**实 验 二**

**课程名称** 移动应用系统

**实验项目** 移动平台数据持久化

**实验仪器** 台式电脑

**系 别** 计算机科学与技术系

**专 业** 计算机科学与技术

**班级/学号** 计科1905 / 2019010132

**学生姓名** 陈浩

**实验日期** 2021/11/23

**成 绩**

**指导教师** 陈雷

实验二 移动平台数据持久化

1. 实验目的：

1、通过本实验，进一步熟悉Android Studio的用法

2、通过本实验，掌握Android系统中适配器视图（ListView）的使用方法

3、通过本实验，掌握Android系统中数据持久化的方法，包括使用文件、SharedPreferences以及SQLite实现数据持久化的方法。

1. 实验内容及要求：

【注：可以自己设计题目，但需要经过指导教师同意】

【基本要求】设计并实现一个个人日记本程序，具体要求如下【70分】：

1）支持新增、修改和删除日记。日记为纯文本格式，日记的内容存储在SQLite数据库中，数据库表应该至少包含日记标题，日记内容，日记的创建时间，日记的作者等信息。

2）以ListView的形式显示当前用户的日记列表，点击某一条记录，可以打开一个新的界面供用户查看、编辑日记。

【扩展要求】在基本要求的基础上，增加下列功能【30分】：

1）设计用户界面，维护日记的作者信息（至少包含作者的姓名），并存入SharedPreferences中。当应用程序启动时读取SharedPreferences中的作者信息，若不存在作者信息，则设置一个默认的作者姓名，用于新建日记的“作者”字段的值（20分）

2）允许用户添加照片到日记中，方法是允许用户通过拍照或读取系统相册的方式添加照片，并将选中的照片拷贝到日记本应用程序的内部存储目录下，将照片文件名保存入数据库中与对应的日记记录关联在一起。用户查看日记时也显示日记中包含的照片（10分）。

【评分标准】：

1）实现了基本要求及扩展要求中所要求的程序功能，将得到对应的分数

2）实现功能的同时，应该尽可能保证程序的稳定性、易用性，出现下列情形，将扣去对应的分数

（1）程序崩溃/闪退，扣10分

（2）程序缺少必要的文字提示，扣5分

3）编写程序的过程中，可以参考网上的资料，但必须能够看懂程序代码并回答老师提出的问题，如果无法解释清楚程序代码，扣15分。

4）雷同的实验报告，提交时间较晚的同学成绩为0

1. 程序代码及运行结果：

注：提供完整的程序代码以及运行结果截图

程序代码：

1. MainActivity.Java

*package* com.example.diary;  
  
*import* android.content.Intent;  
*import* android.database.Cursor;  
*import* android.database.sqlite.SQLiteDatabase;  
  
*import* androidx.swiperefreshlayout.widget.SwipeRefreshLayout;  
*import* androidx.appcompat.app.AppCompatActivity;  
*import* android.os.Bundle;  
  
*import* android.view.View;  
*import* android.widget.AdapterView;  
*import* android.widget.ArrayAdapter;  
*import* android.widget.Button;  
*import* android.widget.ListView;  
*import* android.widget.Toast;  
  
*import* java.util.ArrayList;  
*import* java.util.List;  
  
*import static* com.example.diary.DatabaseHelper.DB\_NAME;  
*import static* com.example.diary.DatabaseHelper.TABLE\_NAME;  
*import static* com.example.diary.DatabaseHelper.VERSION;  
  
*public class* MainActivity *extends* AppCompatActivity {  
  
 *public static* DatabaseHelper dbHelper;  
 *private* SQLiteDatabase db;  
 *// 日记标题列表  
 private* List<String> diary = *new* ArrayList<>();  
 *// 插入 (1) 和 更新 (0)  
 public static final int* TAG\_INSERT = 1;  
 *public static final int* TAG\_UPDATE = 0;  
 *// 点中所选项 (标题)  
 private* String select\_item;  
 *// 日记id  
 private int* Id;  
 *// 以Listview排序* ListView listView;  
 *// 数组适配器：绑定单一格式数据 (此处未Title)* ArrayAdapter<String> adapter;  
 *// 下拉刷新  
 private* SwipeRefreshLayout swipeRefresh;  
  
 *// 获取数据库  
 public static* DatabaseHelper getDbHelper(){  
 *return* dbHelper;  
 }  
  
 *@Override  
 protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 *// 获取主活动各类组件* Button add = findViewById(R.id.add);  
 *// 监听 swipe\_refresh 设置下拉即刷新页面* swipeRefresh = findViewById(R.id.swipe\_refresh);  
 swipeRefresh.setOnRefreshListener(*new* SwipeRefreshLayout  
 .OnRefreshListener() {  
 *@Override  
 public void* onRefresh() {  
 refresh();  
 Toast.makeText(MainActivity.*this*,"Refresh Successfully!", Toast.LENGTH\_SHORT).show();  
 }  
 });  
  
 *// 创建数据库* dbHelper = *new* DatabaseHelper(MainActivity.*this*, DB\_NAME, *null*, VERSION);  
 dbHelper.getWritableDatabase();  
 init();  
  
 *// 设置日记编辑监听 (用于日记的创建)* add.setOnClickListener(*new* View.OnClickListener() {  
 *@Override  
 public void* onClick(View v) {  
 *// 到detail类中进行插入* Intent intent = *new* Intent(MainActivity.*this*, Detail.*class*);  
 intent.putExtra("TAG", TAG\_INSERT);  
 startActivity(intent);  
 }  
 });  
  
 *// 设置日记更新监听 (用于日记的更新)* listView.setOnItemClickListener(*new* AdapterView.OnItemClickListener() {  
 *@Override  
 public void* onItemClick(AdapterView<?> parent, View view,  
 *int* position, *long* id) {  
 *// 到detail类中进行插入* Intent intent = *new* Intent(MainActivity.*this*, Detail.*class*);  
 Id = getDiaryId(position);  
 *// 获取需要更新日记的id* intent.putExtra("ID", Id);  
 intent.putExtra("TAG", TAG\_UPDATE);  
 startActivity(intent);  
 }  
 });  
 }  
  
 *private void* init(){  
 db = dbHelper.getWritableDatabase();  
 diary.clear();  
 *// 查询数据库，先将title一列添加到列表项目中  
 // 为了接下来要在主页面用Listview显示所有的日记标题* Cursor cursor = db.query(TABLE\_NAME,*null*,*null*,*null*,*null*,*null*,*null*);  
 *if*(cursor.moveToFirst()){  
 String diary\_item;  
 *do*{  
 diary\_item = cursor.getString(cursor.getColumnIndex("title"));  
 diary.add(diary\_item);  
 }*while*(cursor.moveToNext());  
 }  
 cursor.close();  
 *// 用于显示Listview的内容* adapter = *new* ArrayAdapter<String>(  
 MainActivity.*this*, android.R.layout.simple\_list\_item\_1, diary);  
 listView = findViewById(R.id.list\_item);  
 listView.setAdapter(adapter);  
 }  
  
 *// 自动刷新列表  
 private void* refresh(){  
 *new* Thread(*new* Runnable(){  
 *@Override  
 public void* run() {  
 *try*{  
 Thread.sleep(1000);  
 }*catch*(InterruptedException e){  
 e.printStackTrace();  
 }  
 runOnUiThread(*new* Runnable() {  
 *@Override  
 public void* run() {  
 init();  
 swipeRefresh.setRefreshing(*false*);  
 }  
 });  
 }  
 }).start();  
 }  
  
 *// 获取当前点击的日记id  
 private int* getDiaryId(*int* position){  
 *int* Id;  
 select\_item = diary.get(position);  
 *// 获取数据库数据* db = dbHelper.getWritableDatabase();  
 *// 查表 (目标为当前标题)  
 try* (Cursor cursor = db.query(TABLE\_NAME, *new* String[]{"id"}, "title=?",  
 *new* String[]{select\_item}, *null*, *null*, *null*)) {  
 cursor.moveToFirst();  
 *// 获取id* Id = cursor.getInt(cursor.getColumnIndex("id"));  
 }  
 *return* Id;  
 }  
}

1. Login.Java

*package* com.example.diary;  
  
*import* android.content.Intent;  
*import* android.content.SharedPreferences;  
*import* android.graphics.Color;  
*import* android.os.Build;  
*import* android.preference.PreferenceManager;  
*import* androidx.appcompat.app.AppCompatActivity;  
*import* android.os.Bundle;  
*import* android.text.method.HideReturnsTransformationMethod;  
*import* android.text.method.PasswordTransformationMethod;  
*import* android.view.View;  
*import* android.widget.Button;  
*import* android.widget.CheckBox;  
*import* android.widget.EditText;  
*import* android.widget.Toast;  
  
*public class* Login *extends* AppCompatActivity{  
  
 *// 定义一个SharedPreferences对象 (主要用户存储用户名和密码)  
 private* SharedPreferences pref;  
 *// 调用SharedPreferences对象的edit()方法来获取一个SharedPreferences.Editor对象  
 // 用以添加要保存的数据  
 private* SharedPreferences.Editor editor;  
 *private* EditText adminEdit; *// 用户名输入框  
 private* EditText passwordEdit; *// 密码输入框  
 private* Button login; *// 登陆选项  
 private* CheckBox savePassword; *// 保存密码复选框  
 private* CheckBox showPassword; *// 显示或隐藏密码复选框* @Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_login);  
 *// 获取各组件或对象的实例* pref = PreferenceManager.getDefaultSharedPreferences(*this*);  
 *// 登录按钮 用户名 密码 是否显示和记住密码* login = findViewById(R.id.login\_button);  
 adminEdit = findViewById(R.id.admin);  
 passwordEdit = findViewById(R.id.password);  
 savePassword = findViewById(R.id.save\_password);  
 showPassword = findViewById(R.id.show\_password);  
  
 *// 默认为不记住密码  
 final boolean* isSave = pref.getBoolean("save\_password", *false*);  
 *// 当 "Rem psw" 勾选时, 从SharedPreferences对象中读出保存的内容, 并显示出来  
 if*(isSave){  
 String account = pref.getString("account", "");  
 String password = pref.getString("password", "");  
 adminEdit.setText(account);  
 passwordEdit.setText(password);  
 *//把光标移到文本末尾处* adminEdit.setSelection(adminEdit.getText().length());  
 passwordEdit.setSelection(passwordEdit.getText().length());  
 savePassword.setChecked(*true*);  
 }  
  
 *// 用户点击登录时的处理事件* login.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 *// 获取输入的用户名和密码* String account = adminEdit.getText().toString();  
 String password = passwordEdit.getText().toString();  
 *// 用户名和密码正确  
 if*(account.equals("Misaki")&&password.equals("20011104")){  
 *// 将密码存入pref用于显示密码* editor = pref.edit();  
 *// "Rem psw" 勾选  
 if*(savePassword.isChecked()){  
 editor.putBoolean("save\_password",*true*);  
 editor.putString("account",account);  
 editor.putString("password",password);  
 }  
 *// 否则pref不存储密码 下次还需要自行输入  
 else*{  
 editor.clear();  
 }  
 *// 提交进行数据存储* editor.apply();  
  
 *// Toast显示登入成功* Toast.makeText(Login.*this*,"Root!",Toast.LENGTH\_SHORT).show();  
 *// 启动活动* Intent intent = *new* Intent(Login.*this*, MainActivity.*class*);  
 startActivity(intent);  
 *// 活动结束* finish();  
 }  
 *// 若用户名或密码错误  
 // 则Toast显示错误, 请重新输入  
 else*{  
 Toast.makeText(Login.*this*,"Error! Plz sign again!",Toast.LENGTH\_SHORT).show();  
 }  
 }  
 });  
  
 *// 用户点击 "show psw" 复选框* showPassword.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 *// 点击复选框即显示密码* showOrhidePassword(passwordEdit, showPassword.isChecked());  
 }  
 });  
  
 }  
  
 *// 当用户退出界面时, 检测是否勾选记住密码  
 // 若勾选则保存用户输入的用户名及密码* @Override  
 *protected void* onDestroy() {  
 *super*.onDestroy();  
 editor = pref.edit();  
 String account = adminEdit.getText().toString();  
 String password = passwordEdit.getText().toString();  
  
 *// 点击了保存密码则存储账号和密码  
 if*(savePassword.isChecked()){  
 editor.putBoolean("save\_password", *true*);  
 editor.putString("account", account);  
 editor.putString("password", password);  
 }  
 *// 未点击则清除保存  
 else*{  
 editor.clear();  
 }  
 editor.apply();  
 }  
  
 *// 显示或隐藏密码  
 private void* showOrhidePassword(EditText passwordEdit, *boolean* isShow){  
 *// 需要记住光标开始的位置  
 int* pos = passwordEdit.getSelectionStart();  
 *if*(isShow) {  
 passwordEdit.setTransformationMethod(HideReturnsTransformationMethod.getInstance());  
 }  
 *else* {  
 passwordEdit.setTransformationMethod(PasswordTransformationMethod.getInstance());  
 }  
 passwordEdit.setSelection(pos);  
 }  
  
}

1. Detail

*package* com.example.diary;  
  
*import* android.Manifest;  
*import* android.annotation.TargetApi;  
*import* android.content.ContentUris;  
*import* android.content.ContentValues;  
*import* android.content.Intent;  
*import* android.content.SharedPreferences;  
*import* android.content.pm.PackageManager;  
*import* android.database.Cursor;  
*import* android.database.sqlite.SQLiteDatabase;  
  
*import* androidx.appcompat.app.AppCompatActivity;  
*import* androidx.appcompat.widget.Toolbar;  
*import* androidx.core.app.ActivityCompat;  
*import* androidx.core.content.ContextCompat;  
  
*import* android.graphics.Bitmap;  
*import* android.graphics.BitmapFactory;  
*import* android.graphics.drawable.BitmapDrawable;  
*import* android.net.Uri;  
*import* android.os.Build;  
*import* android.os.Bundle;  
  
*import* android.preference.PreferenceManager;  
*import* android.provider.DocumentsContract;  
*import* android.provider.MediaStore;  
*import* android.view.Menu;  
*import* android.view.MenuItem;  
  
*import* android.view.View;  
*import* android.widget.Button;  
*import* android.widget.EditText;  
*import* android.widget.ImageView;  
*import* android.widget.TextView;  
*import* android.widget.Toast;  
  
*import* java.io.ByteArrayOutputStream;  
*import* java.sql.Date;  
*import* java.text.SimpleDateFormat;  
  
  
*import static* com.example.diary.DatabaseHelper.TABLE\_NAME;  
*import static* com.example.diary.MainActivity.TAG\_INSERT;  
*import static* com.example.diary.MainActivity.TAG\_UPDATE;  
*import static* com.example.diary.MainActivity.dbHelper;  
*import static* com.example.diary.MainActivity.getDbHelper;  
  
*public class* Detail *extends* AppCompatActivity {  
  
 *private* SQLiteDatabase db; *// 数据库* EditText title; *// 标题* EditText author; *// 作者* TextView time; *// 时间* EditText content; *// 内容* Button pictureChoice; *// 从相册选择照片* ImageView picture; *// 选择的照片  
  
 private* SharedPreferences pref; *// 临时存放区  
 private int* tag; *// tag 用来区分数据库的插入和更新操作  
 private int* id; *// 日记id  
 private* Toolbar toolbar; *// 自建标题栏  
 private static final int* CHOICE\_PHOTO = 2; *// 主活动2* @Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 *// 加载活动和组件* setContentView(R.layout.activity\_detail);  
 toolbar = findViewById(R.id.toolbar); *// Toolbar加载* setSupportActionBar(toolbar);  
 pref = PreferenceManager.getDefaultSharedPreferences(*this*);  
 title = findViewById(R.id.detail\_title); *// 标题* author = findViewById(R.id.detail\_author); *// 作者* time = findViewById(R.id.detail\_time); *// 保存时间* content = findViewById(R.id.detail\_content); *// 内容* pictureChoice =findViewById(R.id.detail\_pictureChoice); *// 照片选择按钮* picture = findViewById(R.id.detail\_picture); *// 所选照片  
  
 // 默认作者名* editor = pref.edit();  
 editor.putString("author", "Misaki");  
 editor.apply();  
 String temp\_author = pref.getString("author", "");  
 author.setText(temp\_author);

*// 默认时间格式 (系统当前时间)* SimpleDateFormat simpleDateFormat = *new* SimpleDateFormat("yyyy年MM月dd日 HH:mm:ss");  
 Date date = *new* Date(System.currentTimeMillis());  
 time.setText(simpleDateFormat.format(date));  
  
 *// 点击选择照片监听* pictureChoice.setOnClickListener(*new* View.OnClickListener() {  
 *@Override  
 public void* onClick(View v) {  
 *// 检查是否具有开启相册的权限  
 if* (ContextCompat.checkSelfPermission(Detail.*this*, Manifest.permission.WRITE\_EXTERNAL\_STORAGE)  
 != PackageManager.PERMISSION\_GRANTED) {  
 ActivityCompat.requestPermissions(Detail.*this*, *new* String[]{Manifest.permission.WRITE\_EXTERNAL\_STORAGE}, 1);  
 }  
 *else* {  
 openAlbum();  
 }  
 }  
 });  
  
 *// 将数据写进数据库* db = dbHelper.getWritableDatabase();  
 Intent intent = getIntent();  
 tag = intent.getIntExtra("TAG",-1);  
 *switch*(tag){  
 *// TAG == 1  
 case* TAG\_INSERT:  
 *break*;  
 *// TAG == 0  
 case* TAG\_UPDATE:  
 *// 获取新id* id = intent.getIntExtra("ID",-1);  
 *// 在原数据库中以每行为代价搜索 id == 新id 的行  
 try* (Cursor cursor = db.query(TABLE\_NAME, *null*, "id=?",  
 *new* String[]{String.valueOf(id)}, *null*, *null*, *null*)) {  
 *// 进行从数据库行头插入  
 if* (cursor.moveToFirst()) {  
 String select\_title = cursor.getString(cursor.getColumnIndex("title"));  
 String select\_author = cursor.getString(cursor.getColumnIndex("author"));  
 String select\_content = cursor.getString(cursor.getColumnIndex("content"));  
 title.setText(select\_title);  
 author.setText(select\_author);  
 content.setText(select\_content);  
 *// 特殊：以字节数组存储 Blob类型 的照片文件  
 byte*[] in = cursor.getBlob(cursor.getColumnIndex("picture"));  
 Bitmap bitmap = BitmapFactory.decodeByteArray(in, 0, in.length);  
 *// 以大图形式保存 (部分图片可能会溢出)* picture.setImageBitmap(bitmap);  
  
 }  
 }  
 *break*;  
 *default*:  
 *break*;  
 }  
 }  
  
 *// 打开系统相册  
 private void* openAlbum(){  
 Intent intent = *new* Intent("android.intent.action.GET\_CONTENT");  
 intent.setType("image/\*");  
 *// 第2个主活动 标明与MainActivity同等关系* startActivityForResult(intent, CHOICE\_PHOTO);  
 }  
  
 *// 判断是否有权限打开系统相册 (需要在Manifest中赋予相关权限)* @Override  
 *public void* onRequestPermissionsResult(*int* requestCode, String[] permissions, *int*[] grantResults){  
 *switch* (requestCode){  
 *case* 1:  
 *if*(grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED){  
 openAlbum();  
 }  
 *else* {  
 Toast.makeText(*this*,"You denied the permission ?",Toast.LENGTH\_SHORT).show();  
 }  
 *break*;  
 *default*:  
 }  
 }  
  
 *// 关于 Android4.4 之前和之后的照片处理* @Override  
 *protected void* onActivityResult(*int* requestCode, *int* resultCode, Intent data) {  
 *super*.onActivityResult(requestCode, resultCode, data);  
 *switch* (requestCode) {  
 *case* CHOICE\_PHOTO:  
 *if* (resultCode == RESULT\_OK) {  
 *// 判断虚拟机版本号 ( ?> v4.4)  
 if* (Build.VERSION.SDK\_INT >= 19) {  
 *// v4.4 及其以上版本采用此方法处理照片* handleImageOnKitKat(data);  
 } *else* {  
 *// v4.4 以下采用此方法处理照片* handleImageBeforeKitKat(data);  
 }  
 }  
 *break*;  
 *default*:  
 *break*;  
 }  
 }  
  
 *// 地址解析: ( >= v4.4)* @TargetApi(19)  
 *private void* handleImageOnKitKat(Intent data){  
 String imagePath = *null*;  
 *// 创建URI* Uri uri = data.getData();  
  
 *if*(DocumentsContract.isDocumentUri(*this*, uri)){  
 String docId = DocumentsContract.getDocumentId(uri);  
 *if*("com.android.providers.media.documents".equals(uri.getAuthority())){  
 String id = docId.split(":")[1];  
 String selection = MediaStore.Images.Media.\_ID + "=" + id;  
 imagePath = getImagePath(MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI, selection);  
 }  
 *else if*("com.android.providers.downloads.documents".equals(uri.getAuthority())) {  
 Uri contentUri = ContentUris.withAppendedId(Uri.parse("content://downloads/public\_downloads"), Long.valueOf(docId));  
 imagePath = getImagePath(contentUri, *null*);  
 }  
 }  
 *else if*("content".equalsIgnoreCase(uri.getScheme())){  
 imagePath = getImagePath(uri,*null*);  
 }  
 *else if*("file".equalsIgnoreCase(uri.getScheme())){  
 imagePath = uri.getPath();  
 }  
 displayImage(imagePath); *// 显示所选中照片* }  
  
 *// 地址解析 ( < v4.4)  
 private void* handleImageBeforeKitKat(Intent data){  
 *// 创建URI* Uri uri = data.getData();  
 String imagePath = getImagePath(uri,*null*);  
 *// 显示所选中照片* displayImage(imagePath);  
 }  
  
 *// 获取图片的路径  
 private* String getImagePath(Uri uri, String selection){  
 String path = *null*;  
 Cursor cursor = getContentResolver().query(uri, *null*, selection,*null*, *null*);  
 *if*(cursor != *null*) {  
 *if* (cursor.moveToFirst()) {  
 path = cursor.getString(cursor.getColumnIndex(MediaStore.Images.Media.DATA));  
 }  
 cursor.close();  
 }  
 *return* path;  
 }  
  
 *// 显示照片  
 private void* displayImage(String imagePath){  
 *if*(imagePath != *null*){  
 Bitmap bitmap = BitmapFactory.decodeFile(imagePath);  
 *// 以大图显示:* picture.setImageBitmap(bitmap);  
 }*else*{  
 Toast.makeText(*this*,"failed to get image", Toast.LENGTH\_SHORT).show();  
 }  
 }  
  
 *// 将menu中的toolbar添加进来 (最上方标题栏)* @Override  
 *public boolean* onCreateOptionsMenu(Menu menu) {  
 toolbar.inflateMenu(R.menu.menu\_main);  
 *return super*.onCreateOptionsMenu(menu);  
 }  
  
 *// 设置 "Save" 、 "Delete" 、 "Return" 按钮点击事件* @Override  
 *public boolean* onOptionsItemSelected(MenuItem item) {  
 *switch*(item.getItemId()){  
 *// 保存日记  
 case* R.id.save:  
 *// 添加日记 (新建)  
 if*(tag == TAG\_INSERT) {  
 *// 将所写内容临时保存到 ContextValues 中 (类似于HashTable: key - values)* ContentValues values = *new* ContentValues();  
 values.put("title", title.getText().toString());  
 values.put("author", author.getText().toString());  
 values.put("content", content.getText().toString());  
 *final* ByteArrayOutputStream os = *new* ByteArrayOutputStream();  
 *// 图片压缩 (默认1 : 1)* Bitmap bitmap = ((BitmapDrawable)picture.getDrawable()).getBitmap();  
 bitmap.compress(Bitmap.CompressFormat.PNG, 100, os);  
 values.put("picture", os.toByteArray());  
 *// 将所有数值插入数据库* db.insert(TABLE\_NAME, *null*, values);  
 *// 清除临时保存的数据* values.clear();  
 Toast.makeText(*this*, "Save!", Toast.LENGTH\_SHORT).show();  
 finish();  
 *break*;  
 }  
 *// 修改日记 (原有)  
 else if*(tag == TAG\_UPDATE){  
 *// 将修改的数据据保存与ContentValues中* ContentValues values = *new* ContentValues();  
 values.put("title", title.getText().toString());  
 values.put("author", author.getText().toString());  
 values.put("content", content.getText().toString());  
 *final* ByteArrayOutputStream os = *new* ByteArrayOutputStream();  
 *// 图片压缩 (默认1 : 1)* Bitmap bitmap = ((BitmapDrawable)picture.getDrawable()).getBitmap();  
 bitmap.compress(Bitmap.CompressFormat.PNG, 100, os);  
 values.put("picture",os.toByteArray());  
 *// 进行更新 (id)* db.update(TABLE\_NAME, values, "id=?", *new* String[]{String.valueOf(id)});  
 finish();  
 *break*;  
 }  
 *// 删除日记  
 case* R.id.delete:  
 *// 根据 id 删除当前选中日记* db.delete(TABLE\_NAME, "id=?", *new* String[]{String.valueOf(id)});  
 Toast.makeText(*this*, "Delete!", Toast.LENGTH\_SHORT).show();  
 finish();  
 *break*;  
 *// 返回页面  
 case* R.id.goBack:  
 finish();  
 Toast.makeText(*this*, "Return!", Toast.LENGTH\_SHORT).show();  
 *break*;  
 *default*:  
 *break*;  
 }  
 *return true*;  
 }  
}

1. DatabaseHelper

*package* com.example.diary;  
  
*import* android.content.Context;  
*import* android.database.sqlite.SQLiteDatabase;  
*import* android.database.sqlite.SQLiteOpenHelper;  
  
*public class* DatabaseHelper *extends* SQLiteOpenHelper{  
  
 *public static final* String DB\_NAME = "Minakami.db";  
 *public static final int* VERSION = 1;  
 *public static final* String TABLE\_NAME = "Diary";  
 *//建表语句  
 public static final* String CREATE\_DIARY = "create table Diary(" +  
 "id integer primary key autoincrement," +  
 "title text," +  
 "time text," +  
 "author text," +  
 "content text," +  
 "picture BLOB)";  
  
 *public* DatabaseHelper(Context context,String name,  
 SQLiteDatabase.CursorFactory factory,*int* version){  
 *super*(context, name, factory, version);  
  
 }  
 *@Override  
 public void* onCreate(SQLiteDatabase db) {  
 db.execSQL(CREATE\_DIARY);  
 }  
  
 *@Override  
 public void* onUpgrade(SQLiteDatabase db, *int* oldVersion, *int* newVersion) {  
  
 }  
}

1. 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:id="@+id/activity\_main"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 *<!--实现下拉自动刷新动画-->* <androidx.swiperefreshlayout.widget.SwipeRefreshLayout  
 android:id="@+id/swipe\_refresh"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 android:orientation="vertical">  
  
 <androidx.appcompat.widget.Toolbar  
 android:background="@color/black"  
 app:title="~Diary!"  
 app:titleTextColor="@color/white"  
 android:id="@+id/toolbar"  
 app:subtitle="by Misaki"  
 app:subtitleTextColor="@color/purple\_200"  
 app:titleMarginStart="95dp"  
 app:navigationIcon="@drawable/ic\_baseline\_ac\_yuki\_24"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"/>  
  
 <ListView  
 android:id="@+id/list\_item"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
 </ListView>  
  
 *<!--分割-->* <View  
 android:layout\_width="match\_parent"  
 android:layout\_height="1dp"  
 android:background="#EEE9E9"/>  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="15dp"  
 android:text=""/>  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:textSize="20sp"  
 android:text="No More!"/>  
 <Button  
 android:id="@+id/add"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 android:text="ADD ONE DIARY"  
 android:textSize="16sp"  
 android:gravity="center"  
 android:layout\_alignParentBottom="true" />  
  
 </LinearLayout>  
 </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>  
  
</RelativeLayout>

1. activity\_login.xml

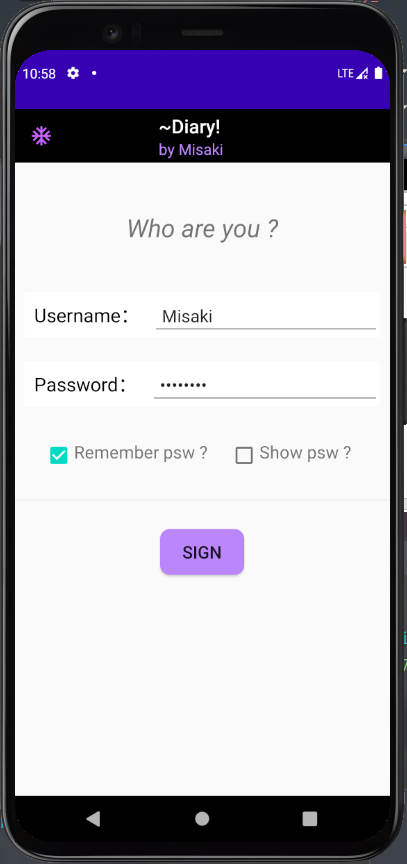
<?xml version="1.0" encoding="utf-8"?>  
*<!--登录界面总体布局为垂直线性布局-->*<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
*<!--页面布局-->*<LinearLayout  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true">  
  
 <androidx.appcompat.widget.Toolbar  
 android:background="@color/black"  
 app:title="~Diary!"  
 app:titleTextColor="@color/white"  
 android:id="@+id/toolbar"  
 app:subtitle="by Misaki"  
 app:subtitleTextColor="@color/purple\_200"  
 app:titleMarginStart="95dp"  
 app:navigationIcon="@drawable/ic\_baseline\_ac\_yuki\_24"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"/>  
  
 *<!--留出一些空白-->* <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:text=""/>  
  
 *<!--登录界面标题-->* <TextView  
 android:id="@+id/login\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_horizontal"  
 android:textSize="26sp"  
 android:textStyle="italic"  
 android:text="Who are you ?"  
 />  
  
 *<!--留出一些空白-->* <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="30dp"  
 android:text=""/>  
  
 *<!--用户名和密码部分采用垂直线性布局-->* <LinearLayout  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 *<!--用户名部分采用线性水平布局-->* <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginBottom="5dp"  
 android:background="#ffffff"  
 android:orientation="horizontal">  
 *<!--用户名-->* <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Username："  
 android:textSize="20sp"  
 android:textColor="#000"  
 android:padding="10dp"/>  
 *<!--用户名输入-->* <EditText  
 android:id="@+id/admin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="3dp"  
 android:hint="Name plz"  
 android:padding="10dp"/>  
 </LinearLayout>  
  
 *<!--密码部分采用线性水平布局-->* <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginEnd="10dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginBottom="5dp"  
 android:background="#ffffff"  
 android:orientation="horizontal">  
  
 *<!--密码-->* <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Password："  
 android:textSize="20sp"  
 android:textColor="#000"  
 android:padding="10dp"/>  
  
 *<!--密码输入-->* <EditText  
 android:id="@+id/password"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="3dp"  
 android:hint="pwd plz"  
 android:password="true"  
 android:padding="10dp"/>  
 </LinearLayout>  
 *<!--密码部分-->* </LinearLayout>  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="30dp"  
 android:text=""/>  
  
 *<!--留出一些空白-->* <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="30dp"  
 android:layout\_marginEnd="40dp">  
  
 *<!--'是否记住密码'复选框-->* <CheckBox  
 android:id="@+id/save\_password"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true" />  
 *<!--复选框提示信息-->* <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_toEndOf="@id/save\_password"  
 android:text="Remember psw ?"  
 android:textSize="18sp" />  
  
 *<!--复选框提示信息-->* <TextView  
 android:id="@+id/show\_password\_info"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentRight="true"  
 android:text="Show psw ?"  
 android:textSize="18sp" />  
  
 *<!--'是否明文显示密码'复选框-->* <CheckBox  
 android:id="@+id/show\_password"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_toLeftOf="@id/show\_password\_info" />  
  
 </RelativeLayout>  
  
 *<!--留白-->* <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="30dp"  
 android:text=""/>  
  
 *<!--画一条分割线-->* <View  
 android:layout\_width="match\_parent"  
 android:layout\_height="1dp"  
 android:background="#EEE9E9"/>  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="15dp"  
 android:text=""/>  
  
 *<!--留出一些空白-->* <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="15dp"  
 android:text=""/>  
  
 *<!--登录按钮-->* <Button  
 android:id="@+id/login\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="18sp"  
 android:text="Sign"  
 android:layout\_gravity="center"  
 android:background="@drawable/buttonshape" />  
  
</LinearLayout>  
</FrameLayout>

1. activity\_detail.xml

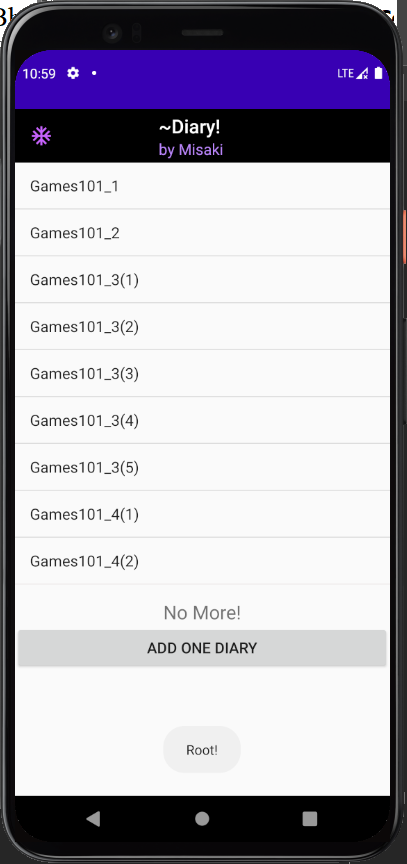
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/activity\_detail"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <androidx.appcompat.widget.Toolbar  
 android:background="@color/black"  
 app:title="~Diary!"  
 app:titleTextColor="@color/white"  
 android:id="@+id/toolbar"  
 app:subtitle="by Misaki"  
 app:subtitleTextColor="@color/purple\_200"  
 app:titleMarginStart="95dp"  
 app:navigationIcon="@drawable/ic\_baseline\_ac\_yuki\_24"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 app:menu="@menu/menu\_main"  
 android:theme="@style/Theme.AppCompat.NoActionBar"/>  
  
 <EditText  
 android:id="@+id/detail\_title"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 android:gravity="center"  
 android:hint="Title"  
 android:background="#D1EEEE"  
 android:textSize="30sp"/>  
  
 <EditText  
 android:id="@+id/detail\_author"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 android:gravity="center"  
 android:hint="Author"  
 android:textSize="18sp"  
 android:layout\_marginTop="10dp"/>  
  
 <TextView  
 android:id="@+id/detail\_time"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 android:gravity="center"  
 android:hint="Time"  
 android:textSize="18sp"  
 android:layout\_marginTop="10dp"/>  
  
 <EditText  
 android:id="@+id/detail\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 android:gravity="center"  
 android:hint="Content"  
 android:textSize="18sp"  
 android:layout\_marginTop="10dp"/>  
  
 <Button  
 android:id="@+id/detail\_pictureChoice"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="CHOOSE ONE PHOTO"  
 />  
  
 <ImageView  
 android:id="@+id/detail\_picture"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="30dp"  
 android:layout\_marginRight="30dp"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginBottom="40dp"  
 android:layout\_gravity="center\_horizontal"/>  
  
</LinearLayout>

运行结果：

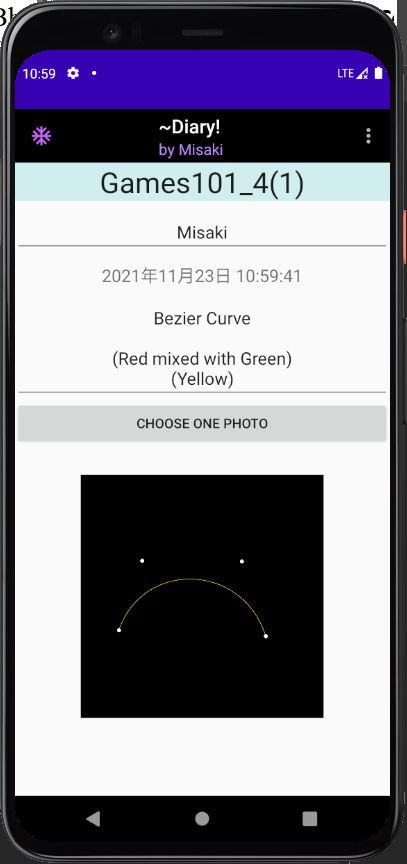
Login：



MainActivity：



Detail：



仓库：

<https://github.com/Minakami-Yuuki/Assignments.git>

1. 实验总结：

在本次实验中，主要要求实现日记的存储功能，并需要添加照片，将其存入SQLite轻量数据库中。

本次实验难点：如何正确的获取虚拟机相册权限，如何构建并连接SQLite数据库，如何采用SharePreferences进行数据持久化，保存日记的作者信息。

本次实验，我采用了相对布局、线性布局和单帧布局，并采用碎片化作为主布局，进行对各个子活动的调用。在Java代码实现中，我加入了一个登录界面，搭配SharePreferences记录登录的作者名称，并将其保存到日记的作者栏。在日记创建方面，共有：标题、作者、当前时间、内容和图片；我将其存至数据库保存，以便进行增删改查。

本次实验我了解到了更多的Android开发知识，并能够在界面上进一步改进，希望以后能够有更过机会接触Android开发。