# 基于移动终端和二维码的系统登录技术研究

摘要：随着网络技术的普及，人们的生活已经离不开网络了，而在各种软件应用中，少不了要进行个人账户的登录，这就涉及到了个人隐私的安全问题，研究并提出更安全可靠的登录方式成为了需要。本文针对这一问题，提出了基于二维码的登录方式，大致过程就是用户要在网站上进行用户登录操作时，网页上会根据Web服务端的信息生成一个二维码，用户通过移动端，如android手机等打开相应的应用程序进行扫码登录，从而达到快速安全的登录效果。本文将对这一过程的设计和实现进行详细的讲解，并给出相关代码，方便其他软件开发者对自己的软件安全性进行完善，将二维码登录技术快速的集成到已存在的系统中，减少相关投入，实现工业效益的最大化。

关键字：QRcode Android JavaEE

一、引言

当今时代是互联网+的时代，人们生活的各个方面都实现了互联网化，各个软件的使用都离不开注册和登录，这一过程的安全性将直接影响到客户对该软件的信任程度。与此同时，不难发现二维码在我们的生活中也是无处不在的，人们现在使用最多的支付方式无疑就是扫描二维码进行支付，这个过程快捷方便。除此之外，还可以发现现在很多信息都是隐藏在二维码中的，比如街上的宣传海报上都会有一个二维码，有的是扫描加群，有的是扫描转到一个网站，这样会为用户省去很多麻烦，不用手动输入网址，只需拿出手机扫描即可。

在以前许多Web应用程序的登录都是采用输入账号和密码的方式，外加一个随机验证码来排除非人工的登录。可想而知，每次的登录都要输入账号和自己为了防止盗号而想出的各种复杂的密码，这样的登录方式给用户带来了诸多不便，而且安全性也得不到保障。现在先进的几种登录方式有手机短信或邮箱进行验证登录，用户在登录时输入账号，这个账号通常为绑定的手机号或者是电子邮箱，系统在进行账号验证，如果存在该账户则会给该账户绑定的手机号或者是邮箱发送一个随机字符的验证码，再由用户输入登录。除了邮箱短信验证，另外一种现在常用的登录方式就是本文所说的扫描二维码进行登录的方式，本文提出了通过移动终端加上二维码的方式进行登录的方案，旨在为用户带来方便安全快捷的登录体验。

二维码技术早在上个世纪九十年代就已经产生了，但是近几年随着移动终端的迅速普及才开始用到这个技术。二维码的编码方式是对外开放的，研究者可以根据相关协议对二维码进行快速的解析，得到编码前的内容。在本文的设计方案中，二维码在Web服务器中进行生成，然后用户通过android手机进行二维码的扫描和解析，并进行相关登录信息的提交到服务器，再由服务器提交到数据库。

本文将先后进行设计整体说明，对系统的结构以及验证过程进行阐述；分析二维码技术发展，以及本文用到的二维码开源框架ZXing；对比Android系统和IOS系统进行比较，对Android系统做大致的介绍，以及逐步介绍移动端功能的实现；对系统后台的功能和设计进行详细说明，介绍javaEE中servlet的用法，javaScript脚本的使用，jdbc的使用，以及数据库的设计和使用。

二、基于移动终端登录系统的整体设计

2.1、系统结构

二维码扫描登录系统的整体结构如图1所示，分为四个部分，即浏览器、Web系统、移动端登录软件、Web系统数据库。

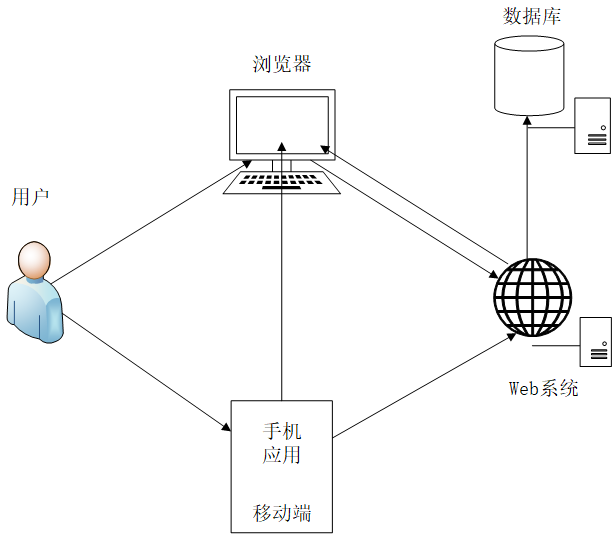


图1 基于移动终端登录系统的结构图

该图中，由于本设计针对的是Web系统，所以浏览器就是应用系统的客户端，而Web系统依托于Tomcat服务器存在，为用户提供设计好的服务，通过互联网进行信息传递，本文只针对其中的登录功能进行设计。图中的手机应用为用户登录时需要下载安装的应用，而数据库存储了用户的信息和登录相关的数据。

该系统的验证过程可以分为以下几步：

（1）用户打开计算器上的浏览器访问该Web系统，如果是第一次访问，Web系统还没有记下登录状态，则需要用户进行登录操作。Web系统会在网页上输出一个二维码图片供用户扫描。而二维码的内容则是本次会话的标识数据，在本设计中就是一个该Web系统的网址链接上本次访问的session id。

（2）当网站上出现二维码后，用户可以打开手机上进行登录的应用进行登录操作。该应用需要用户采用传统的账号密码进行登录，登录成功后才能使用扫描二维码的功能。该应用扫描二维码后会根据和Web系统协商好的数据格式对二维码中的数据进行解析，验证该二维码是否为本Web系统登录使用的二维码，若是，则将二维码中的session id包含在Web系统的登录请求中发送出去。

（3）Web系统中扫描二维码登录的网页会不断重复的给Web系统发送查询登录状态的请求，若移动端已经将登录请求发送成功，则当Web系统查询到该用户已经跳转都已登录状态，则页面会跳转为登录成功后的页面。

2.2、系统主要功能的实现

2.2.1系统整体流程图

分析上文系统进行的登录验证的过程，可以将整个系统的功能大致分为以下三个部分，即登录信息的传递、移动终端的登录、登录状态的转变。方案流程图如图2所示。

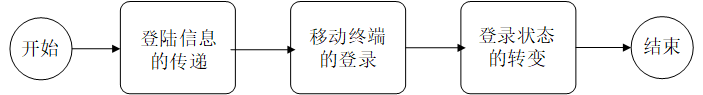


图2 方案整体流程图

2.2.2 登录信息的传递

在整个登录过程中，二维码的功能尤为重要，它使得登录信息能够存储其中并不能被其他系统所用。整个系统把二维码的生成和识别分为两个模块，二维码的生成在Web系统中完成，二维码的扫描在手机应用上完成。生成和扫描的二维码中的数据格式需要协商统一，才能正确的识别信息。

二维码作为信息传递的关键要素，其中存储了两种数据，一个则是登录所需的地址，用来验证是否为此系统生成的二维码，用以确定后续是否向系统发送登录请求。另外一个则是会话标识符，即Web系统中的session id，这一数据是整个系统中最为重要的数据，其能够让Web系统知道是哪个浏览器进行了登录操作，是将使用手机的用户和浏览器用户联系起来的关键数据。以上两种数据采用“，”进行链接。后续移动终端扫描到信息后也将按照此规则解析出登录地址和会话标识符，并验证登录地址后再向系统提交会话标识符，否则就终止登录过程。

2.2.3 移动终端的登录

用户在使用移动端的应用时，同样需要进行登录操作，可以看做是把Web端的登录转移到移动端的登录上，简化Web端的登录操作。用户在移动端登录成功后，来到个人信息的界面，并可以进行二维码扫描，扫描成功后根据信息将账户号以及会话标识符再提交到Web服务器上。

2.2.4 登录状态的转变

Web服务器接收到移动端发送来的账户号和会话标识符将查询数据库将会话标识符即session id保存到指定账户号对应的用户表记录中，并将该条记录中的登录状态转变为已登录状态。此时进行登录的网页将在后台不断地根据本地的session id查询数据库中是否有该session id，并判断其登录状态，若都符合则登录成功，页面进行跳转。除此之外，登录成功后还要设计一个用户注销的按钮，来注销数据库中的用户登录状态。

三、二维码技术分析

3.1、二维码技术的发展

3.2、二维码开源框架——ZXing

四、移动端的设计方案

4.1、Android和IOS系统的比较

现如今主流的移动终端的操作系统有Android和IOS两种，发展到现在已经对于哪个系统更加优秀已经有很大的争议了，各自有自己的长处和优势。可以说IOS开创了智能手机的先河，由于其当时非同凡响的操作体验，和多种多样炫酷实用的应用，将传统手机市场打破。而当时在手机行业称霸的诺基亚公司现如今已经被微软公司收购，摩托罗拉公司也被谷歌公司收购了。IOS打开了智能手机的大门，而Android系统则不断的推进智能手机的发展，使其壮大。由于Android系统的开源特性，各个手机厂商都能够进行Android手机的开发，这样一来整个智能移动终端的行业也就得到了飞速发展。

在IOS系统发布的初期，其功能是不够完善的，甚至是许多在传统手机上很容易实现的功能，IOS也无法完成。如IOS在早期4.0的版本是做不到更换手机壁纸的，而这样的功能对于传统手机来说是非常容易实现的。除此之外，IOS系统还缺少查看手机来电归属地的功能，只能通过手机越狱来获得，而这样无疑会给手机带来安全性的问题。除了这些缺点，IOS的优点是非常明显的，用过IOS系统的人都知道，IOS最明显的一个有点就是流畅，这是由其系统架构导致的，IOS系统对屏幕反应的优先级是最高的，其作出响应的顺序是Touch—Media—Service—Core这样一种架构。当用户触摸屏幕后，系统会最优先的去处理这一事件，再后面才是媒体、服务以及核心架构，带给用户非常好的交互体验。除此之外，IOS系统还带来了全新的设计风格，带给用户全新的视觉体验，传统的UI设计都是依照尽量模仿实际物体设计的，而IOS采用了扁平化的设计风格，达到了简单、精美的效果。

Android系统在早期被人们认为是模仿着IOS系统来做的，这也很难否认，毕竟Android系统在早期确实许多功能和设计模式都跟IOS系统大致相同。但是经过不断的发展，现如今，Android系统已经独树一帜，有许多自己创新的技术。Android系统的发展是非常迅速的，这一直接原因就是其开源特性。Android的源代码是开放的，任何人都可以去Github上查看，并对其进行定制，加入全新的组件和布局。因此许多手机厂商都推出了自家的Android系统。虽然开源的特性让Android系统多样化，但是这是把双刃剑，开源的特性也使得各个手机厂商设计出来的手机没有统一的规格，直接导致了各种应用的兼容性问题。如屏幕尺寸不同导致应用中的图片显示存在差异，从而影响外观效果。而且手机硬件配置的不同也使得Android应用开发者在进行应用开发时要考虑那些低端配置手机是否能够流畅运行自己的应用。这也在一定程度上抑制了其发展。

4.2、Android系统体系结构

和Linux、IOS等操作系统一样，Android系统在设计时也将系统分为了几层架构。如图3Android系统的架构图所示，可以将Android系统分为四个层次，其中，最高层为应用程序层，其次是框架层，再其次是系统运行库层，最后就是底层的linux核心层。

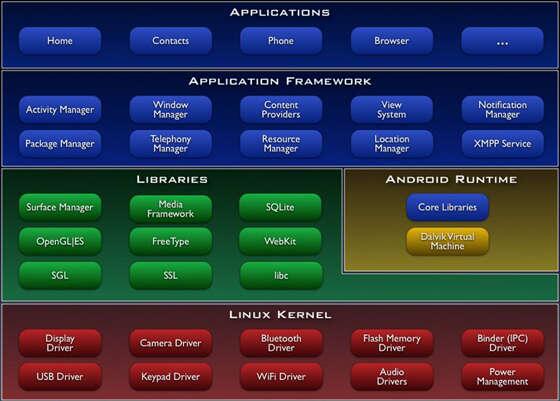


图3 Android系统架构图

4.2.1 应用程序层

这一层是直接和用户打交道的架构层，其包含了手机的基本功能的应用，如电话、邮件、短信、地图、浏览器等Android系统的基本应用，为用户提供最基础的服务。除此之外还使得用户能够安装第三方的应用软件，对Android系统的功能进行扩展，丰富用户体验。这些应用程序早期都是由Java语言编写的，现在越来越多的开发者选用Kotlin语言来进行更快捷安全的开发。

4.2.2 应用框架层

该层提供了Android开发所需要的组件,开发者根据应用程序的需要可以随时调用其API达到快速开发的效果，并且这种架构使得Android开发可以进行划分。一个应用程序就是由开发者对其一个或多个框架进行开发得到的。其中Activity Manager负责管理Activity的生命周期，而Activity可以看做是Android应用的一个界面。View System为Android开发提供了各种控件，和开发windows程序一样，Android开发也是需要大量的控件来完成一个界面的布局，

这些控件最基本的有TextView、Button、ImageView等，其效果分别为文本框、按钮、图片框等。使得Android界面可以显示图片文字等资源，以及响应用户的点击事件，高级的控件还有如TabLayout、ListView等，可以实现选项卡、列表等高级布局，实现更人性化的操作。此外，Notification Manager框架使得应用程序能够向手机的通知栏添加自身的提示信息，Content Providers框架使得某个应用程序能够查询或使用其他应用程序的数据。

4.2.3 系统运行库层

（1）左边绿色部分为Android底层的C/C++库，封装了系统对底层资源的操作。如SGL库为一个底层的2D图形引擎，OpenGL|ES使得Android能够对三维图像进行处理，这个功能在Android游戏开发中用的多。SQLite是一个轻型关系数据库，能够快速进行数据库的创建以及更删改查操作，这个库使得Android能够方便的进行数据存储。Medie Framework库使Android系统能够识别别播放各种多媒体文件，如MPEG4、MP3、JPG等多媒体资源格式。WebKit库为一个web浏览器引擎，使得Android手机实现了浏览器功能，和能够进行web浏览的视图资源。

（2）右边部分主要为Android系统运行在linux系统上提供了可能，Android系统并不是直接存在于linux系统中的，Android系统每个应用的运行都是在Dalvik虚拟机的一个实例上的，而Dalvik虚拟机则相当于linux系统上的一个应用。这种设计使得Android系统的应用能够多线程的并行运行。

4.2.4 linux核心框架

Android系统是运行在linux 2.6内核上的，linux也是开源的一种操作系统，通过对其进行裁剪使得其在满足android系统的核心需求外，能够达到低能耗的要求。Linux内核提供了内存管理、进程管理、驱动管理的功能，是Android系统最底层的框架，直接对硬件进行操作。

4.3、移动端具体功能的实现

根据第二章中系统结构和验证过程进行分析，此Android应用需要三个界面：用户登录界面、注册界面、个人信息及二维码扫描界面。因此创建了三个对应的Activity以及对应的layout布局：MainActivity、RegisterActivity、LoginSuccessActivity。

4.3.1 登录界面

登录界面的布局如图4所示，具体代码见附录I。界面采用线性布局，两个EditView供用户填写用户名和密码，两个按钮监听用户点击事件。



图4 Android登录界面布局

在进行Android开发时，需要有一个Activity的子类，和对应的xml布局文件来完成某个界面的功能。而Activity有其自己的生命周期，在打开应用时，系统会找到该应用的MainActivity作为程序的入口，MainActivity被创建后会进行到onCreate方法中，在onCreate方法中执行布局文件的关联，以及对布局文件中的资源进行初始化。Android提供了findViewById这一方法使布局文件中的控件和java引用对象相绑定。在登录界面中，还需要给登录和注册两个按钮设置点击事件，调用button对象的setOnClickListener方法可以为其设置一个监听事件，该方法需要传入一个View.OnClickListener的实现类，并重写其中的onClick方法来响应该点击事件。

在登录界面中，登录按钮被点击后，需要判断用户名和密码两个EditView中的值是否为空。如果任意一个为空则通过Toast的形式提示用户输入账号和密码。若不为空则进行登录操作。实现登录操作需要Android网络编程相关的知识。本Android应用通过http协议与Web服务端进行通信，再由Web服务器操作数据库并返回查询结果给Android应用，整个过程Android应用不直接操作远程数据库，从而减小了Android应用的资源开销。Android中进行http通讯可以使用HttpURLConnection类，在实现登录逻辑时，需要获得账户名和密码中的内容，并将其添加到GET请求中，所以需要创建一个URL类，在进行实例化时需要传入一个网址作为参数。该网址即是Web系统的网址，在网址后面加上“?account=用户名&password=密码”就将用户名和密码封装到了GET请求中，通过URL的openConnection方法可以获得一个HttpURLConnection的实例，通过HttpURLConnection的setRequestMethod(“GET”)方法可以将此次请求设置为GET方式。将参数设置好后，调用HttpURLConnection的getInputStream方法能够获得Web系统返回数据的字节输入流，并将请求发送给Web服务器。移动端发送请求后，就等待接收Web系统返回的数据，所以Web端和移动端需要协商返回的数据格式。本设计中，Web端在接收到Android应用的登录请求后，会返回一个结果码以及用户信息，两者用“&”连接，其中登录成功会返回字符串“100”，登录失败会返回“110”，若登录失败则用户信息为空。

为了方便数据的传输和解析，本设计将传输的信息封装为Json格式，通过Gson框架来进行实际操作。Json是一种轻量级的数据交换格式，它不是一种程序语言，只是一种数据的约定，表现为字符串格式，在Json的语法中，对象以键值对的形式出现，数据之间通过逗号来进行分割，一对花括号表示一个对象，方括号中存放的是数组。Gson是java的一个jar包，通过Gson我们可以将一个java对象转化成json字符串，同样也可以将json字符串还原成java对象。本设计中将返回码ResCode和用户信息User都封装成类，ResCode类中只有一个属性即返回码，而User类包含了account,password,nickname,age等属性，除了属性外还要添加各个属性的Getter和Setter方法，使其成为一个POJO类。在接收到Web系统返回的json字符串后只需转化成相应的类，然后调用其get方法就能够获得需要的数据。

首先将解析出来的返回码resCode进行比对，如果是登录失败，则通过Toast的形式打印出登录失败的字样。如果登录成功，则需要实例化一个意图类即Intent，来完成Activity的跳转，同时还需要将用户信息添加到Intent对象中传到下一个Activity中。至此Android应用的登录功能已经完成。

4.3.2 注册界面

当用户还没注册账号在登录界面点击注册按钮后，就跳转到RegisterActivity界面中，注册界面的布局如图5所示。布局中有输入用户名、密码和个人信息等编辑框，以及两个按钮。布局文件activity\_register.xml见附录III，RegisterActivity.java见附录IV。



图5 Android应用注册界面

在注册界面中初始化视图资源，为按钮添加监听事件后，当用户点击注册按钮后，先判断信息是否填写完全，如果编辑框中有任何一栏没填写信息，则通过Toast通知用户将信息填写完整，如果信息填写完整，则检查两次输入的密码是否一致，如果不一致则通知用户重新填写。当以上都正确时，将用户信息封装成json字符串，通过POST请求将json字符串发送到Web系统中，具体方法是将

五、后台服务器的设计方案

1、后台开发语言的选择

2、java EE的简介

3、servlet的功能和用法

4、java script的功能和用法

5、后台服务器的具体设计思路