

# Android项目简介

---

组员：张馨月，贾政雄，刘行知，韦卓辉

# 开发背景

---

- 随着当今生活节奏的加快，人们对时间的利用正朝着高效的方向发展。
- 对于当今的大学生而言，繁重的课业压力意味着需要更有效地利用碎片时间，比如在等待外卖的间隙阅读一篇好文章，在课后到饭前的一个小时内跑跑步。而纯粹依靠记忆和自觉来利用碎片时间，往往达不到我们想要的高效效果。因此，越来越多的时间管理APP应运而生。
- 而APP Store里五花八门的时间管理类应用往往不够简洁和便利，一则一些用处不大的功能会占用手机的内存空间，二则很多已经上市的时间管理类APP实际并不会建议用户如何更高效地利用时间，而只停留在记录任务、提醒任务的层面。针对上述现状，本小组提出了名为Slice的时间管理类APP，它能够给用户提供更智能的时间利用建议。

# 目标功能

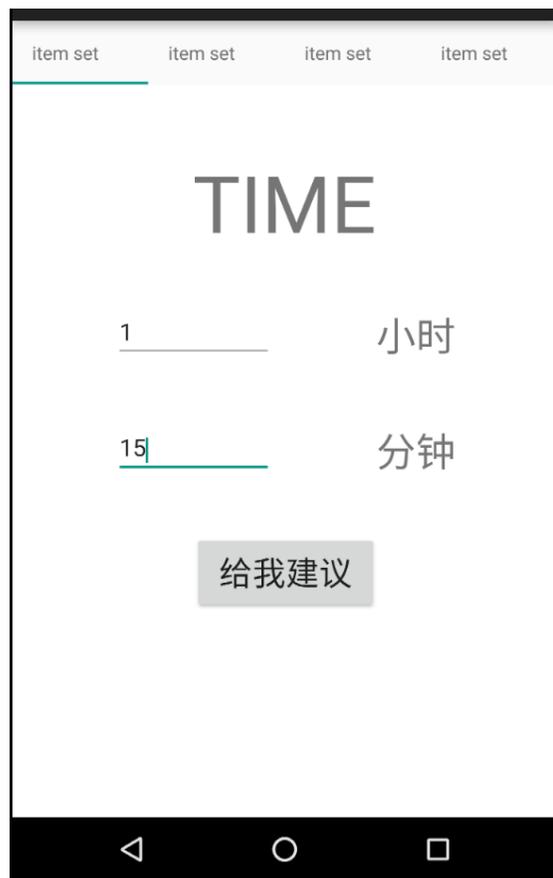
---

针对进行简洁的时间管理的设计目标，我们确定了本APP项目预期实现的基本功能

1. 辅助用户记录待办事项，配以特定的辅助信息后存入本地数据库中。
2. 用户可以输入特定的空闲时间，并通过算法从数据库中抽选出该时间内可以选择进行的事项列表，并通过优先级排序显示。
3. 用户可以根据喜好，调整推荐算法，获得更符合用户意愿的推荐。
4. 用户可以随时查看尚未完成的事项，和最近已经完成的事项。

# APP主界面

---



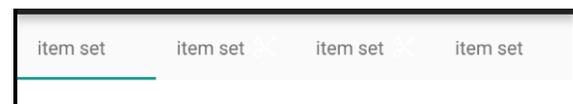
# 项目结构

- ▼ **app**
  - ▶ manifests
  - ▶ java
  - ▼ res
    - ▼ **layout**
      - activity\_third.xml
      - tab1.xml
      - tab2.xml
      - tab3.xml
      - tab4.xml
      - tab\_item.xml
    - ▶ mipmap
    - ▶ values
  - ▶ Gradle Scripts

## activity\_third.xml

- ▼ **LinearLayout (vertical)**
  - sliding\_tabs (CustomView)**
  - viewpager (CustomView) -**

## tab\_item.xml



- ▼ **RelativeLayout**
  - textView - "item set"**
  - imageView - @drawable/**

# Java文件

---

- © DatabaseHelper
- © MyTabItem
- © NextActivity
- © PageFragment
- © SimpleFragmentPagerAdapter
- © ThirdActivity

初始Activity :

ThirdActivity

Tabhost动态生成 :

SimpleFragmentPagerAdapter

PageFragment

MyTabItem

程序运行中隐式启动的Activity :

NextActivity

数据库操作 :

DatabaseHelper

# ThirdActivity.java的onCreate函数

---

```
viewPager = (ViewPager) findViewById(R.id. viewpager);
pagerAdapter = new SimpleFragmentPagerAdapter(getSupportFragmentManager(), this);
viewPager.setAdapter(pagerAdapter);

tabLayout = (TabLayout) findViewById(R.id. sliding_tabs);
tabLayout.setupWithViewPager(viewPager);

tabs = new MyTabItem[tabLayout.getTabCount()];

for (int i = 0; i < tabLayout.getTabCount(); i++) {
    TabLayout.Tab tab = tabLayout.getTabAt(i);
    tabs[i] = new MyTabItem(this);
    tab.setCustomView(tabs[i].getTabView());
}
```

# PageFragment.java

---

## onCreate

- 改变mPage的值，即当前显示的是tabhost中的哪个页面

## onCreateView

- switch(mPage) {
  - case1: 主界面，添加advisebutton的监听 —> 点击后启动NextActivity
  - case2: 显示app给出的建议事件排序结果
  - case3: 添加事件界面，添加submitbutton的监听 —> 点击后将相关数据插入数据库
  - case4: 显示所有事件，其中界面2推荐过的事件高亮置顶
- }

# DatabaseHelper.java

---

## onCreate

```
@Override
public void onCreate(SQLiteDatabase db) {
    String sql = "create table main (name text, time integer, ddl integer, pri integer, type integer, flag integer);";
    db.execSQL(sql);
}
```

## onUpgrade

```
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("drop table if exists job");
    onCreate(db);
}
```

# 数据库的其他操作

---

## 打开数据库

//(可写方式)新建或打开数据库db

```
db = database.getWritableDatabase();
```

//(可读方式)新建或打开数据库db

```
db = database.getReadableDatabase();
```

## Cursor

```
Cursor cursor=db.query("main", null, null, null, null, null, null);  
for(int i=1;i<cursor.getCount();i++) {  
  
    cursor.moveToNext();  
  
    int need_time=cursor.getInt(1);  
    if(need_time<can_time) {  
        rank[count].t_name=cursor.getString(0);  
        rank[count].t_time=cursor.getInt(1);  
        rank[count].t_ddl=cursor.getInt(2);  
        rank[count].t_pri=cursor.getInt(3);  
        rank[count].t_type=cursor.getInt(4);  
        rank[count].t_mark=0;  
        count++;  
    }  
}
```

# 排序算法

---

```
for(int i=0;i<count-1;i++) {  
    for(int j=0;j<count-i-1;j++) {  
        if(rank[j].t_pri<rank[j+1].t_pri) {  
            exchange=rank[j];  
            rank[j]=rank[j+1];  
            rank[j+1]=exchange;  
        }  
    }  
}
```

```
for(int i=0;i<count;i++) {  
    if(can_time>=rank[i].t_time) {  
        can_time=can_time-rank[i].t_time;  
        rank[i].t_mark=1;  
    }  
}
```

算法待改进

目前主要依据：

预估任务时间在要求的范围内

优先级高

# 组员分工

---

界面设计：张馨月，贾政雄

数据库操作：贾政雄

排序逻辑：刘行知

工程整合：张馨月

项目文档：张馨月，韦卓辉

谢谢！