# 熊猫记事本

## 项目描述

熊猫记事本是一个方便用户能够通过多种方式快速记录信息为目的而开发的APP。记录方式包括普通的键盘输入、手写笔记记录、语音记录、本地图片插入记录、拍照记录。为了增强项目的实用性，熊猫记事本APP增加了拓展功能，可以给自己的记事本加锁、把记事分享给好友、设置类似于闹钟的提醒功能。

## 实现

Sqlite数据库+MediaPlayer+自定义ScrollView控件实现

## 系统功能设计

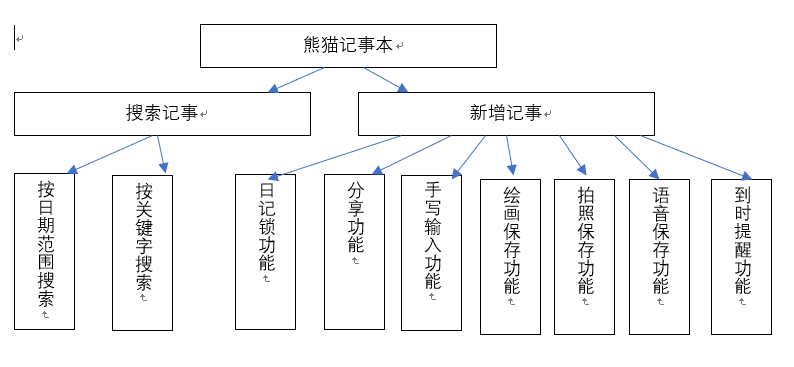


图3.1系统功能结构

熊猫记事本APP包含搜索功能，记事锁功能，分享功能，手写功能，绘画保存功能，语音保存功能，拍照保存功能，到时提醒功能，其系统功能图如图3.1。

## 界面展示

### 系统欢迎界面4.1



图4.1 图4.2

### 输入密码进入记事本4.2

### C:\Users\绘梦\Documents\Tencent Files\318361459\FileRecv\MobileFile\Screenshot_2018-12-23-16-25-19-182_com.mingrisoft.png记事本主界面4.3

图4.3 图4.4

### C:\Users\绘梦\Documents\Tencent Files\318361459\FileRecv\MobileFile\Screenshot_2018-12-23-16-27-46-755_com.mingrisoft.png添加记事

## 主要代码

### Sqlbean

**public class** SQLBean {  
 **private** String **\_id**; *//主键自增id* **private** String **title**; *//日记标题* **private** String **context**; *//日记内容* **private** String **time**; *//日记记录时间* **private** String **datatype**; *//是否设置了到时提醒0代表未提醒* **private** String **datatime**; *//提醒时间* **private** String **locktype**; *//是否添加了密码锁0代表未添加* **private** String **lock**; *//密码锁密码* **public** String get\_id() {*//获取日记id* **return \_id**;  
 }  
 **public void** set\_id(String \_id) {*//设置日记id* **this**.**\_id** = \_id;  
 }  
 **public** String getTitle() {*//获取日记标题* **return title**;  
 }  
 **public void** setTitle(String title) {*//设置日记标题* **this**.**title** = title;  
 }  
 **public** String getContext() {*//获取日记内容* **return context**;  
 }  
 **public void** setContext(String context) {*//设置日记内容* **this**.**context** = context;  
 }  
 **public** String getTime() {*//获取日记添加时间* **return time**;  
 }  
 **public void** setTime(String time) {*//设置日记添加时间* **this**.**time** = time;  
 }  
 **public** String getDatatype() {*//获取是否设置了到时提醒* **return datatype**;  
 }  
 **public void** setDatatype(String datatype) {*//设置日日期到时提醒* **this**.**datatype** = datatype;  
 }  
 **public** String getDatatime() {*//获取提醒时间* **return datatime**;  
 }  
 **public void** setDatatime(String datatime) {*//设置提醒时间* **this**.**datatime** = datatime;  
 }  
 **public** String getLocktype() {*//获取是否添加了日记锁* **return locktype**;  
 }  
 **public void** setLocktype(String locktype) {*//设置日记锁* **this**.**locktype** = locktype;  
 }  
 **public** String getLock() {*//获取日记锁秘密* **return lock**;  
 }  
 **public void** setLock(String lock) {*//设置日记锁秘密* **this**.**lock** = lock;  
 }  
}

### LoginActivity.java

1. **public class** LoginActivity **extends** Activity {  
     
    @Override  
    **protected void** onCreate(Bundle savedInstanceState) {  
    **super**.onCreate(savedInstanceState);  
    setContentView(R.layout.***activity\_login***);  
    *// 查看本app的私有目录* }  
     
    *//登录按钮点击事件* **public void** onLogin(View view){  
    EditText edt = (EditText)findViewById(R.id.***mima***);  
    String pass2 = edt.getText().toString();  
    **if**(!fileIsExists()){  
    *//创建密码“0000”* String mi = **"0000"**;  
    File appDir = getFilesDir();  
    File f = **new** File(appDir, **"abc.txt"**);  
    FileOutputStream fstream = **null**;  
    **try**{  
    fstream = **new** FileOutputStream(f);  
    fstream.write( mi.getBytes(**"UTF-8"**));  
    }**catch**(Exception e){  
    }  
    **finally** {  
    **try**{ fstream.close();} **catch** (Exception e2){}  
    }  
    *//Toast.makeText(this, "已保存", Toast.LENGTH\_SHORT).show();* **if**(pass2.equals(**"0"**)){  
    Log.*d*(**"测试"**,**"登录成功！！！"**);  
    Toast.*makeText*(LoginActivity.**this**,**"登录成功1"**,Toast.***LENGTH\_LONG***).show();  
    startActivity(**new** Intent(LoginActivity.**this**,MainActivity.**class**));  
    finish();  
    }**else** Toast.*makeText*(LoginActivity.**this**,**"密码错误！！"**,Toast.***LENGTH\_LONG***).show();  
    } **else** {  
    File f = **new** File( getFilesDir(), **"abc.txt"**);  
    FileInputStream fstream = **null**;  
    **try**{  
    fstream = **new** FileInputStream(f);  
    **byte**[] data = **new byte**[1024]; *// 当文件很大时，应分作多次读取, 这里是简易写法仅作演示* **int** n = fstream.read(data);  
    **if**(n> 0)  
    {  
    *// 当读取的到byte[]转成String* String str = **new** String(data,0, n, **"UTF-8"**);  
    Log.*d*(**"测试"**, **"读取密码: "** + str);  
    **if**(pass2.equals(str)){  
    Log.*d*(**"测试"**,**"登录成功23123123！！！"**);  
    Toast.*makeText*(LoginActivity.**this**,**"登录成功2"**,Toast.***LENGTH\_LONG***).show();  
    startActivity(**new** Intent(LoginActivity.**this**,MainActivity.**class**));  
    finish();  
    }**else** Toast.*makeText*(LoginActivity.**this**,**"密码错误！！！002"**+str,Toast.***LENGTH\_LONG***).show();  
    }  
    } **catch** (UnsupportedEncodingException e) {  
    e.printStackTrace();  
    } **catch** (FileNotFoundException e) {  
    e.printStackTrace();  
    } **catch** (IOException e) {  
    e.printStackTrace();  
    } **finally** {  
    **try**{ fstream.close();}**catch** (Exception e){}  
    }  
     
     
    }  
   *// if(pass.equals("1111")){  
   // Log.d("测试","登录成功！！！");  
   // Toast.makeText(LoginActivity.this,"登录成功",Toast.LENGTH\_LONG).show();  
   // startActivity(new Intent(LoginActivity.this,Main2Activity.class));  
   // finish();  
   //  
   // }else {  
   // Toast.makeText(LoginActivity.this,"登录失败，密码错误",Toast.LENGTH\_LONG).show();  
   // }* }  
    *//退出按钮点击事件* **public void** onTuichu(View view){  
    **this**.finish();  
     
    }  
    *//点击关于按钮的事件* **public void** onAbout(View view){  
    Toast.*makeText*(LoginActivity.**this**,**"关于！"**,Toast.***LENGTH\_LONG***).show();  
    startActivity(**new** Intent(LoginActivity.**this**,Main2Activity.**class**));  
     
     
    Log.*d*(**"测试"**,**"点击了关于按钮"**);  
     
    }  
    *//判断是否设置过密码* **public boolean** fileIsExists(){  
    **try**{  
    File f=**new** File(getFilesDir()+**"/abc.txt"**);  
    **if**(!f.exists()){  
    Log.*d*(**"测试"**,**"不存在"**);  
    **return false**;  
    }  
    }**catch** (Exception e) {  
    *//* ***TODO: handle exception*** Log.*d*(**"测试"**,**"不存在chucuo"**);  
    **return false**;  
    }  
    Log.*d*(**"测试"**,**"存在"**);  
    **return true**;  
    }  
     
   }

### MainActivity.java

1. **public class** MainActivity **extends** Activity {  
    **private** Button **bt\_add**;*// 添加按钮* **private** SQLiteDatabase **db**;*//数据库对象* **private** DatabaseOperation **dop**;*//自定义数据库类* **private** MyGridView **lv\_notes**;*// 消息列表* **private** TextView **tv\_note\_id**, **tv\_locktype**, **tv\_lock**;  
    **public static** MediaPlayer *mediaPlayer*;*// 音乐播放器* **public static** Vibrator *vibrator*;*//手机震动器* **public** AlarmManager **am**;*// 消息管理者* **public** EditText **et\_keyword**;*// 搜索框* **public** MainAdapter **adapter**;  
     
    @Override  
    **protected void** onCreate(Bundle savedInstanceState) {  
    **super**.onCreate(savedInstanceState);  
    setContentView(R.layout.***activity\_main***);  
    **bt\_add** = (Button) findViewById(R.id.***bt\_add***);  
    **bt\_add**.setOnClickListener(**new** ClickEvent());  
    **et\_keyword** = (EditText) findViewById(R.id.***et\_keyword***);  
    *// 数据库操作* **dop** = **new** DatabaseOperation(**this**, **db**);  
    **lv\_notes** = (MyGridView) findViewById(R.id.***lv\_notes***);  
    **if** (**am** == **null**) {  
    **am** = (AlarmManager) getSystemService(***ALARM\_SERVICE***);  
    }  
    **try** {  
    Intent intent = **new** Intent(MainActivity.**this**, CallAlarm.**class**);  
    PendingIntent sender = PendingIntent.*getBroadcast*(  
    MainActivity.**this**, 0, intent, 0);  
    **am**.setRepeating(AlarmManager.***RTC\_WAKEUP***, 0, 60 \* 1000, sender);  
    } **catch** (Exception e) {  
    e.printStackTrace();  
    }  
    }  
    @Override  
    **protected void** onStart() {  
    *//* ***TODO Auto-generated method stub* super**.onStart();  
    *// 显示记事列表* showNotesList();  
    *// 为记事列表添加监听器* **lv\_notes**.setOnItemClickListener(**new** ItemClickEvent());  
    *// 为记事列表添加长按事件* **lv\_notes**.setOnItemLongClickListener(**new** ItemLongClickEvent());  
    }  
    *// 显示记事列表* **private void** showNotesList() {  
    *// 创建或打开数据库 获取数据* **dop**.create\_db();  
    *//获取数据库内容* Cursor cursor = **dop**.query\_db();  
    **if** (cursor.getCount() > 0) {  
    List<SQLBean> list = **new** ArrayList<SQLBean>();*//日记信息集合里* **while** (cursor.moveToNext()) {*// 光标移动成功  
    // 把数据取出* SQLBean bean = **new** SQLBean();*//创建数据库实体类  
    //保存日记信息id到实体类* bean.set\_id(**""** + cursor.getInt(cursor.getColumnIndex(**"\_id"**)));  
    bean.setContext(cursor.getString(cursor  
    .getColumnIndex(**"context"**)));*//保存日记内容到实体类  
    //保存日记标题到实体类* bean.setTitle(cursor.getString(cursor.getColumnIndex(**"title"**)));  
    *//保存日记记录时间到实体类* bean.setTime(cursor.getString(cursor.getColumnIndex(**"time"**)));  
    bean.setDatatype(cursor.getString(cursor  
    .getColumnIndex(**"datatype"**)));*//保存日记是否设置提醒时间到实体类* bean.setDatatime(cursor.getString(cursor  
    .getColumnIndex(**"datatime"**)));*//保存日记提醒时间到实体类* bean.setLocktype(cursor.getString(cursor  
    .getColumnIndex(**"locktype"**)));*//保存日记是否设置了日记锁到实体类  
    //保存日记锁秘密到实体类* bean.setLock(cursor.getString(cursor.getColumnIndex(**"lock"**)));  
    list.add(bean);*//把保存日记信息实体类保存到日记信息集合里* }  
    *//倒序显示数据* Collections.*reverse*(list);  
    **adapter** = **new** MainAdapter(list, **this**);*//装载日记信息到首页* **lv\_notes**.setAdapter(**adapter**);*//日记列表设置日记信息适配器* }  
    **dop**.close\_db();*//关闭数据库* }  
    *// 记事列表长按监听器* **class** ItemLongClickEvent **implements** OnItemLongClickListener {  
    @Override  
    **public boolean** onItemLongClick(AdapterView<?> parent, View view,  
    **int** position, **long** id) {  
    *//初始化日记id保存控件* **tv\_note\_id** = (TextView) view.findViewById(R.id.***tv\_note\_id***);  
    *//初始化是否添加日记锁保存控件* **tv\_locktype** = (TextView) view.findViewById(R.id.***tv\_locktype***);  
    *//初始化日记锁秘密保存信息* **tv\_lock** = (TextView) view.findViewById(R.id.***tv\_lock***);  
    *//获取控件上是否设置日记锁信息* String locktype = **tv\_locktype**.getText().toString();  
    *//获取控件上日记密码信息* String lock = **tv\_lock**.getText().toString();  
    *//获取控件上id信息转换成int类型* **int** item\_id = Integer.*parseInt*(**tv\_note\_id**.getText().toString());  
    *//弹出选择操作框方法* simpleList(item\_id, locktype, lock);  
    **return true**;  
    }  
    }  
    *// 简单列表对话框，用于选择操作* **public void** simpleList(**final int** item\_id, **final** String locktype,  
    **final** String lock) {  
    *//实例化AlertDialog* AlertDialog.Builder alertDialogBuilder = **new** AlertDialog.Builder(**this**,  
    R.style.***custom\_dialog***);  
    *//设置弹窗标题* alertDialogBuilder.setTitle(**"选择操作"**);  
    *//设置弹窗图片* alertDialogBuilder.setIcon(R.mipmap.***ic\_launcher***);  
    *//设置弹窗选项内容* alertDialogBuilder.setItems(R.array.***itemOperation***,  
    **new** android.content.DialogInterface.OnClickListener() {  
    @Override  
    **public void** onClick(DialogInterface dialog, **int** which) {  
    **switch** (which) {  
    *// 编辑* **case** 0:  
    **if** (**"0"**.equals(locktype)) {*//判断是否添加了秘密锁0没有* Intent intent = **new** Intent(MainActivity.**this**,  
    AddActivity.**class**);*//跳转到添加日记页* intent.putExtra(**"editModel"**, **"update"**);*//传递编辑信息* intent.putExtra(**"noteId"**, item\_id);*//传递id信息* startActivity(intent);*//开始跳转* } **else** {*//有秘密锁  
    // 弹出输入密码框* inputTitleDialog(lock, 0, item\_id);  
    }  
    **break**;  
    *// 删除* **case** 1:  
    **if** (**"0"**.equals(locktype)) {*// 判断是否是加密日记 0没有* **dop**.create\_db();*// 打开数据库* **dop**.delete\_db(item\_id);*//删除数据* **dop**.close\_db();*// 关闭数据库  
    // 刷新列表显示* **lv\_notes**.invalidate();  
    showNotesList();  
    } **else** {*//有秘密锁  
    // 弹出输入密码框* inputTitleDialog(lock, 1, item\_id);  
    *// 刷新列表显示* **lv\_notes**.invalidate();  
    *//显示日记列表信息* showNotesList();  
    }  
    **break**;  
    }  
    }  
    });  
    alertDialogBuilder.create();*//创造弹窗* alertDialogBuilder.show();*//显示弹窗* }  
    *// 记事列表单击监听器* **class** ItemClickEvent **implements** OnItemClickListener {  
    @Override  
    **public void** onItemClick(AdapterView<?> parent, View view, **int** position,  
    **long** id) {  
    **tv\_note\_id** = (TextView) view.findViewById(R.id.***tv\_note\_id***);  
    **tv\_locktype** = (TextView) view.findViewById(R.id.***tv\_locktype***);  
    **tv\_lock** = (TextView) view.findViewById(R.id.***tv\_lock***);  
    String locktype = **tv\_locktype**.getText().toString();  
    String lock = **tv\_lock**.getText().toString();  
    **int** item\_id = Integer.*parseInt*(**tv\_note\_id**.getText().toString());  
    **if** (**"0"**.equals(locktype)) {  
    Intent intent = **new** Intent(MainActivity.**this**, AddActivity.**class**);  
    intent.putExtra(**"editModel"**, **"update"**);  
    intent.putExtra(**"noteId"**, item\_id);  
    startActivity(intent);  
    } **else** {  
    inputTitleDialog(lock, 0, item\_id);  
    }  
    }  
    }  
    *// 加密日记打开弹出的输入密码框* **public void** inputTitleDialog(**final** String lock, **final int** idtype,  
    **final int** item\_id) {*// 密码输入框* **final** EditText inputServer = **new** EditText(**this**);  
    inputServer.setInputType(InputType.***TYPE\_CLASS\_TEXT*** | InputType.***TYPE\_TEXT\_VARIATION\_PASSWORD***);  
    inputServer.setFocusable(**true**);  
    AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
    builder.setTitle(**"请输入密码"**).setView(inputServer)  
    .setNegativeButton(**"取消"**, **null**);  
    builder.setPositiveButton(**"确认"**, **new** DialogInterface.OnClickListener() {  
     
    **public void** onClick(DialogInterface dialog, **int** which) {  
    String inputName = inputServer.getText().toString();  
    **if** (**""**.equals(inputName)) {  
    Toast.*makeText*(MainActivity.**this**, **"密码不能为空请重新输入！"**,  
    Toast.***LENGTH\_LONG***).show();  
    } **else** {  
    **if** (inputName.equals(lock)) {  
    **if** (0 == idtype) {  
    Intent intent = **new** Intent(MainActivity.**this**,  
    AddActivity.**class**);  
    intent.putExtra(**"editModel"**, **"update"**);  
    intent.putExtra(**"noteId"**, item\_id);  
    startActivity(intent);  
    } **else if** (1 == idtype) {  
    **dop**.create\_db();  
    **dop**.delete\_db(item\_id);  
    **dop**.close\_db();  
    *// 刷新列表显示* **lv\_notes**.invalidate();  
    showNotesList();  
    }  
    } **else** {  
    Toast.*makeText*(MainActivity.**this**, **"密码不正确！"**,  
    Toast.***LENGTH\_LONG***).show();  
    }  
    }  
    }  
    });  
    builder.show();  
    }  
    *// 点击事件* **class** ClickEvent **implements** OnClickListener {  
    @Override  
    **public void** onClick(View v) {  
    **switch** (v.getId()) {  
    *// 添加记事* **case** R.id.***bt\_add***:  
    Intent intent = **new** Intent(MainActivity.**this**, AddActivity.**class**);  
    intent.putExtra(**"editModel"**, **"newAdd"**);  
    startActivity(intent);  
    }  
    }  
    }  
    *// 搜索功能* **public void** onSearch(View v) {  
    *//获取搜索关键词* String ek = **et\_keyword**.getText().toString();  
    **if** (**""**.equals(ek)) {*//判断搜索关键词是否为空* Toast.*makeText*(MainActivity.**this**, **"请输入关键词！"**, Toast.***LENGTH\_LONG***)  
    .show();  
    } **else** {*//搜索不为空  
    //进入搜索结果页* Intent intent = **new** Intent(MainActivity.**this**, SearchActivity.**class**);  
    intent.putExtra(**"keword"**, ek);*//传递关键词* startActivity(intent);*//开始跳转* }  
    }  
    *// 日期范围搜索* **public void** onData(View v) {  
    *// 最后一个false表示不显示日期，如果要显示日期，最后参数可以是true或者不用输入* Calendar c = Calendar.*getInstance*();  
    **new** DoubleDatePickerDialog(MainActivity.**this**, 0,  
    **new** DoubleDatePickerDialog.OnDateSetListener() {  
    @Override  
    **public void** onDateSet(DatePicker startDatePicker,  
    **int** startYear, **int** startMonthOfYear,  
    **int** startDayOfMonth, DatePicker endDatePicker,  
    **int** endYear, **int** endMonthOfYear, **int** endDayOfMonth) {  
    **if** (startYear < endYear || startYear == endYear  
    && startMonthOfYear <= endMonthOfYear) {  
    **int** st = startMonthOfYear + 1;  
    **int** et = endMonthOfYear + 1;  
    Intent intent = **new** Intent(MainActivity.**this**,  
    DataSearchActivity.**class**);  
    *// sql判断 需要在月份前补0 否则sql语句判断不正确。* **if** (st < 10) {  
    intent.putExtra(**"startData"**, startYear + **"-0"** + st + **"-"** + **"01"**);  
    } **else** {  
    intent.putExtra(**"startData"**, startYear + **"-"** + st + **"-"** + **"01"**);  
    }  
    **if** (et < 10) {  
    intent.putExtra(**"endData"**, endYear + **"-0"** + et  
    + **"-"** + **"30"**);  
    } **else** {  
    intent.putExtra(**"endData"**, endYear + **"-"** + et  
    + **"-"** + **"30"**);  
    }  
    startActivity(intent);  
    } **else** {  
    Toast.*makeText*(MainActivity.**this**, **"日期选择错误请重新选择！"**,  
    Toast.***LENGTH\_LONG***).show();  
    }  
    }  
    }, c.get(Calendar.***YEAR***), c.get(Calendar.***MONTH***), c  
    .get(Calendar.***DATE***), **false**).show();  
    }  
    *// 进入关于页* **public void** onAbout(View v) {  
    Intent intent = **new** Intent(MainActivity.**this**, About.**class**);  
    startActivity(intent);  
    }  
   }