



华中农业大学
HUAZHONG AGRICULTURAL UNIVERSITY

手机平台应用开发

小农夫 Android 应用

专业： 计算机科学与技术专业

姓名： 刘开

学号： 2013310200305

班级： 计科 1403

中国. 武汉

2017 年 6 月

【应用使用说明】

步骤 1(输入账号与密码)

输入账号与密码，设置只能输入 liukai 与 123456 才能进入，后期需要改进，比如增加注册功能。记住密码使用 CheckBox，如果选中该复选框，并且账号与密码均正确，则使用 SharedPreferences 存放以及读取数据，即将“remember_password”设置为 true，并将“account”与“password”对应的值放到 SharedPreferences 文件中并提交。在后期登录时，会自动将账号和密码都设置到文本框中。如果错误则使用 Toast 进行提示。相关代码为类 LoginActivity。相关截图如下：



步骤 2(主界面)

登录成功后进入主界面，在此处可以选择地区+地标+地块+地位增加田地，使用的控件为 Spinner（下拉框），可以使用 xml 文件存放<item>，但这样就固定了无法选择条目。因此使用另一种方法，首先定义一个数组，比如

```
String[] m_Wei = {"前", "后"}; //定义地位数组
```

然后默认选择第一个条目 String strWei = m_Wei[0]; //选择的地位

接着将可选内容与 ArrayAdapter 连接起来，R.layout.spinner_style 是 android 系统自带样式，然后设置下拉列表的风格、添加 SpinnerWei 事件监听，具体代码如下：

```
adapter = new ArrayAdapter<String>(this, R.layout.spinner_style, m_Wei);
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item); //设置下拉列表的风格
spinnerWei = (Spinner) findViewById(R.id.spinner_wei);
spinnerWei.setAdapter(adapter);
//添加 SpinnerWei 事件监听
spinnerWei.setOnItemClickListener(new Spinner.OnItemClickListener() {
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
                                long arg3) {
        strWei = m_Wei[arg2];
    }

    public void onNothingSelected(AdapterView<?> arg0) {
    }
});
```

增加田地后使用 dishesadapter.notifyDataSetChanged() 进行刷新界面。

列表使用 RecyclerView 进行实现，列表显示的内容有田地号、种植的农作物、农作物状态、删除田地几项。在田地没有种植任何农作物时才删除田地，会弹出对话框提示是否继续，相关代码如下：

```
new AlertDialog.Builder(view.getContext()).setTitle("系统提示") //设置对话框标题
    .setMessage("确定要删除吗！") //设置显示的内容
    .setPositiveButton("确定", new DialogInterface.OnClickListener() { //添加确定按钮
        @Override
        public void onClick(DialogInterface dialog, int which) { //确定按钮的响应事件
            mDishesList.remove(position);
            SQLiteDatabase db = databaseHelper.getWritableDatabase();
            db.delete("Dishes", "name = ?", new String[] { dishes.getName() });
            notifyDataSetChanged();
        }
    }).setNegativeButton("取消", new DialogInterface.OnClickListener() { //添加返回按钮
        @Override
        public void onClick(DialogInterface dialog, int which) { //响应事件
```

```
}  
}).show(); //在按钮响应事件中显示此对话框
```

增加的田地会添加到 SQLite 数据库，进入主界面会创建数据库，并且创建表 Dishes（田地农作物表）与 House（仓库表），使用田地名作为 Dishes 表的主键，因此添加已经存在的田地使用 Toast 进行提示，如果存在该数据库则后续将不再进行创建。

```
MyDatabaseHelper databaseHelper = new MyDatabaseHelper(this, "dishesManager.db", null, 1);
```

```
//得到数据库
```

具体使用类 MyDatabaseHelper 进行数据库的创建。

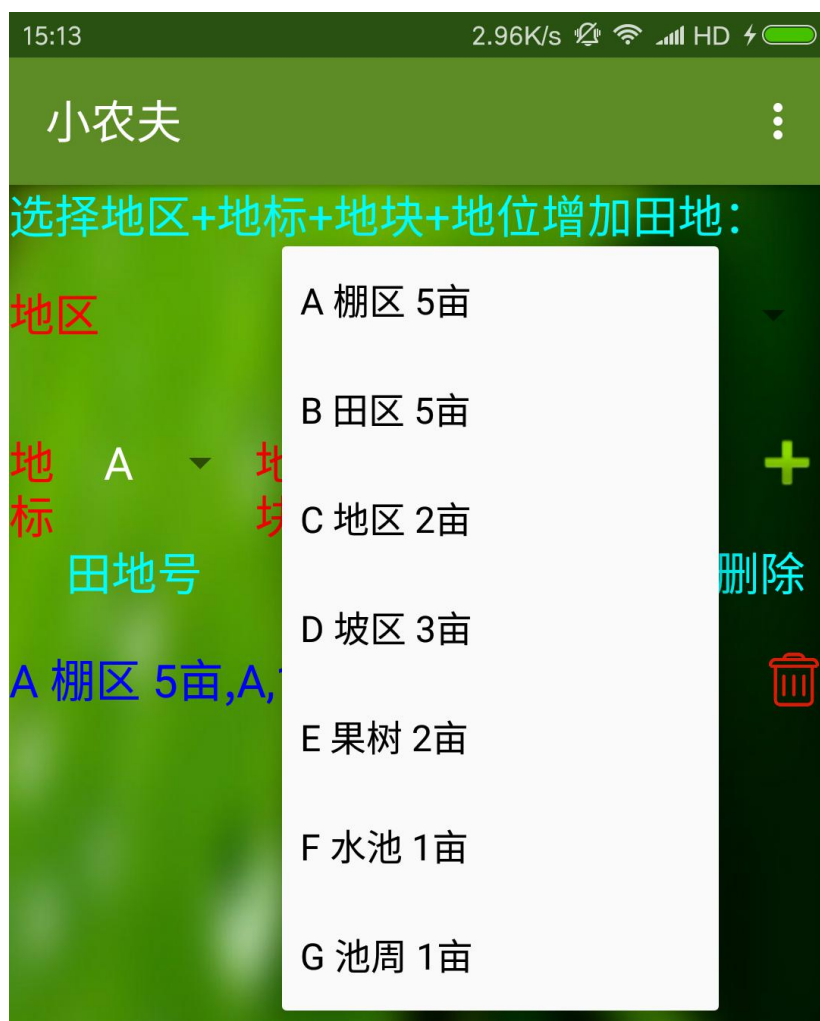
相关截图如下：



系统提示

确定要删除吗！

取消 确定



步骤 3(农作物具体状态)

点击列表的种植项或者状态项进入活动 ScheduleActivity, 用于查看农作物的具体状态, 显示种植面积、种植时间, 以及预计成熟时间, 图片为农作物的照片, 进度条显示成熟进度可以进行种植已经采摘, 默认无任何农作物, 此时只能点击种植按钮, 没有农作物或者农作物没有成熟时点击采摘按钮会进行提示, 相关截图如下:



步骤 4(选择种植的蔬菜)

点击种植按钮会进入活动 `VegetablesActivity`，列表使用 `RecyclerView`，使用下列代码设置为两列，每一项有一张图片和对应输出的名字，可上下滑动进行选择。

```
StaggeredGridLayoutManager layoutManager = new
    StaggeredGridLayoutManager(2, StaggeredGridLayoutManager.VERTICAL); // 设置为两列
recyclerView.setLayoutManager(layoutManager);
```

第一个参数为设置的列数，第二个参数用于设置布局的排列方向。对应的 `activity_vegetables.xml` 文件中通过使用 `<include layout="@layout/title" />` 可将布局栏引入进来，对返回按钮添加返回功能。

```
Button backButton=(Button)findViewById(R.id.title_back);
backButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        onBackPressed();
    }
});
```

使用以下代码可将系统自带的标题栏进行隐藏掉：

```
ActionBar actionBar = getSupportActionBar();
if (actionBar != null) {
    actionBar.hide();
}
```

相关截图如下：



步骤 5(输入种植面积与预计成熟时间)

点击某种蔬菜会先弹出选择对话框，如果点击确定，会弹出输入对话框，提示输入种植面积与预计成熟时间，如果不输入则默认值分别为 1.0 亩和 100 天，输入超过 2 亩也默认为 1 亩。

使用 `android:inputType="numberDecimal"` 限制只能输入小数

使用 `android:inputType="number"` 限制只能输入整数

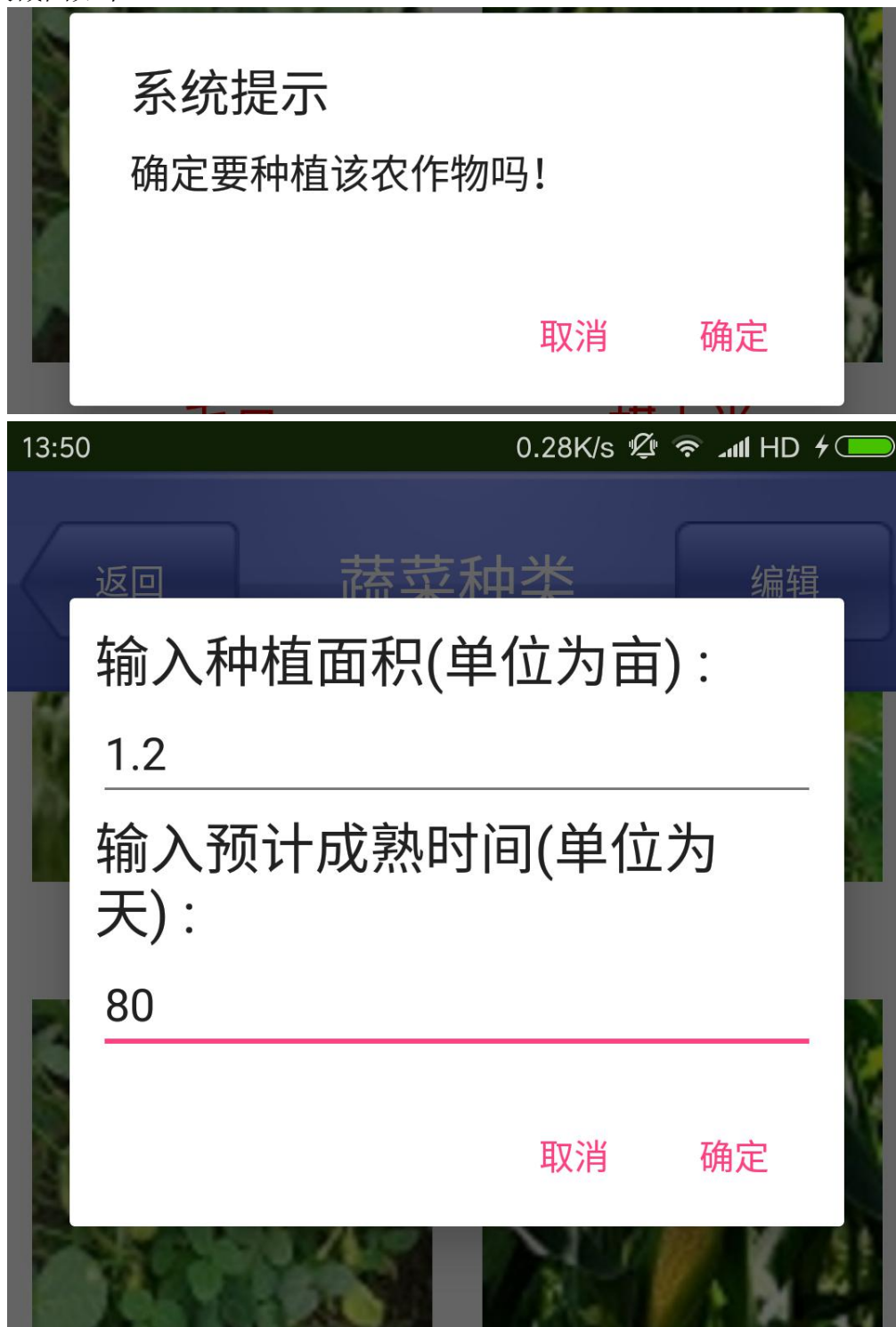
相关代码如下：

// 设置对话框

```
AlertDialogBuilder
    .setCancelable(false)
    .setPositiveButton("确定",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                // 得到输入数据，设置最大输入为2亩
                if (!"".equals(area.getText().toString()) &&
area.getText().toString() != null
                    && Double.parseDouble(area.getText().toString())<=2 &&
Double.parseDouble(area.getText().toString())>0) {
                    areaNumber = Double.parseDouble(area.getText().toString());
                }
                if (!"".equals(time.getText().toString()) &&
time.getText().toString() != null
                    && Integer.parseInt(time.getText().toString())>0) {
                    timeNumber = Integer.parseInt(time.getText().toString());
                }
                SQLiteDatabase db = databaseHelper.getWritableDatabase();
                ContentValues values = new ContentValues();//更新数据

                values.put("crop", vegetable.getName());
                values.put("state", "成长");
                values.put("area", areaNumber);
                values.put("matureTime", timeNumber);
                SimpleDateFormat sf = new SimpleDateFormat("yyyy-MM-dd");
                Date curDate = new Date(System.currentTimeMillis());//获取当前时间
                String str = sf.format(curDate);
                values.put("plantTime", str);
                values.put("image", vegetable.getImageId());//存放图片地址
                db.update("Dishes", values, "name=?", new String[]{name});
                notifyDataSetChanged();
            }
        })
    .setNegativeButton("取消",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                dialog.cancel();
            }
        });
AlertDialog alertDialog = alertDialogBuilder.create();
alertDialog.show();
```


相关截图如下：



步骤 6(更新其他界面)

种植好蔬菜后点击返回后，活动 `ScheduleActivity` 将进行更新，实现过程为在 `ScheduleActivity` 的进行重新初始化界面，将初始化代码放入方法 `init()` 中。

//从另一个活动回来，重新设置，即进行刷新


@Override

```
protected void onResume() {  
    super.onResume();  
    init();  
}
```

此时如果返回活动 MainActivity 也需要进行刷新，将种植项更新为农作物的名字，状态项为成长，使用方法为 MainActivity 进入 ScheduleActivit 便将 MainActivity 进行 finish，然后重新 ScheduleActivit 的 onBackPressed() 方法。另外，为了方便进行测试，将成熟时间改为秒，实际应用时可将相关代码进行注释，更新界面的代码如下：

```
holder.dishesView.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
  
        int position = holder.getAdapterPosition();  
        Dishes dishes = mDishesList.get(position);  
        // Intent intent =new Intent(view.getContext(), ScheduleActivity.class);  
        // view.getContext().startActivity(intent);  
        ScheduleActivity.actionStart(view.getContext(), dishes);  
        ((Activity)context).finish();  
    }  
});  
  
@Override  
public void onBackPressed() {  
    super.onBackPressed();  
    Intent intent = new Intent(ScheduleActivity.this, MainActivity.class);  
    startActivity(intent);  
}
```

相关截图如下：

田地号	种植	状态	删除
A 棚区 5亩,A,1,前	毛豆	成长	

13:51

0.79K/s HD

小农夫



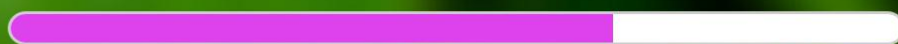
种植面积：1.2亩

种植时间: 2017-06-10

预计成熟时间： 80天



毛豆



种植

采摘

步骤 7(采摘农作物)

当农作物成熟时，进度条会全部变为紫色，然后点击采摘按钮会弹出输入对话框，输入农作物的产量，不输入默认为 1000 公斤，然后重置田地，然后将数据插入到仓库表中，相关代码如下：

// 设置对话框

```
AlertDialogBuilder
    .setCancelable(false)
    .setPositiveButton("确定",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                // 得到数据
                if (!"".equals(yield.getText().toString()) &&
                    yield.getText().toString() != null) {
                    yieldNumber = Integer.parseInt(yield.getText().toString());
                }
                SQLiteDatabase db = databaseHelper.getWritableDatabase();
                ContentValues values = new ContentValues();
                //重置田地
                values.put("crop", "无农作物");
                values.put("state", "无");
                values.put("area", 0);
                values.put("image", 0);
                values.put("matureTime", 0);
                db.update("Dishes", values, "name=?", new String[] {dishes.getName()});

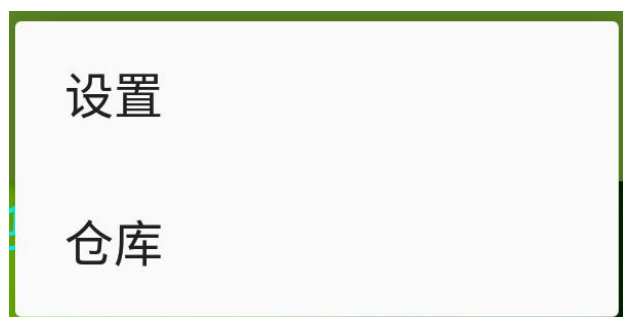
                ContentValues v = new ContentValues();
                // 开始组装数据，插入到仓库表
                v.put("fieldName", dishes.getName());
                v.put("crop", crop);
                v.put("yield", yieldNumber);
                v.put("supply", 0);
                v.put("plantTime", plantTime);
                SimpleDateFormat sf = new SimpleDateFormat("yyyy-MM-dd");
                Date curDate = new Date(System.currentTimeMillis()); //获取当前时间
                String str = sf.format(curDate);
                v.put("pickTime", str);
                v.put("listedTime", "2017-01-01"); //默认
                v.put("offTime", "2017-01-01");
                db.insert("House", null, v); // 插入一条数据
                flag = false;
            }
        })
    .setNegativeButton("取消",
```

```
new DialogInterface.OnClickListener() {  
    public void onClick(DialogInterface dialog, int id) {  
        dialog.cancel();  
    }  
});  
AlertDialog alertDialog = alertDialogBuilder.create();  
alertDialog.show();
```

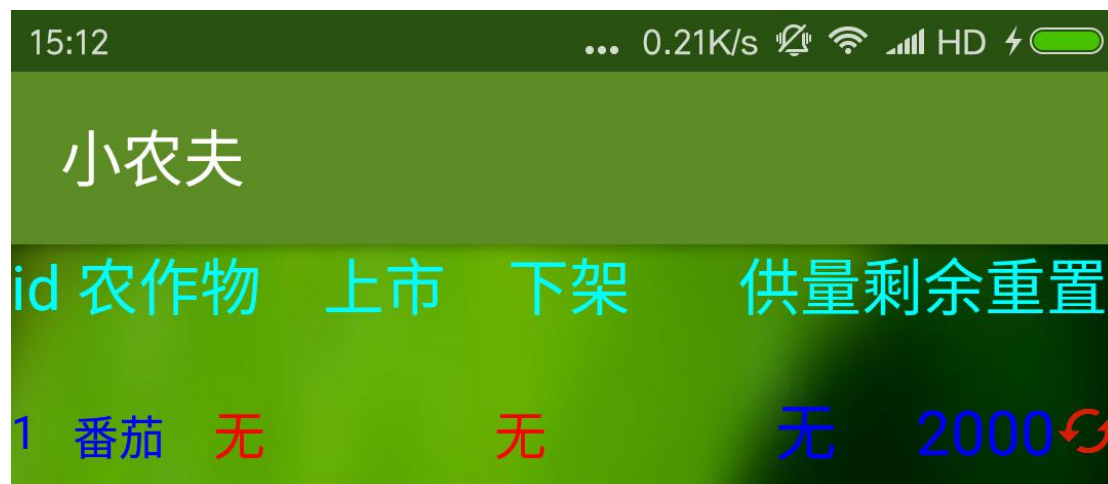
活动 MainActivity 与 ScheduleActivity 均添加了一个菜单项, 点击仓库会进入活动 HouseActivity。



点击右上方的



仓库的列表也是使用 RecyclerView 实现, 项有 id、农作物、上市时间、下架时间、供量、剩余量、重置, House 表使用自动增长的 id 作为主键。相关截图如下:



步骤 8(仓库农作物上市与下架)

点击 HouseActivity 中列表的每一个 Item 的上市项, 会弹出输入对话框输入上市的供量, 不输入默认为剩余量一半, 已经上市过再一次点击将使用 Toast 进行提示。点击下架项, 会弹出输入对话框输入本次上市后的剩余量, 不输入默认为 0。点击后列表更新的方法为在 HouseActivity 定义一个静态方法进行修改,

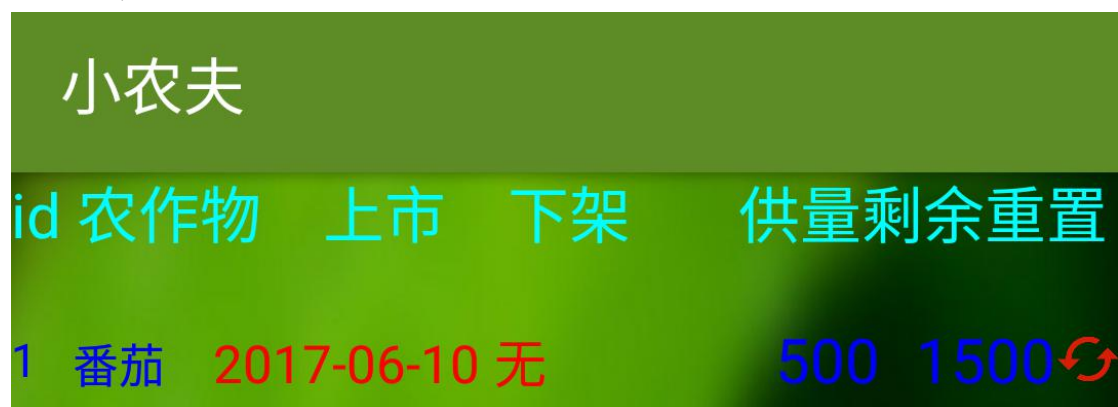
在上市按钮的时间监听中进行调用，主要代码如下：

```
SQLiteDatabase db = databaseHelper.getWritableDatabase();
ContentValues values = new ContentValues();
SimpleDateFormat sf = new SimpleDateFormat("yyyy-MM-dd");
Date curDate = new Date(System.currentTimeMillis()); //获取当前时间
String str = sf.format(curDate);
values.put("supply", supplyNumber);
values.put("yield", (house.getYield() - supplyNumber));
values.put("listedTime", str);
db.update("House", values, "id=?", new String[] {String.valueOf(house.getId())});
init();
houseAdapter.notifyDataSetChanged();

//点击上市
holder.listedTime.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        final int position = holder.getAdapterPosition();
        final House house = mHouseList.get(position);
        if (!"2017-01-01".equals(house.get ListedTime())) {
            Toast.makeText(context, "已经上市，下架后才能再一次上市",
                Toast.LENGTH_SHORT).show();
        } else {
            HouseActivity.myLDialog(context, house);
        }
    }
});
```

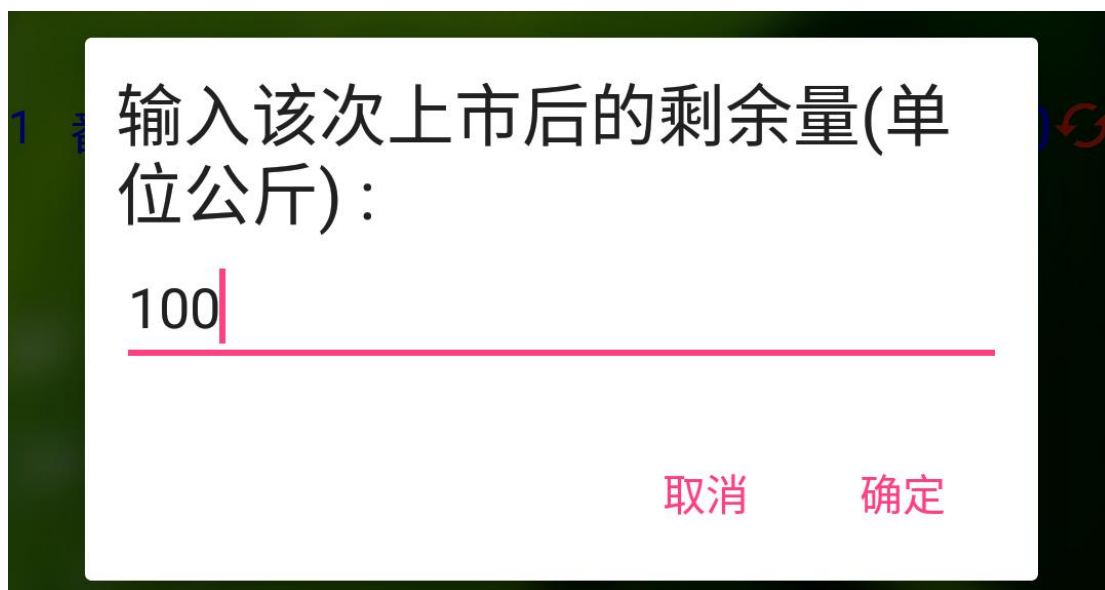
相关截图如下：

点击上市后列表更新：



id	农作物	上市	下架	供量	剩余	重置
1	番茄	2017-06-10	无	500	1500	↻

点击下架后输入剩余量，然后列表自动更新：



id	农作物	上市	下架	供量	剩余重置
1	番茄	2017-06-10	2017-06-10	0	1600

点击重置后列表自动更新:

id	农作物	上市	下架	供量	剩余重置
1	番茄	无	无	无	1600

步骤 9(补充：界面滑动介绍)

在 MainActivity 中可以添加很多田地，进行统一管理，上下滑动查看信息，对于 HouseActivity 中列表 Item，点击除上市、下架、重置项，点击其他项均会使用 Toast 显示田地名、农作物名，以及采摘时间，相关截图如下：

15:170.21K/sHD

小农夫

选择地区+地标+地块+地位增加田地:

地区A 棚区 5亩

地标A

地块1

地位前

+

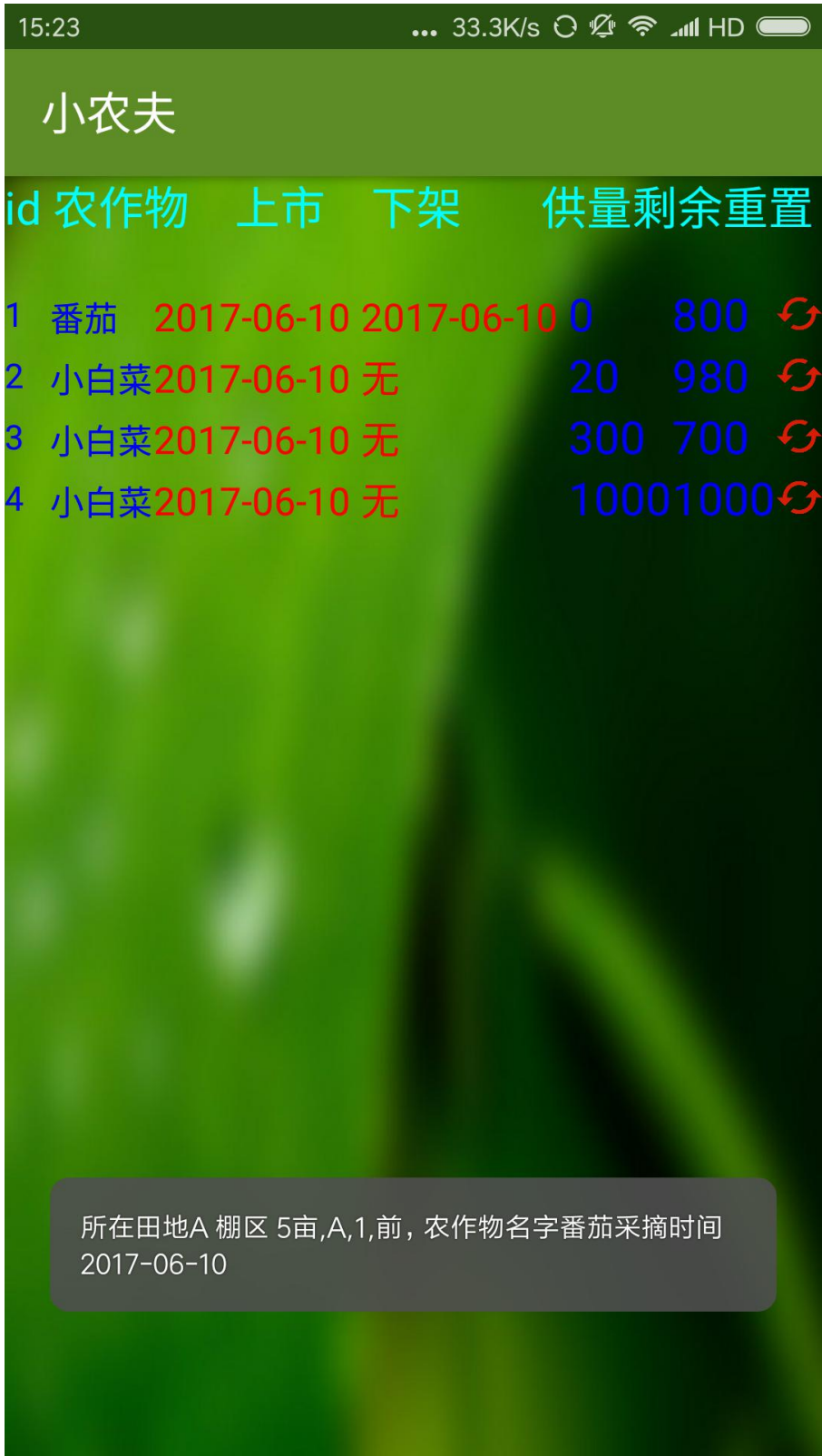
田地号

种植

状态

删除

A 棚区 5亩,A,1,前	小白菜	成长	
E 果树 2亩,B,1,后	小白菜	成长	
D 坡区 3亩,B,1,后	无农作物	无	
C 地区 2亩,B,1,后	育苗	成长	
C 地区 2亩,B,1,前	小白菜	成长	
C 地区 2亩,B,3,前	辣椒	成长	
F 水池 1亩,B,3,前	生菜	成长	
F 水池 1亩,B,3,后	牛心包菜	成长	
F 水池 1亩,B,2,后	牛心包菜	成长	
G 池周 1亩,B,2,后	香菜	成长	
G 池周 1亩,D,2,后	生菜	成长	
C 地区 2亩,D,2,后	小白菜	成长	
C 地区 2亩,D,2,前	茼蒿	成长	
C 地区 2亩,C,2,前	红苋菜	成长	



【项目总结】

通过本次小农夫项目开发实验，使我对于手机平台应用开发的部分知识有了更多的了解，动手编程遇到了问题并查询资料学习到了更多技巧，比如在布局设计时嵌套太多，在使用 LinearLayout 布局时没有是全部布局均加上具体的 orientation，造成界面无法正常显示，还有建立在数据库时，已经存在数据库的情况下，增加表的项时出现了问题，解决办法是将原有数据库删除，可以将模拟器或者手机上的开发应用删掉后，重新安装。还有一个较大的问题就是更改数据库后如何刷新页面，解决方法为步骤 6 的介绍。通过本次应用开发，也学习到了一些老师未讲到的其他有关知识，比如 Spinner（下拉框）的使用、输入对话框的使用等。这些经历让我对于 Android 应用开发有了更详细的认识，是以后学习十分宝贵的经验。项目也存在一些不足之处，比如田地与蔬菜的重新设置并没有实现，界面也不是太美观，后续还需要继续进行完善。