

基于 J2ME/J2EE 移动预约挂号系统的设计与实现

任晓尘, 孙 涌

(苏州大学 计算机科学与技术学院, 江苏 苏州 215006)

摘 要: 医院长期存在的“三长两短”中的挂号排队问题已经成为限制医院发展的瓶颈。文中将移动应用技术与门诊预约挂号相结合, 设计了移动预约挂号系统, 方便了患者的同时也为医院带来了便利。考虑到可维护性、可重用性与可扩展性, 系统采用了 J2EE 四层架构; 针对移动预约挂号系统特点, 在移动客户端应用程序的设计上采用了 MVC 等设计模式, 这些设计模式改善了应用程序结构的同时提高了系统性能; 接着给出了系统各层关键部分的详细设计与实现; 最后展示了系统的运行效果。

关键词: J2ME/J2EE; MVC; 预约挂号

中图分类号: TP399

文献标识码: A

文章编号: 1673-629X(2006)12-0187-03

Design and Implementation of Mobile Appointment System Based on J2ME/J2EE

REN Xiao-chen, SUN Yong

(School of Computer Science & Technology, Soochow University, Suzhou 215006, China)

Abstract: The problem of “Three Long and Two Short” in hospital’s appointment becomes more and more important nowadays. The article applies the mobile application technique to hospital appointment and designs the mobile appointment system. It can bring convenience for both hospitals and patients. With consideration of maintaining, reusing and extending of the system, J2EE 4-tier architecture is adopted in the system. On the designing of mobile client application, different design patterns are adopted as MVC because of the characteristic of mobile appointment system. Then the article describes the designing and implementing process of key part of system in details. At last, the dissertation demonstrates the running effect of the mobile appointment system.

Key words: J2ME/J2EE; MVC; appointment

0 引言

国内医疗改革不断深入使得各医院之间的竞争也不断加剧, 患者对医院要求也逐渐提高。2005年4月, 卫生部与国家中医药管理局联合下文, 要求大中型医院要“创造条件开展预约挂号服务”。预约挂号, 最大的好处是节省时间, 高效利用医疗服务资源。

随着无线技术的发展, GPRS 网络在全球的普及和移动设备无线功能的增强, 使得基于 GPRS 通信的应用成为可能, 另外, 基于 Java 技术的 J2ME 手机设备日益体现出其功能强大、易于扩展等优势, 成为下一代移动计算的首选。文中提出基于 GPRS 和 J2ME/J2EE 构建移动预约挂号系统, 使病人随时随地都能与医院的挂号系统进行联络, 选择科室、专家、就诊时段, 实施预约挂号, 满足了病人对挂号方式的选择, 实现了医院门诊的挂号系统的多样性。

1 技术背景

1.1 J2ME 体系结构

J2ME 开辟了移动互联网新的应用环境, 能更好地为用户提供全新图形化、动态化的移动增值服务^[1,2]。J2ME 包括两种类型的组件, 即配置(configuration)和概要(profile)^[3]。配置是一系列低层次的 API 和一种为系列设备优化的虚拟机。概要是一种规范, 它详细描述了架构在配置之上并使用配置的一系列 API, 它们一起为专用设备类提供完整的运行环境。

遵循 J2ME 规范, 使用移动通讯设备概要(Mobile Information Device Profile)开发移动设备上使用的 MIDlet (指那些使用由 MIDP 和 CLDC 规范定义的 API 编写的应用程序)是无线 Internet 领域中的一种流行趋势, 它构建在 CLDC 之上, 为那些运行在像移动电话和 PDA 的设备上的应用提供完整的 J2ME 运行环境。MIDP 致力于解决用户界面、持久存储、联网和应用程序文件生命周期这样的问题。

1.2 J2EE 架构

J2EE 是当前用于快速构建企业分布式系统——电子

收稿日期: 2006-03-16

作者简介: 任晓尘(1980-), 男, 山西大同人, 硕士研究生, 研究方向为软件工程。

商务的最新技术^[4],是使用 Java 技术开发企业级应用的一种事实上的工业标准,它是 Java 技术不断适应和促进企业级应用过程中的产物。Sun 推出 J2EE 的目的是为应用 Java 技术开发服务器端应用提供一个平台独立的、可移植的、多用户的、安全的和基于标准的企业级平台,从而简化企业应用的开发、管理和部署。J2EE 规范定义了四种不同类型的容器。

(1)Applet 容器:运行和管理 Applet;

(2)应用客户端容器:运行和管理标准 Java 应用客户端程序;

(3)Web 容器:运行和管理表现逻辑层的 Servlet 组件和 JSP 组件;

(4)EJB 容器:运行和管理业务逻辑层的 EJB 组件。

2 移动预约挂号系统设计与实现

移动预约挂号系统(MAS, Mobile Appointment System)是一个基于 J2ME 和 J2EE 平台以及 GPRS 网络的无线应用系统,用户只需拥有支持 J2ME 的手机,即可在移动网络覆盖的区域内进行预约挂号,实现挂号、查询及取消等基本功能。它进一步扩展、增强和完善了现有预约挂号系统,实现了随时随地(只要无线网络覆盖)轻松挂号,减少病人即时挂号、排队等候的烦恼;实现了提前预约、随到随诊;还能减少患者在医院内非医疗性停留时间,减少患者的交叉感染机会,合理安排用户的就诊时间。

2.1 系统功能设计

移动预约挂号系统可以实现预约挂号、挂号查询、取消预约等基本功能,图 1 描述了本系统的用例。移动预约挂号系统与网上预约挂号系统的用例图并无明显差别,是因为移动预约挂号系统是基于现有预约挂号系统之上增强的应用,它并非是要取代现有的预约挂号系统或脱离现有的预约挂号系统而独立存在。

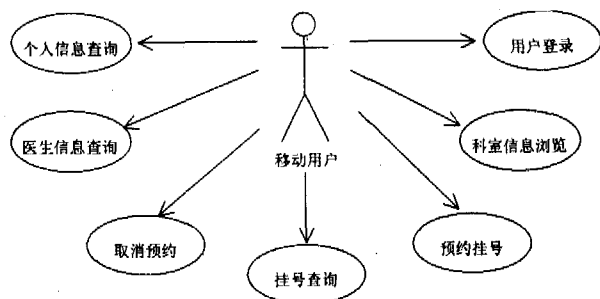


图 1 移动预约挂号系统用例图

2.2 系统总体架构

移动预约挂号系统体系结构采用基于 J2ME/J2EE 四层体系结构,即客户层/应用表示层/应用逻辑处理层/应用数据存储层,所有的访问并不直接访问后台的应用资源和数据资源,而是通过中间层进行访问,这样既可以保证后台数据的安全性,又可以通过中间层的调度保证性能和

事务的一致性。图 2 为本系统的总体架构图,左侧为 J2ME 客户层,右侧为应用服务器层(包括 Web 层与 EJB 层)及医院信息系统层。

2.3 详细设计与实现

移动预约挂号系统是预约挂号系统在移动设备上的扩展,功能上与网上预约挂号系统相似,性能上却存在较大差异。首先,移动设备的局限性为无线应用程序的设计带来了诸多限制,如:屏幕小、内存小、输入性能差、处理能力低等;其次,由于依赖于无线网络,使得系统 MIDP 应用程序面临的限制更为明显,因为无线网络带来的限制要比普通浏览器连接有线网络连接到互联网时所遇到的限制大得多,例如:带宽有限、等待时间较长、时断时续的连接等。为了解决这些限制,系统 MIDP 应用程序设计时应采取以下对策:只在需要时连接网络;只使用必需的网络数据;在断开连接时保持有用;进行相应的性能优化。

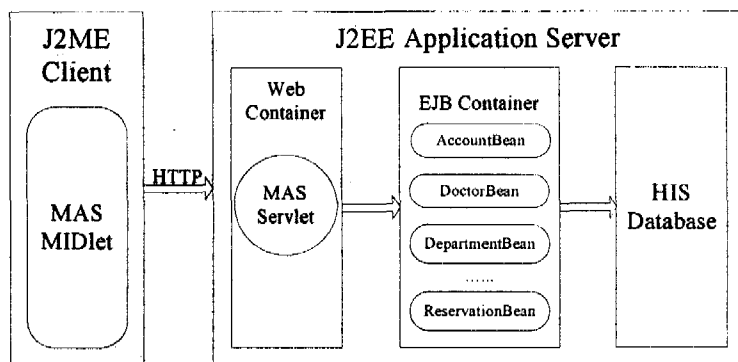


图 2 移动预约挂号系统总体架构图

2.3.1 客户端设计

在移动预约挂号系统的 MIDP 应用程序中,会出现多个用户界面,如用户登陆界面(LoginUI)、科室列表界面(DepartmentUI)和医生信息列表(DoctorUI)等,大量的用户界面使得界面导航变得异常繁杂,而经典的 MVC(Model-View-Controller,模型-视图-控制器)^[5]可用于界面导航,它采用“分治”的思想,即分离用户界面和用户数据,因为用户界面的更改通常要比数据存储系统的更改频繁得多。移动预约挂号系统在客户端采用了 MVC 模式。

(1)模型:用于管理应用程序的行为和数据,并响应来自视图的请求与控制器发出的更改状态的命令。MIDP 中使用 MVC 的一个最重要的原因:为了节约使用网络并在断开连接时仍保持有用,客户机必须确定何时从服务器获得数据,何时从本地存储中获得数据。为此本系统的模型包括本地数据模型和远程数据模型。本地数据模型(LocalModel)用于对本地存储数据(如离线状态个人基本信息)进行存取、删除等各种处理;远程数据模型(Remote-Model)主要封装了与服务器端数据的各种业务处理,这里采用 ModelFacade 门面模型提供对控制器的统一接口。

(2)视图:用于管理信息的显示,它是模型的外在表现。如:主菜单界面(MainMenuUI)、用户登陆界面

(LoginUI)、科室列表界面(DepartmentUI)和医生信息列表(DoctorUI)等。

(3)控制器:提取通过视图传输进来的外部信息,并将其转化成相应事件,对模型进行更新;同时,模型的更新与修改也将通过控制器来通知视图,从而保持视图与模型的一致性。本系统中对应于图 3 中的 UIController。

2.3.2 服务器端设计

移动预约挂号系统应用服务器端采用了 J2EE 的多层架构,包括应用表示层(Web 层)与业务逻辑层(EJB 层)。本系统采用 JBOSS 服务器,支持 EJB 2.0 的规范,是一个管理 EJB 容器的服务器,核心服务是提供 EJB 服务器。JBOSS 不包括 Servlet/JSP page 的 Web 容器,在本系统开发中绑定 Apache Tomcat 作为系统的 Web 服务器。

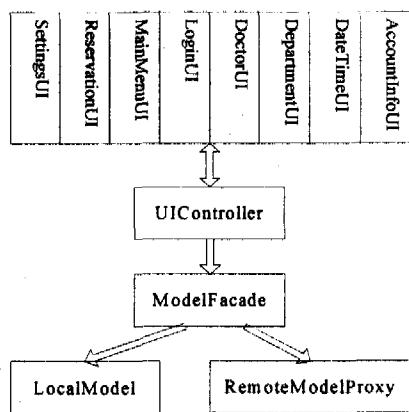


图 3 移动预约挂号系统客户端 MIDP 结构

(1)Web 层。

Web 服务器作为服务器系统的访问入口,处理用户发送的 HTTP 请求和结果数据的发送模块作为服务器端的 Web 模块,也是 EJB 的客户端,负责接受用户请求和发送响应,具体通过 MASServlet 实现。MASServlet 中方法列表如图 4 所示。

```

◆ handleCall() : int
◆ doPost() : void
◆ cancelReservation() : void
◆ createAccount() : void
◆ getDepartments() : void
◆ getDepartmentSchedule() : void
◆ getDoctor() : void
◆ getDoctorShowtimes() : void
◆ getLocales() : void
◆ getResourceBundle() : void
◆ login() : void
◆ reserveDoctor() : void
◆ updateAccount() : void

```

图 4 MASServlet 方法列表

MASServlet 中 doPost()方法代码如下:

```

protected void doPost (HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{ HttpSession session = request.getSession(true);
MASBD masBD =

```

```

(MASBD) session.getAttribute(SESSION_ATTRIBUTE_
MAS_BD);
response.setContentType("application/octet - stream");
try { int contentLength =
handleCall (masBD, request.getInputStream(), response.
getOutputStream());
response.setContentLength(contentLength);
if (session.isNew()) {
response.setStatus(HttpServletResponse.SC_CREATED);
response.setHeader("Location",
response.encodeURL(request.getRequestURL().toString
()));
}
} catch (ApplicationException ae) {
ae.printStackTrace();
response.setStatus(HttpServletResponse.SC_INTERNAL_
SERVER_ERROR);
}
finally {}
return;
}

```

(2)EJB 层。

应用服务器负责完成整个系统的业务逻辑处理,如预约挂号、挂号查询、取消预约等。每个具体的功能操作由一个具体的逻辑处理单元负责。每个逻辑处理单元基于组件技术设计,使系统具有灵活性、可扩展性和逻辑复用性。业务逻辑处理根据 Web 服务器解析的 XML 消息内容进行 JNDI 调用,根据消息中的参数信息进行计算处理,并且根据需要访问数据库服务器,完成增、删、改、查等数据库操作。本系统设计并实现了 AccountBean, DoctorBean, DepartmentBean 和 ReservationBean。AccountBean 是一个实体 Bean,负责用户信息认证。DoctorBean 与 DepartmentBean 类似,均为实体 Bean,分别用于记录医生信息和科室信息。ReservationBean 是一个无状态会话 Bean,负责挂号业务,包括预约挂号、挂号查询及取消预约。

2.3.3 医院信息系统(HIS)层

数据库作为数据仓库存储系统中所用的静态信息,如用户基本信息、医生出诊信息等。应用服务器以组件的形式封装数据库访问方法,该组件通过 JDBC 数据库接口驱动访问数据记录,并且对外提供访问机制。本系统中使用 MySQL 数据库。MySQL 是一种完全网络化的跨平台关系型数据库系统,同时也是基于客户机/服务器 CS 体系结构的分布式数据库管理系统。它具有功能丰富、使用简便、管理方便、运行速度快、安全可靠性强等优点,同时 MySQL 数据库还支持多种语言编写程序访问。

3 运行效果

图 5 为系统运行效果部分界面展示,依次为科室选择

(下转第 192 页)

```

Temp=SR;
Temp=SR && 0x04
}
while (Temp!=0x04);
EFF=0x81; //发送扩展帧格式数据,数据长度 1 个字节
ID0=0x00; //标识符赋数值
ID1=0x00;
ID2=0x00;
ID3=0x00;
TXDATA1=0x55;
CMR=0x01; //启动 SJA1000 发送 /* 01 */
}

```

2.3 接收程序模块

SJA1000 自动接收信息,信息被接收放入接收缓冲器,并将接收缓冲器状态标识位置“1”,AT89S52 将信息保存在程序设置的信息缓冲器中,通过置位 SJA1000 的命令寄存器来释放接收缓冲器,即 CMR=0x04,接收程序的设计也可以通过中断方式和检测 SJA1000 的标识控制方式,但由于接收出于被动,如采用检测法需要耗费大量时间检测接收状态,所以采用中断方式进行控制来提高实时性^[5]。下面就以中断方式说明:如果接收到信息,信息通过接收过滤并存放在接收队列中,产生一个接收中断后 AT89S52 立即把信息送到信息寄存器,并写入 CMR=0x04,发送释放接收缓冲器,下一条信息进入接收缓冲器前将会产生新的中断后进入。

```
void recievedata(void) interrupt 0 using 0
```

```

temp=temp&&XBYTE[0x7F10];
temp=temp && 0x0f;
temp=temp+4;
for(i=0;i<temp;i++)
{
a[i]=XBYTE[0x7F10+i];
}
CMR=0x04;
}

```

3 结束语

系统应用于多主对等系统中,提高数据传输的实时性、可靠性,满足了国内市场的一定需求,为其他工程应用也提供了可借鉴的例证。

参考文献:

- [1] Philips semiconductors. SJA1000 stand-alone CAN controller [R]. Holand: Philips, 2000.
- [2] Atmel Corporation. AT89S52 Datasheets[EB/OL]. 2001. <http://www.21ic.com/search/pdf/download.asp?id=137366>.
- [3] 吴繁红. 基于 AT89C51 和 SJA1000 控制器实现 CAN 总线接口[J]. 电子技术应用, 2003, 29(7): 51-52.
- [4] 饶运涛, 邹继军, 郑勇芸. 现场总线 CAN 原理与应用技术[M]. 北京: 北京航空航天大学出版社, 2003.
- [5] 张培仁, 孙占辉, 张欣, 等. 基于 C 语言编程 MCS251 单片机原理与应用[M]. 北京: 清华大学出版社, 2003.

(上接第 189 页)

界面、预约挂号界面和成功返回界面。

与现有的挂号方式相比,更具亲和力、吸引力和竞争力,能够在新经济时代为医患双方创造共同价值。

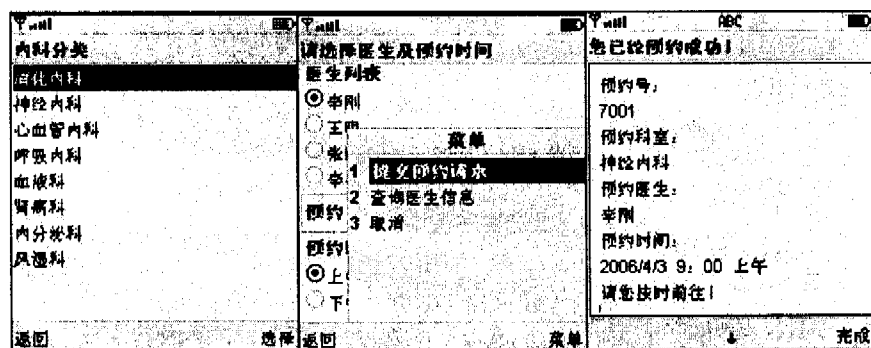


图 5 系统运行效果展示

4 结束语

医院预约服务为患者提供了一种人性化服务,体现了以患者为中心的服务宗旨,为医院节约了大量的人力和物力,提高了医院的经济效益和社会效益。移动挂号的出现并非为了完全取代传统挂号方式以及网上挂号,恰恰相反,它是以上挂号方式的扩充和弥补。移动预约挂号方式

参考文献:

- [1] 吴延昌, 冯萍, 苏聪. 基于 J2ME 技术的应用开发与研究[J]. 微电子学与计算机, 2005 (9): 71-73.
- [2] 陈赞. 嵌入式无线公交查询系统的设计与研究[J]. 计算机应用研究, 2005(7): 185-187.

- [3] 詹建飞. J2ME 开发精解[M]. 北京: 电子工业出版社, 2006.
- [4] Moertiyoso N, Yow K C. Designing wireless enterprise applications on mobile devices[D]. Singapore: School of Computer Engineering, NTU, 2002.
- [5] Gamma E. 设计模式—可复用面向对象软件基础[M]. 李英军, 等译. 北京: 机械工业出版社, 2000.