

移动学生成绩查询系统 ——移动电子商务系统的开发

学习目标

- 理解移动学生成绩查询系统的需求及其分析。
- 理解移动电子商务平台的架构。
- 掌握移动学生成绩查询系统客户端的设计和实现方法。
- 掌握移动学生成绩查询系统服务器端的设计和实现方法。
- 掌握访问网络时,多线程的设计方法。
- 掌握后台数据库的设计和连接方法。
- 能够开发类似的移动电子商务系统。

本章利用 J2ME 技术开发一个移动电子商务系统——移动学生成绩查询系统,用户可以通过在手机上运行移动学生成绩查询系统查询到学生的课程成绩。该系统涵盖的主要内容包括高级用户界面设计、J2ME 网络编程、多线程的设计等。通过本章的项目开发,读者应该掌握类似的移动电子商务平台的设计和实现方法。

3.1 项目描述

现阶段,用户查询所需要的资料,通常是通过打电话得到的,例如查询电话号码信息的 114 电话,查询天气信息的 112 电话,还有 160 电话等。这种查询方式通常要通过人工接听电话,查询费用比较高。如果在手机上运用 Java 程序通过 GPRS(通用分组无线业务)上网查询所需的资料,可使用户以较低的费用,达到同一目的,甚至有更强的功能。我们的项目正是在这样的情况下产生的。随着手机在广大民众、特别是校园里的学生和老

师中的普及,类似的通过手机进行信息查询的系统有着很好的应用前景。

本章要开发的移动学生成绩查询系统是一个主要面向学生和老师的 Java 手机应用系统,用户(学生或老师)只要将移动学生成绩查询系统安装在支持 Java 技术的手机上,就可随时随地查询服务器上保存的学生的课程成绩。

该系统的产品定位范围广泛,从用户市场来看,主要是针对校园里的需要查询学生成绩的老师和学生;从地域市场来看,该系统不仅可以在国内使用,只要有 GPRS 网络覆盖到的地方都可以使用;从行业市场来看,该系统不仅适用于教育行业,经过改造后的类似系统可以适用于政府、金融、通信、体育等各个行业。

系统的主要界面如图 3-1 所示。用户只要输入要查询学生的学号和要查询的学期,就可以查询到该学生指定学期的成绩。

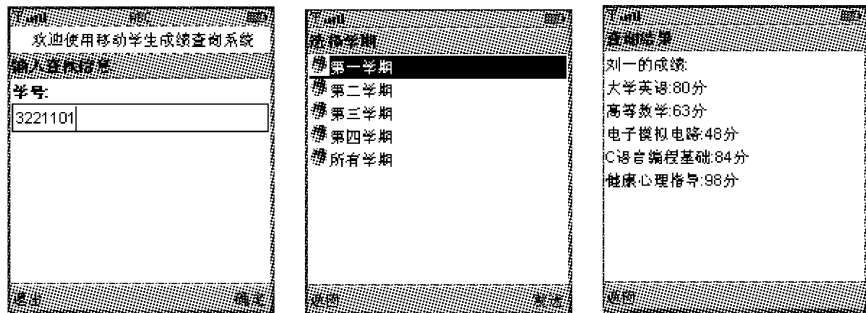


图 3-1 移动学生成绩查询系统主要界面

3.2 项目需求

3.2.1 项目功能需求

本项目的对应程序主要分成三个部分:客户端、服务器端和数据库。

1. 客户端

客户端(手机端)用于接收用户的输入,并将用户输入的查询内容发送到服务器端;接收服务器端的响应后把查询结果显示在屏幕上,如图 3-1 所示。

2. 服务器端

服务器从客户端得到查询内容后,连接到数据库进行查询,最后将查询结果返回到客户端。

3. 数据库

本项目中,全部学生的成绩保存在 Access 数据库中。保存的成绩包括:学生的学

号、姓名、课程的编号、课程名称、该课程是属于哪个学期的、学生所修课程的成绩。

3.2.2 技术支持

本项目主要用了如下技术：J2ME、GPRS 和 Servlet。

1. J2ME

J2ME(Java 2 Micro Edition)是 Java 家族中重要的一员,它是当前使用比较多的基于移动设备开发技术。J2ME 构建于 MIDP(Mobile Information Device Profile)和 CLDC(Connected Limited Device Configuration)或 CDC(Connected Device Configuration)基础之上。

目前国内外 J2ME 比较多的应用包括车载信息和计算、数字电视、电视机顶盒、Internet 访问设备、家庭 Audio/Video、无线手持设备等。

J2ME 具有很多 Java 技术特性,主要有如下几点:

- 可以在各种支持 Java 的设备上运行。
- 代码短小。
- 充分利用 Java 语言的优势。
- 安全性好。
- 用 J2ME 实现的应用可以方便地升级到 J2SE(Java 2 标准版)、J2EE(Java 2 企业版)。

2. GPRS

GPRS(General Packet Radio Service)是通用分组无线业务的英文简称,是一种基于 GSM(Global System for Mobile Communication,全球移动通信系统,简称“全球通”)系统的无线分组交换技术,提供端到端的、广域的无线 IP 连接。通俗地讲,GPRS 是一项高速数据处理技术,其实现的方法是以“分组”的形式将资料传送到用户手上。

虽然 GPRS 是在 GSM 基础上发展起来的一种分组交换的数据承载和传输方式,但是与原有的 GSM 相比,GPRS 在数据业务的承载和支持上具有非常明显的优势:更有效地利用无线网络信道资源,特别适合于突发性、频繁的小流量数据传输;支持的数据传输速率更高,理论峰值可以达到 115Kbps;计费方式更加灵活,支持按数据流量来进行计费,收费更合理;GPRS 还支持在进行数据传输的同时进行语音通话等。具有“实时在线”、“按量计费”、“快捷登录”、“高速传输”、“自如切换”的优点。GPRS 为无线数据传送提供了一条高速公路,只要能接入 GPRS 网络,就能使用无线数据业务,因此 GPRS 终端将不仅仅局限于现有的手机,还可能应用于随时在线的游戏机、做工精美的无线电子书、无线掌上电脑等。

从 2002 年 5 月 17 日起 GPRS 正式商用。目前中国移动 GPRS 网络已覆盖全国所有

省、直辖市、自治区,网络遍及 240 多个城市。无论在繁华的闹市,还是在僻静的边远地区,GPRS 都能够最大限度地为商务人士和各行业集团客户提供高速数据服务。全球 200 多个最大的运营商都选择了 GPRS 网络,GPRS 数据业务可以遍及世界各地。同时中国移动的用户已经可以与中国香港、中国台湾、新加坡实现 GPRS 自由通信,实现真正的移动数据无线互联的概念。

3. Servlet

Servlet 是服务器端语言,它的优点是自动开启多个线程接受多个用户访问。

Servlet 的生命周期从 `init()` 方法开始,Servlet 一旦创建,它便会保存在服务器中,除了第一个用户应用 `init()` 方法之外,其他用户都不需要激活该方法。当服务器关闭该 Servlet 的应用或者服务器关闭时,Servlet 生命便结束,Servlet 生命结束之前会激活方法 `destroy()`。Servlet 这一特性与全局 EJB(企业 Java 组件模型)十分相似,我们可以使用 Servlet 这一特性,建立数据库连接的缓冲池。

客户端程序可以决定使用 `doGet()` 或者 `doPost()` 方法向服务器传递数据,客户端通过 form 标识的 method 参数定义 `get` 或者 `post` 方法。`doGet()` 方法是默认的方法,但是该方法传递的数据量不能太大;如果要传递大量数据,就要使用 `doPost()`。Servlet 的 `service()` 方法可以接受 `get()`、`post()` 或者其他方法传入服务器的数据。如果不能确认用户使用哪种方法向服务器传入数据,可以使用 `service()` 方法接收不同类型方法传入的参数。Servlet 使用 `getParameter()` 取得用户变量。

3.3 概要设计

本项目是一个典型的移动电子商务系统,它基于 J2ME 技术实现客户端程序,基于 J2EE(Java 2 Enterprise Edition)技术实现服务器端程序。因此,在具体地进行项目的概要设计之前,我们先学习一下移动电子商务平台的架构。

3.3.1 移动电子商务平台的架构

J2EE™(The Java™ 2 Platform, Enterprise Edition)是目前比较流行的多层企业应用技术。它的优点之一是能够适应不同类型的客户端,如 Web 浏览器、Java Applets 和 Java 应用程序,这些客户端能够很容易地部署在笔记本电脑、台式电脑和工作站上,实现跨平台。

J2ME™(The Java™ 2 Platform, Micro Edition)是当前使用比较多的基于移动设备开发技术。它提供了一种新型的企业客户端类型,如手机、PDA 等手持设备。J2ME 构建于 MIDP(Mobile Information Device Profile)和 CLDC(Connected Limited Device

Configuration)或 CDC(Connected Device Configuration)基础之上。MIDP 是设备生产厂商根据需要,构造一系列 Java APIs,不同的移动设备厂商可以根据不同的需要构造不同的 APIs 来满足设备开发需要。而 CLDC 或 CDC 是由 Sun 公司提供的所有移动设备必须支持的 Java APIs。因此移动设备的开发难点之一在于如何开发出具有通用性的程序。

结合 J2ME 和 J2EE 技术,我们可以建立一个完整的移动电子商务平台。移动电子商务(Mobile Commerce,简称 M-Commerce)是传统电子商务在移动网络中的应用,但是 M-Commerce 提供了比 E-Commerce 更灵活、更方便、更及时的信息服务。随着无线接入网络 GPRS(2.5G)和 3G 的发展,在无线网络中数据传送速度越来越快,为移动电子商务提供了良好的网络通信环境。目前无线终端访问网络主要有三种模式:WAP 模式、i-Mode 模式和 J2ME 模式。

WAP(Wireless Application Protocol,无线应用协议)是使移动通信设备可靠地接入互联网的公认标准之一。使用 WML(无线标记语言),WAP 协议可以广泛地运用于 GSM、CDMA、TDMA、3G 等多种网络,但是 WAP 也有它的局限性,一方面是 WAP 设备受到 CPU、RAM、ROM 和处理速度的限制,只能处理有限的数据输入;另一方面,WAP 承载网络是低功率的网络,带宽小于 10Kbps,可靠性不高、稳定性不高和不可预测性。

i-Mode(Information Mode)是由日本电报电话公司(NTT)和移动通信公司 DoCoMo 公司于 1999 年 2 月推出的移动互联网技术,这种技术使得用户能够通过蜂窝电话使用 Internet 服务。i-Mode 的带宽为 9600bps,它也具有可靠性不高、稳定性不高和不可预测性等局限性。

J2ME 结合目前 GPRS 和 3G 网络,具有较高的数据传送能力,Java 机制和优化的 HTTPS 和 KSSL 技术保障了数据在网络上的安全性,同时,它提供直接访问网络的能力,而 WAP 和 i-Mode 模式必须要通过网关(Gateway)才能访问网络。

在基于 J2ME-J2EE 技术的电子商务中,数据的传输需要经过无线网络和有线网络。在通信过程中,移动设备端发送数据到无线基站,无线基站将接收到的数据发送到 GPRS/3G 网络中,根据 GPRS/3G 的特点,它们可以很方便地接入到 Internet 中,在 Internet 上数据进行传输,发送到企业网络中进行会话。其网络结构示意图如图 3-2 所示。

在建立移动电子商务平台时,由于移动终端设备资源有限,它不提供直接对后台数据库的访问能力。因此,要想使移动设备终端能够访问到数据库系统,必须要使用 J2EE 技术。根据需求,一个移动电子商务系统必须建立移动终端程序(使用 J2ME 技术)和服务端应用程序(使用 J2EE 技术)。基于 J2ME-J2EE 的移动电子商务系统的架构如图 3-3

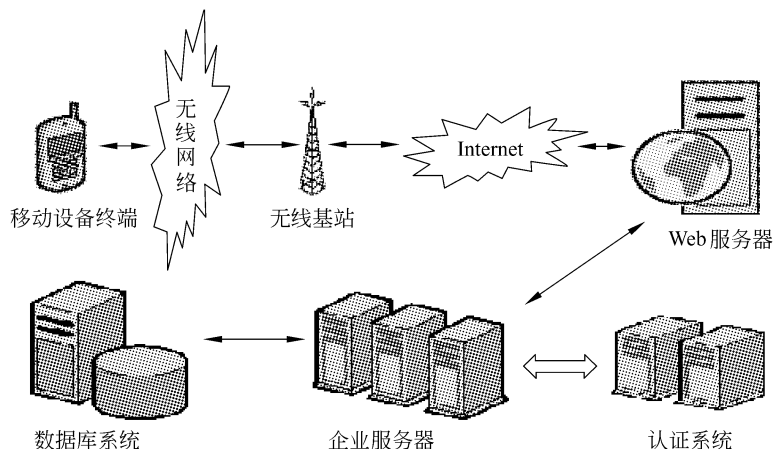


图 3-2 移动电子商务平台网络结构示意图

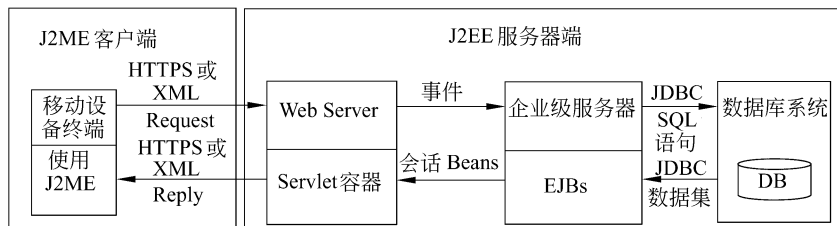


图 3-3 移动电子商务应用系统的架构

所示。

从图 3-3 可以看到,在 J2ME 客户端,移动设备通过 HTTP(HyperText Transfer Protocol)协议或 HTTPS(HyperText Transfer Protocol Secure)协议或者 XML(Extensible Markup Language)连接到 Web Server 上,Web Server 使用 Servlet 容器来处理客户端的连接和请求,然后再将这些请求发送到企业级服务器上,企业级服务器使用 EJB 进行会话,处理请求,并通过 JDBC(Java 数据库连接性)连接到数据库系统上。当数据库系统处理完 SQL 语句后,将会形成一个数据集,并将这个数据集发送到企业级服务器,企业级服务器把接收到的信息提交给 Web Server,Web Server 再通过 HTTP 或 HTTPS 或 XML 将信息发送到客户端。

3.3.2 项目的概要设计

服务器端程序接收客户端发送的查询内容,连接到 Access 数据库上使用 SELECT 语句进行查询,并将查询结果返回给客户端。其算法设计如图 3-4 所示。

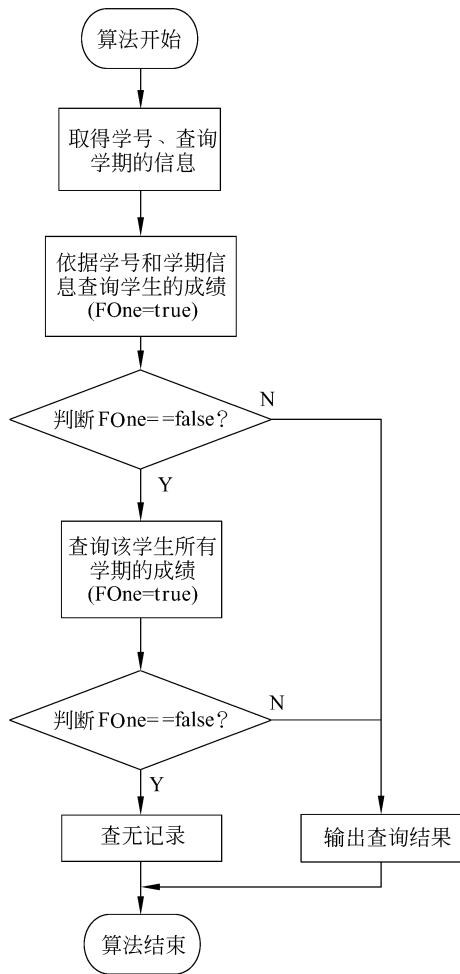


图 3-4 查询成绩算法流程图

3.4 客户端程序(MIDlet)的设计及实现

客户端程序(MIDlet)的设计包括用户界面的设计、连接服务器(网络编程)两大方面。当然,在进行具体的设计和实现工作之前,必须先创建一个工程。

3.4.1 创建工程

在 J2ME Wireless Toolkit 中创建一个名为 ScoreQuery(成绩查询)的工程,如图 3-5 所示。

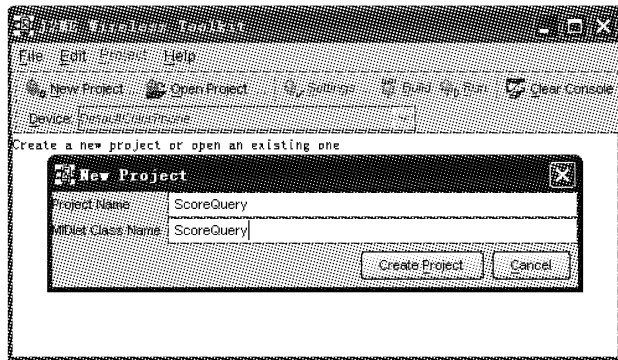


图 3-5 创建一个名为 ScoreQuery 的工程

接下来,还必须完成下面的两个工作:

(1) 在 J2ME Wireless Toolkit 的安装目录下找到 apps 目录,在 apps 目录下找到刚刚创建的工程目录 ScoreQuery,在其中的 src 目录中创建一个名为 ScoreQuery.java 的文件作为该项目的 Java 源文件。后面的客户端程序的实现就是通过编辑这个 Java 源文件来进行的。

(2) 将该工程用到的图片文件 welcome.png 和 symbol.png 文件都放在 ScoreQuery 目录下的 res 目录中,这两个 PNG 文件将分别用作该工程的欢迎图片和学期的图标图片,如图 3-6 所示。



图 3-6 工程 ScoreQuery 所用到的图片

该工程中所用到的所有图片格式仍然是 MIDlet 所支持的 PNG 图像格式。

3.4.2 用户界面设计

在移动设备终端,由于资源等条件的限制,要尽量减少使用对象的个数。同时,为了客户能够方便快捷地操作,客户端的设计也需要尽量地简化和“傻瓜”化。基于以上两点因素的考虑,在移动学生成绩查询系统的客户端,分别使用了 Alert 类、Ticker 类、Form 类、TextField 类、List 类和 Command 类。其中,在系统打开时,通过一个 Alert 对象描述软件的版权信息;Ticker 对象用于显示软件的提示信息;用户通过在 Form 对象中放置的 TextField 对象输入所要查询的学生学号,以及通过 List 对象选择所要查询的学期,以此输入查询内容;查询结果显示在另外一个 Form 对象中;在需要增加按钮的地方通过 Command 对象类实现。客户端用户界面设计如图 3-7 所示。

3.4.3 连接服务器

1. HTTP 协议的工作原理

从 3.3.1 小节的移动电子商务应用系统的架构图(图 3-3)中可以看出,J2ME 客户端



图 3-7 移动学生成绩查询系统的客户端用户界面设计

程序是通过连接到 J2EE 服务器端程序来获取信息的。同样地,在本章的移动学生成绩查询系统中,客户端程序(MIDlet)通过 HTTP 协议向服务器端程序(Servlet)发出查询学生成绩请求,服务器端程序接收到请求后,将查询结果(响应)返回给客户端。

MIDP 2.0 的规范要求支持 HTTP 协议,它是一种获取浏览器上的网页、图像、文件和表格等信息的协议。HTTP 协议是一种请求—响应协议:客户端发送请求,服务器端回复响应,其工作原理如图 3-8 所示。

HTTP 的工作步骤如下:

- (1) 客户端发送请求到服务器。
- (2) 服务器接受客户端发送的请求。
- (3) 采用 doGet()/doPost()方法来访问网页、Servlet、JSP 等。
- (4) 网页、Servlet、JSP 等响应请求,以 HTML 格式发送到服务器。
- (5) 服务器将响应请求通过网络发送到客户端。
- (6) 客户端得到请求内容并关闭数据流。

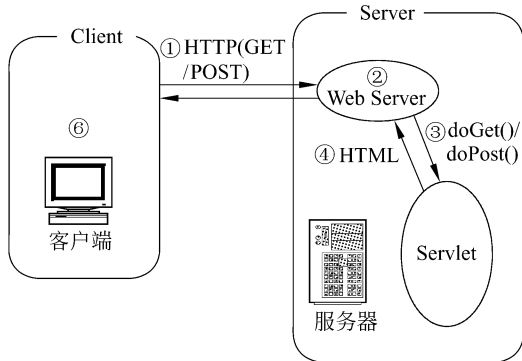


图 3-8 HTTP 的工作原理

使用 HTTP 协议连接服务器使得客户端和服务端之间可以保持连接状态(keep alive)。MIDlet 和 Web 服务器间始终使用同一个 HTTP 连接来多次传递数据。这种特性非常适合于像成绩查询这样的应用系统。但同时也意味着服务器不能初始化客户端的连接,除了客户端向它发出请求之外别无他法。

在 MIDP 中,使用 HTTP 协议的 API 是 HttpURLConnection 接口,它是 javax.microedition.io 包提供的一个接口,它扩展了 ContentConnection 接口,定义了 HTTP 连接所需要的方法和常量。

2. 客户端连接服务器端程序的设计

本系统采用 HttpURLConnection 接口来进行网络连接。在 MIDlet 进行 HTTP 编程的基本步骤如下:

(1) 使用 Connector 类的静态方法 open,将指向简单的 HTML 文档或 Web 服务器应用的地址传递给 open 方法的 url 参数。

(2) 将 Connector.open() 方法返回的 Connector 类型的数据强制转换为 HttpURLConnection 类型的数据。

(3) 使用 setRequestMethod 方法设置请求方式,使用 GET、POST 或 HEAD 这三种请求方式中的一种向服务器发出请求。

(4) 使用 setRequestProperty 方法进行必要的请求属性的设置。

(5) 如果连接成功,则客户端可以向服务器发送请求内容(对于 POST 请求方式);或者接收服务器的响应数据;或者调用服务器端的应用程序,例如 Servlet/JSP,或 ASP、CGI 等。

(6) 关闭 HTTP 连接。

本系统中,客户端连接服务器程序的设计遵循以上的几个步骤。详细介绍如下:

步骤一:导入包。由于在网络设计过程中需要使用 HttpURLConnection 接口,因此,必须导入 javax.microedition.io 这个包。同时,数据在网络传输过程中需要输入流和输出流,因此也必须导入 java.io 包。

```
import java.io.*;
import javax.microedition.io.*;
```

步骤二:设置请求方式。本系统采用的是 POST 的请求方式,因为客户端程序需要传递学生学号、所查询的学期这两个参数到服务器,使用 POST 方法较为简单、方便。当然,如果只有一个参数时,使用 GET 方法可能较为方便。

注 默认的请求方式是 GET。GET 请求方式从服务器获得静态的资源(如果重复同样的 GET 请求,获得的响应应该是一样的)。对于 GET 方式,输入的值与 URL 一起