

教育部高等学校大学计算机课程教学指导委员会

中国大学生计算机设计大赛



软件开发类作品文档简要要求

作品编号： 63536

作品名称： T-Campus 智慧视觉校园

作 者： 邱翼钦 罗港霞 马昊霖

版本编号： V 1.0.0

填写日期： 2019-5-11

填写说明：

- 1、本文档适用于**所有**涉及软件开发的作品，包括：软件应用与开发、大数据、人工智能、物联网应用；
- 2、正文一律用五号宋体，一级标题为二号黑体，其他级别标题如有需要，可根据需要设置；
- 3、本文档为简要文档，不宜长篇大论简明扼要为上；
- 4、提交文档时，以 PDF 格式提交本文档；
- 5、本文档内容是正式参赛内容组成部分，务必真实填写。如不属实，将导致奖项等级降低甚至终止本作品参加比赛。

目 录

| | |
|-----------------|----|
| 第一章 需求分析..... | 3 |
| 第二章 概要设计..... | 3 |
| 第三章 详细设计..... | 6 |
| 第四章 测试报告..... | 14 |
| 第五章 安装及使用 | 25 |
| 第六章 项目总结..... | 29 |

第一章 需求分析

1.1 项目简概

该应用主要面向对象为在校学生以及校方管理人员,旨在为校园内的信息流动提供一个综合性的移动应用平台,囊括信息共享,信息处理等关键性功能。

1.2 项目背景及可行性分析

经过调研及资料查询,我们发现大多数校园服务平台上的信息流动渠道都为单向性,即学生向应用提交信息,仅后台管理人员可见,且不同信息类别提交需要不同接口,大大降低了功能的便捷度,平台不能以一种快速,简便,直观的方式去让使用者直接反映校园中遇到的问题或是想要交流的信息,并且基于这个问题进行快速的反馈以及联系有关方面进行处理,例如公共设施损坏的报修。

信息可读性几乎为零,在校使用者往往无法通过一种简单直观的方式,实时获知周边出现什么问题,有什么样的信息,缺乏一个信息共享的交互平台,缺少统一的渠道分享信息,导致许多学生无法有效的获取所需的相关信息,例如座谈会的举办时间及地点。

在校使用者只参与信息的提交,无法获取其它提交信息,这对校园的建设无疑是不利的。所以我们希望搭建一个综合性的智慧校园平台应用,使校园内的各式信息与应用使用者有更良好的交互。

1.3 项目功能需求分析

1.3.1 为了提供更方便快捷的信息提交方式,校园使用者可以在应用上选择事件分类,通过定位以图,文,视频等多种方式发布校园中的信息,并在应用后端被校方管理人员审核及处理。

1.3.2 为了使信息有更直观的表现方式及交互能力,应用以地图为信息载体,将所有已通过审核的信息以地图气泡的方式直接显示在地图界面。

1.3.3 为了提高信息的公开及交互功能,每个校园使用者都能通过地图事件气泡框看到周边出现了什么问题,以及问题所在的处理阶段,或是活动举办的通知,以上信息都将以基于地图的定位点的形式展现,并附有信息详情界面,信息高度公开,交互清晰直观。

第二章 概要设计

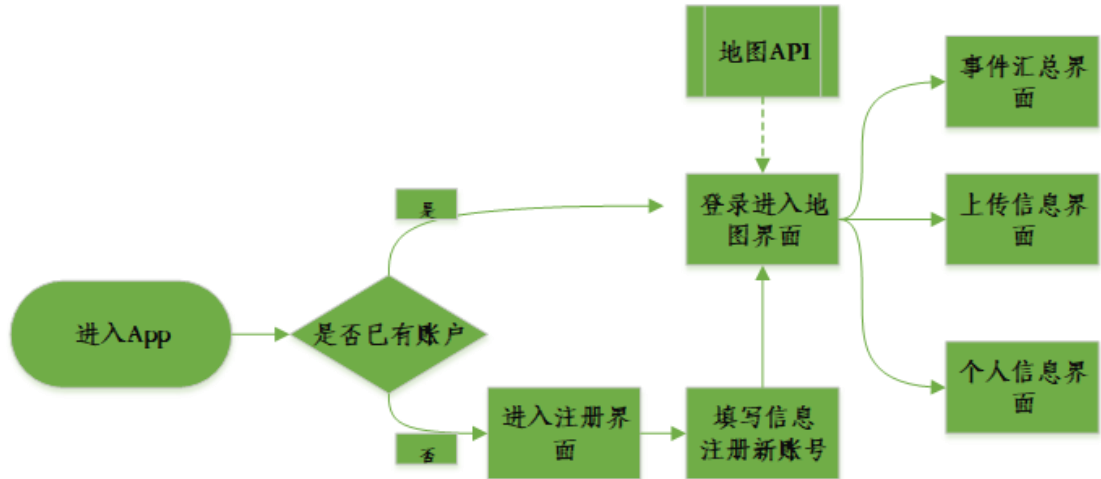
2.1 设计简概

该智慧校园服务平台应用开发分为前台用户手机 app 和后台 web 端管理系统两部分。

前台 app 目前支持 Android 5.0 以上系统,主要模块为地图信息载体、用户上传信息、

事件列表、数据可视化和个人中心五个模块。地图信息载体模块包括定位模块和地图显示所有用户所上传的信息。用户上传信息模块支持用户在当前定位点发布事件，经后台审核显示与地图上。数据可视化模块是由后台数据分析结果经可视化处理后显示给用户。事件列表里包括所有用户所发布的事件，根据事件紧急程度排序。个人中心模块包括用户注册登录和个人信息。

后台管理系统包括审核用户上传事件，是否需要如果介入处理，以及派出校方人员到达定位点处理事件。数据分析模块负责分析处理所有用户上传的信息，展示于用户 app 端的数据可视化模块。



图表 二-1 应用流程简览

2.2 总体设计

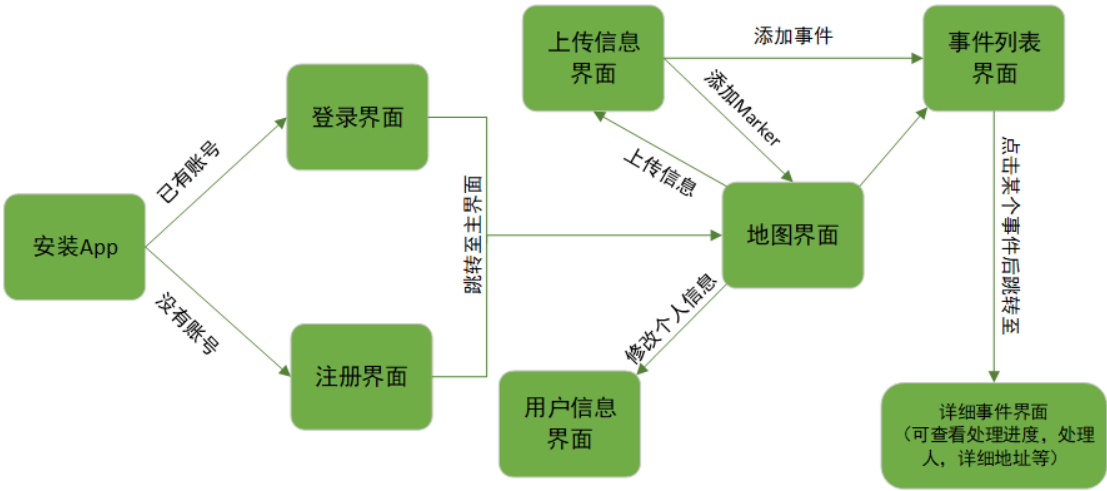
2.2.1 需求规定

详见需求分析

2.2.2 运行环境

软件环境：Android 5.0 及以上

2.2.3 软件结构



图表 二-2 软件结构

2.2.4 人工处理过程

由使用者上传问题，针对于实时定位、上传信息，管理控制、和后台处理这四个服务模块来划分功能，使用者根据输入信息，如果使用者上传信息经过后台审核符合要求，地图上会根据使用者所在位置添加标记点，如果需要进行后台工作人员处理，后台收到信息后派出校方相应人员前往处理。

2.3 接口设计

2.3.1 外部接口

高德地图 API

2.3.2 内部接口

- a. 上传信息与管理控制接口
- b. 管理控制与后台处理接口
- c. 数据返回和处理接口

2.4 运行设计

2.4.1 运行模块

实时定位、上传信息、管理控制、和后台处理四个服务模块。

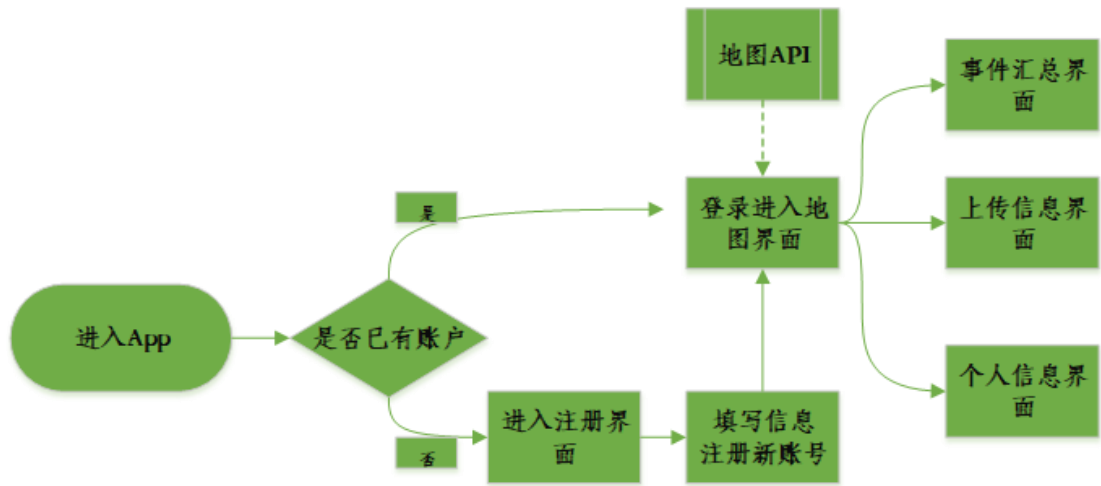
2.4.2 相互关联模块

地图信息载体模块加载位置信息，为使用者上传信息添加地理坐标，信息上传至管制控制模块进行后台审核，审核通过后进入后台处理模块，派出校方人员解决用户问题。

第三章 详细设计

3.1 App 界面设计

3.1.1 使用流程总概：



图表 三-1：流程总概图

如图 1，这是这个项目的整体流程图，全程 APP 界面主要分为三个部分，包括登陆注册界面、地图界面以及事件汇总界面，后台由大数据储存中心进行数据的存储以及加密处理等管理操作。根据用户是否拥有账号选择登录或注册进入 APP 主界面，然后会有地图显示用户实时地址；当用户发现问题后，可在主界面选择拍照并直接上传并表述事件具体问题，上传成功后，后台将会加密并储存；同时，上传的事件会直接在当地社区管理者的显示屏上显示，并进行解决。

3.1.2 登录注册界面：

如图 2，作为 APP 的初始界面，本界面采用居中对称的简约设计以保证对不同设备的

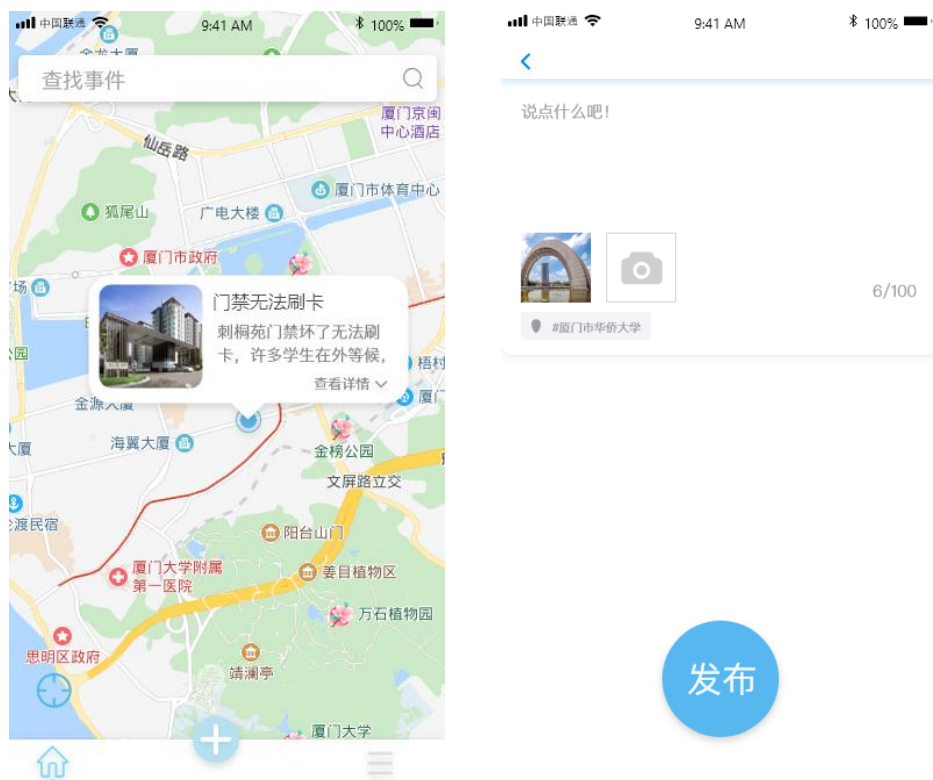
适配性，并给用户一个良好的交互体验。在这里用户可以输入个人的“用户账户”和“密码”，系统会自动确定用户的角色，并设置相应的权限。用户账户的安全性是首要考虑，在此基础上，为了减少无用重复操作，添加上自动登陆机制。



图表 三-2：登录注册界面

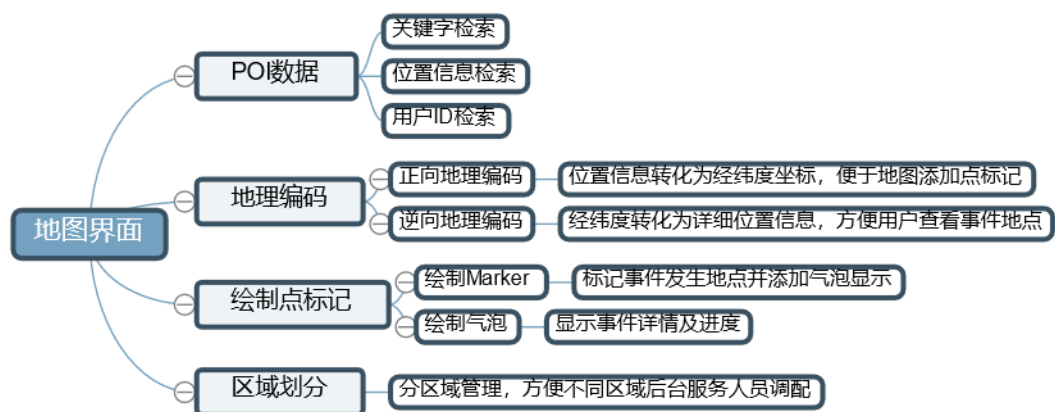
3.1.3 地图界面：

集成高德地图 API，如图表 3 显示当前区域地图通过实时定位方便用户获得当前位置并上传信息，通过绘制 **Marker** 显示当前地点用户上传的事件和事件处理进度。长按地图会加入新的标记点，显示当前标记点位置，同时也可以在此标记点添加事件详情。点击气泡可以进入详细信息界面，其他用户可以留言帮助事件发起人解决问题，也可以关注该事件，随时了解事件进展。



图表三-3：地图界面和信息上传界面

为了解决并完成以上目的，我们将本模块分为几个小模块阐述：



图表 三-4：地图界面技术路线

- **搜索模块：**

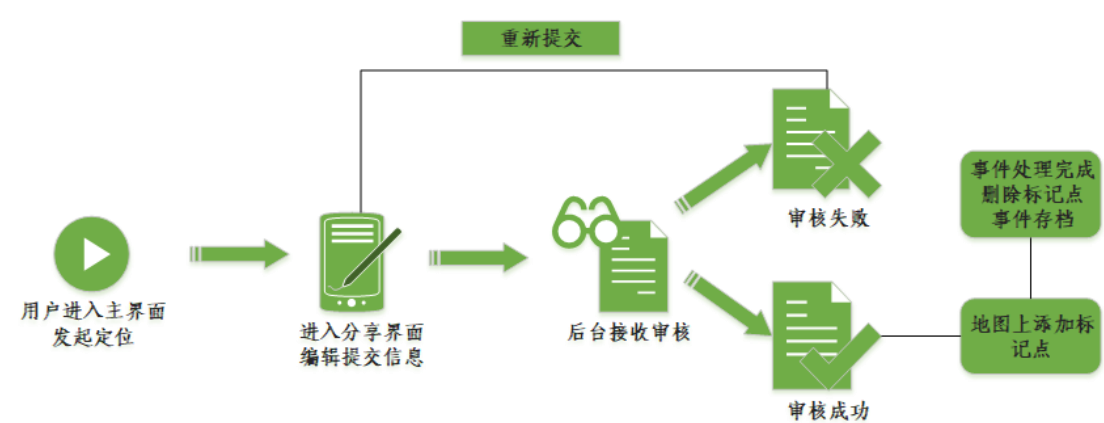
通过 POI 数据检索功能以及配合正向地理编码功能获取搜索事件、地点的位置信息。也可以通过用户 ID 搜索，查看该用户上传的事件。

- **区域划分模块：**

通过多边形区域划分，不同区域后台单独管理，只显示该区域内上报的事件，加快后台处理速度。方便社区、学校、公司等独立管理内部事件，提高效率防止区域之间混乱。

● **信息上传模块:**

用户自定义文字描述、加入照片和位置信息，通过连接数据库匹配用户信息并上传事件详情，方便后台服务人员审核发布，然后在地图界面调用并显示事件详情。



图表 三-5: 信息上传模块设计图

如图 5，用户首先进入主界面发起定位，定位完成后进入分享界面编辑提交信息，将信息上传至后台进行审核，审核若是失败则需重新编辑并提交信息，若是成功地图上则会自动添加标记点用来提供当前事件点信息的入口，点击便可查看用户上传的信息，事件处理完成后则会自动删除标记点事件存档。

3.1.4 事件汇总界面:

此模块界面为用户附近所有已提交事件的汇总，为了更直观的将事件呈现出来，分为描述主图片，事件标题，用户关注度三个部分，并将其列表化显示在界面，呈现的数据量较为庞大，为了保持数据加载的完整及快捷，我们决定使用 Android -Paging 库。



图表三-6：事件汇总界面

Paging 库中的几个类简化了请求数据的过程。这几个类还可以和组件化架构进行无缝对接。

- **将数据加载进内存：**

PagedList 类加载的数据来自于 DataSource。你可以配置一次加载多少数据，可以预先获取多少数据，尽量减少加载数据的时间。此类也可以向其他类提供数据更新的信号，比如：RecyclerView.Adapter 一样，为 RecyclerView 页允许数据加载。

- **将数据显示到 UI 上：**

PagedListAdapter 类实现自 RecyclerView.Adapter，并且从 PagedList 中加载数据。例如：当一个新的页面被加载 PagedListAdapter 就会发信号通知 RecyclerView 数据已经到了，然后 RecyclerView 就是显示和数据量相等的 item，并执行适当的动画。

PagedListAdapter 对于来自一个 PagedList 的下一条数据会在后台线程进行计算。（例如：将数据库的新数据更新到 PagedList 中），调用 notifyItem..()方法更新需要的数据列表的内容。然后 RecyclerView 再执行必要的更改。例如：item 的 position 的顺序的改变。

- **观察数据的更新:**

Paging 使用下列类来构造 PagedList 容器实时更新

LivePagedListBuilder 此类是从 DataSource.Factory 构建 LiveData<PagedList>。如果使用 Room 来管理数据库，Dao 可以生成 DataSource.Factory，使用 PositionalDataSource。

2.RXPagedListBuilder 此类支持 RxJava。

- **创建数据流:**

Paging 的数据流是在后台线程生产，在 UI 线程中显示。例如：当一条新的 item 插入到数据库中，DataSource 被初始化，LiveData<PagedList>或者 Flowable<PagedList>后台线程就会创建一个新的 PagedList。Paging 在后台线程中完成大部分工作，所以不会给主线程带来负担。

这个新建的 PagedList 会被发送到 UI 线程的 PagedListAdapter。PagedListAdapter 使用 DiffUtil 在对比现在的表单和新建表单的差异。当对比结束，PagedListAdapter 通过调用 RecyclerView.Adapter.notifyItemInserted()将新的 item 插入到适当的位置。RecyclerView 就会知道他需要绑定一个新的 item，并将其显示。

3.2 设计重难点

3.2.1 登录注册模块:

对于我们这一款有着社交需求的 App 来说是需要有账户体系的支持的，自然也就需要注册和登录。登陆注册的方式也有很多，例如也可以使用第三方账户如 QQ，微信等进行登陆，而不论是哪一种方式，对于一个没有安全保证的产品，用户的安全感就无从谈起，所以登陆注册过程应避免漏洞，尽可能保证用户账户的安全。

作为面向社区各用户的应用，我们首要考虑的就是用户密码信息的安全性。在本项目中我们采用的是 **MD5 加密算法**，其基本算法原理为：在一些初始化处理后，MD5 以 512 位分组来处理输入文本，每一分组又划分为 16 个 32 位子分组。算法的输出由四个 32 位分组组成，将它们级联形成一个 128 位散列值。

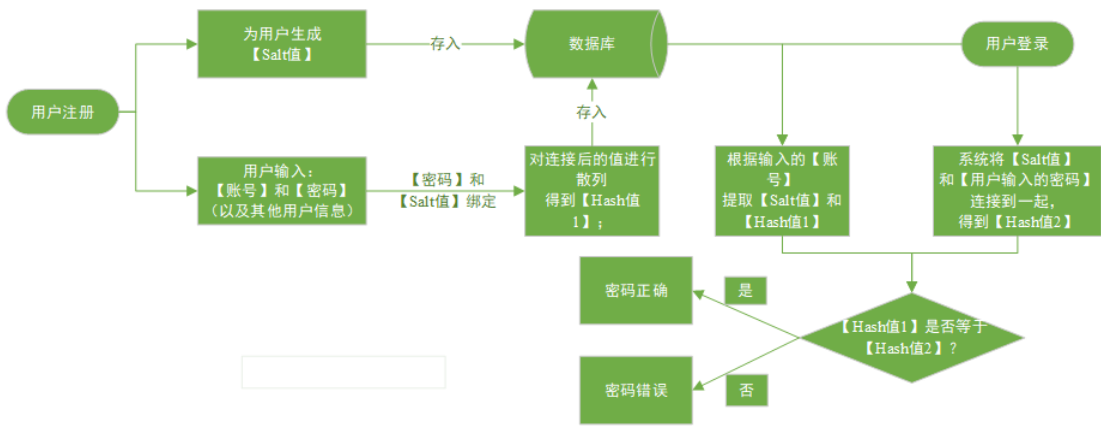
四个 32 位变量初始化为：

A=0x01234567 B=0x89abcdef C=0xfedcba98 D=0x76543210

它们称为**链接变量 (chaining variable)** 接着进行**算法的主循环**，循环的次数是消息中

512 位消息分组的数目。 主循环有四轮（MD4 只有三轮），每轮很相似。

加 Salt 可以一定程度上防止黑客破解密码。其基本想法是这样的：当用户首次提供密码时（通常是注册时），由系统自动往这个密码里 salt 值，然后再散列。而当用户登录时，系统为用户提供的代码加上同样的 salt 值，然后散列，再比较散列值，已确定密码是否正确。



图表 三-7：MD5（加 salt)原理图

3.2.2 地图模块:

本款基于地理功能的 app 需要用到地图功能，用户通过 app 所提交的并且审核通过的事件都会定位在地图界面上并显示出来，通过这种方式可以使得事件更加直观地反映给其他用户以及相关部门。这里采用的是高德地图 API，通过 POI 数据检索功能以及配合正向地理编码功能获取搜索事件、地点的位置信息。关键技术主要有以下几点：

a. 全局广播机制：Android 广播机制包含三个基本要素：**广播(Broadcast)** - 用于发送广播；**广播接收器(BroadcastReceiver)** - 用于接收广播；**意图内容(Intent)**-用于保存广播相关信息的媒介。**Broadcast** 是一种广泛运用的在应用程序之间传输信息的机制。而 BroadcastReceiver 是对发送出来的 Broadcast 进行过滤接受并响应的一类组件。

b. 数据持久化技术：**SharePr2eferences** 储存以及**数据库储存**，使用户数据储存更安全。SharedPreferences 是使用键值对的方式来存储数据的。也就是说当保存一条数据的时候，需要给这条数据提供一个对应的**键**，这样在读取数据的时候就可以通过这个键把相应的值取出来。而且 SharedPreferences 还支持多种不同的数据类型存储，如果存储的数据类

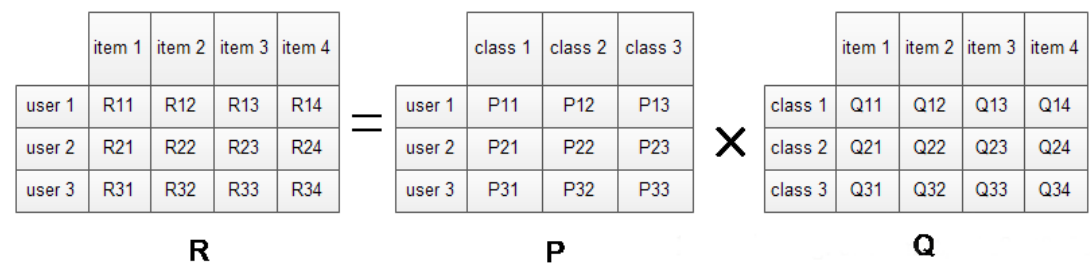
型是整型，那么读取出来的数据也是整型的，存储的数据是一个字符串，读取出来的数据仍然是字符串

3.2.3 事件推荐优先级模块：

此模块为事件汇总界面向用户推送事件优先程度的重要依赖，模块依据用户浏览历史及关注记录来按序汇总事件，可以很大程度上提升用户对该界面的关注度，算法复杂度较高，根据团队讨论，我们采用的是在推荐系统领域应用较为广泛的隐语义模型（LFM）。

LFM 核心思想在于通过隐含的特征联系用户兴趣和事件，即基于用户行为的统计数据来自动聚类，发现事件中隐含的类别，从而用于推荐。在推荐系统中它能够基于用户的行为对 item 进行自动聚类，也就是把 item 划分到不同类别或主题，而这些主题或类别可以理解为用户的兴趣。

LFM 从用户的行为数据出发，自动找到那些类然后进行个性化推荐。如下图所示：



图表三-8： user-item 矩阵

LFM 根据如下公式来计算用户 U 对物品 I 的兴趣度：

$$\text{Preference}(u,i) = r_{ui} + p_u^T q_i = \sum_{k=1}^k p_{u,k} q_{i,k}$$

优化损失函数可以定义如下：

$$\text{LossC} = \sum_{(u,i) \in R} (r_{ui} - \hat{r}_{ui})^2 + \lambda ||p_u||^2 + \lambda ||q_i||^2$$

第四章 测试报告

测试主要根据用户需求及软件需求以及相应的功能进行系统测试,包括功能测试、性能测试、网络测试、安全性和访问控制测试、用户界面测试以及兼容性测试等。

主要功能包括: 用户登录注册、个人信息、地图界面及信息提交、事件汇总。

4.1 测试类型

| 测试类型 | 测试内容 | 测试目的 | 所用测试工具及方法 |
|------|----------------------------|--|--------------|
| 功能测试 | 用户登录注册、个人信息、地图界面及信息提交、事件汇总 | 核实所有功能均已正常实现,即可按用户的需求使用软件: 1、 业务流程检验: 各个业务流程能够满足用户需求,用户使用不会产生疑问; 2、 数据准确: 显示的信息与用户提交信息一致,不会出现信息错误或漏发、重复发送。 | 黑盒测试 手工测试 |

| | | | |
|---------------|--|--|--------------|
| 性能测试 | 1、 注册、登录 响应速度 2、 各功能模块 切换响应速度 | 核实程序在多数 数据处理、手机多 程序运行的情况 下软件性能的稳 定性，不造成程 序崩溃或相关的 异常现象。 | 黑盒测试 手工测试 |
| 网络测试 | 1、模拟用户使用 网络环境，检 验程序在实际 网络环境中使 用情况及进行 业务操作。 2、程序在各运营 商及不同情况 下不同频段网 络下的响应情 况 | 核实程序在不同 网络情况下不同 功能界面的响应 情况，避免出现 在某种网络下程 序无法正常运行 的状况。 | 手工测试 |
| 安全及访问控制 测试 | 1、 注册、登 录、退出登录 2、 权限限制 3、 通过修改 URL 非法访问 4、 登录超时限 制 | 核实用户只能操 作其所拥有权限 能操作的功能。 | 黑盒测试 手工测试 |

| | | | |
|--------|--|--|--------------|
| 用户界面测试 | 1、 页面结构： 包括菜单、背景、颜色、字体、按钮、Title、提示信息的一致性等 2、 友好性、易用性、合理性、一致性、正确性 | 核实软件风格符合可接受标准，能够保证用户界面友好性、易操作性，符合用户操作习惯。 | 手工测试 |
| 兼容性测试 | 采用不同型号的安卓手机，以及不同的屏幕尺寸，不同版本Android | 核实软件在不同的系统中是否运行稳定。 | 黑盒测试 手工测试 |

4.2 测试环境及配置

| 资源名称/类型 | 配置 |
|----------|------------------------------------|
| 测试移动端 | CPU: Kirin 960 ROM:64GB RAM:6GB |
| 数据库管理系统 | MySQL, SQLite |
| 应用软件操作系统 | Android |

4.3 APP 测试点

4.3.1 安全测试

4.3.1.1 软件权限

1) 扣费风险：包括发送短信、拨打电话、连接网络等

- 2) 隐私泄露风险：包括访问手机信息、访问联系人信息等
- 3) 对 App 的输入有效性校验、认证、授权、敏感数据存储、数据加密等方面进行检测
- 4) 限制/允许使用手机功能接入互联网
- 5) 限制/允许使用手机发送接受信息功能
- 6) 限制/允许应用程序来注册自动启动应用程序
- 7) 限制或使用本地连接
- 8) 限制/允许使用手机拍照或录音
- 9) 限制/允许使用手机读取用户数据
- 10) 限制/允许使用手机写入用户数据
- 11) 检测 App 的用户授权级别、数据泄漏、非法授权访问等

4.3.1.2 安装与卸载安全性

- 1) 应用程序应能正确安装到设备驱动程序上
- 2) 能够在安装设备驱动程序上找到应用程序的相应图标
- 3) 是否包含数字签名信息
- 4) 安装路径应能指定
- 5) 没有用户的允许, 应用程序不能预先设定自动启动
- 6) 卸载是否安全, 其安装进去的文件是否全部卸载
- 7) 卸载用户使用过程中产生的文件是否有提示
- 8) 其修改的配置信息是否复原
- 9) 卸载是否影响其他软件的功能
- 10) 卸载应该移除所有的文件

4.3.1.3 数据安全性

- 1) 当将密码或其他敏感数据输入到应用程序时, 其不会被储存在设备中, 同时密码也不会被解码
- 2) 输入的密码将不以明文形式进行显示
- 3) 密码或其他敏感数据将不被储存在它们预输入的位置上
- 4) 当将敏感数据输入到应用程序时, 其不会被储存在设备中

- 5) 备份应该加密, 恢复数据应考虑恢复过程的异常通讯中断等, 数据恢复后再使用前应该经过校验
- 6) “取消”命令操作能够按照设计要求实现其功能
- 7) 应用程序读和写数据正确。
- 8) 应用程序应当有异常保护。
- 9) 如果数据库中重要的数据正要被重写, 应及时告知用户
- 10) 能合理地处理出现的错误
- 11) 意外情况下应提示用户

4.3.1.4 通讯安全性

- 1) 在运行其软件过程中, 如果有来电、、蓝牙、红外等通讯或充电时, 是否能暂停程序, 优先处理通信, 并在处理完毕后能正常恢复软件, 继续其原来的功能
- 2) 当创立连接时, 应用程序能够处理因为网络连接中断, 进而告诉用户连接中断的情况
- 3) 应能处理通讯延时或中断
- 4) 应用程序将保持工作到通讯超时, 进而发送给用户一个错误信息指示有连接错误
- 5) 应能处理网络异常和及时将异常情况通报用户
- 6) 应用程序关闭或网络连接不再使用时应及时关闭) 断开

4.3.1.5 人机接口安全性

- 1) 返回菜单总保持可用
- 2) 命令有优先权顺序
- 3) 声音的设置不影响应用程序的功能
- 4) 应用程序必需利用目标设备适用的全屏尺寸来显示上述内容
- 5) 应用程序必需能够处理不可预知的用户操作, 例如错误的操作和同时按下多个键

4.3.2 安装、卸载测试

4.3.2.1 安装

- 1) 软件在不同操作系统下安装是否正常。

- 2) 软件安装后的是否能够正常运行，安装后的文件夹及文件是否写到了指定的目录里。
- 3) 软件安装各个选项的组合是否符合概要设计说明
- 4) 软件安装向导的 UI 测试
- 5) 软件安装过程是否可以取消，点击取消后，写入的文件是否如概要设计说明处理
- 6) 软件安装过程中意外情况的处理是否符合需求（如死机，重启，断电）
- 7) 安装空间不足时是否有相应提示
- 8) 安装后没有生成多余的目录结构和文件
- 9) 对于需要通过网络验证之类的安装，在断网情况下尝试一下
- 10) 还需要对安装手册进行测试，依照安装手册是否能顺利安装

4.3.2.2 卸载

- 1) 直接删除安装文件夹卸载是否有提示信息。
- 2) 测试系统直接卸载程序是否有提示信息。
- 3) 测试卸载后文件是否全部删除所有的安装文件夹。
- 4) 卸载过程中出现的意外情况的测试（如死机、断电、重启）。
- 5) 卸载是否支持取消功能，单击取消后软件卸载的情况。
- 6) 系统直接卸载 UI 测试，是否有卸载状态进度条提示。

4.3.3 功能测试

4.3.3.1 运行

- 1) App 安装完成后的试运行，可正常打开软件。
- 2) App 打开测试，是否有加载状态进度提示。
- 3) App 打开速度测试，速度是否可观。
- 4) App 页面间的切换是否流畅，逻辑是否正确
- 5) 注册
 - 同表单编辑页面
 - 用户名密码长度
 - 注册后的提示页面

--前台注册页面和后台的管理页面数据是否一致

--注册后，在后台管理中页面提示

6) 登录

--使用合法的用户登录系统。

--系统是否允许多次非法的登陆，是否有次数限制。

--使用已经登陆的账号登陆系统是否正确处理。

--使用禁用的账号登陆系统是否正确处理。

--用户名、口令（密码）错误或漏填时能否登陆。

--删除或修改后的用户，原用户登陆。

--不输入用户口令和用户、重复点（确定或取消按钮）是否允许登陆。

--登陆后，页面中登陆信息。

--页面中有注销按钮。

--登陆超时的处理。

7) 注销

--注销原模块，新的模块系统能否正确处理。

--终止注销能否返回原模块，原用户。

--注销原用户，新用户系统能否正确处理。

--使用错误的账号、口令、无权限的被禁用的账号进行注销

8) 地图信息提交是否正确显示，地图中信息框是否正确展示，并能实现相应点击事件

9) 个人资料是否能修改并正确显示，出现信息冲突能否正常交互

10) 信息汇总是否能够正确显示事件并实时更新

4.3.3.2 应用的前后台切换

1) APP 切换到后台，再回到 app，检查是否停留在上一次操作界面。

2) APP 切换到后台，再回到 app，检查功能及应用状态是否正常。

3) app 切换到后台，再回到前台时，注意程序是否崩溃，功能状态是否正常，尤其是对于从后台切换回前台数据有自动更新的时候。

4) 手机锁屏解屏后进入 app 注意是否会崩溃，功能状态是否正常，尤其是对于从后台切换回前台数据有自动更新的时候。

- 5) 当 App 使用过程中有电话进来中断后再切换到 app，功能状态是否正常。
- 6) 当杀掉 app 进程后，再开启 app，app 能否正常启动。
- 7) 出现必须处理的提示框后，切换到后台，再切换回来，检查提示框是否还存在，有时候会出现应用自动跳过提示框的缺陷。
- 8) 对于有数据交换的页面，每个页面都必需要进行前后台切换、锁屏的测试，这种页面最容易出现崩溃。

4.3.3.3 免登录

- 1) app 有免登录功能时，需要考虑 Android 版本差异。
- 2) 考虑无网络情况时能否正常进入免登录状态。
- 3) 切换用户登录后，要校验用户登录信息及数据内容是否相应更新，确保原用户退出。
- 4) 一个账户只允许登录一台机器，所以，需要检查一个账户登录多台手机的情况。原手机里的用户需要被踢出，给出友好提示。
- 5) app 切换到后台，再切回前台的校验
- 6) 切换到后台，再切换回前台的测试
- 7) 密码更换后，检查有数据交换时是否进行了有效身份的校验
- 8) 支持自动登录的应用在进行数据交换时，检查系统是否能自动登录成功并且数据操作无误。
- 9) 检查用户主动退出登录后，下次启动 app，应停留在登录界面

4.3.3.4 数据更新

- 1) 需要确定哪些地方需要提供手动刷新，哪些地方需要自动刷新，哪些地方需要手动+自动刷新。
- 2) 确定哪些地方从后台切换回前台时需要进行数据更新。
- 3) 根据业务、速度及流量的合理分配，确定哪些内容需要实时更新，哪些需要定时更新。
- 4) 确定数据展示部分的处理逻辑，是每次从服务端请求，还是有缓存到本地，这样才能有针对性的进行相应测试。
- 5) 检查有数据交换的地方，均有相应的异常处理。

4.3.3.5 离线浏览

- 1) 在无网络情况可以浏览本地数据
- 2) 退出 app 再开启 app 时能正常浏览
- 3) 切换到后台再切回前台可以正常浏览
- 4) 锁屏后再解屏回到应用前台可以正常浏览
- 5) 在对服务端的数据有更新时会给予离线的相应提示

4.3.3.6 App 更新

- 1) 当客户端有新版本时，有更新提示。
- 2) 当版本为非强制升级版时，用户可以取消更新，老版本能正常使用。用户在下次启动 app 时，仍能出现更新提示。
- 3) 当版本为强制升级版时，当给出强制更新后用户没有做更新时，退出客户端。下次启动 app 时，仍出现强制升级提示。
- 4) 当客户端有新版本时，在本地不删除客户端的情况下，直接更新检查是否能正常更新。
- 5) 当客户端有新版本时，在本地不删除客户端的情况下，检查更新后的客户端功能是否是新版本。
- 6) 当客户端有新版本时，在本地不删除客户端的情况下，检查资源同名文件如图片是否能正常更新成最新版本。如果以上无法更新成功的，也都属于缺陷。

4.3.3.7 定位、照相机服务

- 1) App 有用到相机，定位服务时，需要注意系统版本差异
- 2) 有用到定位服务、照相机服务的地方，需要进行前后台的切换测试，检查应用是否正常。
- 3) 当定位服务没有开启时，使用定位服务，会友好性弹出是否允许设置定位提示。当确定允许开启定位时，能自动跳转到定位设置中开启定位服务。

4.3.4 性能测试

- 1) 极限测试：在各种边界压力情况下，如电池、存储、网速等，验证 App 是否能正确响应。
—内存满时安装 App

--运行 App 时手机断电

--运行 App 时断掉网络

2) 响应能力测试：测试 App 中的各类操作是否满足用户响应时间要求。

--App 安装、卸载的响应时间

--App 各类功能性操作的影响时间

3) 压力测试：反复/长期操作下、系统资源是否占用异常。

--App 反复进行安装卸载，查看系统资源是否正常

--其他功能反复进行操作，查看系统资源是否正常

4) 性能评估：评估典型用户应用场景下，系统资源的使用情况。

4.3.5 交叉事件测试

1) 多个 App 同时运行是否影响正常功能

2) App 运行时前/后台切换是否影响正常功能

3) App 运行时拨打/接听电话

4) App 运行时发送/接收信息

5) App 运行时发送/收取邮件

6) App 运行时切换网络

7) App 运行时浏览网络

8) App 运行时使用蓝牙传送/接收数据

9) App 运行时使用相机、计算器等手机自带设备

4.3.6 兼容测试

1) 与本地及主流 App 是否兼容

2) 基于开发环境和生产环境的不同，检验在各种网络连接下 App 的数据和运用是否正确

3) 与各种设备是否兼容，若有跨系统支持则需要检验是否在各系统下，各种行为是否一致

--不同操作系统的兼容性，是否适配

--不同手机屏幕分辨率的兼容性

—不同手机品牌的兼容性

4.3.7 硬件环境测试

4.3.7.1 手势操作测试

- 1) 手机开锁屏对运行中的 App 的影响
- 2) 切换网络对运行中的 App 的影响
- 3) 运行中的 App 前后台切换的影响
- 4) 多个运行中的 App 的切换
- 5) App 运行时关机
- 6) App 运行时重启系统
- 7) App 运行时充电
- 8) App 运行时 kill 掉进程再打开

4.3.7.2 网络环境

- 1) 无网络时，执行需要网络的操作，给予友好提示，确保程序不出现 crash。
- 2) 内网测试时，要注意选择到外网操作时的异常情况处理。
- 3) 在网络信号不好时，检查功能状态是否正常，确保不因提交数据失败而造成 crash。
- 4) 在网络信号不好时，检查数据是否会一直处于提交中的状态，有无超时限制。如遇数据交换失败时要给予提示。
- 5) 在网络信号不好时，执行操作后，在回调没有完成的情况下，退出本页面或者执行其他操作的情况，有无异常情况。此问题也会经常出现程序 crash。

4.4 测试结果及解决

经过以上项目的多次测试，最终发现，除已通过测试的项目外，还出现了以下问题，之后，我们采取有效措施将问题解绝：

| 存在问题 | 解决情况 |
|--------------------|------------------|
| 1. 使用相同手机号注册未出现错误提 | 修改相应逻辑代码，经调试后正常使 |

| | |
|------------------------------------|------------------------------------|
| 示 | 用 |
| 2. 地图定位功能出现较明显偏差 | |
| 3. 当地图中信息框较多时，加载时间大幅加长，有时信息框不能主动显示 | |
| 4. 修改个人资料绑定手机一栏，相应信息并未正确修改 | |
| 5. 信息汇总未能按优先算法正确排序事件 | |
| 6. 某些信息输入功能未有错误操作提醒 | 修改逻辑代码及相应布局文件，经调试后正常使用 |
| 7. app 各界面切换及整体加载时间相对偏慢 | 减少 Activity 使用数量，优化整体加载结构，经调试后正常使用 |

经过多方面修改及调试，我们的 App 在大多数情况下都能正常使用，另外我们的 App 还需要拥有更丰富的交互以及对大量数据处理的能力，以方便并优化用户的体验感。

第五章 安装及使用

5.1 安装环境要求

| | |
|-----|-----------------|
| 系统 | Android 5.0 或以上 |
| RAM | 128MB |
| ROM | 20MB |
| 权限 | 网络，GPS |

5.2 安装过程

打开安装包，点击安装进行安装，安装完成后打开设置，进入应用权限，将定位设置为开启，打开手机 GPS 即可使用该应用。

5.3 使用流程

5.3.1 注册:

- 1、打开 app 进入到登录界面（图一）
- 2、点击立即注册，进入注册界面（图二），输入合法用户名、密码、邮箱、手机号进行注册。注册成功后自动返回登录界面。



（图一）



（图二）

5.3.2 登录:

输入已有的用户名和密码点击登录即可，登陆成功后跳转到主界面（图三）



（图三）

5.3.3 个人信息：

- 1、在主界面时，向右滑动屏幕即可实现“主界面”→“个人中心”（图四）的跳转
- 2、点击“编辑资料”进行个人信息的修改
- 3、点击“浏览记录”可查看过去浏览的事件
- 4、点击安全出行，填写出发地和目的地即可根据路况规划路径。
- 5、设置中心可以对 app 的 ui 配色等进行更改。



（图四）

5.3.4 查看事件：

- 1、在主界面地图上能够显示附近发生的事件
- 2、点击主界面右下角图标即可切换视图，进入事件事件查询界面（图五）
- 3、点击想查看的事件即可查看事件的详细情况以及处理进度（图六）



(图五)



(图六)

5.3.5 发布消息:

- 1、进入主界面（图三）或者事件查询界面（图四）
- 2、点击屏幕最下方的加号即可进入上传界面（图七）
- 3、点击文本框→编辑文本；点击相机图标→拍照上传图片或录制上传小视屏；点击图片下方定位可以修改当前位置
- 4、点击发布，在联网状态下即可发布消息，在主界面，或者个人中心可以查看发布的事件详情以及处理进度



发布

(图七)

第六章 项目总结

6.1 前期策划

在前期策划中，我们分为实地走访研究、方案制定、雏形开发三个阶段

6.1.1 实地走访研究

在信息时代的大背景下，智慧校园这一概念早已不再新鲜，但通过走访校内宿舍楼区、教学区以及休闲娱乐场所发现，现在大部分地方并没有有效充分的利用互联网来将日常联系起来，并且校团委、学生组织等开展行动时并没有进行紧密的联系和沟通，学生教师以及校外人员遇到疑难事情，各部门或互相推诿，无法及时解决。因此，智慧校园显得有些名存实亡。于是，在前期策划中，我们着重于如何使校园变得智能起来，以及如何使互联网技术也能与日常小事有紧密联系。

最后，我们在走访校园的师生所了解到，因为现在校园流动人口多，许多为外来人员，校园处理事务繁琐，许多地方出现的问题无法一一详细的了解到，同时，现在快节奏的生活，大多数人也没有时间专门跑去校内组织部门反映，所以问题一直无法及时解决。所以，师生本身的诉求是能用一种方便快捷的方式直观的反应发现的问题，并加以解决。

这些前期的走访调查研究了解到广大师生最强烈直观的诉求，对我们后续作品开发的灵感上给予许多的帮助。

6.1.2 方案制定

我们通过高德地图得到灵感，我们设想通过手机的定位功能，每一个人为校园神经网络的节点，建立一个校园的基于地图的问题发布，处理，反馈，信息共享以及需求数据采集和分析的平台。旨在切实快速解决校园生活中的实际问题，实现信息共享，挖掘校园内存在的需求，为智慧校园建设提供数据和决策支持。

以下对方案的技术进行说明：

(1) 实时地图定位技术

基于校园范围大，每个区域点的管理人手少，故我们将传统的手机地图功能与校园事件反映相结合，因此我们通过对高德地图进行了研究与分析：

- 1) 高德地图以手机定位，以 2D、3D、卫星等多种图形构建数据可视化平台
- 2) 用隐语义模型（LMF）自动分析用户的兴趣点与关注点
- 3) 统计用户数据并推荐有效事件
- 4) 以 Android Paging Library 库高效快速的提取用户所需信息

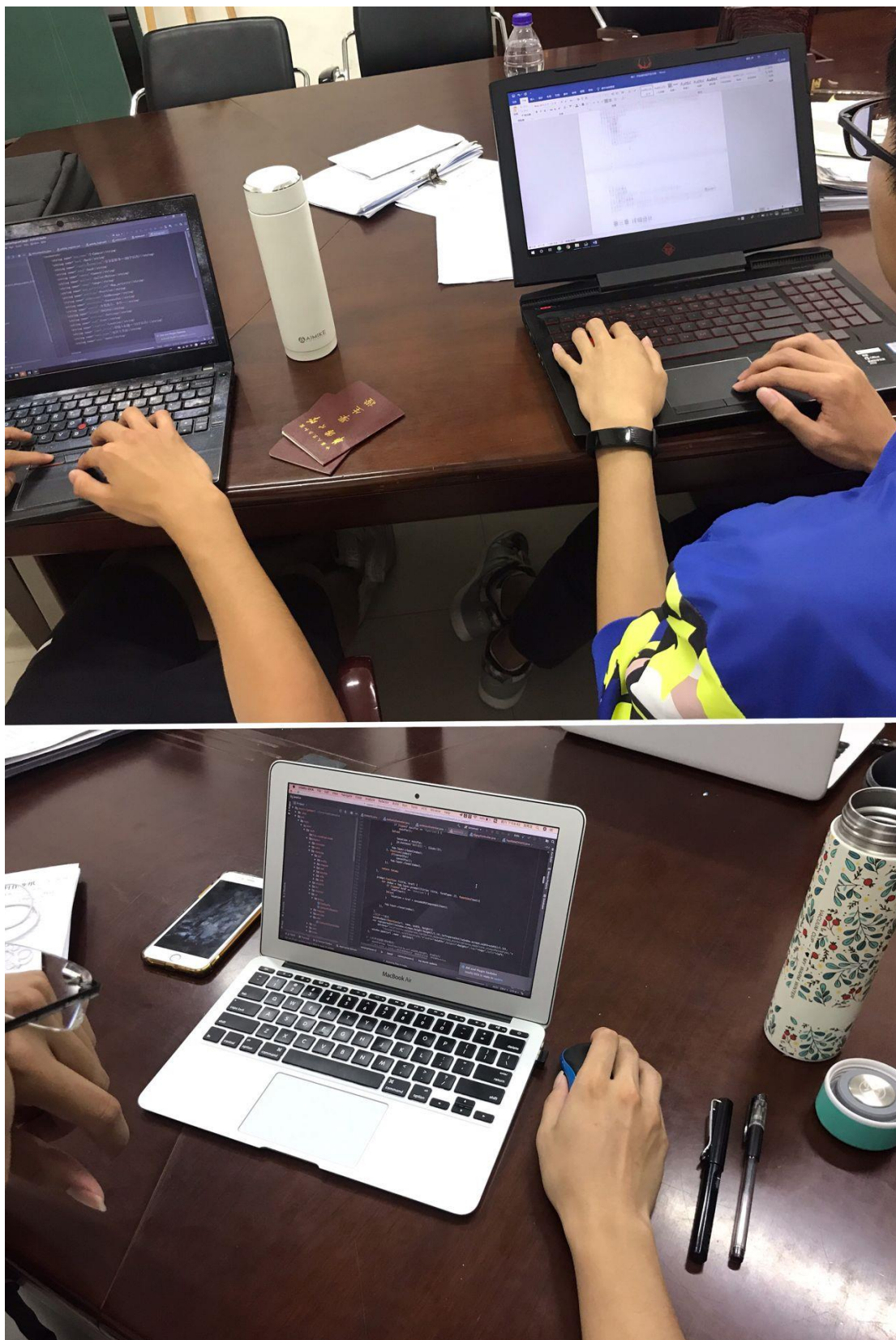
基于地图的定位功能，团队发现，我们可以利用定位和事件发生的地点结合，通过用户拍照实时上传将事件直接发布，app 后台可直接接收到事件发生的地点、详情。



图表 六-1 地图显示界面

6.1.3 雏形开发

1) 前期设计



图表 六-2 团队策划讨论

如图六-2 所示：在前期方案制定环节中，我们确定了方案便开始 APP 的雏形设计了，我们通过对 APP 的功能的分析以及对多个 APP 的参考和总结，我们初步确定了七个界面，搭建出

APP 的雏形。

2) 中期制作

在确定了 APP 的功能以及界面之后，我们在中期就开始着手具体的 APP 开发，目标在做出一个 demo 出来，以下是对 demo 的制作重点进行阐述：

- a. 需要与使用者有良好的交互性，各种控件要清晰可见，并且对操作要有明显的阐述与限定，如密码长度要求，字数限制等。
- b. UI 力求简洁，给使用者一个简洁明了的使用体验。
- c. 各界面切换转场要有一个合适的过度动画或其它效果，动画或效果自然有序。

3) 后期完善

在 APP 制作完成后，我们将把平台安装程序（apk 文件）导入 Android 手机中，并进行多次测试，并且会将测试完成的程序在 Android 的多个版本中进行测试运行。如果发现问题，将会及时返回到安卓安开发软件（Android studio）中修改，直到应用程序可以在多个 Android 手机上以及多数个 Android 版本中运行。