

# 浙江大学

## 智能终端软件开发期末大作业报告



应用名称: Wanderlust 浪迹

课程名称: 智能终端软件开发

任课教师: 张 寅

组 长: 臧天铠

学 号: 3180104057

联系方式: 13968796267

组 员: 徐 珂

学 号: 3180103434

日 期: 2020.11-2021.1

学 号: 3180104057

## 目录

一、项目介绍 .....	3
二、总体架构 .....	4
三、功能设计 .....	6
四、功能实现与关键结构、算法.....	9
五、项目分工 .....	22
六、心得与讨论 .....	23

## 一、项目介绍

### 1.1 项目背景

旅游，出差，探亲……人们出于各种各样的目的，经常需要去往很多地方。而珍贵的一些回忆、琐碎的行程却常常招致遗忘，留下遗憾；有时想统计去过的地方，无从清晰地标记。

本 app 专为解决这些问题而开发，旨在帮助用户以简明而直观的方式管理、规划自己的行程。用户可以记录去过的地方、某一天的行程、在某个地方写下的笔记。通过简单的操作，用户可以查看过去某一天曾去过的地方、上传的照片、记下的“流水账”；也可以为未来的某一天安排行程。此外，还可以根据去过的地方，点亮地图的颜色，呈现可视化的“个人地图”。

本项目适合旅游爱好者、经常外出者以及各种有该类需求的用户。

### 1.2 与同类 APP 对比

同种类型的 App，包括各种日记、手账型的软件；高铁管家等购票软件有类似的行程记录功能；另一种程度上，todo list 类型的 App 也有相似之处。

不同于普通的记录类软件，以地图为核心的开发相对较少。该项目则是明确以“地图”、“地点”为开发重心和创新点，集成各类型软件相关功能，更多地将设计落实在不同地点、跨地区的应用功能，更具倾向性。该项目还支持以可视化的方式呈现去过的地区，可以设置颜色等操作点亮来进行管理。

相比之下，项目可以作为一个专门的平台，用户可以在此汇总来自不同购票软件的行程记录，生成自己的数据库，便于查询与管理。同时，相比 todo list，在该项目 app 上的记录，可以与工作学习任务清单、日常日记分开，清爽简明地专门记录旅游、出行日志，设置未来行程或想去的地方。所有旅游出行相关的事项，可以在该项目中形成一个独立的生态。

### 1.3 运行环境

开发过程中可以正常使用的环境包括：

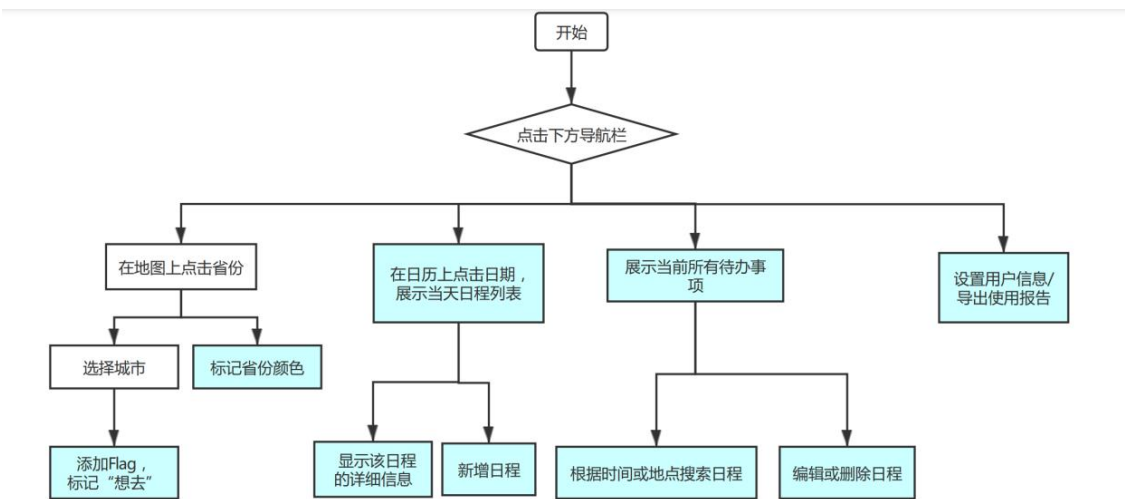
Pixel 2 API 25  
Pixel 2 API 29

## 二、总体架构

### 2.1 软件总体设计

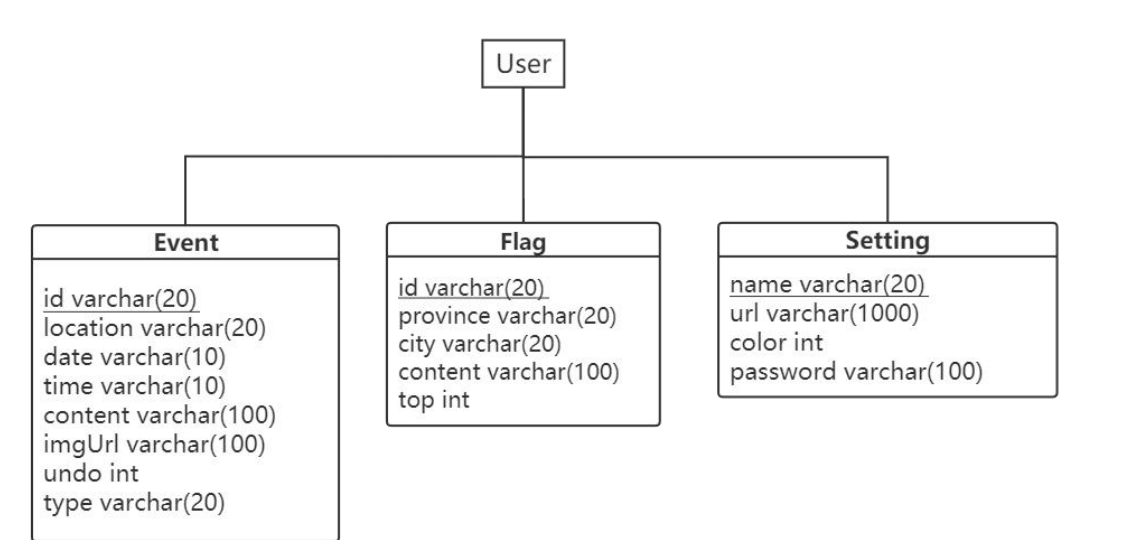
本应用一共分为四个页面，软件执行流程图如下。

在流程图中标蓝的部分均需要从数据库中读取信息或者向数据库中写入信息，整个应用与数据库有着紧密的联系，各个页面之间通过数据库实现了数据的互通。



### 2.2 数据库设计

在本应用中，我们设计的数据库主要涉及三张表。用户可以通过应用的图形界面改变表中的信息，并且利用“设置”页面中的 Report 功能，统计各个表中的信息，生成个人数据统计报告。



(1) Event 用于存储日程信息。

- id 字段是 Event 表的 Primary key，用于标识不同的日程，每个日程拥有一个唯一的 id。

- location 字段用于存储日程发生的地点。
- date 字段用于存储日程发生的日期。
- time 字段用于存储日程发生的具体时间。
- content 字段用于存储日程的具体内容。
- imgUrl 字段用于存储日程相关的图片链接。
- undo 字段用于存储日程的状态（待办/过期）。
- type 字段用于存储日程的类型（工作/交通/旅游）。

(2) Flag 用于存储期望访问的城市信息，在每一个被用户标记为“想去”的城市，我们都会添加 Flag 信息。

- id 字段是 Flag 表的 Primary key，用于标识不同的旗帜。
- province 字段用于存储该旗帜所在的省份。
- city 字段用于存储该旗帜所在的城市。
- content 字段用于存储该旗帜的具体备注信息。
- top 字段用于标识该旗帜是否被置顶。

(3) Setting 表用于存储系统设置信息。

- name 字段是 Setting 表的 Primary key，用于存储用户昵称。
- url 字段用于存储用户头像的图片链接。
- color 字段用于存储当前设置的主题颜色。
- password 字段用于存储用户密码。

### 三、 功能设计

根据功能的实现与效果，App 大致可以分为四个 Bar 模块。下面依照改设计思路依次介绍功能设计。

#### 3.1 可视化地图管理

在该视图板块，Wanderlust 拟呈现中国省份地图，以及进行操作的控制面板。

下面依次介绍相关功能。

##### 3.1.1 省份城市选择

在可视化地图上选择某一省份，对应省份会呈现黄色高亮。

选中后可以在下拉框中选择具体的城市，例如：浙江省-杭州市。

##### 3.1.2 按地区查看事件

选中城市后，点击确定，即可进入展示页面，按时间顺序查看这一城市所有的笔记、图片。

##### 3.1.3 修改省份颜色

所有省份最开始默认为灰色。用户可以选中不同的省份后，手动进行高亮与颜色的设置；清晰明了地显示去过的地方，设置自己喜欢的颜色，呈现出个性化的地图，给予用户满满的成就感。

##### 3.1.4 查看、新增 flag

选中城市后，点击小旗子图标，则可以进入相应界面查看“当前省份”的所有 flag 事件。

在页面下方，输入需要新增的信息，并点击 ADD 按钮，即可添加“当前城市”的 flag 信息，即设置为“想去”。

##### 3.1.5 删除、置顶 flag

在查看 flag 事件的界面，向左拖动事件，可以选择[置顶](#)或[删除](#)记录。

置顶后小旗颜色会改变，并且在上方优先显示。

## 3.2 按日期操作事件

### 3.2.1 选择日期

点击日历插件的不同日期，可以选择日期，选中的日期会被标出。

点击右上角的日历可以直接跳转到今天的日期。

日历上拉可以收起，可以左右滑动。

### 3.2.2 查看事件

选中日期后，会在日历下方以列表的形式显示所有在这个日期发生的事件，并且显示相关的信息（类型、内容、是否待办）。

### 3.2.3 添加事件

点击左上角的 ADD ITEM 按钮，可以新增一个事件。

进入新增时间界面后，需要输入文字，选择输入最多 9 张图片，以及地点和事件类型。完成后点击添加，即创建成功。

## 3.3 待办与搜索

### 3.3.1 查看待办事项

在该 Bar 中，会按时间远近顺序显示出所有待办的事项。

### 3.3.2 编辑和删除待办事项

长按某一事件，可以选择对该事件进行编辑，如：修改文本内容、增删图片等。

长按某一事件，可以选择删除该事件。

### 3.3.3 搜索事件

在上方的搜索框内输入日期或地点，则可以显示所有相关的事件。

### 3.4 用户与设置

#### 3.4.1 设置用户昵称、头像

点击可进入修改界面，可以修改用户头像和昵称。

#### 3.4.2 查看个人报告

可以查看根据用户所有记录统计生成的个人报告。

包括，去过城市的数量、最喜欢去的城市、记录的数量等等等等...

#### 3.4.3 修改私人密码

输入原密码，并输入两次新的密码，如果原密码正确且两次新密码相同，则诚信设置密码成功。

#### 3.4.4 查看开发者信息

点击“关于我们”查看开发者的信息。



## 四、功能实现与关键结构、算法

### 4.0 数据库接口类设计

在设计数据库时，我们分别设计了 DatabaseHelper 类和 ScheduleDao 类。

DatabaseHelper 类主要实现了数据库的创建和表的建立功能。

ScheduleDao 类则实现了日程相关操作与数据库之间的接口，这是因为在项目开发前期，数据库设计主要由一位同学实现，而整个项目的实现又高度依赖于数据库，增加这个接口可以方便我们对数据库的操作，我们只需要对该接口做一定的修改就可以满足需求，这在开发过程中减少了互相等待和修改的时间。

### 4.1 可视化地图管理模块

#### 4.1.1 整体结构

Bar1 的显示主要由两部分组成。从上到下分别为 MapView 和 Bar\_1\_Selector 两部分组成，分别对应地图和控制面板。两个 Fragment 通过动态绑定链接到 Bar\_1 的主界面上。





```

public boolean isTouch(float x, float y) {
    //创建一个矩形
    RectF rectF = new RectF();
    //获取到当前省份的矩形边界
    path.computeBounds(rectF, exact: true);
    //创建一个区域对象
    Region region = new Region();
    //将path对象放入到Region区域对象中
    region.setPath(path, new Region((int)rectF.left, (int)rectF.top, (int)rectF.right, (int)rectF.bottom));
    //返回是否这个区域包含传进来的坐标
    return region.contains((int)x, (int)y);
}

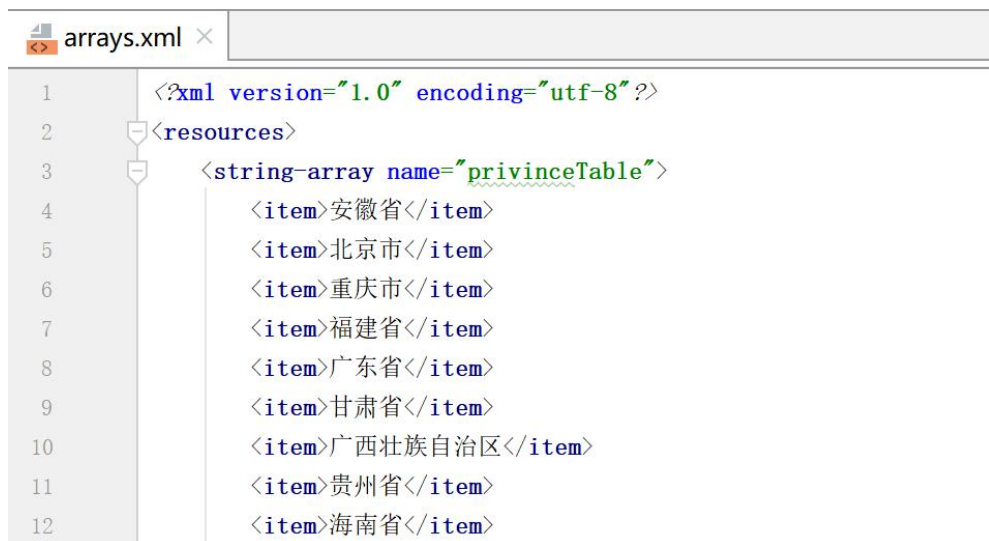
```

如果某一省份被选中后，则将 MapView 中的 select 对象设为该省的对象，并且绘制时显示不同的颜色（黄色）。

当选中一个省份后，MapView 会调用 Selector 中的下拉列表的适配器进行刷新，即用当前的省份名来寻找对应的城市。

相关的数据，我们储存在 array.xml 中，在 Selector 启动时读取成 ArrayList 并交给适配器来管理。

array.xml 中的数据如下：



```

1      <?xml version="1.0" encoding="utf-8"?>
2      <resources>
3          <string-array name="privinceTable">
4              <item>安徽省</item>
5              <item>北京市</item>
6              <item>重庆市</item>
7              <item>福建省</item>
8              <item>广东省</item>
9              <item>甘肃省</item>
10             <item>广西壮族自治区</item>
11             <item>贵州省</item>
12             <item>海南省</item>

```

...



根据城市名更新下拉适配器的代码如下：

```

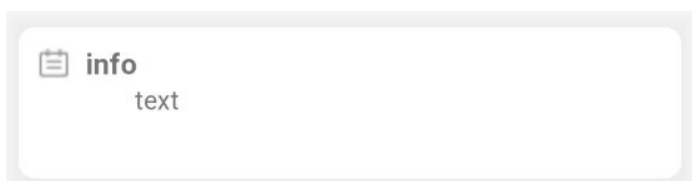
// renew the chosen information
public void renewAdapter(String n){
    name = n;
    int resId = getResId(name, R.array.class);
    if(resId== -1) return;
    adapter = ArrayAdapter.createFromResource(getActivity(), resId, android.R.layout.simple_spinner_item);
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spinner.setAdapter(adapter);
}

```

#### 4.1.4 查看事件

用户点击确定后，在 Bar\_1\_Display 界面中显示一个城市的所有事件，其通过 RecyclerView 以列表的方式呈现，每一个事件为一个 Bar\_1\_Item，下面介绍该类。

该类中，包含一个时间的日期、时间、文本和 bmp 图片（最多 9 张）信息。其对应的布局如下：



在 Bar\_1\_Display 中，主要思想是在数据库中搜索 location 和当前省份+城市对应的所有时间，然后对每个事件都新建一个 Bar\_1\_Item 放入一个列表中，而这个列表绑定有一个 adapter 进行管理并负责在 RecyclerView 显示。

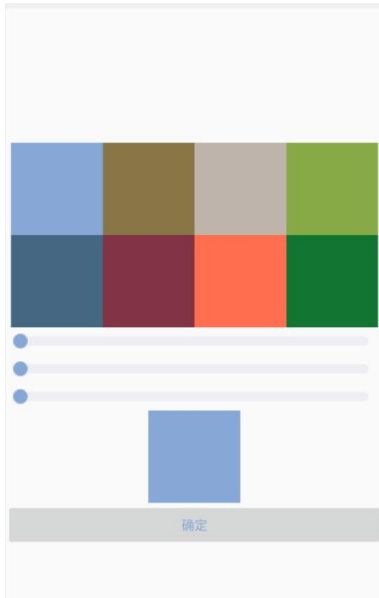
该类为 Bar\_1\_ItemAdapter，其功能主要为，为每一个 Bar\_1\_Item 建立绑定布局 view，并且设置相关的 TextView 显示时间、内容和图片信息。

其中图片的显示由我们自己编写的 NineGridLayout 实现，该类继承自 GridLayout，可以根据向其中添加的图片，自动调整行列，显示 0-9 张图片。

#### 4.1.5 修改省份颜色

选中省份后，点击选择颜色按钮可以设置当前选中省份的颜色。

颜色设置界面如下：



其中包含 8 个，固定的颜色供选择，点击即可设置为当前颜色，显示在下方预览中。

另外有三个 bar 分别对应 rgb 三个 channel 用户可以自己调配喜欢的颜色，结果同样会显示在下方预览中。

点击确定，则会设置成当前颜色。

关于颜色的储存，有一个专门的数据库 table，储存设置过的省份名和颜色。在地图初始化的时候，会便利这个表，为每一个省份选择颜色，如果搜索结果为空，则为默认灰色。

如果多次修改颜色，则会删除原来的记录，再重新添加，避免重复。

MapView 有一个 public 方法接受一个省份名和颜色值来进行当前颜色的设置和数据库颜色的修改。在颜色选择 Activity 结束后，Bar\_1\_Selector 会调用该函数进行修改。

#### 4.1.6 查看、新增 flag



选中省份城市后，点击小旗子标志，即可显示当前省份的所有 Flag 事件，并且可以选择添加当前城市的新 Flag。

Flag 信息的显示，同样通过 RecyclerView 来实现，需要一个 item 和 adapter 配合来管理显示，此处不多赘述，原理和 4.1.4 中所述大同小异，具体可参见 Bar\_1\_Flag\_Item 和 Bar\_1\_Flag\_ItemAdapter 代源码。

#### 4.1.7 删除、置顶 flag

为了美观和用户友好，此处采用了 github 上一个开源的组件——SwipeDelMenuLayout，主要作用是可实现 item 左拉后呈现置顶和删除按钮。

点击删除按钮后，会根据 Flag 的 ID，从数据库的 Flag table 中删除这一记录，然后通知 Adapter 进行修改。

点击置顶按钮后，会根据 Flag 的 ID，设置数据库的 Flag table 中这一记录的 top 属性为 1，此时 icon 的小旗子会变为绿色。再次点击归零，然后通知 Adapter 进行修改。

最终的显示结果，会根据 top 降序排序，因此，置顶的 item 会显示在更前面。



## 4.2 按日期操作事件

### 4.2.1 选择日期

为了实现美观且易于交互的界面，在本页面中使用了 GitHub 上的一个开源日历组件 -NCalendar，在项目的 app/build.gradle 文件中添加对该开源组件的包依赖后，即可使用该组件的功能。

```
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
    implementation 'androidx.appcompat:appcompat:1.0.0'  
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'  
    implementation 'com.google.android.material:material:1.0.0'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.0'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'  
    implementation 'androidx.recyclerview:recyclerview:1.1.0'  
    //日历  
    implementation 'com.necer.ncalendar:ncalendar:4.3.8'  
    //
```

日历页面的布局文件为 fragment\_bar\_2.xml，在布局页面中添加了对该组件中 Miu9Calendar 对象的使用。根据 Github 上的 NCalendar 项目说明，通过在日历组件中嵌套使用 RecyclerView 组件可以实现灵活的瀑布式页面布局，实现日历上拉收起的功能。基于该日历组件的原生功能，左右滑动日历可以切换月份。



```

116         <com.necer.calendar.Miui9Calendar
117             android:id="@+id/myCalendar"
118             android:layout_width="match_parent"
119             android:layout_height="match_parent"
120             android:layout_marginLeft="10dp"
121             android:layout_marginRight="10dp"
122             app:bgCalendarColor="#44eaf2f8"
123             app:pointColor="#5dade2"
124             app:selectCircleColor="#1f618d">
125
126             <androidx.recyclerview.widget.RecyclerView
127                 android:id="@+id/recyclerView"
128                 android:layout_width="match_parent"
129                 android:layout_height="match_parent" />
130         </com.necer.calendar.Miui9Calendar>

```

该页面类用 Bar\_2.java 文件实现，我们基于该日历组件的原生功能，添加了一些个性化的设计，实现了良好的页面交互特性。

在日历中，可以将当前选中的日期用圆圈标识出来；利用 Bar\_2.java 中的 localDate2DateString 函数我们将日期信息转换成字符串，以便将当前所选的日期显示在标题栏中。

```

129     @Override
130     public void onResume() {
131         super.onResume();
132         // 查询当前日期的列表内容
133         setRecyclListData();
134     }
135
136     public void setRecyclListData() {
137         // bar2ItemnewList = scheduleDao.queyrByDateScheduleList(mDate);
138         bar2ItemnewList=scheduleDao.queyrByDateScheduleList(String.valueOf(StringUtils.parse(mDate).getTime()));
139         adapter.setNewData(bar2ItemnewList);
140     }
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

在页面的右上角，我们设置了一个日历形状的按钮，通过对该按钮添加点击事件监听，实现了点击该按钮后跳转到今天的日期的功能。

```

69         btnAddItem.setOnClickListener(this);
70         today.setOnClickListener(new View.OnClickListener() {
71             @Override
72             public void onClick(View view) { myCalendar.toToday(); }
73         });
74
75

```



#### 4.2.2 查看事件

选中日期后，会在日历下方以列表的形式显示所有在这个日期发生的事件。  
这一步的具体实现流程如下：

(1) 我们设计了 Bar\_2\_Item 类，用于存储日程信息。日程信息包括日程内容、日程发生的日期和具体时间、日程图片、日程类型、日程地点。

(2) 我们在 Bar\_2 类中设置了一个成员变量的 List，用于存放 Bar\_2\_Item 类对象 bar2ItemnewList，该列表中每一个的 Bar\_2\_Item 类对象都对应一条日程信息。

```
37 public class Bar_2 extends Fragment implements View.OnClickListener {
38
39     Miui9Calendar myCalendar;
40     RecyclerView recyclerView;
41     ImageView image;
42     TextView date;
43     ImageView today;
44     ImageView back;
45
46     private boolean isResumeFirst = false;
47
48     private LocalDate selectedDate;
49     private String mDate; // 当前日期
50     private Button btnAddItem;
51     private ScheduleDao scheduleDao;
52     private List<Bar_2_Item> bar2ItemnewList = new ArrayList<>();
53     private ScheduleListAdapter adapter;
```

(3) 在 Bar\_2 的页面类中，我们定义了一个 ScheduleListAdapter 类对象 adapter。该成员变量用于控制日历页面中，日历组件下方的日程列表显示。在页面创建时，我们用当前选中的日期对应的日程信息列表对该 adapter 进行初始化，同时我们需要将该 adapter 纳入 recyclerView 的布局中，保证日历组件向上收起时，下方的日程列表可以正常显示。

```
93     adapter = new ScheduleListAdapter(bar2ItemnewList);
94     recyclerView.setAdapter(adapter);
```

(4) 当页面每次重新加载时，调用 onResume 函数并嵌套调用 setRecycllListData 函数，在该函数中调用数据库接口 scheduleDao 查询当前所选中的日期对应的日程列表，并且更新 Bar\_2 类对象中用于存储当前页面显示的日程信息列表类成员，即 bar2ItemnewList。

```
129     @Override
130     public void onResume() {
131         super.onResume();
132         // 查询当前日期的列表内容
133         setRecycllListData();
134     }
135
136     public void setRecycllListData() {
137         // bar2ItemnewList = scheduleDao.queryrByDateScheduleList(mDate);
138         bar2ItemnewList=scheduleDao.queryrByDateScheduleList(String.valueOf(StringUtils.parse(mDate).getTime()));
139         adapter.setNewData(bar2ItemnewList);
140     }
```

### 4.2.3 添加日程

在 Bar\_2.java 中，我们添加了对左上角的 ADD ITEM 按钮的点击事件监听，当监听到点击动作时，调用 InsertAttendanceInfoActivity，可以对当前选中的日期新增一项日程。

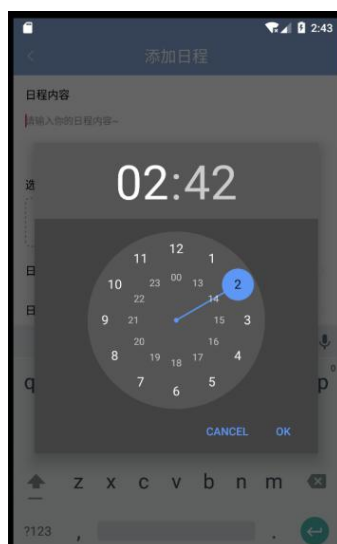
```
143         @Override
144         public void onClick(View v) {
145             switch (v.getId()) {
146                 case R.id.bar_2_additem:
147                     startActivity(new Intent(getActivity(), InsertAttendanceInfoActivity.class).
148                         putExtra( name: "currentDay",mDate));
149                     break;
150             }
151         }
```

新增日程页面的布局文件为 activity\_insert\_attendance\_info.xml，新增日程的动作则通过 InsertAttendanceInfoActivity 类实现。

在新增日程时，我们利用 setToNow() 函数设置默认的日程时间为系统当前的时间；同时我们将通过 LocationUtils 类获取当前用户所在的位置，将当前所在的城市默认为日程地点，该类的实现主要通过参考 Github 和 CSDN 上的开源代码实现。

```
88         String location= LocationUtils.getInstance().getLocations( context: this);
89         tvChooseCity.setText(location);
```

点击日程时间的数字，将进入日程时间设置页面，调用安卓的原生的时间选择器组件 TimePickerDialog，方便用户选择日程的具体时间，进行个性化的设置。



点击日程的具体地点，将进入日程地点设置页面，调用 cityPicker 组件，方便用户选择地点。注意该组件的使用需要添加依赖，项目的 build.gradle 文件中需要添加如下的配置语句：

```
implementation 'liji.library.dev:citypickerview:0.7.0'
```

我们添加了竖直滑动选择器实现日程类型的选择，滑动选择器的具体实现可以参考 WheelViewDataBean 类。

我们可以为日程添加图片，图片添加包含拍照和从相册获取两种模式。如果使用相机模式则需要获取系统的相机权限；图片的加载和显示参考了 GitHub 上的开源项目，通过 GlideEngine 类实现。

当日程信息填写完毕后，点击提交按钮，调用 commitSchedule() 函数，我们将利用数据库的接口类对象 dao 将新增的日程信息加入到数据库中。

```
320 ScheduleDao dao = new ScheduleDao( context: InsertAttendanceInfoActivity.this);
321 dao.add(bar2Itemnew);
```

## 4.3 待办与搜索

### 4.3.1 查看日程

待办页面由 Bar\_3.java 类控制，布局文件为 fragment\_bar\_3.xml，在该页面中，显示用户创建的所有日程，按照日期顺序排列。系统根据日期判定日程状态，早于系统当前日期的日程将被判定为过期；等于或者晚于当前日期的日程将被判定为待办。

与 Bar\_2 类似，在 Bar\_3 类对象中我们创建了一个 Bar\_2\_Item 类的列表用于存放日程信息，并通过 recyclerview 布局 and ScheduleListAdapter 配合显示。

```
45 @  private void initView(View view) {
46     scheduleDao = new ScheduleDao(getActivity());
47     recyclerView = view.findViewById(R.id.recyclerView);
48     recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()), LinearLayoutManager.VERTICAL, reverseLayout);
49     adapter = new ScheduleListAdapter(bar2ItemList);
50     recyclerView.setAdapter(adapter);
```

### 4.3.2 编辑和删除待办事项

长按某一日程，可以对该日程进行编辑，编辑按钮的点击事件与 EditScheduleActivity 类绑定，EditScheduleActivity 类的实现与新增日程的 InsertAttendanceActivity 类十分类似，只不过将对数据库的“增”操作变更为了“改”操作。

长按某一日程，可以选择删除某一日程，当监听到删除按钮的点击动作时，直接调用数据库接口类对象中的删除方法删除对应的日程。

```
111 long _id = bar2ItemList.get(position).get_id();
112 //删除数据
113 scheduleDao.deleteScheduleData(_id);
```

事

### 4.3.3 搜索事件

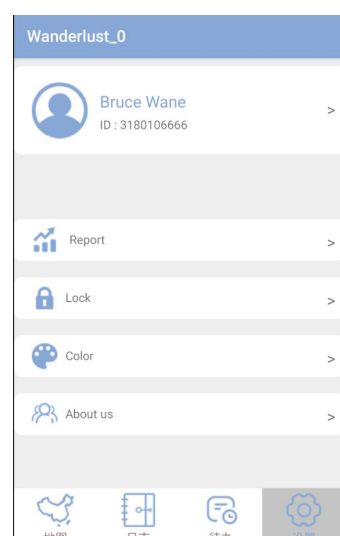
点击页面上方搜索框，触发 SearchActivity，该类对日程信息显示的管理模式类似之前的几个页面，综合利用了 List、ScheduleListAdapter 和 RecyclerView，同样通过 ScheduleDao 类接口访问数据库。

```
31 public class SearchActivity extends FragmentActivity implements View.OnClickListener {
32
33     private EditText etSearch;
34     private ScheduleDao scheduleDao;
35     private RecyclerView recyclerView;
36     private ScheduleListAdapter adapter;
37     private List<Bar_2_Item> bar2ItemList = new ArrayList<>();
38     private List<Bar_2_Item> listAll;
```

在搜索框内输入文本后点击搜索按钮，调用 filterData() 函数；在次函数中我们首先执行 clear 语句，清空搜索缓存，接着根据输入的内容在数据库中搜寻对应的日程，并且将对应日程加入当前页面的日程显示列表中。

```
116 private void filterData() {
117     bar2ItemList.clear();
118     String search = etSearch.getText().toString().trim();
119     if (TextUtils.isEmpty(search)) {
120         Toast.makeText(context, SearchActivity.this, text: "请输入搜索内容", Toast.LENGTH_SHORT).show();
121         return;
122     }
123     for (int i = 0; i < listAll.size(); i++) {
124         Bar_2_Item bean = listAll.get(i);
125         if (StringUtils.getStringDate(bean.getScheduleDate()).contains(search) ||
126             bean.getScheduleLocation().contains(search)) {
127             bar2ItemList.add(bean);
128         }
129     }
130     adapter.setNewData(bar2ItemList);
131 }
```

## 4.4 用户与设置模块实现



#### 4.4.1 用户头像与昵称设置

点击最上面一栏可以进入修改。

修改的过程非常简单，点击编辑昵称后，会将输入的新名字写入数据库中。

点击编辑头像，则会跳出拍摄或相册选取，相应地调用相机或相册，并返回图像的 URL 存入数据库中。



#### 4.4.2 用户个人报告

点击报告栏，可以生成个人的浪迹报告。

主要的实现思路，就是进行一些相对复杂的 SQL 搜索，并将结果插入对应的文本中。

其中需要注意搜索结果为空、为 0 的时候，需要显示不同的内容。



### 4.4.3 设置密码

要求输入原密码和两次相同新密码，有错则跳出 Toast 提示，无误则通过 SQL 语句修改数据库中的密码为新的密码。

### 4.4.4 关于我们

呈现准备好的开发者信息的 Activity。

## 五、项目分工

在本次项目过程中，小组内部实现了合理有序的分工，实现了良好的团队协作。在项目的前期和中期，我们一共组织了三次比较充分的线下讨论，在三次讨论中，我们分别确定了项目选题、项目的开发思路（包括用户使用流程、界面跳转和初步的数据库设计），探讨了开发过程中遇到的问题并且依据遇到的问题对原有的思路进行了一些修改。

在小组分工中，徐珂同学主要负责开屏动画、日志页面和待办页面的相关功能实现，臧天铠同学主要负责数据库接口整合、地图页面和设置页面的相关功能实现。

在项目的开发过程中，我们充分利用 GitHub、GitHub Desktop 等工具实现了远程协作和版本控制，项目开发流程清晰。

项目的 GitHub 链接为 <https://github.com/CosetteX/Wanderlust>。

The screenshot displays the GitHub Desktop application interface. On the left, the 'Changes' panel shows a list of commits. The 'Added on 1.5' commit is highlighted, showing it was made by 'Zang' 19h ago. The main area shows a diff view for the 'Wanderlust\_1\_...Activity.java' file. The diff highlights changes in the 'InsertAttendanceInfoActivity' class, showing additions and modifications to various fields and methods. The changes include updates to the 'mList1' and 'mList2' lists, the 'baseDialog' field, and the 'tvChooseCity' and 'tvChooseTime' views. The diff also shows changes to the 'onItemClick' method in the 'ImageGoodsAdapter' class.

## 六、心得与讨论

徐珂：

通过本次课程项目的开发，我和队友一起完成了我们的第一个安卓 app。开发的过程和之前我在微信小程序课程上的体验比较类似，但是也有不同的地方；在本次项目中我们尝试了对数据库的使用，一开始由于对数据库操作不熟悉遇到了较大的困难，而且两个人分头工作也导致数据库不太容易整合，后来通过阅读实验指导书和查询网络上的资料找到了比较好的解决方法。

同时，在开发过程中，我发现网络上有许多实用的安卓开源组件，可以减轻我们的开发成本，使安卓应用的开发变得更加简单，这是我之前在微信小程序课程中没有深刻体会的。尤其是本次使用的日历组件，在 Github 上面可以看到这个日历组件项目的迭代过程，通过阅读他们的代码我也感受到我们的代码规范性还有待提高；网络上优秀的开源组件往往能提供非常易于使用的接口，通过接口降低代码的耦合度，这是值得我们学习的。

最后要感谢这门课程让我初次体验了安卓开发的乐趣，并且感谢我的队友，本次作业的合作过程非常愉快~

臧天铠：

本次项目开发的周期相对比较长，我们通过 github 进行协同开发。

这是我第一次接触移动端的软件开发。通过本次大作业，我对于安卓开发的各个方面的技术细节，比如各类空间、动画、数据库、自定义 View、线程等等，因此上课教授的理论有了进一步的理解和实践，并且通过查阅资料，也学习了如何借鉴他人的开源控件，并从中学习到一些设计的思想和结构。另外，也体会到了从头开始设计、实现一个 APP 的过程，感受到了其中很多思想和乐趣。

如徐同学所述，数据库是开发过程中的一个重点，我们在开发的过程中也经历了多次的修改和迭代，从中的启发是，最好一开始有比较完善、可扩展、符合范式的数据库设计，这样对于后面的修改需求，能提供极大的便利。

总体感受是，对于一些大的设计不是特别困难，但是在一些细节的地方，确实需要比较多的时间去打磨和调试。例如，在修改数据库的时候，也需要实时修改显示的内容，否则会造成一些不统一；以及 bar 控件的 UI 美化我花费了较多的时间。

另外由于我们两个人虚拟机版本的不同，中途造成了一些小的问题，造成闪退等现象，例如系统自带 icon 格式的不一致等。但最终通过调试和阅读档案发现问题并解决。

最后感谢我的队友的辛勤付出和老师一学期的指导。