PasswordKeeper开发报告

Group02：

16340218 王阔

16340212 王季宁

16340211 王广浩

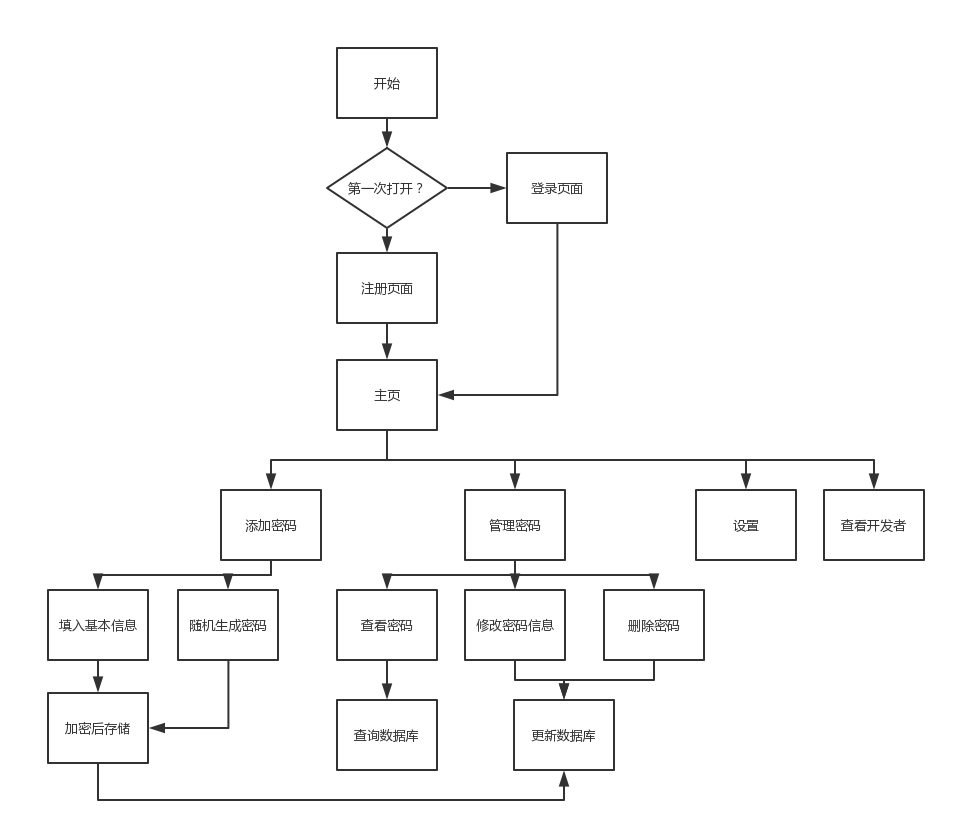
1. **背景**

本次期末项目完全是出于自身的需要来进行设计的。我们发现自己有很多的密码需要保管，在登录游戏的时候往往忘记了自己的密码是多少十分尴尬。现在市场上的密码管理软件基本上都要充值才能解锁全部功能，作为穷学生的我们是没钱充值的。于是我们便自己开发了这个产品。

1. **概述**

这是一个密码管理软件，用来保存你的各种密码。每个密码都有自己的分类，名称等信息。通过这些信息你可以很快地找到自己的密码，再也不用担心忘记密码的问题。你首先要登录之后才能看到自己的密码，所以实际上只有你可以看到自己的密码。密码的保存方式是先加密再存储的，保证了你的密码的安全性。你可以修改自己保存的密码，删除自己的保存的密码。还可以自动地生成一个密码来作为你的密码，自动生成的密码的格式是可以进行设置的。实现了密码管理软件的所有基本功能。

1. **使用流程图**

****

1. **使用方式**

见用户手册

1. **技术重点**
2. 加密算法：使用Bcrypt加密算法，单向加密，不可逆，单间加密过程较长，可防御彩虹桥攻击以及基本的撞库攻击。安全性较高。详情见code中的
3. 设计算法调用的接口：

/创建表，参数是数据库和判断用户是否存在的布尔值

createAllTables(Database db, boolean ifNotExists)

//删除表，参数是数据库和判断用户是否存在的布尔值

public static void dropAllTables(Database db, boolean ifExists)

//初始化Session

public static DaoSession newDevSession(Context context, String name)

//初始化DaoMaster

public DaoMaster(SQLiteDatabase db)

//初始化Daomaster

//获取新的Session

public DaoSession newSession(IdentityScopeType type)

// 对数据库进行更新

public void onUpgrade(Database db, int oldVersion, int newVersion)

// 向数据库中添加值

protected final void bindValues(DatabaseStatement stmt, Item entity)

//向数据库添加值

protected final void bindValues(SQLiteStatement stmt, Item entity)

// 读取数据库键值，输入参数为游标和偏移量

public Long readKey(Cursor cursor, int offset)

// 读取数据库实体，输入参数为游标和偏移量

public Item readEntity(Cursor cursor, int offset)

// 读取数据库实体并赋值给新的实体，输入参数为游标、实体和偏移量

public void readEntity(Cursor cursor, Item entity, int offset)

// 获取实体对应的键值，输入参数为实体

public Long getKey(Item entity)

// 对用户名和密码进行校验

public static boolean authenticateUser(String username, String password)

// 判断条目是否重复

public static boolean isItemRepeated(String mSpinnerStr, String name)

//获取类别id

private static int getCategoryId(String mSpinnerStr)

// 验证老密码是否正确

public static boolean isOldPasswordCorrect(String oldPassword)

// 对输入字符串进行base64编码，输入参数为字符数组和长度

private static String encode\_base64(byte d[], int len)

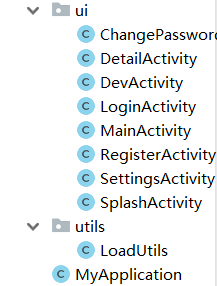
// 对输入字符串进行base64解码，输入参数为字符串和长度

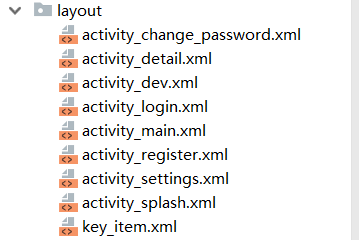
private static byte[] decode\_base64(String s, int maxolen)

1. UI在Manifast文件中依次列举各个界面所在的文件夹，为了充分利用前端编程语言在绘制界面效果方面的优势，使用webview来直接调用写好了的HTML文件。同时为了让画面更加地整洁大方，取消了安卓默认的上边框等等设计。
2. 数据存储：使用数据库访问ORM框架GreenDao，使用简单的面相对象的API来存储，更新，删除和查询Java对象，同时支持数据库加密
3. **代码架构**

代码文件结构截图（就是as左边的那个目录结构），讲下每个文件夹里存了哪些文件，分别有啥用。

**UI部分：**

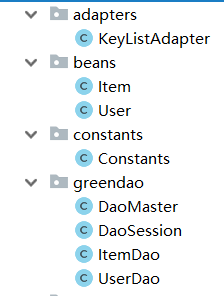




分别定义了开始页面、登录页面、注册页面、设置页面等等的外观和行为，以及各个页面之间复杂的相互跳转的关系。

**存储部分：**

使用GreenDao数据库访问模式



DaoMaster： 保存数据库对象并管理特定模式的DAO类。它有静态方法来创建表或删除它们。

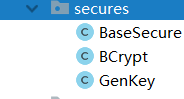
DaoSession：管理特定模式的所有可用DAO对象，使用其中一个getter方法获取该对象。DaoSession还提供了一些通用的持久性方法，如实体的插入，加载，更新，刷新和删除。

ItemDao、UserDao：数据访问对象（DAO）持久存在并查询实体。对于每个实体，greenDAO生成DAO。

Item、User：可持久化对象。通常, 实体对象代表一个数据库。

KeyListAdapter：用于根据数据库中的内容初始化View控件

**加密部分：**



Secury文件夹内是密码管理和随机密码生成的代码。Bcrypt.java是加密算法的实现，BaseSecure.java是加密算法的调用接口，提供了一系列的函数供其他模块调用，GenKey.java用于随机生成密码。

1. **小组分工**

|  |  |
| --- | --- |
| 王阔 16340218 | 加密算法的设计，提供加密算法的调用接口，实现随机密码的生成，连接各个模块的activity文件的编写，将大家分头编写的内容联系起来，使其能够完美运行，以及各种文档的编写。 |
| 王广浩 16340211 | 数据类的实现，负责实现数据存储，同时负责编写adapter与recycleView进行对接，协助文档编写。 |
| 王季宁 16340212 | 整体UI的实现，初始化界面的实现，负责界面之间的切换 |