

## 中文摘要

二维码生成的开发研究主要是为了完善手机二维码的功能，让用户可以随时随地地生成二维码。手机上如果同时拥有二维码扫描和生成两个软件，那么手机用户彼此之间也可以用二维码进行通信，比如，短信的内容以二维码图片的方式发送。

本文在分析讨论 Android 手机软件开发技术原理的基础上，开发出基于 Android 平台的二维码生成软件。此软件可以把输入的字符串生成二维码，还能够将其保存成图片，图片的名字是用输入的字符串命名的，而且本软件还提供一些针对图片的操作，其中包括上下键查找功能，查看图片详细信息功能，搜索功能，删除功能。在这些功能下用户既可以生成二维码，也可以对生成的二维码图片进行相应的操作。

**关键字：**Android；二维码；图片

# Two-dimensional Code Generation of Android

Author:Li Tong Tutor:Li Fuping

## Abstract

The research and development of two-dimensional code generation is to improve mobile phone two-dimensional code function. In another words, it can help users generate two-dimensional code whenever and wherever. If the mobile phone has software which have functions of scanning and generating two-dimensional, the users can communicate with each other through two-dimensional code. For example, users can use two-dimensional codes to send SMS.

This essay is to analyze and discuss Android mobile phone software development technology. Meanwhile, based on the principle of the development technology, the author will try to develop two-dimensional code generation software in the basis of the Android platform. This software can input string to generate two-dimensional code and also can save it into pictures which are the name of the input string. Moreover, the software also provides some image manipulation that includes up down key searching function, viewing detailed information function, searching function and deleting function. In these functions, users can not only generate two-dimensional code but also can generate the corresponding operation for two-dimensional code.

**Keywords:** Android; two-dimensional code; picture

## 目 录

第一章 系统概述.....	1
1.1 项目开发的背景和意义 .....	1
1.1.1 项目开发的背景.....	1
1.1.2 项目开发的意义.....	3
1.2 项目开发的环境 .....	4
1.2.1 硬件环境.....	4
1.2.2 软件环境.....	4
1.3 开发技术的介绍 .....	4
1.3.1 Android 技术介绍.....	4
1.3.2 Android 系统构架介绍.....	4
1.3.3 Eclipse 简介 .....	8
1.3.4 ADT、 Android SDK 简介.....	8
第二章 可行性分析.....	9
2.1 编写目的 .....	9
2.2 可行性研究任务 .....	9
2.2.1 技术可行性 .....	9
2.2.2 经济可行性 .....	9
2.2.3 操作可行性 .....	10
2.3 可行性研究结论 .....	10
第三章 需求分析.....	11
3.1 编写目的 .....	11
3.2 功能需求分析 .....	11
3.3 数据流图 (DFD) .....	12
3.3.1 顶层数据流图 .....	12
3.3.2 一层数据流图 .....	12
3.3.3 二层数据流图 .....	13

3.3.4 三层数据流图 .....	13
3.4 数据字典 (DD) .....	14
第四章 总体设计.....	16
4.1 编写目的 .....	16
4.2 系统功能图 .....	16
4.3 系统流程图 .....	17
4.3.1 总系统流程图.....	17
4.3.2 生成二维码并保存部分流程图.....	18
4.3.3 显示图片部分流程图.....	19
4.4 数据库 .....	20
4.4.1 sdcard 卡对应的数据库.....	20
4.4.2 更新数据库说明 .....	22
第五章 软件设计.....	23
5.1 用户界面设计 .....	23
5.2 文件介绍 .....	23
5.2.1 资源文件中的代码.....	25
5.3 实现功能的代码介绍 .....	29
5.3.1 负责生成二维码的 CaptureZxingActivity.....	29
5.3.2 显示图片的 lookpictureAvtivity.....	36
5.3.3 CaptureZxing Manifest.xml 文件介绍.....	41
第六章 软件测试.....	42
6.1 生成二维码功能测试.....	42
6.1.1 图标.....	42
6.1.2 生成并保存功能 .....	42
6.2 显示图片功能测试.....	44
6.2.1 “下一张”按钮测试.....	44
6.2.2 “上一张”按钮测试.....	44

6.2.3 搜索功能和“图片详情”按钮验证 .....	45
6.2.4 删除功能测试 .....	46
结束语 .....	47
致谢 .....	48
参考文献 .....	49
附录 I 英文资料翻译 .....	50

# 第一章 系统概述

## 1.1 项目开发的背景和意义

### 1.1.1 项目开发的背景

二维条码 (2-dimensional bar code) 是在水平和垂直方向的二维空间存储信息。二维条码具有信息容量大、安全性强、保密性高 (可加密)、识别率高、编码范围广等特点。同一维条码相比, 二维条码也有一些缺点, 如要有专门的生成程序, 识读设备价格比较昂贵, 对于在线扫描, 即先有码后赋值的模式, 不能发挥其特点。

二维条码也有许多不同的编码方法或称码制。就这些码制的编码原理而言, 通常可分为以下两种类型:

线性堆叠式 (或称层排式) 二维码 (stacked bar code): 是在一维条码编码原理的基础上, 将多个一维码在纵向堆叠而产生的。在编码设计、校验原理、识读方式等方面继承了一维条码的特点, 识读设备与条码印刷与一维条码技术兼容, 这类二维条码有 CODE49、PDF417、CODE16K 等。

矩阵式二维码 (dot matrix bar code): 是在一个矩形空间里通过黑、白像素在矩阵中的不同分布进行编码。矩阵式二维条码是建立在计算机图像处理技术、组合编码原理等基础之上的一种新型的图像符号自动识别处理码制。有代表性的有 CODE ONE、DATA MATRIX、CP 码等。

二维条码具有以下特点:

信息容量大, 在一个二维条码中可以存储 1000 个字节以上, 一个载体上可以有几个二维条码; 信息密度高, 同样大小的二维条码是一维条码信息密度的 100 倍以上; 识别率极高, 由于二维码有极强的错误修正技术, 即便破损、玷污 50% 的面积也能正确读出全部信息; 保密性、防伪能力强, 由于二维码的编码技术十分巧妙, 因此非常有效地防止伪造; 编码范围广, 可以将照片、指纹、掌纹、手写签名等凡是可数字化的信息均可编码; 制作容易、使用成本低, 可以打印在普通的纸张、PVC 或其他材料上, 与一维条码的制造成本相当。

二维条码技术标准:

国外对二维条码技术的研究始于 20 世纪 80 年代末, 已研制出多种码制, 全球现有的一、二维条码多达 250 种以上, 其中常见的有 PDF417, QRCode, Code49, Code16K, CodeOne 等 20 余种。二维条码技术标准在全球范围得到了应用和推广。美国讯宝科技公司(Symbol)和日本电装公司(Denso)都是二维条码技术的佼佼者。

目前得到广泛应用的二维码国际标准有 QR 码、PDF417 码、DM 码和 CM 码。

QR 码是由日本 Denso 公司于 1994 年 9 月研制的一种矩阵二维条码符号, 其全称为 Quickly Response, 意思是快速响应。它除具有一维条码及其他二维条码所具有的信息容量大、可靠性高、可表示汉字及图像多种文字信息、保密防伪性强等优点外, 还可高效地表示汉字, 相同内容, 其尺寸小于相同密度的 PDF417 条码。它是目前日本主流的手机二维码技术标准, 目前市场上的大部分条码打印机都支持 QRcode 条码。

PDF417 码是由美籍华人王寅敬(音)博士发明的。PDF 是取英文 Portable Data File 三个单词的首字母的缩写, 意为“便携数据文件”。因为组成条码的每一符号字符都是由 4 个条和 4 个空构成, 如果将组成条码的最窄条或统称为一个模块, 则上述的 4 个条和 4 个空的总模块数一定为 17, 所以称 417 码或 PDF417 码。

DM 码, 其全称为 DataMatrix, 中文名称为数据矩阵。DM 采用了复杂的纠错码技术, 使得该编码具有超强的抗污染能力。主要用于电子行业小零件的标识, 如 Intel 的奔腾处理器的背面就印制了这种码, DM 码由于其优秀的纠错能力成为韩国手机二维码的主流技术。

MC (Maxicode) 码(又称牛眼码), 是一种中等容量、尺寸固定的矩阵式二维条码, 它由紧密相连的六边形模组和位于符号中央位置的定位图形所组成。Maxicode 是特别为高速扫描而设计, 主要应用于包裹搜寻和追踪上。是由美国联合包裹服务(UPS)公司研制的, 用于包裹的分拣和跟踪。Maxicode 的基本特征: 外形近乎正方形, 由位于符号中央的同心圆(或称公牛眼)定位图形 (Finder Pattern), 及其周围六边形蜂巢式结构的资料位元所组成, 这种排列方式使得 Maxicode 可从任意方向快速扫描。

在我国, GM 《二维码网格矩阵码(GM)》SJ/T11349-2006 和 CM 《二维码紧密矩阵码(CM)》SJ/T11350-2006 标准是由原国家信息产业部于 2006 年 5 月 25 日所颁布的两项

国产行业推荐标准。此两项标准于 2006 年 5 月 30 日起实施。

GM 码其全称为网格码 (Grid Matrix Code) 是一种正方形的二维码码制, 该码制的码图由正方形宏模块组成, 每个宏模块由 6 乘 6 个正方形单元模块组成。网格码可以编码存储一定量的数据并提供 5 个用户可选的纠错等级。

CM 码意为“紧密矩阵”, 是英文 Compact Matrix 的缩写。码图采用齿孔定位技术和图像分段技术, 通过分析齿孔定位信息和分段信息可快速完成二维条码图像的识别和处理。

### 1.1.2 项目开发的意义

手机与二维码的结合, 进一步拓展了二维码应用价值, 促进了行业的融合。随着市场、产业链、商业模式的不断成熟, 手机二维码将为通信、媒体以及其他传统行业带来更多的商机, 手机二维码将在未来走进更多人的生活。随着全球信息通信技术的迅猛发展, 以手机为终端的各种应用不断涌现, 在未来, 手机无疑将成为移动商务赖以发展的最重要的终端之一, 而手机二维码的发展将在移动商务的发展中扮演重要角色。

由于手机二维码的广泛应用, 因此用手机终端生成二维码的功能就变得越来越重要了。现在的手机终端关于二维码功能的开发主要围绕的是二维码扫描, 这个功能可以让用户直接扫描二维码后获得名片信息、文本内容、直接上网等等, 很好的方便了用户。那么对于二维码生成功能也就越来越被需要, 此功能简单来讲也就是针对用户输入的信息生成二维码图片。对于手机终端的二维码生成可以随时随地把用户想要生成的信息生成二维码, 然后通过彩信发送给其他人, 或者直接打印分发做宣传。如果这两个功能可以同时放到手机上, 那么对于二维码的功能就相当的完备了。既然已经开发出了扫描二维码的功能, 那么对于开发生成二维码的功能就很理所当然了, 而我们当然也要与时俱进, 开发对社会有意义的软件。

## 1.2 项目开发的环境

### 1.2.1 硬件环境

Intel Celeron 2.4GHz 以上处理器，1GB 以上内存容量，80GB 以上硬盘容量。

### 1.2.2 软件环境

Microsoft Windows XP 操作系统，Eclipse 开发工具，Android SDK（Android 软件开发工具包）。

## 1.3 开发技术的介绍

### 1.3.1 Android 技术介绍

Android 的上层应用程序是用 Java 语言开发，同时还需要基于 Dalvik 虚拟机，所以，Google 公司推荐使用主流的 Java 继承开发环境 Eclipse。只有 Eclipse 还不够，因为是使用 Java 语言进行开发，还应该有由 SUN 公司提供的 Java SDK(其中包括 JRE: Java Runtime Environment)。此外，Android 的应用程序开发和 Java 开发有较大区别的，所以还需要有 Google 提供的 Android SDK。同时，还需要在 Eclipse 安装 ADT，为 Android 开发提供开发工具的升级或者变更，是 Eclipse 下开发工具的升级下载的工具。

简言之，需要以下软件，才能搭建 Android 开发环境，从而进行 Android 应用程序的开发。

- (1) Java SDK
- (2) Eclipse
- (3) Android SDK
- (4) ADT

### 1.3.2 Android 系统构架介绍

Android 作为一个移动设备的平台，其软件层次结构包括了一个操作系统（OS），中间件（MiddleWare）和应用程序（Application）。如图 1.1 所示，根据 Android 的软件框图，其软件层次结构自下而上分为以下几个层次：

第一、操作系统层（OS）；

第二、各种库（Libraries）和 Android 运行环境（RunTime）；

第三、应用程序框架（Application Framework）；

第四、应用程序（Application）；

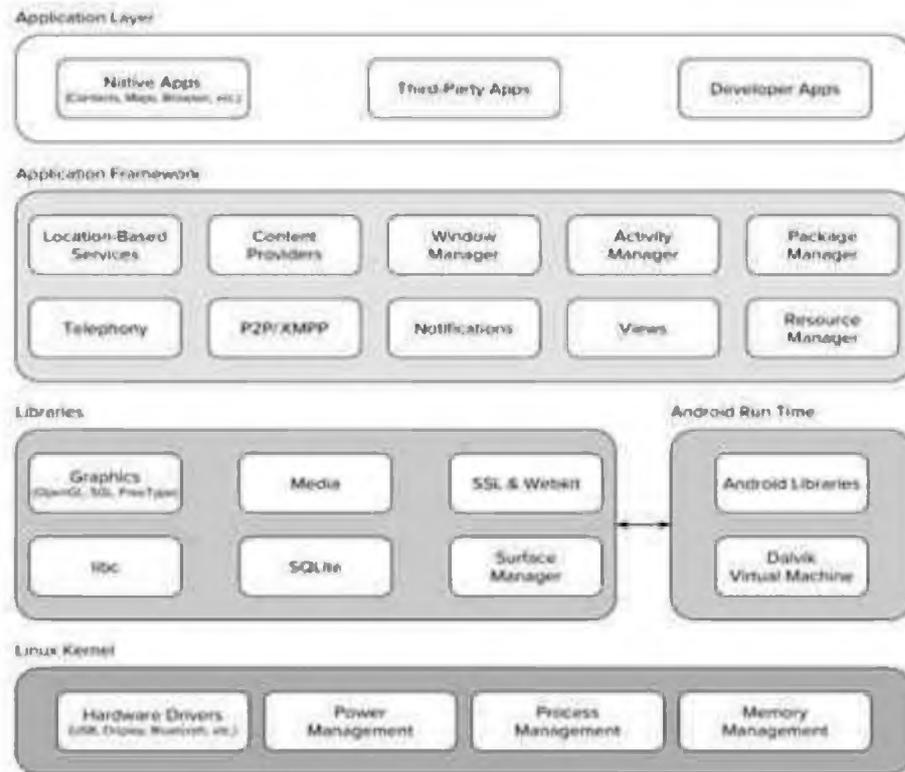


图 1.1 系统构架图

以下分别介绍 Andoid 各个层次的软件的重点及其相关技术：

### (1) 操作系统层（OS）

Android 使用 Linux2.6 作为操作系统，Linux2.6 是一种标准的技术，Linux 也是一个开放的操作系统。Android 对操作系统的使用包括核心和驱动程序两部分，Android 的 Linux 核心为标准的 Linux2.6 内核，Android 更多的是需要一些与移动设备相关的驱动程序。主要的驱动如下所示：

显示驱动（Display Driver）：常用基于 Linux 的帧缓冲（Frame Buffer）驱动；

Flash 内存驱动（Flash Memory Driver）；

照相机驱动（Camera Driver）：常用基于 Linux 的 v4l（Video for ）驱动；

音频驱动（Audio Driver）：常用基于 ALSA（Advanced Linux Sound Architecture，

高级 Linux 声音体系) 驱动;

WiFi 驱动 (Camera Driver): 基于 IEEE 802.11 标准的驱动程序;

键盘驱动 (KeyBoard Driver);

蓝牙驱动 (Bluetooth Driver);

Binder IPC 驱动: Andoid 一个特殊的驱动程序, 具有单独的设备节点, 提供进程间通讯的功能;

Power Management (能源管理)。

(2) 各种库 (Libraries) 和 Android 运行环境 (RunTime)

本层次对应一般嵌入式系统, 相当于中间件层次。Android 的本层次分成两个部分一个是各种库, 另一个是 Android 运行环境。本层的内容大多是使用 C++实现的。在其中, 各种库包括:

C 库 : C 语言的标准库, 这也是系统中一个最为底层的库, C 库是通过 Linux 的系统调用来实现。

多媒体框架 (MediaFramework): 这部分内容是 Android 多媒体的核心部分, 基于 PacketVideo (即 PV) 的 OpenCORE, 从功能上本库一共分为两大部分, 一个部分是音频、视频的回放 (PlayBack), 另一部分是则是音视频的纪录 (Recorder)。

SGL: 2D 图像引擎。

SSL: 即 Secure Socket Layer 位于 TCP/IP 协议与各种应用层协议之间, 为数据通讯提供安全支持。

OpenGL ES 1.0 : 本部分提供了对 3D 的支持。

界面管理工具 (Surface Management): 本部分提供了对管理显示子系统等功能。

SQLite: 一个通用的嵌入式数据库

WebKit: 网络浏览器的核心

FreeType: 位图和矢量字体的功能。

Android 的各种库一般是以系统中间件的形式提供的, 它们均有的一个显著特点就是与移动设备的平台的应用密切相关。

Android 运行环境主要指的虚拟机技术——Dalvik。Dalvik 虚拟机和一般 JAVA 虚

拟机 (Java VM) 不同, 它执行的不是 JAVA 标准的字节码 (bytecode) 而是 Dalvik 可执行格式 (.dex) 中执行文件。在执行的过程中, 每一个应用程序即一个进程 (Linux 的一个 Process)。二者最大的区别在于 Java VM 是以基于栈的虚拟机 (Stack-based), 而 Dalvik 是基于寄存器的虚拟机 (Register-based)。显然, 后者最大的好处在于可以根据硬件实现更大的优化, 这更适合移动设备的特点。

### (3) 应用程序框架 (Application Framework)

Android 的应用程序框架为应用程序层的开发者提供 APIs, 它实际上是一个应用程序的框架。由于上层的应用程序是以 JAVA 构建的, 因此本层次提供的首先包含了 UI 程序中所需要的各种控件: 例如: Views (视图组件) 包括 lists(列表), grids(栅格), text boxes(文本框), buttons(按钮)等。甚至一个嵌入式的 Web 浏览器。

一个 Android 的应用程序可以利用应用程序框架中的以下几个部分:

Activity (活动);

Broadcast Intent Receiver (广播意图接收者);

Service (服务);

Content Provider (内容提供者)。

### (4) 应用程序 (Application);

Android 的应用程序主要是用户界面 (User Interface) 方面的, 通常以 JAVA 程序编写, 其中还可以包含各种资源文件 (放置在 res 目录中) JAVA 程序及相关资源经过编译后, 将生成一个 APK 包。Android 本身提供了主屏幕 (Home), 联系人 (Contact), 电话 (Phone), 浏览器 (Browsers) 等众多的核心应用。同时应用程序的开发者还可以使用应用程序框架层的 API 实现自己的程序。这也是 Android 开源的巨大潜力的体现。

### 1.3.3 Eclipse 简介

Eclipse 是著名的跨平台的自由集成开发环境(IDE)。最初主要用来 Java 语言开发,但是目前亦有人通过插件使其作为其他计算机语言比如 C++和 Python 的开发工具。Eclipse 的本身只是一个框架平台,但是众多插件的支持使得 Eclipse 拥有其他功能相对固定的 IDE 软件很难具有的灵活性。许多软件开发商以 Eclipse 为框架开发自己的 IDE。

Eclipse 是一个开放源代码的软件开发项目,专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台。它主要由 Eclipse 项目、Eclipse 工具项目和 Eclipse 技术项目三个项目组成,具体包括四个部分组成——Eclipse Platform、JDT、CDT 和 PDE。JDT 支持 Java 开发、CDT 支持 C 开发、PDE 用来支持插件开发, Eclipse Platform 则是一个开放的可扩展 IDE,提供了一个通用的开发平台。它提供建造块和构造并运行集成软件开发工具的基础。Eclipse Platform 允许工具建造者独立开发与他人工具无缝集成的工具从而无须分辨一个工具功能在哪里结束,而另一个工具功能在哪里开始。

Eclipse SDK(软件开发者包)是 Eclipse Platform、JDT 和 PDE 所生产的组件合并,它们可以一次下载。这些部分在一起提供了一个具有丰富特性的开发环境,允许开发者有效地建造可以无缝集成到 Eclipse Platform 中的工具。Eclipse SDK 由 Eclipse 项目生产的工具和来自其它开放源代码的第三方软件组合而成。Eclipse 项目生产的软件以 CPL 发布,第三方组件有各自自身的许可协议。

### 1.3.4 ADT、 Android SDK 简介

目前 Android 开发所用的开发工具是 Eclipse,在 Eclipse 编译 IDE 环境中,安装 ADT,为 Android 开发提供开发工具的升级或者变更,简单理解为在 Eclipse 下开发工具的升级下载工具。

#### Android SDK:

在 Android 中,它为开发者提供了库文件以及其它开发所用到的工具。简单理解为开发工具包集合,是整体开发中所用到的工具包,如果你不用 Eclipse 作为你的开发工具,你就不需要下载 ADT,只下载 SDK 即可开发。

## 第二章 可行性分析

### 2.1 编写目的

可行性研究旨在评估目标系统是否值得去开发，问题是否能够解决。

一个项目的开发需要仔细分析，看它在目前的情况下是否可行，如果一个项目开发它具有很好的背景和意义，但是在经济上或者是技术上是不可行的，那么它也只能称之为不能实现的项目。

### 2.2 可行性研究任务

根据社会上对二维码的使用情况，以及一些二维码的生成软件进行可行性研究，主要从以下三个方面进行讨论：

#### 2.2.1 技术可行性

确定现有的技术是否能解决本该开发项目：本项目只是对输入的字符串能够生成二维码，并且要可以保存成图片，在现今社会上已有很多国家研究出多种制码标准，我们可以直接引用函数库，按照相应的机制就可以生成二维码，只要能够成功生成二维码那么对于保存成图片的功能就很简单了，因此在技术上可行。

#### 2.2.2 经济可行性

确定开发项目的开发成本以及估算开发项目所能够产生的经济效益：该软件可以快速的生成二维码，如果用户需要做一些宣传，即可用此软件生成的二维码，进行打印分发或者以短信的形式发送，这样可以起到很好的宣传效果；也可以以同样的方式做优惠券，这样顾客就可以直接扫描来获得优惠；或者有些人需要向对方表达情意而又不想被其他人看到，我们同样可以使用该软件，这些均可以证明该软件有很好的实用价值，因此该软件必定能给用户带来极大的方便，并且可以节约用户宝贵的时间，有很好的实用价值和发展前景，会有广大的使用用户，必定能够产生经济效益，所以本软件在经济上是可行的。

### 2.2.3 操作可行性

确定软件产品是否能够被最终的用户所接受，用户是否能操作最终的产品：由于本软件操作简单明了，界面非常友好，用户只需输入相应的字符串然后点击“生成二维码并保存”按钮即可生成二维码图片而且还可以保存成图片到手机的 sd 卡中，对于用户来说还要能够找到自己生成的图片存在哪里，只有这样他们才能用生成的图片去做一些事情，因此该软件还提供了有关图片的一些功能，如果我们想要查看保存的图片，可以点击查看图片按钮，便可以通过上一张、下一张来浏览图片，而且能够查看图片的详细信息，或者也可以根据图片名字查找图片，还包括删除当前显示的图片的功能。本软件既可以生成二维码也可以将其保存成图片，并且可以提供给用户一些有关图片的基本操作，功能上来说比较完备，实用上来说只是对一些按钮和输入信息的操作，都是一些很简单的操作，用户自己就可以看得懂，在操作上很方便。

## 2.3 可行性研究结论

根据以上几个方面的研究，软件在技术上是是可以实现的，经济上可以带来一定的经济效益而且成本不高，操作上相对比较简单，这些均可以证明该软件是可行的，应该立即进行本项目的开发。

## 第三章 需求分析

### 3.1 编写目的

需求分析是在可行性研究的基础上，将用户对系统的描述，通过开发人员的分析概括，抽象为完整的需求定义，再形成一系列文档的过程。需求分析旨在回答"系统做什么"的问题，确保将来开发出来的软件设备能够真正满足用户的需要。

### 3.2 功能需求分析

本软件主要实现对输入的字符串生成二维码，但是仅仅只是生成二维码的话，那么用户除了能够看到生成的二维码并不能用此二维码做什么，因此该软件至少还需要有保存功能，即把生成的二维码保存成图片，这样用户才可以用生成的二维码完成一些简单的功能，然后马上就会引出另一个问题那就是用户该到哪里去找到保存后的图片呢？针对这个问题让我思考到要给该软件增加一些对图片的操作，当用户把生成的二维码保存成图片后，那么至少可以让用户进入图片管理的操作界面，这些可以借鉴 android 自带的图库软件的功能。

通过以上的分析该软件最少应该具备的功能有：

- (1) 对输入的字符串生成二维码；
- (2) 能够把生成的二维码显示出来并可以保存成图片；
- (3) 可以上下键查找到图片，或者通过输入图片的名字进行查找；

可以查看图片的详细信息，对于上下键查找图片来说必须要通过查看图片的详细信息才能够确定是否为用户要找的图片，因为对于用户来说很难看出每一张的区别那么用户就只能根据图片名字来知道此图片是否为自己要找的图片。同时图片的存储路径也很重要，用户需要知道图片存储位置，这样用户才能够进入文件夹找到图片；

- (4) 删除当前显示的图片；

### 3.3 数据流图 (DFD)

数据流图 (DFD) 是结构化分析方法中用于系统逻辑模型的一种工具, 以图形方式描绘数据在系统中流动和处理过程。

#### 3.3.1 顶层数据流图

顶层流图只包含一个加工, 用以表示被开发的系统, 然后考虑该系统有哪些输入数据、输出数据流。顶层图的作用在于表明被开发系统的范围以及它和周围环境的数据交换关系。本系统的顶层数据流图, 如图 3.1 所示:



图 3.1 顶层数据流图

#### 3.3.2 一层数据流图

此部分的主要输入流为生成二维码并保存成图片请求和显示图片请求, 输出为图片, 如图 3.2 所示:

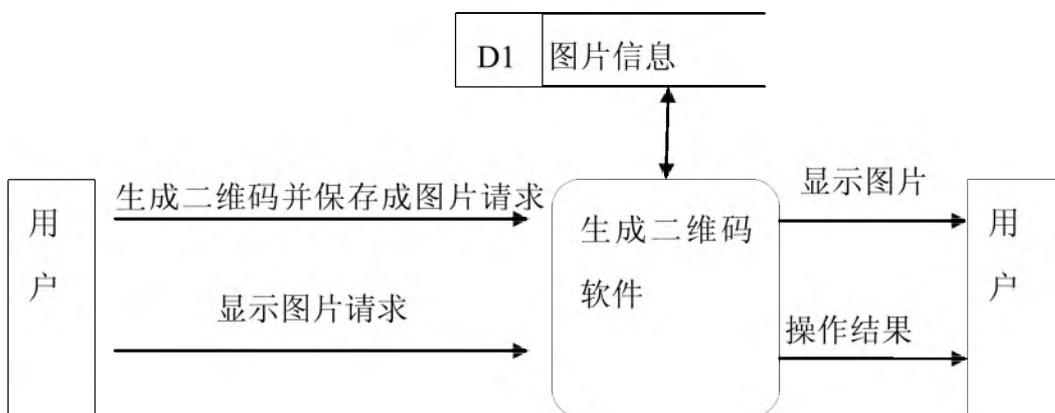


图 3.2 一层数据流图

### 3.3.3 二层数据流图

二层数据流图主要是把二维码生成软件细化成两部分，一部分为生成二维码并保存成图片，另一部分为显示图片部分，其中两个输入流分别为字符串和显示图片请求，输出流分别为图片和操作结果，如图 3.3 所示：

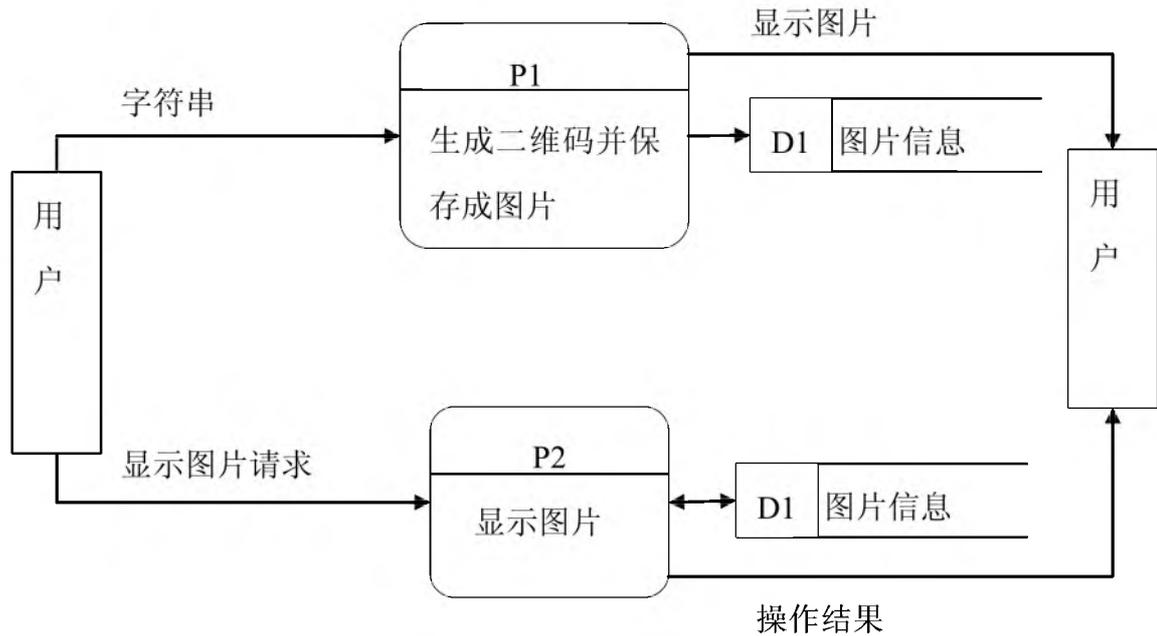


图 3.3 二层数据流图

### 3.3.4 三层数据流图

三层数据流图主要是把显示图片功能进行的细化其中包括上移查找图片功能、下移查找图片功能、删除图片功能、显示详细信息功能和根据图片名查找图片功能，在这里我们可以看到每一部分的输入流和输出流分别是什么，如图 3.4 所示：

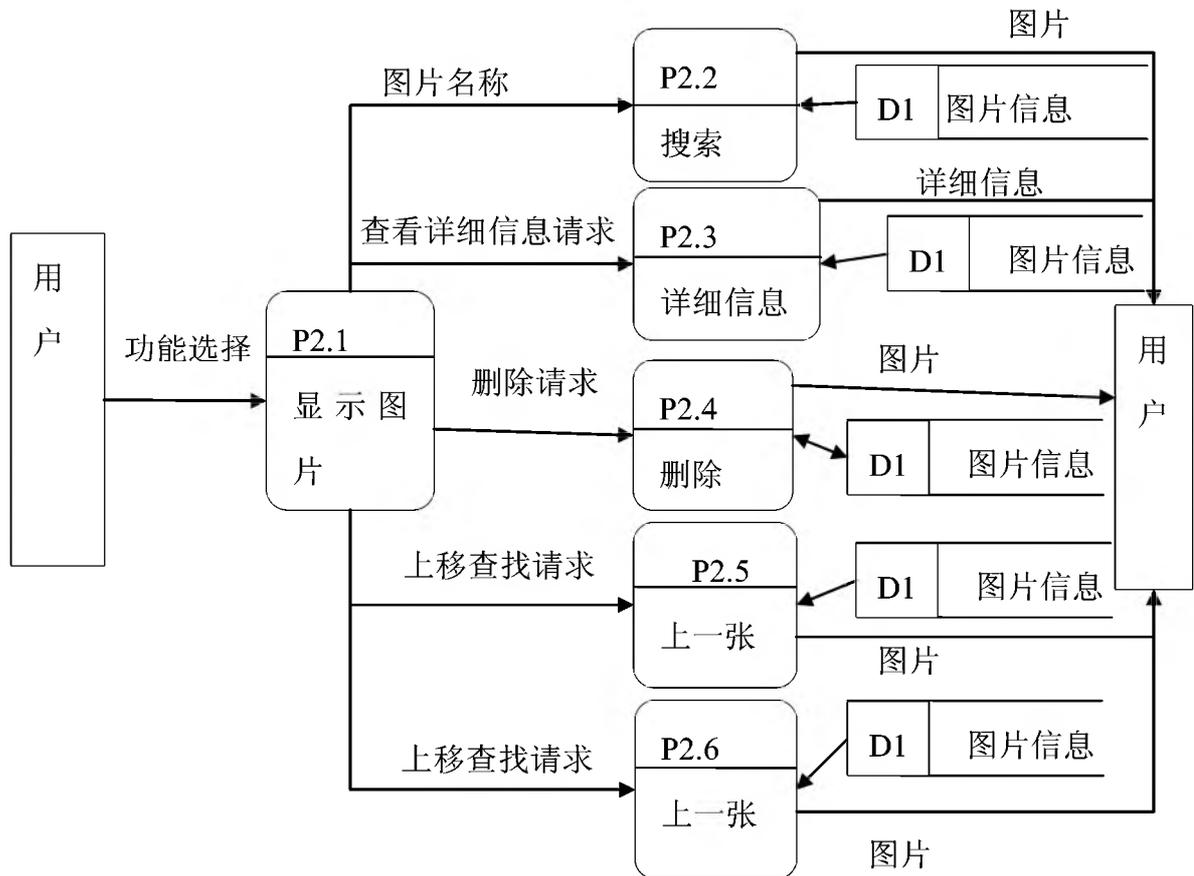


图 3.4 三层数据流图

### 3.4 数据字典 (DD)

数据字典 (DD, Data Dictionary) 数据字典一般应包括对数据流、数据元素 (数据流分量)、数据存储和处理的说明。数据字典是对数据流程图的重要补充和说明。

数据字典是关于数据的信息的集合,也就是对数据流图中包含所有元素的定义的集合。数据字典的作用在软件分析和设计的过程中给人提供关于数据的描述信息。现简要列出本系统的部分数据字典。

1.数据存储卡片的定义，如图 3.5 所示。

名字：图片信息  
描述：存放图片的一些基本信息  
定义：图片信息=图片编号、图片名称、存储路径、图片尺寸  
位置：图片信息表

图 3.5 图片信息

2.处理卡片定义，如图 3.6 所示：

名字：生成二维码并保存成图片  
编号：P1  
输入流：字符串  
输出流：二维码图片  
描述：把输入的字符串生成二维码

名字：搜索图片  
编号：P2.2  
输入流：图片名字  
输出流：图片  
描述：根据图片名字查找到图片

名称：删除图片  
编号：P2.4  
输入流：图片存储路径  
输出流：图片  
描述：根据图片存储路径完成删除当前图片的功能

图 3.6 数据处理卡片

## 第四章 总体设计

### 4.1 编写目的

总体设计是项目开发中的一个重要环节，目标是将上一章节需求分析阶段得到的目标系统的逻辑模型，变换为目标系统的物理模型，此阶段主要是进行功能模块的结构设计和数据库的设计。总体设计阶段重要任务是设计软件的结构，也就是要确定系统中每个程序是有哪些模块组成的，以及这些模块相互间的关系。

### 4.2 系统功能图

对于二维码生成软件主要是帮助用户生成二维码并且可以将其保存成图片，但是还要方便用户找到自己存储的图片位置，再增加一些图片处理功能，如图 4.1 所示：

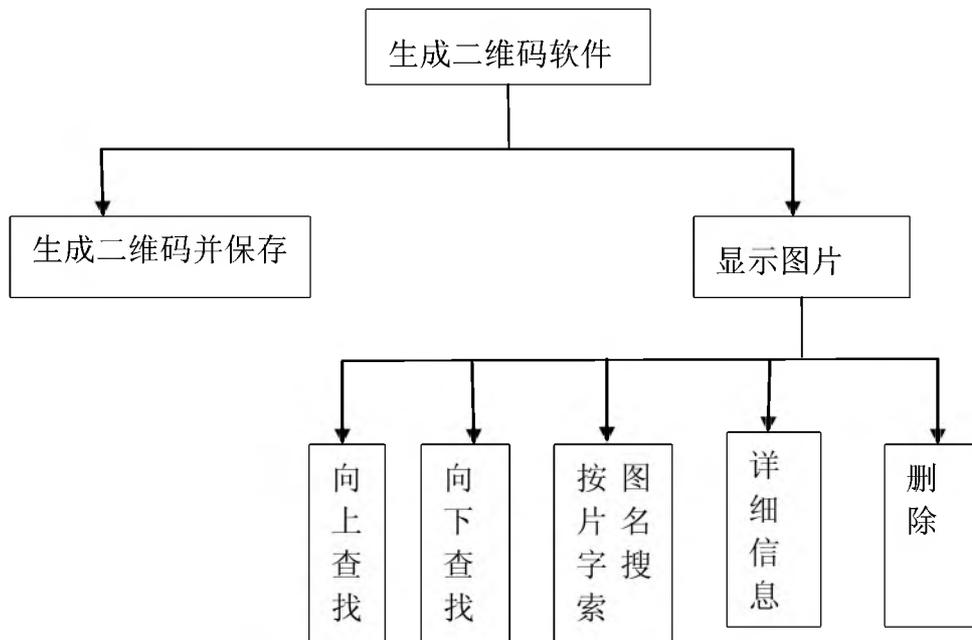


图 4.1 功能图

1. 生成二维码并保存：主要完成对输入的字符串生成二维码，并且保证显示出来以及保存（根据输入的字符串命名图片）。
2. 显示图片部分

- (1) 向上查找：通过“上一个”按钮向上查找图片。
- (2) 向下查找：通过“下一个”按钮向下查找图片。
- (3) 按图片名字搜索：输入图片的名字然后点击“搜索”按钮查找图片。
- (4) 删除：删除当前显示的图片。

### 4.3 系统流程图

#### 4.3.1 总系统流程图

总系统流程图主要展示的是全部的功能，以及输入的信息。如图 4.2 所示：

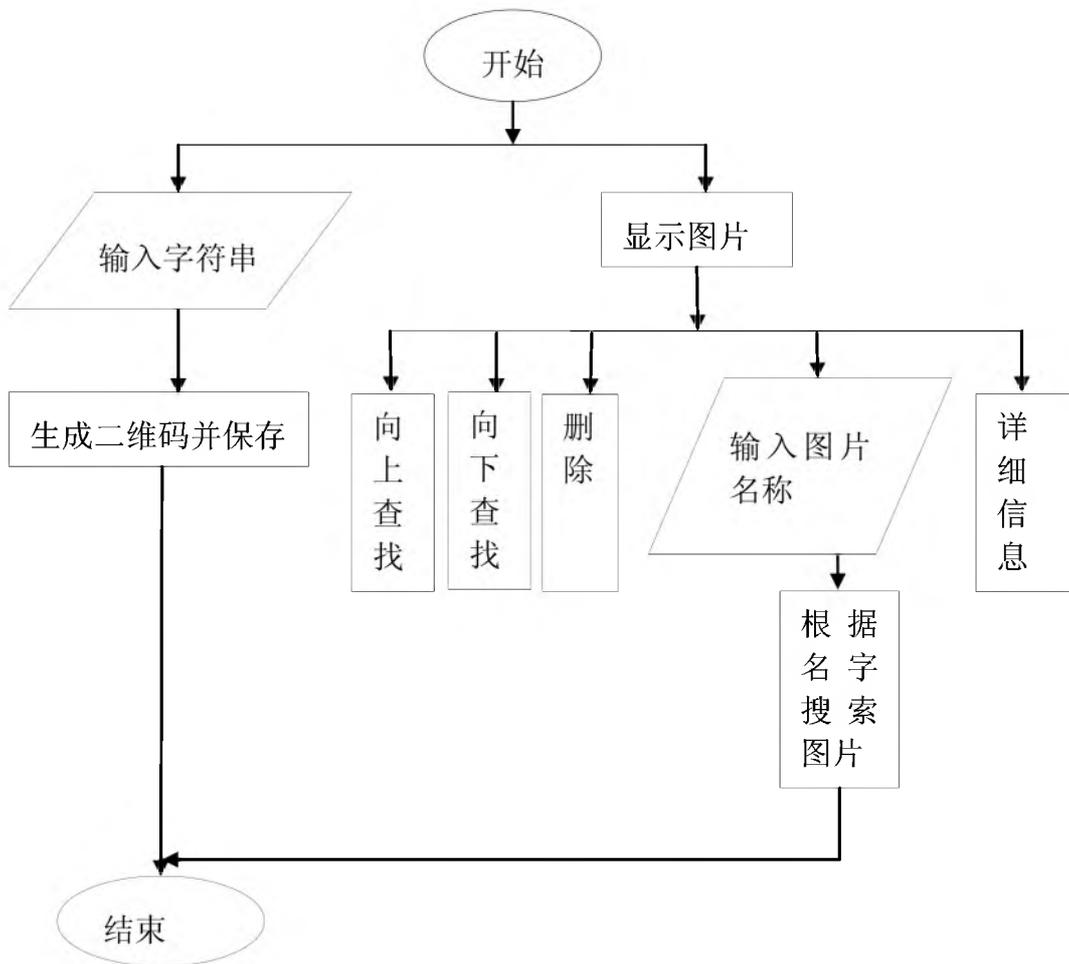


图 4.2 总系统流程图

### 4.3.2 生成二维码并保存部分流程图

此部分数据流图主要是在存储二维码图片时候有分支，其中既要判断是否有 sd 卡还要判断 sd 卡的内存是否足够存储一张图片的，当条件全部满足的时候才能成功保存图片，如图 4.3 所示：

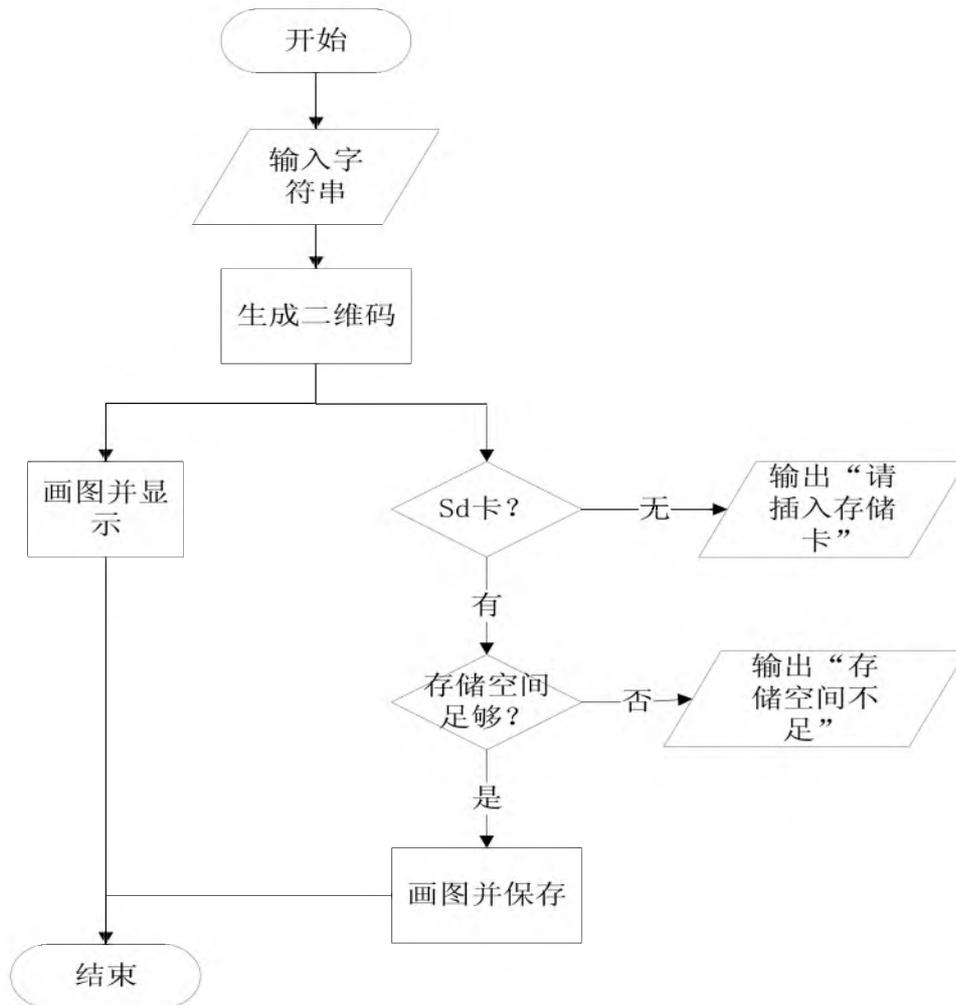


图 4.3 生成二维码并保存部分流程图

### 4.3.3 显示图片部分流程图

此部分主要展示的是显示图片那部分功能的细化流程图，详细的表示了分支部分的情况，如图 4.4 所示：

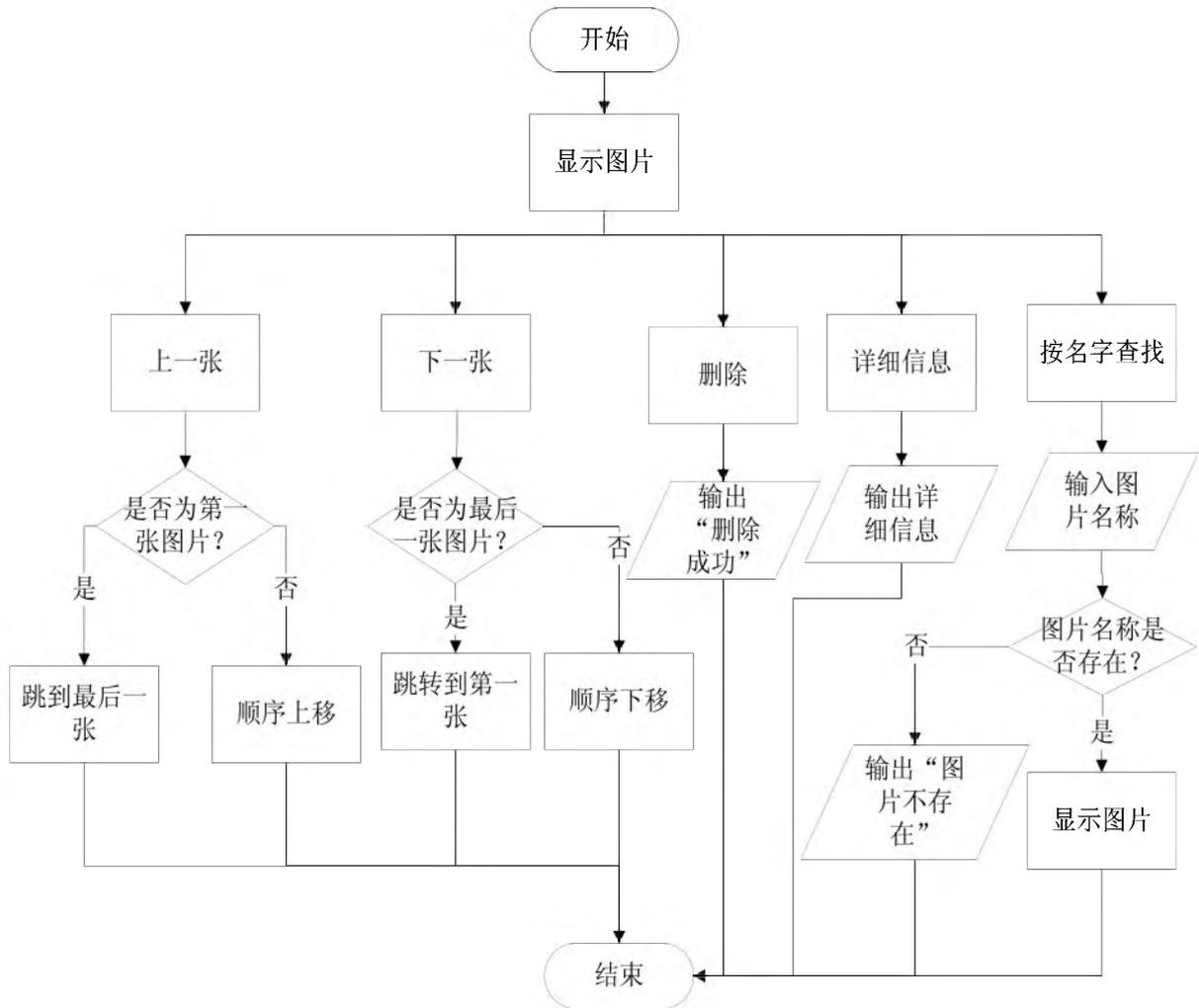


图 4.4 显示图片部分流程图

## 4.4 数据库

### 4.4.1 sdcard 卡对应的数据库

对于本软件的设计并没有用自己设计数据库，而是直接用 Android 自带的多媒体数据库。

当 Android 的系统启动的时候，系统会自动扫描 sdcard (包括大容量存储)跟系统内存的多媒体文件，并把获得的信息保存在两个系统数据库中，以后在其他程序中如果想要访问多媒体文件的信息，其实就是在这两个数据库中进行的，而不是直接去 sdcard 卡中或者手机内存里取。两张数据都是位于/data/data/com.android.providers.media/databases 目录下。如图 4.5 所示：

com.android.launcher		2012-03-17	12:27	drwxr-x--x
com.android.mms		2012-03-17	12:27	drwxr-x--x
com.android.music		2012-03-17	12:26	drwxr-x--x
com.android.netspeed		2012-03-17	12:27	drwxr-x--x
com.android.packageinstaller		2012-03-17	12:27	drwxr-x--x
com.android.phone		2012-03-17	12:27	drwxr-x--x
com.android.protips		2012-04-17	10:35	drwxr-x--x
com.android.providers.applications		2012-03-17	12:26	drwxr-x--x
com.android.providers.contacts		2012-03-17	12:27	drwxr-x--x
com.android.providers.downloads		2012-03-17	12:27	drwxr-x--x
com.android.providers.drm		2012-03-17	12:26	drwxr-x--x
com.android.providers.media		2012-03-17	12:27	drwxr-x--x
databases		2012-05-22	08:19	drwxrwx--x
external-13f50911.db	40960	2012-05-18	03:29	-rw-rw----
external-15f41401.db	40960	2012-05-22	08:19	-rw-rw----
internal.db	34816	2012-04-22	02:24	-rw-rw----
lib		2012-03-17	12:26	drwxr-xr-x
com.android.providers.settings		2012-03-17	12:27	drwxr-x--x

图 4.5 数据库存储位置

名为 external.db 里面存储的是 sdcard(包括大容量存储的媒体信息)上的媒体信息，名为 internal.db 存储的就是手机内存的媒体信息，比如说手机出厂时自带的一些铃声。当然如果有多张 sdcard 更换使用时，这时 external.db 不一定是固定的，但每一个名字确实对应着一张 sdcard，而在本软件用到的是 external-15f41401.db 数据库。

下图为用 SQLite Expert 这个工具打开这个数据库文件看一下，如图 4.6 所示：

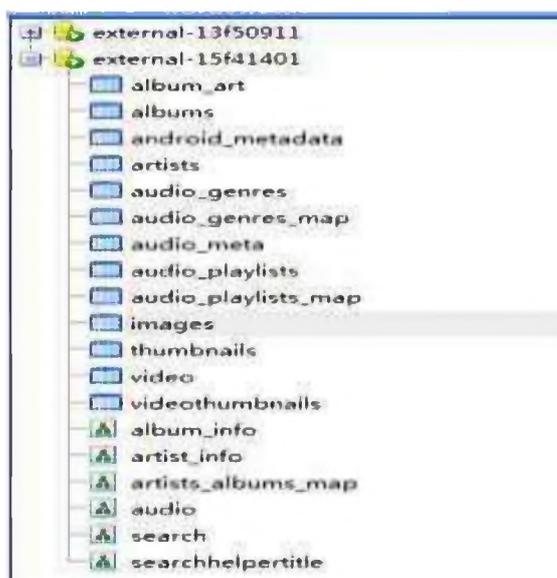


图 4.6 数据库文件

对于本软件主要是对媒体数据库中的 images 表进行操作，接下来看看 images 表的设计，如图 4.7 所示：

Index	Name	Declared Type	Type
1	_id	INTEGER	INTEGER
2	_data	TEXT	TEXT
3	_size	INTEGER	INTEGER
4	_display_name	TEXT	TEXT
5	_mime_type	TEXT	TEXT
6	_title	TEXT	TEXT
7	_date_added	INTEGER	INTEGER
8	_date_modified	INTEGER	INTEGER
9	_description	TEXT	TEXT
10	_person_id	TEXT	TEXT
11	_latitude	INTEGER	INTEGER
12	_latitude	DOUBLE	DOUBLE
13	_longitude	DOUBLE	DOUBLE
14	_date_taken	INTEGER	INTEGER
15	_orientation	INTEGER	INTEGER
16	_mini_thumb_magic	INTEGER	INTEGER
17	_bucket_id	TEXT	TEXT
18	_bucket_display_name	TEXT	TEXT

图 4.7 images 表设计

数据库表字段解析如表 4-1 所示：

表 4-1 图片信息

字段	解析
_id	主键。图片 id, 从 1 开始自增
_data	图片绝对路径
_size	文件大小, 单位为 byte
_display_name	文件名
mime_type	类似于 image/jpeg 的 MIME 类型
title	不带扩展名的文件名
date_added	添加到数据库的时间, 单位秒
date_modified	文件最后修改时间, 单位秒
description	描述
picasa_id	用于 picasa 网络相册
isprivate	私有
latitude	纬度, 需要照片有 GPS 信息
longitude	经度, 需要照片有 GPS 信息
datetaken	取自 EXIF 照片拍摄时间, 若为空则等于文件修改时间, 单位毫秒
orientation	取自 EXIF 旋转角度, 在图库旋转图片也会改变此值
mini_thumb_magic	取小缩略图时生成的一个随机数, 见 MediaThumbRequest
bucket_id	等于 path.toLowerCase.hashCode(), 见 MediaProvider.computeBucketValues()
bucket_display_name	直接包含图片的文件夹就是该图片的 bucket, 就是文件夹名

而在我设计的软件中只是用到了其中的四个字段 `_id`, `_data`, `_size`, `_display_name`, 通过这四个字段找到图片并显示其详细信息, 也可以完成操作。

#### 4.4.2 更新数据库说明

其他程序访问多媒体文件的信息, 是在这两个数据库中进行的, 而不是直接去 `sdcard` 中或者手机内存里取。当生成一张图片完成的时候, 可能就会需要更新图库, 重启机器或者拔了 `sdcard` 再插上或者发送一个广播就会把增加的信息同步进数据库了。

## 第五章 软件设计

### 5.1 用户界面设计

根据需求分析可以知道，此软件应该包含两个主要的界面设计里需要进一步分析每个用户界面中应该包括哪些显示内容。

首先一定是一个界面用来完成生产二维码，另外一个界面是用来查找图片，而且对于这两个界面之间要有联系，因此要用一个按钮的响应来触发另一个界面，下面根据思路设计出如图 5.1 的界面草图：



图 5.1 用户界面设计

### 5.2 文件介绍

首先要创建二维码生成软件的工程，工程名为：CaptureZxing，包名为 com.zxing 其中包含两个 Activity: CaptureZxingActivity 和 lookpictureActivity, CaptureZxingActivity 是软件启动时候的页面，也是生成二维码的页面，lookpictureActivity 是点击显示图片进入后的页面，在这里可以完成查找图片的各种功能。文件夹 gen 中的 R.java 是自动生成

的，对于本软件还引用了外面的函数库 Qrcode.jar，res/layout 中的文件是布局文件，res/values 中的文件是定义的资源文件，AndroidManifest.xml 文件中主要有对 Activity 的定义，还有一些权限的定义等。

接下来看一下二维码生成软件中的文件情况如图 5.2 所示：

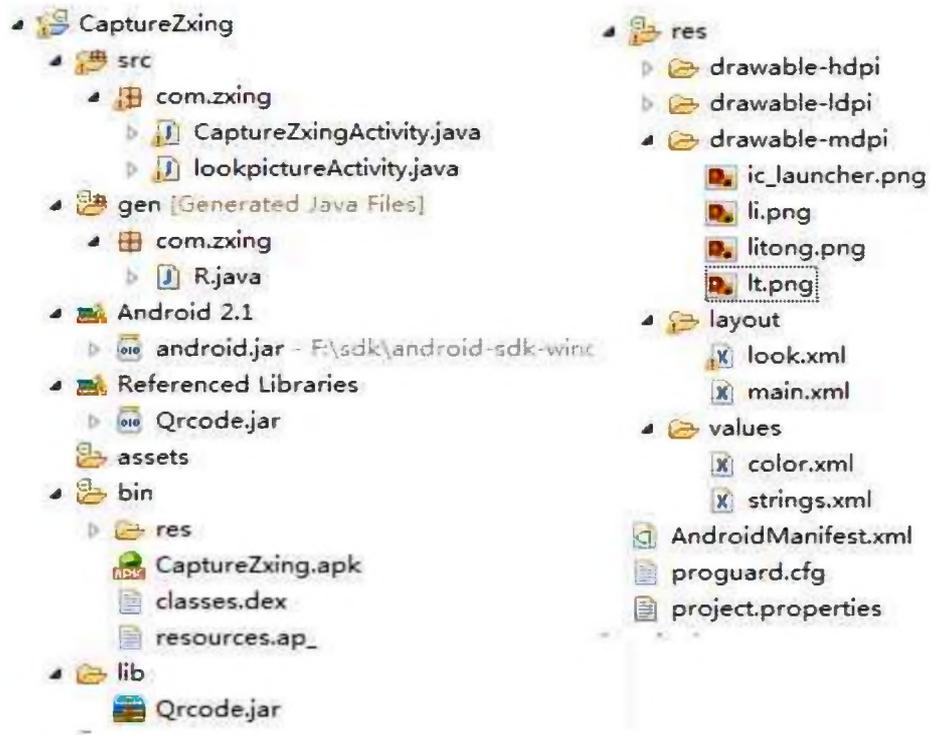


图 5.2 资源文件截图

详细介绍一下各个资源文件的作用，如表 5-1 所示：

表 5-1 资源文件作用表

资源目录	文件	说明
drawable	litong.png	软件图标
	lt.png	背景图片
layout	main.xml	CaptureZxingActivity 的布局文件
	look.xml	lookpictureActivity 的布局文件
values	color.xml	对颜色进行定义的文件
	strings.xml	对字符串定义的文件

### 5.2.1 资源文件中的代码

main.xml 文件中的代码是 CaptureZxingActivity 的布局文件, 在 CaptureZxingActivity 中通过找到 main.xml 中各个控件的 id 来调出整个的布局。

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/lt"
    android:orientation="vertical">
    <TextView
        android:id="@+id/mTextView01"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
        android:textColor="@color/black" />
    <EditText
        android:id="@+id/myEditText01"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:inputType="none" />
    <SurfaceView
        android:id="@+id/mSurfaceView01"
        android:layout_width="320dp"
        android:layout_height="240dp"
        android:visibility="visible"/>
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:id="@+id/mButton01"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/OK"
            android:textColor="@color/black"
            android:layout_weight="1" />
        <Button
            android:id="@+id/mButton02"
            android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:text="@string/look"
        android:textColor="@color/black"
        android:layout_weight="1"></Button>
    </LinearLayout>
</LinearLayout>

```

得到的布局页面，如图 5.3 所示：



图 5.3 生成二维码界面截图

look.xml 文件中的代码是 lookpictureActivity 的布局文件，在 lookpictureActivity 中通过找到 look.xml 中各个控件的 id 来调出整个的布局。

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/lt"
    android:orientation="vertical" >
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >
        <Button
            android:id="@+id/next"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:paddingLeft="30dip"
            android:text="@string/next" />

```

```
<TextView
    android:id="@+id/number"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingLeft="30dip"
    android:textColor="#000000" />
<Button
    android:id="@+id/previous"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingLeft="30dip"
    android:text="@string/previous" />
</LinearLayout>
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >
    <EditText
        android:id="@+id/name"
        android:layout_width="150dip"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/sou"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="30dip"
        android:text="@string/sou" />
</LinearLayout>
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >
    <Button
        android:id="@+id/message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="30dip"
        android:text="@string/xian" />
    <Button
        android:id="@+id/delete"
        android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:paddingLeft="30dip"
        android:text="@string/delete" />
</LinearLayout>
<ImageView
    android:id="@+id/image_view"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" />
</LinearLayout>

```

得到的布局页面，如图 5.4 所示：



图 5.4 显示图片界面截图

values 中 color.xml 是用来定义软件中用到的颜色。

```

<?xml version="1.0" encoding="UTF-8"?>
<resources>
    <color name="black">#000000</color>
    <color name="white">#FFFFFF</color>
</resources>

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/067060044042006056>