

移动应用开发课程实验 – 登录界面的设计与实现

指导老师：曹阳

2017 年 11 月 5 日

华南师范大学计算机学院

计算机科学与技术（软件技术应用方向）

2015 级 5 班

王子彦

20142100012

ziyan.wang@m.scnu.edu.cn

声明：除特殊注明外，本次作业中所有的文档、源代码和相关资源均为本人原创。

该项目的源代码已在 Github 上以 Apache-2.0 协议发布：

<https://github.com/lonelyenvoy/LoginDemo>

王子彦授予曹阳老师对本项目所有文件的查阅、拷贝、修改和使用的权利。

软件名称	LoginDemo
完成人	王子彦
学号	20142100012
完成时间	2017 年 11 月 5 日

一、 软件内容简介

本项目实现了一个国际化的登录演示 app，具有登录、注册和用户管理功能。此 app 支持中英双语，具有优雅的 UI 设计、完善的异常处理机制与友好的提示信息，使用动态加盐 MD5 算法来存储和验证密码，并成功防止了 SQL 注入，保证系统的高度安全性和鲁棒性。

截至本文发布前，本项目代码的最新版本为 1.0.82。

Github 项目 Readme 页面：
您可在[此处](#)查阅本项目的源码。

lonelyenvoy / LoginDemo

Watch

0

Star

0

Fork

0

Code

Issues0

Pull requests0

Projects0

Wiki

Insights

Settings

Branch: master

LoginDemo / README.md

Find file

Copy path

lonelyenvoy update readme

ff49b8a 2 minutes ago

1 contributor

9 lines (6 sloc) | 323 Bytes

Raw


Blame

History

LoginDemo

An internationalized demo Android app for Signing up and login, including simple user administration

Demo



License

[The Apache-2.0 License](#)

二、界面设计

本次 app 的设计中涉及到了四个界面，分别为登录界面、注册界面、用户管理与用户详细信息界面。以下为四个页面的中英双语对照设计图。

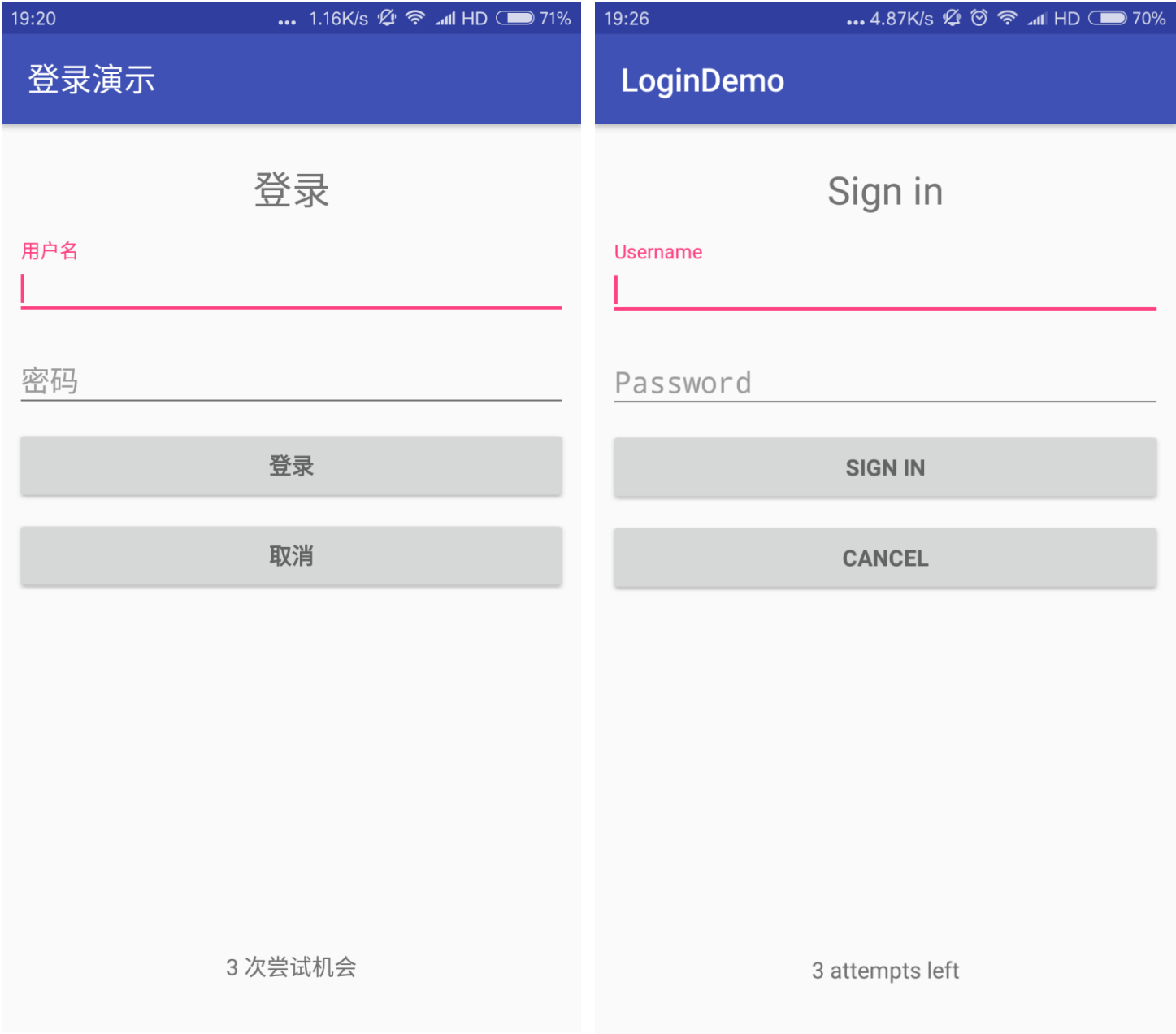


图 2-1 登录界面设计原型图

19:20 ... 1.16K/s 71%

注册

注册

用户名

密码

再次输入密码

注册

取消

19:26 ... 1.13K/s 70%

Sign up

Sign up

Username

Password

Repeat password

SIGN UP

CANCEL

图 2-2 注册界面设计原型图

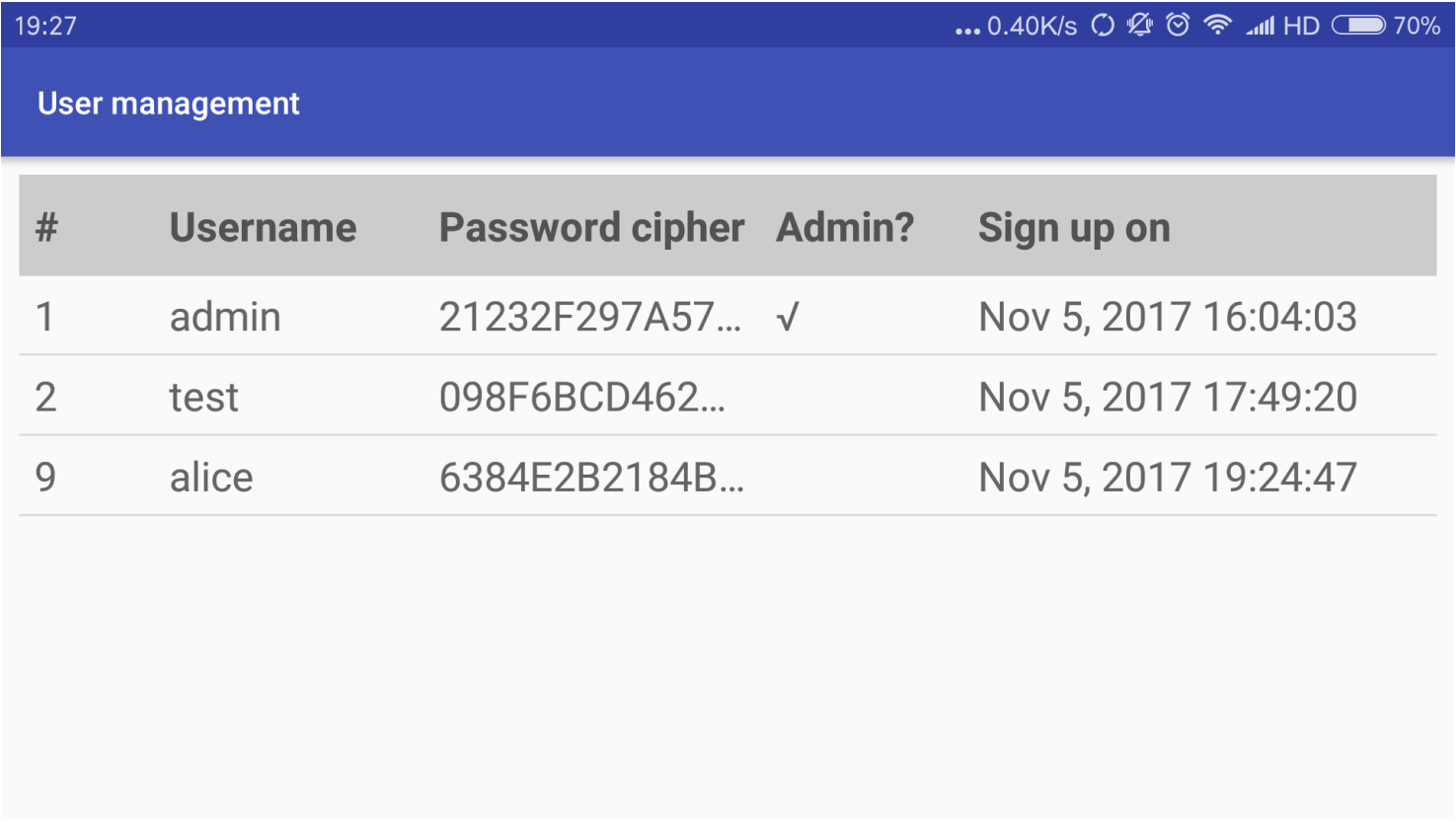


图 2-3 用户管理界面设计原型图（横屏）

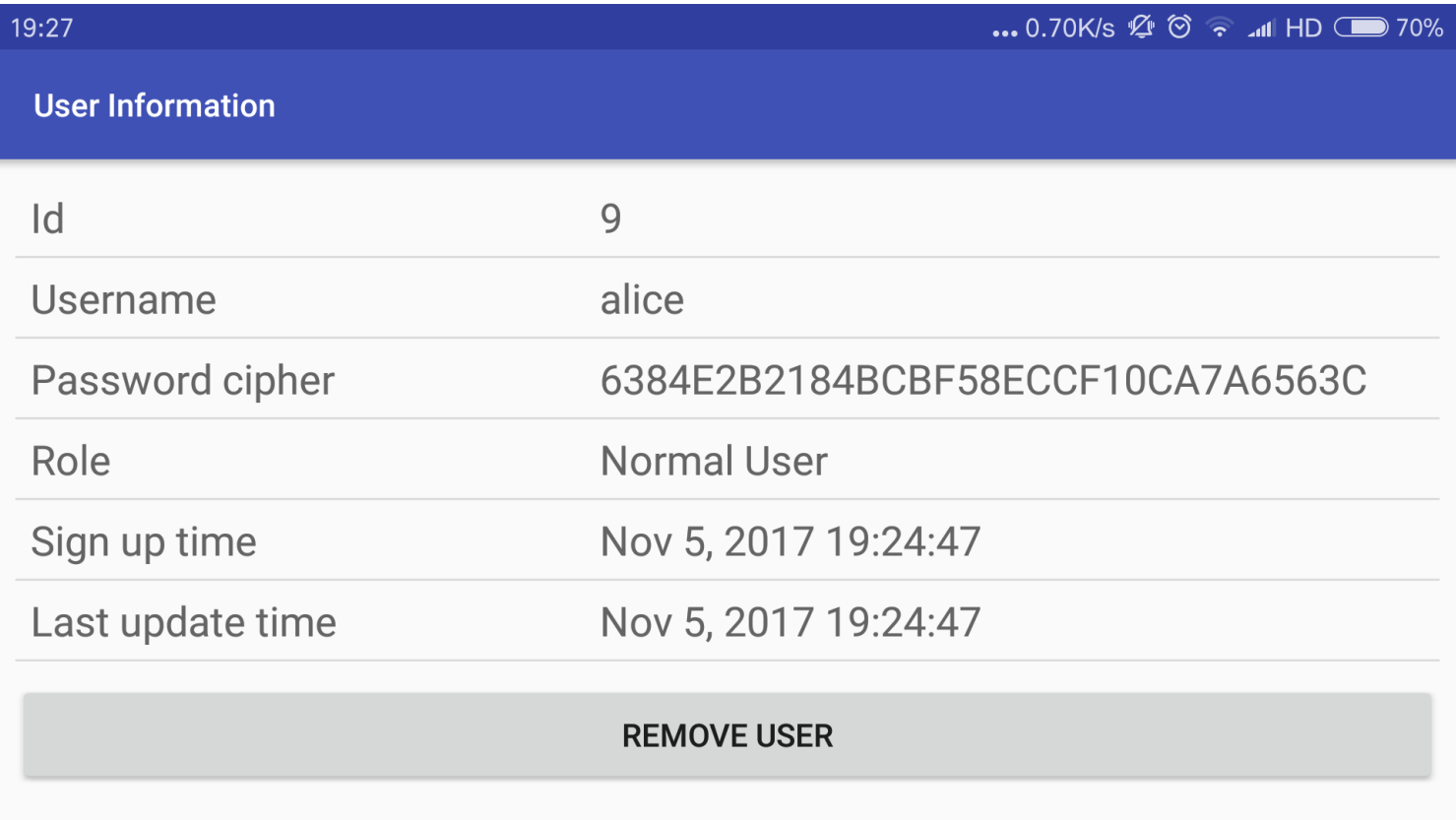


图 2-4 用户详细信息界面设计原型图（横屏）

为了您能更直观地了解本次作业的成果，请查看
同目录下的“[app 完整功能演示图](#)”动画

(约 40 秒)

三、项目设计中的亮点

3.1 滑动注册

在登录界面向左滑，可以进入注册页面。



图 3-1 滑动注册功能演示图

3.2 数据持久化存储

本 app 中所有的数据均在用户手机上永久存储，注册的信息在 app 被关闭后不会丢失。

3.3 较高的安全性

本 app 使用[动态加盐 MD5 算法](#)来存储和验证用户的密码，在数据库中存储密文而非明文，即使系统管理员也无法获取用户的密码，更不用说数据库被盗等情况，能够确保用户个人信息的安全。（算法将在下一节给出）

除此之外，还使用了谷歌提供的类库来[防御](#)恶意用户的[SQL 注入](#)行为。

当用户输错密码超过三次时，系统会进入[锁定](#)状态，并进行[倒计时](#)，用户需等待 [10 秒](#)才能再次登录。

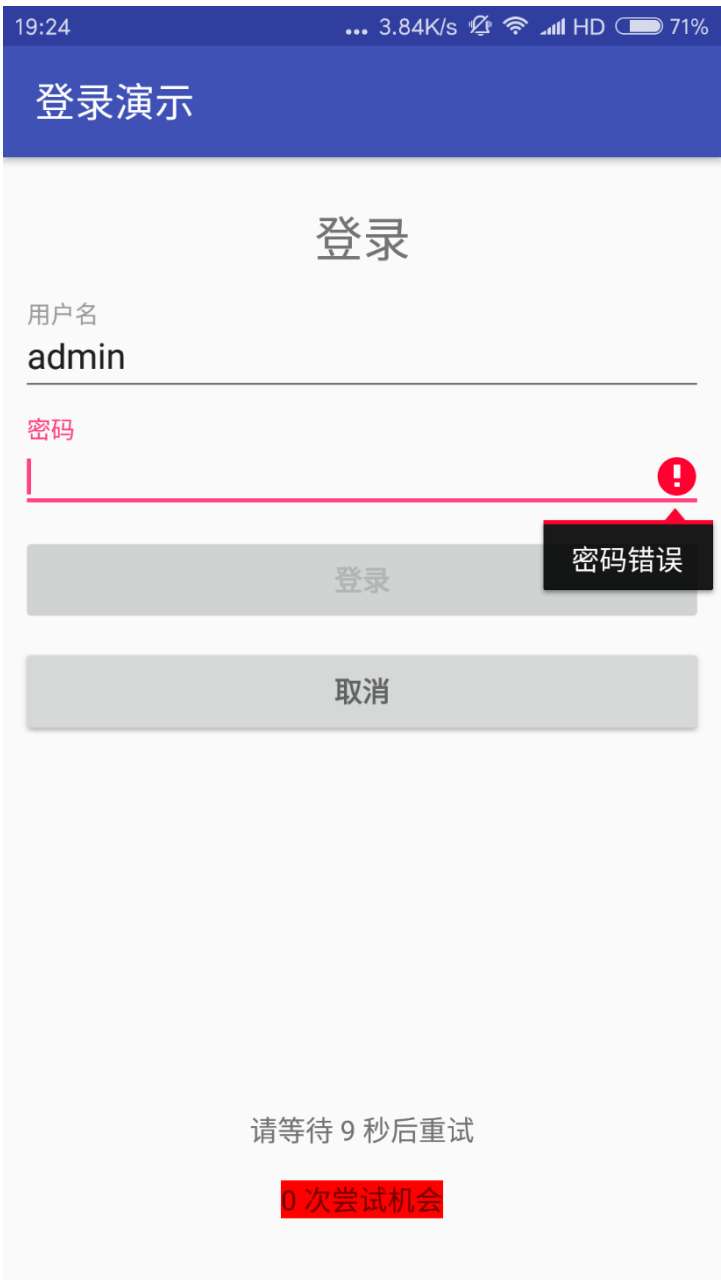


图 3-2 输错密码次数过多的锁定状态示意图

3.4 完善的用户信息

系统管理员可以看到所有用户的注册时间等[详细信息](#)，且可以[删除用户](#)。
普通用户也可以查看自己部分的[非敏感信息](#)。

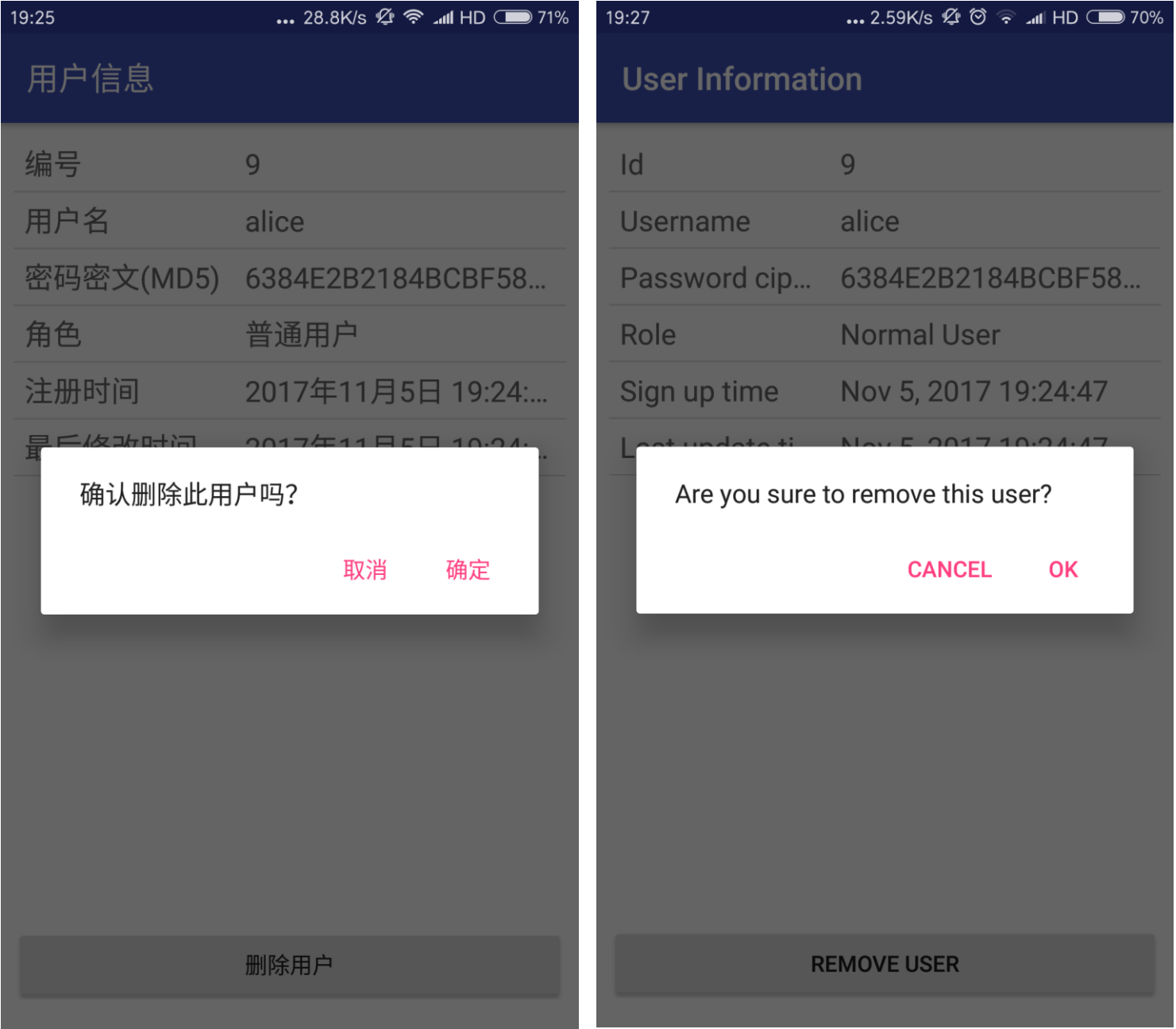


图 3-3 删除用户界面示意图

3.5 完善的异常处理机制与友好的用户提示信息

app 对用户的所有输入进行了大量的处理，确保出错时能够给用户反馈友好的提示信息。



图 3-4 登录时未填写信息的情况



图 3-5 登录时用户名不存在的情况



图 3-6 登录时用户名无效的情况



图 3-7 注册时用户名已存在的情况



图 3-8 注册时用户名太短的情况



图 3-9 注册时用户名太长的情况



图 3-10 注册时两次输入密码不一致的情况

由于情况较多，仅于实验报告中列出其中一部分。

3.6 详尽的国际化支持

本 app 中所有可显示的字符串都经过了双语翻译，以下列出部分。

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="action_ok">OK</string>
    <string name="action_cancel">Cancel</string>

    <string name="app_name">LoginDemo</string>
    <string name="hint_username">Username</string>
    <string name="hint_password">Password</string>
    <string name="action_sign_in">Sign in</string>
    <string name="action_sign_in_short">Sign in</string>

    <string name="error_invalid_username">This username is invalid</string>
    <string name="error_invalid_password_too_short">This password is too short</string>
    <string name="error_invalid_password_too_long">This password is too long</string>
    <string name="error_incorrect_password">This password is incorrect</string>
    <string name="error_field_required">This field is required</string>

    <string name="hint_attempts">attempts left</string>

    <string name="hint_login_successful">login successful</string>
    <string name="hint_login_failed">login failed</string>

    <string name="hint_sign_in_reenable_countdown_front">Retry after</string>
    <string name="hint_sign_in_reenable_countdown_rear">seconds</string>

    <string name="hint_repeat_password">Repeat password</string>
    <string name="action_sign_up">Sign up</string>
    <string name="error_inconsistent_passwords">The two passwords do not match</string>
    <string name="error_invalid_username_too_short">This username is too short</string>
    <string name="error_invalid_username_too_long">This username is too long</string>
    <string name="hint_sign_up_successful">Sign up successful</string>
    <string name="error_sign_up_failed_username_exists">Username occupied</string>
    <string name="error_invalid_username_not_exists">Username does not exist</string>

    <string name="hint_admin_activity_table_header_username">Username</string>
    <string name="hint_admin_activity_table_header_password">Password cipher</string>
    <string name="hint_admin_activity_table_header_isAdmin">Admin?</string>
    <string name="hint_admin_activity_table_header_createdAt">Sign up on</string>

    <string name="field_user_detail_activity_id">Id</string>
    <string name="field_user_detail_activity_username">Username</string>
    <string name="field_user_detail_activity_password">Password cipher</string>
    <string name="field_user_detail_activity_role">Role</string>
```

图 3-11 en\strings.xml 字符串资源英文版（部分）

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="action_ok">确定</string>
    <string name="action_cancel">取消</string>

    <string name="app_name">登录演示</string>
    <string name="hint_username">用户名</string>
    <string name="hint_password">密码</string>
    <string name="action_sign_in">登录</string>
    <string name="action_sign_in_short">登录</string>

    <string name="error_invalid_username">用户名无效</string>
    <string name="error_invalid_password_too_short">密码太短</string>
    <string name="error_invalid_password_too_long">密码太长</string>
    <string name="error_incorrect_password">密码错误</string>
    <string name="error_field_required">此为必填字段</string>

    <string name="hint_attempts">次尝试机会</string>

    <string name="hint_login_successful">登录成功</string>
    <string name="hint_login_failed">登录失败</string>

    <string name="hint_sign_in_reenable_countdown_front">请等待</string>
    <string name="hint_sign_in_reenable_countdown_rear">秒后重试</string>

    <string name="hint_repeat_password">再次输入密码</string>
    <string name="action_sign_up">注册</string>
    <string name="error_inconsistent_passwords">密码不一致</string>
    <string name="error_invalid_username_too_short">用户名太短</string>
    <string name="error_invalid_username_too_long">用户名太长</string>
    <string name="hint_sign_up_successful">注册成功</string>
    <string name="error_sign_up_failed_username_exists">用户名已存在</string>
    <string name="error_invalid_username_not_exists">用户名不存在</string>

    <string name="hint_admin_activity_table_header_username">用户名</string>
    <string name="hint_admin_activity_table_header_password">密码密文</string>
    <string name="hint_admin_activity_table_header_isAdmin">管理员?</string>
    <string name="hint_admin_activity_table_header_createdAt">注册时间</string>

    <string name="field_user_detail_activity_id">编号</string>
    <string name="field_user_detail_activity_username">用户名</string>
    <string name="field_user_detail_activity_password">密码密文(MD5)</string>
    <string name="field_user_detail_activity_role">角色</string>

```

图 3-12 zh\strings.xml 字符串资源中文版（部分）

四、代码设计

由于代码量较大，下面选择几项重要的部分展示。如有兴趣，请在 Android Studio 中 clone 开发者的 Github 代码仓库，打开项目文件进行详细查阅。

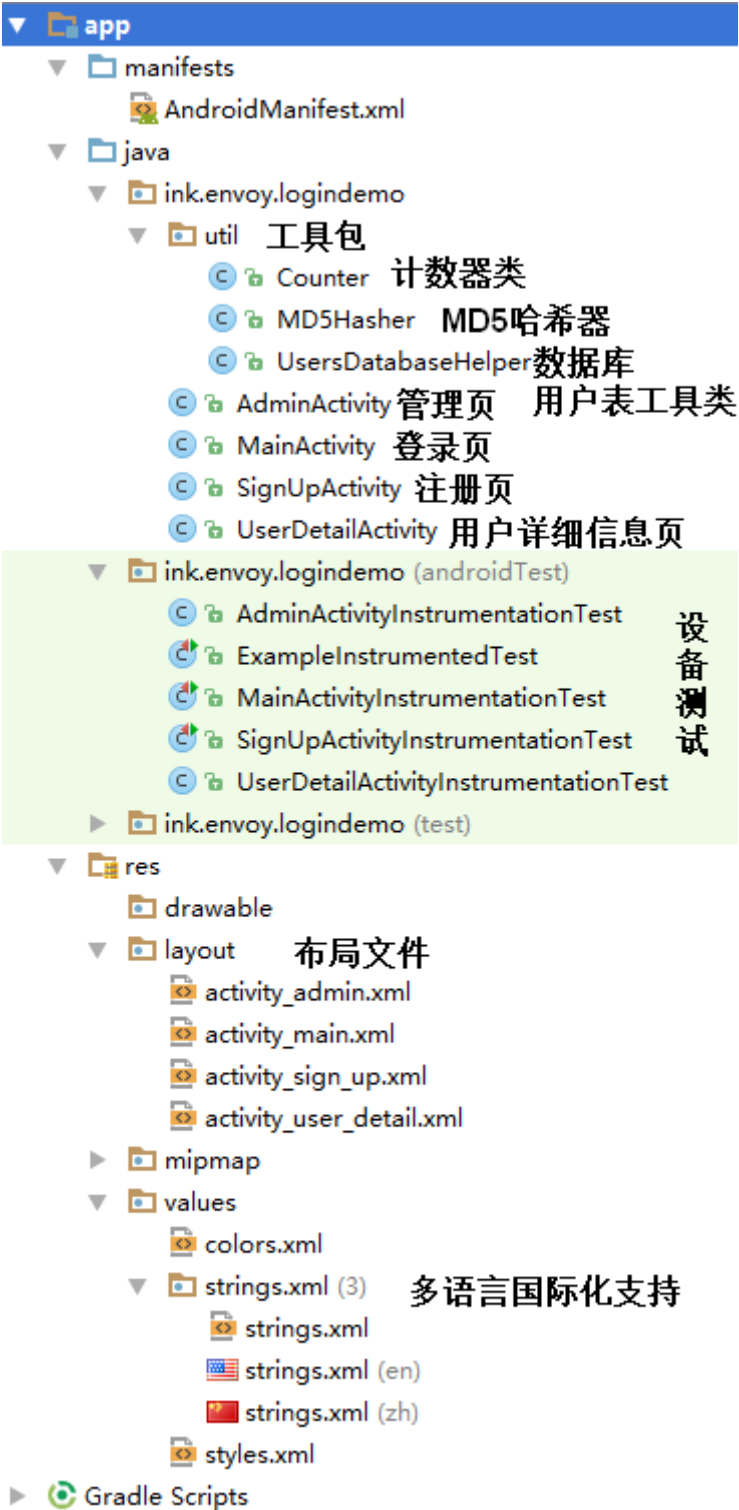


图 4-1 项目文件结构

UI 测试 登录界面 (MainActivity) 测试代码 (此为 18 个测试样例中的 11 个) :

```
package ink.envoy.Logindemo;

import android.support.test.InstrumentationRegistry;
import android.support.test.espresso.action.ViewActions;
import android.support.test.espresso.matcher.ViewMatchers;
import android.support.test.rule.ActivityTestRule;
import android.support.test.runner.AndroidJUnit4;
import android.test.ActivityInstrumentationTestCase2;
import android.test.suitebuilder.annotation.LargeTest;
import android.view.View;

import org.junit.After;
import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

import static android.support.test.espresso.Espresso.onView;
import static android.support.test.espresso.action.ViewActions.click;
import static android.support.test.espresso.action.ViewActions.closeSoftKeyboard;
import static android.support.test.espresso.action.ViewActions.swipeLeft;
import static android.support.test.espresso.action.ViewActions.typeText;
import static android.support.test.espresso.assertion.ViewAssertions.matches;
import static android.support.test.espresso.matcher.ViewMatchers.assertThat;
import static android.support.test.espresso.matcher.ViewMatchers.hasErrorText;
import static android.support.test.espresso.matcher.ViewMatchers.isEnabled;
import static
android.support.test.espresso.matcher.ViewMatchers.withEffectiveVisibility;
import static android.support.test.espresso.matcher.ViewMatchers.withHint;
import static android.support.test.espresso.matcher.ViewMatchers.withId;
import static android.support.test.espresso.matcher.ViewMatchers.withSpinnerText;
import static android.support.test.espresso.matcher.ViewMatchers.withText;

@RunWith(AndroidJUnit4.class)
public class MainActivityInstrumentationTest {

    private static final String ADMIN_USERNAME = "admin";
    private static final String ADMIN_PASSWORD = "admin";
    private static final String TEST_USER_USERNAME = "test";
    private static final String TEST_USER_PASSWORD = "hello_world";
    private static final String NON_EXIST_USERNAME = "hahaha";
```



```

private static final String WHATEVER = "*#@^%!!";
private static final String SHORT_WORD = "aa";
private static final String LONG_WORD = "aaaaaaaaaaaaaaaaaaaaa";

private static final String ERROR_INCORRECT_PASSWORD = "This password is
incorrect";
private static final String ERROR_INVALID_USERNAME_NOT_EXISTS = "Username does
not exist";
private static final String ERROR_INVALID_USERNAME = "This username is invalid";
private static final String ERROR_INVALID_PASSWORD_TOO_SHORT = "This password is
too short";
private static final String ERROR_INVALID_PASSWORD_TOO_LONG = "This password is
too long";
private static final String ERROR_FIELD_REQUIRED = "This field is required";

@Rule
public ActivityTestRule<MainActivity> mActivityRule = new
ActivityTestRule<>(MainActivity.class);

@Test
public void loginWithAdmin(){
    onView(withId(R.id.usernameEditText)).perform(typeText(ADMIN_USERNAME),
closeSoftKeyboard());
    onView(withId(R.id.passwordEditText)).perform(typeText(ADMIN_PASSWORD),
closeSoftKeyboard());
    onView(withId(R.id.signInButton)).perform(click());

onView(withId(R.id.userTableView)).check(matches(withEffectiveVisibility(ViewMatche
r.s.Visibility.VISIBLE)));
}

@Test
public void loginWithTestUser(){
    onView(withId(R.id.usernameEditText)).perform(typeText(TEST_USER_USERNAME),
closeSoftKeyboard());
    onView(withId(R.id.passwordEditText)).perform(typeText(TEST_USER_PASSWORD),
closeSoftKeyboard());
    onView(withId(R.id.signInButton)).perform(click());

onView(withId(R.id.userDetailTableView)).check(matches(withEffectiveVisibility(ViewM
atchers.Visibility.VISIBLE)));
}

@Test

```

```

    public void loginWithEmptyUsername(){
        onView(withId(R.id.passwordEditText)).perform(typeText(WHATEVER),
closeSoftKeyboard());
        onView(withId(R.id.signInButton)).perform(click());

onView(withId(R.id.usernameEditText)).check(matches(hasErrorText(ERROR_FIELD_REQUIRE
D))));
    }

```

```

@Test
    public void loginWithEmptyPassword(){
        onView(withId(R.id.usernameEditText)).perform(typeText(ADMIN_USERNAME),
closeSoftKeyboard());
        onView(withId(R.id.signInButton)).perform(click());

onView(withId(R.id.passwordEditText)).check(matches(hasErrorText(ERROR_FIELD_REQUIRE
D))));
    }

```

```

@Test
    public void loginWithIncorrectPassword(){
        onView(withId(R.id.usernameEditText)).perform(typeText(ADMIN_USERNAME),
closeSoftKeyboard());
        onView(withId(R.id.passwordEditText)).perform(typeText(WHATEVER),
closeSoftKeyboard());
        onView(withId(R.id.signInButton)).perform(click());

onView(withId(R.id.passwordEditText)).check(matches(hasErrorText(ERROR_INCORRECT_PAS
SWORD))));
    }

```

```

@Test
    public void loginWithNonExistUsername(){
        onView(withId(R.id.usernameEditText)).perform(typeText(NON_EXIST_USERNAME),
closeSoftKeyboard());
        onView(withId(R.id.passwordEditText)).perform(typeText(WHATEVER),
closeSoftKeyboard());
        onView(withId(R.id.signInButton)).perform(click());

onView(withId(R.id.usernameEditText)).check(matches(hasErrorText(ERROR_INVALID_USERN
AME_NOT_EXISTS))));
    }

```

```

@Test

```

```

    public void loginWithInvalidUsername() {
        onView(withId(R.id.usernameEditText)).perform(typeText(WHATEVER),
closeSoftKeyboard());
        onView(withId(R.id.passwordEditText)).perform(typeText(WHATEVER),
closeSoftKeyboard());
        onView(withId(R.id.signInButton)).perform(click());

onView(withId(R.id.usernameEditText)).check(matches(hasErrorText(ERROR_INVALID_USERN
AME)));
    }

```

@Test

```

    public void loginWithTooShortPassword() {
        onView(withId(R.id.usernameEditText)).perform(typeText(ADMIN_USERNAME),
closeSoftKeyboard());
        onView(withId(R.id.passwordEditText)).perform(typeText(SHORT_WORD),
closeSoftKeyboard());
        onView(withId(R.id.signInButton)).perform(click());

onView(withId(R.id.passwordEditText)).check(matches(hasErrorText(ERROR_INVALID_PASSW
ORD_TOO_SHORT)));
    }

```

@Test

```

    public void loginWithTooLongPassword() {
        onView(withId(R.id.usernameEditText)).perform(typeText(ADMIN_USERNAME),
closeSoftKeyboard());
        onView(withId(R.id.passwordEditText)).perform(typeText(LONG_WORD),
closeSoftKeyboard());
        onView(withId(R.id.signInButton)).perform(click());

onView(withId(R.id.passwordEditText)).check(matches(hasErrorText(ERROR_INVALID_PASSW
ORD_TOO_LONG)));
    }

```

@Test

```

    public void loginFailsTooManyTimes() {
        onView(withId(R.id.usernameEditText)).perform(typeText(ADMIN_USERNAME),
closeSoftKeyboard());
        for (int i = 0; i < 3; ++i) {
            onView(withId(R.id.passwordEditText)).perform(typeText(WHATEVER),
closeSoftKeyboard());
            onView(withId(R.id.signInButton)).perform(click());

```

```

onView(withId(R.id.passwordEditText)).check(matches(hasErrorText(ERROR_INCORRECT_PASSWORD)));
    }

onView(withId(R.id.hintTextView)).check(matches(withEffectiveVisibility(ViewMatchers
.Visibility.VISIBLE)));
    }

@Test
public void intentToSignUpView() {
    onView(withId(R.id.mainActivityLayout)).perform(swipeLeft());

    onView(withId(R.id.SignUpActivityLayout)).check(matches(withEffectiveVisibility(View
Matchers.Visibility.VISIBLE)));
    }
}

```

图 4-2 UI 测试 登录界面测试用例代码（18 个测试样例中的 11 个）

为了您能更直观地了解本项目的 UI 测试，请查看同目录下的“[登录界面 UI 测试演示图](#)”动画（约 30 秒）

```

package ink.envoy.logindemo.util;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class MD5Hasher {
    public static String hash(String s) {
        char hexDigits[]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};

        try {
            byte[] btInput = s.getBytes();
            MessageDigest mdInst = MessageDigest.getInstance("MD5");
            mdInst.update(btInput);
            byte[] md = mdInst.digest();
            int j = md.length;
            char str[] = new char[j * 2];
            int k = 0;
            for (byte byte0 : md) {
                str[k++] = hexDigits[byte0 >>> 4 & 0xf];
                str[k++] = hexDigits[byte0 & 0xf];
            }
            return new String(str);
        } catch (NoSuchAlgorithmException e) {
            throw new RuntimeException();
        }
    }
}

```

图 4-3 MD5 动态哈希算法

```

private LoginAuthenticationStatus authenticate(String username, String password) {
    SQLiteDatabase db = usersDatabaseHelper.getReadableDatabase();
    Cursor cursor = db.query("users", new String[] { "username", "password", "isAdmin" },
        "username=?", new String[]{ username }, null, null, null);
    try {
        if (cursor.moveToNext()) { // found
            String realPassword = cursor.getString(1);
            boolean isAdmin = cursor.getInt(2) == 1;
            return MD5Hasher.hash(password).equals(realPassword)
                ? (isAdmin
                    ? LoginAuthenticationStatus.SUCCESS_ADMIN
                    : LoginAuthenticationStatus.SUCCESS_NORMAL)
                : LoginAuthenticationStatus.WRONG_PASSWORD;
        } else {
            return LoginAuthenticationStatus.USERNAME_NOT_EXISTS;
        }
    } finally {
        cursor.close();
        db.close();
    }
}

```

图 4-4 能够防范 SQL 注入的用户身份验证逻辑

```

private void loadData() {
    SQLiteDatabase db = usersDatabaseHelper.getReadableDatabase();
    Cursor cursor = db.query(
        "users",
        new String[] { "_id", "username", "password", "isAdmin", "createdAt" },
        "",
        null,
        null,
        null,
        null);
    List<String[]> results = new LinkedList<>();
    while (cursor.moveToNext()) {
        results.add(new String[] {
            cursor.getInt(0) + "", // id
            cursor.getString(1), // username
            cursor.getString(2), // password
            cursor.getInt(3) == 1 ? "√" : "", // isAdmin
            SimpleDateFormat.getInstance().format(new Date(cursor.getLong(4))) // createdAt
        });
    }
    cursor.close();
    db.close();
    userTableView.setHeaderAdapter(new SimpleTableHeaderAdapter(getApplicationContext(),
        "",
        getString(R.string.hint_admin_activity_table_header_username),
        getString(R.string.hint_admin_activity_table_header_password),
        getString(R.string.hint_admin_activity_table_header_isAdmin),
        getString(R.string.hint_admin_activity_table_header_createdAt)));
    userTableView.setDataAdapter(new SimpleTableDataAdapter(getApplicationContext(), results));
    TableColumnWeightModel columnModel = new TableColumnWeightModel(5);
    columnModel.setColumnWeight(0, 2);
    columnModel.setColumnWeight(1, 4);
    columnModel.setColumnWeight(2, 5);
    columnModel.setColumnWeight(3, 3);
    columnModel.setColumnWeight(4, 7);
    userTableView.setColumnModel(columnModel);
}

```

图 4-5 从数据库读取用户信息的逻辑

五、软件操作流程

软件启动后，系统会初始化生成默认管理员账号（用户名和密码都为 admin）。用户可以使用该账号以管理员身份登录，也可以在登录界面[左滑](#)前往[注册](#)页面，在注册页面以普通用户身份注册。

如果用户多次输入错误的密码（超过 3 次），则会被禁止登录，屏幕下方出现 **10 秒的倒数计时**，10 秒后用户可以重新登录。

用户登录后，如果是普通用户身份，则会进入[用户信息](#)界面，用户可以查看自己的个人信息。如果是[管理员](#)身份，则会进入[用户管理](#)界面，管理员可以查看所有用户的信息，并且可以[删除非管理员账户](#)。

六、难点和解决方案

设计用户在登录和注册的界面时，如何防止 SQL 注入是本项目最大的难点。经过思考，可以得出以下方案：先从数据库中获取用户名，然后手动比较用户输入的密码的加盐 MD5 值是否与数据库中存储的相同，并且使用 Google 官方提供的类库，则可完美防止 SQL 注入。具体代码实现请见图 4-4。

七、不足之处和今后的设想

本项目实现了上次项目中所设想的国际化。

本次项目历时较短，未能投入足够多的时间设计和开发 app，有着些许不足之处，只是一个较简单的登录模块，还缺少复杂的用户信息显示与存储功能。但本项目使用了可扩展和易维护的代码架构，如需后期添加功能，能够在短时间内以最少的代码修改量完成任务。

附录

项目依赖于：

Javatuples, 2.1.0

<http://www.javatuples.org/>

SortableTableView, 2.8.0

<https://github.com/ISchwarz23/SortableTableView>

感谢 ISchwarz23 提供技术支持。