|  |
| --- |
| “海上捞”餐饮管理系统 |
| 详细设计说明书 |
| V 2.0 |

|  |
| --- |
| 陈梓轩 王雨宸 李欣 聂俊  2019-7-9 |

目录

[1. 范围 3](#_Toc31697)

[1.1标识 3](#_Toc30214)

[1.1.1 文档适用范围 3](#_Toc11957)

[1.1.2 文档中的缩略词 3](#_Toc47)

[1.2系统概述 3](#_Toc1243)

[1.3文档概述 3](#_Toc2168)

[1.4参考文档 3](#_Toc4592)

[2.设计概述 4](#_Toc7436)

[2.1 需求概述 4](#_Toc6106)

[2.2 运行环境概述 4](#_Toc23292)

[2.3 方法和工具 4](#_Toc7969)

[3.数据库设计 4](#_Toc29059)

[3.1 数据库UML类图 5](#_Toc1952)

[3.2 Dining餐桌实体类 5](#_Toc8300)

[3.2.1空构造函数 5](#_Toc31693)

[3.2.2带参数的构造函数 6](#_Toc4814)

[3.2.3获取餐桌各项参数 6](#_Toc17167)

[3.2.4 设置餐桌各项参数 6](#_Toc16500)

[3.3 Dish菜品实体类 7](#_Toc5117)

[3.3.1空构造函数 7](#_Toc5031)

[3.3.2带参数的构造函数 7](#_Toc19848)

[3.3.3获取菜品各项参数 8](#_Toc15931)

[3.3.4 设置菜品各项参数 8](#_Toc30285)

[3.4 Menu点菜单实体类 9](#_Toc16394)

[3.4.1 空构造函数 9](#_Toc7078)

[3.4.2 带参数的构造函数 9](#_Toc29253)

[3.4.3 获取点菜单各项参数 9](#_Toc13888)

[3.4.4 设置点菜单各项参数 10](#_Toc17981)

[3.5 OrderStream订单实体类 10](#_Toc10285)

[3.5.1 空构造函数 10](#_Toc7064)

[3.5.2 带参的构造函数 10](#_Toc18646)

[3.5.3 获取订单各项参数 11](#_Toc28942)

[3.5.4 设置订单各项参数 11](#_Toc21821)

[3.6 Person用户实体类 12](#_Toc3101)

[3.6.1空构造函数 12](#_Toc3937)

[3.6.2带参数的构造函数 12](#_Toc22364)

[3.6.3获取用户各项参数 13](#_Toc28108)

[3.6.4 设置餐桌各项参数 13](#_Toc29770)

[3.7 Reserves预定列表实体类 14](#_Toc1172)

[3.7.1 空构造函数 14](#_Toc9432)

[3.7.2 带参数的构造函数 14](#_Toc23379)

[3.7.3 获取订单各项参数 14](#_Toc4606)

[3.7.4 设置订单各项参数 15](#_Toc16349)

[4.接口设计 15](#_Toc24589)

[4.1 DAO接口 15](#_Toc25542)

[4.1.1 入座时修改餐桌状态 15](#_Toc18092)

[4.1.2 查看餐桌预定 15](#_Toc17141)

[4.1.3 判断是否该餐桌是否能预定 16](#_Toc20894)

[4.1.4 推荐菜品 16](#_Toc970)

[4.1.5 更改菜单 16](#_Toc22716)

[4.1.6 查找单日流水 16](#_Toc17845)

[4.1.7 查找最近一周流水 17](#_Toc28879)

[4.1.8 查找最近一月流水 17](#_Toc14627)

[4.1.9 根据用户电话获取用户信息 17](#_Toc25646)

[4.1.10 员工离职 17](#_Toc29739)

[4.1.11 获取员工登录时间 18](#_Toc31047)

[4.1.12 编辑员工信息 18](#_Toc20843)

[4.1.13 编辑菜品信息 18](#_Toc16001)

[4.2 JPA提供的统一操作 19](#_Toc21229)

[4.2.1获取所有对象 19](#_Toc3605)

[4.2.2 通过ID查找对象 19](#_Toc4362)

[4.2.3 保存对象 19](#_Toc20947)

[4.2.4 删除对象 19](#_Toc15789)

[4.3 用户服务接口 20](#_Toc3568)

[4.3.1 WaiterService接口 20](#_Toc18018)

[4.3.2 ManagerService接口 21](#_Toc13129)

[5.数据字典 24](#_Toc13326)

[5.1 ORDER\_STREAM 24](#_Toc31939)

[5.2 PERSON 24](#_Toc7498)

[5.3 DINING 25](#_Toc30897)

[5.4 DISH 25](#_Toc26806)

[5.5 MENU 26](#_Toc9833)

# 范围

## 1.1标识

### 1.1.1 文档适用范围

本文档适用于“‘海上捞’餐饮系统”（以下简称“系统”）2.0版本及“系统”2.0版本开发过程中的各个阶段。

### 1.1.2 文档中的缩略词

1. 甲方：“‘海上捞’餐饮集团”，为本项目的客户与出资方。
2. 乙方：本项目开发组。
3. 用户：指所有使用系统的使用者，包括餐饮公司的餐厅经理及服务员。
4. 系统：指本项目中待开发的“‘海上捞’餐饮管理系统”。系统主要由服务器、餐厅经理端和服务员端组成。
5. 服务员端需求(W*x*)：对于服务员端需求的编号，其中*x*为功能编号。
6. 餐厅经理端需求(M*x*)：对于餐厅经理端需求的编号，其中*x*为功能编号。
7. 约束(CO*x*)：指软件开发过程中可能因外部不可抗力带来的需求及功能上的限制。

## 1.2系统概述

为了能够加强餐饮行业的管理，提高办公自动化率，以增加效率，本公司拟开发本系统，用于实现餐厅的点餐、预约及营收情况统计。本项目由“‘海上捞’餐饮集团”出资，主要用户为甲方旗下各家连锁店的餐厅经理及服务员。甲方希望本系统能够及时上线，希望在两周内基本开发完成。

## 1.3文档概述

本文档为“系统详细设计说明书2.0版”，是“系统详细设计说明书1.0版”的改进。在原有说明基础上，增加了对于数据库等部分的更加详细的设计描述。该文档。用于详细陈述系统1.0版本开发过程中，各个模块需要实现的功能及各个接口之间的联系。

本文档作为详细设计的说明，主要使开发工程师在开发过程中明确开发目标，应对外保密，不得外传。

## 1.4参考文档

[1] 《软件需求规格说明书》v2.0

# 2.设计概述

## 2.1 需求概述

本系统旨在提升传统餐饮行业点单方式落后，缺乏统一的信息管理，效率低下的缺陷，开发一套在线的餐饮管理系统。将从原材料控制到顾客点单到生成订单流水的一系列分散的流程集中在同一个系统中实现。本系统的目标是将传统的纸质的点单方式，通过电子化的方式实现。服务员只需要在移动端设备上打开网页，便可以了解餐品的情况，并为顾客点单。餐厅经理也可以实时了解餐厅经营状况，并根据报表调整经营策略。

## 2.2 运行环境概述

为了实现跨平台的适配性，降低开发难度，提高开发质量，本平台将以网页形式，而不是移动端APP形式实现。主要运行环境为当前主流平板电脑（如iPad）或PC机。应运行iOS，Android，Windows或其他支持网络连接的操作系统。

本系统要求在支持flash的浏览器上运行（如Safari, Chrome, Firefox等）。运行设备应至少有2G的内存。

## 2.3 方法和工具

本系统数据库采用MySQL开发，数据保存在阿里云服务器上。

采用GitHub进行版本控制。

采用Eclipse Java EE作为集成开发环境。

# 3.数据库设计

“海上捞”餐饮管理系统采用基于Spring4.0的J2EE框架，数据持久层采用JPA的Hibernate的实现。为每张表建立的单独的DAO工厂接口，该接口提供所有的数据操作包括：新建、删除、更新、SQL查询、HQL查询等，同时注册成Spring的资源库供服务层调用。

## 3.1 数据库UML类图

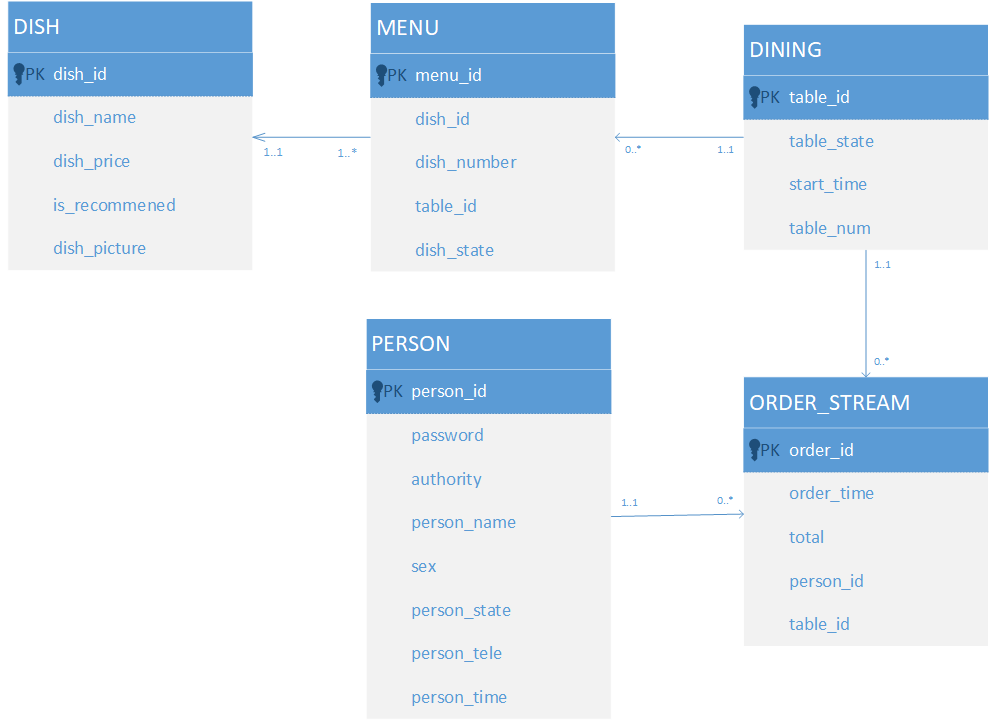


Figure 1 数据库UML类图

## 3.2 Dining餐桌实体类

### 3.2.1空构造函数

a） 功能描述：构造Dining对象。

b） 设计约束与需求：无特殊要求。

**方法：**

public Dining()

**参数：**无。

**返回值：**Dining对象。

**异常：**无。

### 3.2.2带参数的构造函数

a） 功能描述：构造Dining对象。

b） 设计约束与需求：无特殊要求。

**方法：**

public Dining(Long tableId,

int tableNum,

int tableState,

Timestamp startTime)

**参数：**

* tableId – 餐桌编号
* tableNum – 餐桌容纳人数。
* tableState – 餐桌状态（空闲/被占用/被预约）。
* startTime – 餐桌开始时间。

**返回值：**Dining对象。

**异常：**无。

### 3.2.3获取餐桌各项参数

a） 功能描述：

获取对象的各项参数（tableId/ tableNum/ tableState/ startTime）。

b） 设计约束与需求：无特殊要求。

**方法：**public long getTableId();

public int getTableNum();

public int getTableState ();

public timestamp getStartTime ();

**参数：**无。

**返回值：**餐桌的各项参数。

**异常：**无。

### 3.2.4 设置餐桌各项参数

a） 功能描述：

设置对象的各项参数（TableId/ TableNum/ TableState/ StartTime）。

b） 设计约束与需求：无特殊要求。

**方法：**

public void setTableId(long tableId);

public void setTableNum (int tableNum);

public void setTableState (int tableState);

public void setTableId(timestamp StartTime);

**参数：**餐桌实体各项参数

**返回值：**无。

**异常：**无。

## 3.3 Dish菜品实体类

### 3.3.1空构造函数

a） 功能描述：构造Dish对象。

b） 设计约束与需求：无特殊要求。

**方法：**

public Dish()

**参数：**无。

**返回值：**Dish对象。

**异常：**无。

### 3.3.2带参数的构造函数

a） 功能描述：构造Dish对象。

b） 设计约束与需求：无特殊要求。

**方法：**

public Dish(String dishName,

int dishPrice,

String dishDesc,

int isRecommend,

String dishPicture)

**参数：**

* dishName – 菜品名称。
* dishPrice – 菜品价格。
* dishDesc – 菜品描述。
* isRecommend – 菜品是否被推荐。
* dishPicture – 菜品图片。

**返回值：**Dish对象。

**异常：**无。

### 3.3.3获取菜品各项参数

a） 功能描述：

获取对象的各项参数（dishId / dishName / dishPrice / dishDesc / isRecommend/ dishPicture）。

b） 设计约束与需求：无特殊要求。

**方法：**public String getDishId ();

public String getDishName ();

public int getDishPrice ();

public String getDishDesc ();

public int getIsRecommend ();

public String getDishPicture ();

**参数：**无。

**返回值：**菜品的各项参数。

**异常：**无。

### 3.3.4 设置菜品各项参数

a） 功能描述：

设置餐桌各项参数（dishId / dishName / dishPrice / dishDesc / isRecommend/ dishPicture）。

b） 设计约束与需求：无特殊要求。

**方法：**

public void setDishId (Long dishId);

public void setDishName (String dishName);

public void setDishPrice (int dishPrice);

public void setDishDesc (String dishDesc);

public void setIsRecommend (int isRecommend);

public void setDishPicture (String dishPicture);

**参数：**餐桌的各项参数

**返回值：**无。

**异常：**无。

## 3.4 Menu点菜单实体类

### 3.4.1 空构造函数

a） 功能描述：构造Menu对象

b） 设计约束与需求：无特殊要求。

**方法**：

public Menu() ；

**参数**：无；

**返回**：一个menu实体；

**异常**：无；

### 3.4.2 带参数的构造函数

a） 功能描述：构造Menu对象。

b） 设计约束与需求：无特殊要求。

**方法：**

public Menu(long menuId,

long dishId, 、

long tableId,

int dishState,

int dishNumber)；

**参数：**

* menuId – 点菜单编号。
* dishId – 菜品编号。
* tableId – 桌子编号。
* dishState – 菜品状态。
* dishNumber – 菜品数量。

**返回值：**Menu对象。

**异常：**无。

### 3.4.3 获取点菜单各项参数

a) 功能描述：

获取到menu实体类的部分属性值（menuId \ dishId \tableId \ dishState \ dishNumber）

b) 设计约束和需求：无特殊要求。

**方法：**

public int getDishState()；

public long getTableId()；

public long getDishId()；

public long getMenuId() ；

**参数：**无；

**返回：**相应的menu类的属性

**异常：**无。

### 3.4.4 设置点菜单各项参数

a） 功能描述 ：对menu实体类的部分属性进行修改或者实例化

b） 设计约束与需求：无特殊要求。

方法：

public void setMenuId(long menuId)；

public long getDishId() ；

public void setDishId(long dishId)；

public void setTableId(long tableId)；

public void setDishState(int dishState)；

参数：menu的各个属性

返回：一个menu实体

异常;无

## 3.5 OrderStream订单实体类

### 3.5.1 空构造函数

a） 功能描述：构造orderStream对象。

b） 设计约束与需求：无特殊要求。

**方法：**

public OrderStream（）

**参数：**无。

**返回值：**无。

**异常：**无。

### 3.5.2 带参的构造函数

a） 功能描述：通过传入参数构造Dish对象（orderId/personId/total/orderTime/tableId）

b） 设计约束与需求：无特殊要求。

方法：

public OrderStream(long orderId,

long personId,

int total,

Timestamp orderTime,

long tableId)；

参数：

* orderId:订单的ID
* personId：预定人的Id
* total：总人数
* orderTime：预定时间
* table ID：座子的Id

返回值：一个OrderStream实体

异常：无

### 3.5.3 获取订单各项参数

a) 功能描述：获取一个orderStream实体的属性值

b) 设计约束和需求：无特殊要求

方法：

public long getOrderId()；

public long getPersonId()；

public int getTotal()；

public Timestamp getOrderTime()；

public long getTableId()；

参数：无

返回：无

异常：无

### 3.5.4 设置订单各项参数

a) 功能描述：对一个实体Orderstream进行属性修改

b) 设计约束和需求：无特殊要求。

**方法：**

public void setOrderId(long orderId)；

public void setPersonId(long personId)；

public void setTotal(int total)；

public void setOrderTime(Timestamp orderTime)；

public void setTableId(long tableId)；

**参数：**order Stream的实体对应属性

**返回值：**无；

**异常：**无；

## 3.6 Person用户实体类

### 3.6.1空构造函数

a） 功能描述：构造Person对象。

b） 设计约束与需求：无特殊要求。

**方法：**

public Person ()

**参数：**无。

**返回值：**无。

**异常：**无。

### 3.6.2带参数的构造函数

a） 功能描述：构造Person对象。

b） 设计约束与需求：无特殊要求。

**方法：**

public Person (String personName,

String password,

String personTele,

int authority,

int sex,

int personState)

* **参数：**personName – 用户姓名。
* password – 密码。
* personTele – 用户电话。
* authority – 用户权限。
* sex – 用户性别。
* personState – 用户状态（是否在职）

**返回值：**Person对象。

**异常：**无。

### 3.6.3获取用户各项参数

a） 功能描述：

获取对象的各项参数（personId / personName / password / personTele / authority / sex / personState）。

b） 设计约束与需求：无特殊要求。

**方法：**public String getPersonId ();

public String getPersonName ();

public String getPassword ();

public int getPersonTele ();

public String getAuthority ();

public int getSex ();

public String getPersonState ();

**参数：**无。

**返回值：**用户的各项参数。

**异常：**无。

### 3.6.4 设置餐桌各项参数

a） 功能描述：

设置餐桌各项参数（personId / personName / password / personTele / authority / sex / personState）。

b） 设计约束与需求：无特殊要求。

**方法：**public void setPersonId (int personId);

public void setPersonName (String personName);

public void setPassword (String password);

public void setPersonTele (String personTele);

public void setAuthority (int authority);

public void setSex (int sex);

public void setPersonState (int personState);

**参数：**用户实体各项参数

**返回值：**无。

**异常：**无。

## 3.7 Reserves预定列表实体类

### 3.7.1 空构造函数

a) 功能描述：构造一个新的Reserve实体对象

b) 设计约束与需求：无特殊要求。

**方法：**

public Reserves()；

**参数：**无

返回值：一个实例的Reserve对象

**异常：**无

### 3.7.2 带参数的构造函数

a) 功能描述：

reserve含参构造函数(reserveId, tableId, reserveTime, reserveTele)

b) 设计约束与需求：无特殊要求

**方法：**

public Reserves(long reserveId, long tableId, int reserveTime, int reserveTele)

**参数：**

* reserveId：预定的ID
* tableID：座子的ID
* reserveTime :预定时间
* reserveTele：预定电话

**返回值：**一个实例化的Reserve对象

**异常：**无

### 3.7.3 获取订单各项参数

功能描述：获取该reserve实例化对象的相应属性

设计约束和需求：无特殊要求

**方法：**

public long getTableId()；

public timestamp getReserveTime()；

public String getReserveTele()；

public long getReserveId()；

**参数：**无

**返回值：**reserve的相应属性值

**异常：**无

### 3.7.4 设置订单各项参数

a) 功能描述：修改该reserve实体的部分属性

b) 设计约束和需求：无特殊要求

**方法：**

public void setReserveTele(String reserveTele) ；

public void setReserveTime(timestamp reserveTime)；

public void setTableId(long tableId)；

public void setReserveId(long reserveId) ；

**参数：**实体的各项属性

**返回值：**无

**异常：**无

# 4.接口设计

## 4.1 DAO接口

### 4.1.1 入座时修改餐桌状态

* 功能描述：当有人入座时，或预约顾客就座。
* 设计约束与需求：无特殊要求
* 方法：public void takeTable(long id)
* 参数：id：表示需要更新状态的桌号。
* 返回值：无
* 异常：无

### 4.1.2 查看餐桌预定

* 功能描述：经理选择一个餐桌以及预定时间，查看餐桌已有的预定信息。
* 设计约束与需求：无特殊要求
* 方法：public List<Reserves> twoHourInterval(long tableId; Date reserveTime)
* 参数：tableId：表示需要查看预定信息的餐桌；reserveTime：表示需要查看的预定时间。
* 返回值：List<Reserves>：表示该餐桌的预定信息列表。
* 异常：无

### 4.1.3 判断是否该餐桌是否能预定

* 功能描述：当顾客确定预定的餐桌以及时间以后，经理查看该餐桌是否可用，只用在可用时，顾客才能预定该餐桌，否则，顾客不能预定。
* 设计约束与需求：无特殊要求
* 方法：public List<Dining> isAbleReserve(long tableid, Date reserveTime)
* 参数：tableid：表示需要查看是否可预订的桌号；reserveTime：表示需要预定的时间。
* 返回值：List<Dining>:如果其长度大于0，则该餐桌可以预定，否则，餐桌不可预定
* 异常：无

### 4.1.4 推荐菜品

* 功能描述：经理选择某种菜品，将其添加到推荐菜品中。
* 设计约束与需求：无特殊要求
* 方法：public void updateRecom(int i, long id)
* 参数：i：表示菜品在推荐方面更新的状态；id：表示需要修改预定状态的菜品。
* 返回值：无
* 异常：无

### 4.1.5 更改菜单

* 功能描述：服务员在经过顾客确定后，修改订单的状态，只有在顾客确认以后，厨房才会开始按照订单制作菜品。
* 设计约束与需求：无特殊要求
* 方法：public void changeMenuState(long menuId; int i)
* 参数：menuId：表示需要修改的订单；i：表示需要修改的订单的状态。
* 返回值：无
* 异常：无

### 4.1.6 查找单日流水

* 功能描述：经理查看每天的流水，系统将流水清单打印出来。
* 设计约束与需求：无特殊要求
* 方法：public List<OrderStream> findToday()
* 参数：无
* 返回值：List<OrderStream>：当天的流水列表
* 异常：无

### 4.1.7 查找最近一周流水

* 功能描述：经理查看最近一周的流水，系统将流水清单打印出来。
* 设计约束与需求：无特殊要求
* 方法：public List<OrderStream> findWeek()
* 参数：无
* 返回值：List<OrderStream>：最近一周的流水列表
* 异常：无

### 4.1.8 查找最近一月流水

* 功能描述：经理查看最近一月的流水，系统将流水清单打印出来。
* 设计约束与需求：无特殊要求
* 方法：public List<OrderStream> findMonth()
* 参数：无
* 返回值：List<OrderStream>：最近一月的流水列表
* 异常：无

### 4.1.9 根据用户电话获取用户信息

* 功能描述：用户需要通过电话号码进行登录，将用户输入的电话号码与密码传入系统，判断信息是否正确。
* 设计约束与需求：无特殊要求。
* 方法：public Person findByTelePassword(String tele, String password);
* 参数：tele：电话号码，password：密码。
* 返回值：Person: 若信息正确，返回个人信息，否则返回NULL。
* 异常：无。

### 4.1.10 员工离职

* 功能描述：员工离职时应将员工状态修改为“离职”。
* 设计约束与需求：无特殊要求。
* 方法：public void waiterIsNotWork(long id);
* 参数：id：离职员工的编号。
* 返回值：无。
* 异常：根据该id查询不到员工。

### 4.1.11 获取员工登录时间

* 功能描述：经理查看所有员工的登录时间，系统将员工的登录时间列表打印出来。
* 设计约束与需求：无特殊要求。
* 方法：public List<Person> getWorkingPerson();
* 参数：无。
* 返回值：所有员工的登录时间列表。
* 异常：无

### 4.1.12 编辑员工信息

* 功能描述：经理选择某位员工，更改其相关信息。
* 设计约束与需求：无特殊要求
* 方法：public void editinfo(long id; String name; int sex; String telephone; String password)
* 参数：id：表示需要修改的员工编号；name：表示需要修改的员工的名字；sex：表示需要修改的员工的性别；telephone：表示需要修改的员工的电话；password：表示需要修改的员工的密码。
* 返回值：无
* 异常：无

### 4.1.13 编辑菜品信息

* 功能描述：经理选择某种菜品，更改菜品的相关信息。
* 设计约束与需求：无特殊要求
* 方法：public void updateDish(long id; String name; int price; String desc; String pic; int rec)
* 参数：id：表示需要修改的菜品；name：表示需要修改的菜品的名字；price：表示需要修改的菜品的价格；desc：表示需要修改的菜品的描述；pic：表示需要修改的菜品的照片；rec：表示菜品是否被推荐。
* 返回值：无
* 异常：无

## 4.2 JPA提供的统一操作

### 4.2.1获取所有对象

* 功能描述：获取数据库中某一对象的所有实体。
* 设计约束与需求：无特殊要求
* 方法：List<T> findAll();
* 参数：无
* 返回值：List<T>

### 4.2.2 通过ID查找对象

* 功能描述：获取数据库中某一对象的ID标签置为id的实体。
* 设计约束与需求：无特殊要求
* 方法：Optional<T> findById(S ids);
* 参数：S ids
* 返回值：Optional<T>
* 异常：无

### 4.2.3 保存对象

* 功能描述：向数据库中添加一个实体
* 设计约束与需求：无特殊要求
* 方法：<S extends T> S save(S entity);
* 参数：S entity
* 返回值：<S extends T> S
* 异常：无

### 4.2.4 删除对象

* 功能描述：删除数据库中Id值为id的数据项
* 设计约束与需求：无特殊要求
* 方法：void deleteById(T id);
* 参数：T id
* 返回值：void
* 异常：无

## 4.3 用户服务接口

### 4.3.1 WaiterService接口

#### 4.3.1.1 加载餐桌状态

* 功能描述：服务员可以通过该功能，查看当前餐桌状态。
* 设计需求与约束：餐桌应有空闲，有人，已预约三种状态。
* 方法：public Object loadTableStatus(String waiterPage, int start)；
* 参数：waiterPage：返回的网页地址；start：分页标签。
* 返回值：返回一个网页对象。
* 异常：无。

#### 4.3.1.2 展示每桌已点菜品

* 功能描述：服务员可以通过该功能，查看各桌所点菜品及菜品的状态。
* 设计需求与约束：菜品状态应包含已下单，已上菜两种。
* 方法：public Object loadOrderDish(Long tableId)
* 参数：tableId：表示餐桌编号。
* 返回值：返回一个网页对象，表示餐桌对应的菜品列表。
* 异常：无。

#### 4.3.1.3 占桌

* 功能描述：当顾客到达时，服务员可通过该功能为顾客选择座位。
* 设计需求与约束：服务员可以选择空闲或已被预约的桌子。选择已预约的桌子则视为预约人到达餐厅，预约自动取消。
* 方法：public boolean takeTable(long tableId, String tableState)
* 参数：tableId：表示餐桌编号；tableState：表示桌子现在的状态。
* 返回值：boolean类：表示操作是否成功。
* 异常：无。

#### 4.3.1.4 添加菜单

* 功能描述：点餐时添加菜品进入临时订单。
* 设计需求与约束：无特殊要求。
* 方法：public String addMenu(String menus)
* 参数：menus：以json格式标识的字符串，表示点餐的信息。
* 返回值：String：表示点餐是否成功。

#### 4.3.1.5 修改菜品状态

* 功能描述：修改菜品的状态，将已下单改为已上菜。
* 设计需求与约束：无特殊要求。
* 方法：public String changeMenuState(long menuId)
* 参数：menuId：表示菜品的编号。
* 返回值：String：返回修改结果。
* 异常：无。

#### 4.3.1.6 结算订单

* 功能描述：结算订单，将餐桌状态恢复为预约中或者空闲，并将本次餐桌订单信息插入到流水记录中。
* 设计需求与约束：无特殊要求。
* 方法：public String settleAccount(HttpSession session, long tableId, int total)
* 参数：session：用于获取服务员ID；tableId：结算餐桌的ID；total：结算金额 。
* 返回值：String：返回修改结果。
* 异常：无。

#### 4.3.1.7 取消占座

* 功能描述：在客户尚未点菜时可以选择取消占座。
* 设计需求与约束：无特殊要求。
* 方法：public Object houHui(HttpSession session, long tableId)
* 参数：session：用于获取服务员ID；tableId：被取消占座餐桌的ID；。
* 返回值：Object：返回修改结果。
* 异常：无。

### 4.3.2 ManagerService接口

#### 4.3.2.1 添加菜品

* 功能描述：根据所提供的信息，添加新菜品
* 设计需求与约束：如果已存在同名菜品，则不能添加。
* 方法：public int addNewDish (String name,int price,String desc,String pic,int rec)
* 参数：name：菜品名称；price：价格；desc：菜品描述；pic：图片链接；rec：是否为推荐菜。
* 返回值：int：表示添加是否成功。
* 异常：无。

#### 4.3.2.2 修改菜品信息

* 功能描述：根据编号，查找菜品，并更新菜品信息。
* 设计需求与约束：无特殊需求与约束。
* 方法：public void updateDish(long id,Dish d)
* 参数：id：菜品编号；d：修改后的菜品信息。
* 返回值：无。
* 异常：无。

#### 4.3.2.3 修改菜品推荐状态

* 功能描述：根据编号，查找菜品并设置为推荐菜品。
* 设计需求与约束：无特殊需求与约束。
* 方法：public void setRecommend(long dishid,int rec)
* 参数：id：菜品编号；rec：修改为是否推荐。
* 返回值：无。
* 异常：无。

#### 4.3.2.4 展示订单

* 功能描述：展示昨天/一周内/一个月内/全部的订单信息。
* 设计需求与约束：根据用户的选择，展示不同的内容。
* 方法：public Object loadOrderStream(String detail, int start) {
* 参数：detail：展示的周期，仅可以为“今日”，最近一周“，”最近一月“，“全部”；start：分页标签。
* 返回值：返回一个网页对象，表示查找结果。
* 异常：detail不在“今日”，最近一周“，”最近一月“中，应报错。

#### 4.3.2.5 展示服务员

* 功能描述：展示所有服务员的信息。
* 设计需求与约束：无特殊需求与约束。
* 方法：public Object takePerson(String location,int start)
* 参数：location：返回地址；start：分页标签。
* 返回值：返回一个网页对象。
* 异常：无

#### 4.3.2.6 添加服务员

* 功能描述：根据提供的信息，添加服务员。
* 设计需求与约束：服务员的姓名与电话号码不能与已有的服务员重复。
* 方法：public int addPerson(String name,String sex,String telephone,String password)
* 参数：name：服务员姓名；sex：性别；telephone：电话号码；password：密码
* 返回值：int：表示添加是否成功。
* 异常：若存在重复的姓名或电话号码，应提示添加错误。

#### 4.3.2.7 修改服务员

* 功能描述：根据编号，查找服务员，并编辑信息。
* 设计需求与约束：无特殊需求与约束。
* 方法：public int updatePerson(Long id,String name,String sex,String telephone,String password)
* 参数：id：服务员编号；name：服务员姓名；sex：性别；telephone：电话号码；password：密码。
* 返回值：int：表示修改是否成功。
* 异常：无。

#### 4.3.2.8 展示餐桌

* 功能描述：获取所有餐桌信息并展示。
* 设计需求与约束：无特殊需求与约束。
* 方法：public Object loadTables(String root)
* 参数：root：返回地址。
* 返回值：Object：返回一个网页对象。
* 异常：无。

#### 4.3.2.9 预定

* 功能描述：根据预定时间，添加预定。
* 设计需求与约束：需要判断预定时段是否还有空座。
* 方法：public Object loadSearchedTable(String managereserve, String time)
* 参数：time：预定时间；managereserve：返回地址。
* 返回值：Object：返回一个网页对象。
* 异常：无。

#### 4.3.2.10 取消预定

* 功能描述：根据预定信息，取消预定。
* 设计需求与约束：无特殊需求与约束。
* 方法：public Object deleteReserve(long id)
* 参数：id：需要删除的预定的编号。
* 返回值：Object：返回一个网页对象。
* 异常：无

# 5.数据字典

## 5.1 ORDER\_STREAM

ORDER\_STREAM表记录了所有订单的信息，包括餐桌号与订单时间等

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **数据类型** | **默认值** | **允许为空** | **自动递增** | **备注** |
| **order\_id** | bigint |  | NO | YES | 主键，表示点菜内容的编号。 |
| **person\_id** | bigint |  | NO | NO | 外键，代表结算服务员编号 |
| **Total** | Int | 1 | NO | NO | 订单的总价 |
| **table\_id** | bigint |  | NO | NO | 外键，表示该订单执行所在的桌号 |
| **order\_time** | timestamp |  | NO | NO | 订单执行的时间 |

## 5.2 PERSON

PERSON表记录了所有用户的信息。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **数据类型** | **默认值** | **允许为空** | **自动递增** | **备注** |
| **person\_id** | bigint |  | NO | YES | 主键，表示用户的编号，唯一。 |
| **password** | varchar |  | NO | NO | 表示用户的密码，用于登录。 |
| **person\_name** | varchar | NULL | YES | NO | 表示用户的真实姓名。 |
| **sex** | int | 0 | NO | NO | 用户的性别。0表示女，1表示男。 |
| **person\_tele** | varchar | NULL | YES | NO | 表示用户的电话号码。 |
| **authority** | int | 0 | NO | NO | 表示用户的类型，2表示服务员，1表示餐厅经理 |
| **person\_state** | int | 0 | NO | NO | 表示用户当前的状态，0表示用户已离职，不可以登陆系统，1表示用户在职。 |
| **person\_time** | timestamp | NULL | YES | NO | 记录用户本日第一次登录的时间，用于考勤。 |

## 5.3 DINING

DINING表记录了餐桌的信息。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **数据类型** | **默认值** | **允许为空** | **自动递增** | **备注** |
| **table\_id** | bigint |  | NO | YES | 主键，桌子的编号。 |
| **table\_state** | int | 0 | NO | NO | 用int表示桌子的状态。0表示桌子为空，1表示桌子尚有预约，2表示桌子正在被占用。 |
| **start\_time** | timestamp | NULL | YES | NO | 存储餐桌客人就餐开始的时间。 |
| **table\_num** | int | 2 | NO | NO | 表示桌子最多能坐下几名顾客。 |

## 5.4 DISH

DISH表记录了所有菜品的信息。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **数据类型** | **默认值** | **允许为空** | **自动递增** | **备注** |
| **dish\_id** | bigint |  | NO | YES | 主键，菜品的编号。 |
| **dish\_name** | Varchar | NULL | YES | NO | 表示菜品的名称。 |
| **dish\_price** | Int | 0 | NO | NO | 菜品的价格，以整数形式保存。 |
| **dish\_picture** | varchar | NULL | YES | NO | 菜品基本图片存储的名称 |
| **isrecommend** | Int | 0 | NO | NO | 是否为推荐。0：不是推荐菜，1：是推荐菜。 |

## 5.5 MENU

MENU表用于记录所有点菜单的情况。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **数据类型** | **默认值** | **允许为空** | **自动递增** | **备注** |
| **menu\_id** | bigint |  | NO | YES | 主键，点菜单的编号 |
| **dish\_id** | bigint |  | NO | NO | 主键，菜品的编号 |
| **dish\_state** | int | 0 | NO | NO | 已点菜品的状态 |
| **table\_id** | bigint |  | NO | NO | 外键。表示对应的餐桌号。 |