**移动设备开发-实验作业#6-10(合并提交)**

**请同学完成作业后，**以word**档案方式提供给学习委员**

**档案的主题格式必须严格遵守如下所示，否则扣分:**

**[Android] [班级][学号][姓名][实验作业#6-10].docx (注意:”[]”括号符号不能省略)**

**例如: [Android][21大数据1][** **202124114343][王大明][实验作业#6-10].docx (doc也可)**

**实验作业缴交截止日: 2024/5/28**

**实验作业#6:** 请结合教科书课本的范例5-1和5-3 ，编写一个具有较完善功能的后台音乐播放器。(若有能力，可加上自行创意内容或功能。若无想法，简单完成基本功能即可。)

请将设计好之用户使用之GUI页面截图，并提供用户GUI页面的xml档案内容与java编程。此外执行结果也须提供。

**实验结果:**

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

**MainActivity.java**

package com.example.work6;  
  
import com.example.work6.AudioItem;  
import android.content.ContentUris;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.database.Cursor;  
import android.graphics.Color;  
import android.graphics.Paint;  
import android.media.AudioManager;  
import android.media.MediaPlayer;  
import android.media.audiofx.Visualizer;  
import android.os.Build;  
import android.os.Bundle;  
import android.os.Handler;  
import android.os.Looper;  
import android.provider.MediaStore;  
import android.util.Log;  
import android.view.SurfaceHolder;  
import android.view.SurfaceView;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.ImageButton;  
import android.widget.ListView;  
import android.widget.SeekBar;  
import android.widget.TextView;  
import android.Manifest;  
import android.graphics.Canvas;  
import android.graphics.PorterDuff;  
import android.net.Uri;  
import android.widget.\*;  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.core.content.ContextCompat;  
  
import java.io.File;  
import java.io.IOException;  
import java.util.ArrayList;  
  
public class MainActivity extends AppCompatActivity {  
 private MediaPlayer mediaPlayer;  
 private Visualizer visualizer;  
 private ArrayList<AudioItem> playlist = new ArrayList<>();  
 private ArrayAdapter<AudioItem> playlistAdapter;  
 private int currentPosition = 0;  
 private boolean isPlaying = false;  
 private boolean isRepeat = false;  
 private Handler progressHandler = new Handler();  
 private Paint visualizerPaint = new Paint();  
 private SeekBar progressBar;  
 private TextView currentTime;  
 private TextView totalTime;  
 private ImageButton btnRepeat;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_player*);  
  
 // 初始化组件  
 progressBar = findViewById(R.id.*songProgress*);  
 currentTime = findViewById(R.id.*currentTime*);  
 totalTime = findViewById(R.id.*totalTime*);  
 btnRepeat = findViewById(R.id.*btnRepeat*);  
 SeekBar progressBar = findViewById(R.id.*songProgress*);  
 ListView playlistView = findViewById(R.id.*playlist*);  
 SurfaceView visualizerView = findViewById(R.id.*visualizerView*);  
 ImageButton btnPlay = findViewById(R.id.*btnPlay*);  
 ImageButton btnNext = findViewById(R.id.*btnNext*);  
 ImageButton btnPrev = findViewById(R.id.*btnPrev*);  
 ImageButton btnRepeat = findViewById(R.id.*btnRepeat*);  
 Button btnBrowse = findViewById(R.id.*btnBrowse*);  
 Button btnExit = findViewById(R.id.*btnExit*);  
 TextView currentTime = findViewById(R.id.*currentTime*);  
 TextView totalTime = findViewById(R.id.*totalTime*);  
  
 // 初始化播放列表适配器  
 playlistAdapter = new ArrayAdapter<AudioItem>(  
 this,  
 android.R.layout.*simple\_list\_item\_1*,  
 playlist  
 ) {  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
 TextView textView = (TextView) super.getView(position, convertView, parent);  
 textView.setTextColor(Color.*WHITE*);  
 textView.setText(playlist.get(position).getName());  
 return textView;  
 }  
 };  
 playlistView.setAdapter(playlistAdapter);  
  
 // 请求存储权限  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*TIRAMISU*) {  
 // Android 13+ 使用 READ\_MEDIA\_AUDIO  
 if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*READ\_MEDIA\_AUDIO*)  
 != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this,  
 new String[]{Manifest.permission.*READ\_MEDIA\_AUDIO*}, 1);  
 } else {  
 loadAudioFiles(); // 权限已授予，加载音乐  
 }  
 } else {  
 // Android 12 及以下使用 READ\_EXTERNAL\_STORAGE  
 if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*READ\_EXTERNAL\_STORAGE*)  
 != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this,  
 new String[]{Manifest.permission.*READ\_EXTERNAL\_STORAGE*}, 1);  
 } else {  
 loadAudioFiles(); // 权限已授予，加载音乐  
 }  
 }  
  
 // 文件选择监听  
 btnBrowse.setOnClickListener(v -> openFilePicker());  
  
 // 退出应用  
 btnExit.setOnClickListener(v -> finish());  
  
 // 播放控制  
 btnPlay.setOnClickListener(v -> togglePlayback());  
 btnNext.setOnClickListener(v -> playNext());  
 btnPrev.setOnClickListener(v -> playPrevious());  
 btnRepeat.setOnClickListener(v -> toggleRepeatMode());  
  
 // 播放列表点击事件  
 playlistView.setOnItemClickListener((parent, view, position, id) -> {  
 currentPosition = position;  
 AudioItem selectedItem = playlist.get(position); // 这里获取的是AudioItem对象  
 playMusic(selectedItem);  
 });  
  
 // 进度条控制  
 progressBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
 if (fromUser && mediaPlayer != null) {  
 mediaPlayer.seekTo(progress);  
 }  
 }  
  
 @Override public void onStartTrackingTouch(SeekBar seekBar) {}  
 @Override public void onStopTrackingTouch(SeekBar seekBar) {}  
 });  
  
 // 可视化初始化  
 visualizerPaint.setColor(Color.*argb*(200, 76, 175, 80));  
 visualizerView.getHolder().addCallback(new SurfaceHolder.Callback() {  
 @Override  
 public void surfaceCreated(SurfaceHolder holder) {  
 setupVisualizer();  
 }  
  
 @Override  
 public void surfaceChanged(SurfaceHolder holder, int format, int width, int height) {}  
  
 @Override  
 public void surfaceDestroyed(SurfaceHolder holder) {  
 releaseVisualizer();  
 }  
 });  
 }  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
 if (mediaPlayer != null && !isPlaying) {  
 releaseVisualizer();  
 mediaPlayer.release();  
 mediaPlayer = null;  
 }  
 }  
  
 private void openFilePicker() {  
 Intent intent = new Intent(Intent.*ACTION\_OPEN\_DOCUMENT*);  
 intent.addCategory(Intent.*CATEGORY\_OPENABLE*);  
 intent.setType("audio/\*");  
 startActivityForResult(intent, 2);  
 }  
  
 private void loadAudioFiles() {  
 playlist.clear();  
  
 Uri collection = MediaStore.Audio.Media.*EXTERNAL\_CONTENT\_URI*;  
 String[] projection = {  
 MediaStore.Audio.Media.*\_ID*,  
 MediaStore.Audio.Media.*DISPLAY\_NAME* };  
 String selection = MediaStore.Audio.Media.*IS\_MUSIC* + " != 0";  
  
 try (Cursor cursor = getContentResolver().query(  
 collection,  
 projection,  
 selection,  
 null,  
 MediaStore.Audio.Media.*DATE\_ADDED* + " DESC"  
 )) {  
 if (cursor != null) {  
 int idColumn = cursor.getColumnIndexOrThrow(MediaStore.Audio.Media.*\_ID*);  
 int nameColumn = cursor.getColumnIndexOrThrow(MediaStore.Audio.Media.*DISPLAY\_NAME*);  
  
 while (cursor.moveToNext()) {  
 long id = cursor.getLong(idColumn);  
 String name = cursor.getString(nameColumn);  
 playlist.add(new AudioItem(id, name));  
 }  
 playlistAdapter.notifyDataSetChanged();  
 }  
 }  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (requestCode == 2 && resultCode == *RESULT\_OK* && data != null) {  
 Uri uri = data.getData();  
 try {  
 // 直接使用 URI 播放  
 playMusicFromUri(uri);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 private void playMusicFromUri(Uri uri) throws IOException {  
 releaseMediaPlayer();  
 mediaPlayer = new MediaPlayer();  
 mediaPlayer.setDataSource(this, uri); // 使用 URI 设置数据源  
 mediaPlayer.prepare();  
 mediaPlayer.start();  
 isPlaying = true;  
 setupProgressUpdater();  
 updatePlayButton();  
 setupVisualizer();  
 }  
  
 private void togglePlayback() {  
 if (mediaPlayer == null && !playlist.isEmpty()) {  
 playMusic(playlist.get(currentPosition));  
 } else if (mediaPlayer != null) {  
 if (isPlaying) {  
 pauseMusic();  
 } else {  
 mediaPlayer.start();  
 isPlaying = true;  
 setupProgressUpdater(); // 恢复进度更新  
 }  
 updatePlayButton();  
 }  
 }  
  
 private void playMusic(AudioItem audioItem) {  
 try {  
 releaseMediaPlayer();  
 mediaPlayer = new MediaPlayer();  
  
 // 使用MediaStore ID构建URI  
 Uri contentUri = ContentUris.*withAppendedId*(  
 MediaStore.Audio.Media.*EXTERNAL\_CONTENT\_URI*,  
 audioItem.getId()  
 );  
  
 mediaPlayer.setDataSource(this, contentUri);  
 mediaPlayer.setOnPreparedListener(mp -> {  
 int duration = mp.getDuration();  
 progressBar.setMax(duration);  
 totalTime.setText(formatTime(duration));  
 mp.start();  
 isPlaying = true;  
 updatePlayButton();  
 setupProgressUpdater();  
 new Handler(Looper.*getMainLooper*()).postDelayed(() -> {  
 setupVisualizer();  
 }, 200);  
 });  
 mediaPlayer.prepareAsync();  
  
 } catch (IOException e) {  
 e.printStackTrace();  
 Toast.*makeText*(this, "无法播放文件: " + audioItem.getName(), Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 private void setupProgressUpdater() {  
 progressHandler.removeCallbacksAndMessages(null); // 清除旧任务  
  
 progressHandler.post(new Runnable() {  
 @Override  
 public void run() {  
 if (mediaPlayer != null && mediaPlayer.isPlaying()) {  
 int current = mediaPlayer.getCurrentPosition();  
 int total = mediaPlayer.getDuration();  
  
 // 更新进度条  
 progressBar.setProgress(current);  
 progressBar.setMax(total);  
  
 // 更新时间显示  
 currentTime.setText(formatTime(current));  
 totalTime.setText(formatTime(total));  
  
 // 每500ms更新一次  
 progressHandler.postDelayed(this, 500);  
 }  
 }  
 });  
 }  
  
 private String formatTime(int milliseconds) {  
 int seconds = (milliseconds / 1000) % 60;  
 int minutes = (milliseconds / (1000 \* 60)) % 60;  
 return String.*format*("%02d:%02d", minutes, seconds);  
 }  
  
 private void pauseMusic() {  
 if (mediaPlayer != null && mediaPlayer.isPlaying()) {  
 mediaPlayer.pause();  
 isPlaying = false;  
 updatePlayButton();  
 }  
 }  
  
 private void playNext() {  
 if (!playlist.isEmpty()) {  
 currentPosition = (currentPosition + 1) % playlist.size();  
 playMusic(playlist.get(currentPosition));  
 }  
 }  
  
 private void playPrevious() {  
 if (!playlist.isEmpty()) {  
 currentPosition = (currentPosition - 1 < 0) ? playlist.size() - 1 : currentPosition - 1;  
 playMusic(playlist.get(currentPosition));  
 }  
 }  
  
 private void toggleRepeatMode() {  
 isRepeat = !isRepeat;  
 btnRepeat.setImageResource(isRepeat ? R.drawable.*ic\_repeat\_on* : R.drawable.*ic\_repeat\_off*);  
 }  
  
 private void updatePlayButton() {  
 runOnUiThread(() -> {  
 ImageButton btnPlay = findViewById(R.id.*btnPlay*);  
 if (mediaPlayer != null && mediaPlayer.isPlaying()) {  
 btnPlay.setImageResource(R.drawable.*ic\_pause*);  
 btnPlay.setContentDescription("暂停");  
 } else {  
 btnPlay.setImageResource(R.drawable.*ic\_play*);  
 btnPlay.setContentDescription("播放");  
 }  
 });  
 }  
  
 private void setupVisualizer() {  
 if (mediaPlayer == null) return;  
 releaseVisualizer();  
  
 int audioSessionId = mediaPlayer.getAudioSessionId();  
 if (audioSessionId == AudioManager.*ERROR*) { // 检查无效会话 ID  
 Log.*e*("Visualizer", "Invalid audio session ID");  
 return;  
 }  
  
 try {  
 visualizer = new Visualizer(audioSessionId);  
 visualizer.setCaptureSize(Visualizer.*getCaptureSizeRange*()[1]);  
 visualizer.setDataCaptureListener(new Visualizer.OnDataCaptureListener() {  
 @Override  
 public void onWaveFormDataCapture(Visualizer visualizer, byte[] waveform, int samplingRate) {  
 drawWaveform(waveform);  
 }  
  
 @Override  
 public void onFftDataCapture(Visualizer visualizer, byte[] fft, int samplingRate) {}  
 }, Visualizer.*getMaxCaptureRate*(), true, false);  
 visualizer.setEnabled(true);  
 } catch (RuntimeException e) {  
 Log.*e*("Visualizer", "初始化失败: " + e.getMessage());  
 // 处理模拟器兼容性问题  
 if (Build.*FINGERPRINT*.contains("generic")) {  
 Toast.*makeText*(this, "模拟器不支持音频可视化", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 }  
  
 private void drawWaveform(byte[] waveform) {  
 SurfaceHolder holder = ((SurfaceView) findViewById(R.id.*visualizerView*)).getHolder();  
 Canvas canvas = holder.lockCanvas();  
 if (canvas != null) {  
 try {  
 // 清除画布  
 canvas.drawColor(Color.*TRANSPARENT*, PorterDuff.Mode.*CLEAR*);  
  
 // 绘制波形  
 float width = canvas.getWidth();  
 float height = canvas.getHeight();  
 float centerY = height / 2;  
  
 for (int i = 0; i < waveform.length; i++) {  
 float x = i \* width / waveform.length;  
 float y = (Math.*abs*(waveform[i]) / 128.0f) \* height;  
 canvas.drawLine(x, centerY - y, x, centerY + y, visualizerPaint);  
 }  
 } finally {  
 holder.unlockCanvasAndPost(canvas);  
 }  
 }  
 }  
  
 private void releaseMediaPlayer() {  
 if (mediaPlayer != null) {  
 try {  
 mediaPlayer.reset(); // 重置状态  
 mediaPlayer.release(); // 释放资源  
 } catch (Exception e) {  
 Log.*e*("MediaPlayer", "释放异常: " + e.getMessage());  
 }  
 mediaPlayer = null;  
 isPlaying = false;  
 updatePlayButton();  
 }  
 }  
  
 private void releaseVisualizer() {  
 if (visualizer != null) {  
 visualizer.setEnabled(false);  
 visualizer.release();  
 visualizer = null;  
 }  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 releaseMediaPlayer();  
 releaseVisualizer();  
 progressHandler.removeCallbacksAndMessages(null);  
 }  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 if (grantResults.length > 0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) {  
 if (requestCode == 1) {  
 loadAudioFiles(); // 权限授予后加载音乐  
 }  
 } else {  
 Toast.*makeText*(this, "需要权限才能加载音乐", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}

**AudioItem.java**

package com.example.work6;  
  
public class AudioItem {  
 private long id;  
 private String name;  
  
 public AudioItem(long id, String name) {  
 this.id = id;  
 this.name = name;  
 }  
  
 public long getId() { return id; }  
 public String getName() { return name; }  
  
 @Override  
 public String toString() {  
 return name.replace(".mp3", "");  
 }  
}

**activity\_player.xml**

package com.example.work6;  
public class AudioItem {  
 private long id;  
 private String name;  
 public AudioItem(long id, String name) {  
 this.id = id;  
 this.name = name;  
 }  
 public long getId() { return id; }  
 public String getName() { return name; }  
 @Override  
 public String toString() {  
 return name.replace(".mp3", "");  
 }  
}

**实验作业#7:** 请参考教科书课本的范例6-2与6-3 ，编写一个能够调用JavaScript脚本的程序，其脚本能够计算身体BMI数值(数学公式可自己设计或参考网络上信息)，也就是输入身高与体重信息后能够显示BMI数值。

请将设计好之用户使用之GUI页面截图，并提供用户GUI页面的xml档案内容、java编程、html档案内容。此外执行结果也须提供。

**实验结果:**

图形用户界面, 应用程序

AI 生成的内容可能不正确。

**MainActivity.java**

package com.example.work7;  
import android.os.Bundle;  
import android.webkit.WebSettings;  
import android.webkit.WebView;  
import androidx.appcompat.app.AppCompatActivity;  
public class MainActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 WebView webView = findViewById(R.id.*webView*);  
 setupWebView(webView);  
 }  
  
 private void setupWebView(WebView webView) {  
 WebSettings webSettings = webView.getSettings();  
 webSettings.setJavaScriptEnabled(true);  
 webSettings.setAllowFileAccess(true);  
 webSettings.setDomStorageEnabled(true);  
  
 // 加载本地HTML文件  
 webView.loadUrl("file:///android\_asset/bmi\_calculator.html");  
 }  
}

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <WebView  
 android:id="@+id/webView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"/>  
  
</LinearLayout>

**Bmi\_caculator.html**

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="UTF-8">  
 <title>BMI Calculator</title>  
 <style>  
 body {  
 font-family: Arial, sans-serif;  
 max-width: 400px;  
 margin: 20px auto;  
 padding: 20px;  
 background-color: #f5f5f5;  
 }  
 .calculator {  
 background: white;  
 padding: 25px;  
 border-radius: 10px;  
 box-shadow: 0 0 10px rgba(0,0,0,0.1);  
 }  
 input {  
 width: 95%;  
 padding: 10px;  
 margin: 10px 0;  
 border: 1px solid #ddd;  
 border-radius: 5px;  
 font-size: 16px;  
 }  
 button {  
 background-color: #4CAF50;  
 color: white;  
 padding: 12px 24px;  
 border: none;  
 border-radius: 5px;  
 cursor: pointer;  
 width: 100%;  
 font-size: 16px;  
 }  
 #result {  
 margin-top: 20px;  
 padding: 15px;  
 border-radius: 5px;  
 text-align: center;  
 }  
 </style>  
</head>  
<body>  
<div class="calculator">  
 <h2>BMI 计算器</h2>  
 <input type="number" id="height" placeholder="身高（米）" step="0.01">  
 <input type="number" id="weight" placeholder="体重（千克）" step="0.1">  
 <button onclick="calculateBMI()">计算 BMI</button>  
 <div id="result"></div>  
</div>  
  
<script>  
 function calculateBMI() {  
 const height = parseFloat(document.getElementById('height').value);  
 const weight = parseFloat(document.getElementById('weight').value);  
  
 if (!height || !weight || height <= 0 || weight <= 0) {  
 showResult("请输入有效的数值！", "#ff4444");  
 return;  
 }  
  
 const bmi = weight / (height \* height);  
 showResultWithColor(bmi);  
 }  
  
 function showResultWithColor(bmi) {  
 const resultDiv = document.getElementById('result');  
 const category = getBMICategory(bmi);  
  
 resultDiv.innerHTML = `  
 <p style="font-size: 20px; margin: 5px 0;">BMI 值: <strong>${bmi.toFixed(1)}</strong></p>  
 <p style="color: ${category.color}; font-weight: bold;">${category.status}</p>  
 `;  
 resultDiv.style.backgroundColor = category.background;  
 }  
  
 function getBMICategory(bmi) {  
 if (bmi < 18.5) return {  
 status: "体重过轻",  
 color: "#2196F3",  
 background: "#e3f2fd"  
 };  
 if (bmi < 24) return {  
 status: "正常范围",  
 color: "#4CAF50",  
 background: "#e8f5e9"  
 };  
 if (bmi < 28) return {  
 status: "体重过重",  
 color: "#FF9800",  
 background: "#fff3e0"  
 };  
 return {  
 status: "肥胖",  
 color: "#f44336",  
 background: "#ffebee"  
 };  
 }  
</script>  
</body>  
</html>

**实验作业#8:** 编请参考课本第六章内容所学，写一个用户登录程序(具有密码验证功能)，向远程服务器登录。其要求如下:

界面布局设计中，分别设置输入用户名和密码的编辑框，再设置一个“提交”按钮和显示服务器端程序返回的“登录成功”的文本标签与图案，若失败也需响应显示“登录失败”的文本标签与图案。注意其登录验证的程序须写在服务器端程序中。(服务器IP使用本机IP即可)。

请将设计好之用户使用之GUI页面截图，并提供用户GUI页面的xml档案内容与相关java编程(手机端与服务器端程序)。此外执行结果也须提供。

**实验结果:**

图形用户界面, 文本, 应用程序, 聊天或短信

AI 生成的内容可能不正确。文本

AI 生成的内容可能不正确。

图形用户界面, 文本, 应用程序, 聊天或短信

AI 生成的内容可能不正确。文本

AI 生成的内容可能不正确。

**服务器端代码**

**LoginServerApplication.java**

package com.server;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.io.PrintWriter;  
import java.net.ServerSocket;  
import java.net.Socket;  
  
@SpringBootApplication  
public class LoginServerApplication {  
 public static void main(String[] args) throws IOException {  
 ServerSocket serverSocket = new ServerSocket(8080);  
 System.*out*.println("[Server] Running on port 8080...");  
  
 while (true) {  
 try (Socket clientSocket = serverSocket.accept()) {  
 System.*out*.println("[Server] Client connected: "  
 + clientSocket.getInetAddress());  
  
 BufferedReader in = new BufferedReader(  
 new InputStreamReader(clientSocket.getInputStream()));  
 PrintWriter out = new PrintWriter(  
 clientSocket.getOutputStream(), true);  
  
 // 读取客户端数据  
 String inputLine = in.readLine();  
 System.*out*.println("[Server] Received: " + inputLine);  
  
 // 验证逻辑  
 String[] credentials = inputLine.split(":");  
 boolean isValid = "admin".equals(credentials[0])  
 && "123456".equals(credentials[1]);  
  
 // 发送响应  
 String response = isValid ? "SUCCESS" : "FAIL";  
 out.println(response);  
 System.*out*.println("[Server] Sent: " + response);  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 private static boolean validate(String username, String password) {  
 // 示例验证逻辑  
 return "admin".equals(username) && "123456".equals(password);  
 }  
}

**客户端代码**

**MainActivity.java**

package com.example.work8;  
  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.\*;  
import androidx.appcompat.app.AppCompatActivity;  
  
import org.json.JSONException;  
import org.json.JSONObject;  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.OutputStream;  
import java.io.PrintWriter;  
import java.net.HttpURLConnection;  
import java.net.MalformedURLException;  
import java.net.Socket;  
import java.net.SocketTimeoutException;  
import java.net.URL;  
import java.net.URLEncoder;  
import java.nio.charset.StandardCharsets;  
  
public class MainActivity extends AppCompatActivity {  
 private EditText etUsername, etPassword;  
 private Button btnSubmit;  
 private LinearLayout layoutSuccess, layoutFail;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 etUsername = findViewById(R.id.*etUsername*);  
 etPassword = findViewById(R.id.*etPassword*);  
 btnSubmit = findViewById(R.id.*btnSubmit*);  
 layoutSuccess = findViewById(R.id.*layoutSuccess*);  
 layoutFail = findViewById(R.id.*layoutFail*);  
  
 btnSubmit.setOnClickListener(v -> new LoginTask().execute());  
 }  
  
 private class LoginTask extends AsyncTask<Void, Void, String> {  
 @Override  
 protected String doInBackground(Void... voids) {  
 try (Socket socket = new Socket("10.0.2.2", 8080)) { // Android模拟器本机地址  
 // 发送凭证  
 OutputStream os = socket.getOutputStream();  
 PrintWriter pw = new PrintWriter(os);  
 pw.println(etUsername.getText() + ":" + etPassword.getText());  
 pw.flush();  
  
 // 获取响应  
 InputStream is = socket.getInputStream();  
 BufferedReader br = new BufferedReader(new InputStreamReader(is));  
 return br.readLine();  
 } catch (Exception e) {  
 return "ERROR: " + e.getMessage(); // 携带错误信息  
 }  
 }  
  
 @Override  
 protected void onPostExecute(String result) {  
 runOnUiThread(() -> {  
 if(result.equals("SUCCESS")) {  
 layoutSuccess.setVisibility(View.*VISIBLE*);  
 layoutFail.setVisibility(View.*GONE*);  
 } else if(result.equals("FAIL")) {  
 layoutSuccess.setVisibility(View.*GONE*);  
 layoutFail.setVisibility(View.*VISIBLE*);  
 } else {  
 // 显示网络错误  
 Toast.*makeText*(MainActivity.this,  
 "服务器连接失败: "+result,  
 Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }  
 }  
}

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="20dp">  
  
 <EditText  
 android:id="@+id/etUsername"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="用户名"/>  
  
 <EditText  
 android:id="@+id/etPassword"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="密码"  
 android:inputType="textPassword"/>  
  
 <Button  
 android:id="@+id/btnSubmit"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="提交"/>  
  
 <LinearLayout  
 android:id="@+id/layoutSuccess"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:visibility="gone">  
  
 <ImageView  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:src="@drawable/ic\_success"/>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="登录成功"  
 android:textColor="#00FF00"/>  
 </LinearLayout>  
  
 <LinearLayout  
 android:id="@+id/layoutFail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:visibility="gone">  
  
 <ImageView  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:src="@drawable/ic\_failure"/>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="登录失败"  
 android:textColor="#FF0000"/>  
 </LinearLayout>  
  
</LinearLayout>

**Ic\_failure.xml**

<vector xmlns:android="http://schemas.android.com/apk/res/android"  
 android:width="24dp"  
 android:height="24dp"  
 android:viewportWidth="24"  
 android:viewportHeight="24">  
 <path  
 android:fillColor="#F44336"  
 android:pathData="M19,6.41L17.59,5 12,10.59 6.41,5 5,6.41 10.59,12 5,17.59 6.41,19 12,13.41 17.59,19 19,17.59 13.41,12 19,6.41z"/>  
</vector>

**Ic\_success.xml**

<vector xmlns:android="http://schemas.android.com/apk/res/android"  
 android:width="24dp"  
 android:height="24dp"  
 android:viewportWidth="24"  
 android:viewportHeight="24">  
 <path  
 android:fillColor="#4CAF50"  
 android:pathData="M21,7L9,19l-5.5-5.5 1.41-1.41L9,16.17 19.59,5.59 21,7z"/>  
</vector>

**实验作业#9:** 编请参考课本第七章内容所学，编写一个通过列表组件ListView显示Json数组数据的程序。其数据可用JSON数组表达：

[{"sid":1001, "name":"张大山"}, {"sid":1002, "name":"李小丽"} ];

|  |  |
| --- | --- |
| 画面设计 | 执行结果 |
|  |  |

**实验结果:**

电脑萤幕的截图

AI 生成的内容可能不正确。图形用户界面, 应用程序

AI 生成的内容可能不正确。

**MainActivity.java**

package com.example.work9;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Parcel;  
import android.os.Parcelable;  
import android.util.Log;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.ListView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class MainActivity extends AppCompatActivity {  
 private ListView listView;  
 private Button btnParse;  
 private int selectedPosition = -1;  
  
 private static final String *JSON\_DATA* = "["  
 + "{\"sid\":1001,\"name\":\"张大山\",\"age\":21},"  
 + "{\"sid\":1002,\"name\":\"李小丽\",\"age\":22},"  
 + "{\"sid\":1003,\"name\":\"王强\",\"age\":23}"  
 + "]";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 listView = findViewById(R.id.*listView*);  
 btnParse = findViewById(R.id.*btnParse*);  
  
 // 解析原始JSON数据  
 List<Student> students = parseJsonData(*JSON\_DATA*);  
  
 // 设置列表适配器  
 ArrayAdapter<Student> adapter = new ArrayAdapter<Student>(  
 this,  
 android.R.layout.*simple\_list\_item\_activated\_1*,  
 students  
 ) {  
 @NonNull  
 @Override  
 public View getView(int position, View convertView, @NonNull ViewGroup parent) {  
 TextView textView = (TextView) super.getView(position, convertView, parent);  
 Student student = getItem(position);  
 textView.setText(student.name + " - 学号：" + student.sid);  
 return textView;  
 }  
 };  
  
 listView.setAdapter(adapter);  
  
 // 列表选择监听  
 listView.setOnItemClickListener((parent, view, position, id) -> {  
 selectedPosition = position;  
 view.setSelected(true);  
 });  
  
 // 解析按钮点击事件  
 btnParse.setOnClickListener(v -> {  
 if (selectedPosition != -1) {  
 Student selected = (Student) listView.getItemAtPosition(selectedPosition);  
 Intent intent = new Intent(MainActivity.this, DetailActivity.class);  
 intent.putExtra("student", selected);  
 startActivity(intent);  
 overridePendingTransition(R.drawable.*slide\_in\_right*, R.drawable.*slide\_out\_left*);  
 } else {  
 Toast.*makeText*(this, "请先选择要解析的数据", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
  
 private List<Student> parseJsonData(String json) {  
 List<Student> students = new ArrayList<>();  
 try {  
 JSONArray jsonArray = new JSONArray(json);  
 for (int i = 0; i < jsonArray.length(); i++) {  
 JSONObject obj = jsonArray.getJSONObject(i);  
  
 // 关键点：严格匹配字段名称  
 Student student = new Student(  
 obj.getInt("sid"), // 必须与JSON字段名一致  
 obj.getString("name"), // 注意大小写  
 obj.getInt("age") // 新增字段  
 );  
  
 // 调试日志  
 Log.*d*("JSON\_PARSE", "解析结果: " + student);  
 students.add(student);  
 }  
 } catch (JSONException e) {  
 Log.*e*("JSON\_ERROR", "解析失败: " + e.getMessage());  
 }  
 return students;  
 }  
  
 // 实现Parcelable的数据类  
 public static class Student implements Parcelable {  
 public final int sid;  
 public final String name;  
 public final int age;  
  
 protected Student(Parcel in) {  
 sid = in.readInt();  
 name = in.readString();  
 age = in.readInt();  
 }  
  
 public static final Creator<Student> *CREATOR* = new Creator<Student>() {  
 @Override  
 public Student createFromParcel(Parcel in) {  
 return new Student(in);  
 }  
  
 @Override  
 public Student[] newArray(int size) {  
 return new Student[size];  
 }  
 };  
  
 public Student(int sid, String name, int age) {  
 this.sid = sid;  
 this.name = name;  
 this.age = age;  
 }  
  
 @Override  
 public int describeContents() {  
 return 0;  
 }  
  
 @Override  
 public void writeToParcel(Parcel dest, int flags) {  
 dest.writeInt(sid);  
 dest.writeString(name);  
 dest.writeInt(age);  
 }  
 }  
}

**DetailActivity.java**

package com.example.work9;  
  
import android.os.Bundle;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class DetailActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_detail*);  
  
 MainActivity.Student student = getIntent().getParcelableExtra("student");  
  
 ListView resultList = findViewById(R.id.*resultList*);  
 ArrayAdapter<String> adapter = new ArrayAdapter<String>(  
 this,  
 android.R.layout.*simple\_list\_item\_1*,  
 new String[]{  
 "姓名：" + student.name,  
 "学号：" + student.sid,  
 "年龄：" + student.age  
 }  
 );  
  
 resultList.setAdapter(adapter);  
 }  
}

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <!-- 标题栏 -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="#2196F3"  
 android:orientation="horizontal"  
 android:paddingHorizontal="16dp">  
  
 <TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="t2\_12"  
 android:textColor="#FFFFFF"  
 android:textSize="20sp"/>  
  
 <Button  
 android:id="@+id/btnParse"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:background="?attr/selectableItemBackground"  
 android:text="解析数据"  
 android:textAllCaps="false"  
 android:textColor="#FFFFFF"/>  
  
 </LinearLayout>  
  
 <!-- 数据列表 -->  
 <ListView  
 android:id="@+id/listView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:choiceMode="singleChoice"  
 android:divider="#CCCCCC"  
 android:dividerHeight="1dp"/>  
  
</LinearLayout>

**Activity\_detail.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <!-- 标题栏 -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="#2196F3"  
 android:gravity="center\_vertical"  
 android:paddingHorizontal="16dp">  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="解析结果"  
 android:textColor="#FFFFFF"  
 android:textSize="20sp"/>  
  
 </LinearLayout>  
  
 <!-- 解析结果列表 -->  
 <ListView  
 android:id="@+id/resultList"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:divider="#CCCCCC"  
 android:dividerHeight="1dp"/>  
  
</LinearLayout>

**实验作业#10:** 编请参考课本第八章SQLite数据库内容所学，写一个用户个人通讯簿APP，其要求如下:

需要能记录: 用户的姓名、电话、住址、微信号、Email。并能浏览所有已记入数据。

请将设计好之用户使用之GUI页面截图，并提供用户GUI页面的xml档案内容与相关java编程。此外执行结果也须提供。

**实验结果:**

电脑萤幕的截图

AI 生成的内容可能不正确。 图形用户界面, 文本, 应用程序, 聊天或短信

AI 生成的内容可能不正确。手机屏幕的截图

AI 生成的内容可能不正确。

**MianActivity.java**

package com.example.work10;  
  
import android.content.Intent;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
  
public class MainActivity extends AppCompatActivity {  
 private RecyclerView recyclerView;  
 private DatabaseHelper dbHelper;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 dbHelper = new DatabaseHelper(this);  
 recyclerView = findViewById(R.id.*recyclerView*);  
 recyclerView.setLayoutManager(new LinearLayoutManager(this));  
  
 FloatingActionButton fab = findViewById(R.id.*fab*);  
 fab.setOnClickListener(view -> {  
 Intent intent = new Intent(MainActivity.this, AddContactActivity.class);  
 startActivity(intent);  
 });  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
 updateContactList();  
 }  
  
 private void updateContactList() {  
 SQLiteDatabase db = dbHelper.getReadableDatabase();  
 Cursor cursor = db.query("contacts", null, null, null, null, null, null);  
 ContactAdapter adapter = new ContactAdapter(cursor);  
 recyclerView.setAdapter(adapter);  
 }  
}

**DatabaseHelper.java**

package com.example.work10;  
  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
public class DatabaseHelper extends SQLiteOpenHelper {  
 private static final String *DATABASE\_NAME* = "contacts.db";  
 private static final int *DATABASE\_VERSION* = 1;  
  
 public DatabaseHelper(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 db.execSQL("CREATE TABLE contacts (" +  
 "\_id INTEGER PRIMARY KEY AUTOINCREMENT," +  
 "name TEXT," +  
 "phone TEXT," +  
 "address TEXT," +  
 "wechat TEXT," +  
 "email TEXT);");  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 db.execSQL("DROP TABLE IF EXISTS contacts");  
 onCreate(db);  
 }  
}

**ContactAdapter.java**

package com.example.work10;  
  
import android.database.Cursor;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import androidx.recyclerview.widget.RecyclerView;  
  
public class ContactAdapter extends RecyclerView.Adapter<ContactAdapter.ViewHolder> {  
 private Cursor cursor;  
  
 public ContactAdapter(Cursor cursor) {  
 this.cursor = cursor;  
 }  
  
 public static class ViewHolder extends RecyclerView.ViewHolder {  
 TextView tvName, tvPhone, tvAddress, tvWechat, tvEmail;  
  
 public ViewHolder(View itemView) {  
 super(itemView);  
 tvName = itemView.findViewById(R.id.*tvName*);  
 tvPhone = itemView.findViewById(R.id.*tvPhone*);  
 tvAddress = itemView.findViewById(R.id.*tvAddress*);  
 tvWechat = itemView.findViewById(R.id.*tvWechat*);  
 tvEmail = itemView.findViewById(R.id.*tvEmail*);  
 }  
 }  
  
 @Override  
 public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*item\_contact*, parent, false);  
 return new ViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(ViewHolder holder, int position) {  
 if (!cursor.moveToPosition(position)) return;  
  
 holder.tvName.setText(cursor.getString(cursor.getColumnIndex("name")));  
 holder.tvPhone.setText(cursor.getString(cursor.getColumnIndex("phone")));  
 holder.tvAddress.setText(cursor.getString(cursor.getColumnIndex("address")));  
 holder.tvWechat.setText(cursor.getString(cursor.getColumnIndex("wechat")));  
 holder.tvEmail.setText(cursor.getString(cursor.getColumnIndex("email")));  
 }  
  
 @Override  
 public int getItemCount() {  
 return cursor.getCount();  
 }  
}

**AddContactActivity.java**

package com.example.work10;  
  
import android.content.ContentValues;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.EditText;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class AddContactActivity extends AppCompatActivity {  
 private EditText etName, etPhone, etAddress, etWechat, etEmail;  
 private DatabaseHelper dbHelper;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_add\_contact*);  
  
 dbHelper = new DatabaseHelper(this);  
 initializeViews();  
  
 Button btnSave = findViewById(R.id.*btnSave*);  
 btnSave.setOnClickListener(v -> saveContact());  
 }  
  
 private void initializeViews() {  
 etName = findViewById(R.id.*etName*);  
 etPhone = findViewById(R.id.*etPhone*);  
 etAddress = findViewById(R.id.*etAddress*);  
 etWechat = findViewById(R.id.*etWechat*);  
 etEmail = findViewById(R.id.*etEmail*);  
 }  
  
 private void saveContact() {  
 SQLiteDatabase db = dbHelper.getWritableDatabase();  
 ContentValues values = new ContentValues();  
  
 values.put("name", etName.getText().toString());  
 values.put("phone", etPhone.getText().toString());  
 values.put("address", etAddress.getText().toString());  
 values.put("wechat", etWechat.getText().toString());  
 values.put("email", etEmail.getText().toString());  
  
 db.insert("contacts", null, values);  
 finish();  
 }  
}

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"/>  
  
 <com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:id="@+id/fab"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_margin="16dp"  
 app:srcCompat="@android:drawable/ic\_menu\_add"/>  
</RelativeLayout>

**Activity\_add\_contact.xml**

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <!-- 姓名 -->  
 <EditText  
 android:id="@+id/etName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="姓名"  
 android:inputType="textPersonName"/>  
  
 <!-- 电话 -->  
 <EditText  
 android:id="@+id/etPhone"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:hint="电话"  
 android:inputType="phone"/>  
  
 <!-- 地址 -->  
 <EditText  
 android:id="@+id/etAddress"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:hint="地址"  
 android:inputType="textPostalAddress"/>  
  
 <!-- 微信 -->  
 <EditText  
 android:id="@+id/etWechat"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:hint="微信"  
 android:inputType="text"/>  
  
 <!-- Email -->  
 <EditText  
 android:id="@+id/etEmail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:hint="Email"  
 android:inputType="textEmailAddress"/>  
  
 <!-- 保存按钮 -->  
 <Button  
 android:id="@+id/btnSave"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="24dp"  
 android:text="保存联系人"/>  
 </LinearLayout>  
</ScrollView>

**Item\_contact.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <TextView  
 android:id="@+id/tvName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="18sp"  
 android:textStyle="bold"/>  
  
 <TextView  
 android:id="@+id/tvPhone"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="#666"/>  
  
 <TextView  
 android:id="@+id/tvAddress"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="#666"/>  
  
 <TextView  
 android:id="@+id/tvWechat"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="#666"/>  
  
 <TextView  
 android:id="@+id/tvEmail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="#666"/>  
</LinearLayout>