

# 餐馆订餐系统

姓名	学号	备注
鲍黎	2015335550094	组长
徐红	2015335550084	

目录

餐馆订餐系统.....1

用例文档.....4

1. 系统简介.....4

    1.1 系统名称.....4

    1.2 功能简介.....4

2. 系统业务列表.....4

3. 用例图.....5

4. 用例描述.....5

    4.1 UC1: 登录.....6

    4.2 UC2 : 记录预约.....6

    4.3 UC3: 记录到达.....7

    4.4 UC4: 查询预约.....7

    4.5 UC5: 取消预约.....7

    4.6 UC6: 临时分配.....8

    4.7 UC7 : 调换餐桌.....8

    4.8 UC8: 用餐完成.....8

    4.9 UC9: 注销.....9

类图文档.....9

1 类之间的联系图.....9

    2.1 员工类: .....10

    2.4 顾客类: .....12

    2.5 餐桌类: .....12

    2.6 预定记录类: .....13

    2.7 查询记录类.....13

    2.8 增加记录类.....14

    2.9 删除记录类.....14

    2.10 调换餐桌类.....14

    2.11 临时分配类.....15

    2.12 界面类.....15

    2.13 显示界面类.....16

    2.14 隐藏界面类.....16

    2.15 关闭界面类.....17

    2.16 BookingSystem 类.....18

时序图 .....18

    1. 顾客预约.....18

    2. 改变餐桌.....20

    3. 临时分配餐桌.....21

    4. 查询预约记录.....22

    5. 删除预约记录.....23

协作图 .....23

    1. 顾客预约.....23

    2. 改变餐桌.....25

---

3. 临时分配餐桌.....	26
4. 查询预约记录.....	27
5. 删除预约记录.....	28
总结 .....	28

# 用例文档

## 1. 系统简介

### 1.1 系统名称

餐馆订餐系统

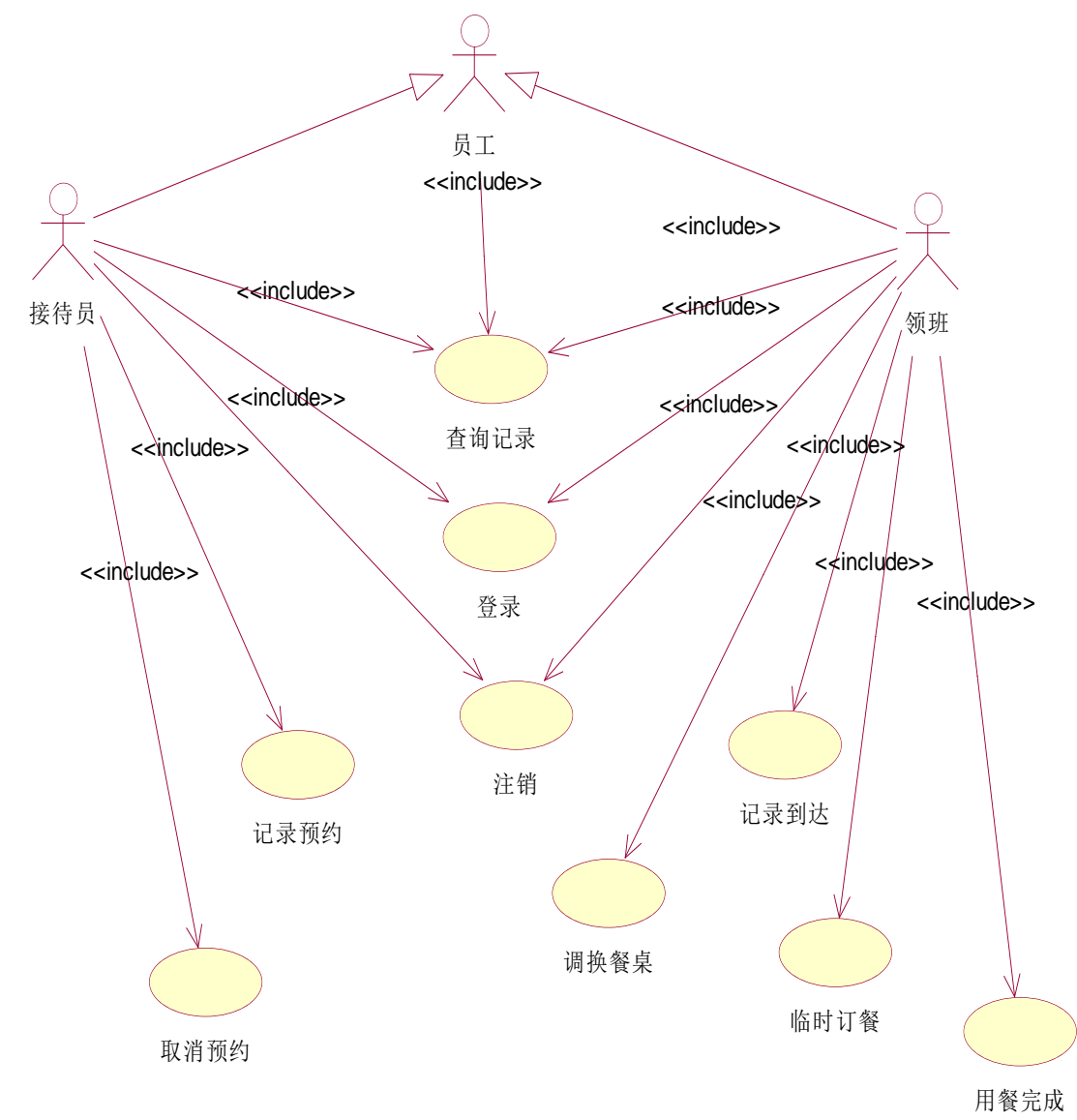
### 1.2 功能简介

本系统用于餐馆日常预约订餐，主要分为两类员工使用本系统：接待员和领班。接待员接受顾客请求，使用该系统为顾客预定餐桌，取消顾客已预定的餐桌记录；领班使用本系统，确认顾客的到来，无预定的顾客临时到来就餐分配临时餐桌，顾客就餐时调换餐桌，确认顾客用餐完毕；接待人员与领班都可查询已订餐的记录。

## 2. 系统业务列表

参与者	用例	说明
接待员	登录	登录系统
	查询预约	查询某时间的预约情况
	查询餐桌	查询某时间的餐桌使用
	记录预约	增加预约餐桌记录
	取消预约	取消某次餐桌预约记录
	注销	退出系统
领班	登录	登录系统
	查询预约	查询某时间的餐桌预约情况
	记录到达	顾客到达餐馆，更新记录、餐桌的状态
	临时订餐	顾客的临时订餐
	调换餐桌	顾客的调换餐桌请求
	用餐完成	顾客用餐完成，修改餐桌状态
	注销	退出系统

3. 用例图



4. 用例描述

4.1 UC1：登录

接待员   领班 • 登录
登录用户选择用户类型，输入账号密码 If 用户类型是接待员 then 查询数据库，匹配账号密码 if 账号密码有误 then 系统提示账号或密码错误 else then 系统跳转到接待员主菜单界面 end if else then 查询数据库，匹配账号密码 if 账号密码有误 then 系统提示账号或密码错误 else then 系统跳转到领班主菜单界面 end if end if

4.2 UC2 ：记录预约

接待员 • 记录预约
接待员输入要预约的日期 系统显示该日的预约 if 有合适的餐桌可以使用 then 接待员输入顾客的姓名和电话、预约的日期、用餐人数和餐桌号 系统增加此项记录，显示预约成功 返回预约编号 else if 有餐桌但餐桌用餐人数小于顾客需要的用餐人数 then 接待员输入顾客的姓名和电话、预约的时间，用餐人数和餐桌号 预约用餐人数大于餐桌能容纳的人数，系统发出警告信息询问用户是否预约 if 回答否 then 终止预约 else then 确定预约，并附有警告标志 返回预约编号 end if else then 没有合适的餐桌可以使用 用例终止 end if

#### 4.3 UC3：记录到达

领班•记录到达
领班输入当前日期 系统显示当天的预约 输入预约编号 领班确认一个预约已经到达 系统对此记录并更新餐桌状态 顾客无预约，还有合适未分配的餐桌

#### 4.4 UC4：查询预约

员工•查询预约
选择查询类型 if 选择类型为日期 then 输入一个日期 if 日期无效 then 系统提示输入的日期有误，请重新输入 else then 系统查询数据库并显示指定日期的预约信息 end if else then 输入预约编号 if 数据库查询无此记录 then 系统提示无此记录 else then 系统显示此预约记录的信息 end if end if

#### 4.5 UC5：取消预约

接待员•取消预约
接待员输入需要取消预约的预约编号 系统在数据库中查询此预约编号 if 系统查询无此预约编号 then 系统提示出错：无此预约编号 else then 接待员选择该预约

接待员取消预约 系统询问接待员确认取消 接待员回答是，系统记录取消并更新显示 end if
--

#### 4.6 UC6：临时分配

领班•临时分配
领班输入当前日期 系统显示当前预约 if 有合适的餐桌 then 领班输入到达时间，用餐人数和餐桌号 系统创建一个临时就餐记录 系统记录并更新餐桌状态 else then 无合适的餐桌 用例终止

#### 4.7 UC7：调换餐桌

领班•调换餐桌
领班查询预约 if 有合适的餐桌 then 取消当前预约 领班输入当前时间、用餐人数和餐桌号 系统记录，返回预约编号并显示预约成功 else then 没有合适的餐桌可调换，用例终止 end if

#### 4.8 UC8：用餐完成

领班•用餐完成
顾客用餐完成 领班输入餐桌号 系统修改餐桌状态 系统查询数据库是否有预约记录 if 有预约记录 then 修改预约记录 end if

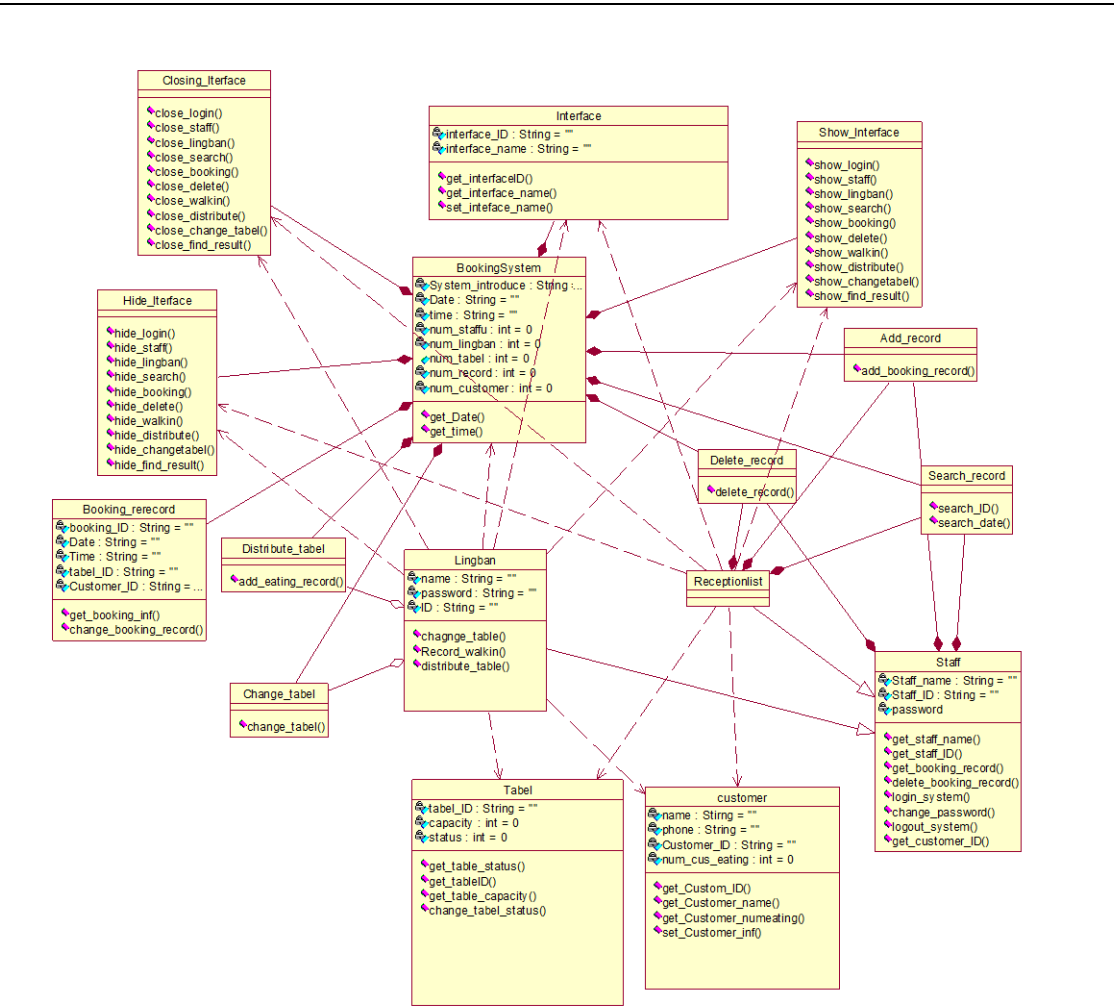


4.9 UC9：注销

接待员   领班 • 注销
用户选择注销 系统退出当前账户登录 返回登录页面

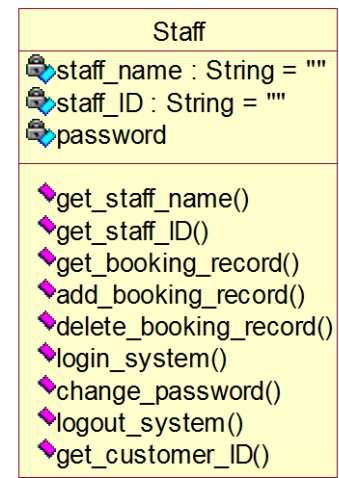
类图文档

1 类之间的联系图



2 系统类

2.1 员工类:



相关信息:

Lingban 类继承于 Staff 类；Receptionist 类继承于 Staff 类

属性：

staff\_ID——餐馆员工的 ID 号

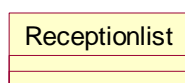
staff\_name——餐馆员工的姓名

password——餐馆员工登录餐馆预约订餐系统时的密码

方法：

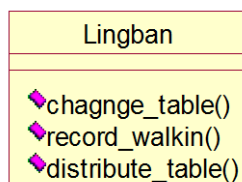
- (1) get\_staff\_name() ——获取餐馆员工的姓名
- (2) get\_staff\_ID() ——获取餐馆员工的 ID 号
- (3) find\_booking\_record() ——接待员或者领班查询餐馆的预约记录
- (4) delete\_booking\_record() ——接待员或者领班在餐馆预约系统中删除一条已有预定记录
- (5) login\_system() ——接待员或者领班登陆餐馆预约订餐系统
- (6) change\_password() ——接待员或者领班修改登录密码
- (7) logout\_System() ——接待员或者领班注销，退出登录系统
- (8) get\_customer\_ID() ——获取顾客预定餐桌时的预约 ID 号
- (9) add\_Booking\_record() ——接待员或者领班在餐馆预约系统中增加一条新的预定记录

## 2.2 接待员类：



相关信息： Receptionist 类继承于 Staff 类

## 2.3 领班类：

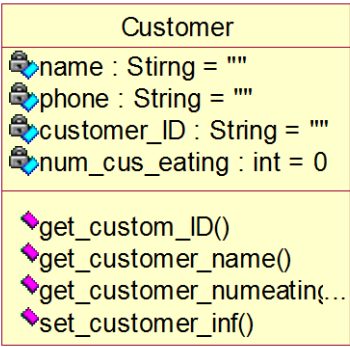


相关信息： Lingban 类继承于 Staff 类

方法：

- (1) change\_table() ——对于顾客就餐人数改变的情况，领班为顾客调换餐桌
- (2) record\_walking() ——记录到达（顾客到达餐厅）
- (3) distribute\_tabel() ——对于未进行预约的顾客，领班临时为顾客分配餐桌进行就餐

2.4 顾客类:



相关信息： Customer 类与 Lingban 类、Staff 类、Receptionist 类是依赖关系

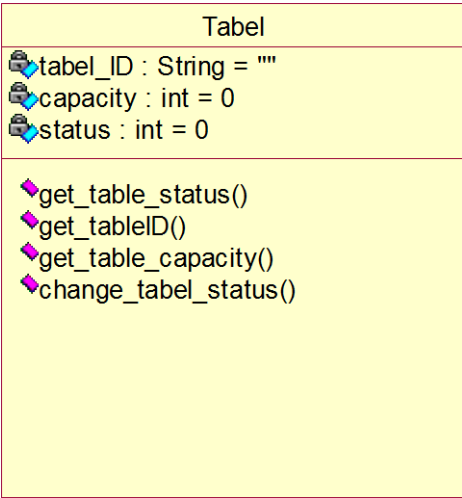
属性:

- phone——顾客的姓名
- name——顾客的联系电话
- customer\_ID——顾客预约时的预约 ID 号
- num\_cus\_eating——顾客的就餐人数

方法:

- (1) get\_customer\_ID() ——获取顾客预约时的预约号
- (2) get\_customer\_name() ——获取顾客的姓名
- (3) get\_customer\_numeating() ——获取顾客的就餐人数
- (4) set\_customer\_inf() ——登记顾客的信息（如：姓名，就餐人数，联系方式等）

2.5 餐桌类:



相关信息： Tabel 类与 Lingban 类、Receptionist 类是依赖关系

属性:

table\_ID——餐桌的 ID 号

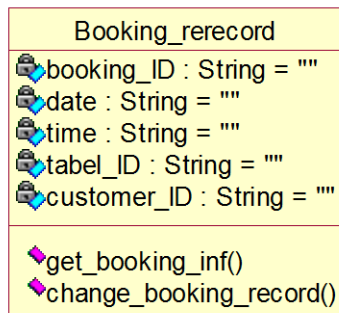
capacity——餐桌所能容纳的人数

status——餐桌此时的状态（如：已预约、正在使用、空闲）

方法:

- (1) get\_tabel\_status() ——获取餐桌此时的状态（如：已预约、正在使用、空闲）
- (2) get\_tabelID() ——获取餐桌的 ID 号
- (3) get\_tabel\_capacity() ——获取餐桌所能容纳的人数
- (4) change\_tabel\_status() ——改变餐桌的状态（如：将“空闲状态”→“已预约”）

## 2.6 预定记录类:



相关信息:Booking\_record 类与 Lingban 类、Receptionist 类、BookingSystem 类是依赖关系

属性:

booking\_ID——餐馆预约系统中预约记录的编号 ID

date——顾客预约的日期

time——顾客预约的时间

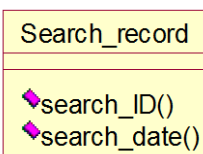
tabel\_ID——顾客预约时，接待员分配给顾客的餐桌号 ID（即顾客就餐的餐桌号）

customer\_ID——顾客预约时的预约号 ID

方法:

- (1) . get\_booking\_inf() ——获取顾客预约时的所登记的预约信息
- (2) . change\_booking\_record() ——接待员修改餐馆预约系统中已有的预定记录

## 2.7 查询记录类

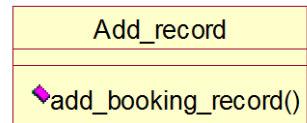


相关信息:Search\_record 类与 Lingban 类、Receptionist 类、BookingSystem 类是依赖关系

方法:

- (1) search\_ID()——领班或接待员按记录 ID 查询餐馆预约系统中顾客的预约信息
- (2) search\_date()——领班或接待员按时间查询餐馆预约系统中顾客的预约信息

2.8 增加记录类

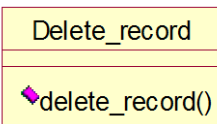


相关信息: Add\_record 类与 Lingban 类、Receptionist 类、BookingSystem 类是依赖关系

方法:

- (1) add\_booking\_record()——领班或接待员为餐馆预约系统添加一条新的预约记录

2.9 删除记录类

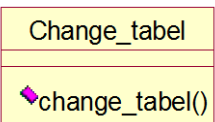


相关信息:Delete\_record 类与 Lingban 类、Receptionist 类、BookingSystem 类是依赖关系

方法:

- (1) delete\_record()——领班或接待员删除餐馆预约系统中已有的预定记录

2.10 调换餐桌类

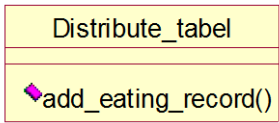


相关信息: Change\_tabel 类与 Lingban 类、BookingSystem 类是依赖关系

方法:

- (1) . Change\_tabel()——对于顾客就餐人数改变的情况，领班为顾客调换餐桌

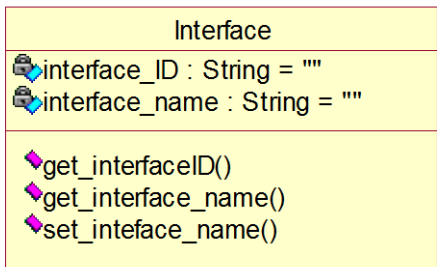
## 2.11 临时分配类



相关信息：Change\_tabel 类与 Lingban 类、BookingSystem 类是依赖关系  
方法：

(1) . add\_eatig\_record()——对于未预约的顾客，领班为顾客临时分配一个餐桌，在餐馆预约系统中增加一条新的就餐记录

## 2.12 界面类



相关信息：Interface 类与 Lingban 类、Receptionist 类、BookingSystem 类是依赖关系

属性：

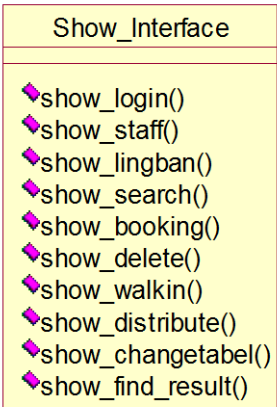
interface\_ID——各种不同界面的界面号 ID（如：登录界面、查询记录界面等）

interface\_name——界面的名称（如：登录界面、查询记录界面等）

方法：

- (1) get\_interfaceID()——获取界面的界面号 ID
- (2) get\_interface\_name()——获取界面的名称
- (3) set\_interface\_name()——设置界面的名称

### 2.13 显示界面类

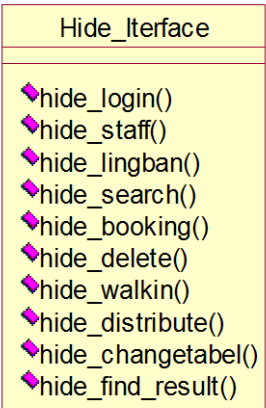


相关信息：Show\_interface 类与 Lingban 类、Receptionist 类、BookingSystem 类是依赖关系

方法：

- (1) show\_login() ——显示登录界面
- (2) show\_staff() ——显示接待员主菜单界面
- (3) show\_lingban() ——显示领班主菜单界面
- (4) show\_search() ——显示查询记录界面
- (5) show\_booking() ——显示预定记录界面
- (6) show\_delete() ——显示删除预定界面
- (7) show\_walkin() ——显示处理记录到达界面
- (8) show\_distribute() ——显示临时分配餐桌界面
- (9) show\_changetabel() ——显示调换餐桌界面
- (10) show\_find\_result() ——显示查询结果界面

### 2.14 隐藏界面类



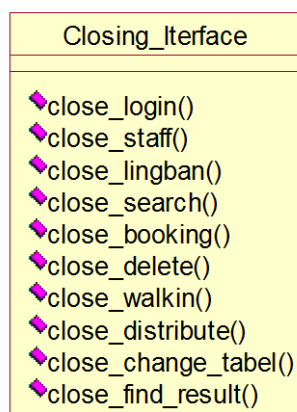
相关信息：Hide\_interface 类与 Lingban 类、Receptionist 类、BookingSystem 类是依赖关系



方法:

- (1) hide\_login()——隐藏登录界面
- (2) hide\_staff()——隐藏接待员主菜单界面
- (3) hide\_lingban()——隐藏领班主菜单界面
- (4) hide\_search()——隐藏查询记录界面
- (5) hide\_booking()——隐藏预定记录界面
- (6) hide\_delete()——隐藏删除预定界面
- (7) hide\_walkin()——隐藏处理记录到达界面
- (8) hide\_distribute()——隐藏临时分配餐桌界面
- (9) hide\_changetable()——隐藏调换餐桌界面
- (10) hide\_find\_result()——隐藏查询结果界面

## 2.15 关闭界面类

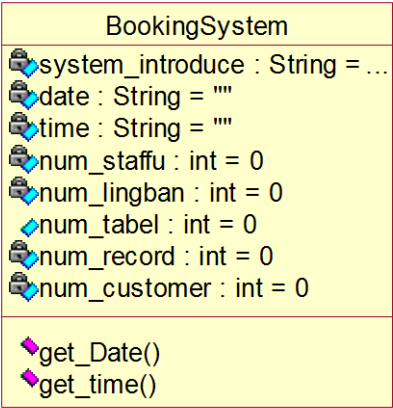


相关信息: Closing\_Interface 类与 Lingban 类、Receptionist 类、BookingSystem 类是依赖关系

方法:

- (1) close\_login()——关闭登录界面
- (2) close\_staff()——关闭接待员主菜单界面
- (3) close\_lingban()——关闭领班主菜单界面
- (4) close\_search()——关闭查询记录界面
- (5) close\_booking()——关闭预定记录界面
- (6) close\_delete()——关闭删除预定界面
- (7) close\_walkin()——关闭处理记录到达界面
- (8) close\_distribute()——关闭临时分配餐桌界面
- (9) close\_changetable()——关闭调换餐桌界面
- (10) close\_find\_result()——关闭查询结果界面

### 2.16 BookingSystem 类



相关信息：Interface 类与 Lingban 类、Receptionist 类、BookingSystem 类是依赖关系

属性：

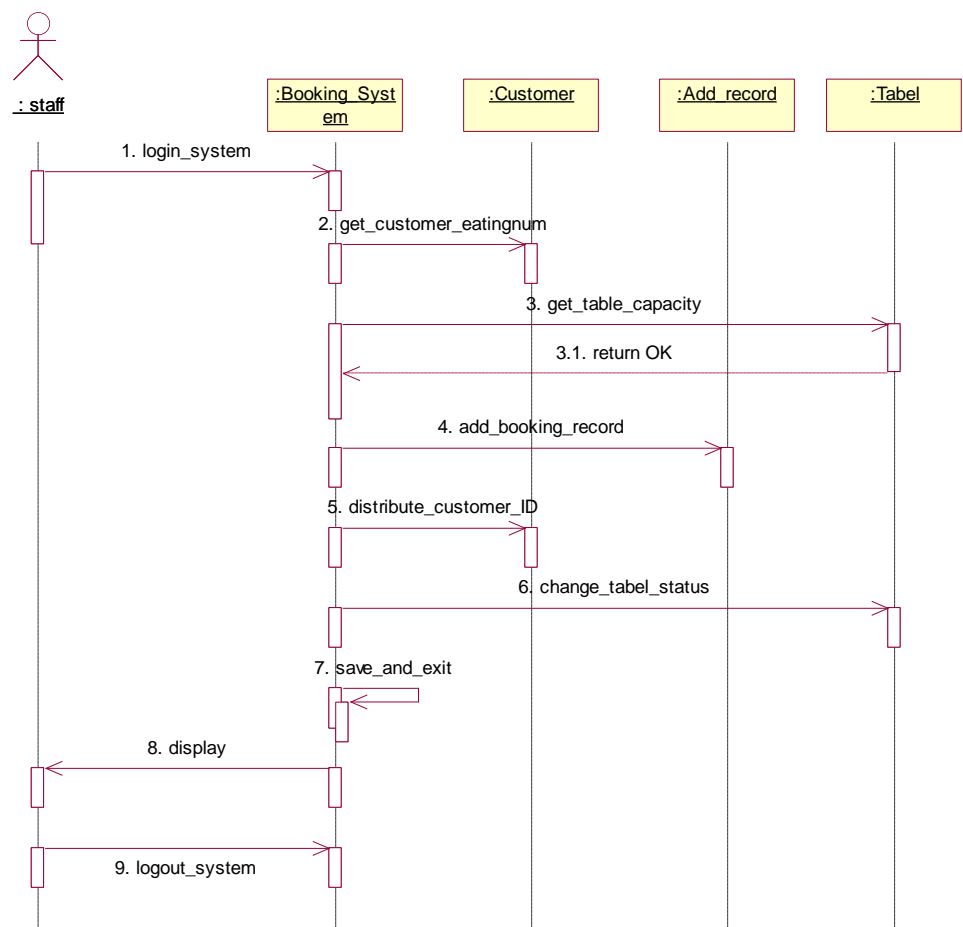
系统简介、日期、时间、接待员数量、领班数量、桌子数量、记录数量、顾客数量

方法：

- (1) 获取日期
- (2) 获取时间

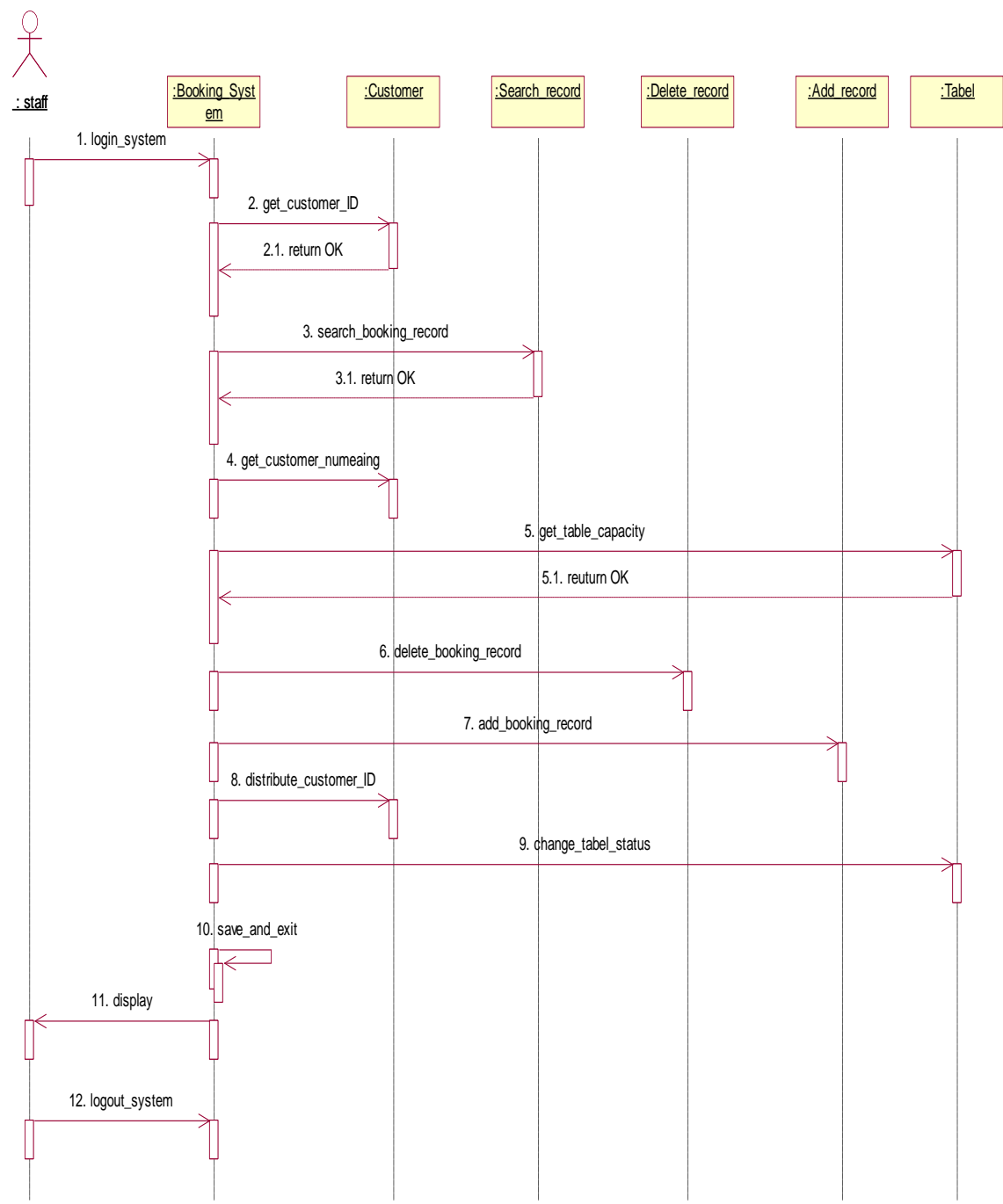
## 时序图

### 1. 顾客预约



当有顾客要订餐时，员工登陆餐馆预约系统——询问顾客订餐人数——查看当前空闲餐桌是否满足顾客需求——满足（若不满足，员工则退出餐馆预约系统）——添加一条新的预约记录——分配顾客一个预约 ID 号——分配餐桌，改变餐桌的状态——餐馆预约系统更新并保存——员工退出餐馆预约系统

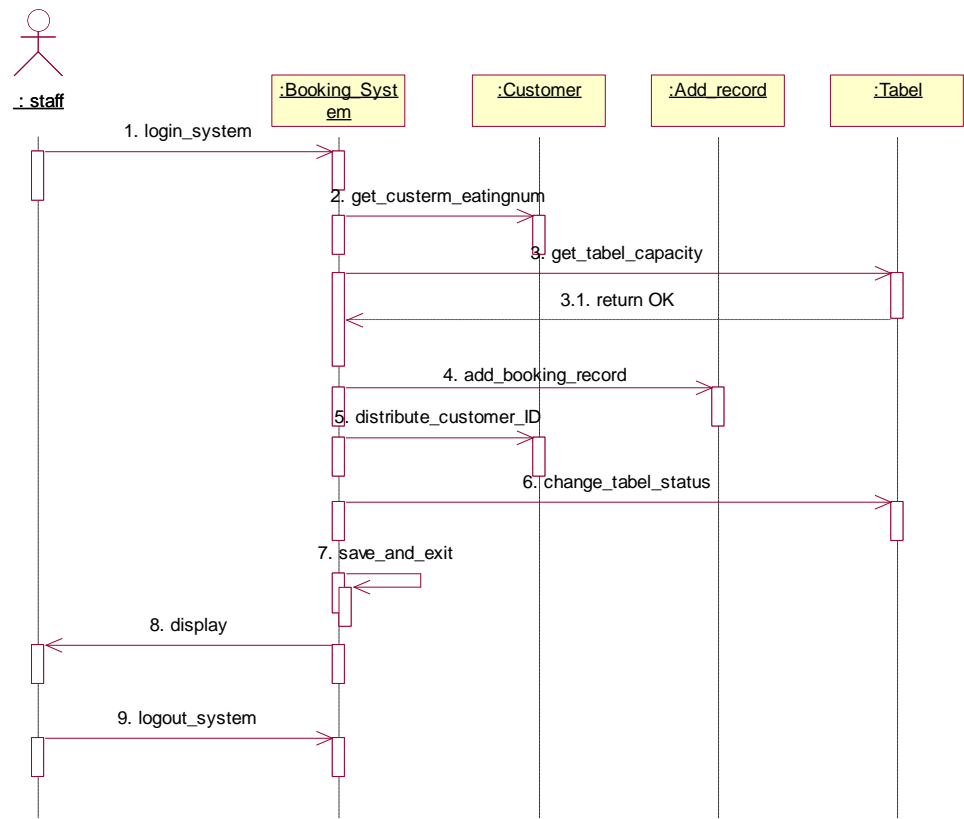
2. 改变餐桌



当顾客要求改变预约信息（即改变餐桌）时，员工登陆餐馆预约系统——询问顾客预约时的预约 ID 号——查询该预约——查询成功（若查询不成功，员工则退出餐馆预约系统）——询问顾客订餐人数——查询当前空闲餐桌——满足顾客需求（若不满足顾客需求，员工则退出餐馆预约系统）删除顾客原

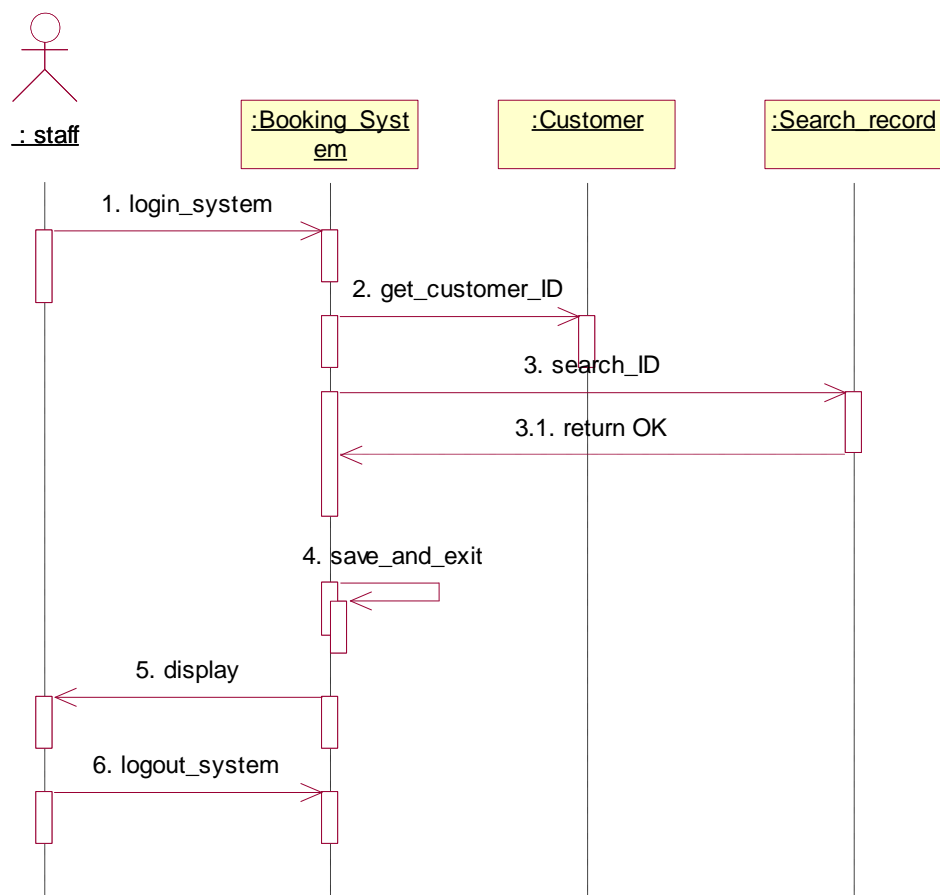
有的记录——添加一条新的预约记录——分配给顾客一个预约 ID 号——分配餐桌，改变餐桌的状态——餐馆预约系统更新并保存——员工退出餐馆预约系统

3. 临时分配餐桌



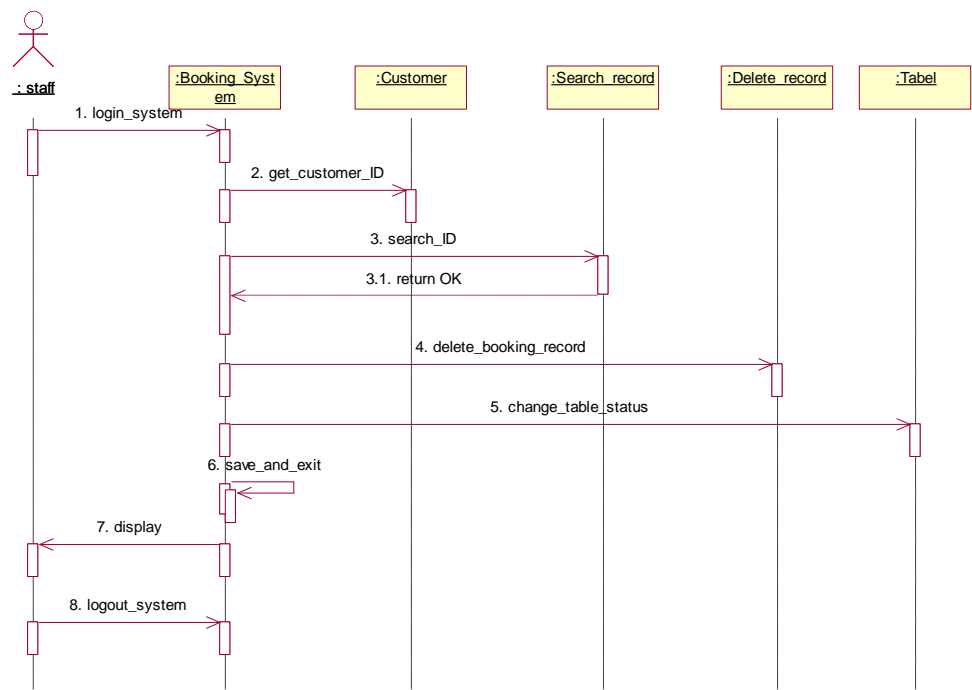
当有顾客临时到达餐厅，员工登陆餐馆预约系统——询问顾客就餐人数——查看当前空闲餐桌是否满足顾客需求——满足（若不满足，员工则退出餐馆预约系统）——添加一条新的预约记录——分配顾客一个预约 ID 号——分配餐桌，改变餐桌的状态——餐馆预约系统更新并保存——员工退出餐馆预约系统

#### 4. 查询预约记录



当需要查询餐厅预约信息时，员工登陆餐馆预约系统——询问顾客预约时的预约ID号——查询该预约——查询成功（若查询不成功，员工则退出餐馆预约系统）——餐馆预约系统更新并保存——员工退出餐馆预约系统

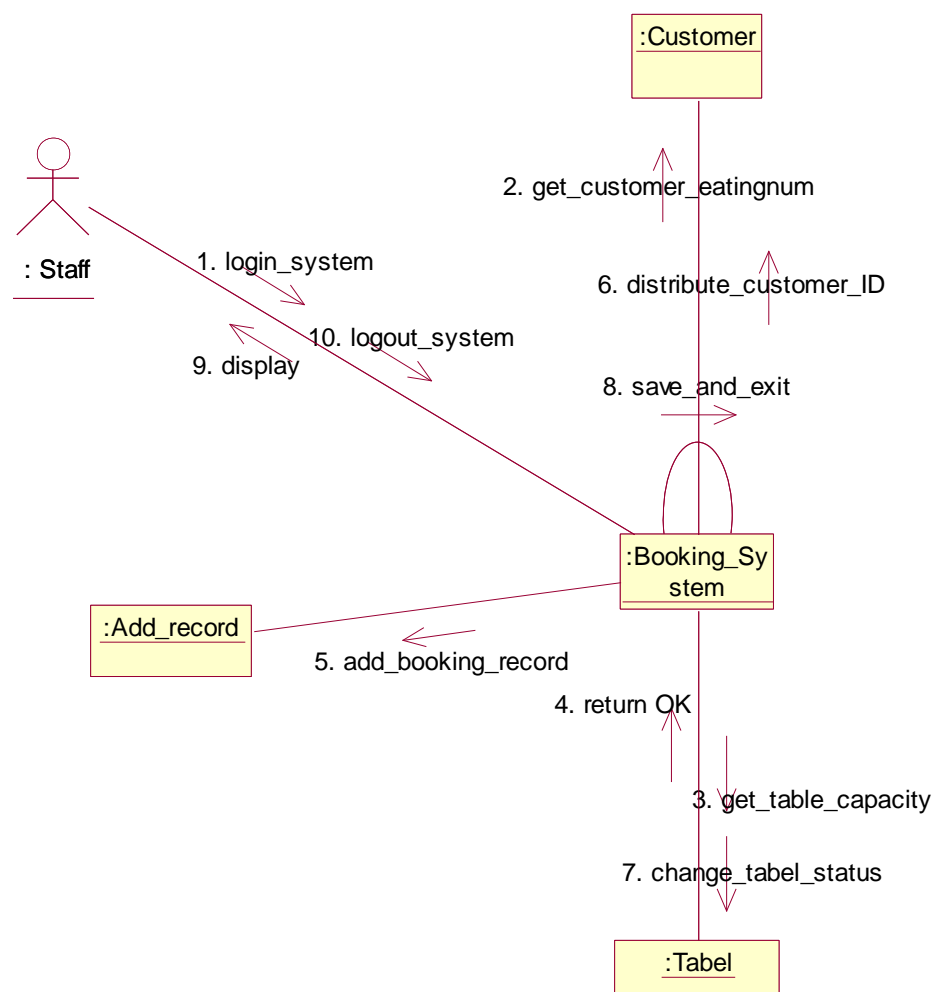
5. 删除预约记录



当需要删除餐厅预约信息时（如：顾客就餐结束），员工登陆餐馆预约系统——询问顾客预约时的预约 ID 号——查询该预约——查询成功（若查询不成功，员工则退出餐馆预约系统）——删除该预约信息——餐馆预约系统更新并保存——员工退出餐馆预约系统

协作图

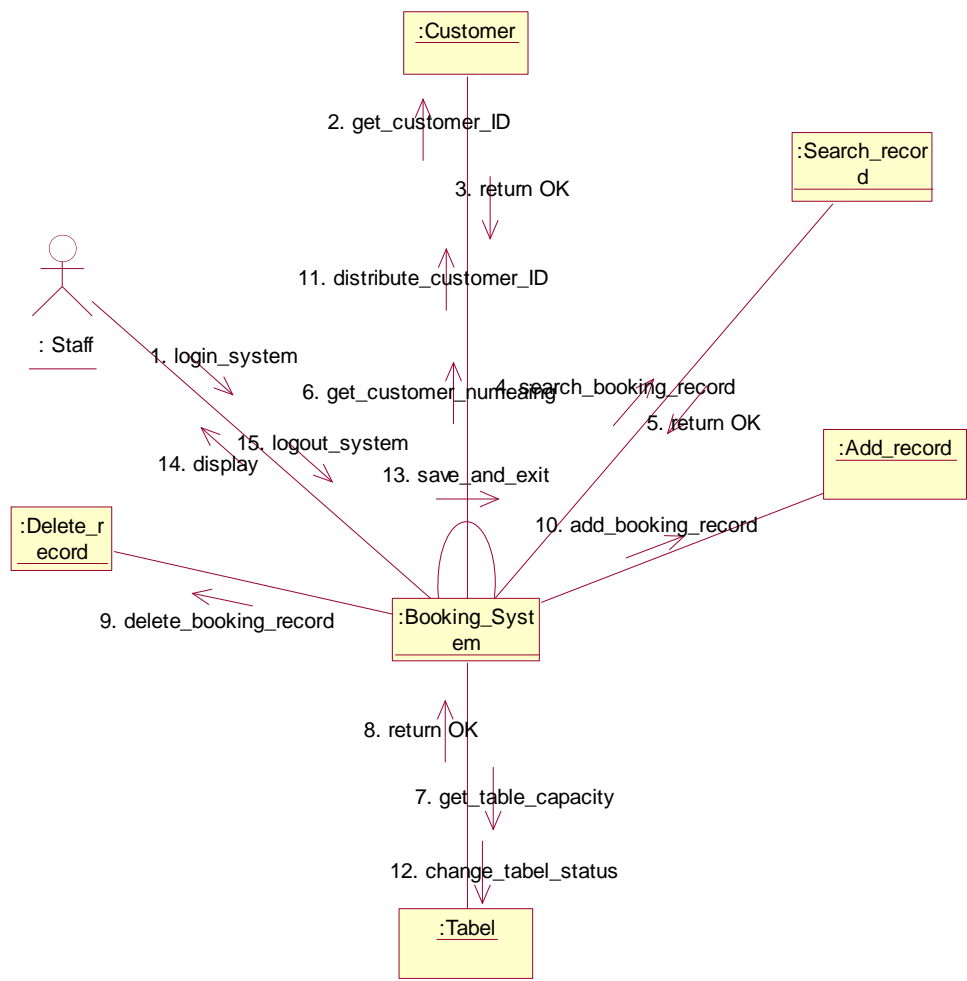
1. 顾客预约



当有顾客要订餐时，员工登陆餐馆预约系统——询问顾客订餐人数——查看当前空闲餐桌是否满足顾客需求——满足（若不满足，员工则退出餐馆预约系统）——添加一条新的预约记录——分配顾客一个预约 ID 号——分配餐桌，改变餐桌的状态——餐馆预约系统更新并保存——员工退出餐馆预约系统

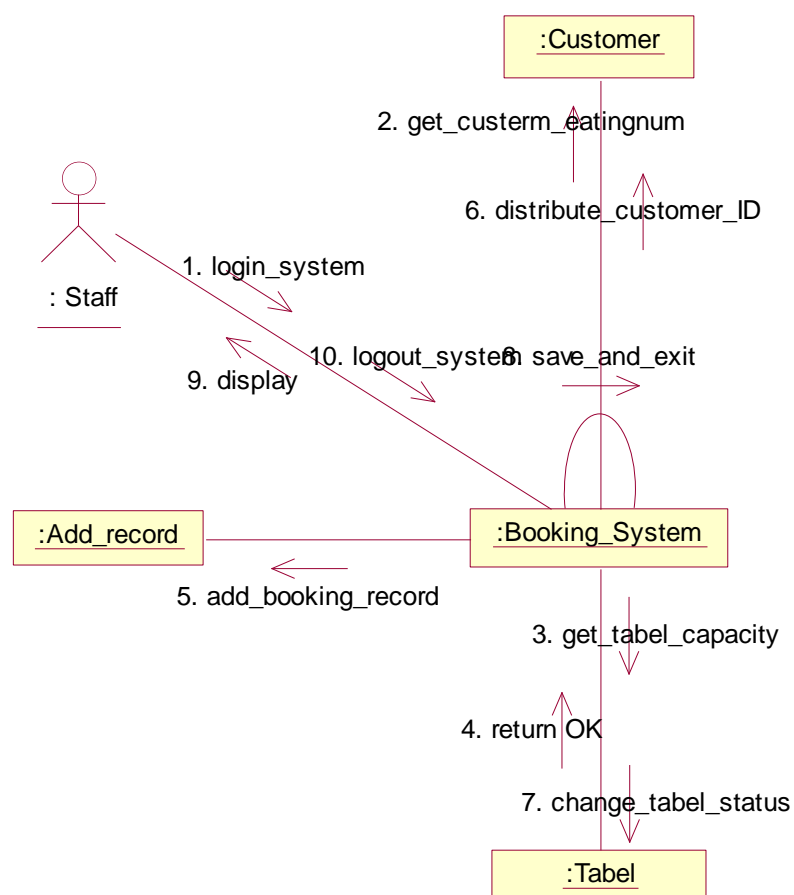


2. 改变餐桌



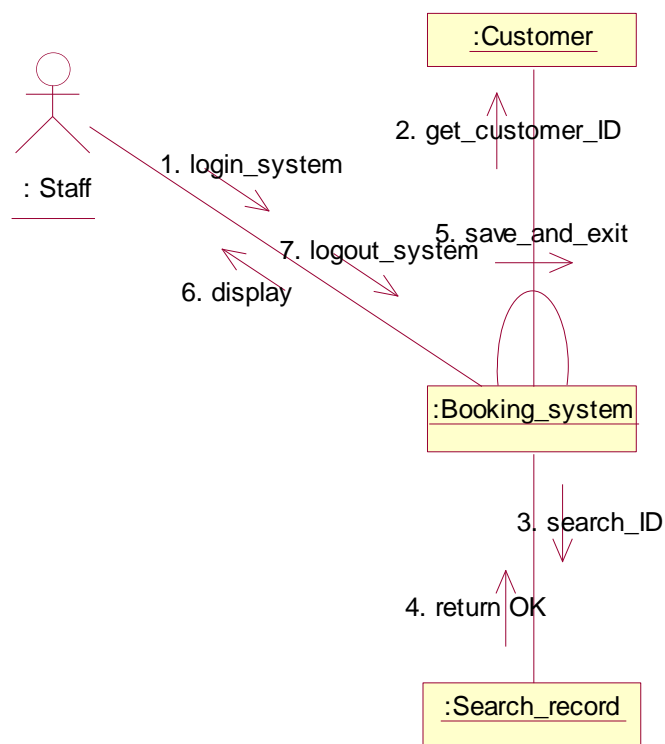
当顾客要求改变预约信息（即改变餐桌）时，员工登陆餐馆预约系统——询问顾客预约时的预约 ID 号——查询该预约——查询成功（若查询不成功，员工则退出餐馆预约系统）——询问顾客订餐人数——查询当前空闲餐桌——满足顾客需求（若不满足顾客需求，员工则退出餐馆预约系统）删除顾客原有的记录——添加一条新的预约记录——分配给顾客一个预约 ID 号——分配餐桌，改变餐桌的状态——餐馆预约系统更新并保存——员工退出餐馆预约系统

### 3. 临时分配餐桌



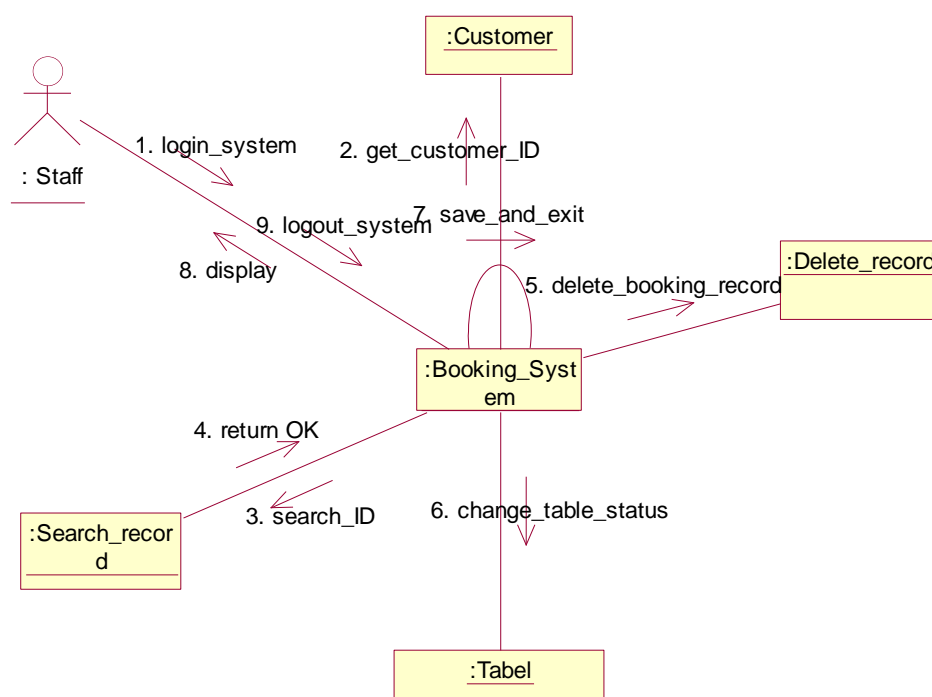
当有顾客临时到达餐厅，员工登陆餐馆预约系统——询问顾客就餐人数——查看当前空闲餐桌是否满足顾客需求——满足（若不满足，员工则退出餐馆预约系统）——添加一条新的预约记录——分配顾客一个预约 ID 号——分配餐桌，改变餐桌的状态——餐馆预约系统更新并保存——员工退出餐馆预约系统

#### 4. 查询预约记录



当需要查询餐厅预约信息时，员工登陆餐馆预约系统——询问顾客预约时的预约ID号——查询该预约——查询成功（若查询不成功，员工则退出餐馆预约系统）——餐馆预约系统更新并保存——员工退出餐馆预约系统

## 5. 删除预约记录



当需要删除餐厅预约信息时（如：顾客就餐结束），员工登陆餐馆预约系统——询问顾客预约时的预约 ID 号——查询该预约——查询成功（若查询不成功，员工则退出餐馆预约系统）——删除该预约信息——餐馆预约系统更新并保存——员工退出餐馆预约系统

## 总结

老实说，第一次上课的时候老师给我们将要做什么软件的时候，我是茫然的，原因如下：1、没有学过软件制作 2、没有写过软件界面 3、老师推荐的语言是 C#，没有学过。但是没有办法，必须得硬着头皮上。

首先是读老师给的资料，大概读了 3 遍吧，基本看懂老师的要求，要求我们写一个餐馆的订餐系统。然后就是做用例分析，读文档查资料，把用例都整出来了，大概花了 3 周吧，我觉得用例分析是第一步所以做得很慢，要考虑的事情很多很杂，还有就是第一次做这些事情没经验在黑暗中摸索着前进，用例图是用 IBM 公司的 rose 做，感觉很好用能画好多精美图，很好用的工具，在网上搜了几篇博客，跟着做，画好了用例图。这是做软件的第一步，后面一步是类的设计，我们组抽象出了 16 个类，包括实际存在的还是抽象类，我们都做了归纳，类的设计主要交给了另一位同学负责，这样使得我们组的任务分配平均。然后是协作图的设计，时序图的设计。

前期的工作基本完成，然后就是设计数据库了，数据库的设计还是比较简单的，在本系统中并没有用到很复杂的数据，所以数据库设计也很简

单，设计了 5 张表，分别是：Receptionist、Leader、Desk、Customer、Record。表与表之间有联系的，Receptionist 表是接待员表，记录接待员的基本信息，同理 Leader 表记录领班的基本信息，Desk 表记录餐桌的基本信息，特别需要强调的是 Desk 表，Desk 表是按照日期排序的，新的一天到来时，Desk 表会自动增加一天课预约的餐桌，Customer 表是记录顾客的基本信息表，Record 表是餐馆的记录表，记录预定，临时订餐记录，消费记录等。

最后一项是用面向对象的程序语言实现餐馆订餐，面对众多的面向对象程序设计语言，经过权衡考虑，我们组选择的是 C# 语言，简单易用，但是组员都没有去学过 C#，只要自己去自学 C#。虽然我们没学过，但是学起来还是挺轻松的，因为 C# 跟 C++ 非常像，而我也是非常爱 C++ 的，所以用了大概 4 周的时间学习了 C#，所学的知识够用来写个图形化界面的程序了，就开始动手写。我们没有采用老师给的文档的写法，只写了，图形部分，数据库部分，和主程序部分，因为我们觉得这样写方便些。写的过程中会发现一些问题，比如用例不全，就回去修改用例文档，各种修修补补。其实些代码的时间用不了多久，因为前期的分析很全面了，按照这个思路写代码用不了多久，所以我觉得前期的分析要占很重要的以部分，应该重视，后期写代码就会很快。程序写完之后就是测试，测试也是很重要的一部分，你不能将有 bug 的软件给用户吧，这样会流失用户。按照面向对象软件测试的方法，测试了几天，然后就宣告“餐桌订餐系统”制作完毕。

总结本次实验，收获挺多的。