

# 第五周周报

组号	班级	姓名	学号
27	2022211305	胡宇杭	2022212408
27	2022211305	孟林	2022210484
27	2022211305	陈炳璇	2022211479

## 前端部分

### 一. 环境配置

本组项目的前端部分使用Vue2脚手架环境进行编写, IDE为VsCode

### 二. 模块设置

首先我们配置了一个脚手架环境, 配置好路由和Vuex仓库  
其中我们的模块和路由配置分为8各部分:

- 1. 首页-Home
- 2. 登录页-Login: 用户输入学号和密码执行登录操作
- 3. 日记首页-HomeDiaries: 负责展示推荐的日记, 顶部配有搜索栏, 可以输入关键字进行日记检索
- 4. 地图首页-HomeMap: 可以展示地图和推荐的地点, 顶部配有搜索栏, 可以输入关键字进行地点检索
- 5. 搜索界面-SearchIndex: 是点击地图首页的搜索栏后进入的界面, 可以展示搜索结果
- 6. 地图详情页面-MapDetail: 负责展示地图详情部分, 显示地点内部的场景和道路, 显示路径规划功能
- 7. 日记详情页-DiaryDetail: 点击日记首页的日记后会根据日记ID自动跳转, 展示日记详情, 可以执行点赞, 评分, 评论等功能
- 8. 日记编辑界面-DiaryEdit: 用户使用登录信息进行日记的编写和发布功能

> mock

▼ router ●

JS index.js M

▼ store ●

▼ diary ●

JS index.js U

▼ login ●

JS index.js U

JS index.js M

▼ views ●

▼ DiaryDetail.vue M

▼ DiaryEdit.vue M

▼ HomeDiaries.... M

▼ HomeIndex.v... M

▼ HomeMap.vue M

▼ LoginIndex.vue M




▼ MapDetail.vue M

▼ SearchIndex.... M

▼ SearchResult... M

▼ App.vue M

JS main.js M

 .gitignore  
 babel.config.js  
 jsconfig.json

```
src > router > JS index.js > [🔍] default
4   import MapDetail from '@views/MapDetail.vue'
5   import Search from '@views/SearchIndex.vue'
6   import SearchResults from '@views/SearchResults.vue'
7   import DiaryDetail from '@views/DiaryDetail.vue'
8   import DiaryEdit from '@views/DiaryEdit.vue'
9   import Home from '@views/HomeIndex.vue'
10  import HomeMap from '@views/HomeMap.vue'
11  import HomeIndex from '@views/HomeIndex.vue'
12  import HomeDiaries from '@views/HomeDiaries.vue'
13
14  Vue.use(Router)
15
16  export default new Router({
17    mode: 'history',
18    routes: [
19      {
20        path: '/login',
21        name: 'login',
22        component: Login,
23      },
24      {
25        path: '/mapDetail',
26        name: 'mapDetail',
27        component: MapDetail,
28      },
29      {
30        path: '/search',
31        name: 'search',
32        component: Search,
33      },
34      {
```

## 三. 细节展示

目前由于后端部分还在进行配置, 所以我们的所有静态测试数据均存放在Vuex仓库中. 当后续后端接口完成, 我们将改为动态从后台拿数据到Vuex中, 再被Vue组件来使用.

目前已经完成的组件和功能:

### 1. 登录页面和首页联动

可以输入学号和密码进行登录操作, 目前在后台只存放了一个用户的登录数据, 输入正确的学号和密码后进入首页页面

学号	请输入学号
密码	请输入密码
登录	
返回	

# 头页

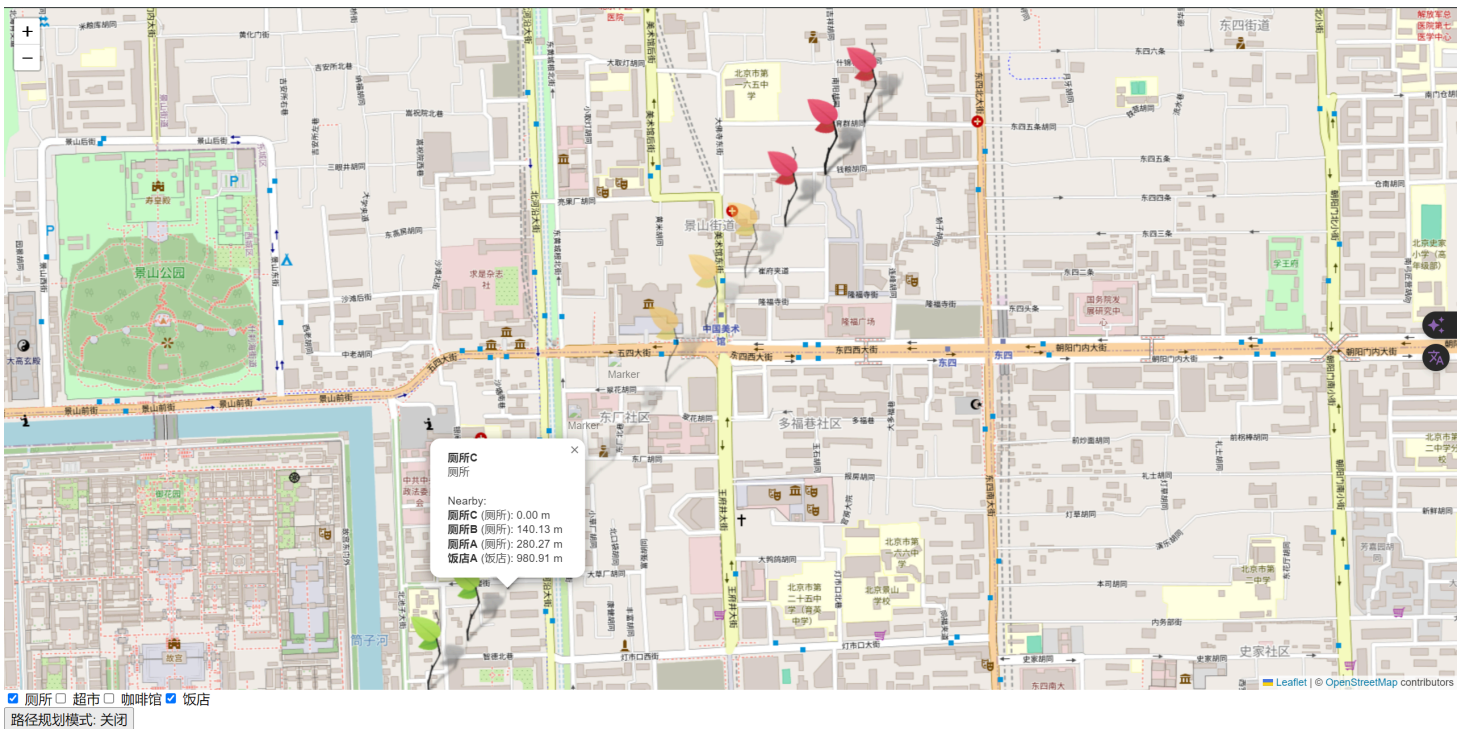


各务原抚子

退出登记

## 2. 地图详情页

我们使用了leaflet地图库进行地图展示和地点标识. 目前已经实现的功能: 可以从Vuex中获取静态的点位数据并摆放在地图上, 点击地图上的图标, 勾选下方的多选框也能展示周围已经选中类型的地点的距离(并按照距离从近到远进行排序)



```
src > store > login > JS index.js > ...
1  // store/login/index.js
2  const mockUser = {
3    | studentId: '111111',
4    | password: '123456',
5    | nickname: "各務原なでしこ",
6    | avatar: require('@/assets/fujisan.jpg'),
7    | diary: {
8    |   | title: "四尾连湖でのゆるゆるキャンプ日記: 抚子と凜の小さな冒険",
9    |   | text: "今日は凜と一緒に、四尾连湖へキャンプに来たよ..."
10   | }
11 };
12
13 export default {
14   | namespace: true,
15   | state: () => ({
16   |   | isLoggedIn: !!localStorage.getItem('user'),
17   |   | user: localStorage.getItem('user') ? JSON.parse(localStorage.getItem('user')) : mockUser, // 初始
18   | },
19   | mutations: {
20   |   | setLoginState(state, { isLoggedIn, user }) {
21   |   |   | state.isLoggedIn = isLoggedIn;
22   |   |   | state.user = user;
23   |   |   | if (isLoggedIn) {
24   |   |   |   | localStorage.setItem('user', JSON.stringify(user));
25   |   |   | } else {
26   |   |   |   | localStorage.removeItem('user');
27   |   |   | }
28   |   | },
29   |   | updateDiary(state, { title, text }) {
30   |   |   | state.user.diary.title = title;
31   |   |   | state.user.diary.text = text;
```

### 3. 日记部分(已经完成CSS修饰工作)

日记部分分为三块: 日记首页, 日记详情页和日记编辑页面

日记首页可以从vuex中的两个文件夹下的js文件获取用户的登录信息(为了执行点赞, 打分和评论的操作)和日记信息



```

src > store > diary > JS index.js > [E] default > getters
1  const mockDiaryData = [
13    comments: [
27      nickname: "耐摔王",
28      avatar: require('@assets/laoda.png'),
29      time: "2023-01-03 09:45",
30      text: "孩子们这并不好笑",
31      likes: 3,
32      liked: false,
33    ]
34  ]
35  }
36  // 可以添加更多的日记数据
37  ];
38
39  // store/diary/index.js
40  export default {
41    namespace: true,
42    state: () => ({
43      diaries: mockDiaryData,
44    }),
45    mutations: {
46      setRating(state, { diaryId, rating }) {
47        const diary = state.diaries.find(d => d.id === diaryId);
48        if (diary) {
49          diary.rating = rating;
50        }
51      },
52      deleteComment(state, { diaryId, commentId }) {
53        const diary = state.diaries.find(d => d.id === diaryId);
54        if (diary) {
55          diary.comments = diary.comments.filter(c => c.id !== commentId);

```

点击首页的某个日记(目前仅有一条测试数据)后会根据该日记的ID跳转到该日记的详情界面, 详情界面会展示日记全文, 点赞数量, 打分选项和评论功能, 该页面也抓取了用户的登录信息, 用户可以发表评论和删除评论, 或者给其他评论点赞

[< 返回](#)

编辑日记



各務原なでしこ

2024-03-28

## 四尾连湖でのゆるゆるキャンプ日記：抚子と凜の小さな冒険

日記へ、今日は凜と一緒に、四尾连湖へキャンプに来たよ...

♡ 99

💬 评论



志摩リン

2023-01-02 14:30

次も一緒に行こう！

♡ 5



耐摔王

2023-01-03 09:45

孩子们这并不好笑

♡ 3



各務原なでしこ

2024-03-31 13:49

我简直要笑晕过去了

♡ 0

🗑 删除

最后是日记编辑功能, 用户可以编辑日记并发表.



[< 返回](#)

编辑日记



各務原なでしこ

日记标题

日记内容

保存

## 四、下周计划

1. 完成日记模块
2. 完成地图主页和搜索模块的主要功能和CSS修饰
3. 完成所有模块的路由部分

## 后端部分

### 总览

后端部分，我们决定花大约 3 周时间实现数据库和网络通信模块，因为这是我们项目的基石，然后再转入对具体项目模块的编写。

### 一. 目前工作进度

1. 配置了 *rbenv*、*Ruby* 和 *Rspec* 环境
2. 学习了如何使用 *Ruby* 编写 *Rspec* 测试程序对项目模块进行测试

3. 学习了数据库的基本结构，阅读了部分 *SQLite* 源码
4. 实现了数据库的基本存储功能
  - 实现了 *table*、*pager*、*cursor*、“虚拟机”类及相关方法的编写
  - 实现了基于 *B+* 树的节点插入功能
  - 实现了数据的序列化和反序列化功能
  - 实现了数据持久化

## 二、下周工作规划

1. 继续学习数据库相关知识
2. 实现 *B+* 树的节点分裂功能
3. 实现基于 *B+* 树的数据搜索功能
4. 实现对数据缓冲区的动态调整（基于 MySQL 的事务机制）
5. 实现基于 *B-* 树的联合检索功能
6. 基于 *libhv* 网络库和 *json* 库实现与前端的通信