# 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。

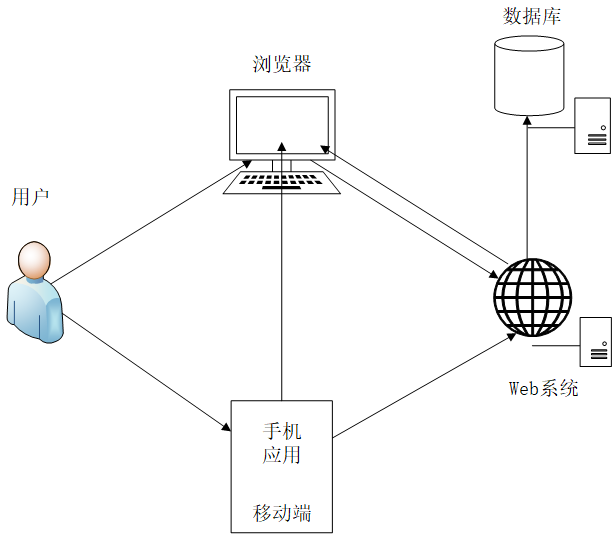


图1 基于移动终端登录系统的结构图

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。

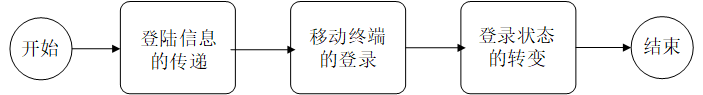


图2 方案整体流程图

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、QR二维码概述

QR二维码又称快速响应矩阵码，是一种矩阵式二维码，它在二维方向上都存贮了信息。QR二维码具有信息容量大、编码范围广、错误修正水平强、全方位识别、可加密等特点，这使得QR码在东亚以及欧美快速发展，直至目前QR码已经是二维码家族中最为流行的一员。QR码的基本结构如下图9所示：

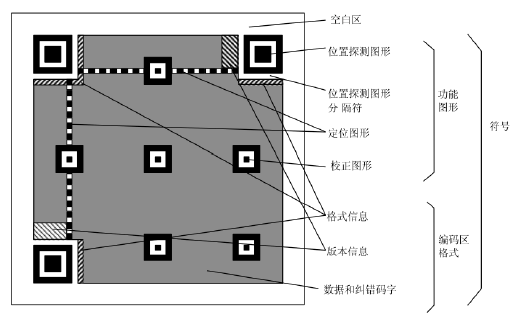


图9 QR码基本结构图

⚫位置探测图形

在二维码图形的右上区域、左上区域和左下区域各有一个“回”字型的的图形符号，我们称之为位置探测图形。在最外层的黑色正方形模块大小为7x7，中间的白色正方形模块大小为5x5，最里层的黑色正方形模块大小为3x3，因此模块宽度比为1：1：3：1：1，以黑白间隔排列。在二维码的其他区域几乎不可能找到这样比列构成的模块，因此当识别到该类型时便可确定是位置探测符，即可确定二维码的边界。

⚫位置探测图形分隔符

在位置探测图像周围一个单位宽的白色条状区域为探测图像分隔符，用来把位置探测图形和编码区域分隔开

⚫定位图形

在两个位置探测图形之间各有一条一个模块宽的黑白相间的条状区域称之为定位图形。在水平方向和垂直方向各有一定位图形，用来确定二维码图形中像素块的坐标。

⚫校正图形

校正图形可看成由5x5大小的黑色模块、3x3白色模块及中间的一单位宽黑色模块组成的重叠同心正方形构成，数量随版本变化，版本越高数量越多。校正图形主要作用是对不规则的图形失真进行校正。

⚫编码区域

编码区域包括格式信息、版本信息、数字码字和纠错码字四个部分。表示的数据大小跟版本号有关。

⚫格式信息

格式信息包含纠错水平等级和掩模图形等信息，会出现两次，目的是提供冗余信息。5个数据位和10个纠错位构成15位的格式信息。

⚫版本信息

版本信息分布在条码的两个特定位置，在译码时非常重要，故会出现两次来提供冗余的信息。6个数据位和12个纠错位构成了18位的版本信息。纠错位可通过BCH（18，6）得到。

⚫空白区域

空白区域由环绕在二维码周围的4单位换的白色区域构成。

3.2、二维码开源框架——ZXing

ZXing是谷歌一个开放源码的，用Java实现的多种格式的1D/2D条码图像处理库，它包含了联系到其他语言的端口。Zxing可以实现使用手机的内置的摄像头完成条形码的扫描及解码。ZXing支持的二维码标准主要有QR Code、Data Matrix PDF417等码制，同时ZXing库提供了android、c#、java等多种平台的类库。在将Zxing导入到JAVA项目工程之前应该先去下载ZXing源码，不过用maven自动从网络上下载需要用到的包。下载解压后得到ZXing的源码结构如图10所示：



图10 ZXing源码目录

下面介绍文件部分文件夹的部分内容：

core 文件夹：这是核心包，打开该文件夹里面有个预编译好了的 core.jar 包，该包的重点内容是各种码制的编码和解码程序，是本项目所做的 QR 二维码的生成与解码的基础程序。

Android 文件夹：存放的是基于 ZXing 核心构建的 Android 应用，也是本设计中实现Android部分扫码功能的核心内容，需要对官方的Android代码进行裁剪并当作第三方库导入到自己的Android工程中。

Javase 文件夹：用 java SE 构建的图形用户界面例子，主要是把 Bit Matrix 图像转换成需要的图片格式。

Android- integration 文件夹：封装简化了调用 Barcode Scanner以及扫描识别后返回的结果。

Google ZXing库是一个大类库，它包含了各种各样的功能，我们在做基于JAVA语言QR二维码生成与识别系统时只需用到生成和识别功能即可。主要是用到zxing\core\src\main\java\com\google\zxing\qrcode目录下的代码，如下图11所示。

encoder包下是编码类二维码，实现二维码的生成。decoder包下是解码相关代码。

QRcode类为编译码接口，外部功能通过调用QRCodeReader和QRCodeWriter类实现。

Detector是摄像头抓取图像时检测到QR码，寻找校正符、定位符。之后把QR码符号部分提取出来，是能否实现正确解码的关键。Decoder类是依照解码规范把之前检测到的QR码符号部分进行解码，将图像解析为真实的信息。ByteMatrix.类负责图像的二值化，其中包括自适应阔值算法代码。将二值化后的矩阵传递给Detector，其中Detect方法是相应的接口，调用该接口就可以取得符号码矩阵。

FinderPatternFinder类实现寻找位置探测图形，在图像中隔一定步长采样一行。在该采样行把连续相同颜色的像素个数计入数组，数组的长度是5位，即找黑-白-黒-白-黑的的数组。填满迭5位数组元素后，看比例是否满足允许有1/2的误差。

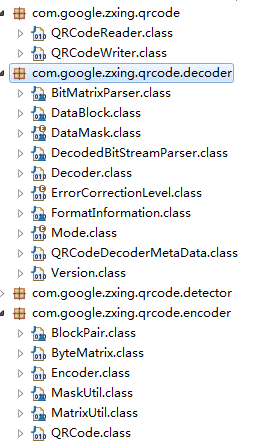


图11 QR码相关类目录

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。

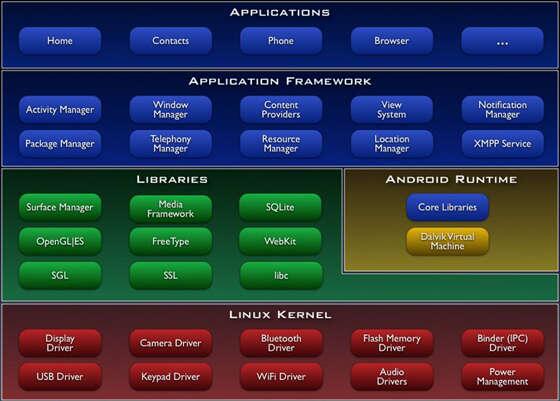


图3 Android系统架构图

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，具体方法是通过HttpURLConnection的对象获得其字节输出流，并将其包装成DataOutputStream对象，通过writeBytes()方法将json字符串发送出去，当然此时还只是存在缓冲区中，当调用HttpURLConnection的getInputStream方法后，数据就真正发出去了，并等待服务器返回执行结果。若返回结果为“210”即静态类Constant中的Constant.REGISTER\_FAIL常量，则表示注册失败，原因可能是Web端不能正常对数据库进行操作；若返回结果为“200”即Constant.EXIST\_THE\_ACCOUNT常量，则表示用户名已被注册过；若返回结果为“300”即Constant.REGISTER\_SUCCESS常量，则表示注册成功，并弹出对话框提示用户已经注册成功，用户点击确定按钮后调用Activity的finish()方法结束当前Activity返回到登录界面中。

4.3.3 扫码界面

当在登录界面进行登录并成功后，会跳转到扫码界面，扫码界面的布局如图6所示，该布局中有一个文本框用来显示用户的信息，也确保用户登录信息的获取是完整的。除了显示用户信息的文本框，还有两个按钮，一个用于调出实际用于扫描二维码的界面，另外一个用来退出当前登录。扫码界面的布局文件activity\_login\_success.xml见附录V，LoginSuccessActivity.java见附录VI。



图6 Android应用扫描界面

在LoginSuccessActivity中，调用Activity的getIntent()方法可以得到在登录界面传来的用户信息的数据。由于封装数据用intent.putExtra(String name, String value)方法存储的键值对，所以在取值的时候需要调用intent.getStringExtra(String name)，通过传入键名来获得对应的值，获取到账户名、昵称、年龄等用户信息后再文本框中显示出来。当用户点击扫码按钮后，通过显性意图的方式来启动一个Activity，与上文中的startActivity方式不同，这里需要调用intent的startActivityForResult方法可以获得下一个页面回传过来的数据。当下一个Activity调用finish方法来结束自身的时候，会执行onActivityResult方法将二维码扫描的结果通过Intent对象回传给上一个Activity。当LoginSuccessActivity得到正确的二维码扫描结果后，要首先对其内容进行判断，看是否是Web系统产生的二维码。在框架设计中提到过，Web系统产生的二维码是由Web系统的网址和session id通过“，”组成的。所以需要调用String方法的split(String regex)方法来进行字符串分割，这里将“，”作为regex的实参，讲二维码结果分割成两部分result[0]和result[1]。首先比较result[0]中的内容是否是Web系统的网址，如果不是，则将二维码的内容通过Toast打印输出，若是，则result[1]中存放的就是session id，此时需要将当前用户的用户名和session id通过GET请求的方式发送到Web系统进行处理。当Web系统将session id保存到数据库中该用户名的一条记录中后，整个Android端的任务就完成了。

五、后台服务器的设计方案

5.1、后台开发语言的选择

进行Web开发前需要了解集中Web后台的开发语言，现在比较热门的后台开发语言有PHP、RUBY以及JAVA等。RUBY语言在网上的评价很高，其语言优雅、开发便捷，但并没有成为主流的后台开发语言。现在比较主流的后台开发语言还是PHP和JAVA，而PHP和JAVA在设计理念上存在着很大的差异，这导致两种语言在进行开发时会得到不同的结果。JAVA是一个非常完善的程序设计语言，它在访问数据库以及进行数学计算方面有先天的优势，JAVA WEB的开发有自己的一套流程，在开发时会用到SSH框架，SSH框架指的是Struts框架、Spring框架、Hibernate框架。其中Struts框架用来管理项目用以实现MVC的设计模式，将视图、逻辑实现分离，降低代码耦合度。Spring框架用于业务逻辑层。Hibernate框架能够使得开发者方便快捷的进行数据持久化操作，即将内存中的数据存储到数据库中。反观PHP语言，其设计理念为面向过程的，并以客户作为开发的核心，能够进行快速的进行Web开发，且现在市面上的许多虚拟主机都是以支持PHP为主，这就使得PHP的开发要容易的多。但是PHP在进行大型的网站开发时，面对极其复杂的业务逻辑，非常容易出错。所以在进行大的项目开发时还是会使用到JAVA语言。由于Android开发中也使用的JAVA语言，所以本设计也将采用JAVA语言进行Web后台的设计。

5.2、J2EE的简介

J2EE 技术，即Java 2平台企业版（Java 2 Platform Enterprise Edition），是一种B/S模式的体系结构，是利用 Java技术简化企业应用系统开发、管理和部署等复杂问题的一套完整的体系结构。该技术由SUN公司推出，基于Java 2平台，它继承了Java语言方便、灵活、跨平台等优势，还提供了对EJB(Enterprise Java Bean)、Java Servlets、JSP（Java Server Pagers）、XML 、API等技术的支持，从而即降低了开发多层应用系统的难度和复杂性，又提供了对现有应用系统的强大支持。

J2EE技术一般分为四个层级：客户层、WEB表示层、业务逻辑层、数据层（EIS）。如下图7所示。

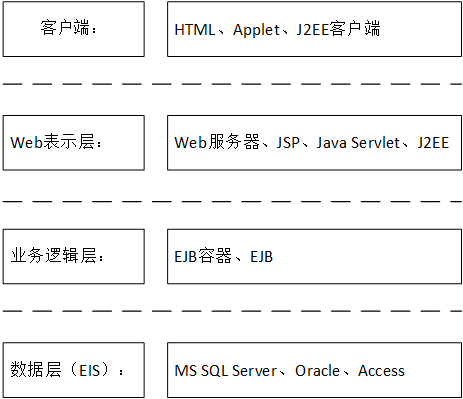


图7 J2EE层次结构图

客户层：可以包括瘦客户端、胖客户端，用户可以与该层进行交互作用，是系统的用户界面，可以呈现出适当的视图，收集查询，显示最终结果，是用户在本地所使用的最后的模型成品。

Web表示层：接收客户层组件的要求，处理请求，然后将请求路由到业务层的适当组件；接受业务层传來的结果，计算一个适当的视图，然后将视图路由到对应客户端。

业务逻辑层：这一层由若干运行在EJB容器下的EJB组件和应用服务器组成。EJB封装了业务逻辑，应用服务器提供了JDBC技术，来访问数据库。

Web表示层和业务逻辑层是给 J2EE 发布工作任务的层次，主要是通过服务器完成工作。这两层是 J2EE 的核心层次。

数据层（EIS）: 将前端业务逻辑层的组件与后端数据库层连接起来，收集好的信息放在该层，供用户访问。

J2EE优势：

1.跨平台性。Java具有平台无关性，可以运行在Windows、Linux、Unix等不同的操作系统上，因此，基于J2EE技术开发的应用程序不受特定的硬件、操作系统、服务器等外部环境的约束，可以实现异构环境下应用系统的开发、部署和维护。

2.稳定性。基于J2EE技术的应用系统拥有较强的可靠性，可以满足用户24\*365 的不间断使用，从而满足用户所需的实时性和稳定性要求。

3.开发效率高。J2EE 技术通过将通用的、复杂的、繁琐的服务端系统安排给中间件供应商完成，从而使普通系统开发人员设计并实施普通的商业逻辑部分，有效地降低了系统开发难度，缩短了开发周期。

4.具有可伸缩性。基于J2EE 技术的应用系统在开发完毕后，可以根据需要部署在各种硬件服务器和操作系统之上，从普通的PC机到高端服务器，甚至是大型机系统中，并且可以使用灵活的负载平衡策略以实现伸缩性。

5.3、MySQL数据库介绍

数据库是数据和数据库对象的集合，其中数据库对象指表（Table）、视图（View）、存储过程（Stored Procedure）和触发器（Trigger）等。数据库通过SQL（Structured Query Language）来对数据进行操作和管理，这里包括一些基本的操作如select、delete、insert、update语句。

ADO（ActiveX Data Objects）是微软开发数据库应用程序的数据库访问技术。它被设计用来同新的数据库访问层OLE DB Provider一起协同工作，以提供通用数据访问（Universal Data MySQL）。OLE DB是一个底层的数据库访问接口，用它可以访问各种数据源，包括传统的关系数据库。ADO封装了OLE DB程序中使用的大量COM接口，所以是一种高层访问技术。关于数据库管理系统和接口的原理如图8所示。

MySQL是一个快速、多线程、多用户的小型关系型数据库管理系统。它支持正规的SQL查询语言和采用多种数据类型，能对数据进行各种详细的查询等。MySQL提供了一全套的数据库创建和访问机制，通过很直观的方式就可以创建、访问、修改数据库的表和项，并且能建立它们之间的各种数据关系。MySQL系列从90年代就发展起来，经过了多年的反展，它已经成为非常成熟的技术，面向中小型企业级应用。

MySQL数据库的特点是数据库文件小而简单，不需要运行或者启动数据库服务进程就可以使用。通常MySQL数据库文件可以随网页文件一起方便地放在网站的目录中，正是由于这些特点，MySQL数据库通常被用来作为网站开发的数据库支持技术。

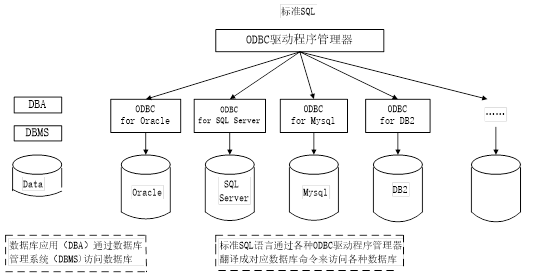


图8 数据库管理系统和接口的原理

5.4、后台服务器的具体设计思路