吴德文

[公司名称][公司地址]

酒店房间选择器

酒店房间选择器

[第一章 创意起源 2](#_Toc71624569)

[第二章 市场调查 2](#_Toc71624570)

[一、 应用市场调研 2](#_Toc71624571)

[第三章 应用设计图 2](#_Toc71624572)

[一、 原始设计 2](#_Toc71624573)

[1、 功能设计 2](#_Toc71624574)

[2、 界面设计 2](#_Toc71624575)

[二、 设计反馈 2](#_Toc71624576)

[三、 改善设计 2](#_Toc71624577)

[第四章 开发计划 3](#_Toc71624578)

[一、 模块设计 3](#_Toc71624579)

[二、 迭代开发进度 3](#_Toc71624580)

[第五章 开发步骤 3](#_Toc71624581)

[一、 项目雏形 3](#_Toc71624582)

# 创意起源

# 市场调查

这么需求想法是非常小的场景中产生的，它只能算是一个小应用，理论上来说，市场上应该不会有同类型的应用。

## 应用市场调研

经过在华为应用市场、腾讯应用宝、小米市场……等应用市场中查找了一番，确实是没有找到类似的应用。

居然还真找到了和我一样想法的产品。下载产品进行横向比较，*诸如此类、之乎者也*。

# 应用设计图

## 原始设计

### 功能设计

### 界面设计

## 设计反馈

## 改善设计

# 开发计划

## 模块设计

## 迭代开发进度

## 模型设计

### 酒店房间（RoomItem）

酒店房间是项目中最核心的数据，抽象得到的特征包括房间号、楼层、朝向等等。

在原型开发时期，我们的界面只需要把房间号显示出来，因此只要用字符串来表达房间模型即可。

当进行功能迭代完善时，定义一个房间类（为避免和系统 Room 类混淆，类名取为RoomItem），类中包含房间号（number）、楼层（floor）、标签（label）等属性。

# 开发步骤

## 原型开发

根据开发计划，先开发一个显示所有房间（号）的原型 App，如图 1所示。可以知道App只需要一个界面即可，由于房间信息只有存为字符串的房间号，因此界面中显示的是字符串的列表。

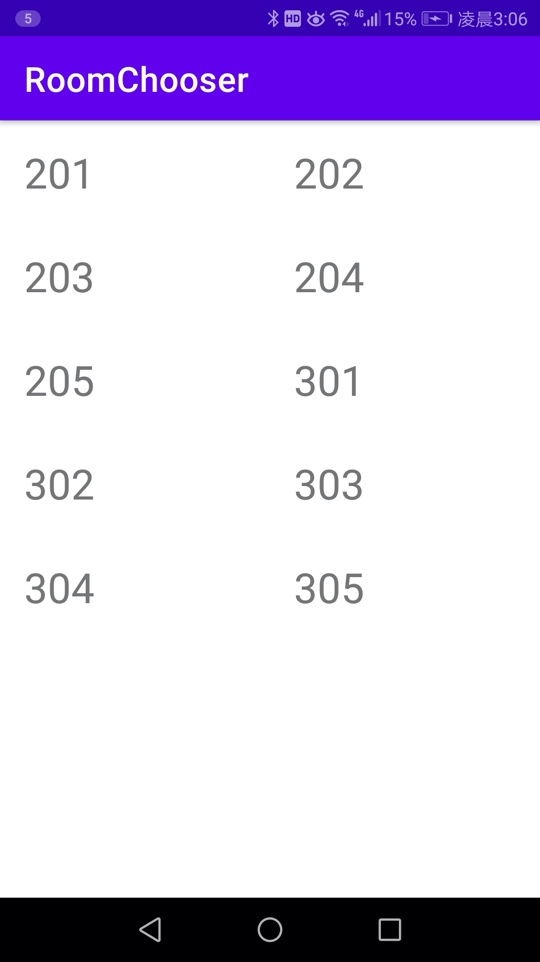


图 1 原型 App 运行效果

### 界面设计（res/layout/activity\_main.xml）

界面只需要一个列表显示区域，也就是只需要一个RecyclerView就可以，因此使用LinearLayout布局就足够——理论上，把RecyclerView作为布局的 Root 节点也是可以的，但为了便于后续扩展，还是套上一层布局为好。

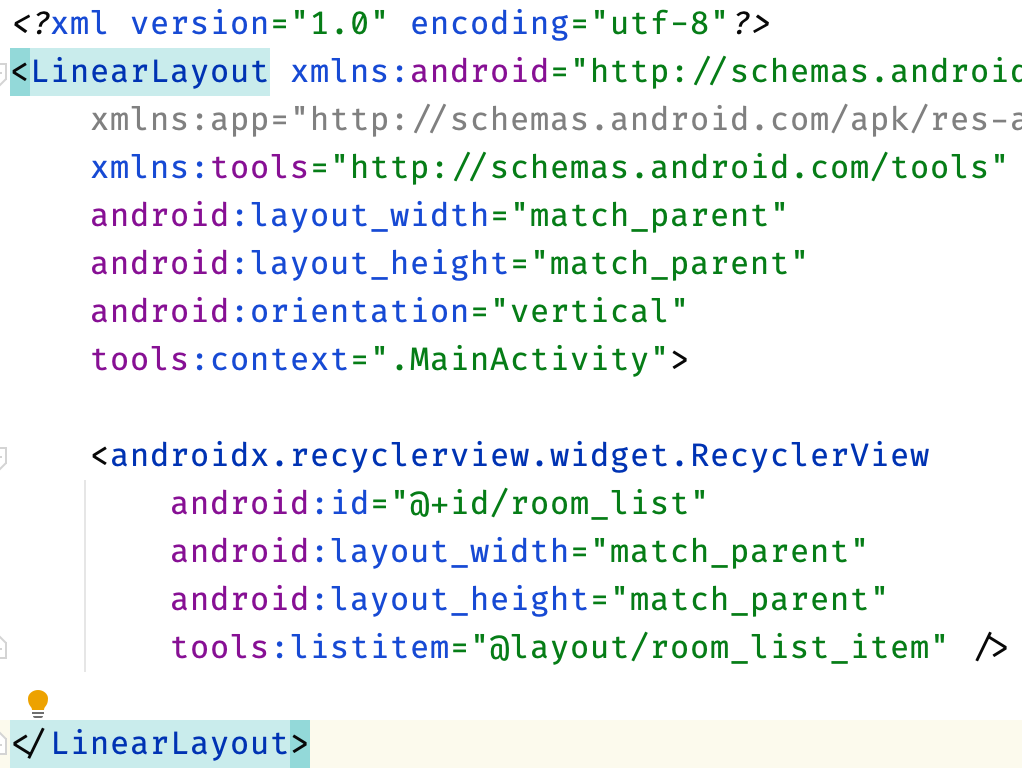
重新截图

图 2 activity\_main.xml

图 2即为首页的布局，一个包含了RecyclerView的LinearLayout。RecyclerView说明这是动态内容显示区域，而具体内容如何展示，还需要有一个列表元素的布局文件来描述，我们将这个展示房间（号）的布局文件命名为room\_list\_item.xml。

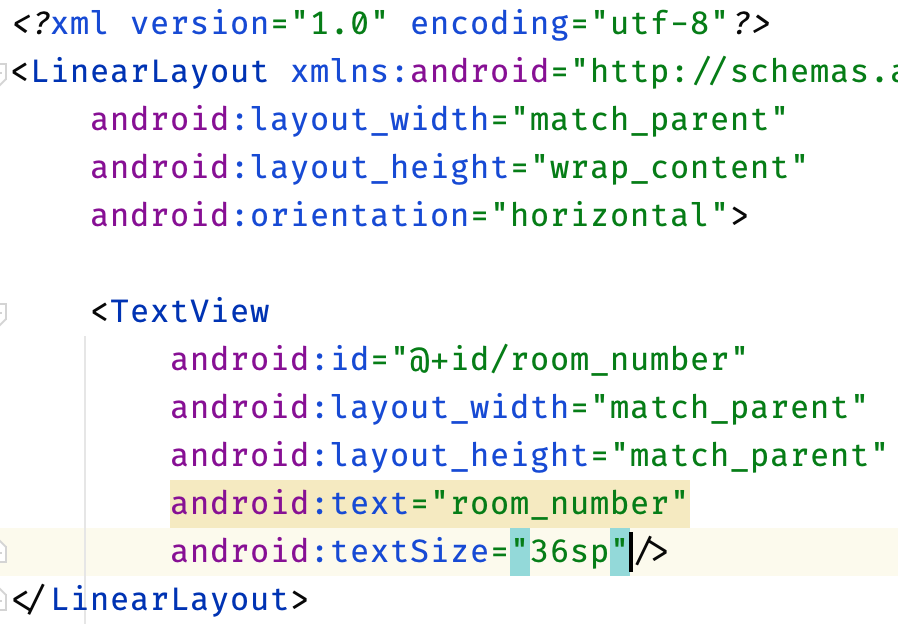
重新截图

图 3 room\_list\_item.xml

图 3即为界面中每个房间（号）如何显示的布局文件。由于我们只有一个字符串即房间号可以显示，布局文件中只需要一个文本框即可。同页面布局文件，可以直接将 TextView 作为文件的根节点，但是为了便于后续扩展，就还是将 TextView 放在布局中。

RecyclerView的设置中使用tools:listitem关联此文件之后，图 2 activity\_main.xml 在设计视图中就显示为如图 4所示。

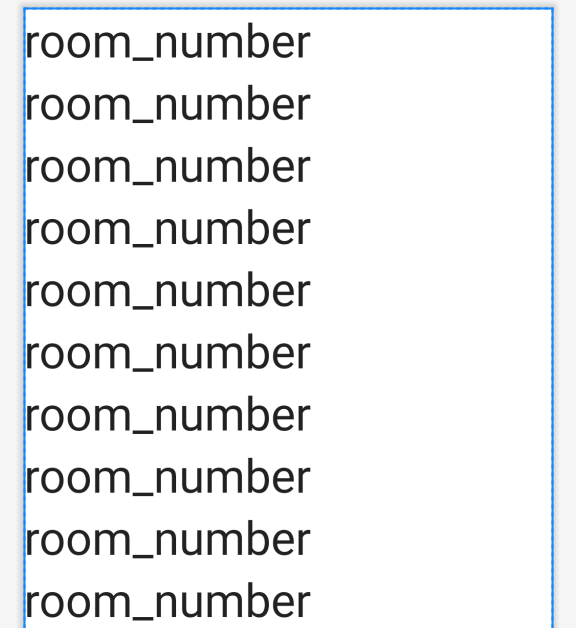
重新截图

图 4 界面布局设计预览

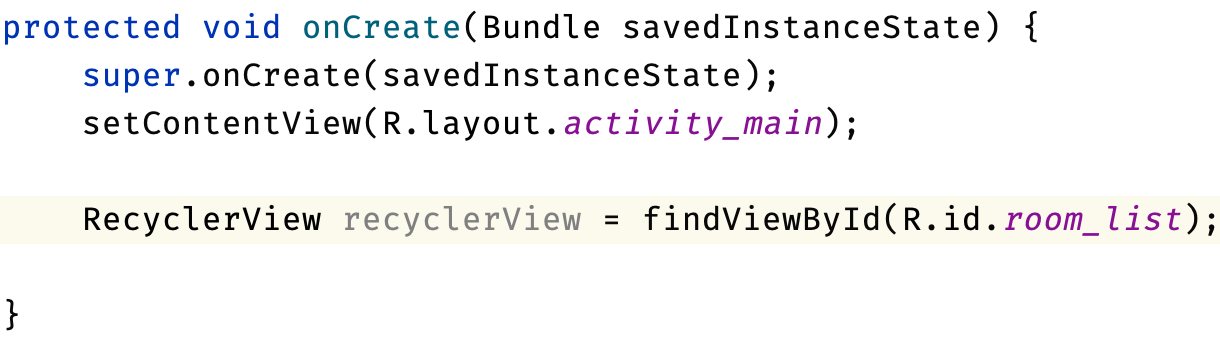
### 逻辑开发

RecyclerView的使用分为几个步骤。首先，通过findViewById获得RecyclerView的引用；然后为RecyclerView设置将房间数据与房间 Layout 绑定的Adapter；将房间数据送进RecyclerView。

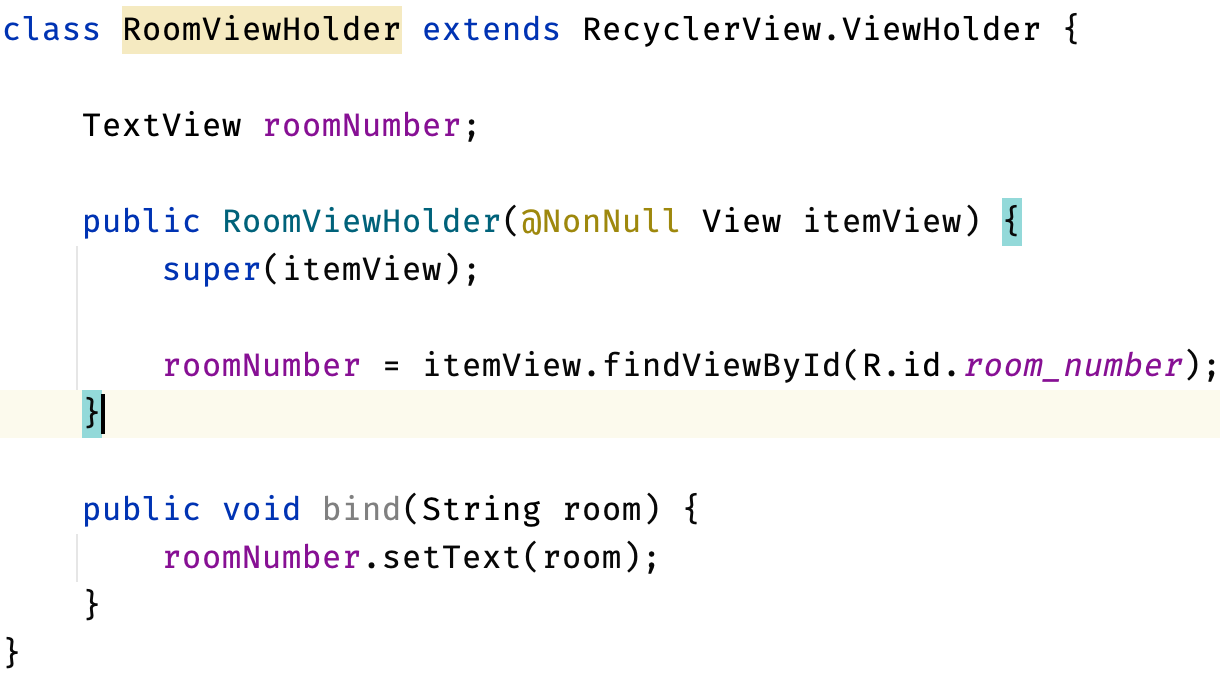
图：

RecyclerView <-> (R.layout.room\_list\_item) RoomAdapter (List<String> roomList ) <-> DataSource (getRoomList)

#### 获得R.id.room\_list的引用



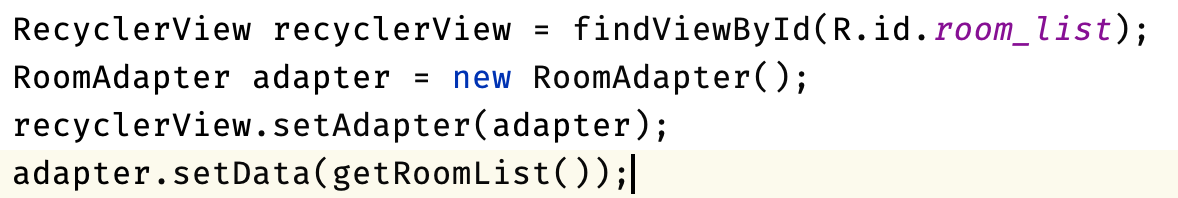
#### 定制 RecyclerView.ViewHolder，即每个元素的布局 View 的引用。



#### 定制RecyclerView.Adapter



#### 在MainActivity中把几个东西串起来。



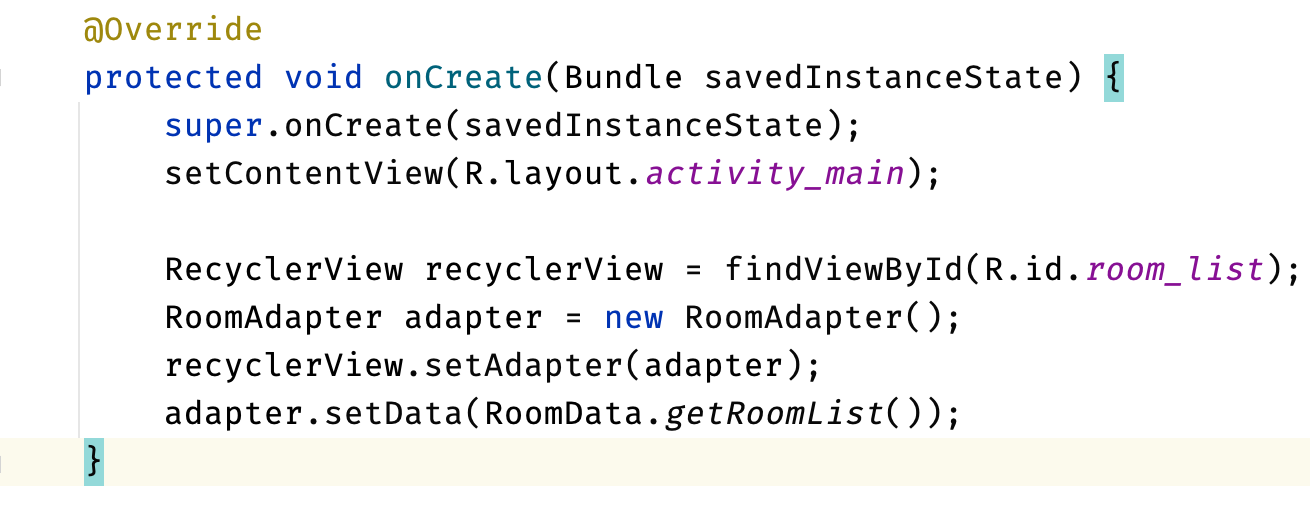
由于是原型，我们并没有实现读取和存储房间数据的细节，只是通过自己创建的方法来提供演示数据。



## 迭代改进：管理界面数据。

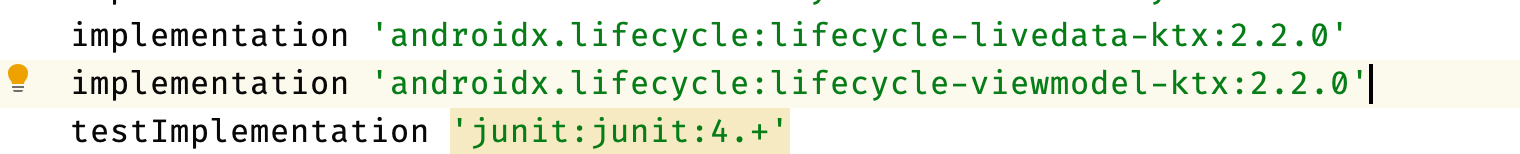
### 使用功能类管理数据

getRoomList方法只是提供演示数据，其功能和应用主逻辑没有关系，也不应该存在于MainActivity中，应该将其抽取出来，放到独立的数据功能类中。





### 使用 ViewModel 来维护界面数据





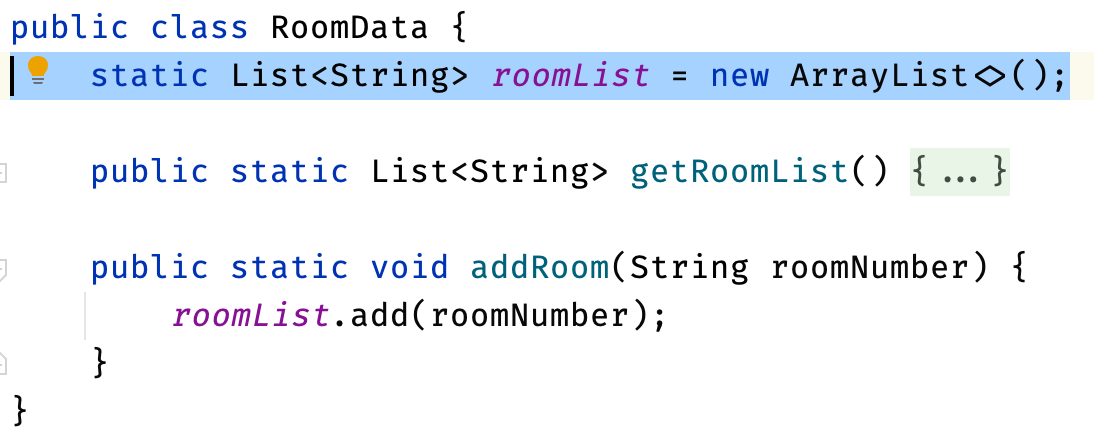


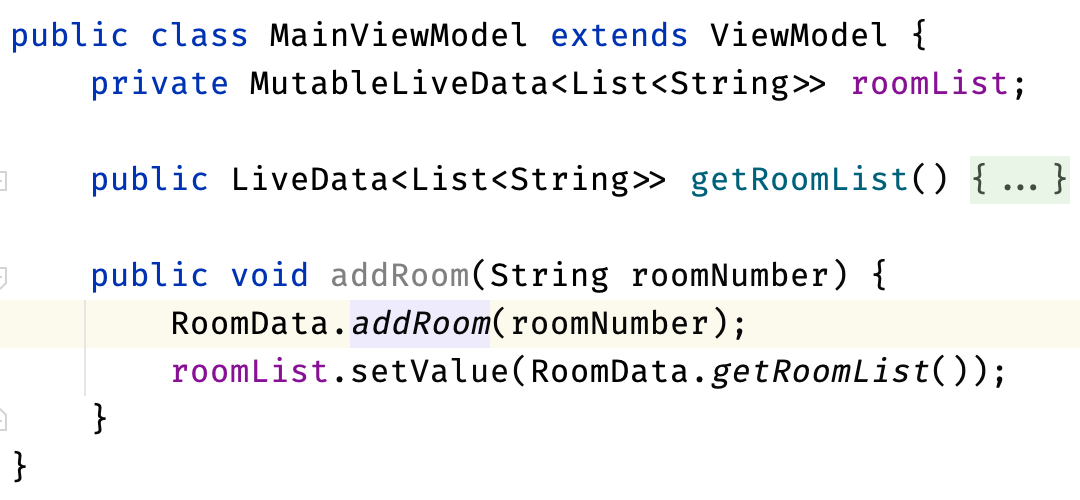
这个lambda 的写法可以进一步优化为：

viewModel.getRoomList().observe(this, adapter::setData);

## 迭代改进：添加房间

### 增加添加房间的方法





### 设计添加房间的界面

#### 直接在现有的界面顶端加上输入框和按钮

