙河校区",   //该校区的名称

            "spot": [   //该校区的所有地点的信息

                {

                    "id": 1,   //该地点的ID

                    "name": "沙河大门",   //该地点的名称

                    "classroom": […], //该地点的所有教室

                    "x": 116.29036,   //该地点的经度

                    "y": 40.163069   //该地点的纬度

                },

                ...

            ]

        },

        ...

]

**1.5** 后端数据结构说明

课程表信息

type Schedule\_atom struct {

    Name string   //课程名称

    Info []Schedule\_Info   //课程信息

}

type Schedule\_Info struct {

    Week      **int**   //时间-周

    Order     **int**   //当天第几节课

    IsExam    bool   //是否是考试

    District  string   //校区

    Spot      string   //地点

    Classroom string   //教室

}

课程时间表

type ScheduleTimeSlot\_atom struct {

    StartHour   **int**  //这节课开始时间-小时

    StartMinute **int**  //这节课开始时间-分钟

    EndHour     **int**  //这节课结束时间-小时

    EndMinute   **int**  //这节课结束时间-分钟

}

已布置作业信息

type Homework struct {

    Title       string   //已布置作业标题

    Description string   //已布置作业描述

    Year        **int**   //已布置作业时间-年

    Month       **int**   //已布置作业时间-月

    Hour        **int**   //已布置作业时间-时

    Minute      **int**   //已布置作业时间-分

    Day         **int**   //已布置作业时间-日

    HasFinished bool   //已布置作业是否已完成

    Uploaded    []Uploaded   //已布置作业的已上传作业

}

已上传作业信息

type Uploaded struct {

    HomeworkName string   //已上传作业名称

    Remark       string   //已上传作业备注

    Year         **int**   //已上传作业时间-年

    Month        **int**   //已上传作业时间-月

    Day          **int**   //已上传作业时间-日

    Hour         **int**   //已上传作业时间-时

    Minute       **int**   //已上传作业时间-分

    Version      **int**   //已上传作业版本

}

已上传文件信息

type Resource struct {

    ResourceName string   //资料名称

    Remark       string   //资料备注

    Year         **int**   //资料时间-年

    Month        **int**   //资料时间-月

    Day          **int**   //资料时间-日

    Hour         **int**   //资料时间-时

    Minute       **int**   //资料时间-分

}

课程信息

type Curriculums\_atom struct {

    Name       string   //课程名称

    Instructor string   //课程教师

    Group      string   //课程QQ群

    Homework   []Homework   //课程的已布置作业

    Resource   []Resource   //课程的资料

}

闹钟信息

type ClockList []Clock

type Clock struct {

    Week       **int**   //闹钟时间-星期

    Hour       **int**   //闹钟时间-时

    Minute     **int**   //闹钟时间-分

    Remark     string   //闹钟备注

    ClockCycle string   //闹钟类型：仅响一次/每天一次/每周一次

}

活动信息

type ActivityList []Activity

type Activity struct {

    Week            **int**   //活动时间-星期

    Hour            **int**   //活动时间-时

    Minute          **int**   //活动时间-分

    District        string   //活动地点-校区

    Spot            string   //活动地点-地点

    Classroom       string   //活动地点-教室

    Activity        string   //活动内容

    ActivityType    string   //活动类型

    ActivityContent string   //活动类型的具体内容

}

时间信息

type Time struct {

    Year   **int**  //年

    Month  **int**  //月

    Day    **int**  //日

    Week   **int**  //星期

    Hour   **int**  //时

    Minute **int**  //分

}

控制元素信息

type Controller struct {

    IsAdmin     bool   //是否是管理员

    IsPausing   bool   //是否暂停

    Multi\_speed **int**   //倍速

}

校区信息

type Coordinate struct {

    Name string   //校区名称

    Spot []struct {   //校区的地点的集合

        Name      string   //该地点的名称

        Classroom []string   //该地点的教室的集合

        X         float64   //该地点的经度

        Y         float64   //该地点的纬度

        ID        **int**   //该地点的ID

    }

}

活动类型信息

type ActivityType\_atom struct {

    Type    string   //活动类型

    Content []string   //活动类型的具体内容的集合

}

路径阶段信息

type Section struct {

    FromID        **int**

    ToID          **int**

    FromX         float64

    FromY         float64

    ToX           float64

    ToY           float64

    IsBycle       bool

    FootDuration  float32 //用时

    BycleDuration float32

    SectionLength **int** //总长度

}

导航信息

type Travel struct {

    ID            string    //旅程编号

    TotalLength   **int**       // 总长度

    TotalDuration float32   // 总时间

    Path          []Section // 旅行路径

}

点信息

type Location struct {

    Lname string

    LID   **int**

    X     float64

    Y     float64

}

边信息

type Road struct {

    RID     **int**

    FromL   **int**

    ToL     **int**

    IsBycle bool   //是否是自行车道

    Crowd   float32   //拥挤度

    Length  **int**   //长度

}

巴士时间信息

type BusSchedule\_atom struct {

    StartHour   **int**  //发车时间-时

    StartMinute **int**  //发车时间-分

    Duration    **int**  //途径时间

}

文件错误信息

type FileError struct {

    IsDuplicated bool  //文件是否已存在

}

导航Log信息

type TravelInfo struct {

    TravelContent string  //通过该string判断该输出哪种Log

}

压缩的哈夫曼树

type TreeNode struct {

    Val   **int**

    Times **int**

    Left  \*TreeNode

    Right \*TreeNode

}

type treeHeap []\*TreeNode

前缀树的节点

type Node **struct** {

    //      若干个子节点: 字符(中文操作)： 节点

    Children map[rune]\*Node

    //      标识节点是否是一个单词最后的一个字符

    isWord **bool**

    //      标识对应的id，即在对应切片里的位置

    id **int**

}

计算海明距离的哈希值结构

type SimHash **struct** {

    IntSimHash int64

    HashBits   **int**

}

**2** 功能需求

校园内建筑物（教学楼、办公楼、宿舍楼）数不少于20个，其它服务设施不少于5种，共20个；

**建立校园内部道路图：**包括各种建筑物、服务设施等信息；不能太简单（边数不能少于200条）；校区个数至少2个；

**课程数目不少于10门：**每门课程包括上课时间、上课地点、课程教师、电子资料、纸质资料、作业信息和考试信息等；

**课外活动不少于20个：**课外活动包括个人活动和集体活动，每个活动包括活动时间、活动地点等信息。

2.1 课程信息管理和查询

学生可以输入课程名称，或者通过课程表查询课程的上课时间、上课地点、课程资料、当前进度、已交作业、待交作业、课程群、考试时间和考试地点等信息。学生可以上传和更新课程资料、上传完成的作业内容。对于课程资料和作业可以进行查询、压缩和去重。系统管理员统一发布考试时间和考试地点，并可以发布和更改课程上课的时间和上课地点。

2.2 课外信息管理和查询

学生可以输入课外活动信息，课外活动包括个人活动和集体活动；个人活动可以包括：自习、锻炼、外出等，集体活动可以包括：班会、小组作业、创新创业、聚餐等。学生可以设定活动闹钟，闹钟可以是一次性的和周期性的，用于活动提醒。系统可以检测个人活动、集体活动和课程的时间冲突，并给出提示。

2.3 课程导航

学生向系统输入课程名称、上课时间或者上课地点，系统自动进行路径规划。

课程名称可以是《数据结构》，系统会自动根据学生的班级信息和最近的上课时间（未开始）查询上课地点；

- 上课时间可以是“周五10点”系统会自动更据学生的班级信息和最近的上课时间（未开始）查询上课地点；

- 上课地点是上课的物理位置，例如 “教三楼111教室”；

- 起点和终点可以在不同校区，需要考虑校区间的交通方式；

- 起点和终点间可以选择最多三个途径点，但此时所有点都必须在同一校区；

- 校区间的交通方式为：定点班车（可以自行规划班次时刻表）和公共汽车（可等间隔发车）。

关于导航策略（前两种策略默认校区内步行；第四种，校区内可选交通工具）

- 最短距离策略：距离最短即可；

- 最短时间策略：假设每条道路拥挤度不一样，在这种情况下时间最短即可；拥挤度为小于等于1的一个正数，真实速度=拥挤度\*理想速度；拥挤度与理想速度自拟；

- 交通工具的最短时间策略：校区内选择自行车时，只能走自行车道路，默认自行车在校区任何地点都有；在考虑不同拥挤度的情况下时间最短；

最后，按照选择的策略输出线路。

2.4 模拟系统时间

系统依据时钟向前推进，时间精度为小时，且以计算机的10秒作为模拟系统的1小时（可以支持快进），人机交互时暂停系统时间推进（例如用户输入信息时），可以通过加入时钟暂停按钮或者命令来实现。

2.5 建立日志文件

记录学生课程和活动状态变化，系统提醒的信息，输出的导航信息，以及学生输入的信息和各种查询操作。

2.6 选做及拓展

选做一：设计各种功能的图形界面；

选做二：途径最短距离策略：途径多个地点的最短距离路径；

选做三：能够使用课表图形界面方式进行课程管理和查询；

选做四：能够对课程作业和资料进行版本管理

对于以上基础功能及拓展要求，小组成员均已按质按量实现。

**3** 模块介绍

前端

index.html: 展视前端页面内容。

GUI/js/main.js: 初始化 index.html 页面的动态内容。

GUI/js/curriculum.js: 包含了课程信息部分逻辑相关的实现。

GUI/js/activityAndClock.js: 包含了课外信息部分逻辑相关的实现。

GUI/js/locationAndNavigation.js: 包含了导航部分逻辑相关的实现。

GUI/js/timeAndController.js: 包含了时间以及部分控制信息，例如倍速、管理员等功能的实现。

后端

main.go: 声明后端所需变量，完成对其他文件中函数的调用。

server.go: 启动服务器，完成对前端信息的接收、处理与发送。

algorithm.go: 保存除文件存储、导航以外的其他部分的实现。例如对闹钟的添加与删除、对活动的添加与删除、搜索等功能的实现。

fileStorage.go: 关于文件存储的实现。

guide.go: 利用了djkstra算法和全排列算法，关于导航的实现。

dataStructure.go: 定义了所有需要用到的数据结构。

loadConfig.go: 实现了对json文件的读和写，实现了数据的持久化。

logger.go: log输出的实现。

3.2 前端

负责GUI的呈现。对于搜索、文件相关、寻路等核心功能，通过将请求数据发送给后端服务器，后端计算后将结果返回给前端，前端获取该数据后经过处理呈现于GUI；其余功能由前端实现的同时将数据发送给后端服务器，由后端保存，从而每次打开前端都能够从后端读取已保存的数据呈现在GUI上。

**3.2.1** 数据保存功能

每次前端页面加载成功时，利用$.getJSON方法从后端读取各类数据；每次前端产生新的数据或需要删除数据时，利用$.get方法将当前时间数据与对应数据传递到后端处理。点击页面下方“保存”按钮可通知后端将当前页面的数据写入Json文件。在页面关闭时向后端传递当前时间数据。

**3.2.2** 计时与倍速功能

当页面加载成功时，对计时按钮绑定startTimer与pauseTimer两个函数。其中startTimer函数内部实例化了一个定时器setInterval，赋值给time。其中setTime函数对右下角的时间按分钟进行递增。

//以下是所要更新的全部数据

    time = setInterval(**function**(){

        //设置右下角的时间

        setTime();

        //设置闹钟相关

        setClock();

        //刷新 今日课程

        refreshTodayClass();

        //刷新 今日活动

        refreshTodayActivities();

        //删除已过期的考试

        deleteExam();

        //当正在导航时，每次调用该setInterval（过了一分钟）则让导航的剩余时间navMinute减一

**if**(isNavigating == **true**){

**if**(navMinute > 0){

                --navMinute;

            }

        }

    }, millisecondsPerMinute);

pauseTimer函数删除了time上绑定的setInterval，也就实现了时间流动的暂停。

当页面加载成功时，对切换倍速按钮绑定switchTimeRatio函数。该函数通过改变当前multi\_speed的值来改变startTimer函数中的millisecondsPerMinute，从而实现倍速的切换。

**3.2.3** 管理员功能

当管理员切换按钮被点击时，触发switchAdmin函数，改变isAdmin的值。当isAdmin为true时，将三个管理员功能按钮设置为可点击。

其中发布考试绑定了assignExam函数，当当前考试时间段与课程表中课程无冲突时，将新发布的考试加入到curriculumSchedul数组中属性Name为该考试课程名的元素中。

其中修改课程时间绑定了changeClassInfo函数，当当前时间确有该课程，且目标时间段与课程表中课程无冲突时，删除curriculumSchedul数组中属性Name为该课程名的元素，再向其中添加新课程。

其中布置作业绑定了assignHomework函数，找到curriculumInfo数组中属性Name为该课程名的元素，将新布置的作业添加到该元素的Homework数组中。

**3.2.4** 导航功能

使用百度地图开放api实现地图的呈现、标志建筑物的标记和路径的绘制功能。当开始导航按钮被点击时，触发navigation函数。

navigation函数中需要判断目的地输入的正确性，当目的地信息无误时，需要判断当前位置与目标位置是否处于同一校区。

当处于同一校区时，直接向后端发送起始地点与终止地点信息，后端经过计算返回路径等信息。之后调用startNavigation函数进行导航与路线的绘制。

当处于不同校区时，第一次导航向后端请求到达车站的路径并调用startNavigation进行导航与路线的绘制。当到达目的地后监听时间，若当天已无巴士时，则退出此次导航。若还有巴士，则监听持续到下班巴士到达。当巴士到达时继续监听时间，直到巴士到达目的地。此时开始第二次导航，向后端请求到达目的地的路径并调用startNavigation进行导航与路线的绘制。

若此时正在导航，则点击导航按钮或关闭页面将调用cancelNavigation函数取消导航。

若此时正在导航且时间流动，则点击计时按钮将调用pauseNavigation暂停导航。

若此时正在导航且时间暂停，则点击计时按钮将调用continueNavigation继续导航。

**3.2.5** 课程信息功能

点击所有课程窗口的条目和课程表窗口中的单元都会调用showSpecificCurriculum函数，向课程信息框中填充各类信息，之后弹出课程信息框。

showSpecificCurriculum = **function**(name){

    //将弹窗标题设置为该科目名称

    $('#specificCurriculum .modal-title').html(name);

    currentPageCurriculumName = name;   //当前科目名称

    refreshSpecClassAndExam(name);      //刷新该科目的课程和考试信息

    refreshSpecHomework(name);          //刷新该科目的已布置作业

    refreshSpecUploadHomework(name);    //刷新该科目的已上传作业

    refreshSpecUploadResource(name);    //刷新该科目的已上传资料

    $('#specificCurriculum').modal('show'); //弹出课程信息窗口

}

refreshSpecClassAndExam函数读取curriculumInfo数组中属性Name为该课程名称的元素中的共有信息，再从curriculumSchedul数组中读取属性Name为该课程名称的元素的所有课程信息，设置“课程信息”栏与“考试信息”栏的条目。

refreshSpecHomework函数读取curriculumInfo中属性Name为该课程名称的元素中的Homework数组，将其中所有元素用于设置“作业记录”栏的条目。

refreshSpecUploadHomework函数读取curriculumInfo中属性Name为该课程名称的元素中的Homework数组，将其中的uploaded数组中的所有元素用于设置“上传文件-作业”栏的条目。

refreshSpecUploadResource函数读取curriculumInfo中属性Name为该课程名称的元素中的Resource数组，将其中的所有元素用于设置“上传文件-资料”栏的条目。

在“上传文件-作业”栏中可进行作业上传，当点击确定提交时，触发uploadHomework函数，若已选择一项已布置作业、该提交作业文件类型为bmp或word，且该提交作业名称未与已提交作业名称重复，则将文件信息传递给后端进行处理。若后端查重发现该提交作业内容与已提交作业重复，则给前端发送信号进行提醒，且不保存该提交作业信息。反之则前后端都存入该提交作业信息。

在“上传文件-资料”栏中可进行资料上传，当点击确定提交时，触发uploadResource函数，若该资料文件类型为bmp或word，且该资料名称未与已有资料名称重复，则将文件信息传递给后端进行处理。若后端查重发现该资料内容与已有资料重复，则给前端发送信号进行提醒，且不保存该资料信息。反之则前后端都存入该资料信息。

refreshTodayClass函数读取curriculumSchedul数组中的元素，若该元素中存在属性Week与当前星期dayOfWeek相等的子元素（即一节课），且当前时间未超过该课程结束时间，则将其呈现在今日课程中。

**3.2.6** 活动信息功能

在我的活动窗口中可新增活动，当点击确认新增按钮时，触发addActivity函数，向activity数组中加入新活动，在按时间排序的同时调用refreshActivities函数刷新我的活动窗口。若新增活动的时间与课程冲突，会在窗口中出现红字提示，但依然能够添加该活动。

refreshActivities函数读取activity数组中的所有元素，将其逐个呈现于我的活动窗口中。

refreshTodayActivities函数读取activity数组中的元素，若存在属性Week与当前星期dayOfWeek相等的元素，且当前时间未超过该课程时间，则将其呈现在今日活动中。

当在我的活动窗口中点击某个活动右侧的“×”时，触发deleteActivity函数，从activity数组中删除该活动。

**3.2.7** 闹钟功能

在设置闹钟窗口中可新增闹钟，当点击确认新增按钮时，触发addClock函数，向clock数组中加入新闹钟，在按时间排序的同时调用refreshClocks函数刷新设置闹钟窗口。

refreshClocks函数读取clock数组中的所有元素，将其逐个呈现于设置闹钟窗口中。

当在设置闹钟窗口中点击某个闹钟右侧的“×”时，或仅响一次的闹钟时间到，触发clockTimeUp函数时，都会触发deleteClock函数，从clock数组中删除该闹钟。

前文提到过，time是setInterval的一个实例，会监听当前时间，当闹钟时间到时，调用clockTimeUp函数，弹出窗口进行信息提示。若该闹钟是仅响一次的闹钟，会调用deleteClock函数将该闹钟从clock数组中删除。

**3.2.8** 搜索功能

课程信息中的“作业记录”栏、“上传文件-作业”栏、“上传文件-资料”栏，所有课程窗口，与我的活动窗口这几处中都设置了搜索功能。

在课程信息中的“作业记录”栏，搜索键绑定refreshSearchedSpecHomework函数。若搜索框为空时按下搜索键，则不会给后端发送请求，而是将curriculumInfo中属性Name为当前课程名的元素中Homework数组中的所有已布置作业信息呈现在“作业记录”栏中；否则，则会向后端发送当前课程名与搜索框中的字符串，后端对已布置作业名进行匹配，返回给前端包含所有匹配成功的已布置作业的数组，前端将该数组中的所有元素呈现在“作业记录”栏中。

在课程信息中的“上传文件-作业”栏，搜索键绑定refreshSearchedSpecUploadHomework函数。若搜索框为空时按下搜索键，则不会给后端发送请求，而是将对应课程的所有已布置作业的所有已提交作业呈现在“上传文件-作业”栏中；否则，则会向后端发送当前课程名与搜索框中的字符串，后端对数据进行分析后返回给前端包含所有匹配成功的已布置作业的数组，前端将该数组中已布置作业的所有已提交作业呈现在“上传文件-作业”栏中。

在课程信息中的“上传文件-资料”栏，搜索键绑定refreshSearchedSpecUploadResource函数。若搜索框为空时按下搜索键，则不会给后端发送请求，而是将curriculumInfo中属性Name为当前课程名的元素中Resource数组中的所有资料呈现在“上传文件-资料”栏中；否则，则会向后端发送当前课程名与搜索框中的字符串，后端对数据进行分析后返回给前端包含所有匹配成功的资料的数组，前端将该数组中的所有资料呈现在“上传文件-资料”栏中。

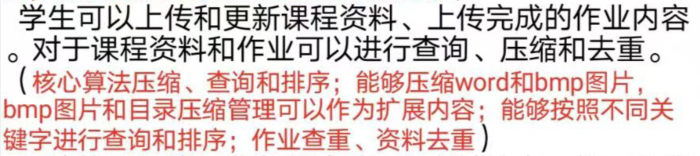
在所有课程窗口中，搜索键绑定refreshSearchedSubjects函数。若搜索框为空时按下搜索健，则不会给后端发送请求，而是将curriculumInfo中所有元素的Name属性作为每个条目的名称，给每个条目绑定单击弹出对应课程信息框的事件后，将条目加入所有课程窗口中；否则，向后端发送搜索框中的字符串，后端返回包含所有匹配成功的课程名称的数组，前端同上处理。

在我的活动窗口中，按下搜索键则调用refreshSearchedActivities函数，给后端发送请求，若搜索框为空则返回包含所有活动类型匹配成功的活动信息的数组，若搜索框不为空则返回包含所有既匹配成功活动类型、活动内容与搜索框内字符串又能匹配成功的活动信息的数组，前端将数组中所有信息呈现在我的活动窗口中。

**3.3** 后端

**3.3.1** 文件管理功能

任务要求对于此部分的要求是“学生可以上传和更新课程资料、上传完成的作业内容。对于课程资料和作业可以进行查询、压缩和去重。”



基本功能：压缩、管理、去重。

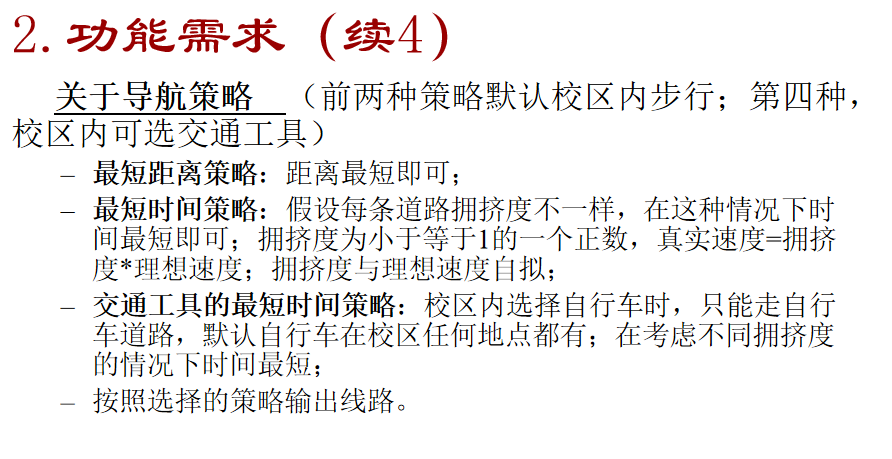
拓展功能：压缩word和bmp图片[[[2]](#endnote-1)]；目录压缩管理；MD5去重[[[3]](#endnote-2)]

实现方法：压缩功能使用的是上学期修改后的哈夫曼树，对于word功能，最多可以压缩5000000个字符的文本；目录压缩管理导入的Go语言的zip包，并**重写了**Zip方法，在前端页面按下“保存”即可生成file.zip；去重功能在前端对于文件名进行检测，如果相同则提示用户重复。

拓展：改进后的哈夫曼编码算法，使得bmp压缩同样能够实现；去重功能在文件名去重的基础上设置了MD5校验去重，在对不同的文件夹建立不同的哈希表，通过计算查表的方式进行唯一性校验。文本查重算法，我们了解到的有余弦相似度、TF-IDF模型、DSSM、编辑距离计算、杰卡德系数计算等，我们选择的是海明距离算法，因为查询资料得知，在相似度达到一定程度的两个字符串哈希值也几乎相似，我们先计算出两个文件字节流的哈希值（在Go中这个数据类型为int64），随后进行相除（大除小），最后用海明距离算法进行海明距离计算，给出查重率。[[[4]](#endnote-3)]

**3.3.2**  **导航**功能

任务要求此部分可以实现多种类的导航，如下：



基本功能：最短距离导航，最短时间导航，多交通方式导航

拓展功能：途径最短距离策略：途径多个地点的最短距离路径策略

实现方法：采用的是上学期所学习过的迪杰斯特拉算法，并且使用最小堆优化了其中的排序过程，使得算法复杂度降为了O(elogv)。(e为最大边数，v为顶点数)同时设定了步行理想速度为3m/s，自行车理想速度10m/s，通过前端的传入参数来选择函数中不同的导航模式。最后将导航的距离，时间以及途径点封装到一个Section结构内，转换为json后传递给前端。

拓展：对于途径多个点的最短距离算法，我们采用的思路是，首先对要求的途径点进行全排列（全排列采用递归进行），之后对每一个可能的排列方式都进行距离的测算，从中选择最短的距离途径顺序，并由此构建具体路径。

**3.3.3** 日志功能

任务要求为：“记录学生课程和活动状态变化，系统提醒的信息，输出的导航信息，以及学生输入的信息和各种查询操作。”

任务要求为：“记录学生课程和活动状态变化，系统提醒的信息，输出的导航信息，以及学生输入的信息和各种查询操作。”

基本功能：记录操作，保存记录

实现方法：日志模块位于 logger.go 中，包含 InitLog() 和 Log(content string) 两个函数。主程序在初始化阶段会首先调用 InitLog() 函数，完成对日志文件的创建，日志文件的命名格式为 yymmdd\_hhmmss.log，即系统的启动时间，以此区分不同的log文档。如果无法正常创建日志文件，则向控制台输出错误信息并结束程序。

在程序运行过程中，任何函数都可以随时调用 Log() 函数以产生日志，该函数可以将记录到的用户操作保存到之前建立的txt文档中，以此记录了用户的各种操作。而为了实现数据持久化的操作，我们采用的方法是覆写json文件，以此确保了服务端关闭重启也不会导致数据的丢失。

**3.3.4** 搜索功能

该模块采用分为两种搜索算法，一种为Trie（前缀树）模糊搜索，一种为RK判别搜索。其中Tire模糊搜索为高效查找算法，复杂度为O(n)。而RK判别搜索是在未要求核心算法部分采用的模糊搜索（比Trie模糊搜索更模糊），复杂度为O(n^2)

基本功能：Trie模糊搜索，RK判别模糊搜索

实现方法：

在题目要求高效查找的课程搜索和活动搜索部分，采用前缀树搜索。在搜索前就会遍历课程/活动名称列表来进行Trie建树操作，该过程复杂度为O(n)。之后，对用户所搜索的关键字进行前缀树模糊搜索，并将所有满足前缀的课程/活动都加入返回列表中，该过程复杂度为O(n)。其中活动和列表的结果将根据时间排序。

在题目未要求高效查找算法的资料搜索和作业搜索部分，采用RK判别搜索。也就是遍历所有成员的名称，对名称和搜索关键词进行RK算法匹配，并将满足条件的成员放入返回列表中。该算法比较前缀树模糊搜索的优势在于，该算法不仅可以匹配前缀，只要关键词在对应成员名称中出现过，均会出现在返回列表中。

**3.3.5** 排序功能

排序功能主要体现在课程、活动、作业和资料搜索的结果排序中。

对于课程搜索结果，提供了编号排序和拼音排序两种方式，对于活动搜索结果，提供了时间排序和拼音排序两种方式。而对于资料和作业搜索结果，默认为时间排序。

所有排序都采用的是选择插入排序，复杂度最差为O(n^2)。不同的排序方式仅是比较函数有所区别，其余地方一致。

**3.3.6** 配置文件读取

配置文件读取模块位于 loadConfig.go 中，包含函数 LoadConfig()，在主程序初始化阶段被调用。LoadConfig() 函数主要完成三项任务：打开配置文件、解析 JSON 和存储信息。

在执行以上三个步骤时，如果发生错误（如找不到配置文件、JSON 格式错误），会调用日志函数 Log() 产生错误日志，并退出程序。配置文件加载成功后，同样会输出日志。

**3.3.7** HTTP服务

HTTP 服务模块位于 server.go 中，包含 StartServer()、ServerTest() 和 APIHandler(w http.ResponseWriter, req \*http.Request) 三个函数。

主程序在初始化阶段完成配置文件加载后调用 StartServer() 函数以启动 HTTP 服务，若启动失败，会产生错误日志并告知用户可能的错误原因。由于 HTTP 监听进程是阻塞的，所以主函数需单独开启一个 HTTP 服务进程，以保证其他模块可以并行执行。

在开始监听后，StartServer() 函数会调用 ServerTest() 函数以测试服务是否已经在指定端口正常运行，确认服务正常后，产生日志并提示用户通过浏览器访问系统。

HTTP 服务的路由由 APIHandler() 函数负责，并且处理前端发来的各种 API 请求。

**4** 用户指南

**4.1** 安装依赖

所需环境： [Go](https://mirrors.tuna.tsinghua.edu.cn/qt/official_releases/online_installers/) 1.18

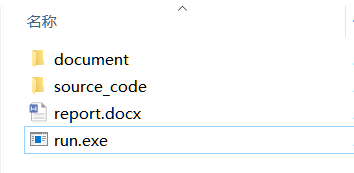
在构建前需要确保您安装了 1.18 或以上版本的 Go 语言环境。可以通过设置镜像源的方法加快下载速度。

**4.2** 编译项目

在 src 目录中执行以下命令来编译（以 windows10 为例）：

|  |
| --- |
| cd src |
| go build -o run.exe |

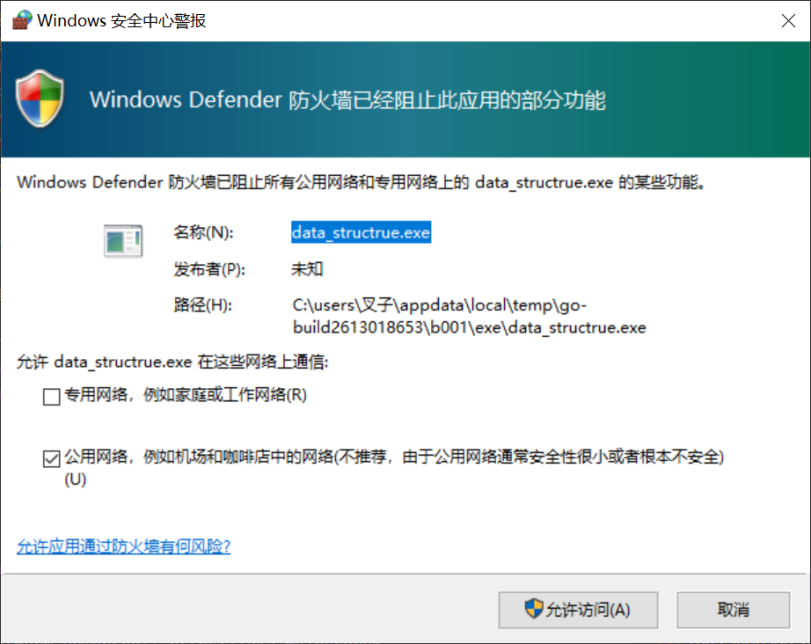
在构建完成后，需要将run.exe移至根目录下。如图所示：



**4.3** 使用说明

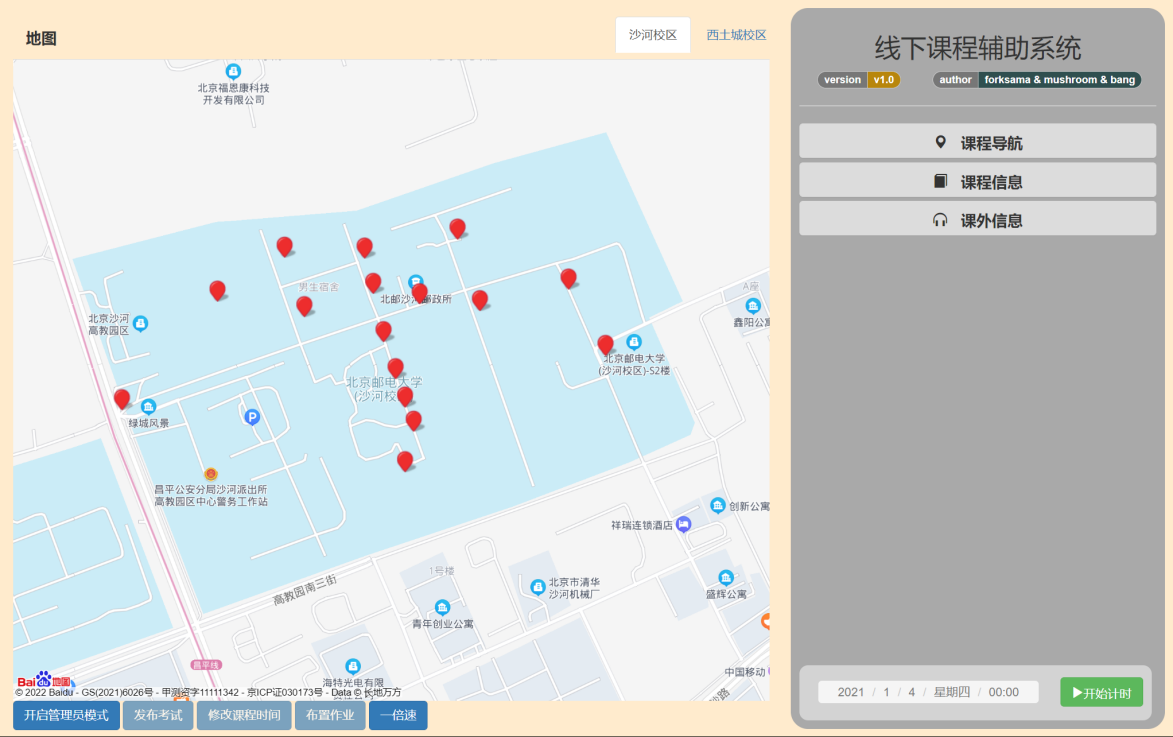
**4.3.1** 运行系统

在控制台中执行代码 go run .\ 命令即可启动线下课程辅助系统的后端服务器，或者直接运行我们准备好的run.exe文件。如果您使用的是Windows操作系统，那么在初次运行时可能会被防火墙阻挡导致程序无法正常工作，请您**允许**程序通过防火墙并在网络中通信（程序的导航系统需要接入互联网）。



当在控制台中看到“初始化完成”及“Web 服务启动成功”字样的日志时，代表系统正常启动，您可以按照提示在浏览器中访问 <http://localhost:port> 以使用系统，其中 port 替换为您在配置文件中设置的端口号（在config.json中，默认为9090）。

**建议使用Chrome浏览器访问该网址**，其他浏览器可能因为不支持某些特性导致显示异常。



进入 Web UI 后，您可以看到左侧的**地图面板**、右侧的**信息面板**。

在**地图面板**中，您可以看到在百度API下地图上所有被标记出的标志地点，单击标志可以显示地点名称。点击地图面板右上方的两个校区按钮将切换校区。此外，当导航正在进行中时，导航的线路将实时显示在地图面板中。

在**信息面板**中，包含课程导航、课程信息和课外信息。您可以在信息面板中看到模拟器当前的时间（精确到分钟）。在课程导航面板中，您可以进行导航信息的设置，同时可以查看当前导航的状态。在课程信息面板中，您可以实时查看当天尚未结束的课程，同时可以查看课程表。在课外信息面板中，您可以实时查看当天尚未结束的活动，同时可以设置活动、闹钟。



**4.3.2** 导航规划

若您想要进行导航，请点击打开右侧信息面板中的课程导航，选择您的当前位置、选择一种目标位置，并采用一种出行方式。

我们以在 x 时 y 分 z 秒从 AA 到 BB ，途径 CC 且不考虑拥挤度的步行策略为例，点击“开始导航”按钮后，系统即可为您进行路线规划。若导航按钮样式变为红色，且内部文字变为取消导航，则说明规划成功：



**4.3.3** 课程表

用户可以点击课程信息的“课程表”按钮查看当前课程表，点击课程表的某一节课可以跳转到相应的课程信息窗口。管理员模式下修改的课程信息和发布的考试会实时反馈在课程表中。



**4.3.4** 所有课程

用户可以点击课程信息的“所有课程”按钮查看所有课程，点击其中某一个条目可以跳转到相应的课程信息窗口。



在课程上方的搜索框中输入课程名称的前缀，选择排序方法，点击搜索，窗口中的条目会变为匹配成功的所有课程。



**4.3.5** 课程信息

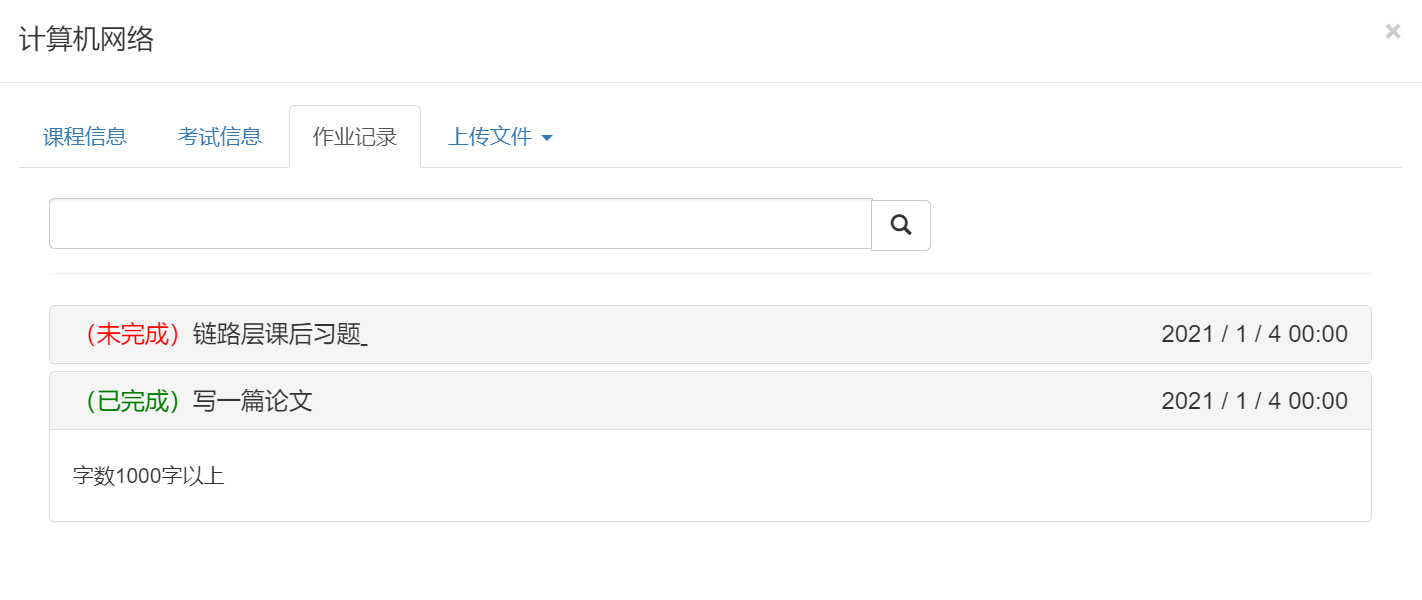
通过课程表窗口和所有课程窗口都可以进入对应科目的课程信息窗口。在“课程信息”栏中罗列了该科目的所有课程信息。



在“考试信息”栏中罗列了该科目已发布但尚未结束的考试信息。



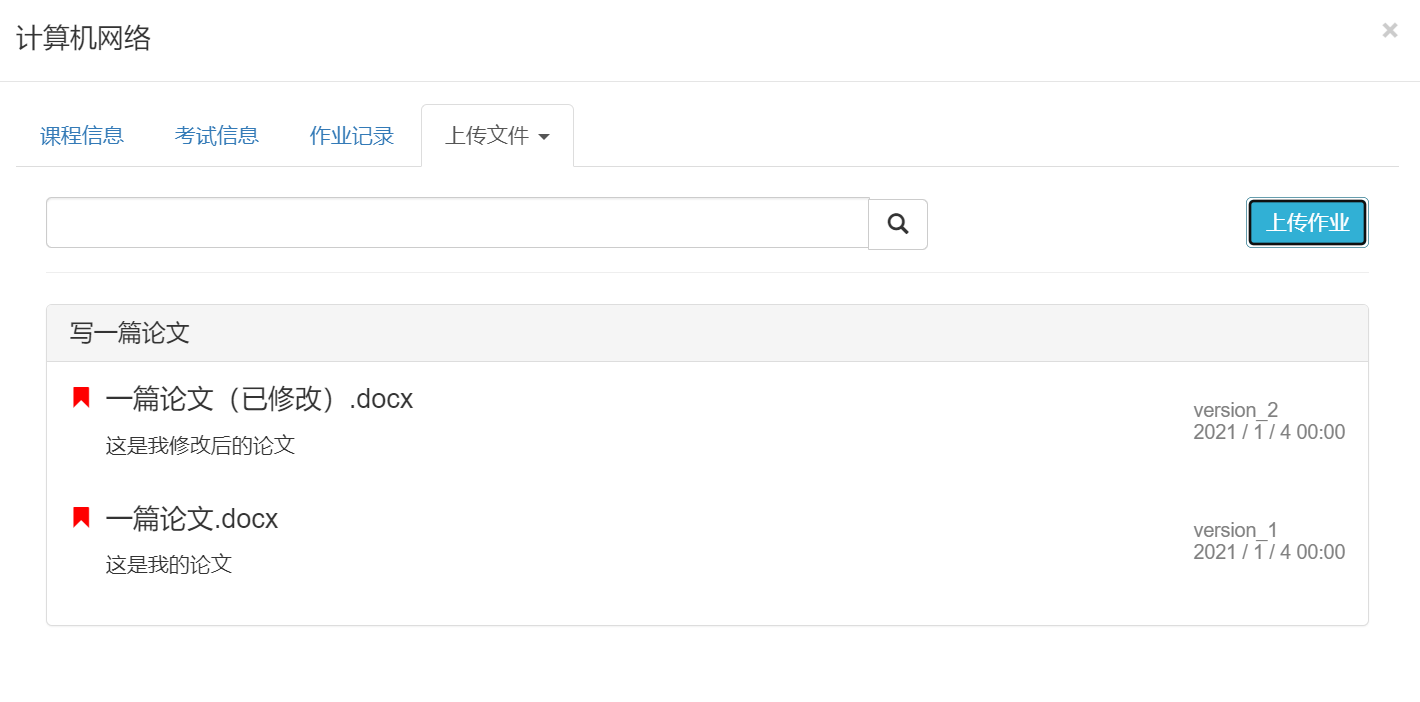
在“作业记录”栏中罗列了该科目已布置的所有作业信息，点击作业名称可展开查看作业内容。可以在上方的搜索栏中输入已布置作业的名称进行搜索。



点击“上传文件”，出现包含“作业”与“资料”的下拉框，选择“作业”则进入上传作业模块，选择“资料”则进入上传资料模块。



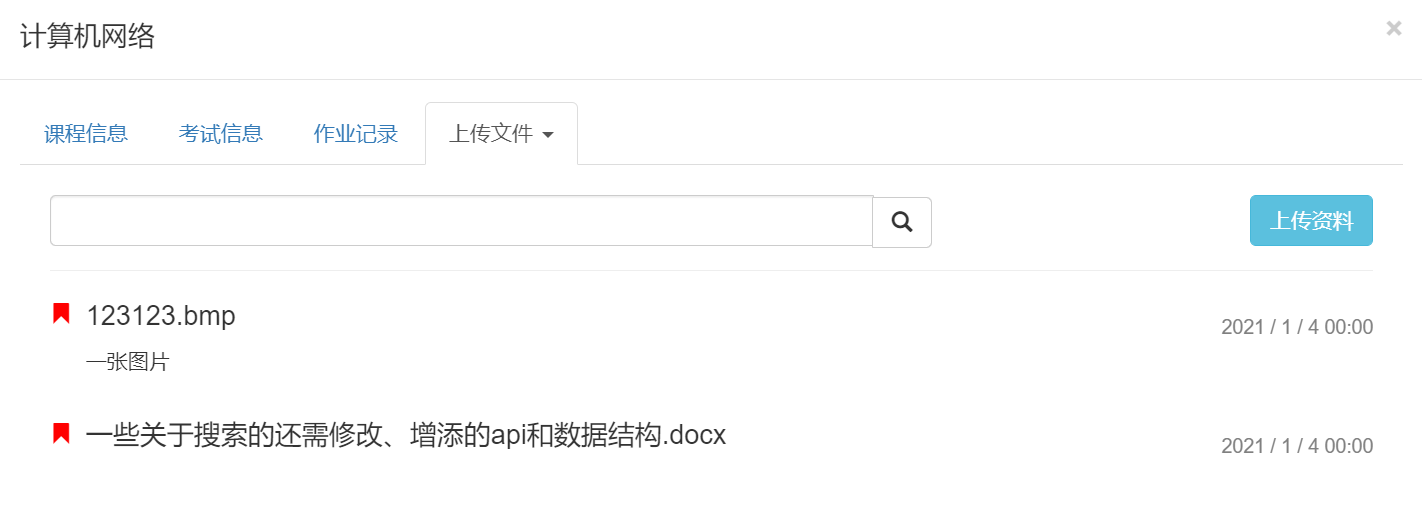
在上传作业模块中可查看当前已提交的作业，点击已布置作业名展开已提交作业条目。其中可见每次提交的作业时间、版本等信息。可以在上方的搜索栏中输入已布置作业的名称进行搜索。



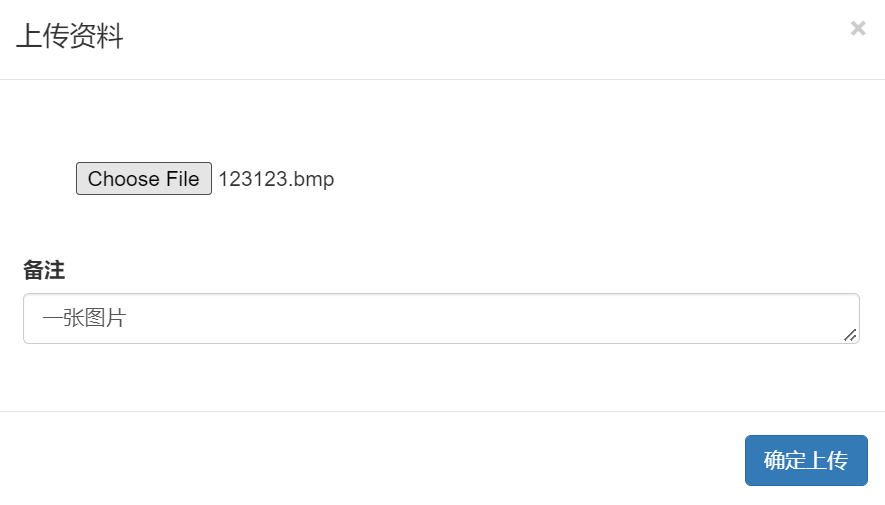
在上传作业模块中，点击右上角的上传作业按钮，可进行作业提交。提交的作业保存在路径./file/Homework 文件夹中。当文件夹中已有作业与待提交作业重名或内容重复时，在左下方提示提交失败。



在上传资料模块中可查看当前已上传的资料，可见上传资料的时间等信息。可以在上方的搜索栏中输入资料名称进行搜索。



在上传作业模块中，点击右上角的上传资料按钮，可进行资料上传。上传的资料保存在路径./file/Resource 文件夹中。当文件夹中已有作业与待提交作业重名或内容重复时，在左下方提示提交失败。



**4.3.6** 今日课程

用户可以在课程信息面板中查看当日尚未结束的课程信息。



**4.3.7** 我的活动

用户可以点击课外信息的“我的活动”按钮查看当前所有活动，点击某活动右侧的“×”可以删除该活动。



点击窗口最下方的“新增活动”按钮，可以新增活动。若活动时间与课程之间存在冲突，会在窗口左下角提示出来。



**4.3.8** 设置闹钟

用户可以点击课外信息的“设置闹钟”按钮查看当前所有已设置的闹钟，点击某闹钟右侧的“×”可以删除该闹钟。



点击窗口最下方的“新增闹钟”按钮，可以新增闹钟。闹钟分三种类型：仅响一次、每天一次、每周一次。仅响一次的闹钟到点提示后自动删除，而每天一次与每周一次的闹钟不会自动删除。



到达闹钟设定时间，会弹出窗口提示，同时时间暂停流动。若该闹钟为仅响一次的闹钟，此时闹钟会被删除。



**4.3.9** 今日活动

用户可以在课外信息面板中查看当日尚未结束的活动信息。



**4.3.10** 时间控制

用户可以在信息面板右下角查看当前时间，以及开始、暂停计时。若当前正在进行导航规划，则导航路线动画随着时间流动的暂停与继续变化。点击页面下方“倍速”按钮可调节时间流动速度。



**4.3.11** 管理员模式

点击页面左下方“开启管理员模式”按钮后，可进行发布考试、修改课程时间、布置作业三项操作。



1 开启管理员模式前



2 开启管理员模式后

管理员可以点击“发布考试”按钮，于某个时间段在某个地点发布考试。考试可在课程表中查看。考试将在进行结束被自动删除。请注意：若该时间段已存在课程或考试，将提示时间冲突，且无法确认发布。



管理员可以点击“修改课程时间”按钮，修改某个时间的课程信息。可以将课程移动到另一个时间，也可以修改课程的地点。请注意：若该时间不存在该课程，将提示时间有误，且无法确认修改。



管理员可以点击“布置作业”按钮，布置某个科目的作业。



**5** 测试评估

**5.1** 范例执行结果及说明

测试样例一

**导航规划**：

规划时间：2021年1月5日 星期日 08:26

当前位置：西土城校区-星光老年之家

目标地点：西土城校区-西门

途径地点：西土城校区-报刊亭、西土城校区-公寓10号楼

出行方案：步行（不考虑拥挤度）

**导航路线：**



**导航分析：**

同校区内导航，用时 11 min，于 08:43 到达目的地。

测试样例二

**导航规划**：

规划时间：2021年1月6日 星期一 08:00

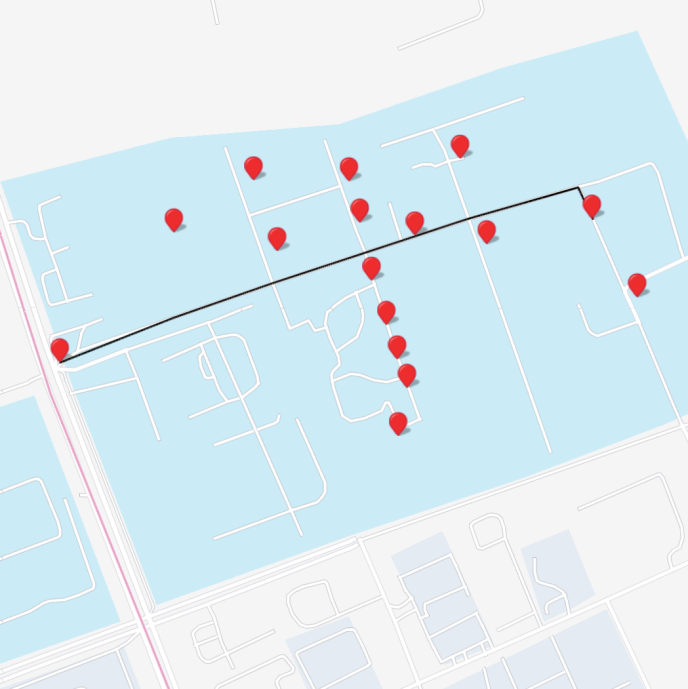
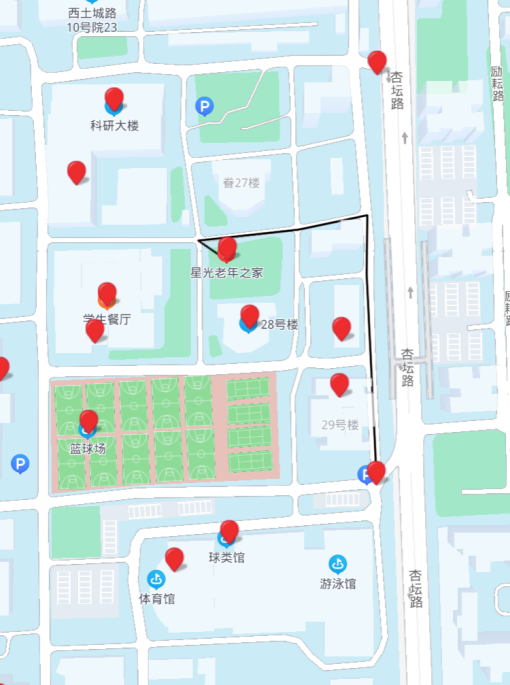
当前位置：西土城校区-星光老年之家

目标课程：计算机组成原理

途径地点：无

出行方案：自行车

**导航路线：**



**导航分析**

西土城内导航至东门车站，用时 1min。

等待最近时间 9:50 的巴士到来。

巴士通往沙河校区用时 1h。

沙河校区内从沙河大门开始导航，用时3min。

共计用时 2h 53min，于 10:53 到达目的地。

**5.2** 评价和改进

评价

以下将从系统的**健壮性、正确性、高效性**和**易读性**四个方面对该系统进行评价：

**健壮性**

程序逻辑严谨，模块分工明确，在各个环节都对所有可能出现的情况进行了周密考虑，且对可能出现错误的环节（如配置文件读取、日志文件写入等）进行了错误处理，在资源文件夹不存在时，检测并创建，最大程度避免了程序致命错误的产生。系统对进行和变量进行严格的控制，避免了内存泄漏等问题。

**正确性**

规划算法使用堆优化的 Dijkstra 算法，保证所得路线为最优解。系统各环节均能保证正确运行，无重大错误产生。

**高效性**

规划算法使用堆优化的 Dijkstra 算法，一般情况下的时间复杂度为 O(N·logN)，而在多个途经点下的时间复杂度为阶乘复杂度。在各种复杂交通线路配置下都具有良好的性能。

**易读性**

在编码过程中符合规范 Golang 官方编码规范，程序采用多文件结构，各模块间符合高内聚、低耦合的设计原则，且代码中有较为详细的注释，具有良好的可读性。

改进

旅行规划与模拟系统接下来可从以下几个方面进行改进：

- 对 Web UI 进行优化，提升用户体验；

- 完善默认配置文件预设的地图及交通信息，增加巴士班次数量以更加贴近实际；

- 提供可定制化的日志服务。

**6** 实验总结

**6.1** 实现的难点、遇到的问题和解决方案

导航动画问题

通过百度地图API实现寻路动画的绘制虽然方便快捷，但是也出现了一些问题。由于API功能上的缺失，导致程序难以监听导航动画结束这一事件，从而直接影响了动画结束时导航按钮的样式的转变、倍速按钮可点击状态的恢复、起始与终点校区不同时当前状态转变为等待巴士等一系列功能设想。

经过一系列探索，最终解决方案是设置全局变量 navMinute 等于导航动画时间，当时间流动时且 navMinute > 0 时，在 startTimer 函数中的 setInterval 递减该 navMinute 变量，监测到 navMinute == 0 时，执行导航动画结束应执行的功能。

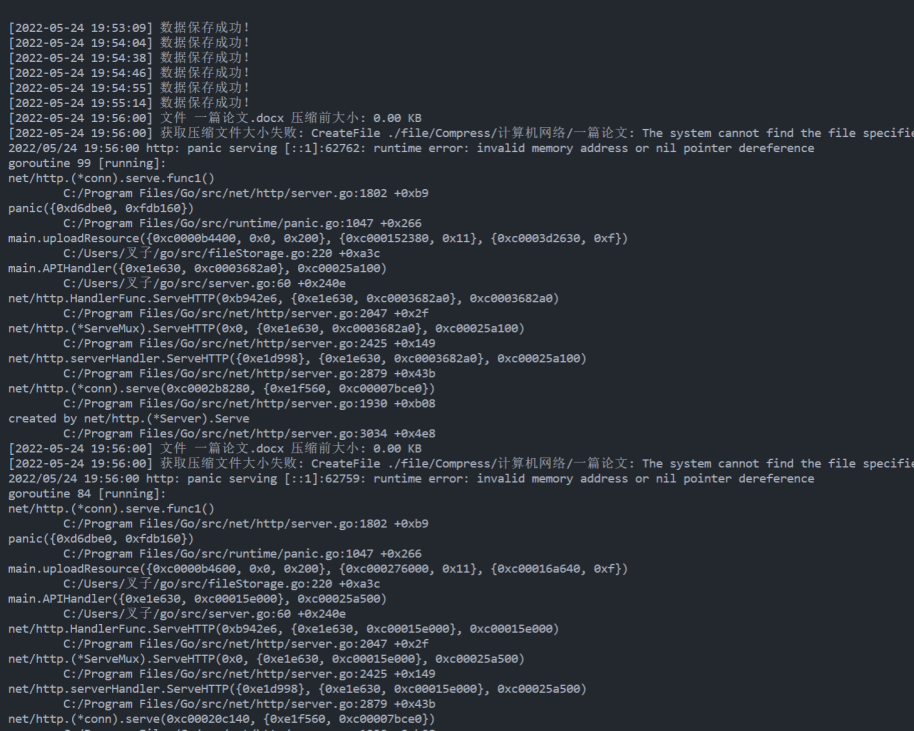
然而在 navigation 函数中，对于导航起始与终点校区不同的情况，由于为了监听事件发生而嵌套了多个 setInterval 在上一级的回调函数中，导致代码阅读性减弱、效率降低。可以考虑引入能够监听变量是否等于特定值的框架，或是修改程序架构，在给须监听变量赋值时，绑定一个事件。这么做实现了变量值的监听，提高了代码的可读性。

文件夹建立问题

文件管理函数在运行初期，会对相应的文件夹进行存在检测，如果不存在则创建。在特定情况下，创建文件夹时会出现报错的情况。

首先我们考虑到可能是中文文件名的问题，遂改成英文，但是无果，另寻他路。

然后，我们考虑到可能是文件夹层级太多的问题，减少了文件夹层级再次运行成功。（报错截图如下）



前后端通信问题

本组成员之前考虑采用Qt框架构建服务端和客户端的通信，但本次项目采用Go语言和前端三件套，因此服务端的构建是初期难住我们的一个难点。

通过查阅资料，后端在短时间内实现了Go语言的服务端的建立，但在前端框架的采用上出现了分歧。既有组员提议采用vue框架使用的axios，也有组员认为serverlet是不错的的选择。

最后经过讨论，我们一致认为使用目前较为流行且上手难度不大的jqeury作为前端框架的模板。事实证明，这一选择确实省去了我们许多的麻烦，将前端繁冗的通讯构建代码缩减为了几行代码。

多线程处理

为了提升项目的易用性和可操作性，我们最初设计了多线程架构，即支持多个用户同时进行导航，但受限于百度地图API的限制，一次只能显示一个用户的导航路线，无法明显展示多线程处理，最后舍弃了多线程的设计。

**6.2** 成员心得

**陈禹璋：**在本次实验中负责前端代码的编写与GUI的设计，这是一项未曾经历过的工作，因此于此次实验中锻炼了自己对代码架构的认识，如何优化代码、进行注释才能提高代码的可阅读、可维护性，以及该怎样对代码的输入进行全方面的测试这几项能力。在该项目中对代码进行了大量测试分析，在不同条件下严苛测试了算法的健壮性，并多次对前端代码进行重新编纂以优化代码的可读性与代码的效率。比较可惜的是即使如此，前端算法的时间复杂度并不算低，可以考虑采用更加优化的方法。实验是大二以来全新的一次体验，在项目推进过程中难免会遇到一些阻力。但正因如此，认识到团队意见高于一意孤行的道理。

**吴建军：**在项目推进和代码修改中进行了全局的数据分析，对整体功能进行了构思之后，设计整体代码模块划分、框架架构以及部分后端代码编写的任务，认识到了在小组项目中代码设计是重中之重，必须精意覃思而不可虎头蛇尾；利用更加高内聚、低耦合的方式设计代码是提高程序模块的可重用性、简洁明了程度的关键。于此同时，团队合作沟通也是其中的不可或缺的一部分，成品代码的易于维护、及高水平算法四个评价标准的实现正是小组成员中交流沟通的结果。同时，本次实验对于Web前端开发工程与Go语言底层原理有了更好的了解。

**吴镇均：**在对所需功能进行合理分析之后，设计了整体数据结构和后端具体模块分工，用Go语言完成了后端框架的构建和文件操作以外的所有算法代码的编写。Go相比cpp有独特的库和方法，因此考虑采用使用Go语言进行后端服务器源程序的编写，自学一门全新的语言不仅让我对于数据结构和算法的熟练程度更加深刻、对输入输出流设计理念的理解更加精进，对于后端服务器的构架也有了更上一层楼的看法。提出了输入输出可视化的理念，并在小组成员的积极讨论下将此理念具现化。于此次小组合作课程设计实验中学会了更加主动地查阅相关资料、积极与组员、老师进行沟通交流，也正因此在同他人交流沟通、自我学习两方面的能力得到了进一步的提升。

7 致谢

光阴荏苒，物换星移，转眼间大二下学期即将结束，全程线上的一个学期给课程带来了许多挑战和压力。虽然因为疫情的影响不能回到美丽的北邮校园，但这一学期也我们也付出了相当多的努力。

首先要感谢计算机学院的老师们，辛苦备课和设计题目，全面考查了数据结构课程的所学知识，并提出拓展要求，极大提高了我们的自学能力和资料搜寻能力，在不断学习中有所收获。

其次，要感谢杨震老师和助教老师。老师们认真负责，积极解答同学们的问题，热心分享学习资料，有效化解了同学们的疑问，是我们学习生活中的催化剂。

最后要感谢各位小组成员，大家积极提出问题、共同讨论解答，携手并进，一起解决了一个又一个难题，最终翻越崇山峻岭、登峰造极，看到了最美丽的风景。

感谢大家！小组成员们的成果或不尽至臻，但已赴全力。若有不及人意之处，还尽老师提出指点。在此再次向杨老师、助教老师及课程设计中所给予支持建议的学长们致以崇高的谢意。在今后的日子里，我们仍然会不懈努力追求，勇攀高峰！

陈禹璋 吴建军 吴镇均

8 参考资料

1. [] [Go 中文开发手册 - 开发者手册](https://www.baidu.com/link?url=zYwaomo9eaUDfDc3BbcVJgPCX6OotXhv6Y88JUxipcDXkzF-y-wsF-hlCAATENpyYudyVvP9nicrQPYhdwFvwa&wd=&eqid=acfbbbf1000ca41300000006628c2a44" \t "https://www.baidu.com/_blank)：https://cloud.tencent.com/developer/doc/1101 [↑](#endnote-ref-0)
2. [] BMP图像文件完全解析：https://zhuanlan.zhihu.com/p/260702527 [↑](#endnote-ref-1)
3. [] MD5校验：https://baike.baidu.com/item/MD5%E6%A0%A1%E9%AA%8C/4966883?fr=aladdin [↑](#endnote-ref-2)
4. [] 计算文本相似度的常用算法：https://blog.csdn.net/weixin\_42510210/article/details/114483739 [↑](#endnote-ref-3)