《数据库系统原理》课程设计

系统实现报告

题目名称： 托福做题系统

学号及姓名：\_\_\_\_\_15231037 张卓\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

15071030 张夫洋\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2017年 12 月 28 日

**一．系统功能需求分析**

1、用户可以通过账号密码或者游客模式登入托福练习系统，进行阅读和写作模块的练习。注册账号后，练习的相关信息能够进行保存并在日后查询。用户可以选择分套题训练或者分题型训练。

2、当用户选择分套题训练时，提供两种模式，一种是成套训练，每次连续给出一套模拟题的阅读题和写作题，共三篇阅读，一篇综合写作，一篇独立写作；第二种是分项训练，用户自行选择要训练的题目，一次给出一篇阅读或一篇写作。

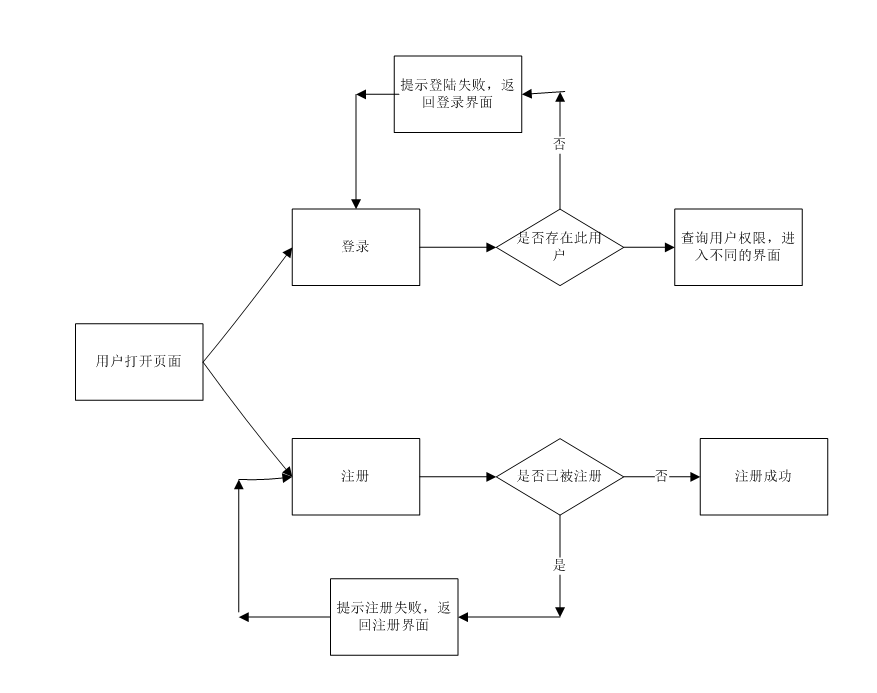
3、当用户选择分题型训练时，可以针对整篇阅读或写作的学科主题选择，并集中训练，也可以针对某道选择题的题型来集中训练。

4、阅读答题界面给出倒计时提示，同时

4、后台人员拥有特殊权限账号，可以在系统种添加套题、

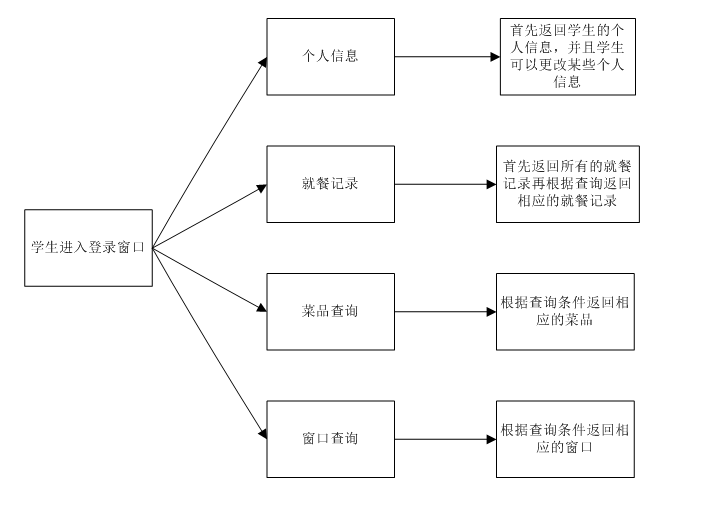
## 二．系统功能结构设计

1、顶层功能

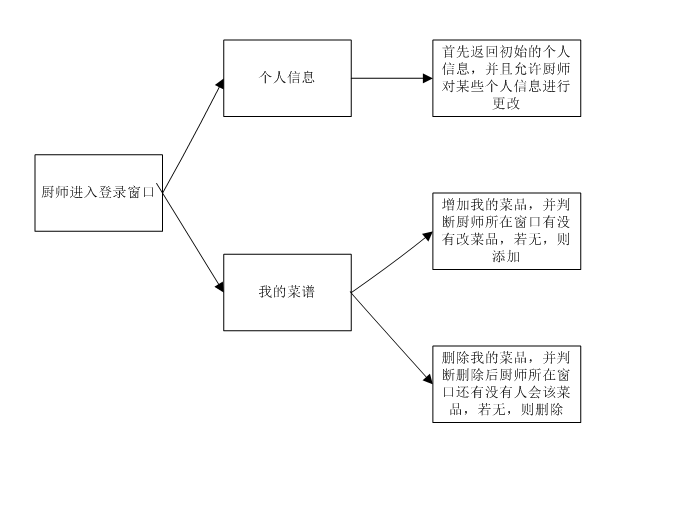


2、一层功能

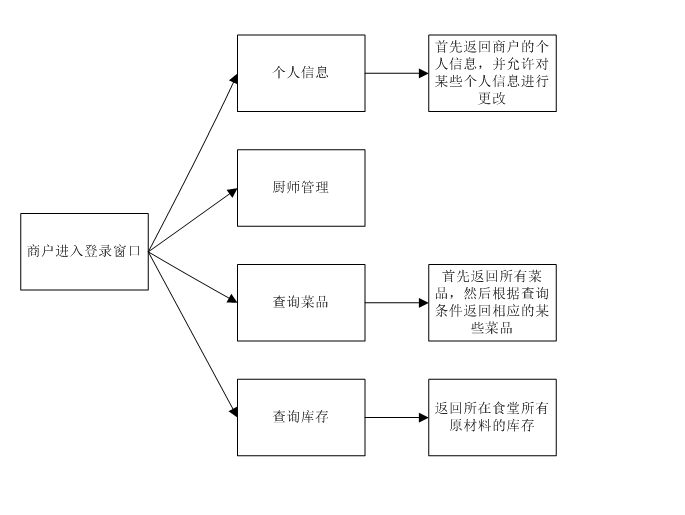
1）学生登录：



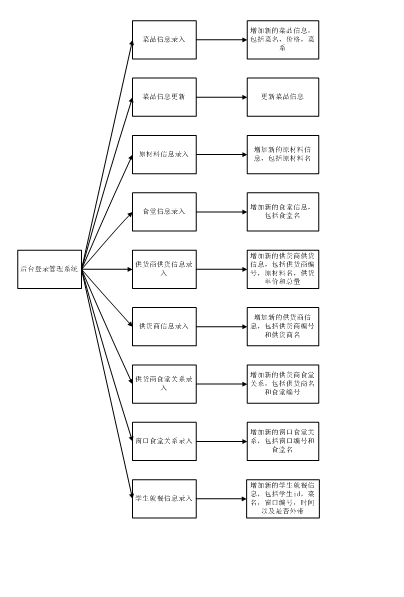
2）厨师登录



3）商户登录

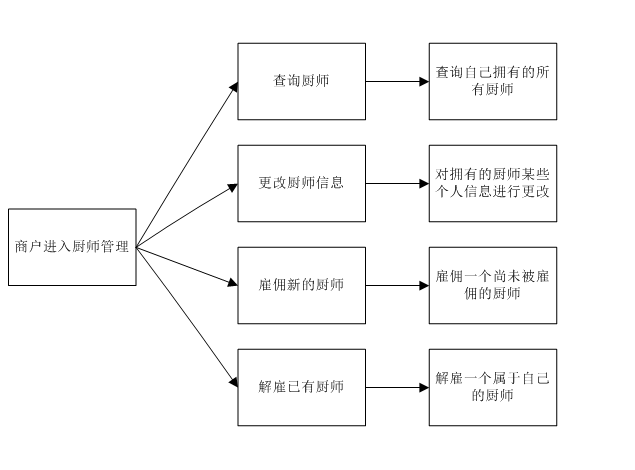


4）后台登录



3、二层功能

商户进入厨师管理界面



**三、数据库基本表的定义**

|  |  |  |  |
| --- | --- | --- | --- |
| 表名：Student | | | |
| 数据名 | 类型 | 约束 | 备注 |
| name | nvarchar(10) | | 学生姓名 |
| id | char(8) | 主键 | 学生学号 |
| sex | nchar(1) | in（男，女） | 性别 |
| age | tinyint | 大于10小于40 | 年龄 |
| birthplace | nvarchar(3) |  | 籍贯 |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名：Dish | | | |
| 数据名 | 类型 | 约束 | 备注 |
| name | nvarchar(10) | 主键 | 菜名 |
| price | float | 非负 | 价格 |
| category | nvarchar(1) | in（鲁、川、粤、苏、闽、浙、湘、徽、其他） | 菜系 |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名：Window | | | |
| 数据名 | 类型 | 约束 | 备注 |
| id | char(4) | 主键 | 窗口编号 |
| winName | nvarchar(10) |  | 窗口名称 |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名：Order | | | |
| 数据名 | 类型 | 约束 | 备注 |
| stuId | char(8) | Student表的外键，主键 | 学生学号 |
| dishName | nvarchar(10) | Dish表的外键 | 菜名 |
| winId | char(4) | Window表的外键 | 窗口编号 |
| date | date |  | 点餐时间 |
| takeaway | Boolean |  | 是否外带 |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名：Win\_Dish | | | |
| 数据名 | 类型 | 约束 | 备注 |
| winId | char(4) | 主键，外键 | 窗口编号 |
| dishName | nvarchar(10) | 主键，外键 | 菜名 |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名：Chief | | | |
| 数据名 | 类型 | 约束 | 备注 |
| name | nvarchar(10) | | 厨师名字 |
| id | char(8) | 主键 | 编号 |
| sex | nchar(1) | in（男，女） | 性别 |
| age | tinyint | 大于10小于40 | 年龄 |
| salary | smallint |  | 工资 |
| master\_id | char(8) | 外键 | 厨师长编号 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名：Win\_Rest | | | | |
| 数据名 | 类型 | 约束 | 备注 |
| winId | char(4) | 主键，外键 | 窗口编号 | |
| restName | nvarchar(10) | 外键 | 食堂名字 | |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名：Dish\_Chief | | | |
| 数据名 | 类型 | 约束 | 备注 |
| dishName | nvarchar(10) | 主键，外键 | 菜名 |
| chiefId | char(8) | 主键，外键 | 窗口编号 |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名：Chief\_Win | | | |
| 数据名 | 类型 | 约束 | 备注 |
| chiefId | char(8) | 外键 | 厨师编号 |
| winId | char(4) | 主键，外键 | 窗口编号 |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名：Supplier | | | |
| 数据名 | 类型 | 约束 | 备注 |
| name | nvarchar(10) | | 供货商名字 |
| id | char(4) | 主键 | 供货商编号 |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名：Sup\_Mat | | | |
| 数据名 | 类型 | 约束 | 备注 |
| supId | char(4) | 主键，外键 | 供货商编号 |
| matName | nvarchar(10) | 主键，外键 | 原料名字 |
| price | float |  | 供应单价 |
| amount | int |  | 供应数量 |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名：Sup\_Rest | | | |
| 数据名 | 类型 | 约束 | 备注 |
| supId | char(4) | 主键，外键 | 供货商编号 |
| restName | nvarchar(10)主键，外键 | | 食堂名字 |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名：Material | | | |
| 数据名 | 类型 | 约束 | 备注 |
| Name | nvarchar(10) | 主键 | 原料名字 |

|  |  |  |  |
| --- | --- | --- | --- |
| 表名：Rest | | | |
| 数据名 | 类型 | 约束 | 备注 |
| Name | nvarchar(10) | 主键 | 食堂名字 |

**四、触发器的定义与实现**

用户表触发器：（给用户表插入记录之后会给相应的不同用户子表插入记录）

create trigger triggerUserList

on userList

after insert

as

begin

declare @id numeric(8,0);

declare @permission nvarchar(2);

select @permission=Permission,@id=Id from inserted;

if(@permission='学生')

begin

if not exists (select \* from Student where Id=@id)

insert into Student(Id) values(@id);

end

else if(@permission='厨师')

begin

if not exists (select \* from Chief where Id=@id)

insert into Chief(Id) values(@id);

end

else if(@permission='商户')

begin

if not exists (select \* from Window where Id=@id)

insert into Window(Id) values(@id);

end

end

菜品信息触发器：（菜品信息录入时，如果所填菜系不属于八大菜系，则将其菜系归为其他）

create trigger triggerDish

on Dish

after insert

as

begin

declare @dishname nvarchar(10);

declare @category nvarchar(50);

select @dishname=Dish,@category=Catagory from inserted;

if(@category!='川' and @category!='鲁'and @category!='粤'and @category!='苏'and @category!='浙'and @category!='闽'and @category!='湘'and @category!='徽')

begin

update dbo.Dish set Catagory='其他' where Dish=@dishname;

end

end

厨师菜品触发器：（当厨师更改其菜谱时，会同时修改厨师所在窗口提供的菜品信息）  
create trigger triggerDishChief  
on Dish\_Chief  
after insert  
as  
begin  
 declare @dishname nvarchar(10);  
 declare @chiefid numeric(8,0);  
 declare @wid numeric(8,0);  
  
 select @dishname=DishName,@chiefid=ChiefId from inserted;  
 select @wid=WinId from Chief\_Win where ChiefId=@chiefid;  
 if(@dishname not in (select DishName from Win\_Dish where WinId=@wid))  
 begin  
 insert into Win\_Dish values(@wid,@dishname);  
 end  
end

**五、存储过程的定义与实现**

CREATE PROCEDURE [dbo].[inUserList]

@username numeric(8,0)

AS

BEGIN

select Id from dbo.userList where Id=@username

END

CREATE PROCEDURE [dbo].[pswMatch]

@username numeric(8,0)，

@psw nchar(20)

AS

BEGIN

select \* from dbo.userList where Id=@username and Password=@psw

END

CREATE PROCEDURE [dbo].[getPermission]

@username numeric(8,0)

AS

BEGIN

select Permission from dbo.userList where Id=@username

END

CREATE PROCEDURE [dbo].[add2UserList]

@username numeric(8,0)，

@psw nchar(20)，

@permission nvarchar(2)

AS

BEGIN

insert into dbo.userList values(@username,@psw,@permission)

END

CREATE PROCEDURE [dbo].[getWinNameByWinId]

@wid numeric(8,0)

AS

BEGIN

select WinName from dbo.Window where Id=@wid

END

CREATE PROCEDURE [dbo].[getAllWinIdRestNamePairs]

AS

BEGIN

select \* from dbo.Win\_Rest

END

CREATE PROCEDURE [dbo].[getWinIdRestNamePairsByRestName]

@restName nvarchar(10)

AS

BEGIN

select \* from dbo.Win\_Rest where RestName=@restName

END

CREATE PROCEDURE [dbo].[UpdateStdMsg]

@id numeric(8,0)，

@name nvarchar(10)，

@sex nchar(1)，

@age tinyint，

@birthplace nvarchar(3)

AS

BEGIN

update dbo.Student set Name=@name,Age=@age,Sex=@sex,Birthplace=@birthplace where Id=@id

END

CREATE PROCEDURE [dbo].[InsertStdMsg]

@id numeric(8,0)，

@name nvarchar(10)，

@sex nchar(1)，

@age tinyint，

@birthplace nvarchar(3)

AS

BEGIN

insert into dbo.Student values(@name,@id,@sex,@age,@birthplace)

END

CREATE PROCEDURE [dbo].[getStdMsg]

@username

AS

BEGIN

select \* from dbo.Student where Id=@username

END

CREATE PROCEDURE [dbo].[getStdRec]

AS

BEGIN

END

CREATE PROCEDURE [dbo].[getDishInfoByWinId]

@wid numeric(8,0)

AS

BEGIN

select Dish,Price,Catagory from dbo.Win\_Dish,dbo.Dish where WinId=@wid and DishName=Dish

END

CREATE PROCEDURE [dbo].[getDishInfoByDishName]

@DishName nvarchar(10)

AS

BEGIN

select \* from dbo.Dish where Dish=@DishName

END

CREATE PROCEDURE [dbo].[getAllDish]

AS

BEGIN

select \* from dbo.Dish

END

CREATE PROCEDURE [dbo].[getWinNamefromDishName]

@DishName nvarchar(10)

AS

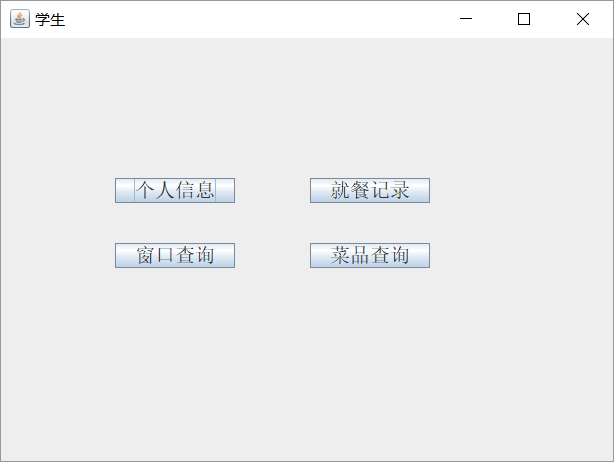
BEGIN

select distinct Window.Id,Window.WinName from dbo.Window,dbo.Win\_Dish,dbo.Dish where dbo.Window.Id = dbo.Win\_Dish.WinId and Win\_Dish.Dishname=@DishName

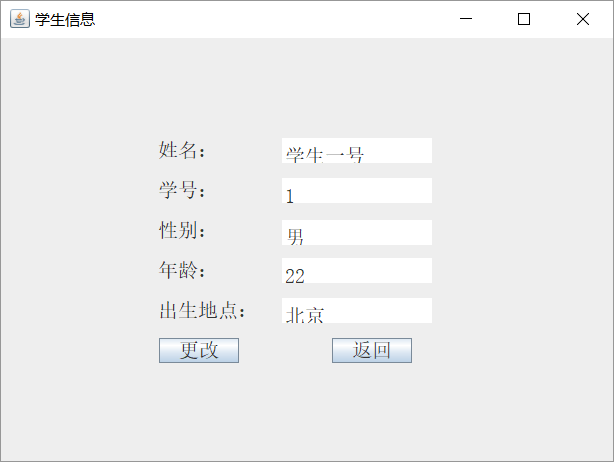
END

**六、系统实现结果**

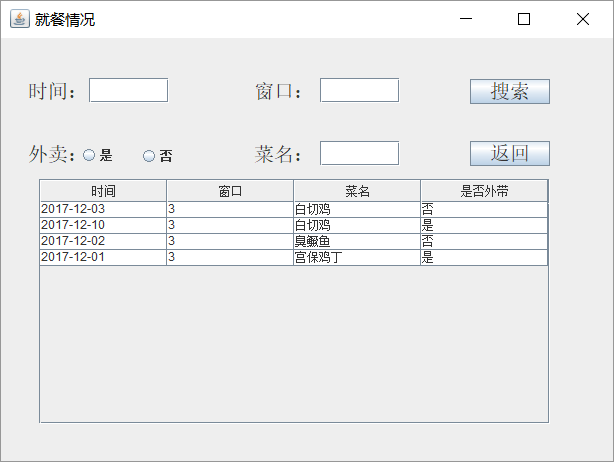
以学生身份登陆数据库



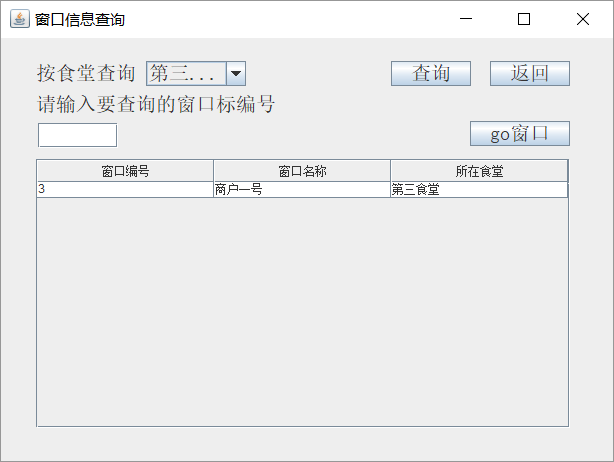
可以有如下功能：



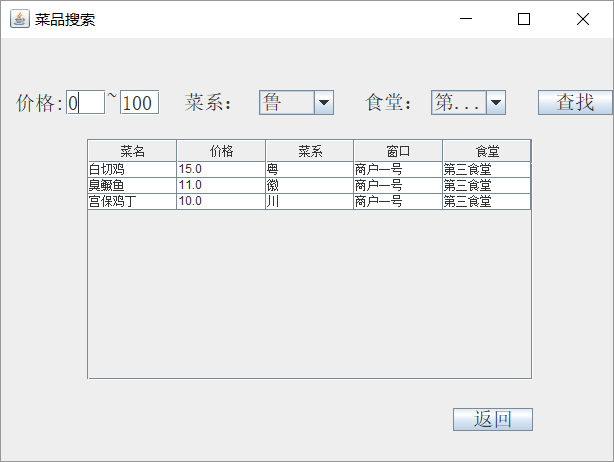
1. 查询以及修改个人信息



1. 查询自己的就餐记录，并可以通过添加约束条件进行自定义查询



1. 查看所有食堂窗口以及其提供的菜品信息，并可以按食堂查看该食堂的所有窗口



1. 查看所有食堂的菜品，并可以按价格，菜系，食堂，进行自定义查询

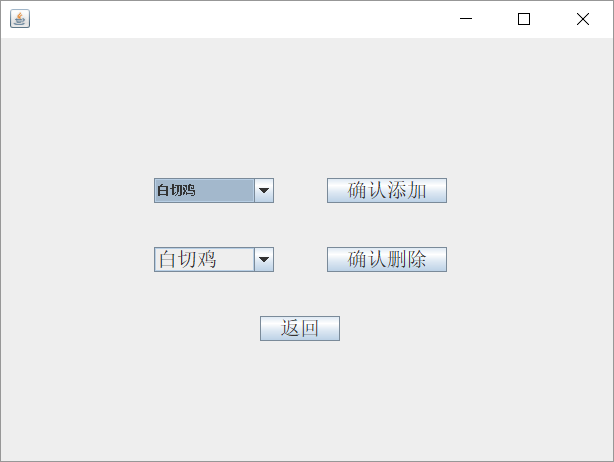
以厨师身份登陆数据库



可以有如下功能：

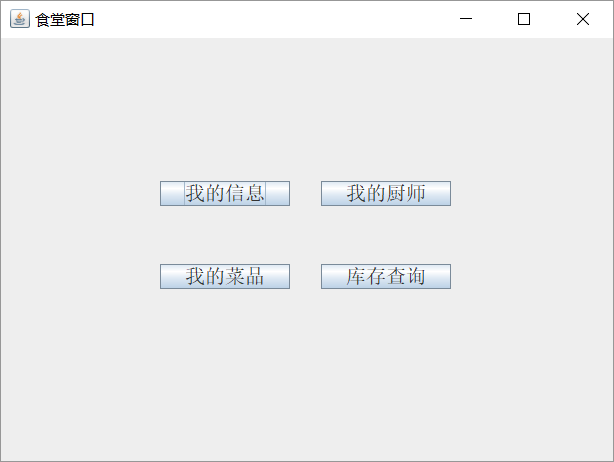


1. 查询个人以及更改个人信息，其中工资，所属窗口，以及编号，不允许厨师个人更改



1. 查看我的菜谱，并且可以实现对菜谱的更新（即添加菜品到我的菜谱，或者从我的菜谱中删除某菜品）

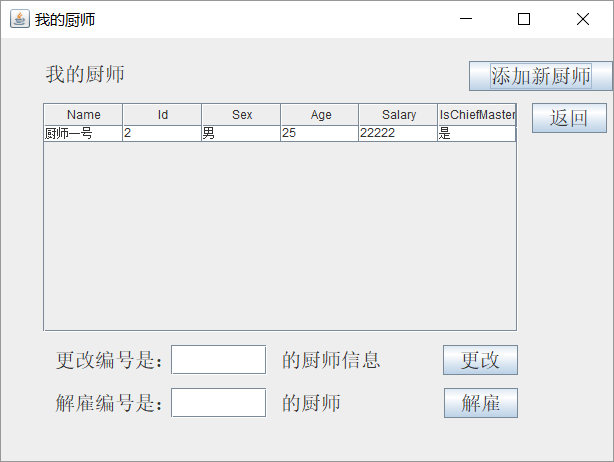
以窗口商户身份登陆数据库



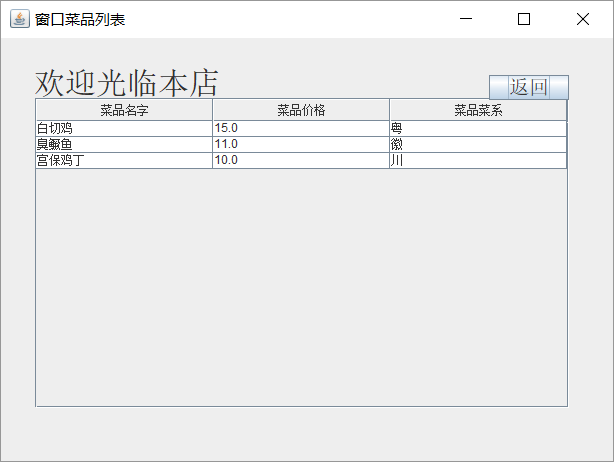
可以有如下功能：



1. 查询或者更改商户信息



1. 查看我的厨师的信息；添加新厨师，即聘用该厨师；更改某厨师的信息，即修改该厨师的薪资，或者将其升职为厨师长；解雇某厨师



1. 查看本窗口提供的菜品的信息，即本窗口的厨师的菜谱并集



1. 查询本窗口所在食堂的食材库存情况

以后台管理员身份登陆数据库



主要涉及如下功能：

1. 将菜品信息，即菜品名字，价格，所属菜系，添加入库，或者进行更新
2. 将食堂名字添加入库
3. 将供货商信息，即供货商ID，供货商名字，添加入库
4. 将供货商提供的食材添加入库
5. 将哪个供货商提供哪些食材，单价，数量等信息添加入库
6. 将供货商和食堂的对应关系添加入库
7. 将学生的就餐信息添加入库
8. 将某窗口和某食堂的对应关系添加入库

**七、总结**

通过这次数据库实验，让我们对于书上很多的内容掌握的更好，比如书上的触发器，在没有使用之前，很多我们想要达到的复杂约束条件都很难以实现，同时我们也明白了一个数据库的设计并不是简简单单的创几个表格就可以了，其中有非常多的约束条件需要我们考虑，而且怎么样才能让数据库拥有更少的冗余信息只有自己实践后才知道，包括在实践阶段发现我们对于一些之前想的内容也有点太过于简单，并没有真正的理解某些设计的含义导致实现数据库的时候出现了错误。总之，我们在这次数据库的实现还是受益颇多的