



(12) 发明专利申请

(10) 申请公布号 CN 103246720 A

(43) 申请公布日 2013.08.14

(21) 申请号 201310156461.3

(22) 申请日 2013.04.28

(71) 申请人 西安交通大学

地址 710049 陕西省西安市咸宁西路 28 号

(72) 发明人 李凡 王明慧 张丹阳 李娜

(74) 专利代理机构 西安通大专利代理有限责任
公司 61200

代理人 汪人和

(51) Int. Cl.

G06F 17/30 (2006.01)

G06Q 50/12 (2012.01)

权利要求书2页 说明书8页 附图2页

(54) 发明名称

一种基于移动终端的餐厅推荐及点餐方法

(57) 摘要

本发明公开了一种基于移动终端的餐厅推荐及点餐方法,通过餐厅端、移动终端与服务器端的配合,实现了对于大部分餐厅信息的整合,并且根据用户需求向其推送距其最近的餐厅供用户选择;同时用户能够选择其中某个餐厅完成点菜和预订餐桌座位的功能,显著的提高了就餐、订餐的效率。

1. 一种基于移动终端的餐厅推荐及点餐方法,其特征在于,包括以下步骤:

移动终端根据自带 GPS 获取用户的实时位置信息,同时移动终端向服务器端发送用户的实时位置信息;

服务器端接收移动终端发送的用户的实时位置信息,然后将所述实时位置信息与服务器端的数据库中存储的餐厅位置信息进行比对,然后根据用户的实时位置与餐厅位置的距离选择出备选餐厅,并按照用户的选择将备选餐厅的餐厅信息返回给移动终端供用户查看,餐厅信息包括在服务器端维护的餐厅的实时菜谱信息、餐桌座位使用信息和用户订单有效性信息;

备选餐厅以地图标注形式或者以列表的形式显示在移动终端上,用户通过移动终端选择、查看备选餐厅的餐厅信息,同时,移动终端根据用户的选择生成相应的订单,然后将订单发送给服务器端,订单包括点菜和预订餐桌及座位的请求;

各餐厅管理人员通过餐厅端的 JSP 页面对餐厅信息进行维护并反馈给服务器端,维护包括修改餐厅基本信息,添加、删除和修改菜谱信息,以及根据订单状态更新餐桌和座位使用状态信息。

2. 如权利要求 1 所述一种基于移动终端的餐厅推荐及点餐方法,其特征在于:

服务器端接收移动终端发来的用户的实时位置信息,然后将所述实时位置信息解析为经纬度坐标值,将所述经纬度坐标值与数据库中存在的餐厅经纬度坐标值进行比对,然后选择出距离用户较近的一系列餐厅作为备选餐厅。

3. 如权利要求 1 所述一种基于移动终端的餐厅推荐及点餐方法,其特征在于:

移动终端向服务器端发送注册和登录请求,然后根据服务器端返回的结果得到注册或者登录是否成功的信息,用户登录成功后,移动终端向服务器端发送用户的实时位置信息。

4. 如权利要求 1 所述一种基于移动终端的餐厅推荐及点餐方法,其特征在于:

移动终端根据用户的实时位置信息和服务器端返回的餐厅位置信息得到用户和餐厅的经纬度坐标信息,然后将用户和餐厅的位置信息标注在地图上,当服务器端接收到移动终端发来的用户查看相应餐厅信息的请求,服务器端根据用户的请求参数利用相应的 HQL 语言检索数据库,然后从数据库中检索出的数据封装成 JSON 格式的数据,然后将 JSON 格式的数据以响应用户请求的方式发送给移动终端,移动终端通过解析 JSON 格式的数据得到用户需要查看的餐厅信息,并将餐厅信息以列表的形式展现出来。

5. 如权利要求 1 所述一种基于移动终端的餐厅推荐及点餐方法,其特征在于:

用户通过移动终端查看备选餐厅信息,以及各备选餐厅的菜谱和空余餐桌座位信息,然后选择其中一个备选餐厅进行就餐,移动终端生成订单并发送给服务器端,订单包括用户编号,餐厅编号和订单生成时间,服务器端接收到订单后将相应的数据存入数据库中,同时服务器端开启另外一个用于计时的线程,自订单生成超过设定时间且用户没有前往餐厅则服务器端自动将该订单设置为无效状态。

6. 如权利要求 5 所述一种基于移动终端的餐厅推荐及点餐方法,其特征在于:

用户首先通过移动终端向服务器端发送订单,然后在通过移动终端浏览菜谱的同时,点击菜谱列表选择是否将菜加入到订单中,然后向服务器端发送数据,服务器端将数据存入用户订单详细列表中;用户在浏览餐厅空余餐桌和座位的同时选择是否预订餐桌和座位,并向服务器端发送数据,服务器端将用户的预订餐桌和座位的信息存入订单详细列表

中。

7. 如权利要求 1 所述一种基于移动终端的餐厅推荐及点餐方法,其特征在于:移动终端使用 Http 协议实现移动终端到服务器端的通信,由移动终端解析用户的操作、翻译成 Http 请求,服务器端通过 Struts2 进行请求拦截,并执行相应的业务。

8. 如权利要求 1 所述一种基于移动终端的餐厅推荐及点餐方法,其特征在于:

服务器端在取得餐厅授权后通过数据库保存餐厅信息,每个餐厅通过 JSP 页面管理和维护各自的信息。

一种基于移动终端的餐厅推荐及点餐方法

技术领域

[0001] 本发明属于移动互联网领域,涉及一种基于移动终端的餐厅推荐及点餐方法。

背景技术

[0002] 在如今生活节奏加快的年代,人们的生活方式发生了很大的变化,这些变化体现在衣食住行等方方面面。其中吃饭一直是人们较为关心的话题,现在人们选择就餐的方式有很多种,有订餐上门服务,电话或者短信提前预定餐厅的,这些方法都有其自身的优点和缺点。现有的移动点菜软件存在如下两个问题:现有的移动点菜软件大多数只是针对特定餐厅的,即每个餐厅开发属于本餐厅的软件,该软件只能在一个餐厅内进行查看,预订等操作,这样就给用户带来了选择的复杂性,即用户需要为每个餐厅安装不同的软件,这无疑增加了用户的负担。除此之外,大部分的软件能够调用地图并将餐厅显示出来,但是用户还需要在地图上找到自己的位置,并且寻找最近的餐厅,这样在无形之中也增加了用户的时间消耗。

发明内容

[0003] 本发明的目的在于提供一种基于移动终端的餐厅推荐及点餐方法。

[0004] 为达到上述目的,本发明采用了以下技术方案:

[0005] 移动终端根据自带 GPS 获取用户的实时位置信息,同时移动终端向服务器端发送用户的实时位置信息;

[0006] 服务器端接收移动终端发送的用户的实时位置信息,然后将所述实时位置信息与服务器端的数据库中存储的餐厅位置信息进行比对,然后根据用户的实时位置与餐厅位置的距离选择出备选餐厅,并按照用户的选择将备选餐厅的餐厅信息返回给移动终端供用户查看,餐厅信息包括在服务器端维护的餐厅的实时菜谱信息、餐桌座位使用信息和用户订单有效性信息;

[0007] 备选餐厅以地图标注形式或者以列表的形式显示在移动终端上,用户通过移动终端选择、查看备选餐厅的餐厅信息,同时,移动终端根据用户的选择生成相应的订单,然后将订单发送给服务器端,订单包括点菜和预订餐桌及座位的请求;

[0008] 各餐厅管理人员通过餐厅端的 JSP 页面对餐厅信息进行维护并反馈给服务器端,维护包括修改餐厅基本信息,添加、删除和修改菜谱信息,以及根据订单状态更新餐桌和座位使用状态信息。

[0009] 服务器端接收移动终端发来的用户的实时位置信息,然后将所述实时位置信息解析为经纬度坐标值,将所述经纬度坐标值与数据库中存在的餐厅经纬度坐标值进行比对,然后选择出距离用户较近的一系列餐厅作为备选餐厅。

[0010] 移动终端向服务器端发送注册和登录请求,然后根据服务器端返回的结果得到注册或者登录是否成功的信息,用户登录成功后,移动终端向服务器端发送用户的实时位置信息。

[0011] 移动终端根据用户的实时位置信息和服务器端返回的餐厅位置信息得到用户和餐厅的经纬度坐标信息,然后将用户和餐厅的位置信息标注在地图上,当服务器端接收到移动终端发来的用户查看相应餐厅信息的请求,服务器端根据用户的请求参数利用相应的 HQL 语言检索数据库,然后将从数据库中检索出的数据封装成 JSON 格式的数据,然后将 JSON 格式的数据以响应用户请求的方式发送给移动终端,移动终端通过解析 JSON 格式的数据得到用户需要查看的餐厅信息,并将餐厅信息以列表的形式展现出来。

[0012] 用户通过移动终端查看备选餐厅信息,以及各备选餐厅的菜谱和空余餐桌座位信息,然后选择其中一个备选餐厅进行就餐,移动终端生成订单并发送给服务器端,订单包括用户编号,餐厅编号和订单生成时间,服务器端接收到订单后将相应的数据存入数据库中,同时服务器端开启另外一个用于计时的线程,自订单生成超过设定时间且用户没有前往餐厅则服务器端自动将该订单设置为无效状态。

[0013] 用户首先通过移动终端向服务器端发送订单,然后在通过移动终端浏览菜谱的同时,点击菜谱列表选择是否将菜加入到订单中,然后向服务器端发送数据,服务器端将数据存入用户订单详细列表中;用户在浏览餐厅空余餐桌和座位的同时选择是否预订餐桌和座位,并向服务器端发送数据,服务器端将用户的预订餐桌和座位的信息存入订单详细列表中。

[0014] 移动终端使用 Http 协议实现移动终端到服务器端的通信,由移动终端解析用户的操作、翻译成 Http 请求,服务器端通过 Struts2 进行请求拦截,并执行相应的业务。

[0015] 服务器端在取得餐厅授权后通过数据库保存餐厅信息,每个餐厅通过 JSP 页面管理和维护各自的信息。

[0016] 与现有技术相比,本发明具有以下有益的技术效果:

[0017] 本发明所述基于移动终端的餐厅推荐及点餐方法是一种基于移动终端的餐厅推荐及点餐系统,其通过移动终端、餐厅端与服务器端的配合完成向用户推送周围最近餐厅信息,用户选择餐厅进行点菜、预订餐桌座位以及餐厅维护餐厅数据的功能。

[0018] 本发明通过移动终端应用和服务器端应用配合实现应用整体功能,采取终端请求服务器端响应的模式;移动终端作为连接用户与服务器中数据的桥梁,主要负责与用户的交互工作,并将服务器返回的数据显示出来;服务器端应用主要负责处理数据,响应移动终端的请求对数据库进行检索和读写操作,并反馈移动终端所需要的数据。

[0019] 本发明提供基于移动终端的餐厅推荐及点餐方法,移动终端通过向服务器发送请求能获取并显示餐厅的信息,同时能完成用户的点菜和餐桌座位预订功能。移动终端调用 GPS 得到用户的实时位置,为服务器的推送周围餐厅信息提供经纬度位置数据。得到了服务器推送的最近餐厅信息后,终端将餐厅信息以及用户位置同时在地图上面直观的标注出来,能够帮助用户最快地做出选择。移动终端能够进行用户注册、登陆和修改用户信息等用户业务,并与服务器端交互这些信息。

[0020] 本发明提供基于移动终端的餐厅推荐及点餐方法,餐厅端通过 JSP 页面向服务器发送请求修改餐厅基本信息,和更新餐厅的动态信息。

[0021] 本发明提供的基于移动终端的餐厅推荐及点餐方法,服务器端聚合了大部分的餐厅信息,包括餐厅基本信息,菜谱信息和空余餐桌座位信息等,根据移动终端发送的用户实时位置信息,系统自动向用户推送最近的餐厅供用户选择。同时服务器端开启后台服务进

程对用户生成的订单有效性进行监视,若订单超出有效时限则将订单标示为失效。服务器端维护管理着用户的基本信息,响应移动终端的请求完成用户的登录注册和修改等功能。

[0022] 本发明用户通过 GPS 技术获取当前位置信息并通过互联网向服务器发送请求查询周边较近餐厅信息,最后基于移动终端与服务器的交互来完成点菜以及餐桌座位预订功能。

[0023] 本发明克服了现有技术存在的问题,通过将很多餐厅的信息进行聚合,包含了各个餐厅的基本信息、菜谱信息以及空余餐桌和座位信息。用户只需移动终端就可以完成对于大部分餐厅的预订功能;系统调用 GPS 功能获取用户的实时位置,并根据此向用户推送距离其最近的餐厅,同时在地图上显示出来,显著的提高了就餐、订餐的效率。

附图说明

[0024] 图 1 是本发明的工作流程图。

[0025] 图 2 是本发明的结构总体示意图。

[0026] 图 3 是移动终端应用示意图。

[0027] 图 4 是服务器端应用功能结构和请求响应流程图。

具体实施方式

[0028] 下面结合附图对本发明做进一步说明,所述是对本发明的解释而不是限定。

[0029] 一种基于移动终端的餐厅推荐及点餐方法,包括以下步骤:

[0030] 基于 GPS 技术并通过无线网络向移动终端提供点餐服务,移动终端根据自带 GPS 获取用户的实时位置信息,同时移动终端向服务器端发送用户的实时位置信息;

[0031] 服务器端接收移动终端发送的用户的实时位置信息,然后将所述实时位置信息与服务器端的数据库中存储的餐厅位置信息进行比对,然后根据用户的实时位置与餐厅位置的距离选择出备选餐厅,并按照用户的选择将备选餐厅的餐厅信息返回给移动终端供用户查看,餐厅信息包括在服务器端维护的餐厅的实时菜谱信息、餐桌座位使用信息和用户订单有效性信息;

[0032] 移动终端调用 Android 的地图功能将备选餐厅以可视化的地图标注形式或者以列表的形式显示在移动终端上,给用户以直观的感受,用户通过移动终端选择、查看备选餐厅的餐厅信息,同时,移动终端根据用户的选择生成相应的订单,然后将订单发送给服务器端,来完成预订功能,订单包括点菜和预订餐桌及座位的请求;

[0033] 各餐厅管理人员通过餐厅端的 JSP 页面对餐厅信息进行维护并反馈给服务器端,维护包括修改餐厅基本信息,添加、删除和修改菜谱信息,以及根据订单状态更新餐桌和座位使用状态信息。

[0034] 服务器端接收移动终端发来的用户的实时位置信息,然后将所述实时位置信息解析为经纬度坐标值,将所述经纬度坐标值与数据库中存在的餐厅经纬度坐标值进行比对,然后选择出距离用户较近的(比如以用户为中心周围 1 千米范围内)一系列餐厅作为备选餐厅。

[0035] 移动终端向服务器端发送注册和登录请求,然后根据服务器端返回的结果得到注册或者登录是否成功的信息,从而继续下一步的动作。用户登录成功后,移动终端向服务器

端发送用户的实时位置信息,同时,用户登录成功后,可以通过移动终端修改个人信息,进行用户订单管理(查看、删除订单)等一系列的操作;

[0036] 在服务器端对于移动终端的账户管理为:

[0037] 移动终端向服务器端发送注册或登录的请求,服务器端处理移动终端发送来的请求,进行用户业务的操作,将用户信息保存在数据库,进行判断并返回动作执行成功或者失败的响应;对于已经登录的用户可以选择是否修改个人信息并发送修改请求,服务器端处理移动终端发来的修改请求,并返回是否修改成功的响应。

[0038] 服务器端的数据库中包括以下数据库表:

[0039] 用户信息表,保存用户的基本账号信息,包括用户名,密码,联系方式等。

[0040] 餐厅基本信息表,保存所有的餐厅基本信息,包括餐厅名称,餐厅地址,餐厅经纬度坐标信息以及餐厅的介绍信息等。

[0041] 餐厅菜谱分类表,保存了各个餐厅的菜谱分类信息,包括所属餐厅编号(外键),分类名称。

[0042] 餐厅菜单表,保存了各个餐厅的详细的菜名信息,包括所属菜谱分类编号(外键),菜名,菜价格,菜描述。

[0043] 餐厅餐桌信息表,保存了各个餐厅的餐桌使用情况,包括所属餐厅编号(外键),餐桌编号,餐桌使用状态,餐桌容纳人数。

[0044] 餐厅座位信息表,保存了各个餐厅的餐桌所对应的座位的使用情况,包括所属餐桌编号(外键),座位使用状态。

[0045] 用户订单表,保存了用户的订单基本信息,包括订单编号,订单生成时间,所属餐厅编号(外键),订单是否有效。

[0046] 订单详细列表,保存了用户分别对应于各个订单的详细资源,包括所订菜编号,数量,餐桌编号,座位编号等。

[0047] 移动终端接收服务器端返回的数据响应,并且在地图上显示用户和周围餐厅:

[0048] 为了能给用户直观的位置信息感受,移动终端根据用户的实时位置信息和服务器端返回的餐厅位置信息得到用户和餐厅的经纬度坐标信息,然后调用 Android 系统的 Google maps 应用,并通过 Overlay (覆盖层)技术将用户和餐厅的位置信息标注在地图上,当服务器端接收到移动终端发来的用户查看相应餐厅信息的请求,例如周围餐厅信息,餐厅的菜谱信息和餐厅的餐桌座位空余状态信息等,服务器端根据用户的请求参数利用相应的 HQL 语言检索数据库,然后将从数据库中检索出的数据封装成 JSON 格式的数据,然后将 JSON 格式的数据以响应用户请求的方式发送给移动终端,移动终端通过解析 JSON 格式的数据得到用户需要查看的餐厅信息,即在数据库中检索的信息,并将这些餐厅信息以列表的形式展现出来。

[0049] 移动终端订单的生成:

[0050] 用户通过移动终端查看备选餐厅信息,以及各备选餐厅的菜谱和空余餐桌座位信息,然后选择其中一个备选餐厅进行就餐,移动终端首先弹出对话框确认用户是否选择该备选餐厅,得到用户确认后,然后移动终端生成订单并发送给服务器端,订单包括用户编号,餐厅编号和订单生成时间,服务器端接收到订单后将相应的数据存入数据库的用户订单表中,同时服务器端开启后台 Service,即启动另外一个用于计时的线程,即从订单生成

时开始计时,自订单生成超过一小时且用户没有前往餐厅,则服务器端自动将该订单设置为无效状态(这个动作是由服务器内部的线程自动完成的,自用户生成订单后服务器会时刻开启一个线程用于计时和修改用户订单有效性的状态)。

[0051] 移动终端的点菜以及餐桌座位的预订功能:

[0052] 用户首先通过移动终端向服务器端发送订单,然后在通过移动终端浏览菜谱的同时,点击菜谱列表中的选项即会弹出对话框提示选择是否将该菜加入到订单中,同时输入需要的份数,然后点击确认按钮向服务器端发送数据,服务器接收请求,将数据存入用户订单详细列表中;与之类似的是用户选择餐桌和座位的步骤,用户同样是在浏览餐厅空余餐桌和座位的同时选择是否预订餐桌和座位,并向服务器端发送数据,服务器端将用户的预订餐桌和座位的信息存入订单详细列表中。

[0053] 移动终端发送请求使用 Http 协议实现移动终端到服务器端的通信,由移动终端解析用户的操作、翻译成 Http 请求,服务器端通过 Struts2 进行请求拦截,并执行相应的业务;对于用户业务数据部分使用 HQL 语句进行用户通过用户名的查找实例。

[0054] 所述的移动终端与用户交互界面包括用户登录界面,用户注册界面,主菜单界面,餐厅信息查看界面,餐厅服务菜单界面,餐厅菜谱查看界面,餐厅空余餐桌空余座位查看界面等。

[0055] 用户登陆界面,用户在该界面通过输入用户名及密码登陆系统,同时该界面有注册按钮,点击注册按钮则进入注册界面。点击退出程序按钮则退出程序。

[0056] 用户注册界面,用户在该界面完成注册功能,输入用户名,密码及重复密码,联系方式等,将这些数据封装起来并发送至服务器,服务器对数据进行处理后存入用户数据表,即完成用户注册功能。

[0057] 主菜单界面,用户登录成功后即进入主菜单界面,可以选择以列表形式查看周围餐厅信息同时完成订单生成以及提前订菜,预订餐桌座位等操作;也可以选择以地图的形式查看周围餐厅信息,得到直观的印象;在主菜单界面,用户可以修改个人信息,在弹出的对话框中输入修改后的信息,请求服务器完成个人信息的修改操作。

[0058] 餐厅信息查看界面,在这个界面用户能够看到周围的餐厅信息以列表的形式展现出来,包括餐厅名称、餐厅地址等信息,分别点击则能进入到各个餐厅的服务菜单界面。

[0059] 餐厅服务菜单界面,用户在该界面能够完成查看菜谱,空余餐桌座位信息,以及点菜,预订餐桌座位等动作。

[0060] 餐厅菜谱查看界面,用户在该界面能够查看各个餐厅菜谱分类,以及详细的菜谱信息,包括价格,介绍,菜名等。

[0061] 餐厅空余餐桌空余座位查看界面,用户在该界面能够查看各个餐厅的实时餐桌和座位的使用情况,同时还能够预订空余餐桌和座位。被预定的餐桌和座位则将被设置状态为使用中。

[0062] 餐厅端对于餐厅基本信息和动态信息的维护:

[0063] 服务器端在取得餐厅授权后通过数据库保存餐厅信息,每个餐厅通过 JSP 页面管理和维护各自的信息,包括修改餐厅基本信息,增加、删除和修改菜谱信息以及实时更新餐厅餐桌和座位使用状态信息等。

[0064] 实施例

[0065] 一种基于移动终端的餐厅推荐及点餐系统,能够根据用户的实时位置为其推送周围较近的餐厅,用户在这些餐厅中间做出选择并完成点菜和餐桌座位的预订。分为移动终端、餐厅端和服务端这三部分配合以达到最终效果,移动终端采用 Android 平台,餐厅端采用 JSP 网页技术,服务器端采用的是 Struts2+Hibernate 架构的 java web 技术。

[0066] 参见图 1,所述基于移动终端的餐厅推荐及点餐系统,包括以下几个步骤:

[0067] 餐厅端通过 JSP 网页向服务器发送请求对餐厅的数据进行维护,服务器拦截请求,并分别处理请求数据,采用 HQL 语言对餐厅数据库进行检索,增加和更新操作。

[0068] 移动终端通过 Http-Post 的方式向服务器发送请求,请求包括登录注册,修改,以及查看餐厅信息,生成订单等。服务器端同样拦截终端请求,并对请求数据进行处理,返回响应给移动终端。

[0069] 服务器端分别处理餐厅端和移动终端的请求,完成对餐厅信息的维护;验证用户注册登录信息,根据终端的业务请求返回相应的数据,实现了向用户推送餐厅信息以及用户预订功能;同时服务器还实时地维护着用户的订单,从用户下单开始计时,超出时限则将订单设置为无效状态。

[0070] 参见图 2,餐厅端通过 JSP 网页向服务器发送数据来维护餐厅的信息,包括餐厅基本信息(餐厅名称,餐厅地址,餐厅介绍和餐厅经纬度坐标等);餐厅动态信息,包括(餐厅菜谱信息,餐厅实时餐桌和座位的使用情况)。当餐厅的菜谱发生变化时,餐厅向服务器发送请求,服务器接收餐厅数据对菜谱数据库进行修改。类似的,餐厅的实时餐桌和座位使用情况也由餐厅服务人员进行统计后发送至服务器,服务器对相应的数据表进行更新。

[0071] 服务器与移动终端的交互过程由图 2 来表示,系统的工作方式可以描述为:

[0072] 对于服务器端,主要功能可以分为以下几个部分,1、接收终端请求实现用户的注册登录即修改个人信息功能;2、接收终端用户实时位置数据并与数据库中的餐厅位置数据进行比较推送较近的餐厅至终端;3、当用户订单生成后,开启后台服务进程进行计时,当订单超出时限后,设置订单状态为失效;4、接收移动终端的点菜,预订餐桌座位的请求,并将请求数据存入订单详细列表中。

[0073] 对于移动终端来说,主要的作用是与用户进行交互,因此其核心部分是各种界面,对于本系统而言主要包含以下几个界面,参见图 3:

[0074] 用户注册登录界面,顾名思义,通过该界面用户能够完成登录系统以及注册成为会员的操作。

[0075] 程序主菜单界面,即用户登录成功后进入的界面,在该界面用户能够直观的了解本系统的主要功能:以列表形式显示周围餐厅信息及预订,以地图形式显示用户及餐厅信息,用户订单管理,用户个人信息修改,返回主菜单。

[0076] 周围餐厅查看界面,该界面由程序主菜单进入,在该界面里用列表的形式显示了距离用户较近的所有餐厅列表,包括餐厅名称,餐厅地址等信息。

[0077] 餐厅服务信息界面,该界面由餐厅查看界面进入,用户点击其中某个餐厅后即进入到该餐厅的服务信息界面。在该界面里用户能够看到餐厅所能提供的服务,包括菜单查询和点菜、空余餐桌座位查询和预订、选择本餐厅并且生成订单。

[0078] 菜单查询和点菜界面,该界面由餐厅服务信息界面进入,在该界面里用户能够看到餐厅的菜谱分类情况,点击其中一个分类则进入到菜谱详细界面,包括菜名,价格,描述

等。用户在浏览详细菜谱的同时点击某一项即会弹出对话框,提示用户是否选择该菜,用户输入份数后点击提交按钮后该菜就会加入到用户的订单详细列表中,从而完成了点菜操作。

[0079] 空余餐桌座位查询和预订界面,该界面由餐厅服务信息界面进入,在该界面里,用户能够看到餐厅的实时餐桌和座位的使用情况。同样的,用户点击相应的座位则会弹出对话框,若该座位正在被使用中或者已被提前预定,则会提示用户该座位目前不可用;若该座位目前空闲则提示用户是否将预订该座位,当用户选择预订后,服务器接收请求,并将该座位加入到用户订单详细列表中,同时修改座位表中的状态为不可用。

[0080] 除了以上界面外,还有用户订单的生成过程,在餐厅服务信息界面里点击选择本餐厅并且生成订单选项后,服务器则会为该用户生成一个订单,订单里包含了用户编号,餐厅编号,订单生成时间等。同时,服务器开启后台服务进程开始计时,当订单生成超过规定时间后用户未能履行订单,则将订单状态设置为无效状态,并且将用户订单详细列表中的预订资源释放。

[0081] 参见图 4,服务器端主要用来处理数据业务,因此服务器端核心为用户数据,餐厅数据,用户订单,用户订单详细列表的数据库。下面将详细介绍本系统中所使用到的数据表:

[0082] 用户数据表,该表记录了用户编号,用户名,用户密码,用户联系方式等信息。该表主要的作用在于服务器验证用户登录,以及作为用户订单表的关联表。

[0083] 餐厅基本数据表,该表记录了餐厅编号,餐厅名称,餐厅地理位置,餐厅的经纬度坐标,餐厅介绍等数据。餐厅编号是该表的主键同时也是餐厅菜谱信息表,餐厅餐桌座位表和订单表的外键。餐厅的经纬度坐标是服务器来向用户推送较近餐厅的一个重要指标,服务器根据用户的经纬度坐标值和餐厅经纬度坐标,从而列出距离用户较近的一系列餐厅。

[0084] 餐厅菜谱分类表,该表记录了各个餐厅的菜谱分类,例如餐厅能够提供的种类有凉菜、炒菜、汤和甜点等。

[0085] 餐厅菜谱详细列表,该表关联菜谱分类表属于其中的某一类,记录了菜名,价格,介绍等信息。

[0086] 餐厅餐桌信息表,该表记录了各个餐厅的餐桌的信息,包括餐桌编号,餐桌所属餐厅(外键),餐桌容纳人数,餐桌使用状态。

[0087] 餐厅座位信息表,该表记录了各个餐厅的座位信息,包括座位编号,座位所属餐桌编号(外键),座位使用状态。

[0088] 用户订单表,该表包括用户编号,餐厅编号,订单生成时间,订单状态。

[0089] 用户订单详细列表,该表包括所属订单编号(外键),菜名,份数,餐桌编号,座位编号。

[0090] 根据上述数据表下面将详细介绍服务器基于这些数据表的主要业务处理过程,如图 4 所示:

[0091] 服务器向终端推送餐厅信息过程:首先用户在程序主菜单界面里选择查看周围餐厅信息,终端调用 GPS 技术得到用户的实时位置信息,向服务器发送位置信息及查看请求,服务器拦截终端请求,将用户的实时经纬度信息与数据库中餐厅基本信息表中存储的餐厅经纬度坐标进行比较,并选出较近的一系列餐厅,并将这些餐厅的数据封装成 JSON 数据返

回给移动终端,终端通过解析 JSON 数据得到了相应的餐厅数据信息,并将这些信息以列表的形式显示出来。与之对应的是调用地图应用并标注出用户及餐厅,该功能的获取较近餐厅数据的过程与前面类似,只是显示方法不同,调用 Google Maps 应用,并且使用 Overlay 技术根据用户和餐厅的经纬度坐标值,将用户和餐厅以符号的形式标注出来。通过这种显示方式能够给用户带来更加直观的感受,缩短用户选择餐厅的时间。

[0092] 用户订单生成过程:用户选择了一个餐厅后,进入该餐厅的服务信息界面,并且点击生成订单,该订单包括用户编号,餐厅编号以及下单时间和订单状态信息。接着用户开始点菜以及预订餐桌和座位,这两步不分先后,可以先点菜也可以先预订餐桌座位。点菜时,用户在浏览餐厅菜谱的同时,可以单击菜谱列表中的某一项,则会弹出对话框提示用户输入需要的份数并确认用户是否提交选择,当用户确认完毕后,终端将用户的选择发送至服务器端,服务器端将数据存入订单详细列表中。与点菜功能相类似的是餐桌和座位的预订功能,用户在浏览某个餐厅的空余餐桌和座位信息时,单击列表中的某一项则会弹出对话框提示用户是否选择该资源,用户确认后发送请求至服务器,服务器接收相关数据并存入订单详细列表中。当用户的订单超出时限而被判定为无效时,服务器则会将用户选择的餐桌及座位资源重新释放,即显示为可用状态。

[0093] 接收用户请求功能。终端上的用户业务都是通过发送请求来完成的,在服务器端应至少完成三个接口:注册接口、登录接口、修改用户信息接口。当收到这些请求时,用户就响应的进行对数据库用户表的操作,实现用户业务。终端发送请求用到的是 Http-Post 协议实现手机终端到服务器的通信,由终端解析用户的操作,翻译成 Http-Post 请求,服务器端通过 Struts2 进行请求拦截,并执行响应的业务。修改用户配置时需要注意的是,多用户查找自己在数据库中的实例并多次修改数据库用户表时会带来数据库回滚的问题。那么对于用户业务数据部分,不能使用事务性管理提交数据,必须使用 HQL 代码进行用户通过用户名的查找实例。

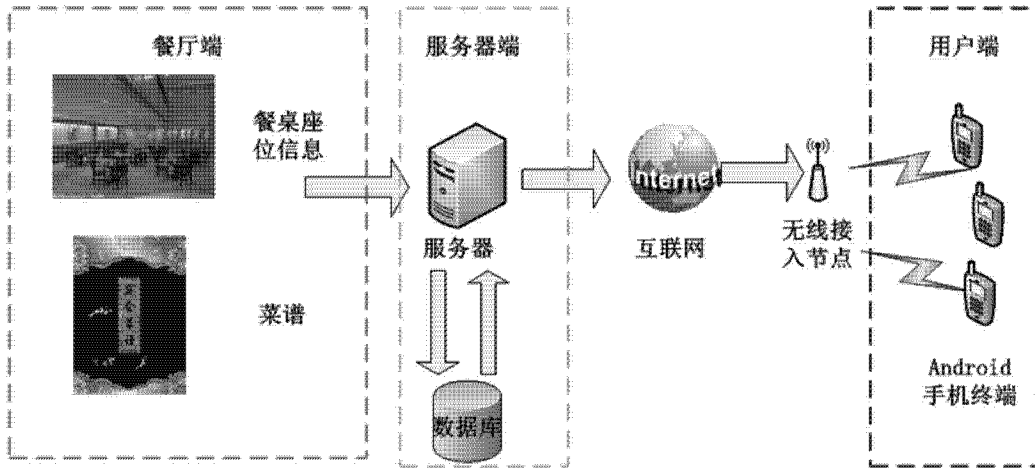


图 1

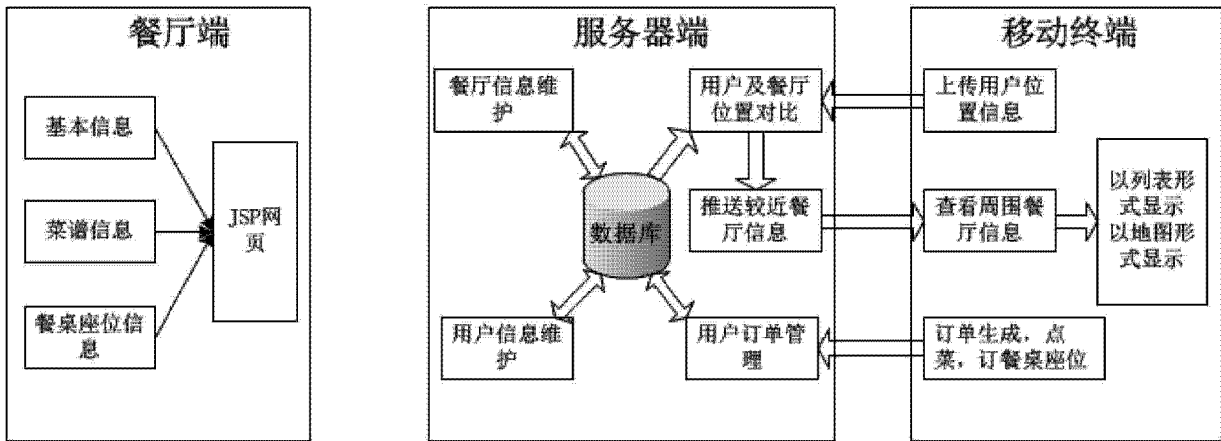


图 2

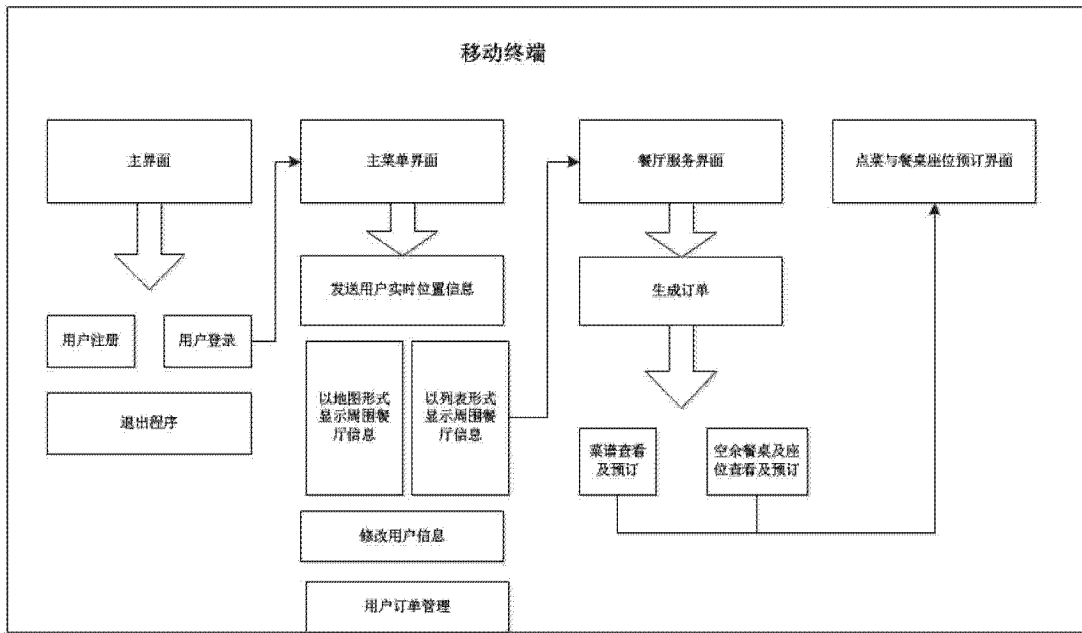


图 3

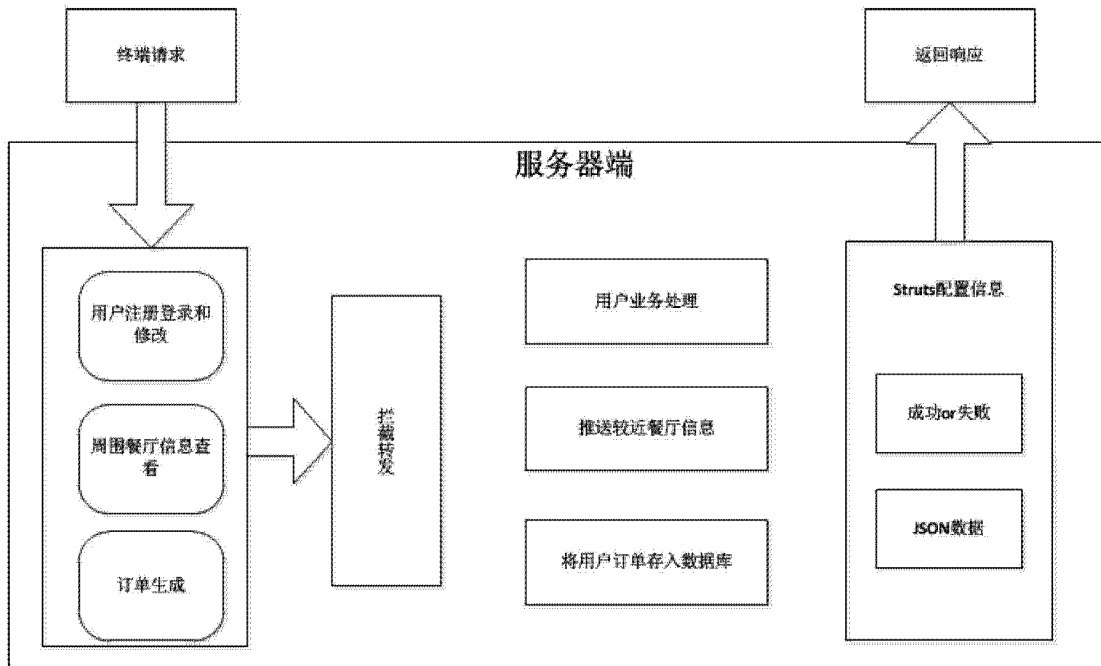


图 4