# 智慧牧场牧场主APP移动端详细设计报告(安卓版)

## 软件功能说明

##### 登录功能

##### 忘记登录密码，重置登录密码功能

##### 登记/接糕/打疫苗，扫描硬件设备二维码，对牲畜进行拍照，图片上传服务器，对牲畜种类进行选择，重量进行选择，年龄进行选择，最后将牲畜信息提交给服务器，完成登记牲畜。

##### 剪毛，对牲畜剪毛后进行拍照，然后将图片提交服务器。

##### 生活照，对牲畜日常生活进行拍照，然后将图片上传至服务器。

##### 工具，机械工具，可以根据当前时间添加相关机械工具，可以对工具数量，类型进行选择。在工具界面生成根据时间顺序添加的工具列表。

##### 草料，和工具功能类似，可以根据当前时间添加相关草料，可以对草料重量，类型进行选择。在草料界面生成根据时间顺序添加的草料列表。

##### 我的牧场，点击我的牧场进入到牧场详情页面，会有牧场详情。点击编辑按钮可以对牧场信息进行编辑，完成后，可以将牧场最新信息上传服务器，信息与服务器进行同步。

##### 牧群页面里，会有牲畜种类和牲畜数量，点击相关种类，你可以看到相关种类列表信息。点击具体条目，会进入到牲畜详情页面。牲畜详情，里面有品种名称、设备号、年龄、一般寿命、牧场、发布时间、牲畜详细介绍、牧场位置（显示地图和牲畜坐标点）、牲畜电子档案。

##### 认领页面，发布认领的牲畜会显示在此页面。点击倒三角型图标可以选择显示全部牲畜，显示未认领牲畜，显示已认领牲畜。每个条目具体信息有牲畜图片，牲畜名称、设备号、年龄、牧场、认领时间。未认领的牲畜信息可以被删除，并且同步到服务器。已认领的牲畜信息不能被删除。已认领的牲畜，后台会自动生成支付订单号，当为已支付时，界面会出现支付确认按钮。点击确认按钮，完成确认支付信息后，再回到认领界面，此订单就完成了交易，界面的相关条目就显示交易成功。

##### 认领详情，和牲畜详情类似，里面有品种名称、设备号、年龄、一般寿命、牧场、发布时间、牲畜详细介绍、牧场位置（显示地图和牲畜坐标点）、牲畜电子档案。

##### 发布牲畜功能，扫描硬件设备二维码，拍照牲畜照片，上传服务器，选择牲畜种类，牲畜重量，年龄，点击确定，发布完成回到主页面，就可以看到刚发布的牲畜了。

##### 我的界面，点击头像，会进入到个人详情页面。点击头像可以重新拍照或者选择本地的照片进行上传服务器，进行头像更新。名字可以修改，性别可以修改，手机号可以修改，并且同步更新到服务器。

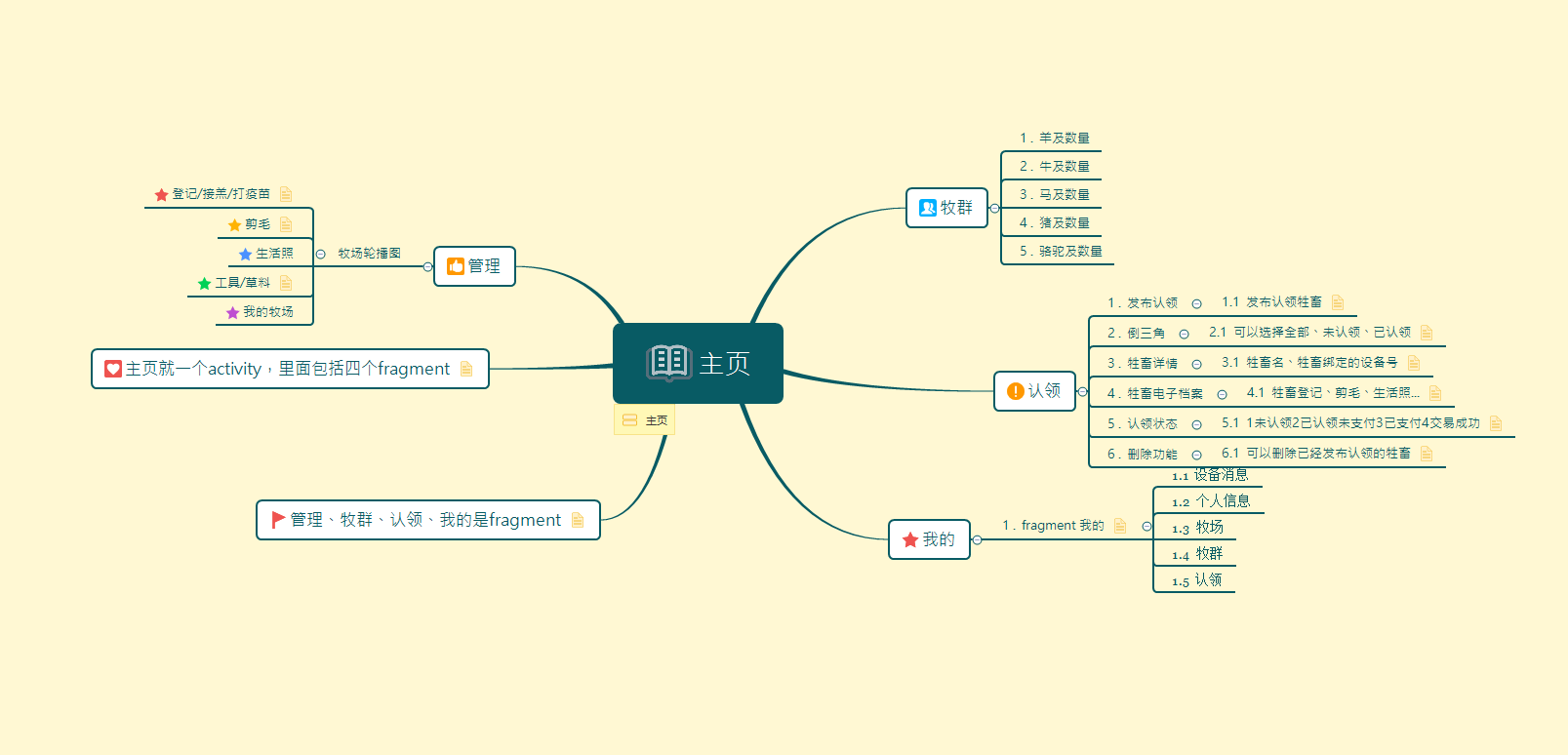
##### 版本检测功能，会显示当前版本号，点击此按钮会检测服务器的最新版本。如果有新版，进行升级。没有，则提示已经是最新版本了。

##### 退出功能，账号退出，然后跳转到登录界面。

##### 设备消息，如果有相关硬件设备电量不足，会将消息显示在此，提示牧场主，哪些设备需要进行充电。继续往下拉，会显示，认领人的手机号和相关牲畜设备号，认领人要干嘛？认领在请求 拍照或者请求录像。点击去拍照按钮，是拍照上传服务器。点击去录像按钮，是录制视频上传服务器。上传的照片和视频 可以在牲畜电子档案里面查看。

## 软件框架（哪些子模块和子模块接口）及业务流程说明

### 1、软件框架



### 业务流程



## 功能模块说明：功能描述、模块/函数接口、模块内部流程、伪代码实现

##### 登记/接羔/打疫苗，扫描二维码，扫描后，拍照，拍照后，把照片上传服务器，选择牲畜种类，选择牲畜类型，选择重量，选择年龄，最后将牲畜信息，上传服务器。

startScanActivity();

openCamera();

UploadFile();

##### 剪毛，对牲畜剪毛后进行拍照，然后将图片提交服务器。

startScanActivity();

openCamera();

UploadFile();

##### 生活照，对牲畜日常生活进行拍照，然后将图片上传至服务器。

openCamera();

UploadFile();

以上三个功能在一个类里面进行实现

package com.jinkun\_innovation.pastureland.ui.fragment;

import android.app.Activity;

import android.content.Intent;

import android.content.res.AssetFileDescriptor;

import android.net.Uri;

import android.os.Build;

import android.os.Bundle;

import android.os.Environment;

import android.provider.MediaStore;

import android.support.annotation.Nullable;

import android.support.v4.app.Fragment;

import android.support.v4.content.FileProvider;

import android.text.TextUtils;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.LinearLayout;

import android.widget.Toast;

import com.daimajia.slider.library.SliderLayout;

import com.daimajia.slider.library.SliderTypes.BaseSliderView;

import com.daimajia.slider.library.SliderTypes.TextSliderView;

import com.google.gson.Gson;

import com.google.zxing.client.android.CaptureActivity2;

import com.jinkun\_innovation.pastureland.R;

import com.jinkun\_innovation.pastureland.bean.IsBinded;

import com.jinkun\_innovation.pastureland.bean.LoginSuccess;

import com.jinkun\_innovation.pastureland.common.Constants;

import com.jinkun\_innovation.pastureland.ui.GrassActivity;

import com.jinkun\_innovation.pastureland.ui.RegisterActivity;

import com.jinkun\_innovation.pastureland.ui.ToolsActivity;

import com.jinkun\_innovation.pastureland.ui.UpLoadActivity;

import com.jinkun\_innovation.pastureland.ui.activity.Muchang2Activity;

import com.jinkun\_innovation.pastureland.utilcode.util.FileUtils;

import com.jinkun\_innovation.pastureland.utilcode.util.LogUtils;

import com.jinkun\_innovation.pastureland.utilcode.util.TimeUtils;

import com.jinkun\_innovation.pastureland.utilcode.util.ToastUtils;

import com.jinkun\_innovation.pastureland.utils.PrefUtils;

import com.jinkun\_innovation.pastureland.utils.StrLengthUtil;

import com.lzy.okgo.OkGo;

import com.lzy.okgo.callback.StringCallback;

import com.lzy.okgo.model.Response;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.util.UUID;

import butterknife.BindView;

import butterknife.ButterKnife;

import butterknife.OnClick;

import butterknife.Unbinder;

import cn.pedant.SweetAlert.SweetAlertDialog;

/\*\*

\* Created by Guan on 2018/3/15.

\*/

public class ManagerFragment extends Fragment {

private static final String TAG1 = ManagerFragment.class.getSimpleName();

@BindView(R.id.llRegister)

LinearLayout llRegister;

@BindView(R.id.llJianMao)

LinearLayout llJianMao;

@BindView(R.id.llLifePhoto)

LinearLayout llLifePhoto;

@BindView(R.id.llTools)

LinearLayout llTools;

@BindView(R.id.llGrass)

LinearLayout llGrass;

@BindView(R.id.llMyMuChang)

LinearLayout llMyMuChang;

Unbinder unbinder;

private static final int REQUEST\_CODE\_ASK\_MULTIPLE\_PERMISSIONS = 3;//权限请求

private static final int SCAN\_REQUEST\_CODE = 100;

private static final int CAMERA\_ACTIVITY = 6; //视频

private String imagePath;

private File photoFile;

private Uri imageUri;//原图保存地址

private static final int REQUEST\_CAPTURE = 2; //拍照

private int checkedItem = 0;

private String scanMessage;

private SliderLayout mSliderShow;

private LoginSuccess mLoginSuccess;

private String mLogin\_success;

private String mUsername;

@Nullable

@Override

public View onCreateView(LayoutInflater inflater,

@Nullable ViewGroup container,

@Nullable Bundle savedInstanceState) {

View view = View.inflate(getActivity(), R.layout.fragment\_manager, null);

unbinder = ButterKnife.bind(this, view);

mLogin\_success = PrefUtils.getString(getActivity(), "login\_success", null);

Gson gson = new Gson();

mLoginSuccess = gson.fromJson(mLogin\_success, LoginSuccess.class);

mUsername = PrefUtils.getString(getActivity(), "username", null);

mSliderShow = view.findViewById(R.id.slider);

TextSliderView textSliderView = new TextSliderView(getActivity());

textSliderView

.description("智慧牧场")

.setScaleType(BaseSliderView.ScaleType.Fit)//图片缩放类型

.image("http://s16.sinaimg.cn/large/4c74484dx738034dca2ef&690");

textSliderView.setOnSliderClickListener(new BaseSliderView.OnSliderClickListener() {

@Override

public void onSliderClick(BaseSliderView slider) {

startActivity(new Intent(getActivity(),Muchang2Activity.class));

}

});

TextSliderView textSliderView1 = new TextSliderView(getActivity());

textSliderView1

.description("科技点亮牧场")

.image("https://goss2.vcg.com/creative/vcg/800/version23/VCG41544531487.jpg");

textSliderView1.setOnSliderClickListener(new BaseSliderView.OnSliderClickListener() {

@Override

public void onSliderClick(BaseSliderView slider) {

startActivity(new Intent(getActivity(),Muchang2Activity.class));

}

});

TextSliderView textSliderView2 = new TextSliderView(getActivity());

textSliderView2

.description("金坤技术")

.image("https://goss1.vcg.com/creative/vcg/800/version23/VCG41544521221.jpg");

textSliderView2.setOnSliderClickListener(new BaseSliderView.OnSliderClickListener() {

@Override

public void onSliderClick(BaseSliderView slider) {

startActivity(new Intent(getActivity(), Muchang2Activity.class));

}

});

mSliderShow.addSlider(textSliderView);

mSliderShow.addSlider(textSliderView1);

mSliderShow.addSlider(textSliderView2);

mSliderShow.setPresetTransformer(SliderLayout.Transformer.RotateUp);

LinearLayout llMyMuChang = (LinearLayout) view.findViewById(R.id.llMyMuChang);

llMyMuChang.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

startActivity(new Intent(getActivity(), Muchang2Activity.class));

}

});

return view;

}

@Override

public void onDestroyView() {

super.onDestroyView();

unbinder.unbind();

}

@OnClick({R.id.llRegister, R.id.llJianMao, R.id.llLifePhoto,

R.id.llTools, R.id.llGrass, R.id.llMyMuChang})

public void onViewClicked(View view) {

switch (view.getId()) {

case R.id.llRegister:

checkedItem = 2;

startScanActivity();

break;

case R.id.llJianMao:

checkedItem = 3;

startScanActivity();

break;

case R.id.llLifePhoto:

checkedItem = 0;

openCamera();

break;

case R.id.llTools:

startActivity(new Intent(getActivity(), ToolsActivity.class));

break;

case R.id.llGrass:

startActivity(new Intent(getActivity(), GrassActivity.class));

break;

case R.id.llMyMuChang:

// startActivity(new Intent(getActivity(), LianJiangPasturelandActivity.class));

break;

}

}

// a为原字符串，b为要插入的字符串，t为插入位置

public String Stringinsert(String a, String b, int t) {

return a.substring(0, t) + b + a.substring(t, a.length());

}

@Override

public void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (resultCode == Activity.RESULT\_OK) {

switch (requestCode) {

case SCAN\_REQUEST\_CODE:

String isbn = data.getStringExtra("CaptureIsbn");

if (!TextUtils.isEmpty(isbn)) {

LogUtils.e(isbn);

Toast.makeText(getActivity(), "解析到的内容为" + isbn, Toast.LENGTH\_LONG).show();

if (StrLengthUtil.length(isbn) == 16) {

scanMessage = isbn;

register(isbn);

} else if (StrLengthUtil.length(isbn) == 15) {

String str = Stringinsert(isbn, "1", 7);

Log.d(TAG1, "15位isbn=" + str);

Log.d(TAG1, "新的长度" + StrLengthUtil.length(str));

scanMessage = str;

register(str);

} else {

ToastUtils.showShort("设备号必须是16位数字");

}

}

break;

case CAMERA\_ACTIVITY:

try {

AssetFileDescriptor videoAsset = getActivity().getContentResolver()

.openAssetFileDescriptor(data.getData(), "r");

FileInputStream fis = videoAsset.createInputStream();

File tmpFile = new File(Environment.getExternalStorageDirectory(),

TimeUtils.getNowString() + ".mp4");

FileOutputStream fos = new FileOutputStream(tmpFile);

byte[] buf = new byte[1024];

int len;

while ((len = fis.read(buf)) > 0) {

fos.write(buf, 0, len);

}

fis.close();

fos.close();

Log.d("ManagerFragment", tmpFile.getAbsolutePath());

PrefUtils.setString(getActivity(), "v1", tmpFile.getAbsolutePath());

} catch (IOException io\_e) {

// TODO: handle error

}

break;

case REQUEST\_CAPTURE://拍照

LogUtils.e(imageUri);

Intent intent = new Intent();

intent.putExtra(getString(R.string.checked\_Item), checkedItem);

intent.putExtra(getString(R.string.img\_Url), photoFile.getAbsolutePath());

intent.putExtra(getString(R.string.scan\_Message), scanMessage);

switch (checkedItem) {

case 2:

intent.setClass(getActivity(), RegisterActivity.class);

break;

case 3:

//剪毛

intent.setClass(getActivity(), UpLoadActivity.class);

case 0:

intent.setClass(getActivity(), UpLoadActivity.class);

break;

}

startActivity(intent);

break;

}

}

}

private void register(final String isbn) {

switch (checkedItem) {

case 2:

/\*Intent intent = new Intent(getActivity()

, RegisterActivity.class);

intent.putExtra(getString(R.string.scan\_Message),

scanMessage);

startActivity(intent);\*/

//接羔

//判断设备是否被绑定

OkGo.<String>post(Constants.ISDEVICEBINDED)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername) //用户手机号

.params("deviceNO", isbn)

.params("ranchID", mLoginSuccess.getRanchID())

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String result = response.body().toString();

Gson gson = new Gson();

IsBinded isBinded = gson.fromJson(result, IsBinded.class);

String code = isBinded.getCode();

boolean msg = isBinded.isMsg();

String msg1 = isBinded.getMsg1();

if (!msg && code.contains("error")) {

//提示被其它牧场主绑定了，跳回主页面

new SweetAlertDialog(getActivity(), SweetAlertDialog.WARNING\_TYPE)

.setTitleText("被其它牧场主绑定了")

.setContentText("无法登记")

.show();

} else if (code.contains("true")) {

//提示此牲畜已发布认领，不可重新发布

new SweetAlertDialog(getActivity(), SweetAlertDialog.WARNING\_TYPE)

.setTitleText("此牲畜已发布认领")

.setContentText("无法重新登记")

.show();

} else if (msg && code.contains("success")) {

//提示设备已经被登记，提示是否解绑

new SweetAlertDialog(getActivity(), SweetAlertDialog.WARNING\_TYPE)

.setTitleText("设备已经被登记，是否解绑?")

.setContentText("解绑重新登记")

.setConfirmText("是")

.setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener() {

@Override

public void onClick(SweetAlertDialog sDialog) {

sDialog.cancel();

Intent intent = new Intent(getActivity()

, RegisterActivity.class);

intent.putExtra(getString(R.string.scan\_Message),

scanMessage);

startActivity(intent);

}

})

.setCancelText("否")

.setCancelClickListener(new SweetAlertDialog.OnSweetClickListener() {

@Override

public void onClick(SweetAlertDialog sDialog) {

sDialog.cancel();

}

})

.show();

} else {

// 未登记

Intent intent = new Intent(getActivity()

, RegisterActivity.class);

intent.putExtra(getString(R.string.scan\_Message),

scanMessage);

startActivity(intent);

}

}

});

break;

case 3:

//剪毛

//判断设备是否被绑定

OkGo.<String>post(Constants.ISDEVICEBINDED)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername) //用户手机号

.params("deviceNO", isbn)

.params("ranchID", mLoginSuccess.getRanchID())

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String result = response.body().toString();

Log.d(TAG1, result);

if (result.contains("true")) {

//已绑定

openCamera();

} else {

//未绑定

new SweetAlertDialog(getActivity())

.setTitleText("未登记设备，不能剪毛!")

.show();

}

}

});

break;

case 0:

//拍照

break;

}

}

private void openCamera() {

String rootDir = "/Pastureland/photo";

FileUtils.createOrExistsDir(new File(Environment.getExternalStorageDirectory(), rootDir));

File file = new File(Environment.getExternalStorageDirectory(), rootDir);

photoFile = new File(file, UUID.randomUUID().toString() + ".jpg");

FileUtils.createOrExistsFile(photoFile);

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.N) {

imageUri = FileProvider.getUriForFile(getActivity(),

"com.jinkun\_innovation.pastureland.fileProvider",

photoFile);//通过FileProvider创建一个content类型的Uri

} else {

imageUri = Uri.fromFile(photoFile);

}

Intent intent = new Intent();

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.N) {

intent.addFlags(Intent.FLAG\_GRANT\_READ\_URI\_PERMISSION); //添加这一句表示对目标应用临时授权该Uri所代表的文件

}

intent.setAction(MediaStore.ACTION\_IMAGE\_CAPTURE);//设置Action为拍照

intent.putExtra(MediaStore.EXTRA\_OUTPUT, imageUri);//将拍取的照片保存到指定URI

startActivityForResult(intent, REQUEST\_CAPTURE);

}

private void startScanActivity() {

Intent intent = new Intent(getActivity(), CaptureActivity2.class);

intent.putExtra(CaptureActivity2.USE\_DEFUALT\_ISBN\_ACTIVITY, true);

intent.putExtra("inputUnable", 0);

intent.putExtra("lightUnable", 1);

intent.putExtra("albumUnable", 1);

intent.putExtra("cancleUnable", 1);

startActivityForResult(intent, SCAN\_REQUEST\_CODE);

}

}

##### 4、认领页面，发布认领的牲畜会显示在此页面。点击倒三角型图标可以选择显示全部牲畜，显示未认领牲畜，显示已认领牲畜。每个条目具体信息有牲畜图片，牲畜名称、设备号、年龄、牧场、认领时间。未认领的牲畜信息可以被删除，并且同步到服务器。已认领的牲畜信息不能被删除。已认领的牲畜，后台会自动生成支付订单号，当为已支付时，界面会出现支付确认按钮。点击确认按钮，完成确认支付信息后，再回到认领界面，此订单就完成了交易，界面的相关条目就显示交易成功。

package com.jinkun\_innovation.pastureland.ui.fragment;

import android.app.Activity;

import android.content.Intent;

import android.net.Uri;

import android.os.Bundle;

import android.support.annotation.NonNull;

import android.support.annotation.Nullable;

import android.support.v4.app.Fragment;

import android.support.v7.widget.LinearLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.text.TextUtils;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.AdapterView;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.Spinner;

import android.widget.TextView;

import com.facebook.drawee.view.SimpleDraweeView;

import com.google.gson.Gson;

import com.google.zxing.client.android.CaptureActivity2;

import com.jinkun\_innovation.pastureland.R;

import com.jinkun\_innovation.pastureland.bean.LoginSuccess;

import com.jinkun\_innovation.pastureland.bean.RenLing;

import com.jinkun\_innovation.pastureland.bean.SelectLivestock;

import com.jinkun\_innovation.pastureland.common.Constants;

import com.jinkun\_innovation.pastureland.ui.PublishClaimActivity;

import com.jinkun\_innovation.pastureland.ui.RenlingDetailActivity;

import com.jinkun\_innovation.pastureland.ui.activity.PayConfirmActivity;

import com.jinkun\_innovation.pastureland.utilcode.constant.TimeConstants;

import com.jinkun\_innovation.pastureland.utilcode.util.TimeUtils;

import com.jinkun\_innovation.pastureland.utilcode.util.ToastUtils;

import com.jinkun\_innovation.pastureland.utils.PrefUtils;

import com.jinkun\_innovation.pastureland.utils.StrLengthUtil;

import com.lzy.okgo.OkGo;

import com.lzy.okgo.callback.StringCallback;

import com.lzy.okgo.model.Response;

import com.scwang.smartrefresh.layout.api.RefreshLayout;

import com.scwang.smartrefresh.layout.listener.OnLoadMoreListener;

import com.scwang.smartrefresh.layout.listener.OnRefreshListener;

import java.util.List;

import cn.pedant.SweetAlert.SweetAlertDialog;

/\*\*

\* Created by Guan on 2018/3/23.

\*/

public class RenlingFragment1 extends Fragment {

private static final String TAG1 = RenlingFragment1.class.getSimpleName();

String mLogin\_success;

LoginSuccess mLoginSuccess;

String mUsername;

private static final int SCAN\_REQUEST\_CODE = 100;

private int checkedItem = 0;

private String scanMessage;

int index = 2;

private List<RenLing.LivestockListBean> mLivestockList;

private String mDeviceNo;

private void startScanActivity() {

Intent intent = new Intent(getActivity(), CaptureActivity2.class);

intent.putExtra(CaptureActivity2.USE\_DEFUALT\_ISBN\_ACTIVITY, true);

intent.putExtra("inputUnable", 0);

intent.putExtra("lightUnable", 1);

intent.putExtra("albumUnable", 1);

intent.putExtra("cancleUnable", 1);

startActivityForResult(intent, SCAN\_REQUEST\_CODE);

}

// a为原字符串，b为要插入的字符串，t为插入位置

public String Stringinsert(String a, String b, int t) {

return a.substring(0, t) + b + a.substring(t, a.length());

}

String isbn;

private String mState;

@Override

public void onActivityResult(final int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (resultCode == Activity.RESULT\_OK) {

switch (requestCode) {

case SCAN\_REQUEST\_CODE:

isbn = data.getStringExtra("CaptureIsbn");

if (!TextUtils.isEmpty(isbn)) {

if (StrLengthUtil.length(isbn) == 16) {

/\*Intent intent = new Intent(getActivity(), PublishClaimActivity.class);

intent.putExtra("isbn", isbn);

startActivityForResult(intent, 1001);\*/

OkGo.<String>get(Constants.SELECT\_LIVE\_STOCK)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("deviceNO", isbn)

.params("ranchID", mLoginSuccess.getRanchID())

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String result = response.body().toString();

Gson gson = new Gson();

SelectLivestock selectLivestock = gson.fromJson(result, SelectLivestock.class);

String msg = selectLivestock.getMsg();

String msg1 = selectLivestock.msg1;

boolean code = selectLivestock.isCode();

if (code) {

//已经发布过,是否重新发布认领

new SweetAlertDialog(getActivity(), SweetAlertDialog.WARNING\_TYPE)

.setTitleText("")

.setContentText("已经发布过,是否重新发布认领?")

.setCancelText("否")

.setConfirmText("是")

.setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener() {

@Override

public void onClick(SweetAlertDialog sweetAlertDialog) {

sweetAlertDialog.cancel();

Intent intent = new Intent(getActivity(), PublishClaimActivity.class);

intent.putExtra("isbn", isbn);

startActivityForResult(intent, 1001);

}

})

.setCancelClickListener(new SweetAlertDialog.OnSweetClickListener() {

@Override

public void onClick(SweetAlertDialog sDialog) {

sDialog.cancel();

}

})

.show();

} else if (msg.equals("false") && msg1.contains("此设备已经被牧场主为")) {

ToastUtils.showShort("此设备已经被别的牧场主所登记");

new SweetAlertDialog(getActivity(), SweetAlertDialog.ERROR\_TYPE)

.setTitleText("无法发布认领")

.setContentText("此设备已经被别的牧场主所登记")

.show();

} else {

//没有发布过

Intent intent = new Intent(getActivity(), PublishClaimActivity.class);

intent.putExtra("isbn", isbn);

startActivityForResult(intent, 1001);

}

}

});

} else if (StrLengthUtil.length(isbn) == 15) {

final String str = Stringinsert(isbn, "1", 7);

Log.d(TAG1, "15位isbn=" + str);

Log.d(TAG1, "新的长度" + StrLengthUtil.length(str));

/\*Intent intent = new Intent(getActivity(), PublishClaimActivity.class);

intent.putExtra("isbn", str);

startActivityForResult(intent, 1001);\*/

OkGo.<String>get(Constants.SELECT\_LIVE\_STOCK)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("deviceNO", str)

.params("ranchID", mLoginSuccess.getRanchID())

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String result = response.body().toString();

Gson gson = new Gson();

SelectLivestock selectLivestock = gson.fromJson(result, SelectLivestock.class);

String msg = selectLivestock.getMsg();

boolean code = selectLivestock.isCode();

if (code) {

//已经发布过,是否重新发布认领

new SweetAlertDialog(getActivity(), SweetAlertDialog.WARNING\_TYPE)

.setTitleText("")

.setContentText("已经发布过,是否重新发布认领")

.setCancelText("否")

.setConfirmText("是")

.setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener() {

@Override

public void onClick(SweetAlertDialog sweetAlertDialog) {

sweetAlertDialog.cancel();

Intent intent = new Intent(getActivity(), PublishClaimActivity.class);

intent.putExtra("isbn", str);

startActivityForResult(intent, 1001);

}

})

.setCancelClickListener(new SweetAlertDialog.OnSweetClickListener() {

@Override

public void onClick(SweetAlertDialog sDialog) {

sDialog.cancel();

}

})

.show();

} else {

//没有发布过

Intent intent = new Intent(getActivity(), PublishClaimActivity.class);

intent.putExtra("isbn", str);

startActivityForResult(intent, 1001);

}

}

});

} else {

ToastUtils.showShort("设备号必须是16位数字");

}

}

case 1001:

//刷新数据

OkGo.<String>post(Constants.LIVE\_STOCK\_CLAIM\_LIST)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("ranchID", mLoginSuccess.getRanchID())

.params("isClaimed", 0)

.params("current", 1)

.params("pagesize", 10)

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String s = response.body().toString();

// Log.d(TAG1, s);

if (s.contains("livestockId")) {

index = 2;

Gson gson1 = new Gson();

RenLing renLing = gson1.fromJson(s, RenLing.class);

mLivestockList = renLing.getLivestockList();

//创建并设置Adapter

mAdapter = new MyAdapter(mLivestockList);

mRecyclerView.setAdapter(mAdapter);

MoveToPosition(mLayoutManager, 0);

mAdapter.setOnItemClickListener(new OnRecyclerViewItemClickListener() {

@Override

public void onItemClick(View view, int position) {

RenLing.LivestockListBean livestockListBean = mLivestockList.get(position);

Intent intent = new Intent(getActivity(), RenlingDetailActivity.class);

intent.putExtra("getDeviceNo", livestockListBean.getDeviceNo());

startActivity(intent);

}

});

}

}

});

break;

case 1002:

//支付成功，刷新界面

//刷新数据

OkGo.<String>post(Constants.LIVE\_STOCK\_CLAIM\_LIST)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("ranchID", mLoginSuccess.getRanchID())

// .params("isClaimed",)

.params("current", 1)

.params("pagesize", 10)

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String s = response.body().toString();

// Log.d(TAG1, s);

if (s.contains("livestockId")) {

index = 2;

Gson gson1 = new Gson();

RenLing renLing = gson1.fromJson(s, RenLing.class);

mLivestockList = renLing.getLivestockList();

//创建并设置Adapter

mAdapter = new MyAdapter(mLivestockList);

mRecyclerView.setAdapter(mAdapter);

MoveToPosition(mLayoutManager, 0);

mAdapter.setOnItemClickListener(new OnRecyclerViewItemClickListener() {

@Override

public void onItemClick(View view, int position) {

RenLing.LivestockListBean livestockListBean = mLivestockList.get(position);

Intent intent = new Intent(getActivity(), RenlingDetailActivity.class);

intent.putExtra("getDeviceNo", livestockListBean.getDeviceNo());

startActivity(intent);

}

});

} else {

}

}

});

break;

}

}

}

@Nullable

@Override

public View onCreateView(final LayoutInflater inflater, @Nullable ViewGroup container,

@Nullable Bundle savedInstanceState) {

View view = View.inflate(getActivity(), R.layout.fragment\_claim, null);

ImageView ivAdd = view.findViewById(R.id.ivAdd);

ivAdd.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

//扫描

checkedItem = 2;

startScanActivity();

}

});

RefreshLayout refreshLayout = view.findViewById(R.id.refreshLayout);

refreshLayout.setOnRefreshListener(new OnRefreshListener() {

@Override

public void onRefresh(RefreshLayout refreshlayout) {

if (mState.equals("全部")) {

OkGo.<String>post(Constants.LIVE\_STOCK\_CLAIM\_LIST)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("ranchID", mLoginSuccess.getRanchID())

// .params("isClaimed",)

.params("current", 1)

.params("pagesize", 10)

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String s = response.body().toString();

// Log.d(TAG1, s);

if (s.contains("livestockId")) {

index = 2;

Gson gson1 = new Gson();

RenLing renLing = gson1.fromJson(s, RenLing.class);

mLivestockList = renLing.getLivestockList();

//创建并设置Adapter

mAdapter = new MyAdapter(mLivestockList);

mRecyclerView.setAdapter(mAdapter);

mAdapter.setOnItemClickListener(new OnRecyclerViewItemClickListener() {

@Override

public void onItemClick(View view, int position) {

Log.d(TAG1, position + "被点击了");

RenLing.LivestockListBean livestockListBean = mLivestockList.get(position);

Intent intent = new Intent(getActivity(), RenlingDetailActivity.class);

intent.putExtra("getImgUrl", livestockListBean.getImgUrl());

intent.putExtra("getLivestockName", livestockListBean.getLivestockName());

intent.putExtra("getDeviceNo", livestockListBean.getDeviceNo());

intent.putExtra("getCharacteristics", livestockListBean.getCharacteristics());

intent.putExtra("getCellphone", livestockListBean.getCellphone());

intent.putExtra("getCreateTime", livestockListBean.getCreateTime());

intent.putExtra("getPrice", livestockListBean.getPrice());

intent.putExtra("getIsClaimed", livestockListBean.getIsClaimed());

intent.putExtra("getLifeTime", livestockListBean.getLifeTime());

intent.putExtra("getBirthTime", livestockListBean.getBirthTime());

intent.putExtra("getClaimTime", livestockListBean.getClaimTime());

startActivity(intent);

}

});

} else {

}

}

});

} else if (mState.equals("未认领")) {

OkGo.<String>post(Constants.LIVE\_STOCK\_CLAIM\_LIST)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("ranchID", mLoginSuccess.getRanchID())

.params("isClaimed", 0)

.params("current", 1)

.params("pagesize", 10)

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String s = response.body().toString();

// Log.d(TAG1, s);

if (s.contains("livestockId")) {

index = 2;

Gson gson1 = new Gson();

RenLing renLing = gson1.fromJson(s, RenLing.class);

mLivestockList = renLing.getLivestockList();

//创建并设置Adapter

mAdapter = new MyAdapter(mLivestockList);

mRecyclerView.setAdapter(mAdapter);

mAdapter.setOnItemClickListener(new OnRecyclerViewItemClickListener() {

@Override

public void onItemClick(View view, int position) {

Log.d(TAG1, position + "被点击了");

RenLing.LivestockListBean livestockListBean = mLivestockList.get(position);

Intent intent = new Intent(getActivity(), RenlingDetailActivity.class);

intent.putExtra("getImgUrl", livestockListBean.getImgUrl());

intent.putExtra("getLivestockName", livestockListBean.getLivestockName());

intent.putExtra("getDeviceNo", livestockListBean.getDeviceNo());

intent.putExtra("getCharacteristics", livestockListBean.getCharacteristics());

intent.putExtra("getCellphone", livestockListBean.getCellphone());

intent.putExtra("getCreateTime", livestockListBean.getCreateTime());

intent.putExtra("getPrice", livestockListBean.getPrice());

intent.putExtra("getIsClaimed", livestockListBean.getIsClaimed());

intent.putExtra("getLifeTime", livestockListBean.getLifeTime());

intent.putExtra("getBirthTime", livestockListBean.getBirthTime());

intent.putExtra("getClaimTime", livestockListBean.getClaimTime());

startActivity(intent);

}

});

} else {

}

}

});

} else if (mState.equals("已认领")) {

OkGo.<String>post(Constants.LIVE\_STOCK\_CLAIM\_LIST)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("ranchID", mLoginSuccess.getRanchID())

.params("isClaimed", 1)

.params("current", 1)

.params("pagesize", 10)

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String s = response.body().toString();

// Log.d(TAG1, s);

if (s.contains("livestockId")) {

index = 2;

Gson gson1 = new Gson();

RenLing renLing = gson1.fromJson(s, RenLing.class);

mLivestockList = renLing.getLivestockList();

//创建并设置Adapter

mAdapter = new MyAdapter(mLivestockList);

mRecyclerView.setAdapter(mAdapter);

mAdapter.setOnItemClickListener(new OnRecyclerViewItemClickListener() {

@Override

public void onItemClick(View view, int position) {

Log.d(TAG1, position + "被点击了");

RenLing.LivestockListBean livestockListBean = mLivestockList.get(position);

Intent intent = new Intent(getActivity(), RenlingDetailActivity.class);

intent.putExtra("getImgUrl", livestockListBean.getImgUrl());

intent.putExtra("getLivestockName", livestockListBean.getLivestockName());

intent.putExtra("getDeviceNo", livestockListBean.getDeviceNo());

intent.putExtra("getCharacteristics", livestockListBean.getCharacteristics());

intent.putExtra("getCellphone", livestockListBean.getCellphone());

intent.putExtra("getCreateTime", livestockListBean.getCreateTime());

intent.putExtra("getPrice", livestockListBean.getPrice());

intent.putExtra("getIsClaimed", livestockListBean.getIsClaimed());

intent.putExtra("getLifeTime", livestockListBean.getLifeTime());

intent.putExtra("getBirthTime", livestockListBean.getBirthTime());

intent.putExtra("getClaimTime", livestockListBean.getClaimTime());

startActivity(intent);

}

});

} else {

}

}

});

}

refreshlayout.finishRefresh(1000);//传入false表示刷新失败

}

});

refreshLayout.setOnLoadMoreListener(new OnLoadMoreListener() {

@Override

public void onLoadMore(@NonNull final RefreshLayout refreshLayout) {

if (mState.equals("全部")) {

OkGo.<String>post(Constants.LIVE\_STOCK\_CLAIM\_LIST)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("ranchID", mLoginSuccess.getRanchID())

.params("current", index)

.params("pagesize", 10)

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String s = response.body().toString();

// Log.d(TAG1, s);

if (s.contains("livestockId")) {

index++;

Gson gson1 = new Gson();

RenLing renLing = gson1.fromJson(s, RenLing.class);

List<RenLing.LivestockListBean> mylist =

renLing.getLivestockList();

if (mylist.size() != 0) {

for (int i = 0; i < mylist.size(); i++) {

mLivestockList.add(mylist.get(i));

}

MoveToPosition(mLayoutManager, 10 \* (index - 1));

}

//创建并设置Adapter

mAdapter = new MyAdapter(mLivestockList);

mRecyclerView.setAdapter(mAdapter);

mAdapter.setOnItemClickListener(new OnRecyclerViewItemClickListener() {

@Override

public void onItemClick(View view, int position) {

RenLing.LivestockListBean livestockListBean = mLivestockList.get(position);

Intent intent = new Intent(getActivity(), RenlingDetailActivity.class);

intent.putExtra("getDeviceNo", livestockListBean.getDeviceNo());

startActivity(intent);

}

});

} else {

ToastUtils.showShort("没有更多数据了");

}

}

});

} else if (mState.equals("未认领")) {

OkGo.<String>post(Constants.LIVE\_STOCK\_CLAIM\_LIST)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("ranchID", mLoginSuccess.getRanchID())

.params("isClaimed", 0)

.params("current", index)

.params("pagesize", 10)

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String s = response.body().toString();

// Log.d(TAG1, s);

if (s.contains("livestockId")) {

index++;

Gson gson1 = new Gson();

RenLing renLing = gson1.fromJson(s, RenLing.class);

List<RenLing.LivestockListBean> mylist =

renLing.getLivestockList();

if (mylist.size() != 0) {

for (int i = 0; i < mylist.size(); i++) {

mLivestockList.add(mylist.get(i));

}

MoveToPosition(mLayoutManager, 10 \* (index - 1));

}

//创建并设置Adapter

mAdapter = new MyAdapter(mLivestockList);

mRecyclerView.setAdapter(mAdapter);

mAdapter.setOnItemClickListener(new OnRecyclerViewItemClickListener() {

@Override

public void onItemClick(View view, int position) {

RenLing.LivestockListBean livestockListBean = mLivestockList.get(position);

Intent intent = new Intent(getActivity(), RenlingDetailActivity.class);

intent.putExtra("getDeviceNo", livestockListBean.getDeviceNo());

startActivity(intent);

}

});

} else {

ToastUtils.showShort("没有更多数据了");

}

}

});

} else if (mState.equals("已认领")) {

OkGo.<String>post(Constants.LIVE\_STOCK\_CLAIM\_LIST)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("ranchID", mLoginSuccess.getRanchID())

.params("current", index)

.params("isClaimed", 1)

.params("pagesize", 10)

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String s = response.body().toString();

// Log.d(TAG1, s);

if (s.contains("livestockId")) {

index++;

Gson gson1 = new Gson();

RenLing renLing = gson1.fromJson(s, RenLing.class);

List<RenLing.LivestockListBean> mylist =

renLing.getLivestockList();

if (mylist.size() != 0) {

for (int i = 0; i < mylist.size(); i++) {

mLivestockList.add(mylist.get(i));

}

MoveToPosition(mLayoutManager, 10 \* (index - 1));

}

//创建并设置Adapter

mAdapter = new MyAdapter(mLivestockList);

mRecyclerView.setAdapter(mAdapter);

mAdapter.setOnItemClickListener(new OnRecyclerViewItemClickListener() {

@Override

public void onItemClick(View view, int position) {

RenLing.LivestockListBean livestockListBean = mLivestockList.get(position);

Intent intent = new Intent(getActivity(), RenlingDetailActivity.class);

intent.putExtra("getDeviceNo", livestockListBean.getDeviceNo());

startActivity(intent);

}

});

} else {

ToastUtils.showShort("没有更多数据了");

}

}

});

}

refreshLayout.finishLoadMore();

}

});

mRecyclerView = view.findViewById(R.id.my\_recycler\_view);

//创建默认的线性LayoutManager

mLayoutManager = new LinearLayoutManager(getActivity());

mRecyclerView.setLayoutManager(mLayoutManager);

//如果可以确定每个item的高度是固定的，设置这个选项可以提高性能

mRecyclerView.setHasFixedSize(true);

mLogin\_success = PrefUtils.getString(getActivity(), "login\_success", null);

Gson gson = new Gson();

mLoginSuccess = gson.fromJson(mLogin\_success, LoginSuccess.class);

mUsername = PrefUtils.getString(getActivity(), "username", null);

OkGo.<String>post(Constants.LIVE\_STOCK\_CLAIM\_LIST)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("ranchID", mLoginSuccess.getRanchID())

// .params("isClaimed",)

.params("current", 1)

.params("pagesize", 10)

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String s = response.body().toString();

// Log.d(TAG1, s);

if (s.contains("livestockId")) {

Gson gson1 = new Gson();

RenLing renLing = gson1.fromJson(s, RenLing.class);

mLivestockList = renLing.getLivestockList();

//创建并设置Adapter

mAdapter = new MyAdapter(mLivestockList);

mRecyclerView.setAdapter(mAdapter);

mAdapter.setOnItemClickListener(new OnRecyclerViewItemClickListener() {

@Override

public void onItemClick(View view, int position) {

Log.d(TAG1, position + "被点击了");

RenLing.LivestockListBean livestockListBean = mLivestockList.get(position);

Intent intent = new Intent(getActivity(), RenlingDetailActivity.class);

intent.putExtra("getDeviceNo", livestockListBean.getDeviceNo());

intent.putExtra("getImgUrl", livestockListBean.getImgUrl());

intent.putExtra("getLivestockName", livestockListBean.getLivestockName());

intent.putExtra("getCharacteristics", livestockListBean.getCharacteristics());

intent.putExtra("getCellphone", livestockListBean.getCellphone());

intent.putExtra("getCreateTime", livestockListBean.getCreateTime());

intent.putExtra("getPrice", livestockListBean.getPrice());

intent.putExtra("getIsClaimed", livestockListBean.getIsClaimed());

intent.putExtra("getLifeTime", livestockListBean.getLifeTime());

intent.putExtra("getBirthTime", livestockListBean.getBirthTime());

intent.putExtra("getClaimTime", livestockListBean.getClaimTime());

startActivity(intent);

}

});

} else {

}

}

});

Spinner spinner = (Spinner) view.findViewById(R.id.spinner1);

spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

@Override

public void onItemSelected(AdapterView<?> parent, View view,

int pos, long id) {

String[] claim = getResources().getStringArray(R.array.claim);

mState = claim[pos];

if (mState.equals("全部")) {

mLogin\_success = PrefUtils.getString(getActivity(), "login\_success", null);

Gson gson = new Gson();

mLoginSuccess = gson.fromJson(mLogin\_success, LoginSuccess.class);

mUsername = PrefUtils.getString(getActivity(), "username", null);

OkGo.<String>post(Constants.LIVE\_STOCK\_CLAIM\_LIST)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("ranchID", mLoginSuccess.getRanchID())

// .params("isClaimed",)

.params("current", 1)

.params("pagesize", 10)

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

startIndex = 0;

String s = response.body().toString();

Log.d(TAG1, s);

if (s.contains("error")) {

return;

} else if (s.contains("成功")) {

Gson gson1 = new Gson();

RenLing renLing = gson1.fromJson(s, RenLing.class);

mLivestockList = renLing.getLivestockList();

//创建并设置Adapter

mAdapter = new MyAdapter(mLivestockList);

mRecyclerView.setAdapter(mAdapter);

mAdapter.setOnItemClickListener(new OnRecyclerViewItemClickListener() {

@Override

public void onItemClick(View view, int data) {

Log.d(TAG1, data + "被点击了");

RenLing.LivestockListBean livestockListBean = mLivestockList.get(data);

Intent intent = new Intent(getActivity(), RenlingDetailActivity.class);

intent.putExtra("getDeviceNo", livestockListBean.getDeviceNo());

startActivity(intent);

}

});

} else {

return;

}

}

});

} else if (claim[pos].equals("未认领")) {

mLogin\_success = PrefUtils.getString(getActivity(), "login\_success", null);

Gson gson = new Gson();

mLoginSuccess = gson.fromJson(mLogin\_success, LoginSuccess.class);

mUsername = PrefUtils.getString(getActivity(), "username", null);

OkGo.<String>post(Constants.LIVE\_STOCK\_CLAIM\_LIST)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("ranchID", mLoginSuccess.getRanchID())

.params("isClaimed", 0) //未认领

.params("current", 1)

.params("pagesize", 10)

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

startIndex = 0;

String s = response.body().toString();

Log.d(TAG1, s);

if (s.contains("未有发布牲畜") || s.contains("获取发布到认领输出信息异常")) {

if (mLivestockList != null) {

mLivestockList.clear();

mAdapter.notifyDataSetChanged();

}

} else if (s.contains("成功")) {

Gson gson1 = new Gson();

RenLing renLing = gson1.fromJson(s, RenLing.class);

mLivestockList = renLing.getLivestockList();

//创建并设置Adapter

mAdapter = new MyAdapter(mLivestockList);

mRecyclerView.setAdapter(mAdapter);

mAdapter.setOnItemClickListener(new OnRecyclerViewItemClickListener() {

@Override

public void onItemClick(View view, int data) {

Log.d(TAG1, data + "被点击了");

RenLing.LivestockListBean livestockListBean = mLivestockList.get(data);

Intent intent = new Intent(getActivity(), RenlingDetailActivity.class);

intent.putExtra("getDeviceNo", livestockListBean.getDeviceNo());

startActivity(intent);

}

});

}

}

});

} else if (claim[pos].equals("已认领")) {

mLogin\_success = PrefUtils.getString(getActivity(), "login\_success", null);

Gson gson = new Gson();

mLoginSuccess = gson.fromJson(mLogin\_success, LoginSuccess.class);

mUsername = PrefUtils.getString(getActivity(), "username", null);

OkGo.<String>post(Constants.LIVE\_STOCK\_CLAIM\_LIST)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("ranchID", mLoginSuccess.getRanchID())

.params("isClaimed", 1)

.params("current", 1)

.params("pagesize", 10)

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

startIndex = 0;

String s = response.body().toString();

Log.d(TAG1, s);

if (s.contains("未有发布牲畜") || s.contains("获取发布到认领输出信息异常")) {

if (mLivestockList != null) {

mLivestockList.clear();

mAdapter.notifyDataSetChanged();

}

} else if (s.contains("成功")) {

Gson gson1 = new Gson();

RenLing renLing = gson1.fromJson(s, RenLing.class);

mLivestockList = renLing.getLivestockList();

//创建并设置Adapter

mAdapter = new MyAdapter(mLivestockList);

mRecyclerView.setAdapter(mAdapter);

mAdapter.setOnItemClickListener(new OnRecyclerViewItemClickListener() {

@Override

public void onItemClick(View view, int data) {

Log.d(TAG1, data + "被点击了");

RenLing.LivestockListBean livestockListBean = mLivestockList.get(data);

Intent intent = new Intent(getActivity(), RenlingDetailActivity.class);

intent.putExtra("getDeviceNo", livestockListBean.getDeviceNo());

startActivity(intent);

}

});

}

}

});

}

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

// Another interface callback

}

});

return view;

}

int startIndex = -1; // 起始页（从0开始）

private String[] getDummyDatas() {

String[] arr = {"北京", "上海", "广州", "深圳"};

return arr;

}

private RecyclerView mRecyclerView;

private LinearLayoutManager mLayoutManager;

private MyAdapter mAdapter;

public static interface OnRecyclerViewItemClickListener {

void onItemClick(View view, int data);

}

/\*\*

\* RecyclerView 移动到当前位置，

\*

\* @param manager 设置RecyclerView对应的manager

\* @param n 要跳转的位置

\*/

public static void MoveToPosition(LinearLayoutManager manager, int n) {

manager.scrollToPositionWithOffset(n, 0);

manager.setStackFromEnd(true);

}

public class MyAdapter extends RecyclerView.Adapter<MyAdapter.ViewHolder> implements View.OnClickListener {

private OnRecyclerViewItemClickListener mOnItemClickListener = null;

public void setOnItemClickListener(OnRecyclerViewItemClickListener listener) {

this.mOnItemClickListener = listener;

}

public List<RenLing.LivestockListBean> datas = null;

public MyAdapter(List<RenLing.LivestockListBean> datas) {

this.datas = datas;

}

//创建新View，被LayoutManager所调用

@Override

public ViewHolder onCreateViewHolder(ViewGroup viewGroup, int viewType) {

View view = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.item\_claim, viewGroup, false);

ViewHolder vh = new ViewHolder(view);

//将创建的View注册点击事件

view.setOnClickListener(this);

return vh;

}

//将数据与界面进行绑定的操作

@Override

public void onBindViewHolder(ViewHolder viewHolder, final int position) {

// viewHolder.tvId.setText(datas.get(position).getDeviceNo());

//将数据保存在itemView的Tag中，以便点击时进行获取

viewHolder.itemView.setTag(position);

String imgUrl = datas.get(position).getImgUrl();

imgUrl = Constants.BASE\_URL + imgUrl;

Log.d(TAG1, imgUrl);

Uri uri = Uri.parse(imgUrl);

viewHolder.ivGhoat.setImageURI(uri);

String livestockName = datas.get(position).getLivestockName();

viewHolder.tvAnimalName.setText(livestockName);

mDeviceNo = datas.get(position).getDeviceNo();

viewHolder.tvId.setText("设备号:" + mDeviceNo);

String createTime = datas.get(position).getCreateTime();

String nowString = TimeUtils.getNowString();

// Log.d(TAG1, "createTime=" + createTime);

// Log.d(TAG1, "nowString=" + nowString);

long timeSpanByNow = TimeUtils.getTimeSpanByNow(createTime, TimeConstants.DAY);

Log.d(TAG1, timeSpanByNow + "天=timeSpanByNow");

int age = (int) timeSpanByNow / 30;

Log.d(TAG1, age + "个月");

if (age == 1) {

age = 2;

} else if (age == 2) {

age = 3;

} else if (age == 0) {

age = 1;

}

viewHolder.tvAnimalAge.setText("年龄：" + age + "个月");

String name = datas.get(position).getName();

if (TextUtils.isEmpty(name)) {

viewHolder.tvMuchangName.setText("牧场：无");

} else {

viewHolder.tvMuchangName.setText("牧场：" + name);

}

String isClaimed = datas.get(position).getIsClaimed();

if (isClaimed.contains("1")) {

viewHolder.tvDel.setText("");

String claimTime = datas.get(position).getClaimTime();

claimTime = claimTime.substring(0, 10);

String finishTime = datas.get(position).getFinishTime();

finishTime = finishTime.substring(0, 10);

//已认领

viewHolder.tvClaimTime.setText("认领时间：" + claimTime + "至" + finishTime);

String tradeStatus = datas.get(position).getTradeStatus();

String orderNo = datas.get(position).getOrderNo();

viewHolder.tvOrderNo.setText("支付订单号：" + orderNo);

if (!TextUtils.isEmpty(tradeStatus)) {

if (tradeStatus.contains("0")) {

//已认领未支付

viewHolder.btnPayConfirm.setVisibility(View.INVISIBLE);

viewHolder.tvState.setText("已认领未支付");

} else if (tradeStatus.contains("1")) {

//已认领未支付

viewHolder.btnPayConfirm.setVisibility(View.VISIBLE);

viewHolder.tvState.setText("已认领已支付");

viewHolder.btnPayConfirm.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new Intent(getActivity(), PayConfirmActivity.class);

mDeviceNo = datas.get(position).getDeviceNo();

intent.putExtra("mDeviceNo", mDeviceNo);

startActivityForResult(intent, 1002);

}

});

} else if (tradeStatus.contains("3")) {

//已认领已支付

viewHolder.btnPayConfirm.setVisibility(View.GONE);

viewHolder.tvState.setText("交易成功");

}

}

viewHolder.tvClaimPeople.setText("认领人：" + datas.get(position).getNickname());

viewHolder.tvPhone.setText("电话：" + datas.get(position).getCellphone());

} else if (isClaimed.contains("0")) {

//未认领

viewHolder.tvClaimTime.setText("发布时间：" + datas.get(position).getBirthTime());

viewHolder.tvState.setText("未认领");

viewHolder.tvOrderNo.setText("");

viewHolder.tvClaimPeople.setText("");

viewHolder.tvPhone.setText("");

viewHolder.btnPayConfirm.setVisibility(View.GONE);

viewHolder.tvDel.setText("删除");

viewHolder.tvDel.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

new SweetAlertDialog(getActivity(), SweetAlertDialog.WARNING\_TYPE)

.setTitleText("删除?")

.setContentText("删除此条目")

.setConfirmText("确定")

.setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener() {

@Override

public void onClick(SweetAlertDialog sDialog) {

sDialog.cancel();

OkGo.<String>get(Constants.delLivestock)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("deviceNo", datas.get(position).getDeviceNo())

.params("ranchID", mLoginSuccess.getRanchID())

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String result = response.body().toString();

if (result.contains("删除成功")) {

ToastUtils.showShort("删除成功");

//刷新数据

OkGo.<String>post(Constants.LIVE\_STOCK\_CLAIM\_LIST)

.tag(this)

.params("token", mLoginSuccess.getToken())

.params("username", mUsername)

.params("ranchID", mLoginSuccess.getRanchID())

//.params("isClaimed",0)

.params("current", 1)

.params("pagesize", 10)

.execute(new StringCallback() {

@Override

public void onSuccess(Response<String> response) {

String s = response.body().toString();

if (s.contains("livestockId")) {

index = 2;

Gson gson1 = new Gson();

RenLing renLing = gson1.fromJson(s, RenLing.class);

mLivestockList = renLing.getLivestockList();

//创建并设置Adapter

mAdapter = new MyAdapter(mLivestockList);

mRecyclerView.setAdapter(mAdapter);

MoveToPosition(mLayoutManager, 0);

mAdapter.setOnItemClickListener(new OnRecyclerViewItemClickListener() {

@Override

public void onItemClick(View view, int position) {

RenLing.LivestockListBean livestockListBean = mLivestockList.get(position);

Intent intent = new Intent(getActivity(), RenlingDetailActivity.class);

intent.putExtra("getDeviceNo", livestockListBean.getDeviceNo());

startActivity(intent);

}

});

}

}

});

} else if (result.contains("删除失败")) {

ToastUtils.showShort("删除失败");

} else if (result.contains("已被认领不可删除")) {

ToastUtils.showShort("已被认领不可删除");

} else {

ToastUtils.showShort("删除异常");

}

}

});

}

})

.setCancelText("取消")

.setCancelClickListener(new SweetAlertDialog.OnSweetClickListener() {

@Override

public void onClick(SweetAlertDialog sDialog) {

sDialog.cancel();

}

})

.show();

}

});

}

}

@Override

public void onClick(View v) {

if (mOnItemClickListener != null) {

//注意这里使用getTag方法获取数据

mOnItemClickListener.onItemClick(v, (int) v.getTag());

}

}

//获取数据的数量

@Override

public int getItemCount() {

return datas.size();

}

//自定义的ViewHolder，持有每个Item的的所有界面元素

public class ViewHolder extends RecyclerView.ViewHolder {

SimpleDraweeView ivGhoat;

TextView tvAnimalName, tvId, tvAnimalAge, tvMuchangName,

tvClaimTime, tvClaimPeople, tvState,

tvPhone, tvOrderNo, tvDel;

Button btnPayConfirm;

public ViewHolder(View view) {

super(view);

// mTextView = view.findViewById(R.id.tvClaim);

ivGhoat = view.findViewById(R.id.ivGhoat);

tvAnimalName = view.findViewById(R.id.tvAnimalName);

tvId = view.findViewById(R.id.tvId);

tvAnimalAge = view.findViewById(R.id.tvAnimalAge);

tvMuchangName = view.findViewById(R.id.tvMuchangName);

tvClaimTime = view.findViewById(R.id.tvClaimTime);

tvClaimPeople = view.findViewById(R.id.tvClaimPeople);

tvState = view.findViewById(R.id.tvState);

tvPhone = view.findViewById(R.id.tvPhone);

tvOrderNo = view.findViewById(R.id.tvOrderNo);

btnPayConfirm = view.findViewById(R.id.btnPayConfirm);

tvDel = view.findViewById(R.id.tvDel);

}

}

}

}

## 数据存储及数据库定义

##### 本地数据库存储

使用sharedPreference ,工具类PrefUtils。

简易使用：

给key 赋值 value

PrefUtils.setString(getApplicationContext(),"key","value");

取值：

String key= PrefUtils.getString(getApplicationContext(),”key”,””);

重要字段信息有 ：

loginState username login\_success

##### 网络后台数据库

public static final String IP = "182.92.3.109";

public static final String PORT = ":10100";

// public static final String IP = "116.95.255.211";

// public static final String PORT = ":10100";

//测试服务器

public static final String BASE\_URL = "http://" + IP + PORT;

//内部正式服务器

//http://182.92.3.109:10100 testranch 12345678 这是109访问地址跟用户名密码

// public static final String BASE\_URL = "http://182.92.3.109:10100";

//客户服务器

// public static final String BASE\_URL ="http://116.95.255.211:10120";

public static final int ranchID = 1;

public static final String token = "ce5cc971-5904-4454-b639-cb8d48215963";

//牧场主找回密码

//1)点击发送验证码：

public static final String VERIFY\_CODE = BASE\_URL + "/mobleClaim/txCode.do";

//2)修改密码:

public static final String MODIFY\_PASSWORD = BASE\_URL + "/ranchLogin/backPass.do";

//登录:

public static final String LOGIN = BASE\_URL + "/ranchLogin/login.do";

//发布认领

//判断是被是否被绑定

public static final String ISDEVICEBINDED = BASE\_URL + "/releaseLivestocks/isDeviceBinded.do";

//1、查看发布情况selectLivestock.do，未发布执行release.do

public static final String SELECT\_LIVE\_STOCK = BASE\_URL + "/releaseLivestocks/selectLivestock.do";

//根据字典表类型选字典表品种(牲畜扫描时选类型时调用)

public static final String SELECTVARIETY = BASE\_URL + "/releaseLivestocks/selectVariety.do";

//直接发布到认领表接口

public static final String RELEASE = BASE\_URL + "/releaseLivestocks/release.do";

//打疫苗登记-新增牲畜

public static final String SAVELIVESTOCK = BASE\_URL + "/releaseLivestocks/saveLivestock.do";

//重新发布到牲畜认领表

public static final String IS\_CLAIMED = BASE\_URL + "/releaseLivestocks/isClaimed.do";

//获取已发布牲畜信息

public static final String LIVE\_STOCK\_CLAIM\_LIST = BASE\_URL + "/releaseLivestocks/livestockClaimList.do";

//剪毛

public static final String SHEARING = BASE\_URL + "/releaseLivestocks/shearing.do";

//牧场拍生活照和视频

public static final String RANCHIMGVIDEO = BASE\_URL + "/releaseLivestocks/ranchImgVideo.do";

//牧场详情

public static final String RANCH = BASE\_URL + "/releaseLivestocks/ranch.do";

//更新牧场

public static final String UPDRANCH = BASE\_URL + "/releaseLivestocks/updRanch.do";

//牲畜详情

public static final String LIVESTOCK = BASE\_URL + "/releaseLivestocks/livestock.do";

//设备消息

public static final String DEVICEMSG = BASE\_URL + "/releaseLivestocks/deviceMsg.do";

//牧场主处理拍照和视频请求

public static final String updLivestockClaim = BASE\_URL + "/releaseLivestocks/updLivestockClaim.do";

//设备解绑（解绑之前要弹出确认提示框）

public static final String DOUNBINDDEVICE = BASE\_URL + "/releaseLivestocks/doUnBindDevice.do";

//个人信息

//1）查询个人信息

public static final String ADMINLIST = BASE\_URL + "/adminMember/adminList.do";

//2）更新个人信息

public static final String UPDADMIN = BASE\_URL + "/adminMember/updAdmin.do";

//文件上传（图片、视频） /adminMember/headImgUrl.do

public static final String HEADIMGURL = BASE\_URL + "/adminMember/headImgUrl.do";

//获取牧场牲畜类型和每种类型牲畜的数量

public static final String QUERYTYPEANDSUM = BASE\_URL + "/releaseLivestocks/queryTypeAndSum.do";

//通过牲畜类型查询所有牲畜

public static final String QUERYLIVESTOCKVARIETYLIST = BASE\_URL + "/releaseLivestocks/queryLivestockVarietyList.do";

//电子档案

public static final String DianziDangan = BASE\_URL + "/claimLivestock/getArchivesElectronicURL.do";

//牧场主确认订单支付

public static final String orderPay = BASE\_URL + "/releaseLivestocks/orderPay.do";

//牲畜删除

public static final String delLivestock= BASE\_URL + "/releaseLivestocks/delLivestock.do";

### 通过访问以上这些接口，可以获取到服务器后台的数据

##### 对服务器的数据，进行解析

例如：QUERYTYPEANDSUM

##### 访问查询牧群种类及数量的接口

*BASE\_URL* + "/releaseLivestocks/queryTypeAndSum.do";

##### 返回结果：

{"msg":"获取牲畜类型和数量成功","code":"success","typeMap":{"1":66,"2":6,"3":4,"4":4,"7":4}}

##### 然后根据以上 json 信息，写 Javabean 类， 将信息 一一对应。

package com.jinkun\_innovation.pastureland.bean;

import com.google.gson.annotations.SerializedName;

/\*\*

\* Created by Guan on 2018/3/26.

\*/

public class MuqunSum {

/\*\*

\* msg : 获取牲畜类型和数量成功

\* code : success

\* typeMap : {"1":235,"2":3,"3":6,"4":8,"7":1}

\*/

private String msg;

private String code;

private TypeMapBean typeMap;

public String getMsg() {

return msg;

}

public void setMsg(String msg) {

this.msg = msg;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public TypeMapBean getTypeMap() {

return typeMap;

}

public void setTypeMap(TypeMapBean typeMap) {

this.typeMap = typeMap;

}

public static class TypeMapBean {

/\*\*

\* 1 : 235

\* 2 : 3

\* 3 : 6

\* 4 : 8

\* 7 : 1

\*/

@SerializedName("1")

private int \_$1;

@SerializedName("2")

private int \_$2;

@SerializedName("3")

private int \_$3;

@SerializedName("4")

private int \_$4;

@SerializedName("7")

private int \_$7;

public int get\_$1() {

return \_$1;

}

public void set\_$1(int \_$1) {

this.\_$1 = \_$1;

}

public int get\_$2() {

return \_$2;

}

public void set\_$2(int \_$2) {

this.\_$2 = \_$2;

}

public int get\_$3() {

return \_$3;

}

public void set\_$3(int \_$3) {

this.\_$3 = \_$3;

}

public int get\_$4() {

return \_$4;

}

public void set\_$4(int \_$4) {

this.\_$4 = \_$4;

}

public int get\_$7() {

return \_$7;

}

public void set\_$7(int \_$7) {

this.\_$7 = \_$7;

}

}

}

Gson gson1 = new Gson();

MuqunSum muqunSum = gson1.fromJson(s, MuqunSum.class);

MuqunSum.TypeMapBean typeMap = muqunSum.getTypeMap();

String s1 = typeMap.get\_$1() + "";

String s2 = typeMap.get\_$2() + "";

String s3 = typeMap.get\_$3() + "";

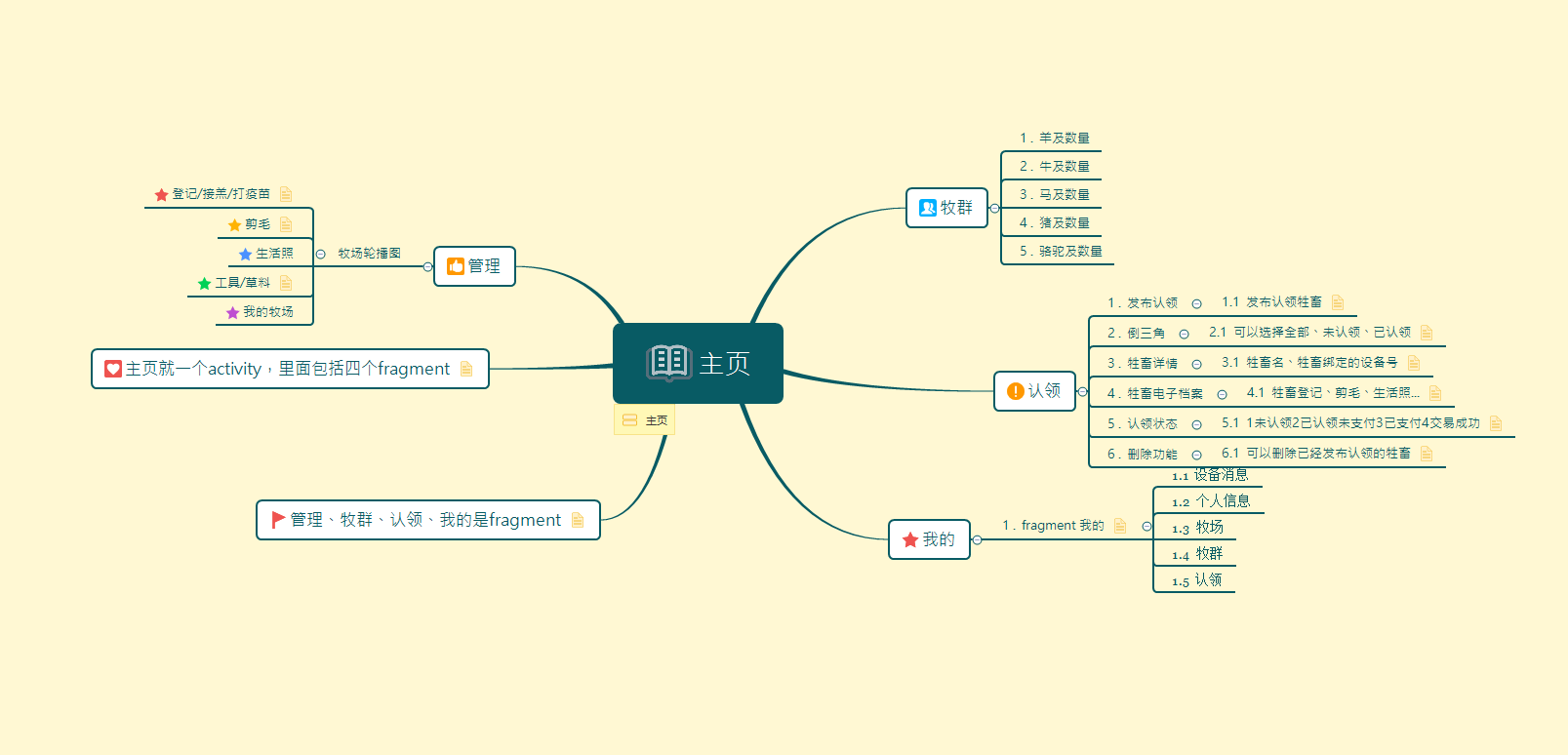
String s4 = typeMap.get\_$4() + "";

String s7 = typeMap.get\_$7() + "";

通过以上的代码，可以轻松将网络数据捕获，然后显示在app 界面上，至此，数据解析完成。

## 手机APP前端实现（菜单及页面显示功能模块介绍）

##### 软件框架：



实现技术:

1、Viewpager + tablayout + fragment 主体布局框架

1. 首页轮播图技术，使用的是 AndroidImageSlider

3、技术点太多，详情见牧场主app 开发文档。