复旦大学高等教育自学考试

（本科）

软件工程实践报告

题目\_\_\_\_长江宾馆客房信息管理系统\_\_\_\_\_\_

专 业\_\_\_\_\_\_\_\_计算机软件\_\_\_\_\_\_\_\_

作 者\_\_\_\_\_\_\_\_查伟昌\_\_\_\_\_\_\_\_\_\_\_\_

准考证号\_\_\_\_\_\_\_374216230112\_\_\_\_\_\_\_\_

指导老师\_\_\_\_\_\_\_\_\_于玉\_\_\_\_\_\_\_\_\_\_\_\_\_

完成时间 2017 年 05 月 12 日

**目录**

[第1章 绪言 3](#_Toc482302008)

[1.1 项目开发背景及信息系统目标 3](#_Toc482302009)

[1.2 信息系统范围 4](#_Toc482302010)

[第2章 建立初步对象模型 4](#_Toc482302011)

[2.1 需求分析 4](#_Toc482302012)

[2.2 确定类 4](#_Toc482302013)

[2.3 确定属性 5](#_Toc482302014)

[2.4 确定关联 8](#_Toc482302015)

[2.5 包图 9](#_Toc482302016)

[2.6 GUI图 10](#_Toc482302017)

[2.7 总图 11](#_Toc482302018)

[第3章 建立时序图 11](#_Toc482302019)

[3.1 客房预订 11](#_Toc482302020)

[3.2 客房入住 13](#_Toc482302021)

[3.3 退房 13](#_Toc482302022)

[第4章 软件模块处理 14](#_Toc482302023)

[4.1 登录界面中验证口令的脚本 14](#_Toc482302024)

[4.2 预订界面中包含的脚本 14](#_Toc482302025)

[4.3 入住界面中包含的脚本 15](#_Toc482302026)

# 绪言

# 项目开发背景及信息系统目标

近年来，宾馆业迅猛发展，市场的竞争日趋激烈，全面提高宾馆的软件管理水准，已成为宾馆业发展的当务之急。尤其是对于星级宾馆，既需要完成前台的一些服务工作，还需要完成后台的管理工作。然而，传统的人工管理模式已经远远不能满足有效、快捷地处理经营中产生的大量信息数据的需要，从而使得企业决策层无法及时、准确地掌握一线资料，继而影响对市场进行正确地分析和预测。像沿海城市三星级以上宾馆引进外方管理，使小部分宾馆管理水准几乎接近或达到国际水平。但对占80%以上的广大中小型宾馆来说，是难以做到的。因此，欲在竞争中甩开对手，取得优势，必须在经营、管理、产品、服务等方面具备独到之处。而对宾馆的经营状况起决定作用的是客房的管理。简单的服务标准已不是制胜的锦囊，只有管理做到最细微之处，才能让顾客体会到宾馆服务的高标准、高质量，而准确、快速、周全往往就是最基本的成功要素。

传统的管理方法已经不能适应现代社会的需要，因此采用电脑管理业务、财务等诸多环节已成为推动宾馆业迅速发展的先决条件，宾馆客房管理信息系统是各大中小型宾馆所需要使用的一个管理系统。

系统目标是建立一套功能完善的管理信息系统，既能满足业务人员日常处理的需要，增强企业经营全过程的数字化管理水平；又能满足管理人员决策分析的需要，提高公司管理层对公司经营反馈信息的响应速度。从而大幅度提高工作效率，提高实施管理的准确性、科学性，使担负管理的工作人员从繁杂的手工劳作中解脱出来。

# 信息系统范围

适用中小类型的酒店，提供基本的预订，入住，退房和人员管理服务。

# 建立初步对象模型

# 需求分析

在分析现行手工操作的基础上，进一步研究客房管理的处理流程，确定系统的总体结构及功能模块，研究各个模块之间的联系，确定模块之间的接口方案。依照实用性、可靠性、可扩充性、经济性的原则，设计有关的各类码表及相关的程序代码，并验证代码的正确性，设计一套合理、完整的测试方案，对系统进行测试，是系统达到用户的使用要求。

系统开发的总体任务是实现酒店客房各种信息的系统化、规范化和可操作性。要解决的几个关键问题：

1. 合理的设计数据库及数据表。

作为酒店客房管理系统，应具有信息的录入，删除，修改，浏览等基本功能，还要求能够根据用户的需要进行操作。

 (2) 设计出友好的界面

  界面的友好与否是用户评价一个软件优劣的重要方面。应设计登陆后系统的主窗口，要求能够显示酒店客房管理的基本信息，桌面尽量少用菜单，数据的录入，修改，删除、查询等功能按钮直接分布在主界面上，且布局合理，美观。尽量减少用户的键盘输入，减少用户的工作量，提高工作效率。

# 确定类

类名称：客房类型，客房，员工，宾馆，预订，入住，入住客户，退房

# 确定属性

类名：Hotel

类描述：宾馆基本信息

类属性：名称，位置，联系电话

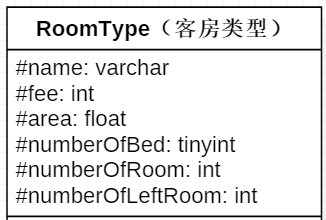


图表 1 类图 - 宾馆

类名：RoomType

类描述：客房类型信息

类属性：房型名称，费用，面积，床位数，房间数

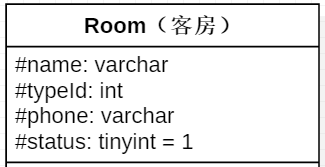


图表 2类图 - 客房类型

类名：Room

类描述：客房信息

类属性：名称，客房类型编号，状态

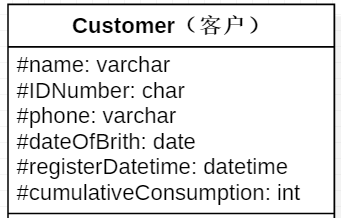


图表 3类图 - 客房

类名：Customer

类描述：用户信息

类属性：姓名，身份证号，电话，生日，注册时间

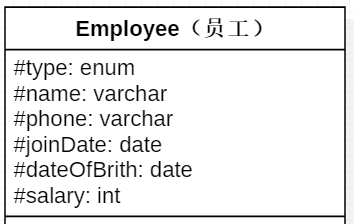


图表 4类图 - 客户

类名：Employee

类描述：员工信息。员工分为负责人，接待员，前台员工，收银员

类属性：类型，姓名，电话，入职时间，生日，基本工资

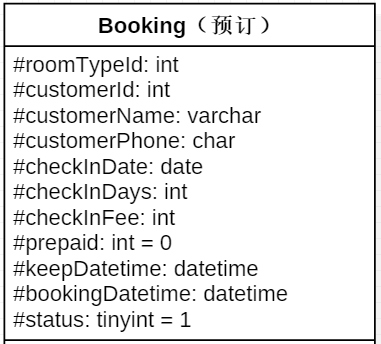


图表 5类图 - 员工

类名：Booking

类描述：客户预订客房信息，包括用户信息和入住信息

类属性：房间类型编号，客户编号，客户姓名，客户手机号，入住日期，入住天数，预计房费，预付金额，最晚预留时间，预订时间，状态

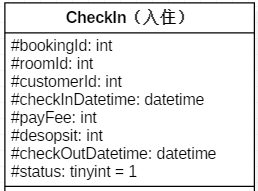


图表 6类图 - 预订

类名：CheckIn

类描述：客户入住信息，包括用户信息和客房信息

类属性：预定编号，客房编号，客户编号，预计房费，已缴房费，押金，预计退房时间，入住时间，状态

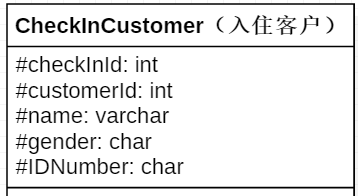


图表 7类图 - 入住

类名：CheckInCustomer

类描述：入住客户信息

类属性：入住编号，姓名，性别，身份证号

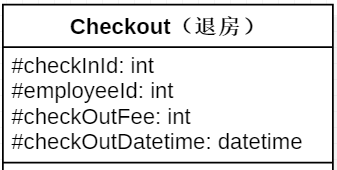


图表 8类图 - 入住客户

类名：Checkout

类描述：客户退房信息

类属性：入住编号，房费，其他消费（消费，物品损坏），退房时间



图表 9类图 - 退房

# 确定关联

a、客房类型与客房，一种客房类型会有0到n个客房

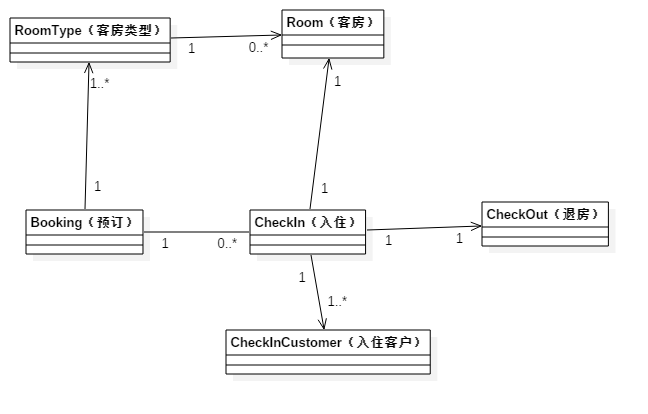
b、预定与客房类型，一次可以预定多个房间类型

c、入住与客房，一一对应

d、预订与入住，没有直接关联。预订可能不会产生入住，也可能会有1到多个入住；入住可能是通过预订来的，也可能不是

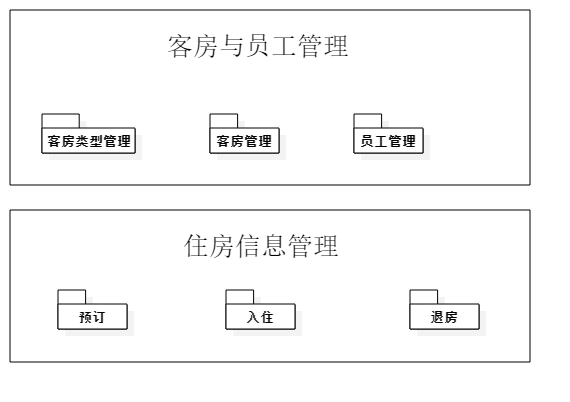
e、入住与入住客户，一对多

f、入住与退房，一对一



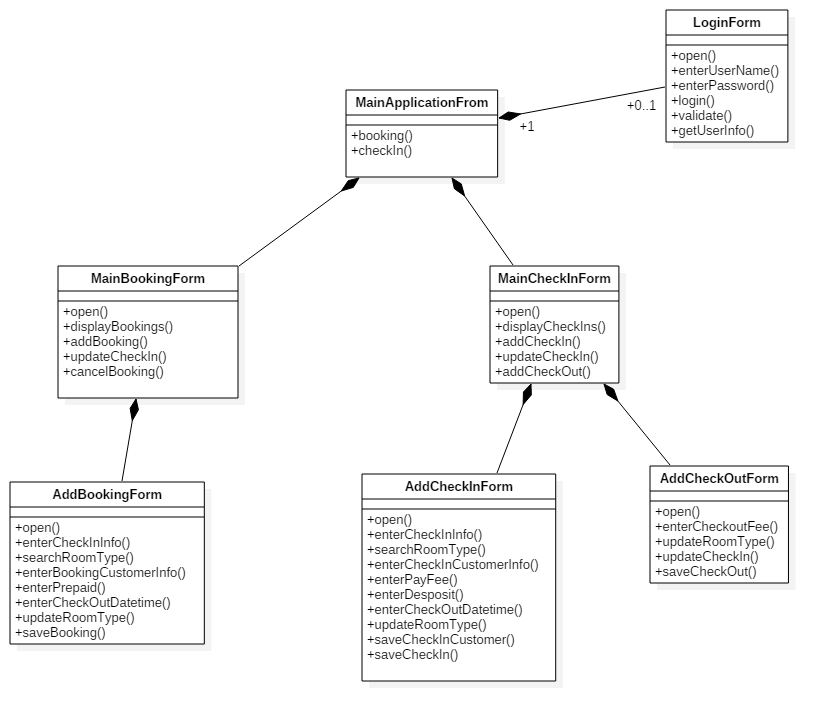
图表 10关系图

# 包图



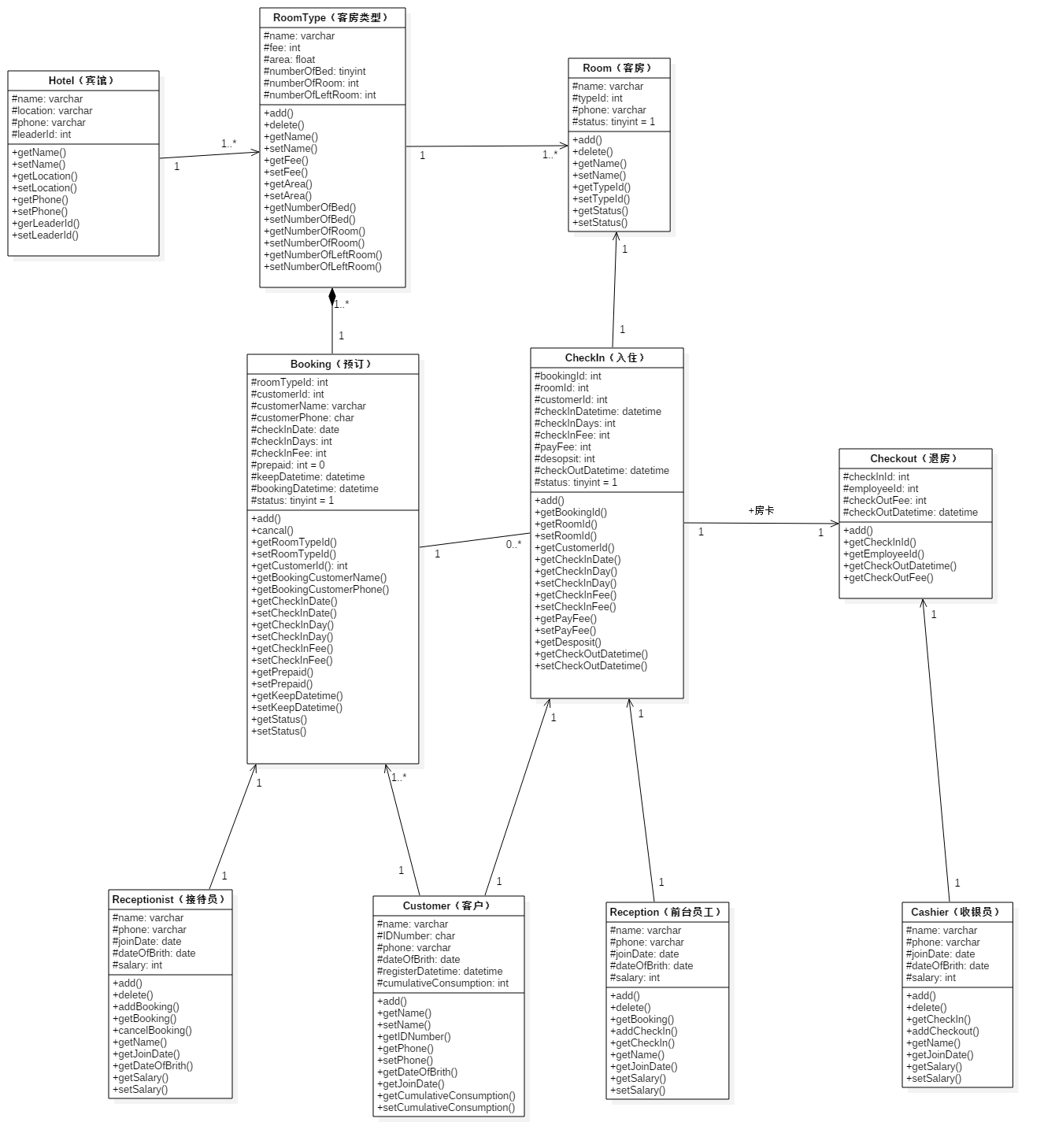
图表 11包图

# GUI图



图表 12 GUI图

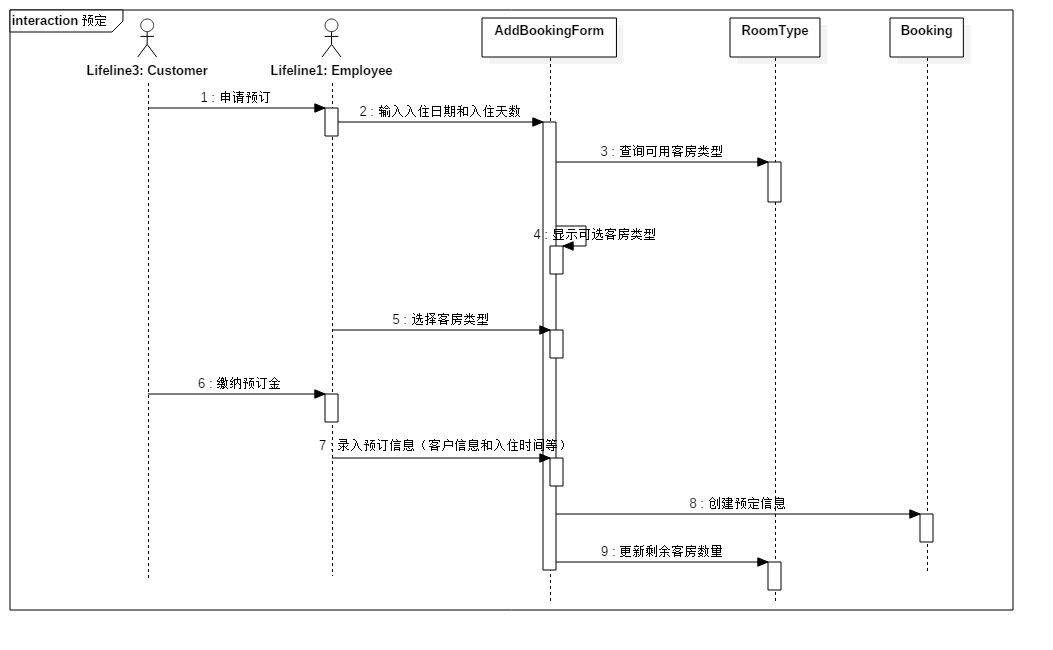
# 总图



图表 13实体类图

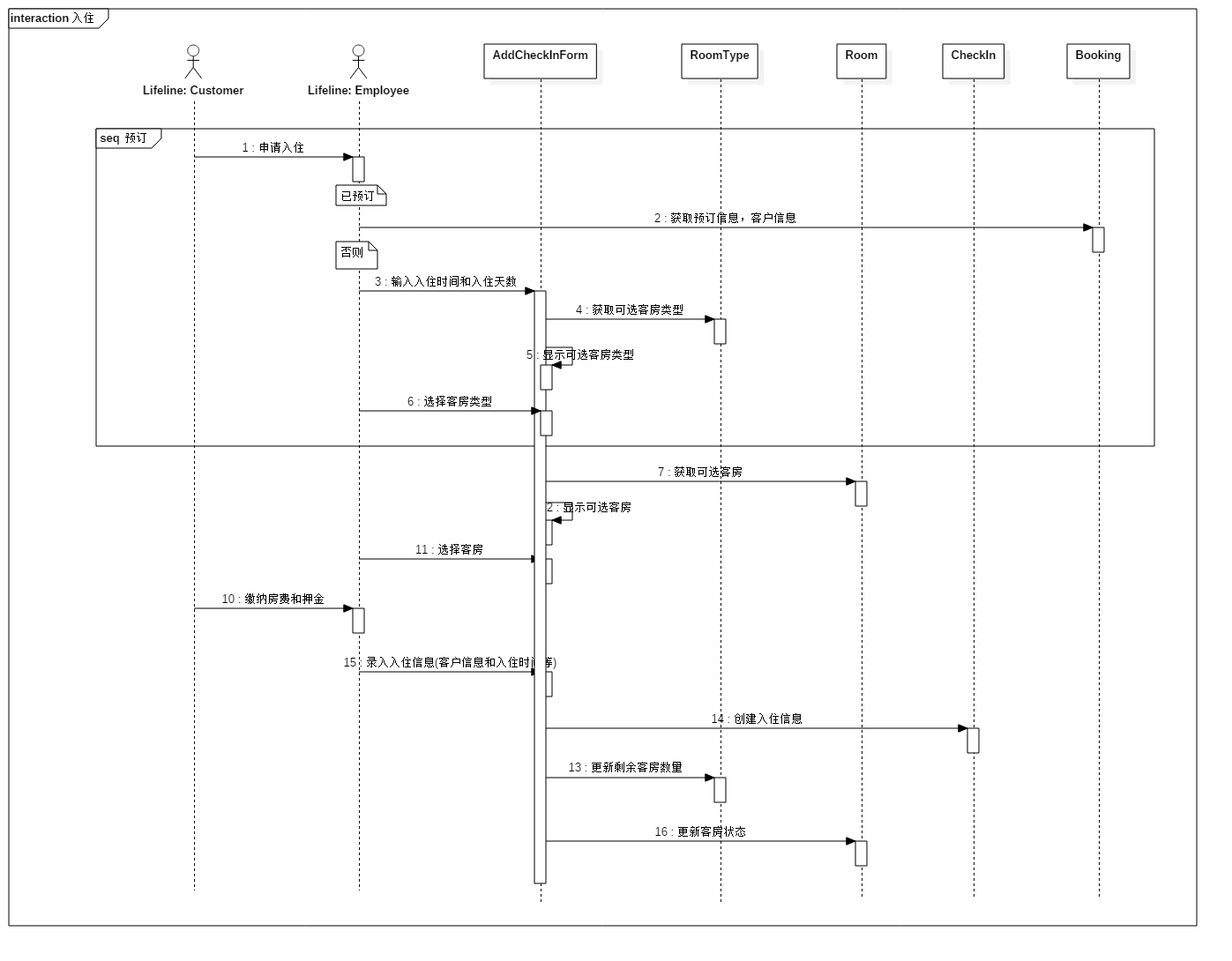
# 建立时序图

# 预订



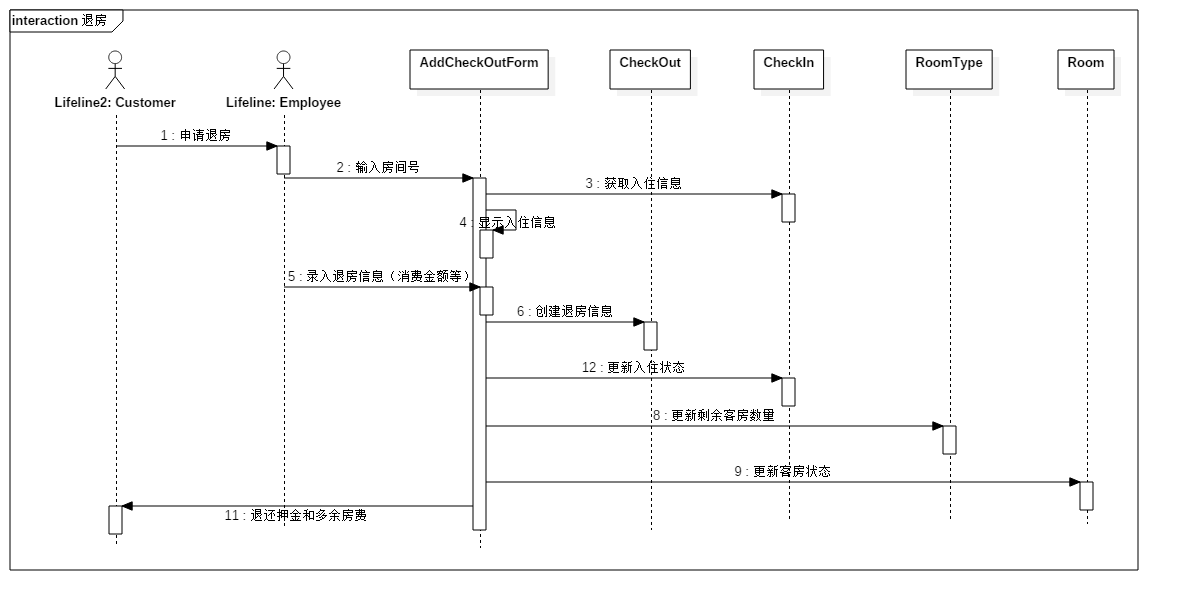
图表 14时序图 - 客房预订

# 客房入住



图表 15时序图 - 客房入住

# 退房



**图表 16时序图 - 退房**

# 软件模块处理

# 登录界面中验证口令的脚本

public function login($name, $password)

{

$result = DB::table('user')

->where('name', $name)

->get()

->first();

if (!$result) {

return "用户不存在";

}

if (password\_verify($password, $result->password)) {

return "登录成功";

} else {

return "用户名或密码错误";

}

}

# 预订界面中包含的脚本

$input = $request->all();

$checkInDate = $input['check\_in\_date'];

$checkInDays = $input['check\_in\_days'];

$amount = $input['amount'];

$i = 0;

$datetime = date("Y-m-d H:i:s");

// 更新剩余房间类型

do {

$date = date("Y-m-d", strtotime("+$i days", strtotime($checkInDate)));

$result = DB::table('room\_type\_date\_check\_in')

->where('room\_type\_id', $input['room\_type\_id'])

->where('date', $date)

->get()

->first();

if ($result) {

DB::table('room\_type\_date\_check\_in')

->where('room\_type\_id', $input['room\_type\_id'])

->where('date', $date)

->update(['number\_of\_check\_in\_room' => $result->number\_of\_check\_in\_room + $amount, 'updated\_at' => $datetime]);

} else {

$dateCheckIn = [

'room\_type\_id' => $input['room\_type\_id'],

'date' => $date,

'number\_of\_check\_in\_room' => $amount,

'created\_at' => $datetime,

'updated\_at' => $datetime,

];

DB::table('room\_type\_date\_check\_in')

->insert($dateCheckIn);

}

$i++;

} while ($i < $checkInDays);

Booking::create($request->all());

return redirect()->route(config('quickadmin.route').'.booking.index');

# 入住界面中包含的脚本

$input = $request->all();

$checkInDate = $datetime = date("Y-m-d");

$checkInDays = $input['check\_in\_days'];

if ($input['booking\_id']) {

DB::table('booking')

->where('id', $input['booking\_id'])

->update(['status' => 2]);

} else {

// 更新剩余房间类型

$amount = $input['amount'];

$i = 0;

$datetime = date("Y-m-d H:i:s");

do {

$date = date("Y-m-d", strtotime("+$i days", strtotime($checkInDate)));

$result = DB::table('room\_type\_date\_check\_in')

->where('room\_type\_id', $input['room\_type\_id'])

->where('date', $date)

->get()

->first();

if ($result) {

DB::table('room\_type\_date\_check\_in')

->where('room\_type\_id', $input['room\_type\_id'])

->where('date', $date)

->update(['number\_of\_check\_in\_room' => $result->number\_of\_check\_in\_room + $amount, 'updated\_at' => $datetime]);

} else {

$dateCheckIn = [

'room\_type\_id' => $input['room\_type\_id'],

'date' => $date,

'number\_of\_check\_in\_room' => $amount,

'created\_at' => $datetime,

'updated\_at' => $datetime,

];

DB::table('room\_type\_date\_check\_in')->insert($dateCheckIn);

}

$i++;

} while ($i < $checkInDays);

}

// 更新房间入住信息

$dateCheckIns = array();

$i = 0;

do {

$date = date("Y-m-d", strtotime("+$i days", strtotime($checkInDate)));

$dateCheckIn[] = [

'room\_id' => $input['room\_id'],

'date' => $date,

'status' => 2,

'created\_at' => $datetime,

'updated\_at' => $datetime,

];

$i++;

} while ($i < $checkInDays);

DB::table('room\_date\_check\_in')->insert($dateCheckIn);

DB::table('room')

->where('id', $input['room\_id'])

->update(['status' => 3]);

CheckIn::create($request->all());

return redirect()->route(config('quickadmin.route').'.checkin.index');