

Software Design Document

1. Introduction

Go & Show is a traveling application, aiming at helping visitors reduce their waiting time in line, avoid crowded attractions in the park and display visitor flow data on a map. In this program, our team choose Guangzhou Panyu, ChimeLong Paradise as the senario. Using the real attraction, as the data to develop our application.

2. Requirements Specification

Functional Requirements:

1. User Registration: The application should allow users to register with their email, phone number or other social media accounts.
2. Attractions Information: The application should provide information about the attractions in the park including introduction, location, opening hours, ticket prices and other details.
3. Reviews and Ratings: The application should allow users to post pictures, reviews, ratings, and other feedback about the attractions.
4. Animation Display: The application should use animations to vividly show the attractions.
5. Heap Map: The application should be able to display visitor flow data on a map and provide recommendations to users based on crowd location in the park.

Non-Functional Requirements:

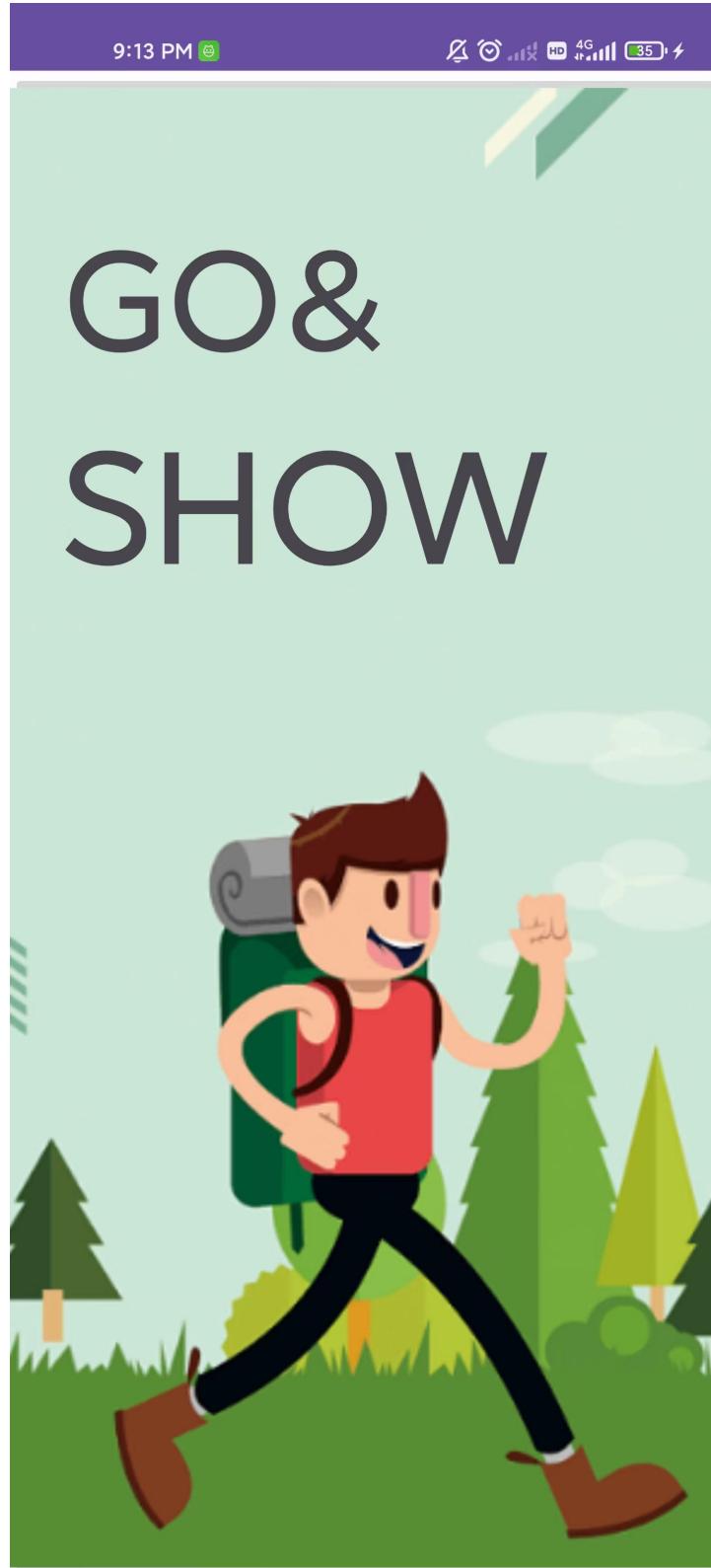
1. Usability: The application should be easy to use and navigate, with a user-friendly interface.
2. Performance: The application should be responsive and fast, with quick loading times and minimal lag.
3. Security: The application should be secure, protecting user data and preventing unauthorized access.
4. Compatibility: The application should be compatible with a wide range of devices and operating systems.

3. Overall Design

- It provides details about the attractions such as introduction, location, opening hours, ticket prices, etc. Users can post emojis, reviews, etc. to the Application. Quality photos and transcripts will be selected and compiled into an album.
- Animations will be used to vividly show the attractions. The Application will have marks for tourists to grow grass, and the higher the recommendation index, the more marks for planting.
- The Heap Map can figure out the crowd location in the park.

4. User Interface Design

After entering our application, a cover of our application logo will appear about 3 seconds. user can skip it by click the screen.



Then you will see the login page. Where you can sign in your own account here. If you never sign up before create your account by clicking "sign up for free" button

9:13 PM



Welcome back

Please enter your details

Email

Password

[Forgot password?](#)

[Login](#)

Don't have an account?

[Sign Up For Free!](#)

9:46 PM



Hello there!

Please enter your details

Username

Enter your username

Email

Enter your email

Password

Enter your password

Confirm Password

Confirm Password

Sign Up Now!

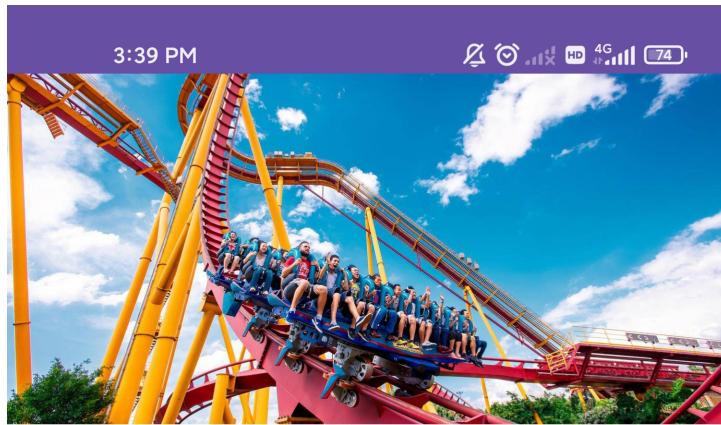
Do have an account?

Login Here

After logging in you will see our main page. In order to let our user easy to use our application, we simplified our main page as much as we can. There are a search bar and setting button on the top of the main page. Additionally, a heap map switch is shown at the bottom of screen, you can open the heap map just switch the button on. User can zoom in or zoom out the map as they want.



If you click the red markers on the map, you can see the detail information of the attraction. User can leave their comments or emojis under here.



垂直过山车

设计灵感和理念来自人类最初的飞翔梦想，它模拟了空中之王——雄鹰的各种空中绝技。最高落差达60米，相当于20层楼的落差，为亚洲垂直下落距离第二高的垂直下坠过山车，轨道全长981米，是世界上最长的垂直下坠过山车，拥有近90度垂直俯冲和冲浪体验，最高时速达112公里/小时，逍遥搏击任飞翔，彩虹绘出真体验，就让我们一起尽情去体验过山车带来的巅峰感受吧。

null: 还没做过好怕

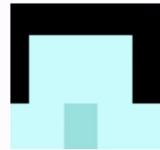
null: 😅 😊

Enter your comment

Submit

In the setting page, user can change there profile portrait just by clicking the portrait on the top. We also set the navigation button like "My profile", "setting", "logout" on this page.

9:14 PM



null

100M Followers

GO

&

SHOW

My Profile

Settings

Logout

5. Key Technology

5.1 Primary Technology

5.1.1 Baidu Map SDK Configuration

In order to use some function in Map we tried several Map SDK including(Amap, Google map, Baidu map). Finally we choose Baidu map as our SDK.



5.1.2 Database Establishment

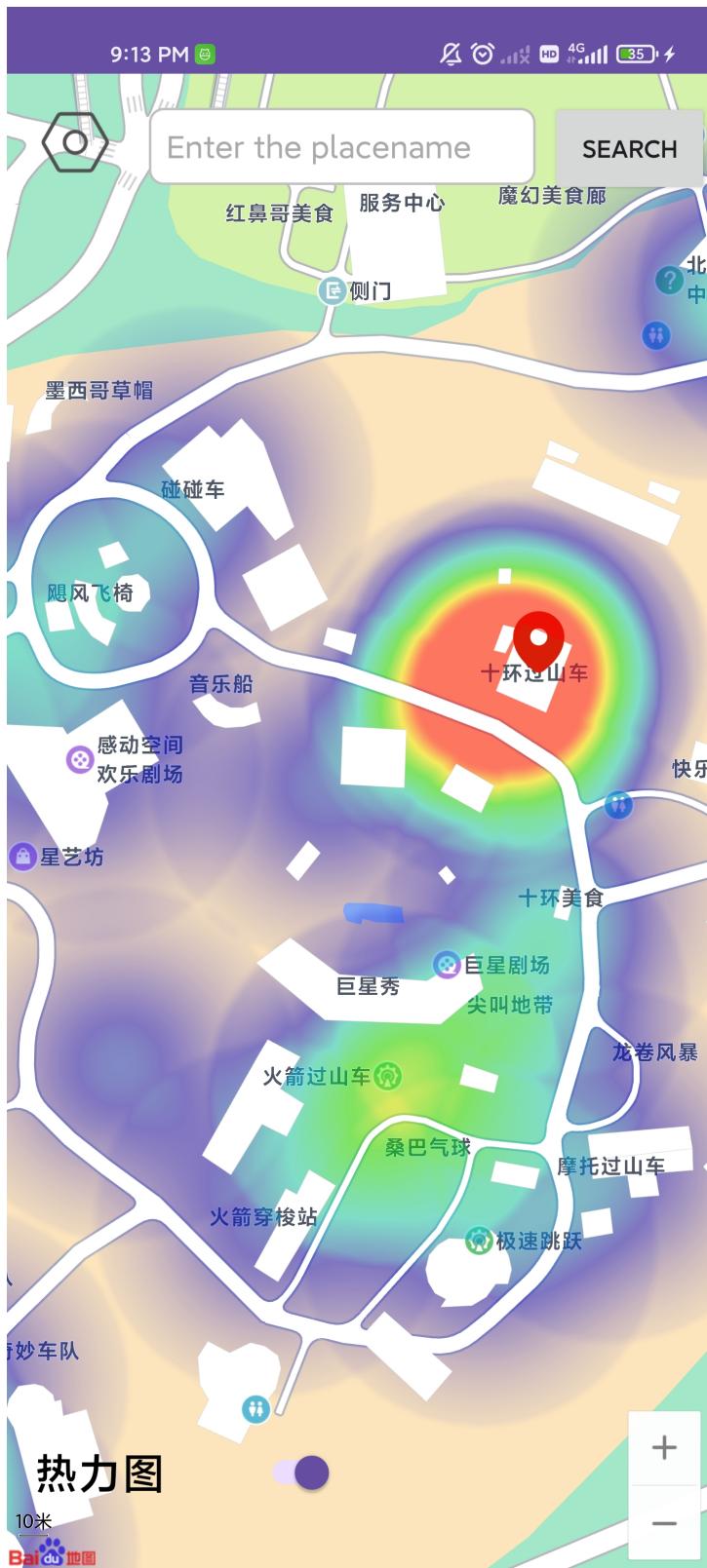
We create a data base to store the data of users.

```
1 class MyDatabaseHelper(val context: Context, name: String, version: Int) :  
2     SQLiteOpenHelper(context, "userInfo", null, 1) {  
3  
4     private val craeteUserAccount = "CREATE TABLE user_account (" +  
5         "id INTEGER PRIMARY KEY AUTOINCREMENT," +  
6         "user_name TEXT," +  
7         "user_email TEXT," +  
8         "user_password TEXT" +  
9         ")"  
10  
11    override fun onCreate(db: SQLiteDatabase) {  
12        db.execSQL(craeteUserAccount)
```

```
13     }
14
15     override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion:
16     Int) {
16         // 在此方法中处理数据库升级逻辑
17         db.execSQL("drop table if exists user_account")
18         onCreate(db)
19     }
20 }
21
```

5.1.4 Heap map funciton

Android 7.5.2 Map SDK supports dynamic heat map function, support to represent the density and distribution of data. You can see the flow of people through the heat map.



```
1 fun setHeatmapProperties(locations: List<LatLang>): HeatMap {  
2     // 获取多帧热力图数据集  
3     val datas = listOf(locations)  
4  
5     // 设置开始动画属性: 开启初始动画, 时长100毫秒, 动画缓动函数类型为线性  
6     val initAnimation = HeatMapAnimation(true, 100,  
7         HeatMapAnimation.AnimationType.Linear)
```

```
8     // 设置帧动画属性: 开启帧动画, 时长800毫秒, 动画缓动函数类型为线性
9     val frameAnimation = HeatMapAnimation(true, 800,
HeatMapAnimation.AnimationType.Linear)
10
11    // 设置热力图渐变色用到的所有颜色数组
12    val colors = intArrayOf(
13        color.rgb(0, 0, 200),           // 红色
14        color.rgb(0, 225, 0),          // 绿色
15        color.rgb(255, 0, 0)          // 蓝色
16    )
17
18    // 设置热力图渐变类
19    val gradient = Gradient(colors, floatArrayOf(0.3f, 0.7f, 1f))
20
21    // 创建热力图对象
22    val heatmap = HeatMap.Builder()
23        .datas(datas)
24        .initAnimation(initAnimation)
25        .frameAnimation(frameAnimation)
26        .gradient(gradient)
27        .maxIntensity(3.1f)
28        .minIntensity(-0.1f)
29        .radius(200)
30        .opacity(0.7)
31        .build()
32
33    return heatmap
34 fun hideHeatmap(locations: List<LatLng>): HeatMap {
35    // 获取多帧热力图数据集
36    val datas = listOf(locations)
37
38    // 设置开始动画属性: 开启初始动画, 时长100毫秒, 动画缓动函数类型为线性
39    val initAnimation = HeatMapAnimation(true, 100,
HeatMapAnimation.AnimationType.Linear)
40
41    // 设置帧动画属性: 开启帧动画, 时长800毫秒, 动画缓动函数类型为线性
42    val frameAnimation = HeatMapAnimation(true, 800,
HeatMapAnimation.AnimationType.Linear)
43
44    // 设置热力图渐变色用到的所有颜色数组
45    val colors = intArrayOf(
46        color.rgb(0, 0, 200),           // 红色
47        color.rgb(0, 225, 0),          // 绿色
48        color.rgb(255, 0, 0)          // 蓝色
49    )
50
51    // 设置热力图渐变类
52    val gradient = Gradient(colors, floatArrayOf(0.3f, 0.7f, 1f))
53
54    // 创建热力图对象
55    val heatmap = HeatMap.Builder()
56        .datas(datas)
```

```
57     .initAnimation(initAnimation)
58     .frameAnimation(frameAnimation)
59     .gradient(gradient)
60     .maxIntensity(3.1f)
61     .minIntensity(-0.1f)
62     .radius(200)
63     .opacity(0.0)
64     .build()
65
66     return heatmap
67 }
68
69 // 添加热力图到地图
70 fun addHeatmapToMap(baiduMap: BaiduMap, heatmap: HeatMap) {
71     baiduMap.addHeatMap(heatmap)
72 }
73
74 // 开启热力图帧动画
75 fun startHeatmapFrameAnimation(baiduMap: BaiduMap) {
76     baiduMap.startHeatMapFrameAnimation()
77 }
78
79 // 停止热力图帧动画
80 fun stopHeatmapFrameAnimation(baiduMap: BaiduMap) {
81     baiduMap.stopHeatMapFrameAnimation()
82 }
83
84 // 设置热力图帧回调
85 fun setHeatmapFrameCallback(baiduMap: BaiduMap, callback:
BaiduMap.OnHeatMapDrawFrameCallBack) {
86     baiduMap.setOnHeatMapDrawFrameCallBack(callback)
87 }
```

5.1.5 Search bar POI function

By typing the place you wanna search in the search box, and click the search button. The map will locate and zoom into the place you want to go. Here we put "大喇叭餐厅" as the default location.



```
1  @Override  
2  public void onGetPoiResult(PoiResult poiResult) {  
3      if (poiResult.error == SearchResult.ERRORNO.NO_ERROR) {  
4          mBaiduMap.clear();  
5  
6          //创建PoiOverlay对象  
7          PoiOverlay poiOverlay = new PoiOverlay(mBaiduMap);  
8      }
```

```

9     //设置Poi检索数据
10    poiOverlay.setData(poiResult);
11
12    //将poiOverlay添加至地图并缩放至合适级别
13    poiOverlay.addToMap();
14    poiOverlay.zoomToSpan();
15 }
16 }

```

5.2 Technical Challenge

5.2.1 Develop References

Some official developer document on Baidu web are out of date. Some functions or methods have been discarded. So we have to search the reference from Website such as CSDN.

5.2.2 SDK configuration problem

It really takes a long time to figure out the map SDK configuration on android studio. Because this is our first time to develop by using third SDK.

5.2.3 Shown Layer problem

Because of using third SDK, we have to spend a substantial of time figuring out the relation ship of each layer, in order to show all the buttons or correctly.

元素压盖顺序

所有叠加或覆盖到地图的内容，我们统称为地图覆盖物。如标注、矢量图形元素（包括：折线、多边形和圆等）、定位图标等。覆盖物有自己的地理坐标，当您拖动或缩放地图时，它们会相应的移动。

百度地图SDK为广大开发者提供的基础地图和上面的各种覆盖物元素，具有一定 的层级压盖关系，具体如下（从上至下的顺序）：

- 1.自定义View (MapView.addView(View));
- 2.弹出窗图层 (InfoWindow)；
- 3.定位图层 (BaiduMap.setMyLocationEnabled(true));
- 4.指南针图层（当地图发生旋转和视角变化时，默认出现在左上角的指南针）；
- 5.标注图层（Marker），文字绘制图层（Text）；
- 6.几何图形图层（点、折线、弧线、圆、多边形）；
- 7.底图标注（指的是基底图上自带的那些POI元素）；
- 8.百度城市热力图 (BaiduMap.setBaiduHeatMapEnabled(true));
- 9.实时路况图图层 (BaiduMap.setTrafficEnabled(true));
- 10.热力图图层 (HeatMap);
- 11.地形图图层 (GroundOverlay)；
- 12.瓦片图层 (TileOverlay)；
- 13.基础底图（包括底图、底图道路、卫星图、室内图等）；

6. Testing and User Experience Analysis

6.1 Testing

We run our application on the third party mobile app usability testing platform, 7 out of 10 mobile phones run successfully.

The test platform we used is wetest: <https://console.wetest.net/app/testlab/automation/test>

The screenshot shows a web-based test analysis interface. On the left, there's a sidebar with 'Automation' at the top, followed by 'Create Test' and 'Test Analysis'. The main area has tabs for 'Device' and 'Performance', with 'Device' selected. A 'Filter' dropdown is set to 'Select...'. Below is a table of test results:

Device ID	Device Status	Device Result	Brand	Category
1461	✓ Complete	Passed	SAMSUNG	C
1477	✓ Complete	Passed	SAMSUNG	C
2543	✓ Complete	Failed	VIVO	F
2736	✓ Complete	Failed	OPPO	C
5482	✓ Complete	Failed	HUAWEI	C
5485	✓ Complete	Passed	XIAOMI	R
5603	✓ Complete	Passed	Google	R
5634	✓ Complete	Passed	XIAOMI	R
5655	✓ Complete	Passed	SAMSUNG	C
5658	✓ Complete	Passed	Xiaomi	F

Show per page. Total: 10

6.2 User Experience

We found 10 classmates to use our app and record their feedback and suggestions after the user experience. Here are some received positive feedback from them. For example, some students reported that it is very convenient to use our app when visiting scenic spots, and there is no need to run around looking for amusement facilities with few people. Also, we get some suggestions about

improving our app and we will add new features to enhance the user experience further.



7. Conclusion

In conclusion, our app GO&SHOW is designed to simplify your daily routine and provide you with a seamless experience of traveling and sharing. We have put in a lot of effort and dedication to develop an app that is user-friendly and intuitive.

We are confident that our app will help you stay organized and productive, and we encourage you to try it out for yourself.

8. References sources

- Kotlin Reference:[[Kotlin Docs | Kotlin Documentation \(kotlinlang.org\)](#)]
(<https://kotlinlang.org/docs/home.html>)
- CSDN: ([25条消息](#)) [CSDN - 专业开发者社区](#)
- Baidu SDK reference: [Android地图SDK | 百度地图API SDK \(baidu.com\)](#)