

## 目 录

1. 引言.....	3
1.1 编写目的.....	3
1.2 项目概述.....	3
1.3 术语定义.....	3
1.4 缩写说明.....	4
1.5 引用文档.....	4
2. 软件设计决策.....	5
2.1 设计目标.....	5
2.2 设计原则.....	5
2.3 设计约束.....	5
2.3.1 遵循标准.....	5
2.3.2 运行环境.....	5
2.3.3 开发环境及工具.....	6
2.3.4 技术限制.....	6
2.3.5 其他.....	6
3. 逻辑架构设计.....	7
3.1 设计决策.....	7
3.2 软件单元.....	7
3.2.1 界面层.....	8
3.2.2 业务逻辑层.....	9
3.2.3 持久化层.....	10
3.3 处理流程.....	11
3.3.1 管理剧目(TTMS_UC_01_1).....	11
3.3.2 管理演出厅(TTMS_UC_02_2).....	11
3.3.3 管理用户(TTMS_UC_02_3).....	12
4. 人机界面设计.....	13
4.1 初步方案(草图).....	13
4.2 最终效果.....	15
4.2.1 登录.....	15
4.2.2 演出厅管理.....	15
4.2.3 座位管理.....	16
4.2.4 用户管理.....	17

4.2.5	新增演出计划.....	17
4.2.6	剧目管理.....	18
4.2.7	售票.....	18
4.2.8	退票.....	20
4.2.9	异常输入判断以及消息提示.....	21
5.	数据存储设计.....	22
5.1	内部数据结构.....	22
5.2	数据库.....	22
5.3	数据文件.....	23
6.	详细设计.....	24
6.1	剧目管理(TTMS_UC_01_1).....	24
6.1.1	功能描述.....	24
6.1.2	处理流程.....	24
6.1.3	内部数据.....	24
6.1.4	异常与错误处理.....	25
6.1.5	测试要点.....	25
6.2	演出厅管理(TTMS_UC_02_1).....	25
6.2.1	功能描述.....	25
6.2.2	处理流程.....	26
6.2.3	内部数据.....	26
6.2.4	异常与错误处理.....	27
6.2.5	测试要点.....	27
6.3	用户管理(TTMS_UC_02_2).....	27
6.3.1	功能描述.....	27
6.3.2	处理流程.....	27
6.3.3	内部数据.....	28
6.3.4	异常与错误处理.....	28
6.3.5	测试要点.....	28
7.	开发架构设计.....	29
7.1	工程结构.....	29
8.	物理架构设计.....	30
8.1	网络环境.....	30
8.2	部署方案.....	30

# “HLW 剧院票务管理系统”软件设计说明

## 1. 引言

### 1.1 编写目的

本文档用于说明 HLW 剧院管理系统软件体系结构设计、接口设计和软件单元详细设计，是 HLW 剧院管理系统软件实现的基础。本文的预期读者包括：

- 开发人员
- 测试人员
- 项目管理人员

### 1.2 项目概述

本项目基本信息如下：

- 项目名称：HLW 剧院票务管理系统；
- 项目编号：HLW.2018.TTMS；
- 投资方：HLW 传媒有限公司；
- 用户：HLW 传媒有限公司下属各剧院；
- 开发方：啦啦啦啦公司。

### 1.3 术语定义

本文中用到的专门术语定义见表 1。

表 1 术语定义

序号	术语	含义
1	Theater Ticket Management System	剧院票务管理系统
2	Unified Modeling Language	统计建模语言
3	Use Case Diagram	用例图
4	View Layer	界面层
5	Service Layer	业务逻辑层
6	Persistence Layer	持久化层

1.4 缩写说明

表 2 英文缩写说明

序号	缩写	原文
1	TTMS	Theater Ticket Management System
2	UML	Unified Modeling Language

1.5 引用文档

本文引用的文档及标准参见表 3。

表 3 引用文档

序号	文档编号	标题	版本号	修订日期	编制单位
1	GB/T11457-2006	信息技术软件工程术语	—	2006/7/1	国务院标准化行政部门
2	GBT14394-1993	计算机软件可靠性和可维护性管理	—	2008/12/1	国务院标准化行政部门

## 2. 软件设计决策

### 2.1 设计目标

HLW 传媒有限责任公司目前使用手动管理剧院业务的方式,不仅需要大量的人力物力,而且很容易出现错误,因此需要开发一套用于数字化管理剧院业务的系统。系统要求简单高效,易于操作,数据安全,能同时满足 50 名以内售票员同时售票的需求。

### 2.2 设计原则

稳定性至上,最大限度保证系统不崩溃,即使系统崩溃,也要保证数据会及时保存下来。

安全性至上,系统的安全性直接影响着甲方业务数据的安全,如果安全性不能保证,那么对甲方来说是灾难性的。

### 2.3 设计约束

#### 2.3.1 遵循标准

本项目设计过程中所遵循的标准有：

1. GB/T 8567-2006 中华人民共和国国家标准 计算机软件文档编制规范
2. GB/T 11457-2006 中华人民共和国国家标准 信息技术 软件工程术语
3. GB/T 20157-2006 中华人民共和国国家标准 信息技术 软件维护
4. GB/Z 20156-2006 中华人民共和国国家标准 软件工程 软件生成周期过程用于项目管理的指南
5. GB/T 20158-2006 中华人民共和国国家标准 信息技术 软件生成周期过程配置管理
6. GB/T 16260-2006 中华人民共和国国家标准 软件工程 产品质量《企业信息化技术规范》 2006

#### 2.3.2 运行环境

数据库：MySQL5.7

操作系统：服务器端：centos7 、客户端：Windows 10/deepin15.6  
平台：jre1.8

### 2.3.3 开发环境及工具

本系统客户端和服务端均采用 JAVA 语言开发,使用 IntelliJ IDEA 作为开发工具。

### 2.3.4 技术限制

为了保证数据的安全性以及系统的可移植性,本系统采用三层 c/s 架构设计开发,客户端与服务器端采用 HTTP 交换数据,因此为了获得较低的操作延迟,对带宽有一定的要求,我们建议将服务器架设在公司内部服务器上,这样即可以保证操作延迟低,又能避免公网带来的数据安全风险。

本系统客户端采用 javafx 编写,尽管 javafx 是一套优秀并且持续维护的桌面软件解决方案,但是众所周知桌面软件行业已经几乎属于夕阳产业,仅凭桌面游戏在苦苦支撑着,因此 javafx 的表现力与大家经常使用的 web 前端界面还有所差距,尽管我们已经在界面设计上下足了功夫,但是显然无法与深耕多年的 web 界面相比,尤其在界面的美观和灵活性上。

### 2.3.5 其他

本次项目周期只有短短的两周时间,因此在项目设计中,必须小心谨慎,不能出现设计失误,因为设计失误会导致开发付出很多无用的劳动,会严重影响项目进度。

尽管这样,为了保证项目按时保质保量完成,我们还是安排了更多的人手加入开发。

### 3. 逻辑架构设计

#### 3.1 设计决策

HLW 剧院票务管理系统软件逻辑架构图如图 3-1 所示，系统采用三层 C/S 架构设计 ,数据首先从界面从产生 ,通过函数参数的形式传递给客户端的转发层，转发层将数据使用 HTTP 协议转发给服务器 ,服务器端的控制器层负责接受来自转发层的数据，并且将对应的请求传递给业务层，业务层根据需求从持久化层获取或者修改、写入数据，而持久化层

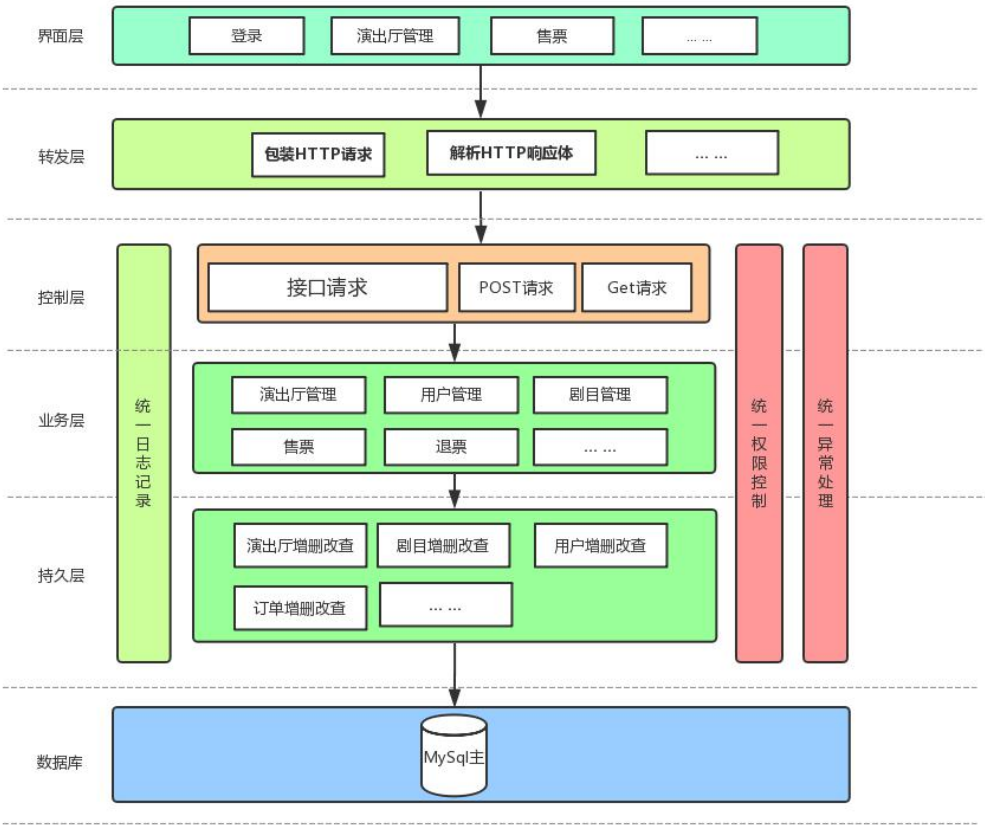


图 3-1 HLW 剧院票务管理系统软件逻辑架构

#### 3.2 软件单元

3.2.1 界面层

界面层的软件单元构成如图 3-2 所示，软件单元的说明见表 4。

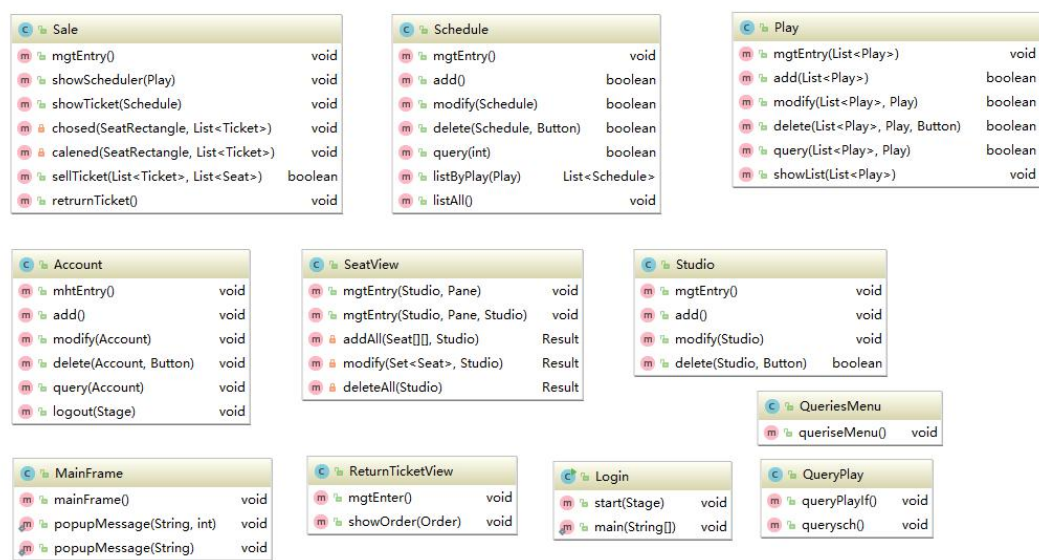


图 3-2 界面层类图

表 4 界面层软件单元构成

序号	软件单元标识符	软件单元(类)名称	功能说明	备注
1.	SU_UI_01	Account	用户管理	
2.	SU_UI_02	Login	登录	
3.	SU_UI_03	MainFrame	界面主框架	
4.	SU_UI_04	Play	剧目管理	
5.	SU_UI_05	ReturnTicketView	退票	
6.	SU_UI_06	Sale	售票	
7.	SU_UI_07	Schedule	演出计划管理	
8.	SU_UI_08	SeatView	座位管理	
9.	SU_UI_09	Studio	演出厅管理	



3.2.2 业务逻辑层

业务逻辑层的软件单元构成如图 3-3 所示，软件单元的说明见表 5。

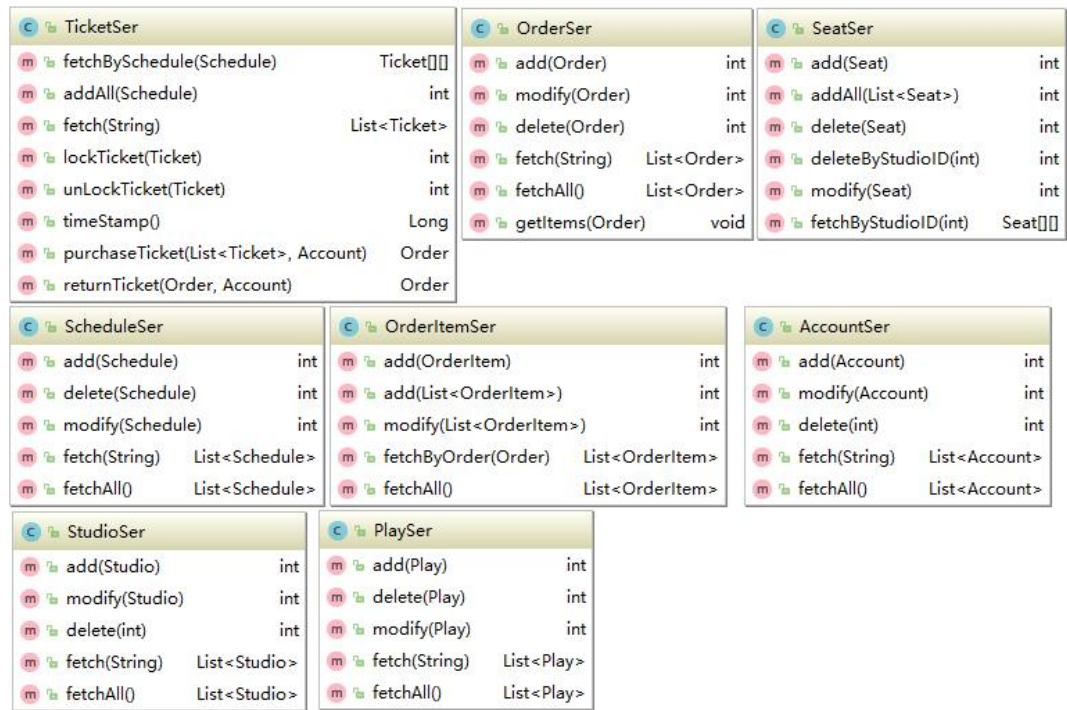


图 3-3 业务逻辑层类图

表 5 界面层软件单元构成

序号	软件单元标识符	软件单元(类)名称	功能说明	备注
1	SU_Ser_01	AccountSer	用户管理	
2	SU_Ser_02	OrderItemSer	订单条目处理	
3	SU_Ser_03	OrderSer	订单处理	
4	SU_Ser_04	PlaySer	剧目管理	
5	SU_Ser_05	ScheduleSer	演出计划管理	
7	SU_Ser_06	SeatSer	座位管理	
8	SU_Ser_09	StudioSer	演出厅管理	
9	SU_Ser_10	TicketSer	票管理	

3.2.3 持久化层

业务逻辑层的软件单元构成如图 3-4 所示，软件单元的说明见表 6。

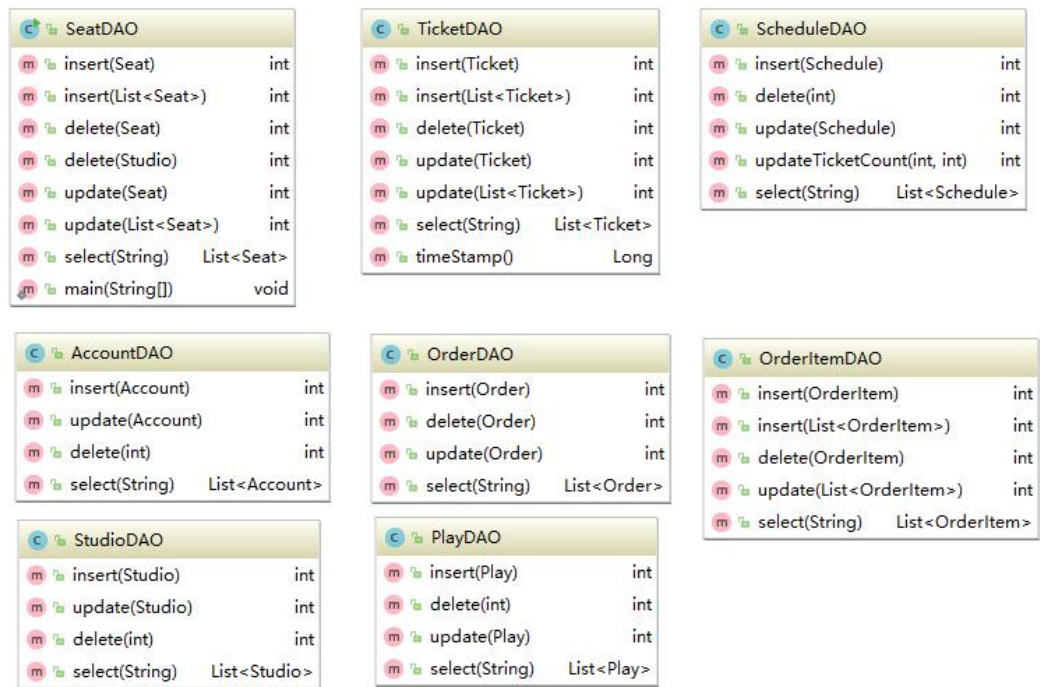


图 3-4 持久化层类图

表 6 持久化层软件单元构成

序号	软件单元标识符	软件单元(类)名称	功能说明	备注
1	SU_DAO_01	AccountDAO	用户管理	
2	SU_DAO_02	OrderDAO	订单处理	
3	SU_DAO_03	OrderItemDAO	订单条目处理	
4	SU_DAO_04	PlayDAO	剧目管理	
5	SU_DAO_05	ScheduleDAO	演出计划管理	
6	SU_DAO_06	SeatDAO	座位管理	
7	SU_DAO_07	StudioDAO	演出厅管理	
8	SU_DAO_08	TicketDAO	票管理	

### 3.3 处理流程

#### 3.3.1 管理剧目(TTMS\_UC\_01\_1)

管理剧目用例的处理流程图如图 3-5 所示，管理员用户在成功登录后，可以对剧目信息进行新增修改删除 ,对应的操作会引起数据库中持久化的数据发生改变。

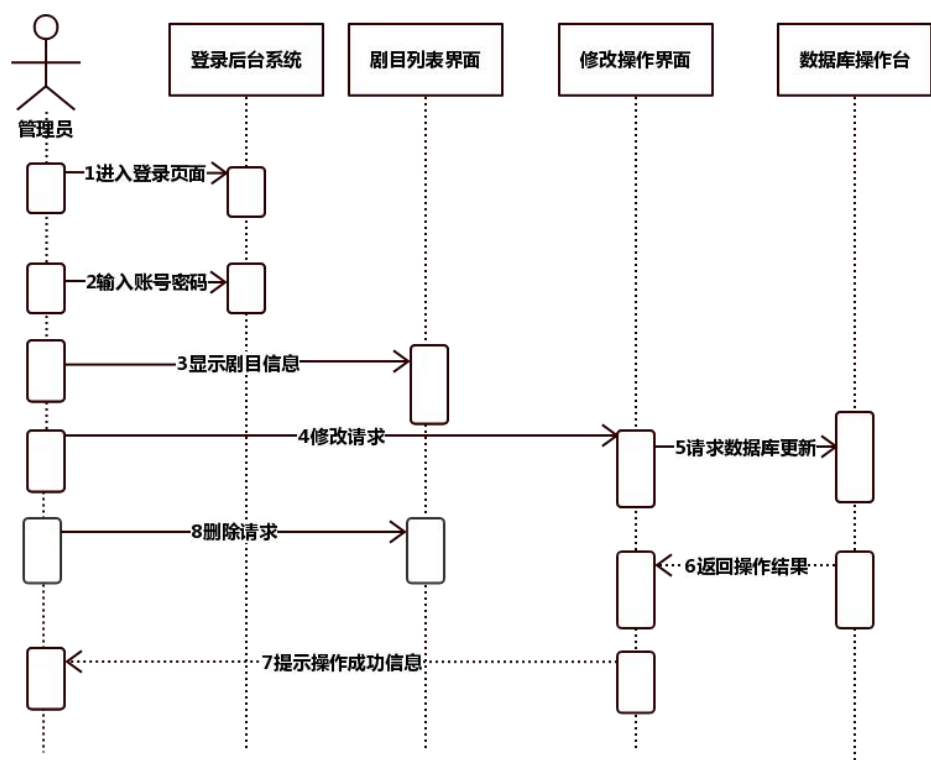


图 3-5 管理剧目用例处理流程

#### 3.3.2 管理演出厅(TTMS\_UC\_02\_2)

管理演出厅用例的处理流程图如图 3-6 所示，管理员用户在成功登录后，可以对演出厅信息进行新增修改删除 ,对应的操作会引起数据库中持久化的数据发生改变。

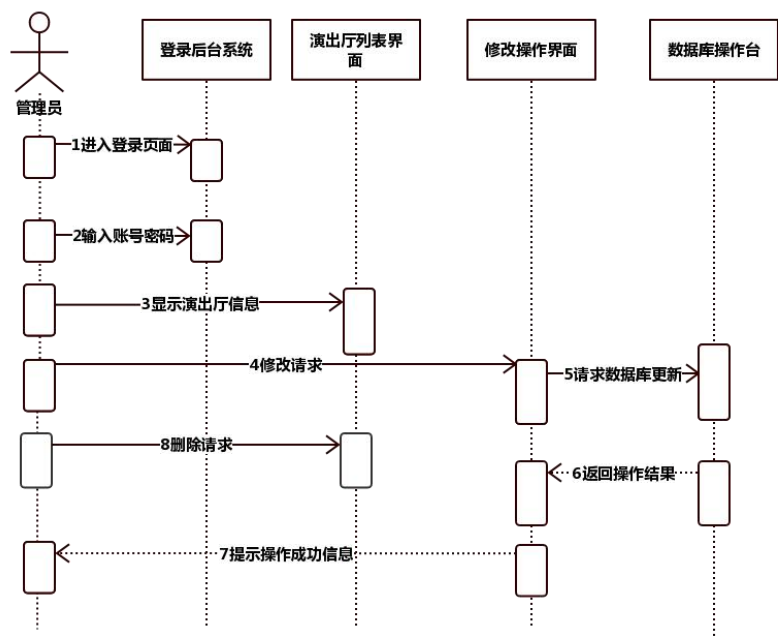


图 3-6 管理演出厅用例处理流程

3.3.3 管理用户(TTMS\_UC\_02\_3)

用户管理用例处理流程图如图 3-7，管理员用户在成功登录后，可以对用户的信息进行新增修改删除，对应的操作会引起数据库中持久化的数据发生改变。

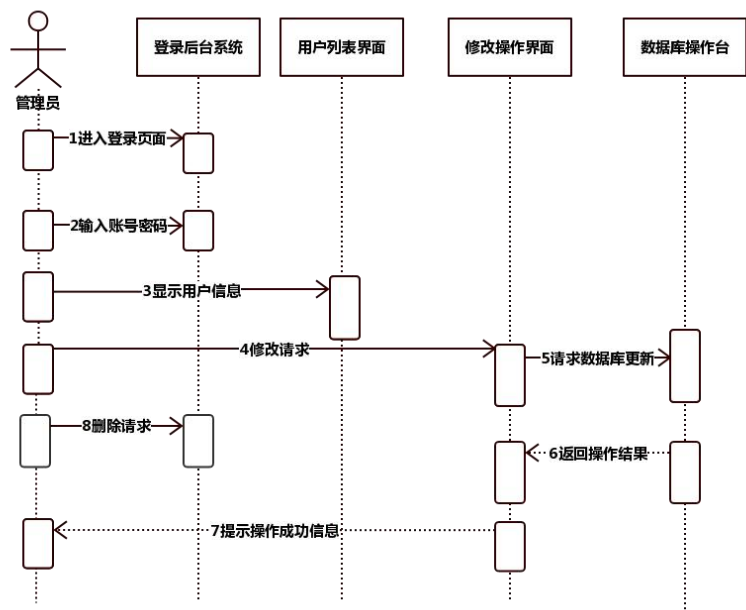
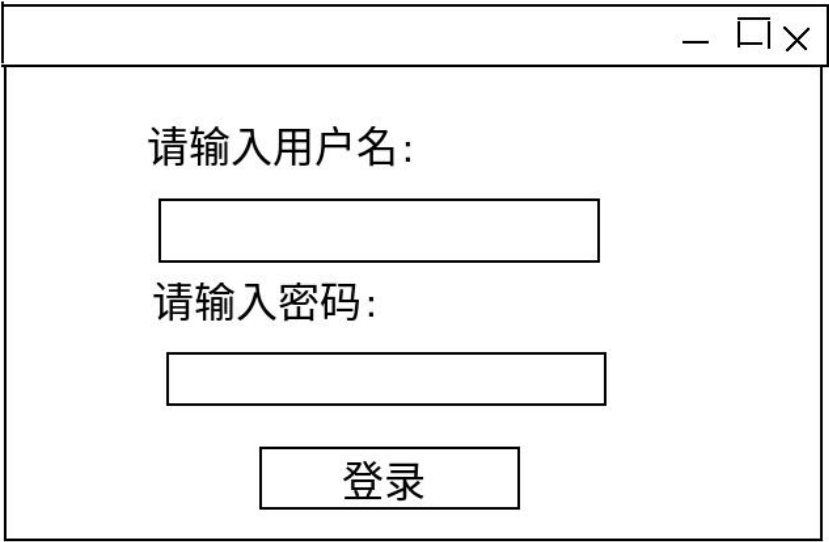


图 3-7 管理用户用例处理流程

## 4. 人机界面设计

### 4.1 初步方案(草图)

程序开始运行首先弹出登录界面，草图如图 4-1



草图 4-1 展示了登录界面的设计。该界面包含以下元素：

- 窗口标题栏：位于顶部，包含最小化、最大化和关闭按钮。
- 用户名输入框：上方标有“请输入用户名：”。
- 密码输入框：下方标有“请输入密码：”。
- 登录按钮：位于底部中央，标有“登录”。

图 4-1 登陆界面草图

登录成功后转入主界面 左侧显示功能列表，草图如图 4-2



草图 4-2 展示了主界面的设计。该界面包含以下元素：

- 窗口标题栏：位于顶部，包含最小化、最大化和关闭按钮。
- 功能列表：位于左侧，包含以下功能项：
  - 用户管理
  - 演出厅管理
  - 座位管理
  - 售票
  - 退票
  - 统计
- 主内容区：位于右侧，用于显示功能操作的结果。

图 4-2 主界面草图

当用户选择具体功能后 比如选择 演出厅管理 则图 4-3



图 4-3 具体功能界面草图

界面上方出现子菜单 可以选择子功能 当用户选择子功能后 界面主体部分

将显示相应功能 例如删除演出厅界面草图如图 4-4

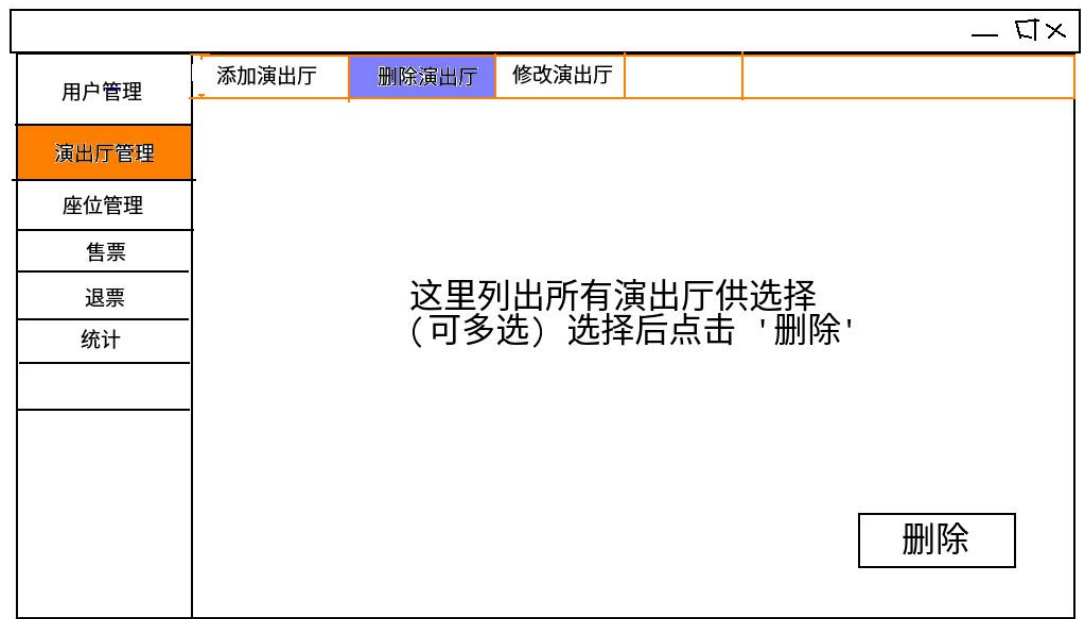


图 4-4 删除演出厅界面草图

整套程序按照这套流程设计 将其他功能入口放在左栏菜单或者顶栏菜单中,而功能的主体部分展示在界面中间,在功能完整的前提下由负责人自由设计

## 4.2 最终效果

### 4.2.1 登录

登录界面如图 4-5 所示,用户输入用户名密码之后点击登录按钮即可登录。



图 4-5 登录界面

### 4.2.2 演出厅管理

演出厅管理界面示意图如图 4-6,在演出厅管理功能中,管理员可以对演出厅进行增删改查操作,系统首先列出所有演出厅,管理员选择需要操作的演出厅,对其进行修改或者删除操作,也可以点击上方的“添加演出厅”按钮来新增一个演出厅。





图 4-6 演出厅管理

4.2.3 座位管理

座位管理界面如图 4-7 所示，在新增/修改演出厅的同时,会对演出厅的座位进行调整，在修改座位的界面，管理员可以点击一个座位，使其在正常，损坏，无座位三种状态之间进行切换，随后将保存提交到服务器。

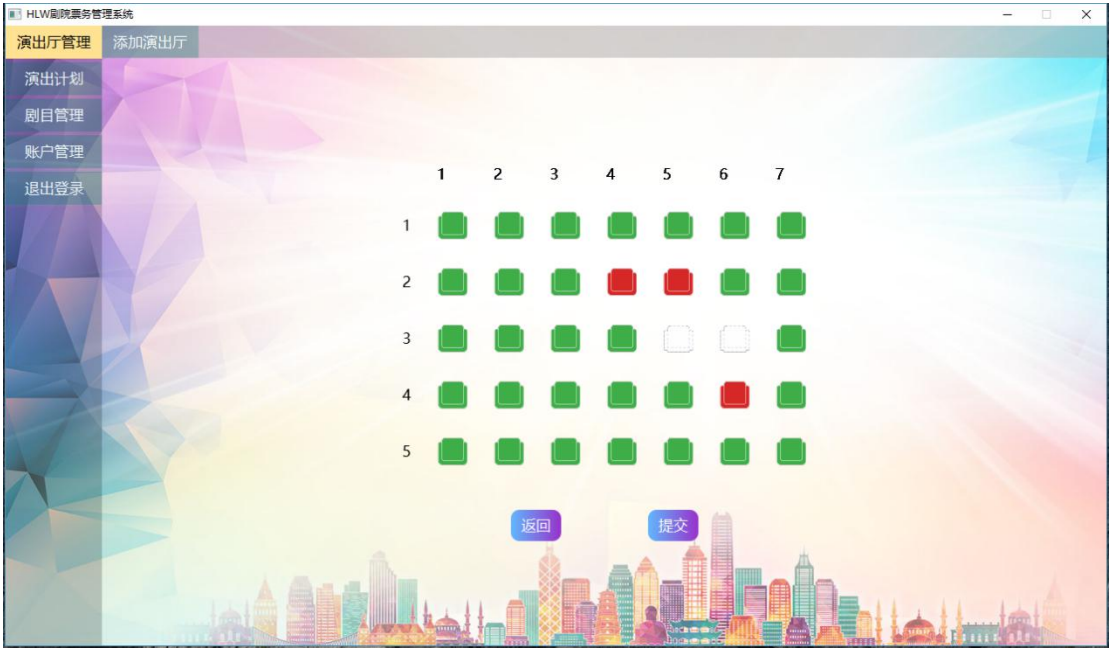


图 4-7 座位管理



4.2.4 用户管理

用户管理界面如图 4-8 所示。



图 4-8 用户管理

4.2.5 新增演出计划

新增演出计划界面如图 4-9 所示，新增演出计划功能使用日期选择器和时间选择器让经理提供日期和时间,避免手动输入产生的错误。



图 4-9 新增演出计划

4.2.6 剧目管理

剧目管理界面如图 4-10 所示。

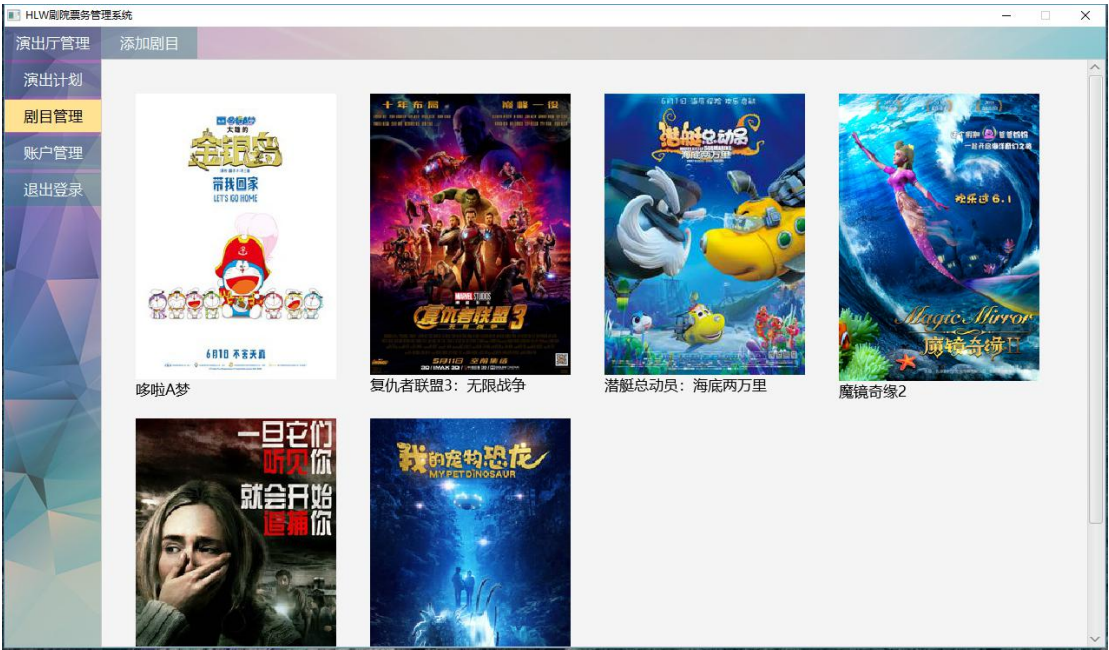


图 4-10 剧目管理

4.2.7 售票

售票界面如图 4-11 所示，售票员每点击一个座位时,就会锁定当前座位,其

他售票员就暂时无法点击这个座位。

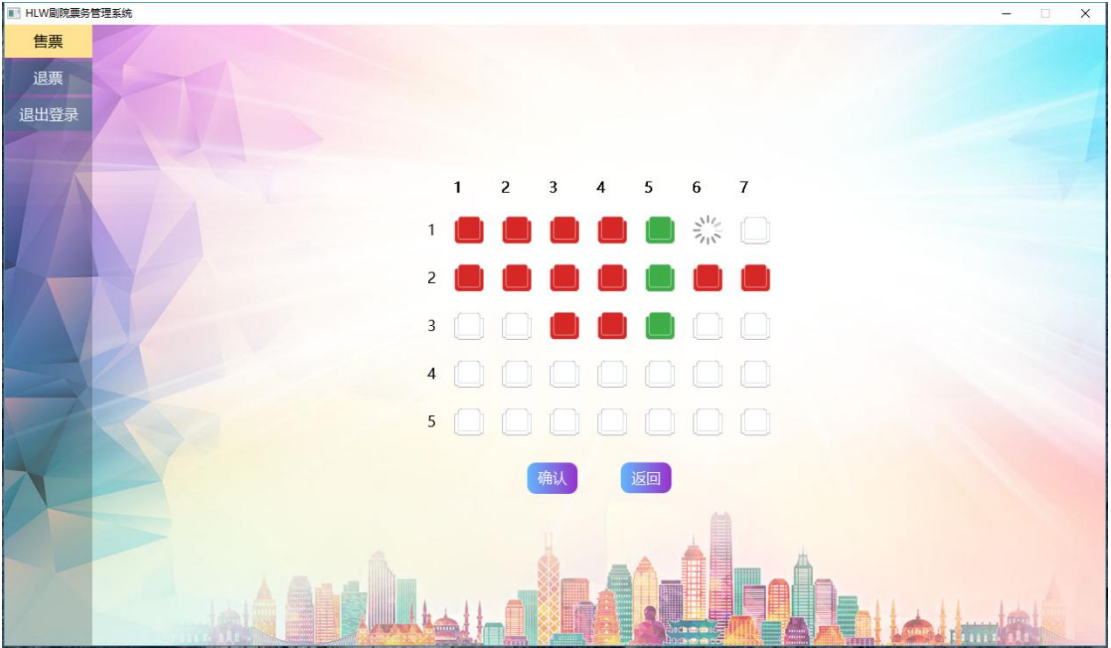


图 4-11 售票

售票成功界面如图 4-12 所示，售票成功后，订单 id 可以作为凭证，只需将此弹窗内容打印出来，加上一定的防伪措施，即可产生实体票。

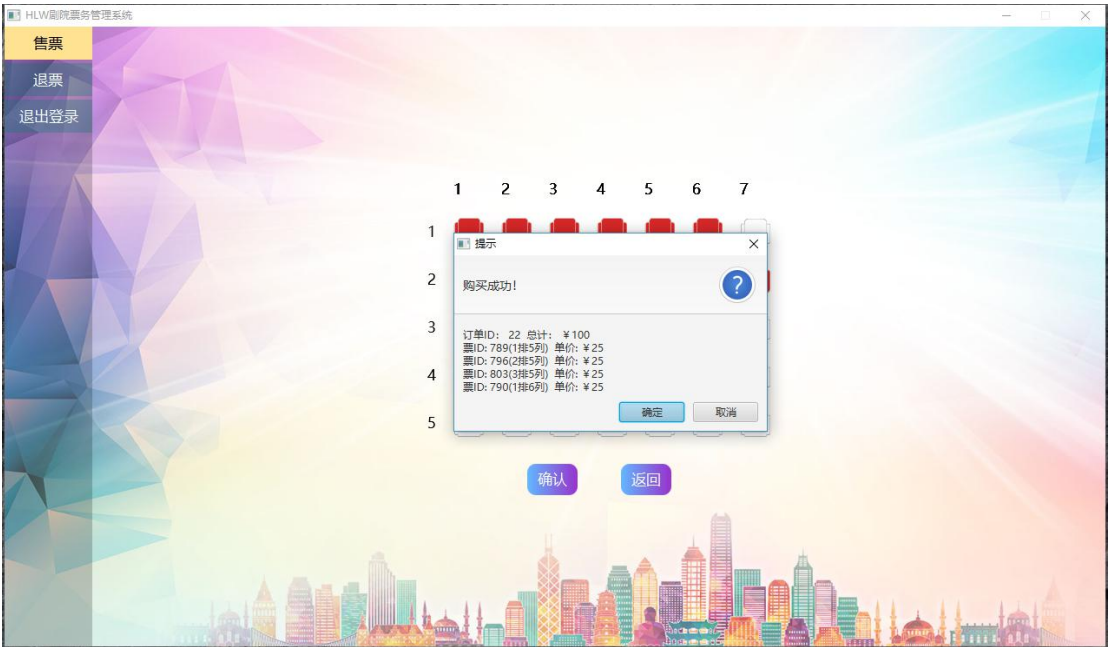


图 4-12 售票成功

4.2.8 退票

退票界面如图 4-13 所示，售票员根据用户提供的票上的订单 id，先对订单进行查询,若是合法订单 id,则会显示订单信息并且显示可用的退票按钮。



图 4-13 退票

退票成功界面如图 4-14 所示，退票成功后,订单状态已经改变,并且退票按钮不再可用。



图 4-14 退票成功



4.2.9 异常输入判断以及消息提示

像图 4-15 这样的输入场景,用户有可能键入不合法的,例如在价格输入框中输入一个字符,或者在封面输入框中输入一个不合法的 URL,这都会对程序运行造成影响,因此系统在,每一处需要用户手动输入的地方加上的异常输入验证,以保证程序不会因为操作失误而终止运行。

还有注意到图上的提示信息,我们为系统做了一个统一的消息提醒框架,在任何一个界面调用,都可以展示出如图上的消息提醒,并且消息提醒还有丰富的动画效果,给整个程序增添了不少生气。



图 4-15 异常输入判断以及消息提示

## 5. 数据存储设计

### 5.1 内部数据结构

TTMS 系统内部数据示意图(类图)如图 5-1 所示。

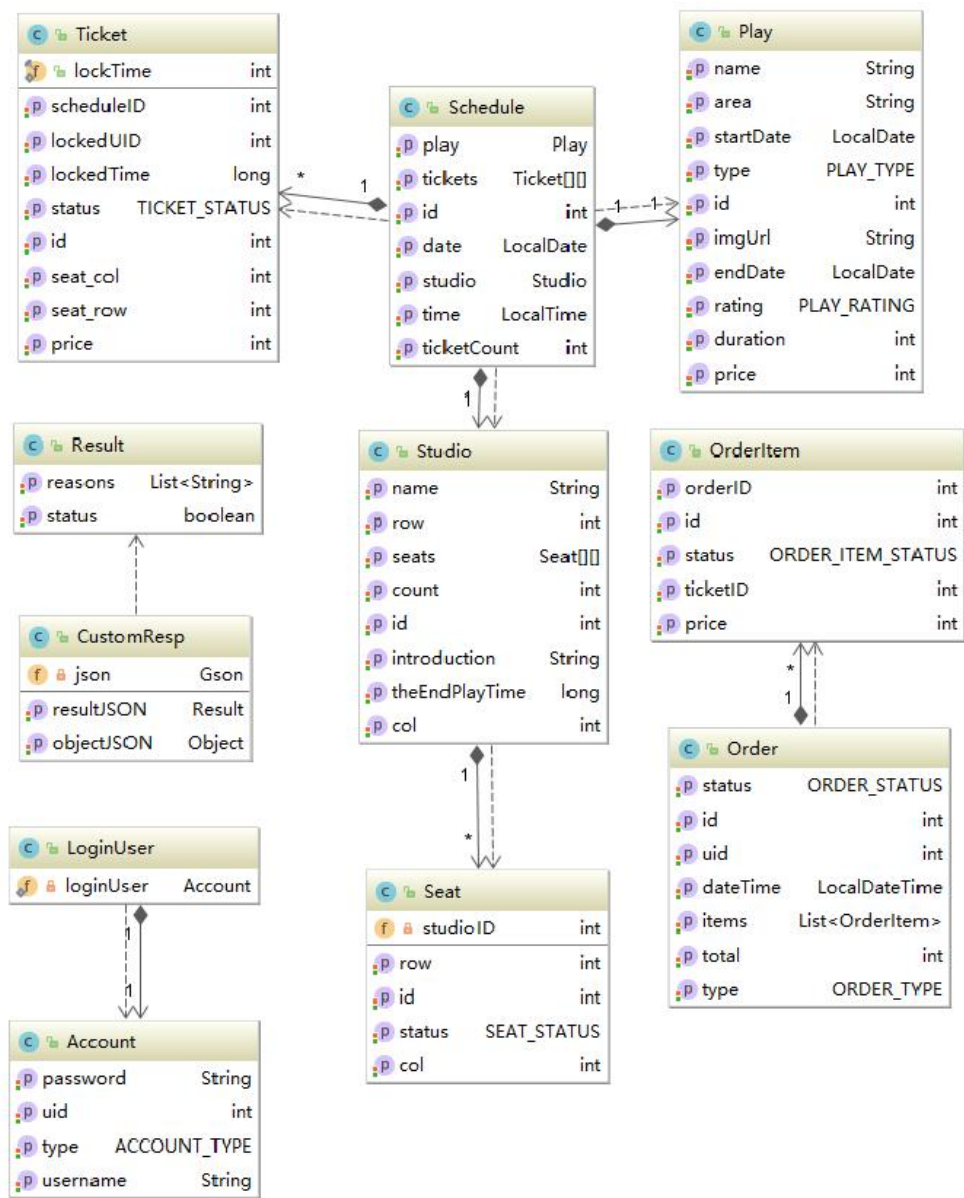


图 5-1 内部数据示意图(类图)

### 5.2 数据库

数据库表关系以及数据表定义图如图 5-2 所示。

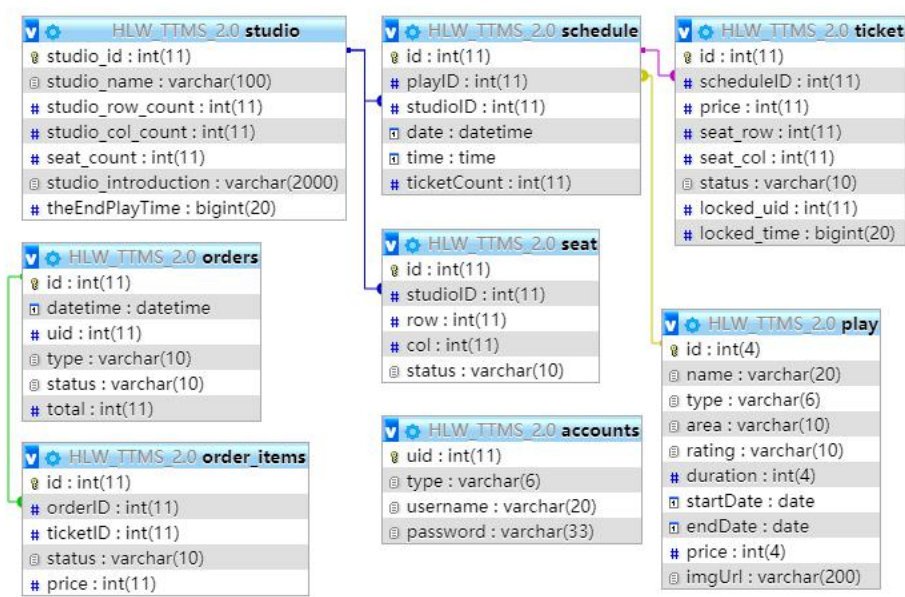


图 5-2 数据库表关系以及数据表定义图

5.3 数据文件

无

## 6. 详细设计

### 6.1 剧目管理(TTMS\_UC\_01\_1)

#### 6.1.1 功能描述

剧目管理类的类图如图 6-1 所示。剧目管理类通过对 Play 对象进行操作来对剧目进行管理。

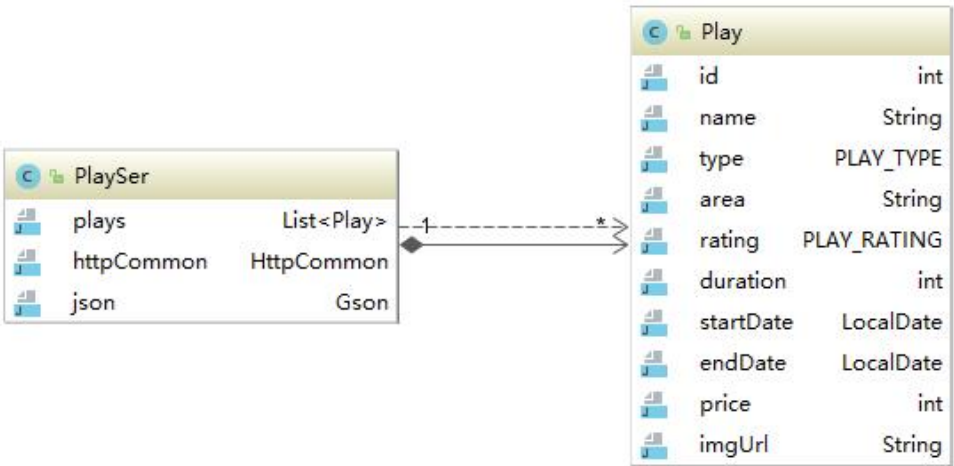


图 6-1 剧目管理类类图

#### 6.1.2 处理流程

管理剧目用例的处理流程图如图 3-5 所示，管理员用户在成功登录后，可以对剧目信息进行新增修改删除，对应的操作会引起数据库中持久化的数据发生改变。

#### 6.1.3 内部数据

剧目管理类的属性及其作用见表 6-1。

表 6-1 剧目管理类属性列表



属性名	用途	取值范围
plays	获取到的剧目列表	-
httpCommon	封装的 http 连接对象	-
json	Json 处理对象	-

#### 6.1.4 异常与错误处理

对于剧目列表为空的情况要进行判断，防止出现空指针异常。

#### 6.1.5 测试要点

添加、修改剧目时涉及到异常输入的地方，尝试边界数据进行测试。

### 6.2 演出厅管理(TTMS\_UC\_02\_1)

#### 6.2.1 功能描述

演出厅管理类以及座位类类图如图 6-2 所示。演出厅管理除了涉及演出厅的基本信息的修改，还涉及到演出厅中座位的管理。

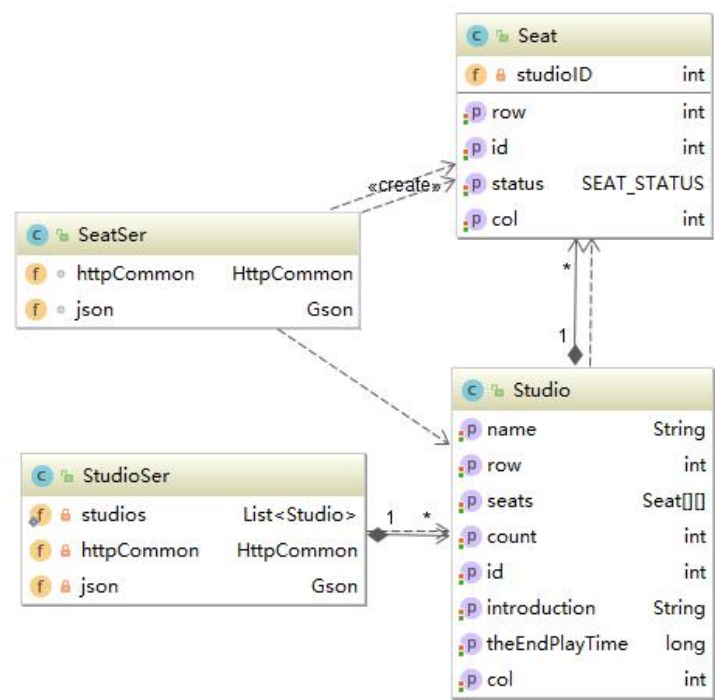


图 6-2 演出厅管理类和座位管理类类图

6.2.2 处理流程

管理演出厅用例的处理流程图如图 3-6 所示，管理员用户在成功登录后，可以对演出厅信息进行新增修改删除，对应的操作会引起数据库中持久化的数据发生改变。

6.2.3 内部数据

剧目管理类的属性及其作用见表 6-2。

表 6-2 剧目管理类属性列表

属性名	用途	取值范围
studios	获取到的演出厅列表	-
httpCommon	封装的 http 连接对象	-
json	Json 处理对象	-

6.2.4 异常与错误处理

在修改座位时，有可能删除原座位失败，但是新的座位已经列出了，就会导致座位列表出现致命的异常。

6.2.5 测试要点

反复的测试修改座位功能，看是否会产生内存泄露这类问题，以及性能瓶颈问题。

6.3 用户管理(TTMS\_UC\_02\_2)

6.3.1 功能描述

用户管理类的类图如图 6-3 所示，用户管理类通过对 Account 对象进行操作来实现用户管理。

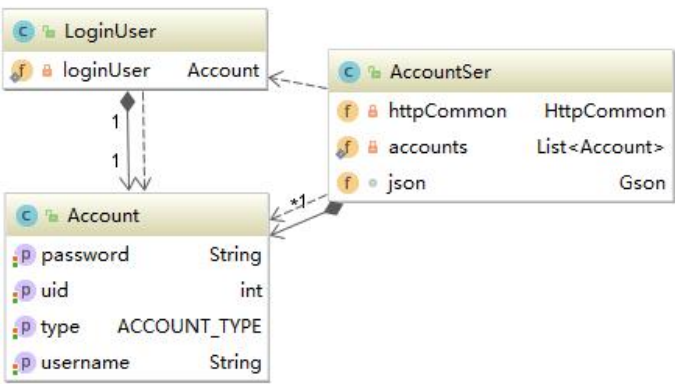


图 6-3 用户管理类类图

6.3.2 处理流程

用户管理用例处理流程图如图 3-7，管理员用户在成功登录后，可以对用户的信息进行新增修改删除，对应的操作会引起数据库中持久化的数据发生改变。

### 6.3.3 内部数据

用户管理类属性一览表见表 6-3。

表 6-3 用户管理类属性列表

属性名	用途	取值范围
accounts	获取到的演出厅列表	-
httpCommon	封装的 http 连接对象	-
json	Json 处理对象	-
loginUser	保存已登录用户	

### 6.3.4 异常与错误处理

用户登录时密码输入错误次数过多应该给予一定的保护措施,防止用户密码被暴力破解。

### 6.3.5 测试要点

用户管理模块是整个系统的生命线,一旦用户系统出现问题,轻则导致业务无法正常办理,重则导致企业数据泄露,数据损坏,因此测试过程中应该着重测试用户管理模块的安全性能。

## 7. 开发架构设计

### 7.1 工程结构

项目客户端目录结构如图 7-1 所示。



图 7-1 项目客户端目录结构图

项目服务器端目录结构如图 7-2 所示。

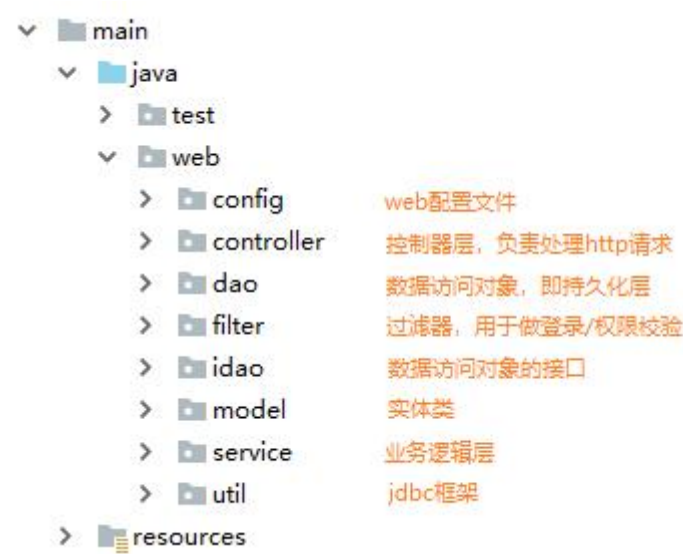


图 7-2 服务器端目录结构

## 8. 物理架构设计

### 8.1 网络环境

系统的网络拓扑图如图 8-1 所示。

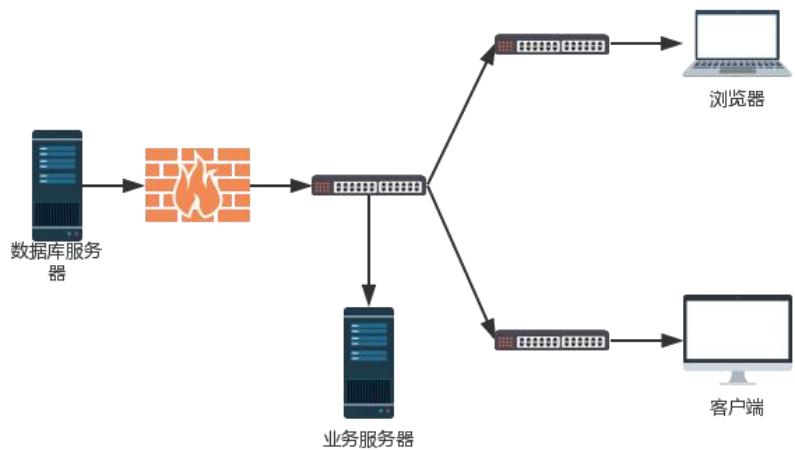


图 8-1 系统网络拓扑图

### 8.2 部署方案

软件部署图如图 8-2 所示。

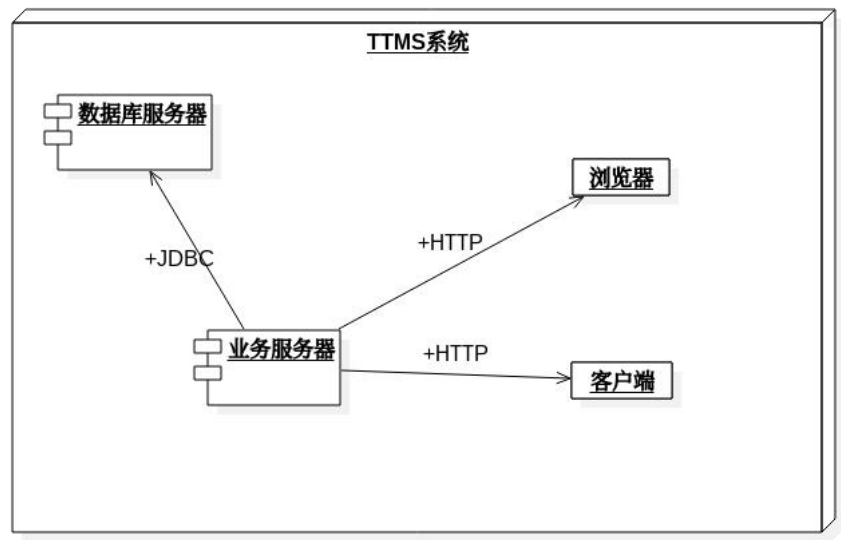


图 8-2 软件部署图