设计思想

# 一.文字信息的录入

## 首页的设计

首页的作用为：

展示备忘录记录列表

设置进入添加备忘录功能

点击查看或长按修改已有记录

对于添加备忘录功能我设置了悬浮窗来进入备忘录编辑界面，在实现悬浮窗功能时，我在**activity\_main.xml**中设置了**<com.google.android.material.floatingactionbutton.FloatingActionButton** 去定义悬浮窗功能，使用**<ListView**创建备忘录列表：

**<ListView**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"**

**android:id="@+id/list\_view"/>**

**//悬浮窗**

**<FrameLayout**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent">**

**<com.google.android.material.floatingactionbutton.FloatingActionButton**

**android:id="@+id/fab"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="end|bottom"**

**android:layout\_margin="16dp"**

**android:src="@android:drawable/ic\_input\_add" />**

**</FrameLayout>**

在**MainActivety.java**中首先为ListView创建成员对象：

**public ArrayList list=new ArrayList();**

**private ArrayAdapter<String> myadapter;**

使用**setContentView**函数应用刚刚定义好的布局，随后定义了悬浮窗按钮并为其建立了点击事件：

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_main);**

**//悬浮窗按钮**

**FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);**

**fab.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View view) {//点击悬浮窗按钮进入下一个页面**

**String data=null;**

**Intent intent=new Intent(MainActivity.this,input\_activity.class);**

**intent.putExtra("extra\_data",data);**

**startActivityForResult(intent,1);**

**}**

**});**

**}**

其中在悬浮窗按钮的点击事件中，我使用了显式intent先创建一个intent变量 其中两个参数为上下文所处的界面以及要传输的目标界面，我通过intent流实现两个页面间的数据传输，**intent.putExtra("extra\_data",data);**函数第一个参数是键，用于后面从intent中取值，第二个参数是要传的数据，通过**startActivityForResult(intent,1);**启动**input\_activity，**其中1为该活动的唯一标识，需确保唯一确定，在活动结束返回时要使用。

在**input\_activity**活动结束并返回时会调用setResult函数：

**setResult(RESULT\_OK,intent);**

其中第一个参数用于向上一个活动返回处理结果，第二个参数将带有数据的intent传递回去。

我们是通过**startActivityForResult**来启动下一个活动，而在下一个活动返回后会调用上一个活动的**onActivityResult（）方法**我们需要在主函数中重写这个方法来得到返回的数据：

**protected void onActivityResult(int requestCode, int resultCode, Intent data) {**

**switch (requestCode){**

**case 1:**

**if(resultCode==RESULT\_OK){**

**String returndata=data.getStringExtra("data\_return");**

**list.add(returndata);**

**ListView listView=(ListView) findViewById(R.id.list\_view);**

**myadapter=new ArrayAdapter<>(MainActivity.this, androidx.appcompat.R.layout.support\_simple\_spinner\_dropdown\_item,list);**

**listView.setAdapter(myadapter);**

**//监听点击事件**

**listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {**

**@Override**

**public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {**

**String data= (String) list.get(i);**

**Intent intent=new Intent(MainActivity.this,input\_activity.class);**

**intent.putExtra("extra\_data",data);**

**list.remove(i);**

**startActivityForResult(intent,2);**

**}**

**});**

**//长安点击事件**

**listView.setOnCreateContextMenuListener(new View.OnCreateContextMenuListener() {**

**@Override**

**public void onCreateContextMenu(ContextMenu menu, View view, ContextMenu.ContextMenuInfo contextMenuInfo) {**

**menu.setHeaderTitle("长按菜单-ContextMenu");**

**menu.add(0, 0, 0, "删除");//组，id,顺序，内容**

**menu.add(0, 1, 0, "其他功能");**

**}**

**});**

**}**

**break;**

**case 2:**

**if(resultCode==RESULT\_OK){**

**String returndata=data.getStringExtra("data\_return");**

**list.add(returndata);**

**ListView listView=(ListView) findViewById(R.id.list\_view);**

**myadapter=new ArrayAdapter<>(MainActivity.this, androidx.appcompat.R.layout.support\_simple\_spinner\_dropdown\_item,list);**

**listView.setAdapter(myadapter);**

**// }**

**}**

**default:**

**}**

**}**

**onActivityResult（）方法**有三个参数，第一个参数requestCode是我们在启动活动时传入的请求码即唯一标识，第二个参数resultCode，即我们在返回数据时传入的处理结果，第三个参数为携带者返回数据的intent。**（在intent传输数据这部分，我参考了郭霖主编的《第一行代码Android》（第二版）P41-P53）**

在返回活动中，可能会接受两个标识，其中标识为1的为点击悬浮窗按钮后的返回动作，表示为2的为点击ListView某一项后的返回动作。

先看case 1：

**case 1:**

**if(resultCode==RESULT\_OK){**

**String returndata=data.getStringExtra("data\_return");**

**list.add(returndata);**

**ListView listView=(ListView) findViewById(R.id.list\_view);**

**myadapter=new ArrayAdapter<>(MainActivity.this, androidx.appcompat.R.layout.support\_simple\_spinner\_dropdown\_item,list);**

**listView.setAdapter(myadapter);**

**//监听点击事件**

**listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {**

**@Override**

**public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {**

**String data= (String) list.get(i);**

**Intent intent=new Intent(MainActivity.this,input\_activity.class);**

**intent.putExtra("extra\_data",data);**

**list.remove(i);**

**startActivityForResult(intent,2);**

**}**

**});**

**//长安点击事件**

**listView.setOnCreateContextMenuListener(new View.OnCreateContextMenuListener() {**

**@Override**

**public void onCreateContextMenu(ContextMenu menu, View view, ContextMenu.ContextMenuInfo contextMenuInfo) {**

**menu.setHeaderTitle("长按菜单-ContextMenu");**

**menu.add(0, 0, 0, "删除");//组，id,顺序，内容**

**menu.add(0, 1, 0, "其他功能");**

**}**

**});**

**}**

**break;**

在case 1中 我们已经接收到来自input\_activity传回来的要保存的备忘录数据，利用**list.add(returndata);**将其加入数组使用 **ListView listView=(ListView) findViewById(R.id.list\_view);**找到之前定义好的ListView，因为数组中的数据是无法直接传递给ListView，所以需要一个适配器，在这里我使用了ArrayAdapter适配器，在 **ArrayAdapter**的构造函数中传入的三个参数依次为当前上下文，list\_item\_1作为ListView子项布局的id，是一个Android内置的布局文件里面有一个TextView，可以简单的显示一段文本。

随后调用setAdapter方法将构建好的适配器对象传进去，就建立了ListView和数据之间的关联，就可以在ListView上展示我们刚刚得到的数据了。

然后设置listView的点击事件，首先是点击修改内容事件，我们还是要启动建立好的input\_data活动去修改内容，但和悬浮窗点击事件不同的是我们刚进入时的文本框内容不为空，而是有点击之前已有的内容，所以我们先调用 **list.get(i)**方法获得我们目前ListView中已有的内容，创建显式intent，先将已有的内容传入intent流中删除现有的记录然后调用 **startActivityForResult**（）函数启动input\_activity此时的唯一标识码设定为2，若修改完成后则会进入case 2中。

在listView的长按点击事件中 **listView.setOnCreateContextMenuListener（）**启动长按点击事件，其中长安点击事件要重写**onCreateContextMenu（）和onContextItemSelected（）**两个函数，其中menu：准备创建的菜单View：触发弹出菜单的组件。以ListView为例，如果ListView的每个Item为一个布局文件[xml](https://so.csdn.net/so/search?q=xml&spm=1001.2101.3001.7020)，则View代表该布局文件生成的View，使用View可直接获取该布局文件中的相应的组件，menuInfo：该对象提供当前选中项item的信息。

在其中调用menu.add()函数即可为菜单添加item，其中传入的参数为第一个int类型的groupID参数，代表的是组概念，你可以将几个菜单项归为一组，以便更好的以组的方式管理你的菜单按钮。第二个int类型的item ID参数，代表的是项目编号。这个参数非常重要，一个itemID对应一个menu中的选项。在后面使用菜单的时候，就靠这个item ID来判断你使用的是哪个选项。第三个int类型的orderID参数，代表的是菜单项的显示顺序。默认是0，表示菜单的显示顺序就是按照add的显示顺序来显示。第四个String类型的title参数，表示选项中显示的文字。

**onContextItemSelected（）函数：**

**public boolean onContextItemSelected(MenuItem item) {**

**int id=item.getItemId();**

**AdapterView.AdapterContextMenuInfo menuInfo = (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();**

**System.out.println("info:"+menuInfo.position);**

**System.out.println("id"+id);**

**switch (id){**

**case 0:**

**list.remove(menuInfo.position);**

**myadapter.notifyDataSetChanged();**

**return true;**

**}**

**return super.onContextItemSelected(item);**

**}**

传入参数为被点击的item,通过**getItemId()**获得当前item的id并且通过不同id执行不同动作，在删除动作中通过**menuInfo.position获得**被长按的ListView的id通过remove删除，并且通过**myadapter.notifyDataSetChanged()**方法刷新ListView。

## 编辑页的设计

编辑页的作用为：

1. 提供一个文本框用于用户输入
2. 提供保存功能
3. 获取主页面传入的初始数据并将其放在文本框
4. 获取系统当前时间并于用户输入的数据一并回传给主页面

在编辑页页面设计上，通过<Button创建一个保存按钮，通过<EditText创建一个输入框用于用户输入：

**<Button**

**android:id="@+id/title\_save"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="center"**

**android:layout\_margin="5dp"**

**android:backgroundTint="#CABCBC"**

**android:text="保存"**

**android:textAlignment="center"**

**android:textColor="#000000"**

**android:textSize="24sp"**

**app:layout\_constraintEnd\_toEndOf="parent"**

**tools:ignore="MissingConstraints"**

**tools:layout\_editor\_absoluteY="2dp" />**

**<EditText**

**android:id="@+id/edit\_text"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:hint="请在这里输入要记录的内容"**

**android:maxLines="4"**

**app:layout\_constraintBottom\_toBottomOf="parent"**

**app:layout\_constraintTop\_toTopOf="parent"**

**app:layout\_constraintVertical\_bias="0.08"**

**tools:ignore="MissingConstraints"**

**tools:layout\_editor\_absoluteX="0dp" />**

在**input\_activity.java**中由于美观问题我首先使用了

**ActionBar actionBar=getSupportActionBar();**方法获取当前页面的标题，然后使用**actionBar.hide();**方法隐藏了当前标题随后使用

**String data=intent1.getStringExtra("extra\_data");**获取主页面传来的数据，先判断是否为空，来确定是通过悬浮窗点击进入还是通过过ListView点击进入，若是通过ListView点击进去则需要扔掉时间变量获取实际内容然后使用**setText（）方法**放入文本框中，其中三个参数为传入的数据，起始位置，结束位置。

随后通过**Calendar calendar = Calendar.getInstance();**创建calender变量然后通过

**int year = calendar.get(Calendar.YEAR);**

**int month = calendar.get(Calendar.MONTH)+1;**

**int day = calendar.get(Calendar.DAY\_OF\_MONTH);**

**int hour = calendar.get(Calendar.HOUR\_OF\_DAY);**

**int minute = calendar.get(Calendar.MINUTE);**

**int second = calendar.get(Calendar.SECOND);**

**String time=year+"年"+month+"月"+day+"日"+hour+":"+minute+":"+second;**

上述方法获得当前的时间信息并以一定形式传给字符串变量time。

随后通过**editText.getText().toString()**方法获得当前文本框中的文字，如果当前用户有输入则将其与时间变量组合并传回给主页面。