package com.heiyue.project.im;

import java.util.ArrayList;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import org.json.JSONArray;

import android.text.TextUtils;

public class EaseAtMessageHelper {

private List<String> toAtUserList = new ArrayList<String>();

private Set<String> atMeGroupList = null;

private static EaseAtMessageHelper instance = null;

public synchronized static EaseAtMessageHelper get(){

if(instance == null){

instance = new EaseAtMessageHelper();

}

return instance;

}

private EaseAtMessageHelper(){

atMeGroupList = EasePreferenceManager.getInstance().getAtMeGroups();

if(atMeGroupList == null)

atMeGroupList = new HashSet<String>();

}

public void addAtUser(String username){

synchronized (toAtUserList) {

if(!toAtUserList.contains(username)){

toAtUserList.add(username);

}

}

}

public boolean containsAtUsername(String content){

if(TextUtils.isEmpty(content)){

return false;

}

synchronized (toAtUserList) {

for(String username : toAtUserList){

String nick = username;

if(EaseUserUtils.getUserInfo(username) != null){

EaseUser user = EaseUserUtils.getUserInfo(username);

if (user != null) {

nick = user.getNick();

}

}

if(content.contains(nick)){

return true;

}

}

}

return false;

}

public boolean containsAtAll(String content){

String atAll = "@" + EaseUI.getInstance().getContext().getString(R.string.action\_settings

// R.string.all\_members

);

if(content.contains(atAll)){

return true;

}

return false;

if(TextUtils.isEmpty(content)){

return null;

}

synchronized (toAtUserList) {

List<String> list = null;

for(String username : toAtUserList){

String nick = username;

if(EaseUserUtils.getUserInfo(username) != null){

EaseUser user = EaseUserUtils.getUserInfo(username);

if (user != null) {

nick = user.getNick();

}

}

if(content.contains(nick)){

if(list == null){

list = new ArrayList<String>();

}

list.add(username);

}

}

return list;

}

}

public void parseMessages(List<EMMessage> messages) {

int size = atMeGroupList.size();

EMMessage[] msgs = messages.toArray(new EMMessage[messages.size()]);

for(EMMessage msg : msgs){

if(msg.getChatType() == ChatType.GroupChat){

String groupId = msg.getTo();

try {

JSONArray jsonArray = msg.getJSONArrayAttribute(EaseConstant.MESSAGE\_ATTR\_AT\_MSG);

for(int i = 0; i < jsonArray.length(); i++){

String username = jsonArray.getString(i);

if(EMClient.getInstance().getCurrentUser().equals(username)){

if(!atMeGroupList.contains(groupId)){

atMeGroupList.add(groupId);

break;

}

}

}

} catch (Exception e1) {

//Determine whether is @ all message

String usernameStr = msg.getStringAttribute(EaseConstant.MESSAGE\_ATTR\_AT\_MSG, null);

if(usernameStr!=null){ if(usernameStr.toUpperCase().equals(EaseConstant.MESSAGE\_ATTR\_VALUE\_AT\_MSG\_ALL)){

if(!atMeGroupList.contains(groupId)){

atMeGroupList.add(groupId);

}

}

}

}

if(atMeGroupList.size() != size){

EasePreferenceManager.getInstance().setAtMeGroups(atMeGroupList);

}

}

}

}

public Set<String> getAtMeGroups(){

return atMeGroupList;

}

public void removeAtMeGroup(String groupId){

if(atMeGroupList.contains(groupId)){

atMeGroupList.remove(groupId);

EasePreferenceManager.getInstance().setAtMeGroups(atMeGroupList);

}

}

public boolean hasAtMeMsg(String groupId){

return atMeGroupList.contains(groupId);

}

public boolean isAtMeMsg(EMMessage message){

EaseUser user = EaseUserUtils.getUserInfo(message.getFrom());

if(user != null){

try {

JSONArray jsonArray = message.getJSONArrayAttribute(EaseConstant.MESSAGE\_ATTR\_AT\_MSG);

for(int i = 0; i < jsonArray.length(); i++){

String username = jsonArray.getString(i);

if(username.equals(EMClient.getInstance().getCurrentUser())){

return true;

}

}

} catch (Exception e) {

//perhaps is a @ all message

String atUsername = message.getStringAttribute(EaseConstant.MESSAGE\_ATTR\_AT\_MSG, null);

if(atUsername != null){

if(atUsername.toUpperCase().equals(EaseConstant.MESSAGE\_ATTR\_VALUE\_AT\_MSG\_ALL)){

return true;

}

}

return false;

}

}

return false;

}

public JSONArray atListToJsonArray(List<String> atList){

JSONArray jArray = new JSONArray();

int size = atList.size();

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

String username = atList.get(i);

jArray.put(username);

}

return jArray;

}

public void cleanToAtUserList(){

synchronized (toAtUserList){

toAtUserList.clear();

}

}

}

package com.heiyue.project.im;  
import android.app.ActivityManager;  
import android.app.ActivityManager.RunningTaskInfo;  
import android.content.Context;  
import android.net.ConnectivityManager;  
import android.net.NetworkInfo;  
import android.text.TextUtils;  
import com.hyphenate.chat.EMConversation.EMConversationType;  
import com.hyphenate.chat.EMMessage;  
import com.hyphenate.chat.EMTextMessageBody;  
import com.hyphenate.chat.Message;  
import com.hyphenate.util.HanziToPinyin;  
import com.hyphenate.util.HanziToPinyin.Token;  
import java.util.ArrayList;  
import java.util.List;  
public class EaseCommonUtils {  
 private static final String TAG = "CommonUtils";  
 public static boolean isNetWorkConnected(Context context) {  
 if (context != null) {  
 ConnectivityManager mConnectivityManager = (ConnectivityManager) context.getSystemService(Context.CONNECTIVITY\_SERVICE);  
 NetworkInfo mNetworkInfo = mConnectivityManager.getActiveNetworkInfo();  
 if (mNetworkInfo != null) {  
 return mNetworkInfo.isAvailable() && mNetworkInfo.isConnected();  
 }  
 }  
 return false;  
 }  
  
 public static boolean isSdcardExist() {  
 return android.os.Environment.getExternalStorageState().equals(android.os.Environment.MEDIA\_MOUNTED);  
 }  
 public static EMMessage createExpressionMessage(String toChatUsername, String expressioName, String identityCode){  
 EMMessage message = EMMessage.createTxtSendMessage("["+expressioName+"]", toChatUsername);  
 if(identityCode != null){  
 message.setAttribute(EaseConstant.MESSAGE\_ATTR\_EXPRESSION\_ID, identityCode);  
 }  
 message.setAttribute(EaseConstant.MESSAGE\_ATTR\_IS\_BIG\_EXPRESSION, true);  
 return message;  
 }

public static String getMessageDigest(EMMessage message, Context context) {  
 String digest = "";  
 switch (message.getType()) {

case LOCATION:  
 if (message.direct() == EMMessage.Direct.RECEIVE) {  
 digest = getString(context, R.string.location\_recv);  
 digest = String.format(digest, message.getFrom());  
 return digest;  
 } else {  
 digest = getString(context, R.string.location\_prefix);  
 }  
 break;  
 case TXT:  
 EMTextMessageBody txtBody = (EMTextMessageBody) message.getBody();  
 if(message.getBooleanAttribute(EaseConstant.MESSAGE\_ATTR\_IS\_VOICE\_CALL, false)){  
 digest = getString(context, R.string.voice\_call) + txtBody.getMessage();  
 }else if(message.getBooleanAttribute(EaseConstant.MESSAGE\_ATTR\_IS\_VIDEO\_CALL, false)){  
 digest = getString(context, R.string.video\_call) + txtBody.getMessage();  
 }else if(message.getBooleanAttribute(EaseConstant.MESSAGE\_ATTR\_IS\_BIG\_EXPRESSION, false)){  
 if(!TextUtils.isEmpty(txtBody.getMessage())){  
 digest = txtBody.getMessage();  
 }else{  
 digest = getString(context, R.string.dynamic\_expression);  
 }  
 }else{  
 digest = txtBody.getMessage();  
// }  
 break;  
// case FILE:  
// digest = getString(context, R.string.file);  
// break;  
// default:  
// EMLog.e(TAG, "error, unknow type");  
// return "";  
 }  
 return digest;  
 }  
 public static String getMessageDigest(Message message, Context context) {  
 String digest = "";  
 switch (message.getType())

case TXT:  
 EMTextMessageBody txtBody = (EMTextMessageBody) message.getBody();  
 if(message.getBooleanAttribute(EaseConstant.MESSAGE\_ATTR\_IS\_VOICE\_CALL, false)){  
 digest = getString(context, R.string.voice\_call) + txtBody.getMessage();  
 }else if(message.getBooleanAttribute(EaseConstant.MESSAGE\_ATTR\_IS\_VIDEO\_CALL, false)){  
 digest = getString(context, R.string.video\_call) + txtBody.getMessage();  
 }else if(message.getBooleanAttribute(EaseConstant.MESSAGE\_ATTR\_IS\_BIG\_EXPRESSION, false)){  
 if(!TextUtils.isEmpty(txtBody.getMessage())){  
 digest = txtBody.getMessage();  
 }else{  
 digest = getString(context, R.string.dynamic\_expression);  
 }  
 }else{  
 digest = txtBody.getMessage();  
 }  
 break;  
 case FILE:  
 digest = getString(context, R.string.file);  
 break;  
 default:  
 EMLog.e(TAG, "error, unknow type");  
 return "";  
 }  
 return digest;  
 }  
 static String getString(Context context, int resId){  
 return context.getResources().getString(resId);  
 }  
 public static String getTopActivity(Context context) {  
 ActivityManager manager = (ActivityManager) context.getSystemService(Context.ACTIVITY\_SERVICE);  
 List<RunningTaskInfo> runningTaskInfos = manager.getRunningTasks(1);  
 if (runningTaskInfos != null)  
 return runningTaskInfos.get(0).topActivity.getClassName();  
 else  
 return "";  
 }

public static void setUserInitialLetter(EaseUser user) {  
 final String DefaultLetter = "#";  
 String letter = DefaultLetter;   
 final class GetInitialLetter {  
 String getLetter(String name) {  
 if (TextUtils.isEmpty(name)) {  
 return DefaultLetter;  
 }  
 char char0 = name.toLowerCase().charAt(0);  
 if (Character.isDigit(char0)) {  
 return DefaultLetter;  
 }  
 ArrayList<Token> l = HanziToPinyin.getInstance().get(name.substring(0, 1));  
 if (l != null && l.size() > 0 && l.get(0).target.length() > 0)  
 {  
 Token token = l.get(0);  
 String letter = token.target.substring(0, 1).toUpperCase();  
 char c = letter.charAt(0);  
 if (c < 'A' || c > 'Z') {  
 return DefaultLetter;  
 }  
 return letter;  
 }  
 return DefaultLetter;  
 }  
 }  
 if ( !TextUtils.isEmpty(user.getNick()) ) {  
 letter = new GetInitialLetter().getLetter(user.getNick());  
 user.setInitialLetter(letter);  
 return;  
 }   
 if (letter.equals(DefaultLetter) && !TextUtils.isEmpty(user.getUsername())) {  
 letter = new GetInitialLetter().getLetter(user.getUsername());  
 }  
 user.setInitialLetter(letter);  
 }  
 public static EMConversationType getConversationType(int chatType) {  
 if (chatType == EaseConstant.CHATTYPE\_SINGLE) {  
 return EMConversationType.Chat;  
 } else if (chatType == EaseConstant.CHATTYPE\_GROUP) {  
 return EMConversationType.GroupChat;  
 } else {  
 return EMConversationType.ChatRoom;  
 }  
 }  
}

package com.heiyue.project.im;  
public class EaseConstant {  
 public static final String MESSAGE\_ATTR\_IS\_VOICE\_CALL = "is\_voice\_call";  
 public static final String MESSAGE\_ATTR\_IS\_VIDEO\_CALL = "is\_video\_call";  
 public static final String MESSAGE\_ATTR\_IS\_BIG\_EXPRESSION = "em\_is\_big\_expression";  
 public static final String MESSAGE\_ATTR\_EXPRESSION\_ID = "em\_expression\_id";  
 public static final String MESSAGE\_ATTR\_AT\_MSG = "em\_at\_list";  
 public static final String MESSAGE\_ATTR\_VALUE\_AT\_MSG\_ALL = "ALL";  
 public static final int CHATTYPE\_SINGLE = 1;  
 public static final int CHATTYPE\_GROUP = 2;  
 public static final int CHATTYPE\_CHATROOM = 3;   
 public static final String EXTRA\_CHAT\_TYPE = "chatType";  
 public static final String EXTRA\_USER\_ID = "userId";  
}

package com.heiyue.project.im;  
import android.app.Notification;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.Context;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.media.AudioManager;  
import android.media.Ringtone;  
import android.media.RingtoneManager;  
import android.net.Uri;  
import android.os.Build;  
import android.os.Vibrator;  
import android.support.v4.app.NotificationCompat;  
import com.heiyue.project.im.EaseUI.EaseSettingsProvider;  
import com.hyphenate.chat.EMMessage;  
import com.hyphenate.util.EMLog;  
import com.hyphenate.util.EasyUtils;  
import java.util.HashSet;  
import java.util.List;  
import java.util.Locale;  
public class EaseNotifier {  
 private final static String TAG = "notify";  
 Ringtone ringtone = null;  
 protected final static String[] msg\_eng = { "sent a message", "sent a picture", "sent a voice",  
 "sent location message", "sent a video", "sent a file", "%1 contacts sent %2 messages"  
 };  
 protected static int notifyID = 0525; // start notification id  
 protected static int foregroundNotifyID = 0555;  
 protected NotificationManager notificationManager = null;  
 protected HashSet<String> fromUsers = new HashSet<String>();  
 protected int notificationNum = 0;  
 protected Context appContext;  
 protected String packageName;  
 protected String[] msgs;  
 protected long lastNotifiyTime;  
 protected AudioManager audioManager;  
 protected Vibrator vibrator;  
 protected EaseNotificationInfoProvider notificationInfoProvider;  
 public EaseNotifier() {  
 }  
 public EaseNotifier init(Context context){  
 appContext = context;  
 notificationManager = (NotificationManager) context.getSystemService(Context.NOTIFICATION\_SERVICE);  
  
 packageName = appContext.getApplicationInfo().packageName;  
 if (Locale.getDefault().getLanguage().equals("zh")) {  
 msgs = msg\_ch;  
 } else {  
 msgs = msg\_eng;  
 }  
 audioManager = (AudioManager) appContext.getSystemService(Context.AUDIO\_SERVICE);  
 vibrator = (Vibrator) appContext.getSystemService(Context.VIBRATOR\_SERVICE);  
 return this;  
 }  
 public void reset(){  
 resetNotificationCount();  
 cancelNotificaton();  
 }  
 void resetNotificationCount() {  
 notificationNum = 0;  
 fromUsers.clear();  
 }  
 void cancelNotificaton() {  
 if (notificationManager != null)  
 notificationManager.cancel(notifyID);  
 }  
 public synchronized void onNewMsg(EMMessage message) {  
 EaseSettingsProvider settingsProvider = EaseUI.getInstance().getSettingsProvider();  
 if(!settingsProvider.isMsgNotifyAllowed(message)){  
 return;  
 }   
 if (!EasyUtils.isAppRunningForeground(appContext)) {  
 EMLog.d(TAG, "app is running in backgroud");  
 sendNotification(message, false);  
 } else {  
 sendNotification(message, true);  
  
 }   
 vibrateAndPlayTone(message);  
 }  
 public synchronized void onNewMesg(List<EMMessage> messages) {  
 EaseSettingsProvider settingsProvider = EaseUI.getInstance().getSettingsProvider();  
 if(!settingsProvider.isMsgNotifyAllowed(null)){  
 return;  
 }  
 if (!EasyUtils.isAppRunningForeground(appContext)) {  
 EMLog.d(TAG, "app is running in backgroud");  
 sendNotification(messages, false);  
 } else {  
 sendNotification(messages, true);  
 }  
 vibrateAndPlayTone(messages.get(messages.size()-1));  
 }  
 protected void sendNotification (List<EMMessage> messages, boolean isForeground){  
 for(EMMessage message : messages){  
 if(!isForeground){  
 notificationNum++;  
 fromUsers.add(message.getFrom());  
 }  
 }  
 sendNotification(messages.get(messages.size()-1), isForeground, false);  
 }  
 protected void sendNotification (EMMessage message, boolean isForeground){  
 sendNotification(message, isForeground, true);  
 }  
 protected void sendNotification(EMMessage message, boolean isForeground, boolean numIncrease) {  
 String username = message.getFrom();  
 try {  
 String notifyText = username + " ";  
 switch (message.getType()) {  
 case VIDEO:  
 notifyText += msgs[4];  
 break;  
 case FILE:  
 notifyText += msgs[5];  
 break;  
 default:  
 break;  
 }  
 PackageManager packageManager = appContext.getPackageManager();  
 String appname = (String) packageManager.getApplicationLabel(appContext.getApplicationInfo());  
 String contentTitle = appname;  
 if (notificationInfoProvider != null) {  
 String customNotifyText = notificationInfoProvider.getDisplayedText(message);  
 String customCotentTitle = notificationInfoProvider.getTitle(message);  
 if (customNotifyText != null){  
 notifyText = customNotifyText;  
 }   
 if (customCotentTitle != null){  
 contentTitle = customCotentTitle;  
 }   
 }  
 NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(appContext)  
 .setSmallIcon(appContext.getApplicationInfo().icon)  
 .setWhen(System.currentTimeMillis())  
 .setAutoCancel(true);  
 Intent msgIntent = appContext.getPackageManager().getLaunchIntentForPackage(packageName);  
 if (notificationInfoProvider != null) {  
 msgIntent = notificationInfoProvider.getLaunchIntent(message);  
 }  
 PendingIntent pendingIntent = PendingIntent.getActivity(appContext, notifyID, msgIntent,PendingIntent.FLAG\_UPDATE\_CURRENT);  
  
 if(numIncrease){  
 // prepare latest event info section  
 if(!isForeground){  
 notificationNum++;  
 fromUsers.add(message.getFrom());  
 }  
 }  
 int fromUsersNum = fromUsers.size();  
 String summaryBody = msgs[6].replaceFirst("%1", Integer.toString(fromUsersNum)).replaceFirst("%2",Integer.toString(notificationNum));  
 if (notificationInfoProvider != null) {  
 // lastest text  
 String customSummaryBody = notificationInfoProvider.getLatestText(message, fromUsersNum,notificationNum);  
 if (customSummaryBody != null){  
 summaryBody = customSummaryBody;  
 }  
 // small icon  
 int smallIcon = notificationInfoProvider.getSmallIcon(message);  
 if (smallIcon != 0){  
 mBuilder.setSmallIcon(smallIcon);  
 }  
 }  
 mBuilder.setContentTitle(contentTitle);  
 mBuilder.setTicker(notifyText);  
 mBuilder.setContentText(summaryBody);  
 mBuilder.setContentIntent(pendingIntent);  
 mBuilder.setNumber(notificationNum);  
 Notification notification = mBuilder.build();  
 if (isForeground) {  
 notificationManager.notify(foregroundNotifyID, notification);  
 notificationManager.cancel(foregroundNotifyID);  
 } else {  
 notificationManager.notify(notifyID, notification);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }

public void vibrateAndPlayTone(EMMessage message) {  
 if(message != null){  
 if(EMClient.getInstance().chatManager().isSilentMessage(message)){  
 return;  
 }   
 }  
 if (System.currentTimeMillis() - lastNotifiyTime < 1000) {  
 // received new messages within 2 seconds, skip play ringtone  
 return;  
 }  
 try {  
 lastNotifiyTime = System.currentTimeMillis();

if (audioManager.getRingerMode() == AudioManager.RINGER\_MODE\_SILENT) {  
 EMLog.e(TAG, "in slient mode now");  
 return;  
 }  
 EaseSettingsProvider settingsProvider = EaseUI.getInstance().getSettingsProvider();  
 if(settingsProvider.isMsgVibrateAllowed(message)){  
 long[] pattern = new long[] { 0, 180, 80, 120 };  
 vibrator.vibrate(pattern, -1);  
 }  
 if(settingsProvider.isMsgSoundAllowed(message)){  
 if (ringtone == null) {  
 Uri notificationUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_NOTIFICATION);  
 ringtone = RingtoneManager.getRingtone(appContext, notificationUri);  
 if (ringtone == null) {  
 EMLog.d(TAG, "cant find ringtone at:" + notificationUri.getPath());  
 return;  
 }  
 }  
 if (!ringtone.isPlaying()) {  
 String vendor = Build.MANUFACTURER;   
 ringtone.play();  
 if (vendor != null && vendor.toLowerCase().contains("samsung")) {  
 Thread ctlThread = new Thread() {  
 public void run() {  
 try {  
 Thread.sleep(3000);  
 if (ringtone.isPlaying()) {  
 ringtone.stop();  
 }  
 } catch (Exception e) {  
 }  
 }  
 };  
 ctlThread.run();  
 }  
 }  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 public void setNotificationInfoProvider(EaseNotificationInfoProvider provider) {  
 notificationInfoProvider = provider;  
 }  
  
 public interface EaseNotificationInfoProvider {  
 String getDisplayedText(EMMessage message);  
  
 String getLatestText(EMMessage message, int fromUsersNum, int messageNum);  
 String getTitle(EMMessage message);

int getSmallIcon(EMMessage message);

Intent getLaunchIntent(EMMessage message);  
 }  
}

package com.heiyue.project.im;  
import java.io.File;  
import java.util.HashMap;  
import java.util.Map;  
import java.util.Map.Entry;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
import android.content.Context;  
import android.net.Uri;  
import android.text.Spannable;  
import android.text.Spannable.Factory;  
import android.text.style.ImageSpan;  
import com.heiyue.project.im.EaseUI.EaseEmojiconInfoProvider;  
public class EaseSmileUtils {  
 public static final String DELETE\_KEY = "em\_delete\_delete\_expression";   
 public static final String ee\_1 = "[):]";  
 public static final String ee\_2 = "[:D]";  
 public static final String ee\_3 = "[;)]";  
 public static final String ee\_4 = "[:-o]";  
 public static final String ee\_5 = "[:p]";  
 public static final String ee\_6 = "[(H)]";  
 public static final String ee\_7 = "[:@]";  
 public static final String ee\_8 = "[:s]";  
 public static final String ee\_9 = "[:$]";  
 public static final String ee\_10 = "[:(]";  
 public static final String ee\_11 = "[:'(]";  
 public static final String ee\_12 = "[:|]";   
 public static final String ee\_13 = "[(a)]";  
 public static final String ee\_14 = "[8o|]";  
 public static final String ee\_15 = "[8-|]";  
 public static final String ee\_16 = "[+o(]";  
 public static final String ee\_17 = "[<o)]";  
 public static final String ee\_18 = "[|-)]";  
 public static final String ee\_19 = "[\*-)]";  
 public static final String ee\_20 = "[:-#]";  
 public static final String ee\_21 = "[:-\*]";  
 public static final String ee\_22 = "[^o)]";  
 public static final String ee\_23 = "[8-)]";  
 public static final String ee\_24 = "[(|)]";  
 public static final String ee\_25 = "[(u)]";  
 public static final String ee\_26 = "[(S)]";  
 public static final String ee\_27 = "[(\*)]";  
 public static final String ee\_28 = "[(#)]";  
 public static final String ee\_29 = "[(R)]";  
 public static final String ee\_30 = "[({)]";  
 public static final String ee\_31 = "[(})]";  
 public static final String ee\_32 = "[(k)]";  
 public static final String ee\_33 = "[(F)]";  
 public static final String ee\_34 = "[(W)]";  
 public static final String ee\_35 = "[(D)]";  
 private static final Factory spannableFactory = Spannable.Factory.getInstance();  
 private static final Map<Pattern, Object> emoticons = new HashMap<Pattern, Object>();  
 static {  
 EaseEmojicon[] emojicons = EaseDefaultEmojiconDatas.getData();  
 for (EaseEmojicon emojicon : emojicons) {  
 addPattern(emojicon.getEmojiText(), emojicon.getIcon());  
 }  
 EaseEmojiconInfoProvider emojiconInfoProvider = EaseUI.getInstance().getEmojiconInfoProvider();  
 if(emojiconInfoProvider != null && emojiconInfoProvider.getTextEmojiconMapping() != null){  
 for(Entry<String, Object> entry : emojiconInfoProvider.getTextEmojiconMapping().entrySet()){  
 addPattern(entry.getKey(), entry.getValue());  
 }  
 }  
 }  
 public static void addPattern(String emojiText, Object icon){  
 emoticons.put(Pattern.compile(Pattern.quote(emojiText)), icon);  
 }  
 public static boolean addSmiles(Context context, Spannable spannable) {  
 boolean hasChanges = false;  
 for (Entry<Pattern, Object> entry : emoticons.entrySet()) {  
 Matcher matcher = entry.getKey().matcher(spannable);  
 while (matcher.find()) {  
 boolean set = true;  
 for (ImageSpan span : spannable.getSpans(matcher.start(),matcher.end(), ImageSpan.class))  
 if (spannable.getSpanStart(span) >= matcher.start()  
 && spannable.getSpanEnd(span) <= matcher.end())  
 spannable.removeSpan(span);  
 else {  
 set = false;  
 break;  
 }  
 if (set) {  
 hasChanges = true;  
 Object value = entry.getValue();  
 if(value instanceof String && !((String) value).startsWith("http")){  
 File file = new File((String) value);  
 if(!file.exists() || file.isDirectory()){  
 return false;  
 }  
 spannable.setSpan(new ImageSpan(context, Uri.fromFile(file)),  
 matcher.start(), matcher.end(),  
 Spannable.SPAN\_EXCLUSIVE\_EXCLUSIVE);  
 }else{  
 spannable.setSpan(new ImageSpan(context, (Integer)value),  
 matcher.start(), matcher.end(),  
 Spannable.SPAN\_EXCLUSIVE\_EXCLUSIVE);  
 }  
 }  
 }  
 }  
 return hasChanges;  
 }  
 public static Spannable getSmiledText(Context context, CharSequence text) {  
 Spannable spannable = spannableFactory.newSpannable(text);  
 addSmiles(context, spannable);  
 return spannable;  
 }  
 public static boolean containsKey(String key){  
 boolean b = false;  
 for (Entry<Pattern, Object> entry : emoticons.entrySet()) {  
 Matcher matcher = entry.getKey().matcher(key);  
 if (matcher.find()) {  
 b = true;  
 break;  
 }  
 }   
 return b;  
 }  
 public static int getSmilesSize(){  
 return emoticons.size();  
 }  
}

package com.heiyue.project.ui;  
import android.app.Activity;  
import android.content.Intent;  
import android.graphics.Color;  
import android.text.TextUtils;  
import android.view.View;  
import android.view.View.OnClickListener;  
import android.widget.EditText;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
import com.heiyue.net.NetResponse;  
import com.heiyue.net.RequestCallBack;  
import com.heiyue.project.base.BaseActivity;  
import com.heiyue.project.bean.ShopAddressBean;  
import com.heiyue.project.bean.ShopAreaInfoBean;  
import com.heiyue.project.util.PubDialogUtil;  
import com.heiyue.ui.ButtonTouch;  
import com.heiyue.util.LogOut;  
import com.heiyue.util.StringUtil;  
import com.qixiaofu.qxf.R;  
import butterknife.BindView;  
public class AddressAddActivity {  
 private ShopAddressBean mAddress;  
 private final int REQUESTCODE\_SELECTAREA = 1;  
 private PubDialogUtil pubDialogUtil;  
 private String prov\_id, city\_id, area\_id;  
 public static void startActivity(Activity activity, ShopAddressBean add, int requestCode) {  
 Intent intent = new Intent(activity, AddressAddActivity.class);  
 intent.putExtra("address", add);  
 activity.startActivityForResult(intent, requestCode);  
 }  
 @Override  
 protected int setContentLayoutView() {  
 return R.layout.user\_activity\_address\_add;  
 }  
 @BindView(R.id.edName)  
 public EditText mEdName;  
 @BindView(R.id.edPhone)  
 public EditText mEdPhone;  
 @BindView(R.id.layAddressSelect)  
 public LinearLayout mLayAddressSelect;  
 @BindView(R.id.tvAddress)  
 public TextView mTvAddress;  
 @BindView(R.id.edAddressDetail)  
 public EditText mEdAddressDetail;  
 @BindView(R.id.btnCommic)  
 public ButtonTouch mBtnCommic;  
 @Override  
 protected void initViews() {  
 }  
 @Override  
 protected void initData() {  
 mAddress = (ShopAddressBean) getIntent().getSerializableExtra("address");  
 if (mAddress == null) {  
 tvTitle.setText("新地址");  
 } else {  
 tvTitle.setText("旧地址");  
 btnRightText.setVisibility(View.VISIBLE);  
 btnRightText.setText("删除");  
 btnRightText.setTextColor(Color.parseColor("#FFFE3824"));  
 mTvAddress.setText(mAddress.area\_info);  
 mEdAddressDetail.setText(mAddress.address);  
 mEdName.setText(mAddress.true\_name);  
 mEdPhone.setText(mAddress.mob\_phone);  
 prov\_id = mAddress.prov\_id;  
 city\_id = mAddress.city\_id;  
 area\_id = mAddress.area\_id;  
 }  
 mLayAddressSelect.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 ChooseAreaActivity.startActivityForResult((Activity) context, REQUESTCODE\_SELECTAREA);  
 }  
 });  
 mBtnCommic.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 commitAdd();  
 }  
 });  
 }  
 private void deleteAdd(ShopAddressBean add) {  
 dao.delShopAddressInfor(add.address\_id, new RequestCallBack<String>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 if (result.netMsg.success) {  
 setResult(RESULT\_OK);  
 ((Activity) context).finish();  
 }  
 }  
 });  
 }  
 @Override  
 protected void setActionBarDetail() {  
 btnRightText.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (mAddress == null) {  
 return;  
 }  
 showDialogNew(mAddress, 1, "数据错误");  
 }  
 });  
 }  
 public void showDialogNew(final ShopAddressBean info, final int type, String message) {  
 if (info == null) {  
 showToast("数据异常");  
 return;  
 }  
 pubDialogUtil = new PubDialogUtil(AddressAddActivity.this);  
 pubDialogUtil.showDialogtitleText("操作提示", message, new OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 deleteAdd(info);  
 pubDialogUtil.dismiss();  
 }  
 }, false, "确定");  
 }  
 private void commitAdd() {  
 final String true\_name = mEdName.getText().toString();  
 final String mob\_phone = mEdPhone.getText().toString().trim();  
 final String address = mEdAddressDetail.getText().toString();  
 final String area\_info = mTvAddress.getText().toString();  
 if (TextUtils.isEmpty(true\_name)) {  
 showToast("请输入姓名");  
 return;  
 }  
 if (TextUtils.isEmpty(mob\_phone)) {  
 showToast("请输入手机号");  
 return;  
 }  
 if (!StringUtil.isValidPhone(mob\_phone)) {  
 showToast("输入手机号不正确");  
 return;  
 }  
 if (TextUtils.isEmpty(address)) {  
 showToast("请输入详细的地址");  
 return;  
 }  
 if (TextUtils.isEmpty(area\_info)) {  
 showToast("请选择所在区域");  
 return;  
 }  
 mBtnCommic.setEnabled(false);  
 if (mAddress == null) {

if (prov\_id == null || city\_id == null || area\_id == null || area\_info == null) {  
 showToast("请选择所在区域");  
 return;  
 }  
 dao.addShopAddressInfor(true\_name, mob\_phone, address, prov\_id,  
 city\_id, area\_id, area\_info,  
 new RequestCallBack<ShopAddressBean>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<ShopAddressBean> result) {  
 mBtnCommic.setEnabled(true);  
 if (result.netMsg.success) {  
 mBtnCommic.setEnabled(false);  
 if (result.content != null) {  
 Intent intent = new Intent();  
 ShopAddressBean content = result.content;  
 content.true\_name = true\_name;  
 content.mob\_phone = mob\_phone;  
 content.address = address;  
 content.prov\_id = prov\_id;  
 content.city\_id = city\_id;  
 content.area\_id = area\_id;  
 content.area\_info = area\_info;  
 intent.putExtra("newAddAddress", content);  
 setResult(RESULT\_OK, intent);  
 ((Activity) context).finish();  
 }  
 setResult(RESULT\_OK);  
 ((Activity) context).finish();  
 }  
 }  
 });  
 } else {  
 dao.editShopAddressInfor(mAddress.address\_id, true\_name, mob\_phone,  
 address, prov\_id, city\_id, area\_id, area\_info,  
 new RequestCallBack<String>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 mBtnCommic.setEnabled(true);  
 if (result.netMsg.success) {  
 mBtnCommic.setEnabled(false);  
 setResult(RESULT\_OK);  
 ((Activity) context).finish();  
 }  
 }  
 });  
 }  
 }  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (resultCode != RESULT\_OK) {  
 return;  
 }  
 switch (requestCode) {  
 case REQUESTCODE\_SELECTAREA:  
 if (data != null) {  
 try {  
 mTvAddress.setText(data.getStringExtra("result\_address"));  
 ShopAreaInfoBean provArea = (ShopAreaInfoBean) data.getSerializableExtra("provArea");  
 ShopAreaInfoBean cityArea = (ShopAreaInfoBean) data.getSerializableExtra("cityArea");  
 ShopAreaInfoBean areaArea = (ShopAreaInfoBean) data.getSerializableExtra("areaArea");  
 prov\_id = provArea.getItemId();  
 city\_id = cityArea.getItemId();  
 area\_id = areaArea.getItemId();  
 } catch (Exception e) {  
 LogOut.d(TAG, "选择 区域成功" + e.toString());  
 }  
 }  
 break;  
 default:  
 break;  
 }  
 }  
}

package com.heiyue.project.ui;  
import android.app.Activity;  
import android.content.Context;  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.net.Uri;  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.support.v4.app.FragmentActivity;  
import android.support.v4.view.PagerAdapter;  
import android.support.v4.view.ViewPager;  
import android.text.TextUtils;  
import android.view.KeyEvent;  
import android.view.View;  
import android.view.View.OnLongClickListener;  
import android.view.ViewGroup;  
import android.view.ViewGroup.LayoutParams;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
import com.heiyue.bean.IBanner;  
import com.heiyue.photoview.PhotoView;  
import com.heiyue.photoview.PhotoViewAttacher;  
import com.heiyue.project.bean.PubBanner;  
import com.heiyue.project.config.CacheManager;  
import com.heiyue.ui.ImageViewTouch;  
import com.heiyue.util.ImageTool;  
import com.heiyue.util.LogOut;  
import com.heiyue.util.Md5Util;  
import com.nostra13.universalimageloader.core.ImageLoader;  
import com.nostra13.universalimageloader.core.assist.FailReason;  
import com.nostra13.universalimageloader.core.listener.ImageLoadingListener;  
import com.qixiaofu.qxf.R;  
import java.io.File;  
import java.io.Serializable;  
import java.util.ArrayList;  
import java.util.List;  
public class PhotoViewActivity extends FragmentActivity {  
 public static final String IMAGE\_URLS\_ARR = "img\_urls";  
 public static final String IMAGE\_POSITION = "img\_position";  
 public static final String IMAGE\_TYPE\_HEAD = "img\_type\_head";  
 public static List<? extends IBanner> imageList;  
 public static List<String> imageDeleteList;  
 private static int requestCode;  
  
 private int currentPos;  
 private TextView mTvSize;  
 private Activity context;  
 private ImageViewTouch mBtBack;  
 private ImageViewTouch mBtDelete;  
 private PhotoAdapter mPhotoAdapter;  
 private RelativeLayout mLayoutBar;  
 private ViewPager mViewPager;  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 LogOut.d("PhotoViewActivity.onCreate()");  
 setContentView(R.layout.pub\_activity\_photo\_view);  
 context = this;  
 mViewPager = (ViewPager) findViewById(R.id.viewpager);  
 mLayoutBar = (RelativeLayout) findViewById(R.id.layoutBar);  
 mBtBack = (ImageViewTouch) findViewById(R.id.btBack);  
 mBtDelete = (ImageViewTouch) findViewById(R.id.btDelete);  
 mTvSize = (TextView) findViewById(R.id.tvSize);  
 currentPos = getIntent().getIntExtra(IMAGE\_POSITION, 0);  
 mTvSize.setText("");  
 if (imageList != null && imageList.size() > 0) {  
 mPhotoAdapter = new PhotoAdapter();  
 mViewPager.setAdapter(mPhotoAdapter);  
 if (currentPos >= 0 && currentPos < imageList.size() && imageList.size() > 1) {  
 mTvSize.setText((currentPos + 1) + " / " + imageList.size());  
 mViewPager.setCurrentItem(currentPos);  
 }  
 }  
 imageDeleteList = new ArrayList<String>();  
 mViewPager.setOnPageChangeListener(new ViewPager.OnPageChangeListener() {  
 @Override  
 public void onPageSelected(int position) {  
 currentPos = position;  
 mTvSize.setText((position + 1) + " / " + imageList.size());  
 }  
  
 @Override  
 public void onPageScrolled(int arg0, float arg1, int arg2) {  
 }  
  
 @Override  
 public void onPageScrollStateChanged(int arg0) {  
 }  
 });  
 mBtBack.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 setPhotoResult();  
 PhotoViewActivity.this.finish();  
 }  
 });  
 mBtDelete.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if(imageList.size() > 0){  
 if(imageList.size() == 1){  
 currentPos = 0;  
 }  
 imageDeleteList.add(imageList.get(currentPos).getUrl());  
 imageList.remove(currentPos);  
 mPhotoAdapter.notifyDataSetChanged();  
 mViewPager.setAdapter(mPhotoAdapter);  
 if(imageList.size() != 0){  
 if(imageList.size() == 1){  
 currentPos = 0;  
 }  
 mTvSize.setText((currentPos + 1) + " / " + imageList.size());  
 mViewPager.setCurrentItem(currentPos);  
 }else{

setPhotoResult();  
 PhotoViewActivity.this.finish();  
 }  
 LogOut.d(""+(currentPos));  
 }  
 }  
 });  
 if (requestCode != 0){  
 mBtDelete.setVisibility(View.VISIBLE);  
 }  
 }  
 @Override  
 public boolean onKeyDown(int keyCode, KeyEvent event) {  
 if(keyCode == KeyEvent.KEYCODE\_BACK){  
 LogOut.d("系统返回按钮");  
 setPhotoResult();  
 }  
 return super.onKeyDown(keyCode, event);  
 }  
 private void setPhotoResult() {  
 Intent intent = new Intent();  
 intent.putExtra("imageList", (Serializable) imageDeleteList);  
 setResult(RESULT\_OK, intent);  
 }  
 public static void startActivity(Activity context, int position, List<? extends IBanner> imgs, int requestCode) {  
 if (position < 0) {  
 position = 0;  
 }  
 Intent intent = new Intent(context, PhotoViewActivity.class);  
 intent.putExtra(IMAGE\_POSITION, position);  
 PhotoViewActivity.imageList = imgs;  
 PhotoViewActivity.requestCode = requestCode;  
 context.startActivityForResult(intent, requestCode);  
 }  
 public static void startActivity(Context context, int position, List<? extends IBanner> imgs) {  
 if (position < 0) {  
 position = 0;  
 }  
 Intent intent = new Intent(context, PhotoViewActivity.class);  
 intent.putExtra(IMAGE\_POSITION, position);  
 PhotoViewActivity.imageList = imgs;  
 context.startActivity(intent);  
 }  
 public static void startActivity(Fragment fragment, int position, List<? extends IBanner> imgs) {  
 if (position < 0) {  
 position = 0;  
 }  
 Intent intent = new Intent(fragment.getActivity(), PhotoViewActivity.class);  
 intent.putExtra(IMAGE\_POSITION, position);  
 PhotoViewActivity.imageList = imgs;  
 fragment.startActivity(intent);  
 }  
 public static void startActivity(Context context, String imageUrl) {  
 Intent intent = new Intent(context, PhotoViewActivity.class);  
 intent.putExtra(IMAGE\_POSITION, 0);  
 List<PubBanner> childs = new ArrayList<PubBanner>();  
 PubBanner child = new PubBanner();  
 child.img = imageUrl;  
 childs.add(child);  
 PhotoViewActivity.imageList = childs;  
 context.startActivity(intent);  
 }  
 class PhotoAdapter extends PagerAdapter {  
 public PhotoAdapter() {  
 }  
 @Override  
 public int getCount() {  
 if (imageList != null) {  
 return imageList.size();  
 }  
 return 0;  
 }  
 @Override  
 public View instantiateItem(ViewGroup container, int position) {  
 final IBanner item = imageList.get(position);  
 PhotoView photoView = new PhotoView(container.getContext());  
 container.addView(photoView, LayoutParams.MATCH\_PARENT, LayoutParams.MATCH\_PARENT);  
 if (item != null) {  
 String photoPath = item.getUrl();  
 if (!TextUtils.isEmpty(photoPath)) {  
 if (photoPath.startsWith("http://") || photoPath.startsWith("https://")) {  
 ImageLoader.getInstance().displayImage(photoPath, photoView, CacheManager.getDefaultDisplay());  
 } else {  
 ImageLoader.getInstance().displayImage(Uri.fromFile(new File(photoPath)).toString(), photoView, CacheManager.getLocalDisplayerOptions4Upload());  
 }  
 } else if (item.getResource() > 0) {  
 ImageLoader.getInstance().displayImage("drawable://" + item.getResource(), photoView, CacheManager.getLocalDisplayerOptions4Upload());  
 }  
 }  
 photoView.setOnViewTapListener(new PhotoViewAttacher.OnViewTapListener() {  
 @Override  
 public void onViewTap(View view, float x, float y) {  
 setPhotoResult();  
 finish();

if (mLayoutBar.getVisibility() == View.VISIBLE) {  
 mLayoutBar.setVisibility(View.GONE);  
 } else {  
 mLayoutBar.setVisibility(View.VISIBLE);  
 setPhotoResult();  
 PhotoViewActivity.this.finish();  
 }  
 }  
 });  
 photoView.setOnLongClickListener(new OnLongClickListener() {  
 @Override  
 public boolean onLongClick(View v) {  
 // showPopuWindow(item.getPhotoPath());  
 currentUrl = item.getUrl();  
 SavePhotoPopActivity.open(context, 1);  
 return true;  
 }  
 });  
 return photoView;  
 }  
 @Override  
 public void destroyItem(ViewGroup container, int position, Object object) {  
 container.removeView((View) object);  
 }  
 @Override  
 public boolean isViewFromObject(View view, Object object) {  
 return view == object;  
 }  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (resultCode == RESULT\_OK) {  
 switch (requestCode) {  
 case 1:  
 if (SavePhotoPopActivity.isSave) {  
 save(currentUrl);  
 }  
 break;  
 default:  
 break;  
 }  
 }  
 }  
 private String currentUrl;  
 private void save(final String url) {  
 String displayUrl = url;  
 if (!url.startsWith("http://")) {  
 displayUrl = Uri.fromFile(new File(url)).toString();  
 }  
 ImageLoader.getInstance().loadImage(displayUrl,  
 new ImageLoadingListener() {  
 @Override  
 public void onLoadingStarted(String imageUri, View view) {  
 }  
 @Override  
 public void onLoadingFailed(String imageUri, View view,  
 FailReason failReason) {  
 }  
 @Override  
 public void onLoadingComplete(String imageUri, View view,  
 Bitmap loadedImage) {  
 try {  
 String dir = CacheManager.getQrcodeDir(getApplicationContext());  
 ImageTool.saveJPGE\_After(loadedImage, dir + Md5Util.md5(url) + ".jpg");  
 ImageTool.scanPhotos(dir + Md5Util.md5(url) + ".jpg", getApplicationContext());  
 Toast.makeText(getApplicationContext(), Toast.LENGTH\_SHORT).show();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 @Override  
 public void onLoadingCancelled(String imageUri, View view) {  
  
 }  
 });  
 }  
}

package com.heiyue.project.ui;  
import android.app.ActionBar.LayoutParams;  
import android.app.Activity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.view.View.OnClickListener;  
import com.bigkoo.pickerview.adapter.ArrayWheelAdapter;  
import com.bigkoo.pickerview.lib.WheelView;  
import com.bigkoo.pickerview.listener.OnItemSelectedListener;  
import com.google.gson.Gson;  
import com.google.gson.reflect.TypeToken;  
import com.heiyue.project.bean.CityBean;  
import com.heiyue.project.bean.WheelBean;  
import com.heiyue.project.dao.ServerDao;  
import com.heiyue.util.StringUtil;  
import com.qixiaofu.qxf.R;  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.List;

public class SetAddressPopActivity extends Activity {  
 public static void startActivity(Activity activity, String defaultCity,  
 String defaultArea, int requestCode) {  
 Intent intent = new Intent(activity, SetAddressPopActivity.class);  
 intent.putExtra("defaultCity", defaultCity);  
 intent.putExtra("defaultArea", defaultArea);  
 activity.startActivityForResult(intent, requestCode);  
 }  
 private WheelView wvCity;  
 private WheelView wvArea;  
 Calendar c = Calendar.getInstance();  
 private ServerDao dao;  
 private String defaultCity;  
 private String defaultArea;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.include\_wheeview\_2);  
 getWindow().setLayout(LayoutParams.MATCH\_PARENT,  
 LayoutParams.WRAP\_CONTENT);  
 wvCity = (WheelView) findViewById(R.id.wheeView\_left);  
 wvArea = (WheelView) findViewById(R.id.wheeView\_right);  
 defaultCity = getIntent().getStringExtra("defaultCity");  
 defaultArea = getIntent().getStringExtra("defaultArea");  
 dao = new ServerDao(this);  
 getCitysDate();  
 initData();  
 }  
 private int checkDefaultCityIndex() {  
 int index = 0;  
 if (!TextUtils.isEmpty(defaultCity) && listCitys != null  
 && listCitys.size() > 0) {  
 for (int i = 0; i < listCitys.size(); i++) {  
 if (defaultCity.equals(listCitys.get(i).getPickerViewText())) {  
 index = i;  
 }  
 }  
 }  
 return index;  
 }  
 private int checkDefaultAreaIndex(String city) {  
 int index = 0;  
 if (!TextUtils.isEmpty(defaultArea) && listArea != null && listArea.size() > 0) {  
 for (int i = 0; i < listArea.size(); i++) {  
 if (defaultArea.equals(listArea.get(i).getPickerViewText())) {  
 index = i;  
 }  
 }  
 }  
 return index;  
 }  
 private ArrayWheelAdapter<WheelBean> adapterCity;  
 private ArrayWheelAdapter<WheelBean> adapterArea;  
  
 private void initData() {  
 if (mDate == null || mDate.size() == 0) {  
 return;  
 }  
 getCity();  
 adapterCity = new ArrayWheelAdapter<WheelBean>(listCitys);  
 wvCity.setAdapter(adapterCity);  
 wvCity.setCurrentItem(checkDefaultCityIndex());  
 getArea(defaultCity);  
 adapterArea = new ArrayWheelAdapter<WheelBean>(listArea);  
 wvArea.setAdapter(adapterArea);  
 wvArea.setCurrentItem(checkDefaultAreaIndex(defaultCity));  
 wvCity.setOnItemSelectedListener(new OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(int index) {  
 defaultCity = adapterCity.getItem(index).getPickerViewText();  
 getArea(defaultCity);  
 adapterArea = new ArrayWheelAdapter<WheelBean>(listArea);  
 wvArea.setAdapter(adapterArea);  
 wvArea.setCurrentItem(checkDefaultAreaIndex(defaultCity));  
 }  
 });  
 findViewById(R.id.linear\_parent).setOnClickListener(  
 new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
 findViewById(R.id.tvcacle).setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
 findViewById(R.id.tvok).setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent data = new Intent();  
 String cityStr = adapterCity.getItem(wvCity.getCurrentItem())  
 .getPickerViewText();  
 String areaStr = adapterArea.getItem(wvArea.getCurrentItem())  
 .getPickerViewText();  
 data.putExtra("cityStr", cityStr);  
 data.putExtra("areaStr", areaStr);  
 setResult(RESULT\_OK, data);  
 finish();  
 }  
 });  
 }  
 public List<CityBean> mDate;  
 private void getCitysDate() {  
 mDate = dao.getAddressData();  
 if (mDate == null || mDate.size() == 0) {  
 String citys = StringUtil.getTextFromAssert(this, "city.txt");  
 mDate = new Gson().fromJson(citys, new TypeToken<List<CityBean>>() {  
 }.getType());  
 if (mDate != null && mDate.size() > 0) {  
 dao.saveAddressData(mDate);  
 }  
 }  
 }  
  
 private ArrayList<WheelBean> listArea;  
 private void getArea(String city) {  
 if (TextUtils.isEmpty(city)) {  
 city = listCitys.get(0).getPickerViewText();  
 }  
 listArea = dao.getAreas(city);  
 if (listArea == null || listArea.size() == 0) {  
 listArea = new ArrayList<WheelBean>();  
 String[] area = null;  
 for (CityBean c : mDate) {  
 if (city.equals(c.city)) {  
 area = c.area;  
 }  
 }  
 if (area == null || area.length == 0) {  
 return;  
 }  
 for (String a : area) {  
 WheelBean bean = new WheelBean(a);  
 listArea.add(bean);  
 }  
 if (listArea.size() > 0) {  
 dao.saveAreas(listArea, city);  
 }  
 }  
 }  
 private ArrayList<WheelBean> listCitys;  
 private void getCity() {  
 listCitys = dao.getCitys();  
 if (listCitys == null || listCitys.size() == 0) {  
 listCitys = new ArrayList<WheelBean>();  
 for (CityBean c : mDate) {  
 WheelBean bean = new WheelBean(c.city);  
 listCitys.add(bean);  
 }  
 if (listCitys.size() > 0) {  
 dao.saveCitys(listCitys);  
 }  
 }  
 }  
}

package com.heiyue.project.ui;  
import java.util.ArrayList;  
import java.util.List;  
import android.app.Activity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.view.View.OnClickListener;  
import android.view.ViewGroup.LayoutParams;  
import android.widget.TextView;  
import com.bigkoo.pickerview.adapter.ArrayWheelAdapter;  
import com.bigkoo.pickerview.lib.WheelView;  
import com.heiyue.project.bean.OrderPushInfo.ServiceForm;  
import com.qixiaofu.qxf.R;  
public class SetOrderFormPopActivity extends Activity {  
 public static void startActivityforResult(Activity activity,  
 String defaultForm, int requestCode,List<ServiceForm> list) {  
 Intent intent = new Intent(activity, SetOrderFormPopActivity.class);  
 intent.putExtra("defaultForm", defaultForm);  
 mList = (ArrayList<ServiceForm>) list;  
 activity.startActivityForResult(intent, requestCode);  
 }  
 private WheelView wheelForm;  
 private TextView tvcacle, tvok;  
 private String defaultForm;  
 private static ArrayList<ServiceForm> mList;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.include\_wheeview\_1);  
 getWindow().setLayout(LayoutParams.MATCH\_PARENT,  
 LayoutParams.WRAP\_CONTENT);  
 getList();  
 initView();  
 initData();  
 }  
 private void initView() {  
 wheelForm = (WheelView) findViewById(R.id.wheeView\_center);  
 tvcacle = (TextView) findViewById(R.id.tvcacle);  
 tvok = (TextView) findViewById(R.id.tvok);  
 }  
 private ArrayWheelAdapter<ServiceForm> adapter;  
 private ArrayList<ServiceForm> formList;  
 private void getList() {  
 formList = new ArrayList<ServiceForm>();  
 formList.addAll(mList);  
 }  
 private int checkDefaultAgeIndex() {  
 int index = 0;  
 if (!TextUtils.isEmpty(defaultForm)) {  
 for (int i = 0; i < formList.size(); i++) {  
 ServiceForm bean = formList.get(i);  
 if (defaultForm.equals(bean.getField\_name())) {  
 index = i;  
 }  
 }  
 }  
 return index;  
 }  
 private void initData() {  
 defaultForm = getIntent().getStringExtra("defaultForm");  
 if (formList == null) {  
 return;  
 }  
 adapter = new ArrayWheelAdapter<ServiceForm>(formList);  
 wheelForm.setAdapter(adapter);  
 wheelForm.setCurrentItem(checkDefaultAgeIndex());  
 tvcacle.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
 findViewById(R.id.layout\_parent).setOnClickListener(  
 new OnClickListener() {  
  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
 tvok.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 ServiceForm form = (ServiceForm) adapter.getItem(wheelForm  
 .getCurrentItem());  
 if (form == null) {  
 return;  
 }  
 Intent data = new Intent();  
 data.putExtra("form", form.getField\_name());  
 data.putExtra("formid", form.getField\_value());  
 setResult(RESULT\_OK, data);  
 finish();  
 }  
 });  
 }  
}

package com.heiyue.project.ui.index;  
import android.app.Activity;  
import android.content.Context;  
import android.content.Intent;  
import android.support.v4.app.Fragment;  
import android.text.Editable;  
import android.text.TextUtils;  
import android.text.TextWatcher;  
import android.view.KeyEvent;  
import android.view.View;  
import android.view.View.OnClickListener;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.LinearLayout;  
import android.widget.ListView;  
import android.widget.PopupWindow;  
import android.widget.TextView;  
import com.amap.map3d.comm.poisearch.PoiKeywordSearchActivity;  
import com.bigkoo.pickerview.TimePickerView;  
import com.google.gson.Gson;  
import com.heiyue.net.NetResponse;  
import com.heiyue.net.RequestCallBack;  
import com.heiyue.project.base.BaseActivity;  
import com.heiyue.project.bean.MyOrderInfoBean;  
import com.heiyue.project.bean.OrderPushInfo;  
import com.heiyue.project.bean.OrderPushInfo.ServiceForm;  
import com.heiyue.project.bean.OrderPushInfo.ServiceType;  
import com.heiyue.project.bean.OrderSubmissionData;  
import com.heiyue.project.bean.ServiceSectorBean;  
import com.heiyue.project.ui.SetOrderFormPopActivity;  
import com.heiyue.project.ui.SetOrderSectorOtherActivity;  
import com.heiyue.project.ui.SetOrderTypeOrSmallStorePopActivity;  
import com.heiyue.project.ui.mine.BrandInputActivity;  
import com.heiyue.project.util.EditTextInputStyleUtil;  
import com.heiyue.project.util.PubDialogUtil;  
import com.heiyue.util.LogOut;  
import com.heiyue.util.TimeUtil;  
import com.qixiaofu.qxf.R;  
import java.util.Calendar;  
import java.util.Date;  
import java.util.List;  
import butterknife.BindView;  
public class OrderPush1Activity {  
 private static final int REQUESTCODE\_SECTOR = 6;  
 private OrderPushInfo orderPushInfo;  
 public static Activity instance = null;  
 private TimePickerView mTimePickerView;  
 private boolean isStartTime = false;  
 private PubDialogUtil mPubDialogUtil;  
 private List<ServiceSectorBean> mSelectedClassList;  
 public static void startActivity(Context context) {  
 Intent intent = new Intent(context, OrderPush1Activity.class);  
 context.startActivity(intent);  
 }  
 public static void startActivity(Activity activity, int requestCode) {  
 Intent intent = new Intent(activity, OrderPush1Activity.class);  
 // context.startActivity(intent);  
 activity.startActivityForResult(intent, requestCode);  
 }  
 public static void startActivity(Fragment fragment) {  
 Intent intent = new Intent(fragment.getActivity(), OrderPush1Activity.class);  
 fragment.startActivity(intent);  
 }  
 public static void enterActivityForResult(Activity activity, String id, int requestCode) {  
 Intent intent = new Intent(activity, OrderPush1Activity.class);  
 intent.putExtra("id", id);  
 activity.startActivityForResult(intent, requestCode);  
 }S  
 public static void enterActivityForResult(Activity activity, boolean isBuDan, int requestCode) {  
 Intent intent = new Intent(activity, OrderPush1Activity.class);  
 intent.putExtra("isBuDan", isBuDan);  
  
 activity.startActivityForResult(intent, requestCode);  
  
 }  
 public static void enterActivityForResult(Fragment fragment, boolean isBuDan, int requestCode) {  
 Intent intent = new Intent(fragment.getActivity(), OrderPush1Activity.class);  
 intent.putExtra("isBuDan", isBuDan);  
 fragment.startActivityForResult(intent, requestCode);  
 }  
 @Override  
 protected int setContentLayoutView() {  
 return R.layout.order\_activity\_push\_1;  
 }  
 @BindView(R.id.tv\_form)  
 public TextView mTvForm;  
 @BindView(R.id.tv\_type)  
 public TextView mTvType;  
 @BindView(R.id.tv\_area)  
 public TextView mTvArea;  
 @BindView(R.id.tv\_time\_start)  
 public TextView mTvTimeStart;  
 @BindView(R.id.tv\_time\_end)  
 public TextView mTvTimeEnd;  
 @BindView(R.id.text\_area\_tech)  
 public TextView mTextAreaTech;  
 @BindView(R.id.tv\_brand)  
 public TextView mTvBrand;  
 @BindView(R.id.et\_project\_name)  
 public EditText mEtProjectName;  
 @BindView(R.id.et\_num)  
 public EditText mEtNum;  
 @BindView(R.id.et\_unit)  
 public EditText mEtUnit;  
 @BindView(R.id.et\_price)  
 public EditText mEtPrice;  
 @BindView(R.id.ll\_project)  
 public LinearLayout llProjectName;  
 @BindView(R.id.ll\_form)  
 public LinearLayout llForm;  
 @BindView(R.id.ll\_type)  
 public LinearLayout llType;  
 @BindView(R.id.ll\_area)  
 public LinearLayout llArea;  
 @BindView(R.id.ll\_time\_start)  
 public LinearLayout llTimeStart;  
 @BindView(R.id.ll\_time\_end)  
 public LinearLayout llTimeEnd;  
 @BindView(R.id.ll\_area\_tech)  
 public LinearLayout llAreaTech;  
 @BindView(R.id.ll\_brand)  
 public LinearLayout llBrand;  
 @BindView(R.id.ll\_order\_push)  
 public LinearLayout llOrderPush;  
 private LinearLayout llAreaTechOther;  
 private TextView mTvAreaTechOther;  
 private ListView lvOrderPush;  
 private Button btnPopupOK;  
 private Button btnPopupNO;  
 private PopupWindow popupWindow;  
 private int mSelectedPosition;  
 private String yearStart;  
 private String monthStart;  
 private String dayStart;

private OrderSubmissionData submissionData;  
 @Override  
 protected void initViews() {  
 instance = this;  
 submissionData = new OrderSubmissionData();  
 llForm.setOnClickListener(this);  
 llType.setOnClickListener(this);  
 llArea.setOnClickListener(this);  
 llTimeStart.setOnClickListener(this);  
 llTimeEnd.setOnClickListener(this);  
 llAreaTech.setOnClickListener(this);  
 llBrand.setOnClickListener(this);  
 // 约束输入的价格的格式  
 EditTextInputStyleUtil.setEditTextInputPriceStyle(mEtPrice);  
 // 时间选择器  
 mTimePickerView = new TimePickerView(this, TimePickerView.Type.YEAR\_MONTH\_DAY\_HOURS);  
 // 控制时间范围  
 Calendar calendar = Calendar.getInstance();  
 mTimePickerView.setRange(calendar.get(Calendar.YEAR), calendar.get(Calendar.YEAR) + 30);  
 mTimePickerView.setTime(new Date());  
 mTimePickerView.setCyclic(false);  
 mTimePickerView.setCancelable(true);  
 // 时间选择后回调  
 mTimePickerView.setOnTimeSelectListener(new TimePickerView.OnTimeSelectListener() {  
 @Override  
 public void onTimeSelect(Date date) {  
 if (isStartTime) {  
 mTvTimeStart.setText(TimeUtil.getTime(date));  
 String startTime = TimeUtil.getTimeStamp(mTvTimeStart.getText().toString(), "yyyy-MM-dd HH:mm") + "";  
 submissionData.setService\_stime(startTime);  
 } else {  
 mTvTimeEnd.setText(TimeUtil.getTime(date));  
 String endTime = TimeUtil.getTimeStamp(mTvTimeEnd.getText().toString(), "yyyy-MM-dd HH:mm") + "";  
 submissionData.setService\_etime(endTime);  
 }  
 }  
 });  
 final LinearLayout.LayoutParams layoutParams = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP\_CONTENT, LinearLayout.LayoutParams.MATCH\_PARENT);  
 mEtProjectName.addTextChangedListener(new TextWatcher() {  
 @Override  
 public void beforeTextChanged(CharSequence s, int start, int count, int after) {  
 }  
 @Override  
 public void onTextChanged(CharSequence s, int start, int before, int count) {  
 if(TextUtils.isEmpty(s.toString())){  
 LogOut.d("输入无："+s);  
 final LinearLayout.LayoutParams layoutParams = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP\_CONTENT, LinearLayout.LayoutParams.MATCH\_PARENT);  
 layoutParams.width = LinearLayout.LayoutParams.WRAP\_CONTENT;  
 mEtProjectName.setLayoutParams(layoutParams);  
 }else{  
 LogOut.d("输入有："+s);  
 layoutParams.width = LinearLayout.LayoutParams.MATCH\_PARENT;  
 mEtProjectName.setLayoutParams(layoutParams);  
 }  
 }  
 @Override  
 public void afterTextChanged(Editable s) {  
 }  
 });  
 LogOut.d("onResume()，显示项目信息");  
 showOrderPushInfo();  
 }  
 private String id;  
 private MyOrderInfoBean myOrdersInfor;  
 private boolean isBuDan;  
 private String serviceIdList, otherServiceIdList;  
 @Override  
 protected void initData() {  
 id = getIntent().getStringExtra("id");  
 isBuDan = getIntent().getBooleanExtra("isBuDan", false);  
 if (!TextUtils.isEmpty(id)) {  
 getMyOrderInfor();  
 }  
 if (isBuDan && TextUtils.isEmpty(id)) {  
 tvTitle.setText("add");  
 } else {  
 tvTitle.setText("send");  
 }  
 getNetData();  
 }  
  
 private void getNetData() {  
 dao.getOrderPushInfo(new RequestCallBack<OrderPushInfo>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<OrderPushInfo> result) {  
 if (result.netMsg.success) {  
 orderPushInfo = result.content;  
 }  
 }  
 });  
 }  
 @Override  
 public void onClick(View v) {  
 if (orderPushInfo == null) {  
 showToast("服务器返回数据为空");  
 return;  
 }  
 switch (v.getId()) {  
 case R.id.ll\_form:  
 List<ServiceForm> service\_form = orderPushInfo.getService\_form();  
 SetOrderFormPopActivity.startActivityforResult(this, "现场服务", 1, service\_form);  
 break;  
 case R.id.ll\_type:  
 List<ServiceType> service\_type = orderPushInfo.getService\_type();  
 SetOrderTypeOrSmallStorePopActivity.startChooseServiceType(this, "服务", 2, service\_type);  
 break;  
 case R.id.ll\_area:

PoiKeywordSearchActivity.startActivity(this, 3);  
 break;  
 case R.id.ll\_time\_start:  
 isStartTime = true;  
 mTimePickerView.show();  
 break;  
 case R.id.ll\_time\_end:  
 isStartTime = false;  
 mTimePickerView.show();  
 break;  
 case R.id.ll\_area\_tech:  
 List<ServiceSectorBean> service\_sector = orderPushInfo.getService\_sector();  
 if (!service\_sector.get(service\_sector.size() - 1).gc\_name.equals("其他")) {  
 ServiceSectorBean ss = new ServiceSectorBean();  
 ss.gc\_id = "";  
 ss.gc\_image = "";  
 ss.gc\_name = "其他";  
 service\_sector.add(ss);  
 }  
 SetOrderSectorOtherActivity.startActivityforResult(this, "", "", 6, SetOrderSectorOtherActivity.getSelectItemList(serviceIdList), service\_sector);  
  
 SetOrderSectorOtherActivity.startActivityforResult(this, "", "", REQUESTCODE\_SECTOR, mSelectedClassList, service\_sector);  
 break;  
 case R.id.ll\_brand:  
 String brand = mTvBrand.getText().toString().trim();  
 BrandInputActivity.enterActivityForResult((Activity) context, brand, 7);  
 break;  
 case R.id.ll\_area\_tech\_other:

List<ServiceSector> service\_sector\_other = orderPushInfo.getService\_sector();  
 if (!service\_sector\_other.get(service\_sector\_other.size() - 1).gc\_name.equals("其他")) {  
 ServiceSector ss = new ServiceSector();  
 ss.gc\_id = "";  
 ss.gc\_image = "";  
 ss.gc\_name = "其他";  
 service\_sector\_other.add(ss);  
 }  
 SetOrderSectorOtherActivity.startActivityforResult(this, "", "", 8, SetOrderSectorOtherActivity.getSelectItemList(otherServiceIdList), service\_sector\_other);  
 break;  
 case R.id.action\_bar\_left\_btn:

check();  
 break;  
 default:  
 break;  
 }  
 }  
 @Override  
 public boolean onKeyDown(int keyCode, KeyEvent event) {  
 if (keyCode == KeyEvent.KEYCODE\_BACK ) {  
 check();  
 }  
 return false;  
 }  
 private void showOrderPushInfo() {  
 String infoCache = dao.getOrderPrePushInfoCache();  
  
 if(!TextUtils.isEmpty(infoCache)){  
 Gson gson = new Gson();  
 submissionData = gson.fromJson(infoCache, OrderSubmissionData.class);  
 mEtProjectName.setText(submissionData.getProjectName());  
 mEtProjectName.setSelection(submissionData.getProjectName().length());  
 mTvForm.setText(submissionData.getServiceFormVal());  
 mTvType.setText(submissionData.getServiceTypeVal());  
 mTvArea.setText(submissionData.getService\_address());  
 mTvTimeStart.setText(TimeUtil.getSimpleDate(submissionData.getService\_stime(), "yyyy-MM-dd HH:mm"));  
 mTvTimeEnd.setText(TimeUtil.getSimpleDate(submissionData.getService\_etime(), "yyyy-MM-dd HH:mm"));  
  
 mTextAreaTech.setText(submissionData.getServiceSectorVal());  
 mTvBrand.setText(submissionData.getService\_brand());  
 mTvAreaTechOther.setText(submissionData.getOther\_service\_sector());  
 mEtPrice.setText(submissionData.getService\_price());  
 mEtNum.setText(submissionData.getNumber());  
 mEtUnit.setText(submissionData.getNumber\_unit());  
 serviceIdList= submissionData.getServiceSectorId();  
 }  
 }  
 private void check(){  
 if(!TextUtils.isEmpty(mEtProjectName.getText().toString().trim()) || !TextUtils.isEmpty(mTvForm.getText().toString().trim()) || !TextUtils.isEmpty(mTvType.getText().toString().trim())){  
 showDialog("您是否保存当前信息？");  
 }else{  
 OrderPush1Activity.this.finish();  
 }  
 }  
 public void showDialog(String message) {  
 mPubDialogUtil = new PubDialogUtil(OrderPush1Activity.this);  
 mPubDialogUtil.showDialogtitleText("操作提示", message, new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 saveOrderInfoCache();  
 mPubDialogUtil.dismiss();  
 OrderPush1Activity.this.finish();  
 }  
 }, new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 dao.clearOrderPrePushInfoCache();  
 mPubDialogUtil.dismiss();  
 OrderPush1Activity.this.finish();  
 }  
 }, "取消", "保存");  
 }

private void saveOrderInfoCache() {  
 if(!TextUtils.isEmpty(mEtProjectName.getText().toString().trim())){  
 submissionData.setProjectName(mEtProjectName.getText().toString().trim());  
 }  
 if(!TextUtils.isEmpty(mTextAreaTech.getText().toString().trim())){  
 submissionData.setTitle(mTextAreaTech.getText().toString().trim());  
 }  
 if(!TextUtils.isEmpty(mTvBrand.getText().toString().trim())){  
 submissionData.setService\_brand(mTvBrand.getText().toString().trim());  
 }  
 if(!TextUtils.isEmpty(mEtNum.getText().toString().trim())){  
 submissionData.setNumber(mEtNum.getText().toString().trim());  
 }  
 if(!TextUtils.isEmpty(mEtPrice.getText().toString().trim())){  
 submissionData.setService\_price(mEtPrice.getText().toString().trim());  
 }  
 if(!TextUtils.isEmpty(mEtUnit.getText().toString().trim())){  
 submissionData.setNumber\_unit(mEtUnit.getText().toString().trim());  
 }  
 if(submissionData != null){  
 Gson gson = new Gson();  
 String infoCache = gson.toJson(submissionData);

LogOut.d(""+infoCache);  
 dao.saveOrderPrePushInfoCache(infoCache);  
 }  
 }  
 private void getMyOrderInfor() {  
 dao.getMyOrderInforRet(id, new RequestCallBack<MyOrderInfoBean>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<MyOrderInfoBean> result) {  
 if (result.netMsg.success) {  
 myOrdersInfor = result.content;  
 showMyOrderInfor();  
 }  
 }  
 });  
 }  
 private void showMyOrderInfor() {  
 if (myOrdersInfor == null) {  
 return;  
 }  
 mEtProjectName.setText(myOrdersInfor.entry\_name);  
 mEtProjectName.setSelection(myOrdersInfor.entry\_name.length());  
 mTvForm.setText(myOrdersInfor.service\_form);  
 mTvType.setText(myOrdersInfor.service\_type);  
 mTvArea.setText(myOrdersInfor.service\_address);  
 mTvTimeStart.setText(TimeUtil.getSimpleDate(myOrdersInfor.service\_stime, "yyyy-MM-dd HH:mm"));  
 mTvTimeEnd.setText(TimeUtil.getSimpleDate(myOrdersInfor.service\_etime, "yyyy-MM-dd HH:mm"));  
 mTextAreaTech.setText(myOrdersInfor.service\_sector);  
 mTvBrand.setText(myOrdersInfor.service\_brand);  
 mTvAreaTechOther.setText(myOrdersInfor.other\_service\_sector);  
 mEtPrice.setText(myOrdersInfor.service\_price);  
 mEtNum.setText(myOrdersInfor.number);  
 mEtUnit.setText(myOrdersInfor.number\_unit);  
 llArea.setEnabled(false);  
 llAreaTech.setEnabled(false);  
 llAreaTechOther.setEnabled(false);  
 llBrand.setEnabled(false);  
 llForm.setEnabled(false);  
 llOrderPush.setEnabled(false);  
 llTimeEnd.setEnabled(false);  
 llTimeStart.setEnabled(false);  
 llType.setEnabled(false);  
 mEtUnit.setEnabled(false);  
 mEtPrice.setEnabled(false);  
 }  
 private boolean commic() {  
 LogOut.d("提交时间：开始：" + submissionData.getService\_stime() + ",结束：" + submissionData.getService\_etime());  
 if (TextUtils.isEmpty(mEtProjectName.getText().toString().trim())) {  
 showToast("请填写项目名称");  
 return false;  
 }  
 if (mEtProjectName.getText().toString().trim().length() > 30) {  
 showToast("项目名称不得超过30字");  
 return false;  
 } else if (TextUtils.isEmpty(mTvForm.getText())) {  
 showToast("请选择服务形式");  
 return false;  
 } lse if (TextUtils.isEmpty(mEtUnit.getText())) {  
 showToast("请选择服务数量单位");  
 return false;  
 } else if (TextUtils.isEmpty(mEtPrice.getText())) {  
 showToast("请选择服务价格");  
 return false;  
 }  
 long start = TimeUtil.getSeconds(mTvTimeStart.getText().toString(), "yyyy-MM-dd HH:mm");  
 long end = TimeUtil.getSeconds(mTvTimeEnd.getText().toString(), "yyyy-MM-dd HH:mm");  
 long current = System.currentTimeMillis() / 1000;  
 LogOut.d("当前时间："+current);  
 if(start < current){  
 showToast("预约开始时间必须在当前时间之后");  
 return false;  
 }  
 if (start > end) {  
 showToast("预约开始时间必须在截止时间之前");  
 return false;  
 }  
 try {  
 double input = Double.parseDouble(mEtPrice.getText().toString());  
 if (input == 0) {  
 showToast("价格不能为0");  
 return false;  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 submissionData.setNumber(mEtNum.getText().toString().trim());  
 submissionData.setService\_price(mEtPrice.getText().toString().trim());  
 submissionData.setProjectName(mEtProjectName.getText().toString().trim());  
 submissionData.setTitle(mTextAreaTech.getText().toString().trim() + "-" + mTvBrand.getText().toString().trim());  
 submissionData.setTitle(mTextAreaTech.getText().toString().trim());  
 submissionData.setNumber\_unit(mEtUnit.getText().toString().trim());  
 saveOrderInfoCache();  
 return true;  
 }  
 @Override  
 protected void setActionBarDetail() {  
 btnRightText.setVisibility(View.VISIBLE);  
 btnRightText.setText("下一步");  
 btnRightText.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (TextUtils.isEmpty(id)) {  
 if (commic()) {  
 if (isBu) {  
 submissionData.setIs\_compe("1");  
 OrderPushPayActivity.enterActivityForResult((Activity) context, submissionData, orderPushInfo == null ? "0" : orderPushInfo.getAvailable\_predeposit(), "0", isBuDan,  
 REQUESTCODE\_TO\_PAY);  
 } else {  
 submissionData.setIs\_compe("0");  
 OrderPush2Activity.startActivity(context, submissionData, orderPushInfo == null ? "0" : orderPushInfo.getTop\_price(),  
 orderPushInfo.getAvailable\_predeposit());  
 }  
 }  
 } else {  
 againRelease();  
 }  
 }  
 });  
 }  
 private void againRelease() {  
 if (myOrdersInfor == null) {  
 showToast("数据异常");  
 return;  
 }  
 if (TextUtils.isEmpty(mTvTimeStart.getText())) {  
 showToast("请选择预约开始时间");  
 return;  
 }  
 if (TextUtils.isEmpty(mTvTimeEnd.getText())) {  
 showToast("请选择预约结束时间");  
 return;  
 }  
 long start = TimeUtil.getSeconds(mTvTimeStart.getText().toString(), "yyyy-MM-dd HH:mm");  
 long end = TimeUtil.getSeconds(mTvTimeEnd.getText().toString(), "yyyy-MM-dd HH:mm");  
 long current = System.currentTimeMillis() / 1000;  
 LogOut.d("当前时间："+current);  
 /\*if(start < current){  
 showToast("预约开始时间必须在当前时间之后");  
 return;  
 }\*/  
 if (start > end) {  
 showToast("预约开始时间必须在结束时间之前");  
 return;  
 }  
 if (TextUtils.isEmpty(mEtNum.getText())) {  
 showToast("请选择服务数量");  
 return;  
 }  
 String startTime = TimeUtil.getTimeStamp(mTvTimeStart.getText().toString(), "yyyy-MM-dd HH:mm") + "";  
 String endTime = TimeUtil.getTimeStamp(mTvTimeEnd.getText().toString(), "yyyy-MM-dd HH:mm") + "";

OrderPush2Activity.enterActivityForResult((Activity) context, id, mEtNum.getText().toString(), startTime, endTime, myOrdersInfor.bill\_desc, myOrdersInfor.image, REQUESTCODE\_ORDER2);  
 }  
 private final int REQUESTCODE\_ORDER2 = 100;// 发单2  
 private final int REQUESTCODE\_TO\_PAY = 10;// 去支付-- 补单的时候用  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (resultCode == RESULT\_OK) {  
 if (data == null) {  
 return;  
 }  
 switch (requestCode) {  
 case 1:  
 String formVal = data.getStringExtra("form");  
 String formId = data.getStringExtra("formid");  
 mTvForm.setText(formVal);  
 submissionData.setServiceFormVal(formVal);  
 submissionData.setServiceFormId(formId);  
 break;  
 case 2:  
 String typeVal = data.getStringExtra("type");  
 String typeId = data.getStringExtra("typeid");  
 mTvType.setText(typeVal);  
 submissionData.setServiceTypeVal(typeVal);  
 submissionData.setServiceTypeId(typeId);  
 break;  
 case 3:  
 String add = PoiKeywordSearchActivity.getData(data);  
 submissionData.setLat(PoiKeywordSearchActivity.getDatalat(data) + "");  
 submissionData.setLng(PoiKeywordSearchActivity.getDatalng(data) + "");  
 submissionData.setService\_city(PoiKeywordSearchActivity.getData(data));  
 submissionData.setService\_address(PoiKeywordSearchActivity.getData(data));  
 if (!TextUtils.isEmpty(add)) {  
 mTvArea.setText(add);  
 }  
 break;  
 case 4:  
 yearStart = data.getStringExtra("yearStr");  
 monthStart = data.getStringExtra("monthStr");  
 dayStart = data.getStringExtra("dayStr");  
 String timeStart = yearStart + monthStart + dayStart;  
 if (!TextUtils.isEmpty(timeStart)) {  
 mTvTimeStart.setText(timeStart);  
 }  
 submissionData.setServiceStimeShow(timeStart);  
 submissionData.setService\_stime(TimeUtil.getSeconds(mTvTimeStart.getText().toString(), "yyyy年MM月dd日") + "");  
 break;  
 case 5:  
 String yearEnd = data.getStringExtra("yearStr");  
 String monthEnd = data.getStringExtra("monthStr");  
 String dayEnd = data.getStringExtra("dayStr");  
 mTvTimeEnd.setText(yearEnd + monthEnd + dayEnd);  
 long end = TimeUtil.getSeconds(mTvTimeEnd.getText().toString(), "yyyy年MM月dd日");  
 end = end + (1 \* 60 \* 60 \* 24 - 60);  
 submissionData.setServiceEtimeShow(yearEnd + monthEnd + dayEnd);  
 submissionData.setService\_etime(end + "");  
 submissionData.setService\_etime(TimeUtil.getSeconds(mTvTimeEnd  
 .getText().toString(), "yyyy年MM月dd日")+ "");  
 break;  
 case 6:  
 mSelectedClassList = (List<ServiceSectorBean>)data.getSerializableExtra("selectedClassList");  
 StringBuffer stringBuffer = new StringBuffer();  
 for(int i = 0; i < mSelectedClassList.size(); i++){  
 ServiceSectorBean pServiceSector = mSelectedClassList.get(i);  
 stringBuffer.append(pServiceSector.gc\_name);  
 if(pServiceSector.list != null && pServiceSector.list.size() > 0){  
 stringBuffer.append("(");  
 for (int j = 0; j < pServiceSector.list.size(); j++){  
 stringBuffer.append(pServiceSector.list.get(j).gc\_name);  
 if(j != pServiceSector.list.size() - 1){  
 stringBuffer.append(",");  
 }  
 }  
 stringBuffer.append(")");  
 }  
 if(i != mSelectedClassList.size() - 1){  
 stringBuffer.append(",");  
 }  
 }  
 mTextAreaTech.setText(""+stringBuffer.toString());  
 serviceIdList = data.getStringExtra("sector");  
 submissionData.setServiceSectorId(serviceIdList);  
 LogOut.d("id名："+serviceIdList);  
 boolean bx = data.getBooleanExtra("isHaveChange", false);  
 if (bx) {  
 String sectorName = data.getStringExtra("sectorname");  
 serviceIdList = data.getStringExtra("sector");  
 mTvAreaTechOther.setText(sectorName);  
 // // 其他服务领域ID  
 submissionData.setOther\_service\_sector(otherServiceIdList);  
 mTextAreaTech.setText(sectorName);  
 submissionData.setServiceSectorVal(sectorName);  
 submissionData.setServiceSectorId(serviceIdList);  
 }  
 break;  
 case 7:  
 String brand = data.getStringExtra("brand");  
 mTvBrand.setText(brand);  
 submissionData.setService\_brand(brand);  
 break;  
 case 8:  
 boolean booleanExtra = data.getBooleanExtra("isHaveChange", false);  
 if (booleanExtra) {  
 String sectorName = data.getStringExtra("sectorname");  
 otherServiceIdList = data.getStringExtra("sector");  
 mTvAreaTechOther.setText(sectorName);  
 // 其他服务领域ID  
 submissionData.setOtherServiceSectorVal(sectorName);  
 submissionData.setOtherServiceSectorId(otherServiceIdList);  
 }  
 break;  
 case REQUESTCODE\_ORDER2:  
 if (data != null) {  
 isHaveChange = data.getBooleanExtra("isHaveChange", false);  
 if (isHaveChange) {  
 setResultCustom();  
 }  
 }  
 break;  
 case REQUESTCODE\_TO\_PAY:  
 finish();  
 break;  
 default:  
 break;  
 }  
 }  
 }  
 @Override  
 protected void onDestroy() {  
 mPubDialogUtil = null;  
 super.onDestroy();  
 }  
}

package com.heiyue.project.ui.index;  
import android.annotation.SuppressLint;  
import android.app.Activity;  
import android.content.Context;  
import android.content.Intent;  
import android.text.TextUtils;  
import android.view.View;  
import android.view.View.OnClickListener;  
import android.widget.CheckBox;  
import android.widget.CompoundButton;  
import android.widget.CompoundButton.OnCheckedChangeListener;  
import android.widget.EditText;  
import android.widget.LinearLayout;  
import android.widget.LinearLayout.LayoutParams;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
import com.heiyue.bean.IBanner;  
import com.heiyue.net.NetResponse;  
import com.heiyue.net.RequestCallBack;  
import com.heiyue.project.adapter.PhotoLvAdapter;  
import com.heiyue.project.base.BaseActivity;  
import com.heiyue.project.bean.AccountBalanceBean;  
import com.heiyue.project.bean.OrderSubmissionData;  
import com.heiyue.project.bean.PhotoBean;  
import com.heiyue.project.bean.ResultImageInfo;  
import com.heiyue.project.config.Constants;  
import com.heiyue.project.ui.PhotoViewActivity;  
import com.heiyue.project.ui.TakePhotoActivity;  
import com.heiyue.ui.ButtonTouch;  
import com.heiyue.ui.HorizontalListView;  
import com.heiyue.util.LogOut;  
import com.qixiaofu.qxf.R;  
import java.io.Serializable;  
import java.util.ArrayList;  
import java.util.Iterator;  
import java.util.List;  
import butterknife.BindView;  
public class OrderPush2Activity {  
 public static Activity instance = null;  
 public static void startActivity(Context context, OrderSubmissionData submissionData, String topprice, String price) {  
 Intent intent = new Intent(context, OrderPush2Activity.class);  
 intent.putExtra("data", submissionData);  
 intent.putExtra("topprice", topprice);  
 intent.putExtra("price", price);  
 context.startActivity(intent);  
 }  
 public static void enterActivityForResult(Activity activity, String id,  
 String number, String service\_stime, String service\_etime,  
 String bill\_desc, List<String> image, int requestCode) {  
 Intent intent = new Intent(activity, OrderPush2Activity.class);  
 intent.putExtra("id", id);  
 intent.putExtra("number", number);  
 intent.putExtra("service\_stime", service\_stime);  
 intent.putExtra("service\_etime", service\_etime);  
 intent.putExtra("bill\_desc", bill\_desc);  
 if (image != null) {  
 intent.putExtra("image", (Serializable) image);  
 }  
 activity.startActivityForResult(intent, requestCode);  
 }  
 @Override  
 protected int setContentLayoutView() {  
 return R.layout.order\_activity\_push\_2;  
 }  
 @BindView(R.id.et\_remark)  
 public EditText mEtRemark;  
 // private GridView mGridAddImage;  
 @BindView(R.id.lv\_add\_image)  
 public HorizontalListView mLvAddImage;  
 @BindView(R.id.cb\_show\_top)  
 public CheckBox mCbShowTop;  
 @BindView(R.id.et\_count)  
 public EditText mEtCount;  
 @BindView(R.id.btn\_is\_commic)  
 public ButtonTouch mBtnIsCommic;  
 @BindView(R.id.rl\_count)  
 public RelativeLayout rlCount;  
 @BindView(R.id.rl\_is\_hint)  
 public RelativeLayout rlIsHint;  
 @BindView(R.id.layTopCheck)  
 public View layTopCheck;  
 @BindView(R.id.topprice)  
 public TextView topprice;  
 @BindView(R.id.ll\_parent)  
 public LinearLayout llParent;  
 @Override  
 protected void initViews() {  
 instance = this;  
 }  
 //关于重新发布  
 private String id;  
 private String number;  
 private String service\_stime;  
 private String service\_etime;

private String bill\_desc;// 备注  
 private List<String> imageList;  
 private List<String> uploadList = new ArrayList<String>();  
 private int currentPosition;  
 private AccountBalanceBean paySettingStatus;  
 private OrderSubmissionData submissionDatas;  
 private String Topprice, price;  
 private String upLoadedImages;  
 private List<Integer> deleteImageIndexList;  
 private PhotoLvAdapter mPhotoAdapter;  
 private final int REQUESTCODE\_TO\_PAY = 10;// 去支付  
 private final int REQUESTCODE\_PHOTOLIST = 100;  
 private final int REQUESTCODE\_PHOTO\_VIEW = 101;  
 @SuppressLint("NewApi")  
 @Override  
 protected void initData() {  
 submissionDatas = (OrderSubmissionData) getIntent().getSerializableExtra("data");  
 Topprice = getIntent().getStringExtra("topprice");  
 price = getIntent().getStringExtra("price");  
 id = getIntent().getStringExtra("id");  
 number = getIntent().getStringExtra("number");  
 service\_stime = getIntent().getStringExtra("service\_stime");  
 service\_etime = getIntent().getStringExtra("service\_etime");  
 bill\_desc = getIntent().getStringExtra("bill\_desc");  
 imageList = (List<String>) getIntent().getSerializableExtra("image");  
 topprice.setText("价格：¥" + Topprice + "元/天");  
 LayoutParams params = new LayoutParams(LayoutParams.MATCH\_PARENT, 0);  
 llParent.setLayoutParams(params);  
 deleteImageIndexList = new ArrayList<Integer>();  
 initAddPhoto();  
 if (!TextUtils.isEmpty(id) && imageList != null && imageList.size() > 0) {  
 List<PhotoBean> list = new ArrayList<PhotoBean>();  
 for (String temp : imageList) {  
 PhotoBean photo = new PhotoBean(temp, 0);  
 list.add(photo);  
 }  
 mPhotoAdapter.setData(list);  
 }

mEtRemark.setText(bill\_desc);  
 mEtRemark.setSelection(TextUtils.isEmpty(bill\_desc) ? 0 : bill\_desc.length());  
 // 显示天数  
 if (!TextUtils.isEmpty(id)) {  
 layTopCheck.setVisibility(View.GONE);  
 llParent.setVisibility(View.VISIBLE);  
 mEtCount.setText(number);  
 mEtCount.setEnabled(false);  
 rlIsHint.setVisibility(View.GONE);  
 mBtnIsCommic.setText("重新发单");  
 mPhotoAdapter.setShowMinus(false);  
 }  
  
 if (adapter.getCount() < Constants.MYORDER\_IMAGE\_MAX\_COUNT) {  
 adapter.add(itemAdd);  
 }  
 mCbShowTop.setOnCheckedChangeListener(new OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged(CompoundButton buttonView,  
 boolean isChecked) {  
 if (!isChecked) {  
 LayoutParams params = new LayoutParams(  
 LinearLayout.LayoutParams.FILL\_PARENT, 0);  
 llParent.setLayoutParams(params);  
 } else {  
 LayoutParams params = new LayoutParams(  
 LinearLayout.LayoutParams.FILL\_PARENT,  
 LinearLayout.LayoutParams.WRAP\_CONTENT);  
 llParent.setLayoutParams(params);  
 }  
 }  
 });  
 mBtnIsCommic.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (TextUtils.isEmpty(id)) {  
 faDanCommitData();  
 } else {  
 againCommit();  
 }  
 }  
 });  
 }  
 private void initAddPhoto() {  
 mPhotoAdapter = new PhotoLvAdapter(context);  
 // 设置图片宽高,dp的大小  
 float dp\_5 = getResources().getDimension(R.dimen.db\_5);  
 mPhotoAdapter.setPhotoSize(4, 0);  
 mLvAddImage.setLayoutParams(new LayoutParams(android.widget.AbsListView.LayoutParams.MATCH\_PARENT, mPhotoAdapter.getImgHeight()));  
 mPhotoAdapter.setOnMinusClickListener(new PhotoLvAdapter.OnPhotoMunisClickListener() {  
 @Override  
 public void onItemClick(IBanner item, int position) {  
 mPhotoAdapter.remove(position);  
 int count = mPhotoAdapter.getCount();  
 if (count > 0) {  
 IBanner temp = mPhotoAdapter.getItem(count - 1);  
 if (!TextUtils.isEmpty(temp.getUrl())) {  
 mPhotoAdapter.add(new PhotoBean(null, R.drawable.btn\_camera));  
 }  
 }  
 }  
 });  
 if (mPhotoAdapter.getCount() < Constants.MYORDER\_IMAGE\_MAX\_COUNT) {  
 mPhotoAdapter.add(new PhotoBean(null, R.drawable.btn\_camera));  
 }  
 mPhotoAdapter.setOnPhotoClickListener(new PhotoLvAdapter.OnPhotoClickListener() {  
 @Override  
 public void onItemClick(IBanner item, final int position) {  
 if (!TextUtils.isEmpty(item.getUrl())) {  
 List<IBanner> imgs = new ArrayList<IBanner>(mPhotoAdapter.getList());  
 if (imgs != null && imgs.size() > 0) {  
 IBanner banner = imgs.get(imgs.size() - 1);  
 if (TextUtils.isEmpty(banner.getUrl())) {imgs.remove(imgs.size() - 1);  
 }  
 PhotoViewActivity.startActivity(context, position, imgs, REQUESTCODE\_PHOTO\_VIEW);  
 }  
 } else {  
 Intent intent = new Intent(context, TakePhotoActivity.class);  
 intent.putExtra(TakePhotoActivity.IS\_CHOOSE\_MANY, true);  
 intent.putExtra(TakePhotoActivity.CHOOSE\_MANY\_CURRENT\_SIZE, mPhotoAdapter.getCount() - 1);  
 intent.putExtra(TakePhotoActivity.key\_manyPhotoMaxCount, Constants.MYORDER\_IMAGE\_MAX\_COUNT);  
 OrderPush2Activity.this.startActivityForResult(intent, REQUESTCODE\_PHOTOLIST);  
 }  
 }  
 @Override  
 public void onItemLongClick(IBanner item, int position) {  
 }  
 });  
 mLvAddImage.setAdapter(mPhotoAdapter);  
 }  
 private void faDanCommitData() {  
 if (mCbShowTop.isChecked() && TextUtils.isEmpty(mEtCount.getText().toString())) {  
 showToast("请输入置顶天数");  
 return;  
 }  
 submissionDatas.setBill\_desc(mEtRemark.getText().toString().trim());  
 submissionDatas.setTop\_day(mEtCount.getText().toString().trim());  
 if (mCbShowTop.isChecked()) {  
 float tp;  
 tp = Integer.valueOf(submissionDatas.getTop\_day()) \* Float.valueOf(Topprice);  
 // intent.putExtra("topprice", tp + "");  
 OrderPushPayActivity.enterActivityForResult((Activity) context, submissionDatas, price, tp + "", REQUESTCODE\_TO\_PAY);  
 } else {  
 OrderPushPayActivity.enterActivityForResult((Activity) context, submissionDatas, price, "0", REQUESTCODE\_TO\_PAY);  
 }  
 }  
 private void getPaySettingStatus() {  
 dao.getPayPwdStatus(dao.getAccountid(), "",  
 new RequestCallBack<AccountBalanceBean>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<AccountBalanceBean> result) {  
 if (result.netMsg.success) {  
 paySettingStatus = result.content;  
 showToast(result == null ? "没有设置支付密码" : "已设置支付密码");  
 } else {  
 paySettingStatus = new AccountBalanceBean();  
 }  
 String statu = paySettingStatus.getStatu();  
 if ("0".equals(statu)) {  
 // 跳转到设置密码界面  
 } else if ("1".equals(statu)) {  
 Intent intent = new Intent(OrderPush2Activity.this, OrderPushPayActivity.class);  
 intent.putExtra("data", submissionDatas);  
 intent.putExtra("price", price);  
 if (mCbShowTop.isChecked()) {  
 float tp = Integer.valueOf(submissionDatas.getTop\_day()) \* Float.valueOf(Topprice);  
 intent.putExtra("topprice", tp + "");  
 } else {  
 intent.putExtra("topprice", "0");  
 }  
 startActivity(intent);  
 }  
 }  
 });  
 }  
 private void againCommit() {  
 if (TextUtils.isEmpty(id)) {  
 return;  
 }  
 bill\_desc = mEtRemark.getText().toString();  
 dao.reSetBill(id, number, bill\_desc, upLoadedImages, service\_stime, service\_etime, new RequestCallBack<String>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 if (result.netMsg.success) {  
 isHaveChange = true;  
 setResultCustom();  
 }  
 }  
 });  
 }  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (resultCode == RESULT\_OK) {  
 switch (requestCode) {  
 case REQUESTCODE\_PHOTOLIST:  
 if (data != null) {  
 uploadList.clear();  
 currentPosition = 0;  
 String dataPath = TakePhotoActivity.getDataPath(data);  
 if (TextUtils.isEmpty(id)) {  
 }  
 if (dataPath != null) {  
 ArrayList<String> dataPathList = new ArrayList<String>();  
 dataPathList.add(dataPath);  
 uploadImgMulti(dataPathList);  
 int count = mPhotoAdapter.getCount();  
 mPhotoAdapter.remove(count - 1);  
 mPhotoAdapter.add(new PhotoBean(dataPath, 0));  
 if (count < Constants.IMG\_COUNT\_MAX) {  
 mPhotoAdapter.add(new PhotoBean(null, R.drawable.btn\_camera));  
 }  
 }  
 ArrayList<String> dataPathArr = TakePhotoActivity.getDataPathArr(data);  
 if (dataPathArr != null) {  
 uploadImgMulti(dataPathArr);  
 for (int i = 0; i < dataPathArr.size(); i++) {  
 int count = mPhotoAdapter.getCount();  
 mPhotoAdapter.remove(count - 1);  
 mPhotoAdapter.add(new PhotoBean(dataPathArr.get(i), 0));  
 if (count < Constants.MYORDER\_IMAGE\_MAX\_COUNT) {  
 mPhotoAdapter.add(new PhotoBean(null, R.drawable.btn\_camera));  
 }  
 }  
 }  
 }  
 break;  
 case REQUESTCODE\_PHOTO\_VIEW:  
 List<String> imageDeleteList = (List<String>) data.getSerializableExtra("imageList");  
 List<? extends IBanner> imageList = mPhotoAdapter.getList();  
 for (String img: imageDeleteList) {  
 Iterator<? extends IBanner> imageIterator = imageList.iterator();  
 while(imageIterator.hasNext()){  
 IBanner image = imageIterator.next();  
 if(img.equals(image.getUrl())){  
 deleteImageIndexList.add(imageList.indexOf(image));  
 imageIterator.remove();  
 }  
 }  
 }  
 mPhotoAdapter.notifyDataSetChanged();  
 String[] upLoadedImageArray = upLoadedImages.split(",");  
 upLoadedImages = "";  
 for(int i = 0; i < upLoadedImageArray.length; i++){  
 for(int j = 0; j < deleteImageIndexList.size(); j++){  
 if(i == deleteImageIndexList.get(j)){  
 upLoadedImageArray[i] = "";  
 }  
 }  
 }  
 upLoadedImages = "";  
 for(int i = 0; i < upLoadedImageArray.length; i++){  
 if(!TextUtils.isEmpty(upLoadedImageArray[i])){  
 if (TextUtils.isEmpty(upLoadedImages)) {  
 upLoadedImages = upLoadedImageArray[i];  
 } else {  
 upLoadedImages += "," + upLoadedImageArray[i];  
 }  
 }  
 }  
 submissionDatas.setImages(upLoadedImages);  
 break;  
 default:  
 break;  
 }  
 }  
 }  
 @Override  
 protected void setActionBarDetail() {  
 }  
 private void uploadImgMulti(List<String> pathList) {  
 dao.uploadImageMulti(pathList, new RequestCallBack<ResultImageInfo>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<ResultImageInfo> result) {  
 if (result.netMsg.success) {  
 if (upLoadedImages == null) {  
 upLoadedImages = result.content.img;  
 } else  
 upLoadedImages += "," + result.content.img;  
 }  
 }  
 });  
 }  
}

package com.heiyue.project.ui.index;  
import android.app.Activity;  
import android.content.Context;  
import android.content.Intent;  
import android.text.TextUtils;  
import android.util.DisplayMetrics;  
import android.view.View;  
import android.view.View.OnClickListener;  
import android.widget.EditText;  
import android.widget.GridView;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
import com.heiyue.net.NetResponse;  
import com.heiyue.net.RequestCallBack;  
import com.heiyue.photoview.RoundedImageView;  
import com.heiyue.project.adapter.AddImageAdapter;  
import com.heiyue.project.base.BaseActivity;  
import com.heiyue.project.bean.BillingDetailsBean;  
import com.heiyue.project.bean.PubImage;  
import com.heiyue.project.bean.ResultImageInfo;  
import com.heiyue.project.bean.SQLinformation;  
import com.heiyue.project.bean.SmallStockBean;  
import com.heiyue.project.config.Constants;  
import com.heiyue.project.ui.CommentActivity;  
import com.heiyue.project.ui.OrderCancleReasonActivity;  
import com.heiyue.project.ui.TakePhotoActivity;  
import com.heiyue.project.ui.WebViewActivity;  
import com.heiyue.project.ui.mine.CheckCompliteActivity;  
import com.heiyue.project.ui.mine.MyOrderPayActivity;  
import com.heiyue.project.ui.shop.SupplementShopOrderSeclecPayTypeActivity;  
import com.heiyue.project.util.ClipBoardUtil;  
import com.heiyue.project.util.EditTextInputStyleUtil;  
import com.heiyue.project.util.PubDialogUtil;  
import com.heiyue.project.util.TimeUtil;  
import com.heiyue.ui.ButtonTouch;  
import com.heiyue.ui.NoScrollGridView;  
import com.heiyue.util.LogOut;  
import com.heiyue.util.ToastUtil;  
import com.hyphenate.chat.EMClient;  
import com.nostra13.universalimageloader.core.ImageLoader;  
import com.qixiaofu.qxf.R;  
import java.util.ArrayList;  
import java.util.List;  
import butterknife.BindView;  
public class OrderReceiveDetailsActivity {  
 private static final String REQUESTCODE\_TO\_ORDER\_RECEIVE\_CANCLE = "10";

private static final String REQUESTCODE\_TO\_ORDER\_SEND\_CANCLE = "20";  
 private static final String REQUESTCODE\_TO\_ORDER\_SEND\_UNFINISHED = "30";  
 private static final int REQUESTCODE\_TO\_ORDER\_CHANGE\_RECEIVER = 40;  
 private static final int REQUESTCODE\_REPORT\_MODIFY = 11;  
 private static final int REQUESTCODE\_REPORT\_ADD = 12;  
 public static void startActivity(Context context, String id, String title) {  
 Intent intent = new Intent(context, OrderReceiveDetailsActivity.class);  
 intent.putExtra("id", id);  
 intent.putExtra("title", title);  
 intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);  
 intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);  
 context.startActivity(intent);  
 }  
 public static void startActivity(Context context, String id, String title, String pay\_statu) {  
 Intent intent = new Intent(context, OrderReceiveDetailsActivity.class);  
 intent.putExtra("id", id);  
 intent.putExtra("title", title);  
 intent.putExtra("pay\_statu", pay\_statu);  
 context.startActivity(intent);  
 }  
 public static void startActivity(Activity activity, String id, String title, String pay\_statu, String is\_eval, String move\_state, String enroll\_state, int requestCode) {  
 Intent intent = new Intent(activity, OrderReceiveDetailsActivity.class);  
 intent.putExtra("id", id);  
 intent.putExtra("title", title);  
 intent.putExtra("pay\_statu", pay\_statu);  
 intent.putExtra("is\_eval", is\_eval);  
 intent.putExtra("move\_state", move\_state);  
 intent.putExtra("enroll\_state", enroll\_state);  
 activity.startActivityForResult(intent, requestCode);  
 }  
 private String Title;  
 private PubDialogUtil pubDialogUtil;  
 private String is\_user\_eval;  
 private AddImageAdapter adapter;  
 private String id;  
 private String pay\_statu;  
 private String enroll\_state;  
 private String move\_state;  
 @BindView(R.id.layout\_content)  
 public LinearLayout mLayoutContent;  
 @BindView(R.id.tv\_number)  
 public TextView tv\_number;  
 @BindView(R.id.tv\_Establishtime)  
 public TextView tv\_Establishtime;  
 @BindView(R.id.imgHeader)  
 public RoundedImageView imgHeader;  
 @BindView(R.id.tvUserName)  
 public TextView tvUserName;  
 @BindView(R.id.tvState)  
 public TextView tvState;  
 @BindView(R.id.tvDate)  
 public TextView tvDate;  
  
 @BindView(R.id.tvProjectName)  
 public TextView tvProjectName;  
 @BindView(R.id.tvTerritory)  
 public TextView tvTerritory;  
 @BindView(R.id.tvBrand)  
 public TextView tvBrand;  
 @BindView(R.id.tvCountUnit)  
 public TextView tvCountUnit;  
 @BindView(R.id.tvForm)  
 public TextView tvForm;  
 @BindView(R.id.tvType)  
 public TextView tvType;  
 @BindView(R.id.tvTime)  
 public TextView tvTime;

@BindView(R.id.tvArea)  
 public TextView tvArea;  
 @BindView(R.id.tvTerritoryOther)  
 public TextView tvTerritoryOther;  
 @BindView(R.id.tvDesc)  
 public TextView tvDesc;  
 @BindView(R.id.tvPrice)  
 public TextView tvPrice;  
 @BindView(R.id.line\_top)  
 public View viewLineTop;  
 @BindView(R.id.line\_bottom)  
 public View viewLineBottom;  
 private LinearLayout llSparePartsName;  
 private LinearLayout llSparePartsSn;  
 private TextView tvSparePartsName;  
 private TextView tvSparePartsSn;  
 @BindView(R.id.mygridview1)  
 public GridView mygridview1;  
 private AddImageAdapter adapter1;  
 private TextView tv\_cancel;  
 @BindView(R.id.kehutiaojia)  
 public LinearLayout kehutiaojia;  
 @BindView(R.id.grid\_add\_image)  
 public NoScrollGridView grid\_add\_image;  
 public AddImageAdapter adapter2;  
 @BindView(R.id.tv\_tiaojiaPrice)  
 public TextView tv\_tiaojiaPrice;  
 public PubImage itemAdd;  
 @BindView(R.id.tv\_refuse)  
 public TextView tv\_refuse;  
 @BindView(R.id.tv\_Agree)  
 public TextView tv\_Agree;  
 @BindView(R.id.tiaojia)  
 public LinearLayout tiaojia;  
 @BindView(R.id.et\_tiaojiaPrice)  
 public EditText et\_tiaojiaPrice;  
 @BindView(R.id.grid\_add\_image1)  
 public NoScrollGridView grid\_add\_image1;  
 @BindView(R.id.tv\_true)  
 public TextView tv\_true;  
 @BindView(R.id.ll\_order\_cancle\_reason)  
 public LinearLayout mLLayoutOrderCancleReason;  
 @BindView(R.id.tvCancleReason)  
 public TextView mOrderCancleReason;  
 @BindView(R.id.btnCancel)  
 public ButtonTouch mBtnCancel;

@BindView(R.id.btnGoPay)  
 public ButtonTouch mBtnGoPay;  
 @BindView(R.id.btnWithdraw)  
 public ButtonTouch mBtnWithdraw;  
 @BindView(R.id.btnAgainNew)  
 public ButtonTouch mBtnAgainNew;  
 @BindView(R.id.btnOk)  
 public ButtonTouch mBtnOk;  
 @BindView(R.id.btnEvaluate)  
 public ButtonTouch mBtnEvaluate;  
 @BindView(R.id.btnEvaluateLook)  
 public ButtonTouch mBtnEvaluateLook;  
 @BindView(R.id.btnDel)  
 public ButtonTouch mBtnDel;  
 @BindView(R.id.btnModifyPrice)  
 public ButtonTouch mBtnModifyPrice;  
 @BindView(R.id.btnNotCompleted)  
 public ButtonTouch mBtnNotCompleted;  
 @BindView(R.id.btnWaitOk)  
 public ButtonTouch mBtnWaitOk;  
 // private com.heiyue.ui.ButtonTouch mBtnAgree;  
 // private com.heiyue.ui.ButtonTouch mBtnDisagree;  
 @BindView(R.id.btnStartWork)  
 public ButtonTouch mBtnStartWork;

@BindView(R.id.btnChangeReicever)  
 public ButtonTouch mBtnChangeReicever;  
 @BindView(R.id.btnServiceReportModify)  
 public ButtonTouch mBtnServiceReportModify;  
 @BindView(R.id.btnServiceReport)  
 public ButtonTouch mBtnServiceReport;  
 @BindView(R.id.btnSmallStockLook)  
 public ButtonTouch mBtnSmallStockLook;  
 @BindView(R.id.btnAgreeReceive)  
 public ButtonTouch mBtnAgreeReceive;  
 @BindView(R.id.btnRefuseReceive)  
 public ButtonTouch mBtnRefuseReceive;  
 public BillingDetailsBean showInfor;  
 @Override  
 protected int setContentLayoutView() {  
 return R.layout.activity\_order\_details;  
 }  
 @Override  
 protected void initViews() {  
 mLayoutContent.setVisibility(View.INVISIBLE);  
 btnRight.setVisibility(View.INVISIBLE);  
 btnRight.setImageResource(R.drawable.icon\_chat\_red);  
 linformation = new SQLinformation();  
 tv\_number.setOnLongClickListener(new View.OnLongClickListener() {  
 @Override  
 public boolean onLongClick(View view) {  
 ClipBoardUtil.copy(OrderReceiveDetailsActivity.this, "" + tv\_number.getText().toString());  
 return false;  
 }  
 });  
 mBtnAgainNew.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 OrderPush1Activity.enterActivityForResult((Activity) context,  
 id, REQUESTCODE\_AGAIN\_NEW);  
 }  
 });  
 mBtnCancel.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if ("0".equals(pay\_statu)) {  
 showDialogNew(showInfor, 3, "你确定要取消订单吗？"); OrderCancleReasonActivity.startActivity(OrderReceiveDetailsActivity.this,REQUESTCODE\_TO\_ORDER\_RECEIVE\_CANCLE, showInfor.id, showInfor.call\_name, 0);  
 } else {  
 showDialogNew(showInfor, 3, "取消订单将扣除服务费用的10%，你确定要取消订单吗？");  
 if (Title.equals("发单")) {  
 OrderCancleReasonActivity.startActivity(OrderReceiveDetailsActivity.this, REQUESTCODE\_TO\_ORDER\_SEND\_CANCLE, showInfor.id, showInfor.call\_name, 0);  
 } else {  
 OrderCancleReasonActivity.startActivity(OrderReceiveDetailsActivity.this, REQUESTCODE\_TO\_ORDER\_RECEIVE\_CANCLE, showInfor.id, showInfor.call\_name, 0);  
 }  
 }  
 }  
 });  
 mBtnGoPay.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 MyOrderPayActivity.enterActivityForResult((Activity) context,  
 showInfor == null ? null : showInfor.id,  
 showInfor == null ? null : showInfor.service\_price,  
 REQUESTCODE\_GO\_TO\_PAY);  
 }  
 });  
 mBtnWithdraw.setOnClickListener(new OnClickListener() {  
  
 @Override  
 public void onClick(View v) {  
 showDialogNew(showInfor, 1, "撤销之后订单无法被找回，你确定要撤销此订单吗？");  
 }  
 });  
 mBtnOk.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 okMyOrders(showInfor);  
 }  
 });  
 mBtnEvaluate.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (Title.equals("接单")) {  
 CommentActivity.startActivity((Activity) context, "2", showInfor == null ? null : showInfor.id, REQUESTCODE\_GO\_TO\_COMMENT);  
 } else if (Title.equals("发单")) {  
 CommentActivity.startActivity((Activity) context, "1", showInfor == null ? null : showInfor.id, REQUESTCODE\_GO\_TO\_COMMENT);  
 }  
 }  
 });  
 mBtnEvaluateLook.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 OrderDetailCommentActivity.startActivity(context, showInfor.id);  
 }  
 });  
 mBtnDel.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 showDialogNew(showInfor, 2, "删除订单后无法找回，你确定删除此订单吗？");  
 }  
 });  
 mBtnModifyPrice.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 // ModifyPrice();  
 if (tiaojia.getVisibility() == View.VISIBLE) {  
 tiaojia.setVisibility(View.GONE);  
 et\_tiaojiaPrice.setText("");  
 imgs = "";  
 adapter.setData(null);  
 adapter.add(itemAdd);  
 } else {  
 tiaojia.setVisibility(View.VISIBLE);  
 }  
 }  
 });  
 mBtnNotCompleted.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
// notCompleted(showInfor);  
 OrderCancleReasonActivity.startActivity(OrderReceiveDetailsActivity.this, REQUESTCODE\_TO\_ORDER\_SEND\_UNFINISHED, showInfor.id, 0);  
 }  
 });  
 tv\_refuse.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 // 拒绝调价  
 AgreePrice(showInfor, "0");  
 }  
 });  
 tv\_Agree.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 // 同意调价  
 AgreePrice(showInfor, "1");  
 }  
 });  
 tv\_true.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 ModifyPrice();  
 }  
 });  
 mBtnStartWork.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startWork(showInfor);  
 }  
 });  
 mBtnChangeReicever.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 OrderChangeActivity.startActivity(OrderReceiveDetailsActivity.this, showInfor.id, REQUESTCODE\_TO\_ORDER\_CHANGE\_RECEIVER);  
 }  
 });  
 mBtnAgreeReceive.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 agreeChangeOrderToMe(showInfor);  
 }  
 });  
 mBtnRefuseReceive.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 refuseChangeOrderToMe(showInfor);  
 }  
 });  
 mBtnServiceReportModify.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 OrderServiceReportActivity.startActivity(OrderReceiveDetailsActivity.this, showInfor, REQUESTCODE\_REPORT\_MODIFY);  
 }  
 });  
 mBtnServiceReport.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 WebViewActivity.startActivity(context, "服务报告", showInfor.report\_html, false, true);  
 }  
 });  
 }  
 private SQLinformation linformation;  
 @Override  
 protected void initData() {  
 Title = getIntent().getStringExtra("title");  
 id = getIntent().getStringExtra("id");  
 is\_user\_eval = getIntent().getStringExtra("is\_eval");  
 pay\_statu = getIntent().getStringExtra("pay\_statu");  
 move\_state = getIntent().getStringExtra("move\_state");  
 enroll\_state = getIntent().getStringExtra("enroll\_state");  
 adapter = new AddImageAdapter(context);  
 adapter1 = new AddImageAdapter(context);  
 adapter2 = new AddImageAdapter(context);  
 grid\_add\_image.setAdapter(adapter2);  
 mygridview1.setAdapter(adapter1);  
 grid\_add\_image1.setAdapter(adapter);  
 itemAdd = new PubImage();  
 itemAdd.isAdd = true;  
 adapter.add(itemAdd);  
 getNetData();  
 EditTextInputStyleUtil.setEditTextInputPriceStyle(et\_tiaojiaPrice);  
 btnRight.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 ChatActivity.startActivity(context, linformation, true);  
 }  
 });  
 }  
 public void getNetData() {  
 if (Title.equals("发单")) {  
 dao.myBillDetail(new RequestCallBack<BillingDetailsBean>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<BillingDetailsBean> result) {  
 if (result.netMsg.success) {  
 showInfor = result.content;  
 linformation.call\_name = result.content.getCall\_name();  
 linformation.call\_nik\_name = result.content.getCall\_nik\_name();  
 linformation.os = result.content.getOs();  
 linformation.bill\_user\_avatar = result.content.getBill\_user\_avatar();  
 linformation.ot\_user\_avatar = result.content.getOt\_user\_avatar();  
 showdata(result.content);  
 }  
 }  
 }, id);  
 } else if (Title.equals("接单")) {  
 dao.myOtDetail(id, move\_state, new RequestCallBack<BillingDetailsBean>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<BillingDetailsBean> result) {  
 if (result.netMsg.success) {  
 showInfor = result.content;  
 linformation.call\_name = result.content.getCall\_name();  
 linformation.call\_nik\_name = result.content.getCall\_nik\_name();  
 linformation.os = result.content.getOs();  
 // linformation.bill\_user\_avatar = result.content  
 // .getBill\_user\_avatar();  
 // linformation.ot\_user\_avatar = result.content  
 // .getOt\_user\_avatar();  
 linformation.bill\_user\_avatar = result.content.getOt\_user\_avatar();  
 linformation.ot\_user\_avatar = result.content.getBill\_user\_avatar();  
 showdata(result.content);  
 }  
 }  
 });  
 }  
 }  
 private String yuanjia;  
 public void showdata(BillingDetailsBean data) {  
 mLayoutContent.setVisibility(View.VISIBLE);  
 yuanjia = data.getService\_price();  
 tv\_number.setText(data.getBill\_sn());  
 tv\_Establishtime.setText("创建时间 " + TimeUtil.getSimpleDate(data.getInputtime(), "yyyy-MM-dd HH:mm"));  
 ImageLoader.getInstance().displayImage(data.getBill\_user\_avatar(), imgHeader);  
 tvUserName.setText(data.getBill\_user\_name());  
 mBtnAgainNew.setVisibility(View.GONE);  
 mBtnCancel.setVisibility(View.GONE);  
 mBtnDel.setVisibility(View.GONE);  
 mBtnEvaluate.setVisibility(View.GONE);  
 mBtnEvaluateLook.setVisibility(View.GONE);  
 mBtnGoPay.setVisibility(View.GONE);  
 mBtnModifyPrice.setVisibility(View.GONE);  
 mBtnNotCompleted.setVisibility(View.GONE);  
 mBtnOk.setVisibility(View.GONE);  
 mBtnWithdraw.setVisibility(View.GONE);  
 mBtnServiceReportModify.setVisibility(View.GONE);  
 mBtnServiceReport.setVisibility(View.GONE);  
 mBtnWaitOk.setVisibility(View.GONE);  
 mBtnStartWork.setVisibility(View.GONE);  
 mBtnChangeReicever.setVisibility(View.GONE);  
 tiaojia.setVisibility(View.GONE);  
 btnRight.setVisibility(View.VISIBLE);  
 if (data.getBill\_statu().equals("0")) {  
 tvState.setText("已撤销");  
 mBtnDel.setVisibility(View.VISIBLE);  
 } else if (data.getBill\_statu().equals("1")) {  
 tvState.setText("待接单");  
 } else if (data.getBill\_statu().equals("2")) {  
 tvState.setText("已接单");  
 if ("0".equals(data.t\_state + "")) {  
 mBtnCancel.setVisibility(View.VISIBLE);  
 mBtnOk.setVisibility(View.VISIBLE);  
 } else if ("4".equals(data.t\_state + "")) {  
 mBtnOk.setVisibility(View.VISIBLE);  
 mBtnCancel.setVisibility(View.VISIBLE);  
 } else {  
 if ("1".equals(data.is\_special)) {  
 mBtnServiceReportModify.setVisibility(View.GONE);  
 } else {  
 mBtnServiceReportModify.setVisibility(View.VISIBLE);  
 }  
 mBtnWaitOk.setVisibility(View.VISIBLE);  
 }  
 } else if (data.getBill\_statu().equals("3")) {  
 tvState.setText("已完成");  
 if ("1".equals(data.is\_have\_report)) {  
 mBtnServiceReport.setVisibility(View.VISIBLE);  
 } else if ("0".equals(data.is\_have\_report)) {  
 mBtnServiceReport.setVisibility(View.GONE);  
 }  
 if (TextUtils.isEmpty(is\_user\_eval)) {  
 data.setIsUserEval(0);  
 }  
 if (0 == data.getIsUserEval()) {  
 mBtnEvaluate.setVisibility(View.VISIBLE);  
 mBtnEvaluateLook.setVisibility(View.GONE);  
 } else if (1 == data.getIsUserEval()) {  
 mBtnEvaluateLook.setVisibility(View.VISIBLE);  
 mBtnEvaluate.setVisibility(View.GONE);  
 }  
 mBtnDel.setVisibility(View.VISIBLE);  
  
 final List<SmallStockBean> smallStockList = data.goods;  
 if (smallStockList != null && smallStockList.size() > 0 && "1".equals(data.is\_special)) {  
 mBtnSmallStockLook.setVisibility(View.VISIBLE);  
 mBtnSmallStockLook.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 OrderDetailUseSmallStockListActivity.startActivity(OrderReceiveDetailsActivity.this, smallStockList, 1);  
 }  
 });  
 }  
 btnRight.setVisibility(View.GONE);  
 EMClient.getInstance().chatManager().deleteConversation(data.call\_name, true);  
 } else if (data.getBill\_statu().equals("4")) {  
 tvState.setText("已失效");  
 } else if (data.getBill\_statu().equals("5")) {  
 tvState.setText("已取消");  
 mLLayoutOrderCancleReason.setVisibility(View.VISIBLE);  
 mOrderCancleReason.setText(data.cancle\_bill);  
 mBtnDel.setVisibility(View.VISIBLE);  
 btnRight.setVisibility(View.GONE);  
 } else if (data.getBill\_statu().equals("6")) {  
 tvState.setText("调价中");  
 if (Title.equals("接单")) {  
 kehutiaojia.setVisibility(View.VISIBLE);  
 List<PubImage> imgs = new ArrayList<PubImage>();  
 for (int i = 0; i < data.getUp\_images().size(); i++) {  
 PubImage image = new PubImage();  
 image.cover = data.getUp\_images().get(i);  
 imgs.add(image);  
 }  
 tv\_tiaojiaPrice.setText("¥" + data.getService\_up\_price());  
 adapter2.setData(imgs);  
 }  
 } else if (data.getBill\_statu().equals("7")) {  
 if ("0".equals(pay\_statu)) {  
 tvState.setText("未支付 ");  
 mBtnCancel.setVisibility(View.VISIBLE);  
 mBtnGoPay.setVisibility(View.VISIBLE);  
 } else if ("1".equals(pay\_statu)) {  
 tvState.setText("已支付 ");  
 mBtnOk.setVisibility(View.VISIBLE);  
 }  
 } else if ("8".equals(data.getBill\_statu())) {  
 tvState.setText("已接单");  
  
 if ("1".equals(data.move\_state)) {  
 if ("1".equals(data.bill\_belong)) {  
 mBtnRefuseReceive.setVisibility(View.VISIBLE);  
 mBtnAgreeReceive.setVisibility(View.VISIBLE);  
 btnRight.setVisibility(View.GONE);  
 } else if ("2".equals(data.bill\_belong)) {  
 mBtnRefuseReceive.setVisibility(View.GONE);  
 mBtnAgreeReceive.setVisibility(View.GONE);  
 }  
 } else if ("2".equals(data.move\_state)) {  
 tvState.setText("来自订单转移");  
 if (data.t\_state == 0) {  
 mBtnCancel.setVisibility(View.VISIBLE);  
 mBtnOk.setVisibility(View.VISIBLE);  
 } else if (data.t\_state == 4) {  
 mBtnOk.setVisibility(View.VISIBLE);  
 mBtnCancel.setVisibility(View.VISIBLE);  
 } else {  
 mBtnWaitOk.setVisibility(View.VISIBLE);  
 }  
 btnRight.setVisibility(View.VISIBLE);  
 if (Integer.valueOf(data.move\_count) < 2) {  
 mBtnChangeReicever.setVisibility(View.VISIBLE);  
 } else {  
 mBtnChangeReicever.setVisibility(View.GONE);  
 }  
 mBtnCancel.setVisibility(View.VISIBLE);  
 mBtnStartWork.setVisibility(View.VISIBLE);  
 mBtnRefuseReceive.setVisibility(View.GONE);  
 mBtnAgreeReceive.setVisibility(View.GONE);  
 } else if ("0".equals(data.move\_state)) {  
 if (Integer.valueOf(data.move\_count) < 2) {  
 mBtnChangeReicever.setVisibility(View.VISIBLE);  
 } else {  
 mBtnChangeReicever.setVisibility(View.GONE);  
 }  
 mBtnCancel.setVisibility(View.VISIBLE);  
 mBtnStartWork.setVisibility(View.VISIBLE);  
 } else {  
 mBtnChangeReicever.setVisibility(View.VISIBLE);  
 mBtnCancel.setVisibility(View.VISIBLE);  
 mBtnStartWork.setVisibility(View.VISIBLE);  
 }  
 }  
 if ("1".equals(enroll\_state)) {  
 tvState.setText("已报名");  
 tvState.setVisibility(View.VISIBLE);  
 viewLineTop.setVisibility(View.GONE);  
 viewLineBottom.setVisibility(View.GONE);  
 }  
  
 if (btnRight.getVisibility() == View.GONE) {  
 EMClient.getInstance().chatManager().deleteConversation(data.call\_name, true);  
 }  
 tvDate.setText(TimeUtil.getSimpleDate(data.getInputtime(), "yyyy-MM-dd HH:mm"));  
 tvProjectName.setText(data.getProjectName());  
 tvTerritory.setText(data.getService\_sector());  
 tvBrand.setText(data.getService\_brand());  
 tvCountUnit.setText(data.getNumber());  
 tvForm.setText(data.getService\_form());  
 tvType.setText(data.getService\_type());  
 tvTime.setText(TimeUtil.getSimpleDate(data.getService\_stime(),  
 "yyyy年MM月dd日") + "-" + TimeUtil.getSimpleDate(data.getService\_etime(), "yyyy年MM月dd日"));  
 tvArea.setText(data.getService\_address());  
 tvTerritoryOther.setText(data.getOther\_service\_sector());  
 tvDesc.setText(data.getBill\_desc());  
 tvPrice.setText("¥" + data.getService\_price());  
 List<PubImage> imgs = new ArrayList<PubImage>();  
 for (int i = 0; i < data.getImage().size(); i++) {  
 PubImage image = new PubImage();  
 image.cover = data.getImage().get(i);  
 imgs.add(image);  
 }  
 adapter1.setData(imgs);

setGridView(mygridview1, 100, adapter1.getCount());  
 }  
 @Override  
 protected void setActionBarDetail() {  
 }  
 private String imgs;  
 private final int REQUESTCODE\_GO\_TO\_PAY = 11;  
 private final int REQUESTCODE\_GO\_TO\_COMMENT = 22;  
 private final int REQUESTCODE\_AGAIN\_NEW = 33;  
 private final int REQUESTCODE\_TIAO\_JIA = 10;  
 private final int REQUESTCODE\_OK\_COMPLITE = 1;  
 private void uploadImgMulti(List<String> pathList) {  
 dao.uploadImageMulti(pathList, new RequestCallBack<ResultImageInfo>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<ResultImageInfo> result) {  
 if (result.netMsg.success) {  
 if (imgs == null) {  
 imgs = result.content.img;  
 } else  
 imgs += "," + result.content.img;  
 }  
 }  
 });  
 }  
 private void cheXiaoFaDan(BillingDetailsBean item) {  
 if (item == null) {  
 showToast("数据异常");  
 return;  
 }  
 dao.myOrdersWithDraw(item.id, new RequestCallBack<String>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 if (result.netMsg.success) {  
 showToast("撤销成功");  
 isHaveChange = true;  
 setResultCustom();  
 getNetData();  
 }  
 }  
 });  
 }  
 private void delMyOrders(BillingDetailsBean item) {  
 if (item == null) {  
 showToast("数据异常");  
 return;  
 }  
 dao.engDelMyOrders(item.id, new RequestCallBack<String>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 if (result.netMsg.success) {  
 showToast("删除成功");  
 isHaveChange = true;  
 setResultCustom();  
 context.finish();  
 }  
 }  
 });  
 }  
 private void cancelMyOrders(final BillingDetailsBean item) {  
 if (item == null) {  
 showToast("数据异常");  
 return;  
 }  
 dao.cancelMyOrders(item.id, "", new RequestCallBack<String>() {  
 @Override  
 public void start() {  
  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 if (result.netMsg.success) {  
 EMClient.getInstance().chatManager().deleteConversation(item.call\_name, false);  
 isHaveChange = true;  
 setResultCustom();  
 getNetData();  
 }  
 }  
 });  
 }  
 private void EngineercancelMyOrders(final BillingDetailsBean item) {  
 if (item == null) {  
 showToast("数据异常");  
 return;  
 }  
 dao.cancelEngOffBill(item.id, "", new RequestCallBack<String>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 if (result.netMsg.success) {  
 EMClient.getInstance().chatManager().deleteConversation(item.call\_name, false);  
 showToast("取消成功");  
 isHaveChange = true;  
 setResultCustom();  
 context.runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 try {  
 Thread.sleep(500);  
 context.finish();  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 }  
 }  
 });  
 }

private void okMyOrders(BillingDetailsBean item) {  
 if (item == null) {  
 showToast("数据异常");  
 return;  
 }  
 if ("接单".equals(Title)) {  
 if ("7".equals(item.bill\_statu + "")) {  
 CheckCompliteActivity.startActivity((Activity) context, item, CheckCompliteActivity.TYPE\_BUDAN, REQUESTCODE\_OK\_COMPLITE);  
 } else {  
// CheckCompliteActivity.startActivity((Activity) context, item, CheckCompliteActivity.TYPE\_OK, REQUESTCODE\_OK\_COMPLITE);  
 if ("0".equals(item.is\_special)) {  
 OrderServiceReportActivity.startActivity(OrderReceiveDetailsActivity.this, item, REQUESTCODE\_REPORT\_ADD);  
 } else {  
 CheckCompliteActivity.startActivity(OrderReceiveDetailsActivity.this, item, CheckCompliteActivity.TYPE\_OK, REQUESTCODE\_OK\_COMPLITE);  
 }  
 }  
 return;  
 }  
 dao.okMyOrders(item.id, new RequestCallBack<String>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 if (result.netMsg.success) {  
 isHaveChange = true;  
 setResultCustom();  
 getNetData();  
 }  
 }  
 });  
 }  
 private void notCompleted(BillingDetailsBean item) {  
 if (item == null) {  
 showToast("数据异常");  
 return;  
 }  
 dao.notCompleted(item.id, "", new RequestCallBack<String>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 if (result.netMsg.success) {  
 isHaveChange = true;  
 setResultCustom();  
 getNetData();  
 }  
 }  
 });  
 }  
  
 private void startWork(final BillingDetailsBean item) {  
 if (item == null) {  
 showToast("数据异常");  
 return;  
 }  
 dao.engStartWorkMyOrders(item.id, new RequestCallBack<String>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 if (result.netMsg.success) {  
 item.bill\_statu = "2";  
 item.t\_state = 4;  
 isHaveChange = true;  
 setResultCustomData(item);  
 }  
 }  
 });  
 }

private void refuseChangeOrderToMe(BillingDetailsBean item) {  
 dao.refuseChangeOrder(item.id, item.move\_state, new RequestCallBack<String>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 if (result.netMsg.success) {  
 isHaveChange = true;  
 setResultCustom();  
 }  
 }  
 });  
 }  
 private void agreeChangeOrderToMe(BillingDetailsBean item) {  
 dao.agreeChangeOrder(item.id, item.move\_state, new RequestCallBack<String>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 if (result.netMsg.success) {  
 isHaveChange = true;  
 setResultCustom();  
 }  
 }  
 });  
 }  
 private void AgreePrice(BillingDetailsBean item, final String type) {  
 if (item == null) {  
 showToast("数据异常");  
 return;  
 }  
 dao.agreePrice(new RequestCallBack<String>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 if (result.netMsg.success) {  
 if ("1".equals(type)) {  
 showToast("同意调价成功");  
 } else if ("0".equals(type)) {  
 showToast("拒绝调价成功");  
 }  
 kehutiaojia.setVisibility(View.GONE);  
 isHaveChange = true;  
 setResultCustom();  
 getNetData();  
 }  
 }  
 }, type, item.id);  
 }  
 private void ModifyPrice() {  
 if (et\_tiaojiaPrice.getText().toString().trim().equals("")) {

return;  
 }  
 if (Float.valueOf(et\_tiaojiaPrice.getText().toString().trim()) < Float.valueOf(yuanjia)) {  
 dao.modifyPricedi(new RequestCallBack<String>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 if (result.netMsg.success) {  
 isHaveChange = true;  
 setResultCustom();  
 }  
 }  
 }, id, et\_tiaojiaPrice.getText().toString().trim(), imgs);  
 } else if (Float.valueOf(et\_tiaojiaPrice.getText().toString().trim()) > Float.valueOf(yuanjia)) {  
 float p = Float.valueOf(et\_tiaojiaPrice.getText().toString().trim()) - Float.valueOf(yuanjia);  
 SupplementShopOrderSeclecPayTypeActivity.startActivity(context,  
 et\_tiaojiaPrice.getText().toString().trim(), p + "", id,  
 imgs, REQUESTCODE\_TIAO\_JIA);  
 } else {  
 showToast("输入调价和原价一样");  
 }  
  
 }

public void showDialogNew(final BillingDetailsBean item, final int type, String message) {  
 if (item == null) {  
 showToast("数据异常");  
 return;  
 }  
 pubDialogUtil = new PubDialogUtil(OrderReceiveDetailsActivity.this);  
 pubDialogUtil.showDialogtitleText("操作提示", message, new OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 switch (type) {  
 case 1:  
 cheXiaoFaDan(item);  
 break;  
 case 2:  
 delMyOrders(item);  
 break;  
 case 3:  
 if (Title.equals("发单")) {  
 cancelMyOrders(item);  
 } else {  
 EngineercancelMyOrders(item);  
 }  
 break;  
 default:  
 break;  
 }  
 pubDialogUtil.dismiss();  
 }  
 }, false, "确定");  
 }

private void setGridView(GridView gv, int length, int size) {  
 DisplayMetrics dm = new DisplayMetrics();  
 getWindowManager().getDefaultDisplay().getMetrics(dm);  
 float density = dm.density;  
 int gridviewWidth = (int) (size \* (length + 4) \* density);  
 int itemWidth = (int) (length \* density);  
 LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(  
 gridviewWidth, LinearLayout.LayoutParams.FILL\_PARENT);  
 gv.setLayoutParams(params); //   
 gv.setColumnWidth(itemWidth); //   
 gv.setHorizontalSpacing(5); //   
 gv.setStretchMode(GridView.NO\_STRETCH);  
 gv.setNumColumns(size);   
 }  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (resultCode == RESULT\_OK) {  
 switch (requestCode) {  
 case 3:  
 String dataPath = TakePhotoActivity.getDataPath(data);  
 if (dataPath != null) {  
 ArrayList<String> dataPathList = new ArrayList<String>();  
 dataPathList.add(dataPath);  
 uploadImgMulti(dataPathList);  
 adapter.remove(0);  
 PubImage aa = new PubImage();  
 aa.cover = dataPath;  
 adapter.getList().add(0, aa);  
 if (adapter.getCount() < Constants.MYORDER\_IMAGE\_MAX\_COUNT) {  
 adapter.getList().add(0, itemAdd);  
 }  
 adapter.notifyDataSetChanged();  
 }  
 break;  
 case REQUESTCODE\_TIAO\_JIA:  
 tiaojia.setVisibility(View.GONE);  
 getNetData();  
 break;  
 case REQUESTCODE\_REPORT\_ADD:  
 case REQUESTCODE\_OK\_COMPLITE:// 确认完成  
 showInfor.bill\_statu = "2";  
 showInfor.t\_state = 1;  
 isHaveChange = true;  
 setResultCustomData(showInfor);  
 break;  
 case REQUESTCODE\_TO\_ORDER\_CHANGE\_RECEIVER:  
 mBtnChangeReicever.setVisibility(View.GONE);  
 getNetData();  
 break;  
 default:  
 break;  
 }  
 }  
 }  
}

package com.heiyue.project.view;  
import android.content.Context;  
import android.support.v4.view.ViewPager;  
import android.util.AttributeSet;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
import com.heiyue.ui.LazyViewPager;  
import com.qixiaofu.qxf.R;  
import java.util.List;  
public class MidTab extends LinearLayout implements View.OnClickListener {  
 private Context mContext;  
 private ViewPager mViewpager;  
 private LazyViewPager mLazyViewpager;  
 private View mView;  
 private LinearLayout mLayout1;  
 private List<String> mTextList;  
 private int mSelectedColor;  
 public MidTab(Context context) {  
 this(context, null);  
 }  
 public MidTab(Context context, AttributeSet attrs) {  
 this(context, attrs, 0);  
 }  
 public MidTab(Context context, AttributeSet attrs, int defStyleAttr) {  
 super(context, attrs, defStyleAttr);  
 this.mContext = context;  
 initView();  
 }  
 private void initView() {  
 mView = (View) LayoutInflater.from(mContext).inflate(R.layout.pub\_middle\_tab, null);  
 mLayout1 = (LinearLayout) mView.findViewById(R.id.lly\_all);  
 mTv1 = (TextView) mView.findViewById(R.id.tv\_all);  
 mLine1 = (View) mView.findViewById(R.id.line1);  
 mLayout2 = (LinearLayout) mView.findViewById(R.id.lly\_expend);  
 mTv2 = (TextView) mView.findViewById(R.id.tv\_expend);  
 mLine2 = (View) mView.findViewById(R.id.line2);  
 mLayout3 = (LinearLayout) mView.findViewById(R.id.lly\_income);  
 mTv3 = (TextView) mView.findViewById(R.id.tv\_income);  
 mLine3 = (View) mView.findViewById(R.id.line3);  
 mLayout1.setOnClickListener(this);  
 mLayout2.setOnClickListener(this);  
 mLayout3.setOnClickListener(this);  
 this.setOrientation(LinearLayout.HORIZONTAL);  
 }  
 public void setTexts(List<String> textList) {  
 this.mTextList = textList;  
 }  
 public void setViewPager(ViewPager pager) {  
 this.mViewpager = pager;  
 }  
 public void setViewPager(LazyViewPager pager) {  
 this.mLazyViewpager = pager;  
 }  
 public void setSelectedColor(int color) {  
 this.mSelectedColor = color;  
 }  
 public void setTab(int position) {  
 setColor();  
 switch (position) {  
 case 0:  
 mTv1.setTextColor(mSelectedColor);  
 mLine1.setBackgroundColor(mSelectedColor);  
 mLine1.setVisibility(View.VISIBLE);  
 break;  
 case 1:  
 mTv2.setTextColor(mSelectedColor);  
 mLine2.setBackgroundColor(mSelectedColor);  
 mLine2.setVisibility(View.VISIBLE);  
 break;  
 case 2:  
 mTv3.setTextColor(mSelectedColor);  
 mLine3.setBackgroundColor(mSelectedColor);  
 mLine3.setVisibility(View.VISIBLE);  
 break;  
 }  
 }  
 private void setText() {  
 if(mTextList != null && mTextList.size() >= 3){  
 mTv1.setText(mTextList.get(0));  
 mTv2.setText(mTextList.get(1));  
 mTv3.setText(mTextList.get(2));  
 }  
 }  
 private void setColor() {  
 mTv1.setTextColor(getResources().getColor(R.color.Black\_33));  
 mTv2.setTextColor(getResources().getColor(R.color.Black\_33));  
 mTv3.setTextColor(getResources().getColor(R.color.Black\_33));  
 mLine1.setVisibility(View.GONE);  
 mLine2.setVisibility(View.GONE);  
 mLine3.setVisibility(View.GONE);  
 }  
 @Override  
 public void onClick(View v) {  
 if (mViewpager == null && mLazyViewpager == null) {  
 throw new IllegalArgumentException("Viewpager can not be null...");  
 }  
 switch (v.getId()) {  
 case R.id.lly\_all:  
 setPagerTab(0);  
 setTab(0);  
 break;  
 case R.id.lly\_expend:  
 setPagerTab(1);  
 setTab(1);  
 break;  
 case R.id.lly\_income:  
 setPagerTab(2);  
 setTab(2);  
 break;  
 }  
 }  
 private void setPagerTab(int p) {  
 if (mLazyViewpager != null) {  
 mLazyViewpager.setCurrentItem(p, false);  
 return;  
 }  
 if (mViewpager != null) {  
 mViewpager.setCurrentItem(p, false);  
 }  
 }  
 public void build(){  
 setText();  
 setTab(0);  
 this.addView(mView);  
 }  
}

package com.rocky.stateview;  
import android.content.Context;  
import android.graphics.Typeface;  
import android.graphics.drawable.Drawable;  
import android.util.AttributeSet;  
import android.util.TypedValue;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.LinearLayout;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
import com.heiyue.R;  
import com.rocky.stateview.bean.StateBean;  
import java.util.List;  
public class HorizontalStatesView extends LinearLayout implements HorizontalStatesViewIndicator.OnDrawIndicatorListener {  
 private final Context mContext;  
 private RelativeLayout mTextContainer;  
 private HorizontalStatesViewIndicator mStepsViewIndicator;  
 private List<StateBean> mStepBeanList;  
 private int mComplectingPosition;  
 private int mUnComplectedTextColor;  
 private int mComplectedTextColor;  
 private int mTextSize = 14;//default textSize  
 private TextView mTextView;  
 public HorizontalStatesView(Context context) {  
 this(context, null);  
 }  
 public HorizontalStatesView(Context context, AttributeSet attrs) {  
 this(context, attrs, 0);  
 }  
 public HorizontalStatesView(Context context, AttributeSet attrs, int defStyleAttr) {  
 super(context, attrs, defStyleAttr);  
 mContext = context;  
 mUnComplectedTextColor = mContext.getResources().getColor(R.color.uncompleted\_text\_color);  
 mComplectedTextColor = mContext.getResources().getColor(android.R.color.white);  
 init();  
 }  
 private void init() {  
 View rootView = LayoutInflater.from(mContext).inflate(R.layout.widget\_horizontal\_stepsview, this);  
 mStepsViewIndicator = (HorizontalStatesViewIndicator) rootView.findViewById(R.id.state\_indicator);  
 mStepsViewIndicator.setOnDrawListener(this);  
 mTextContainer = (RelativeLayout) rootView.findViewById(R.id.rl\_text\_container);  
 }  
 public HorizontalStatesView setStepViewTexts(List<StateBean> stepsBeanList) {  
 mStepBeanList = stepsBeanList;  
 mStepsViewIndicator.setStepNum(mStepBeanList);  
 return this;  
 }  
 public HorizontalStatesView setStepViewUnComplectedTextColor(int unComplectedTextColor) {  
 mUnComplectedTextColor = unComplectedTextColor;  
 return this;  
 }  
 public HorizontalStatesView setStepViewComplectedTextColor(int complectedTextColor) {  
 this.mComplectedTextColor = complectedTextColor;  
 return this;  
 }  
 public HorizontalStatesView setStepsViewIndicatorUnCompletedLineColor(int unCompletedLineColor) {  
 mStepsViewIndicator.setUnCompletedLineColor(unCompletedLineColor);  
 return this;  
 }  
 public HorizontalStatesView setStepsViewIndicatorCompletedLineColor(int completedLineColor) {  
 mStepsViewIndicator.setCompletedLineColor(completedLineColor);  
 return this;  
 }

public HorizontalStatesView setStepsViewIndicatorDefaultIcon(Drawable defaultIcon) {  
 mStepsViewIndicator.setDefaultIcon(defaultIcon);  
 return this;  
 }  
 public HorizontalStatesView setStepsViewIndicatorCompleteIcon(Drawable completeIcon) {  
 mStepsViewIndicator.setCompleteIcon(completeIcon);  
 return this;  
 }  
 public HorizontalStatesView setStepsViewIndicatorAttentionIcon(Drawable attentionIcon) {  
 mStepsViewIndicator.setAttentionIcon(attentionIcon);  
 return this;  
 }

public HorizontalStatesView setTextSize(int textSize) {  
 if (textSize > 0) {  
 mTextSize = textSize;  
 }  
 return this;  
 }  
 @Override  
 public void ondrawIndicator() {  
 if (mTextContainer != null) {  
 mTextContainer.removeAllViews();  
 List<Float> complectedXPosition = mStepsViewIndicator.getCircleCenterPointPositionList();  
 if (mStepBeanList != null && complectedXPosition != null && complectedXPosition.size() > 0) {  
 for (int i = 0; i < mStepBeanList.size(); i++) {  
 mTextView = new TextView(getContext());  
 mTextView.setTextSize(TypedValue.COMPLEX\_UNIT\_SP, mTextSize);  
 mTextView.setText(mStepBeanList.get(i).getName());  
 int spec = MeasureSpec.makeMeasureSpec(0, MeasureSpec.UNSPECIFIED);  
 mTextView.measure(spec, spec);  
 // getMeasuredWidth  
 int measuredWidth = mTextView.getMeasuredWidth();  
 mTextView.setX(complectedXPosition.get(i) - measuredWidth / 2);  
 mTextView.setLayoutParams(new ViewGroup.LayoutParams(ViewGroup.LayoutParams.WRAP\_CONTENT, ViewGroup.LayoutParams.WRAP\_CONTENT));  
 if (i <= mComplectingPosition) {  
 mTextView.setTypeface(null, Typeface.BOLD);  
 mTextView.setTextColor(mComplectedTextColor);  
 } else {  
 mTextView.setTextColor(mUnComplectedTextColor);  
 }  
 if(mStepBeanList.get(i).getState()== StateBean.STEP\_COMPLETED){  
 mTextView.setTypeface(null, Typeface.BOLD);  
 mTextView.setTextColor(mComplectedTextColor);  
 }else{  
 mTextView.setTextColor(mUnComplectedTextColor);  
 }  
 mTextContainer.addView(mTextView);  
 }  
 }  
 }  
 }  
}

package com.heiyue.project.ui.home;  
import android.os.Bundle;  
import android.support.annotation.Nullable;  
import android.support.v7.widget.Toolbar;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.ListView;  
import android.widget.TextView;  
import com.gyf.barlibrary.ImmersionBar;  
import com.handmark.pulltorefresh.library.PullToRefreshBase;  
import com.handmark.pulltorefresh.library.PullToRefreshListView;  
import com.heiyue.base.ListItemClickListener;  
import com.heiyue.net.NetResponse;  
import com.heiyue.net.RequestCallBack;  
import com.heiyue.project.adapter.RepositoryListAdapter;  
import com.heiyue.project.adapter.RepositoryTypeListAdapter;  
import com.heiyue.project.adapter.SortTypeListAdapter;  
import com.heiyue.project.base.BaseFragment;  
import com.heiyue.project.bean.RepositoryInfoBean;  
import com.heiyue.project.bean.RepositoryTypeBean;  
import com.heiyue.project.bean.PubSortType;  
import com.heiyue.project.config.Constants;  
import com.heiyue.project.ui.RepositoryDetailActivity;  
import com.heiyue.project.ui.index.SearchRepositoryActivity;  
import com.heiyue.ui.ButtonTouch;  
import com.heiyue.ui.ImageViewTouch;  
import com.heiyue.util.LogOut;  
import com.qixiaofu.qxf.R;  
import java.util.ArrayList;  
import java.util.List;  
import butterknife.BindView;  
import butterknife.OnClick;  
public class RepositoryFragment extends BaseFragment implements PullToRefreshBase.OnRefreshListener2<ListView> {  
 private static final int REQUESTCODE\_REPOSITORY\_DETAIL = 1;  
 private static final int REQUESTCODE\_SEARCH\_REPOSITORY = 2;  
 @BindView(R.id.btWriteRepository)  
 public ButtonTouch mBtWriteRepository;  
 private View mView;  
 private PullToRefreshListView pullToRefreshListView;  
 private RepositoryListAdapter repositoryAdapter;  
 private int currentPage = 1;  
 private View mEmptyView;  
 @BindView(R.id.toolbar)  
 public Toolbar toolbar;  
 @BindView(R.id.layoutContent)  
 public LinearLayout mLayoutContent;  
 @BindView(R.id.right\_btn)  
 public ImageViewTouch mButtonSearch;  
 @BindView(R.id.ll\_all\_type)  
 public LinearLayout mLLAllType;  
 @BindView(R.id.ll\_all\_brand)  
 public LinearLayout mLLAllBrand;  
 @BindView(R.id.ll\_all\_sort)  
 public LinearLayout mLLAllSort;  
 @BindView(R.id.tv\_all\_type)  
 public TextView mTvAllType;  
 @BindView(R.id.tv\_all\_brand)  
 public TextView mTvAllBrand;  
 @BindView(R.id.tv\_all\_sort)  
 public TextView mTvAllSort;  
 @BindView(R.id.iv\_all\_type)  
 public ImageView mIvAllType;  
 @BindView(R.id.iv\_all\_brand)  
 public ImageView mIvAllBrand;  
 @BindView(R.id.iv\_all\_sort)  
 public ImageView mIvAllSort;  
 @BindView(R.id.ll\_choose)  
 public LinearLayout mChooseView;  
 @BindView(R.id.lv\_all\_type)  
 public ListView mLvAllType;  
 @BindView(R.id.lv\_all\_brand)  
 public ListView mLvAllBrand;  
 @BindView(R.id.lv\_all\_sort)  
 public ListView mLvAllSort;  
 private RepositoryTypeListAdapter mAllTypeListAdapter;  
 private RepositoryTypeListAdapter mAllBrandListAdapter;  
 private SortTypeListAdapter mAllSortListAdapter;  
 private List<RepositoryTypeBean> mAllBrandList;  
 private String mLastTypeId;  
 private String mCurrentTypeId = "0";  
 private String mSortId = "0";  
 private List<RepositoryTypeBean> mAllTypeList;  
 public static RepositoryFragment getInstance() {  
 RepositoryFragment instance = new RepositoryFragment();  
 return instance;  
 }  
 @Override  
 public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  
 mView = getLayoutInflater(getArguments()).inflate(R.layout.fragment\_repository, null);  
 pullToRefreshListView = (PullToRefreshListView) mView.findViewById(R.id.listview);  
 mEmptyView = inflater.inflate(R.layout.include\_empty\_view, null);  
 pullToRefreshListView.setEmptyView(mEmptyView);  
 return mView;  
 }  
 @Override  
 public void onViewCreated(View view, Bundle savedInstanceState) {  
 super.onViewCreated(view, savedInstanceState);  
 iniData();  
 }  
 private void iniData() {  
 mChooseView.setVisibility(View.GONE);  
 mAllTypeListAdapter = new RepositoryTypeListAdapter(getActivity(), new ListItemClickListener() {  
 @Override  
 public void itemClick(int type, Object object, int position, Object params) {  
 RepositoryTypeBean repositoryType = (RepositoryTypeBean) object;  
 mCurrentTypeId = repositoryType.id;  
 mTvAllType.setText(repositoryType.name);  
 if("0".equals(mCurrentTypeId)){  
 mTvAllType.setText("all");  
 mTvAllBrand.setText("all");  
 }  
 mAllBrandList.clear();  
 if(repositoryType.smallList != null){  
 mAllBrandList.add(new RepositoryTypeBean("0", "全部"));  
 mAllBrandList.addAll(repositoryType.smallList);  
 }  
 mAllBrandListAdapter.setData(mAllBrandList);  
 mAllBrandListAdapter.notifyDataSetChanged();  
 mChooseView.setVisibility(View.GONE);  
 mLvAllType.setVisibility(View.INVISIBLE);  
 mIvAllType.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_bottom));  
 currentPage = Constants.PAGE\_START;  
 getRepositoryList();  
 }  
 });  
 mAllTypeList = new ArrayList<RepositoryTypeBean>();  
 mLvAllType.setAdapter(mAllTypeListAdapter);  
 mLvAllType.setVisibility(View.GONE);  
 mAllBrandListAdapter = new RepositoryTypeListAdapter(getActivity(), new ListItemClickListener() {  
 @Override  
 public void itemClick(int type, Object object, int position, Object params) {  
 RepositoryTypeBean repositoryType = (RepositoryTypeBean) object;  
 mCurrentTypeId = repositoryType.id;  
 mTvAllBrand.setText(repositoryType.name);  
 if("0".equals(mCurrentTypeId)){  
 mTvAllBrand.setText("all");  
 }  
 mChooseView.setVisibility(View.GONE);  
 mLvAllBrand.setVisibility(View.INVISIBLE);  
 mIvAllBrand.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_bottom));  
 currentPage = Constants.PAGE\_START;  
 getRepositoryList();  
 }  
 });  
 mAllBrandList = new ArrayList<RepositoryTypeBean>();  
 mAllBrandList.add(new RepositoryTypeBean("0", "all"));  
 mLvAllBrand.setAdapter(mAllBrandListAdapter);  
 mAllSortListAdapter = new SortTypeListAdapter(getActivity(), new ListItemClickListener() {  
 @Override  
 public void itemClick(int type, Object object, int position, Object params) {  
 PubSortType sortType = (PubSortType) object;  
 mSortId = sortType.id;  
 mTvAllSort.setText(sortType.name);  
 mChooseView.setVisibility(View.GONE);  
 mLvAllSort.setVisibility(View.INVISIBLE);  
 mIvAllSort.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_bottom));  
 currentPage = Constants.PAGE\_START;  
 getRepositoryList();  
 }  
 });  
 List<PubSortType> sortTypeList = new ArrayList<PubSortType>();  
 sortTypeList.add(new PubSortType("0", "update"));  
 sortTypeList.add(new PubSortType("1", "delete"));  
 sortTypeList.add(new PubSortType("2", "add"));  
 mAllSortListAdapter.setData(sortTypeList);  
 mLvAllSort.setAdapter(mAllSortListAdapter);  
 pullToRefreshListView.setOnRefreshListener(this);  
 pullToRefreshListView.setMode(PullToRefreshBase.Mode.BOTH);  
 repositoryAdapter = new RepositoryListAdapter(getActivity(), new ListItemClickListener() {  
 @Override  
 public void itemClick(int type, Object object, int position, Object params) {  
 RepositoryInfoBean info = (RepositoryInfoBean) object;  
 switch (type) {  
 case 1:  
 mChooseView.setVisibility(View.GONE);  
 RepositoryDetailActivity.startActivityForResult(getActivity(), info, REQUESTCODE\_REPOSITORY\_DETAIL);  
 break;  
 }  
 }  
 });  
 pullToRefreshListView.setAdapter(repositoryAdapter);  
 }  
 @Override  
 public void setUserVisibleHint(boolean isVisibleToUser) {  
 super.setUserVisibleHint(isVisibleToUser);  
 if (isVisibleToUser) {  
 getRepositoryType();  
 getRepositoryList();  
 }  
 }  
 protected void getRepositoryType() {  
 dao.getRepositoryTypeList(new RequestCallBack<List<RepositoryTypeBean>>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<List<RepositoryTypeBean>> result) {  
 if (result.netMsg.success) {  
 if (result.content != null && result.content.size() > 0) {  
 mAllTypeList.clear();  
 mAllTypeList.add(0, new RepositoryTypeBean("0", "全部"));  
 mAllTypeList.addAll(mAllTypeList.size(), result.content);  
 mAllTypeListAdapter.setData(mAllTypeList);  
 }  
 }  
 }  
 });  
 }  
 private void getRepositoryList() {  
 if (!mCurrentTypeId.equals(mLastTypeId)) {  
 repositoryAdapter.getList().clear();  
 repositoryAdapter.notifyDataSetChanged();  
 }  
 dao.getRepositoryList(mCurrentTypeId, mSortId, null, currentPage, new RequestCallBack<List<RepositoryInfoBean>>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<List<RepositoryInfoBean>> result) {  
 pullToRefreshListView.onRefreshComplete();  
 mLastTypeId = mCurrentTypeId;  
 if (result.netMsg.success) {  
 if (result.content != null && result.content.size() > 0) {  
 if (currentPage == Constants.PAGE\_START ) {  
 repositoryAdapter.setData(result.content);  
 } else {  
 repositoryAdapter.addData(result.content);  
 }  
 if(result.content.size() == Constants.PAGE\_DEFAULT\_COUNT){  
 pullToRefreshListView.setMode(PullToRefreshBase.Mode.BOTH);  
 }else{  
 pullToRefreshListView.setMode(PullToRefreshBase.Mode.PULL\_FROM\_START);  
 }  
 }  
 }  
 }  
 });  
 }  
 @OnClick({R.id.right\_btn, R.id.ll\_choose, R.id.layoutContent, R.id.btWriteRepository, R.id.ll\_all\_type, R.id.ll\_all\_brand, R.id.ll\_all\_sort})  
 public void onClick(View view) {  
 switch (view.getId()) {  
 case R.id.right\_btn:  
 SearchRepositoryActivity.startActivity(getActivity(), REQUESTCODE\_SEARCH\_REPOSITORY);  
 break;  
 case R.id.ll\_choose:  
 mChooseView.setVisibility(View.GONE);  
 mLvAllType.setVisibility(View.GONE);  
 mLvAllBrand.setVisibility(View.GONE);  
 mIvAllType.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_bottom));  
 mIvAllBrand.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_bottom));  
 mIvAllSort.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_bottom));  
  
 break;  
 case R.id.layoutContent:  
// mChooseView.setVisibility(View.GONE);  
 mLvAllType.setVisibility(View.GONE);  
 mLvAllBrand.setVisibility(View.GONE);  
 mLvAllSort.setVisibility(View.GONE);  
 break;  
 case R.id.btWriteRepository:  
 break;  
 case R.id.ll\_all\_type:  
 setState();  
 if (mLvAllType.getVisibility() == View.VISIBLE) {  
 mLvAllType.setVisibility(View.INVISIBLE);  
 mIvAllType.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_bottom));  
 mChooseView.setVisibility(View.GONE);  
 } else {  
 mLvAllType.setVisibility(View.VISIBLE);  
 mIvAllType.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_top));  
 }  
 mLvAllBrand.setVisibility(View.INVISIBLE);  
 mLvAllSort.setVisibility(View.INVISIBLE);  
 break;  
 case R.id.ll\_all\_brand:  
 setState();  
 if(mAllBrandListAdapter.getList().size() == 0){  
 mChooseView.setVisibility(View.GONE);  
 }  
 if (mLvAllBrand.getVisibility() == View.VISIBLE) {  
 mLvAllBrand.setVisibility(View.INVISIBLE);  
 mIvAllBrand.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_bottom));  
 mChooseView.setVisibility(View.GONE);  
 } else {  
 mLvAllBrand.setVisibility(View.VISIBLE);  
 mIvAllBrand.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_top));  
 }  
 mLvAllType.setVisibility(View.INVISIBLE);  
 mLvAllSort.setVisibility(View.INVISIBLE);  
 break;  
 case R.id.ll\_all\_sort:  
 setState();  
 if (mLvAllSort.getVisibility() == View.VISIBLE) {  
 mLvAllSort.setVisibility(View.INVISIBLE);  
 mIvAllSort.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_bottom));  
 mChooseView.setVisibility(View.GONE);  
 } else {  
 mLvAllSort.setVisibility(View.VISIBLE);  
 mIvAllSort.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_top));  
 }  
 mLvAllType.setVisibility(View.INVISIBLE);  
 mLvAllBrand.setVisibility(View.INVISIBLE);  
 break;  
 }  
 }  
 @Override  
 public void onPullDownToRefresh(PullToRefreshBase<ListView> refreshView) {  
 currentPage = Constants.PAGE\_START;  
 getRepositoryList();  
 }  
 @Override  
 public void onPullUpToRefresh(PullToRefreshBase<ListView> refreshView) {  
 currentPage++;  
 getRepositoryList();  
 }  
 private void upGoodVote(final RepositoryInfoBean info) {  
 dao.upRepositoryVote(info.post\_id, new RequestCallBack<String>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<String> result) {  
 if (result.netMsg.success) {  
 info.upvote\_num = String.valueOf(Integer.valueOf(info.upvote\_num) + 1);  
 repositoryAdapter.notifyDataSetChanged();  
 }  
 }  
 });  
 }  
 private void setState() {  
 mChooseView.setVisibility(View.VISIBLE);  
 mIvAllType.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_bottom));  
 mIvAllBrand.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_bottom));  
 mIvAllSort.setImageDrawable(getResources().getDrawable(R.drawable.arrow\_bottom));  
 }  
 @Override  
 protected void immersionInit() {  
 ImmersionBar.with(getActivity())  
 .titleBar(toolbar)  
 .statusBarColor(R.color.black)  
 .statusBarDarkFont(false)  
 .init();  
 }  
}

public class Crowd extends Fragment{  
 private View mView;  
 private PullToRefreshListView mListView;  
 private CrowdTestAdapter mAdapter;  
 private List<String> mData;  
  
 public static Crowd getInstance() {  
 Crowd instance = new Crowd ();  
 return instance;  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  
 mView = inflater.inflate(R.layout.fragment\_crowd\_test, null);  
 initView();  
 return mView;  
 }  
  
 private void initView() {  
 mListView = (PullToRefreshListView)mView.findViewById(R.id.lv);  
 mData = new ArrayList<String>();  
 for(int i=0;i< 10;i++){  
 mData.add(i);  
 }  
 mListView.setOnRefreshListener(new PullToRefreshBase.OnRefreshListener<ListView>() {  
 @Override  
 public void onRefresh(PullToRefreshBase<ListView> refreshView) {  
 initData();  
 }  
 });  
 }  
 private void initData() {  
 ServerDao dao = new ServerDao(getActivity());  
 dao.raiseList(1, new RequestCallBack<List<CrowFundingInfoBean>>() {  
 @Override  
 public void start() {  
 }  
 @Override  
 public void finish(NetResponse<List<CrowFundingInfoBean>> result) {  
 mListView.onRefreshComplete();  
 for(int i=0;i< result.content.size();i++){  
 mData.add(result.content.get(i).title);  
 }  
 mAdapter = new CrowdTestAdapter(mData, getActivity());  
 mListView.setAdapter(mAdapter);  
 mAdapter.notifyDataSetChanged();  
 onNetDataDown(result, adapter, result.content);  
 }  
 });  
 }  
}