

项目编号	XUPTSE.2019.TTMS
文档编号	XUPTSE.2019.TTMS.SDS
密 级	高级保密



西安邮电大学

# 金太阳影院管理系统 软件设计说明

版本：V1.0

## 团队成员：

学号	姓名	角色
04173113	伍亚荣	组员
04173114	王静茹	组员
04173115	刘娅	组长
04173097	杨朋飞	组员

软件 1704 班金太阳软件公司

二〇一九年五月

拟 制：刘娅

审 核：王静茹

标准化：伍亚荣

会 签：杨朋飞，伍亚荣，王静茹，刘娅

批 准：舒新峰



## 目 录

目 录.....	I
1. 引言.....	4
1.1 编写目的.....	4
1.2 项目概述.....	4
1.3 术语定义.....	4
1.4 缩写说明.....	5
1.5 引用文档.....	5
2. 软件设计决策.....	6
2.1 设计目标.....	6
2.2 设计原则.....	6
2.3 设计约束.....	6
2.3.1 遵循标准.....	7
2.3.2 运行环境.....	7
2.3.3 开发环境及工具.....	7
2.3.4 技术限制.....	8
2.3.5 其他.....	8
3. 逻辑架构设计.....	8
3.1 设计决策.....	8
3.2 软件单元.....	9
3.2.1 界面层.....	9
3.2.2 业务逻辑层.....	11
3.2.3 持久化层.....	13
3.3 处理流程.....	13
3.3.1 订单查询 (TTMS_UC_03).....	13
4. 人机界面设计.....	14
5. 数据存储设计.....	15
5.1 内部数据结构.....	15
5.2 数据库.....	15
5.2.1 数据库设计.....	15
5.2.2 数据表定义.....	15
6. 详细设计.....	19
6.1 MovieController (Service_UC_01).....	19

6.1.1 功能描述.....	19
6.1.2 处理流程.....	19
6.1.3 内部数据.....	19
6.1.4 测试要点.....	19
6.2 MovieTypeController（Service_UC_02） .....	19
6.2.1 功能描述.....	19
6.2.2 处理流程.....	20
6.2.3 内部数据.....	20
6.3 PicUpLoadController（Service_UC_03） .....	20
6.3.1 功能描述.....	20
6.3.2 处理流程.....	20
6.3.3 内部数据.....	20
6.3.4 异常与错误处理.....	20
6.3.5 测试要点.....	20
6.4 ScheduleController（Service_UC_04） .....	20
6.4.1 功能描述.....	21
6.4.2 处理流程.....	21
6.4.3 内部数据.....	21
6.4.4 异常与错误处理.....	21
6.4.5 测试要点.....	21
6.5 SeatController（Service_UC_05） .....	21
6.5.1 功能描述.....	21
6.5.2 处理流程.....	22
6.5.3 内部数据.....	22
6.5.4 异常与错误处理.....	22
6.5.5 测试要点.....	22
6.6 StudioController（Service_UC_06） .....	22
6.6.1 功能描述.....	22
6.6.2 处理流程.....	22
6.6.3 内部数据.....	22
6.6.4 测试要点.....	23
6.7 TicketController（Service_UC_07） .....	23
6.7.1 功能描述.....	23
6.7.2 处理流程.....	23

---

6.7.3 内部数据.....	23
6.7.4 测试要点.....	23
7. 开发架构设计.....	23
7.1 工程结构.....	23
7.2 源代码文件.....	24
8. 物理架构设计.....	25
8.1 网络环境.....	25
8.2 部署方案.....	25

# “金太阳”软件设计说明

## 1. 引言

### 1.1 编写目的

本文档是关于用户对“剧院票务管理系统”的功能与非功能需求的详细说明。本文档的编写为“剧院票务管理系统”的设计、开发和测试提供依据，为项目组成员对需求的详尽理解，以及在开发开发过程中的协同工作提供强有力的保证。同时本文档也作为项目评审验收的依据之一。

本文的预期读者包括：

- 用户
- 系统开发人员
- 系统测试人员
- 系统验收人员

### 1.2 项目概述

- 项目名称：剧院票务管理系统；
- 项目编号：XUPTSE.2016.TTMS；
- 投资方：亿达传媒有限公司；
- 用户：亿达传媒有限公司下属个剧院；
- 开发方：金太阳公司。

### 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	影厅	指影院现有可以满足电影放映的影厅
2	排场	安排现有正在上映的电影在什么地方什么时间放映
3	选座	观众可以根据已经选择的场次来选择自己的座位

## 1.4 缩写说明

本文件中用到的英文缩写说明见表 2。

表 2 英文缩写说明

序号	缩写	原文
1	TTMS	剧院票务管理系统
2	UC	用例
3	BD	标书
4	DD	设计文档

## 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/T 11457-2006	信息技术 软件工程术语		2006/7/1	国务院标准 行政部门
2	GB/T 14394-2008	计算机软件可靠性和可维护性管理		2008/12/1	国务院标准 行政部门
3	GB/T 9386-2008	计算机软件测试文件编制规范		2008/9/1	国务院标准 行政部门
4	GB/T 9385-2008	计算机软件需求规格说明规范		2008/9/1	国务院标准 行政部门
5	GB/T 9385-2008	计算机软件测试规范		2008/9/1	国务院标准 行政部门
6	XUPTSE.2016.TTMS.SRS	票淘淘软件需求规格说明书		2019/4/29	西安邮电 大学软件 工程系



## 2. 软件设计决策

### 2.1 设计目标

为了使系统在影院的管理中发挥更大的作用，实现工作过程的计算机化，提高工作效率和工作质量，现提出如下的系统开发目标：

- 1.系统应具有实用性、可靠性和适用性，同时注意到先进性。
- 2.对各个数据库进行动态管理，防止混乱。
- 3.能够按照用户选择的不同的条件进行简单查询和复合查询。
- 4.能够对查询结果进行分类汇总，实现报表打印。
- 5.注意数据的安全性，具有数据备份和恢复的功能。
- 6.方便用户的操作，尽量减少用户的操作。

### 2.2 设计原则

（1）以技术先进和标准化为设计标准，设计开发遵循现有的国家设计规范软件。

（2）以功能实用性为主导思想，开发实现整个规范建设需要展示的内容，满足展示和管理的实用性。

（3）开放性和可扩展性，选择具有较大兼容能力和扩展能力强的 JAVA 技术体系来开发，满足将来不断提升的信息系统建设需求。

（4）以安全性为设计要求，系统设计充分考虑"安全性"。因系统投入后将长期使用，必须保证系统长期正常稳定的运行。采用有效的技术实现方式提高安全系数。

（5）界面应该简洁、有效、显而易见，不要在功能的可用性上妥协保持界面简洁直观，用户需要易于理解和操作的功能。

（6）建立在用户已有的经验上允许用户根据以往的经验操作系统，特别是他们从现实世界中获得的经验。支持不同的交互技术,允许用户选择最适合自身情况的交互方式。

### 2.3 设计约束

### 2.3.1 遵循标准

### 2.3.2 运行环境

#### 1) 硬件配置

数据库服务器：

CPU：Interi5 以上

内存：2G 以上(建议 2G)

硬盘：20G

光驱：普通

打印机：普通喷墨打印机（可选）

网卡：100M

终端：

CPU：Interi3 以上

内存：512M 以上

硬盘：10G

显示器：

屏幕分辨率：800\*600 以上(推荐 1920\*1080)

打印机：普通喷墨打印机（单张打印），专业票务打印机（连打）

网卡：100M

网络设备：100M 集线器或交换机

#### 2) 软件配置

服务器：

操作系统：WINDOWS SERVER2010

数据库：SQLSERVER2017

终端：WINDOWS XP、WINDOWS 7、WINDOWS 10

### 2.3.3 开发环境及工具

后台：

开发语言：Java

开发环境：Windows 10

开发工具：eclipse

前端：

开发语言：HTML、CSS、JavaScript、Ajax

开发环境：node

### 2.3.4 技术限制

存储容量：web 端，储存容量要求不高。

性能：响应时间短、可拓展性强

灵活性：有完整的业务模型，灵活性较高

### 2.3.5 其他

## 3. 逻辑架构设计

### 3.1 设计决策

使用面向对象技术进行系统的分析与设计， 并使用 UML 描述系统的设计模型；

使用 Java 语言进行软件开发；

逻辑架构采用分层体系结构；

物理架构采用 C/S 架构，服务器端为数据库服务器；

数据采用集中式存储，数据库使用关系数据库 MySQL5。

逻辑架构图 1 所示：

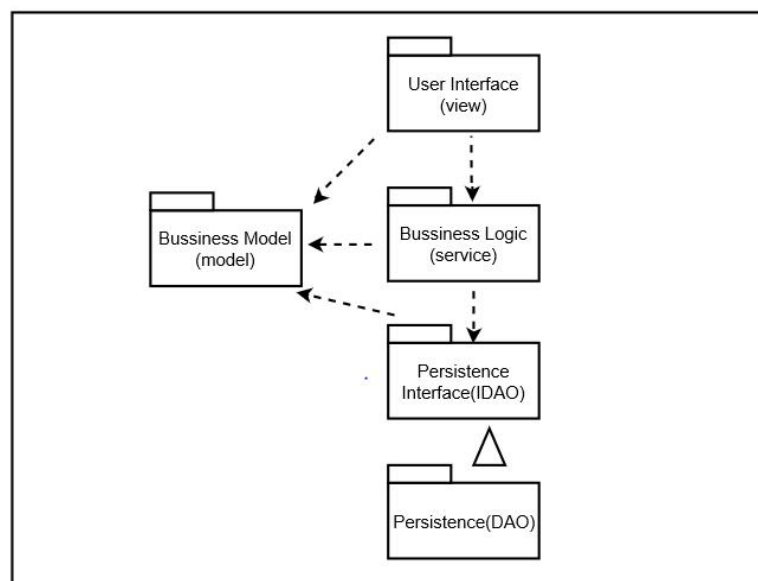


图 1 金太阳软件逻辑架构

## 3.2 软件单元

### 3.2.1 界面层

根据需求规格说明书，软件分成两块用户和管理员模块，管理员界面如图 2 所示。

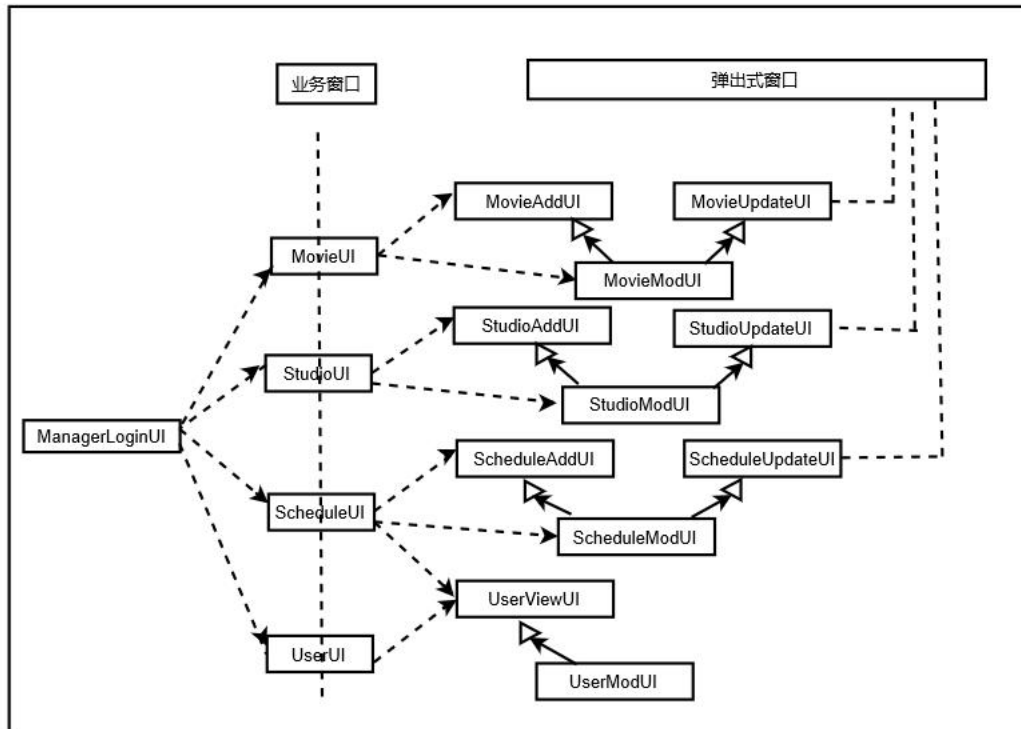


图 2 管理界面层

下表给出了管理员界面层软件单元构成：

表 4 管理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.	Manager_UI_01	ManagerLoginUI	管理端登录界面	
2.	Manager_UI_02	MovieUI	电影管理界面	
3	Manager_UI_03	StudioUI	影厅管理界面	
4	Manager_UI_04	ScheduleUI	演出计划管理界面	
5	Manager_UI_05	UserUI	用户管理界面	
6	Manager_UI_06	MovieAddUI	电影增加界面	
7	Manager_UI_07	MovieUpdateUI	电影修改界面	

8	Manager_UI_08	MovieModUI	电影模态框	
9	Manager_UI_09	StudioAddUI	影厅增加界面	
10	Manager_UI_10	StudioUpdateUI	影厅修改界面	
11	Manager_UI_11	StudioModUI	影厅模态框	
12	Manager_UI_12	ScheduleAddUI	演出计划增加界面	
13	Manager_UI_13	ScheduleUpdateUI	演出计划修改界面	
14	Manager_UI_14	ScheduleModUI	演出计划模态框	
15	Manager_UI_15	UserUI	用户管理界面	
16	Manager_UI_16	UserViewUI	用户查看界面	
17	Manager_UI_17	UserModUI	用户模态框	

用户界面层如图 3 所示：

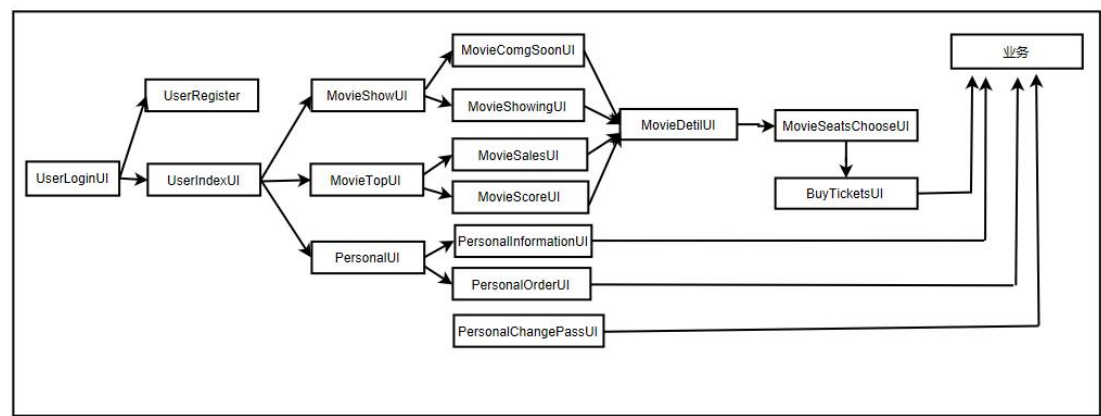


图 3 用户界面层

下表给出了用户界面层软件单元构成：

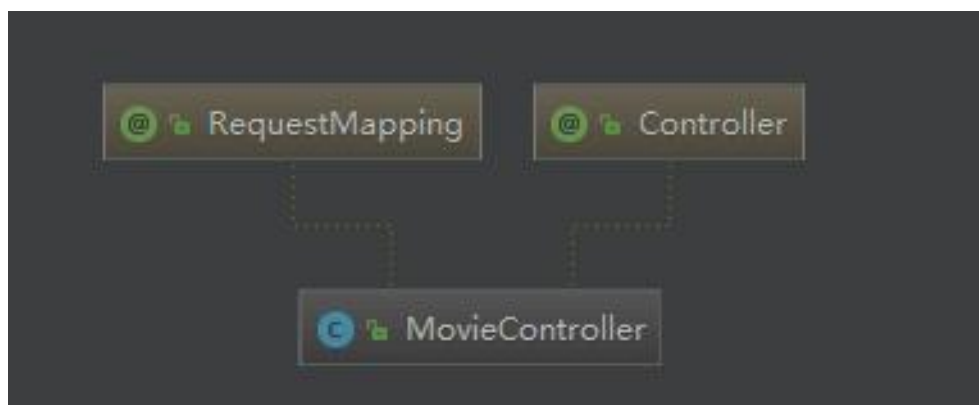
表 5 用户界面层软件单元构成

序号	软件单元标识符	软件单元(类)名称	功能说明	备注
1	User_UI_01	UserLoginUI	用户登录界面	
2	User_UI_02	UserRegister	用户注册界面	
3	User_UI_03	StudioUI	电影展示界面	

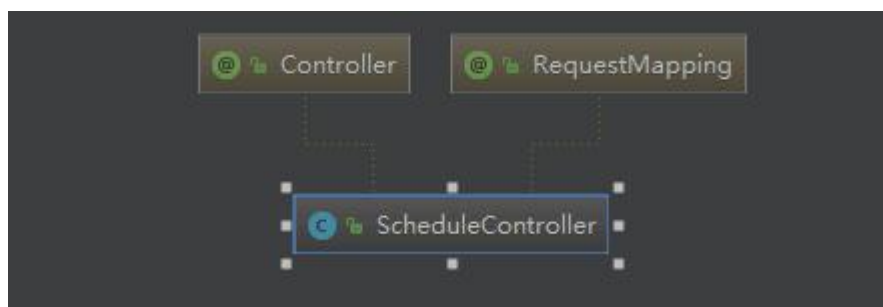
4	User_UI_04	MovieShowUI	电影榜单界面	
5	User_UI_05	MovieTopUI	个人中心界面	
6	User_UI_06	PersonalUI	即将上映界面	
7	User_UI_07	MovieComgSoonUI	正在热映界面	
8	User_UI_08	MovieShowingUI	销量榜单界面	
9	User_UI_09	MovieSalesUI	评分榜单界面	
10	User_UI_10	MovieScoreUI	个人信息界面	
11	User_UI_11	PersonalInformationUI	个人订单界面	
12	User_UI_12	PersonalOrderUI	修改密码界面	
13	User_UI_13	PersonalChangePassUI	电影详情界面	
14	User_UI_14	MovieDetilUI	选座界面	
15	User_UI_15	MovieSeatsChooseUI	购票界面	

### 3.2.2 业务逻辑层

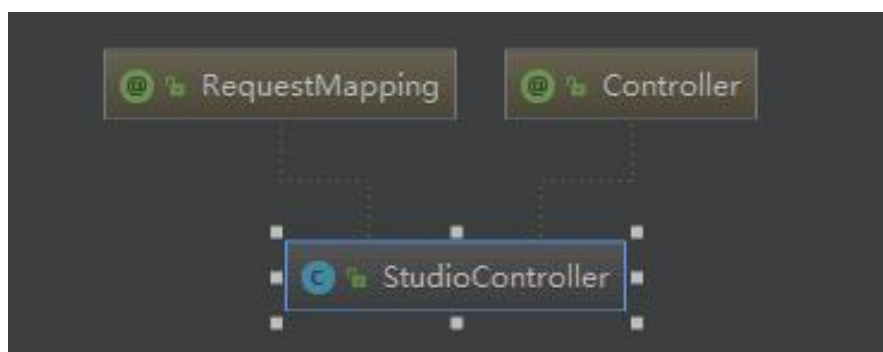
1. 当有人要查询/修改电影信息, 后台需要调用 movie 的持久化层去数据库中查找到此电影的所有信息, 然后返回给页面。业务流程类图如下如:



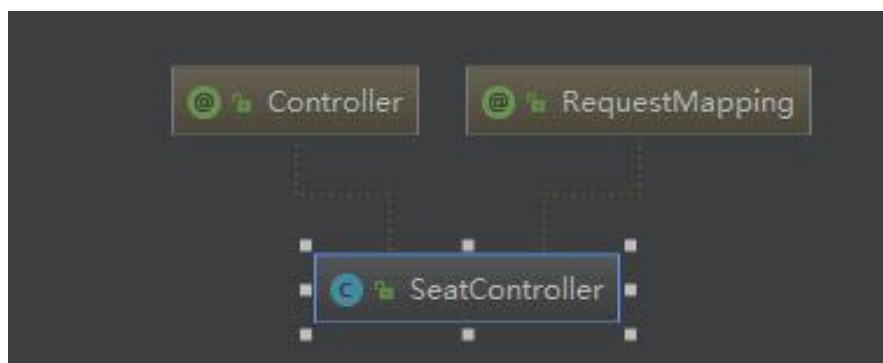
2. 当有人要查询/修改某个电影的演出计划信息, 后台需要调用 movie 的持久化层去数据库中查找到此电影的所有计划信息, 然后返回给页面。业务流程类图如下如:



3. 当有人要查询/修改演出厅信息, 后台需要调用 movie 的持久化层去数据库中查找演出厅的信息, 然后返回给页面。业务流程类图如下如:



4. 当有人要查询/修改某个演出厅座位信息, 后台需要调用 movie 的持久化层去数据库中查找到此演出厅的所有座位信息, 然后返回给页面。业务流程类图如下如:



5. 当有人要查询某个票的信息, 后台需要调用 movie 的持久化层去数据库中查找到此电影票的所有信息, 然后返回给页面。业务流程类图如下如:

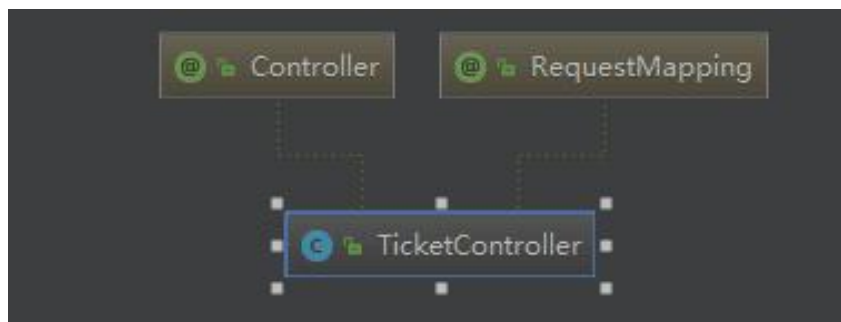


表 6 展示了所有业务逻辑层软件单元：

表 6 业务逻辑层软件单元

序号	软件单元标识符	软件单元(类)名称	功能说明	备注
1	Service_UC_01	MovieController	电影管理	
2	Service_UC_02	MovieTypeController	电影类型管理	
3	Service_UC_03	PicUploadController	图片上传管理	
4	Service_UC_04	ScheduleController	演出计划管理	
5	Service_UC_05	SeatController	座位管理	
6	Service_UC_06	StudioController	影厅管理	
7	Service_UC_07	TicketController	电影票管理	

### 3.2.3 持久化层

## 3.3 处理流程

### 3.3.1 订单查询 (TTMS\_UC\_03)

图 5 展示了订单查询时的全部过程：

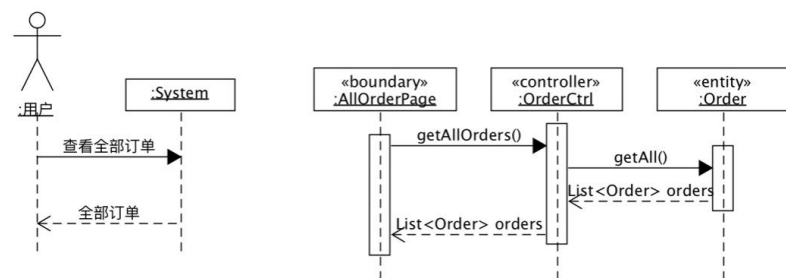


图 5 订单查询用例顺序图

图 6 展示了订单信息的所有用例顺序图：

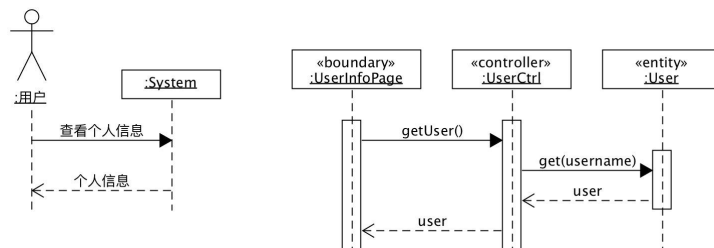


图 6 订单信息用例顺序图



## 4. 人机界面设计

### 界面设计策略：

金太阳剧院票务管理系统界面原则是方便、简洁、美观、一致。各子系统之间 界面风格统一。

让每个操作尽可能简单。

向用户展示接下来的交互。

不同内容的交互界面应该看起来不同。

只显示必要的信息。

清晰。

显示风格：图形界面；

显示方式：1920\*1080；

输出格式：显示布局、打印格式。

### 界面布局方案：

系统登录界面：（如图 7）

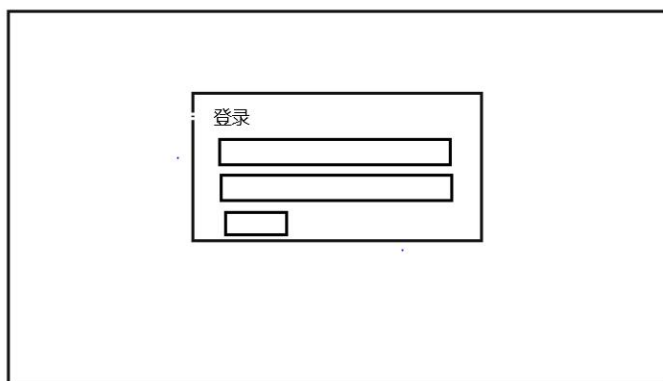


图 7 系统登录界面图

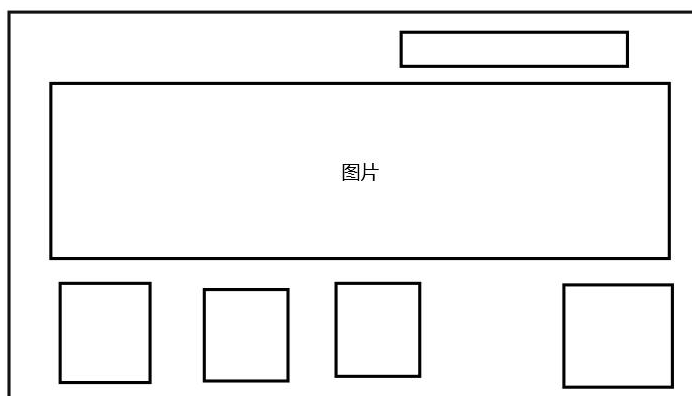


图 8 主界面图

5. 数据存储设计

5.1 内部数据结构

数据字典采用了数据结构中的树型结构

5.2 数据库

5.2.1 数据库设计

数据库表关系图如图 9 所示：

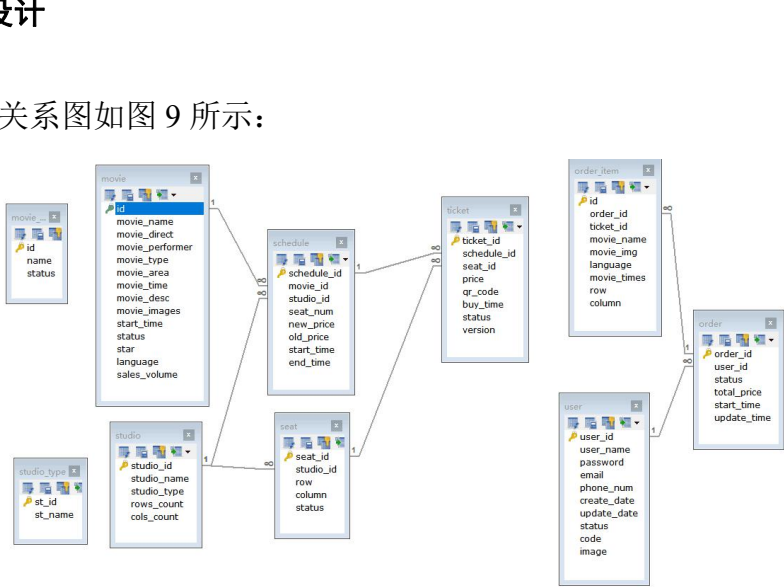


图 9 数据库表关系图

5.2.2 数据表定义

每一个实体都可以成为一张表，实体说明表如表 7 所示：

表 7 实体说明表

编号	表名	定义	功能
表 1	Movie	电影详情表	存储电影的相关信息
表 2	movie_type	电影类型表	存储电影类型相关信息
表 3	Order	订单表	存储订单相关信息
表 4	order_detail	订单详情表	存储订单详情
表 5	order_item	订单票信息表	存储订单票相关信息
表 6	Schedule	演出计划表	存储演出计划相关信息
表 7	Seat	座位表	存储座位相关信息

表 8	Studio	演出厅表	存储演出厅相关信息
表 9	studio_type	演出类型表	存储演出类型相关信息
表 10	Ticket	电影票表	存储电影票相关信息
表 11	User	用户信息表	存储用户信息

数据库中的剧目表设计如下：

表 8 Movie 表

列名	类型	允许为空	描述
id	bigint(20)	NOT NULL	电影 id（主键）
movie_name	varchar(50)	NOT NULL	电影名字
movie_direct	varchar(50)	NOT NULL	导演
movie_performer	varchar(50)	NOT NULL	主演
movie_type	int(11)	NOT NULL	类型
movie_area	varchar(50)	NOT NULL	制片国家/地区
movie_time	int(11)	DEFAULT NULL	片长
movie_desc	varchar(255)	NOT NULL	剧情介绍'
movie_images	varchar(255)	NOT NULL	电影图片
start_time	datetime	NOT NULL	上架时间
status	int(5)	DEFAULT '1'	电影状态 1-正常，2-下架，3-删除
star	double(10,2)	DEFAULT '5.00'	电影评分
language	varchar(20)	DEFAULT '国语',	电影语言
sales_volume	bigint(20)	DEFAULT '0'	电影销量

数据库中的订单表设计如下：

表 10 订单表 order

列名	类型	允许为空	描述
order_id	varchar(50)	NOT NULL	订单号（主键）
user_id	bigint(20)	NOT NULL	用户 id
status	int(10)	DEFAULT NULL	订单状态 1、未付款，2、已付款，3 交易成功、4. 交易关闭
total_price	bigint(20)	DEFAULT NULL	总价
start_time	Datetime	DEFAULT NULL	开始时间

update_time	Datetime	DEFAULT NULL	电影票时间刷新
-------------	----------	--------------	---------

数据库中的订单细节表设计如下：

表 11 订单详情表 order\_detail

列名	类型	允许为空	描述
detail_id	bigint(20)	NOT NULL	订单详情 id（主键）
order_id	varchar(50)	NOT NULL	订单号
row	int(10)	NOT NULL	行号
column	int(10)	NOT NULL	列号

数据库中的演出计划设计如下：

表 13 演出计划信息 schedule 表

列名	类型	允许为空	描述
schedule_id	bigint(20)	NOT NULL	演出计划 id（主键）
movie_id	bigint(20)	NOT NULL	电影 id
studio_id	bigint(20)	NOT NULL	演出厅 id
seat_num	bigint(20)	DEFAULT '0'	座位的总数
new_price	bigint(20)	NOT NULL	现价
old_price	bigint(20)	NOT NULL	原价
start_time	datetime	NOT NULL	开播时间
end_time	datetime	NOT NULL	结束时间

数据库中的订单细节表设计如下：

表 14 座位表 seat

列名	类型	允许为空	描述
seat_id	bigint(20)	NOT NULL	座位id（主键）
studio_id	bigint(20)	NOT NULL	影厅id
row	int(11)	NOT NULL	行号
column	int(11)	NOT NULL	列号
status	int(11)	NOT NULL	1-可选 2-已售 3-情侣座 4-损坏

数据库中的演出厅表设计如下：

表 15 演出厅 studio 表

列名	类型	允许为空	描述
----	----	------	----

studio_id	bigint(20)	NOT NULL	演出厅 id（主键）
studio_name	varchar(50)	NOT NULL	演出厅名称
studio_type	int(5)	NOT NULL	演出厅类型
rows_count	int(11)	NOT NULL	行数
cols_count	int(11)	NOT NULL	列数

数据库中的演出票设计如下：

表 17 电影票 Ticket 表

列名	类型	允许为空	描述
ticket_id	varchar(50)	NOT NULL	电影票 id（主键）
schedule_id	bigint(20)	NOT NULL	演出计划 id
seat_id	bigint(20)	NOT NULL	座位 id
price	bigint(20)	NOT NULL	价格
qr_code	varchar(255)	DEFAULT NULL	票的二维码
buy_time	datetime	DEFAULT NULL	购票时间
status	int(10)	DEFAULT '1'	票的状态 1-未售 3-已售
version	bigint(20)	DEFAULT '0'	票的版本号

数据库中的用户表设计如下：

表 18 用户 user 表

列名	类型	允许为空	描述
user_id	bigint(20)	NOT NULL	用户 id（主键）
user_name	varchar(50)	NOT NULL	用户名
password	varchar(32)	NOT NULL	密码
email	varchar(50)	NOT NULL	电子邮箱
phone_num	varchar(20)	DEFAULT NULL	电话号
create_date	datetime	NOT NULL	创建时间
update_date	datetime	NOT NULL	更新时间
status	tinyint(4)	DEFAULT NULL	1-正常 2-删除
code	varchar(50)	DEFAULT	二维码

		NULL	
image	varchar(100)	DEFAULT NULL	头像

## 6. 详细设计

### 6.1 MovieController (Service\_UC\_01)

#### 6.1.1 功能描述

MovieController:

功能及作用：getMoviesByPage（分页查询电影）

addMovie（添加电影）

updateMoviePre（修改电影前(先根据 id 查找电影)）

updateMovie（修改电影（不为空的字段更新））

deleteMovie（删除电影）

#### 6.1.2 处理流程

对电影进行增加，查找，修改，删除等一系列操作。

#### 6.1.3 内部数据

movieService：处理关于电影增删改查相关业务流程。

#### 6.1.4 测试要点

对电影进行添加、修改、查询、删除等一系列操作。

### 6.2 MovieTypeController (Service\_UC\_02)

#### 6.2.1 功能描述

MovieTypeController：

功能及作用：

getAllMoviesType（返回所有的电影类型）

### 6.2.2 处理流程

getAllMoviesType 返回所有电影类型

### 6.2.3 内部数据

movieTypeService：处理电影类型相关业务。

## 6.3 PicUploadController (Service\_UC\_03)

### 6.3.1 功能描述

PicUploadController：

功能及作用：

Upload（图片上传）

### 6.3.2 处理流程

Upload（图片上传）：上传文件，然后判断文件是否为图片文件，文件是否合法。

### 6.3.3 内部数据

picService：处理图片相关业务。

LOGGER：图片相关日志文件。

### 6.3.4 异常与错误处理

如果上传图片格式不正确或上传为文档，就会出现异常、错误。

### 6.3.5 测试要点

通过上传图片、文件等格式文件测试功能是否正确

## 6.4 ScheduleController (Service\_UC\_04)

### 6.4.1 功能描述

ScheduleController:

功能及作用:

- queryAllScheduleByDate (通过日期获取演出安排)
- queryAllSchedule (获取所有的演出计划)
- addSchedule (添加一个演出计划)
- deleteSchedule (删除一个演出计划)
- updateSchedule (更新一个演出计划)
- queryScheduleId (根据演出计划 id 获取演出计划)

### 6.4.2 处理流程

对演出计划进行一系列添加, 更新, 查询, 删除操作。通过日期、演出计划 id 获取演出计划。

### 6.4.3 内部数据

scheduleService: 处理演出计划相关业务

### 6.4.4 异常与错误处理

若演出计划 id 不正确, 则不能进行查找操作。

### 6.4.5 测试要点

对演出计划进行一系列添加、修改、查找、删除。

## 6.5 SeatController (Service\_UC\_05)

### 6.5.1 功能描述

SeatController:

功能及作用:

- updateSeats (批量更新座位状态)



### 6.5.2 处理流程

updateSeats 通过座位 id 更新座位状态。

### 6.5.3 内部数据

seatService：处理座位相关业务。

LOGGER：座位相关业务日志文件。

### 6.5.4 异常与错误处理

单个座位不能修改为情侣座

### 6.5.5 测试要点

将座位修改为可选，已售，情侣座，损坏这四种状态。

## 6.6 StudioController (Service\_UC\_06)

### 6.6.1 功能描述

StudioController：

功能及作用：

addStudio（添加演出厅）

deleteStudio（删除演出厅）

updateStudio（更新演出厅）

queryStudio（根据演出厅 id 查询演出厅）

queryAllWithAllSeats（查询演出厅所有座位）

### 6.6.2 处理流程

演出厅的添加、查询、修改删除操作，还有对演出厅的更新，查询所有座位。

### 6.6.3 内部数据

studioService：处理演出厅相关业务

#### 6.6.4 测试要点

对演出厅进行添加、查询、删除、更新，以及查询演出厅所有座位。

### 6.7 TicketController (Service\_UC\_07)

#### 6.7.1 功能描述

TicketController:

功能及作用:

queryByScheduleId (通过演出计划 id 查询电影票)

checkTicketState (锁票)

updateTicketsStatus (更新电影票状态)

updateTicketStatusByTicketId (通过电影票 id 更新电影票状态)

#### 6.7.2 处理流程

通过调用演出计划 id 查询电影票，对电影票进行锁定，更新电影票相关状态，通过电影票 id 更新电影票状态。

#### 6.7.3 内部数据

ticketService: 处理电影票相关业务

discoveryClient:

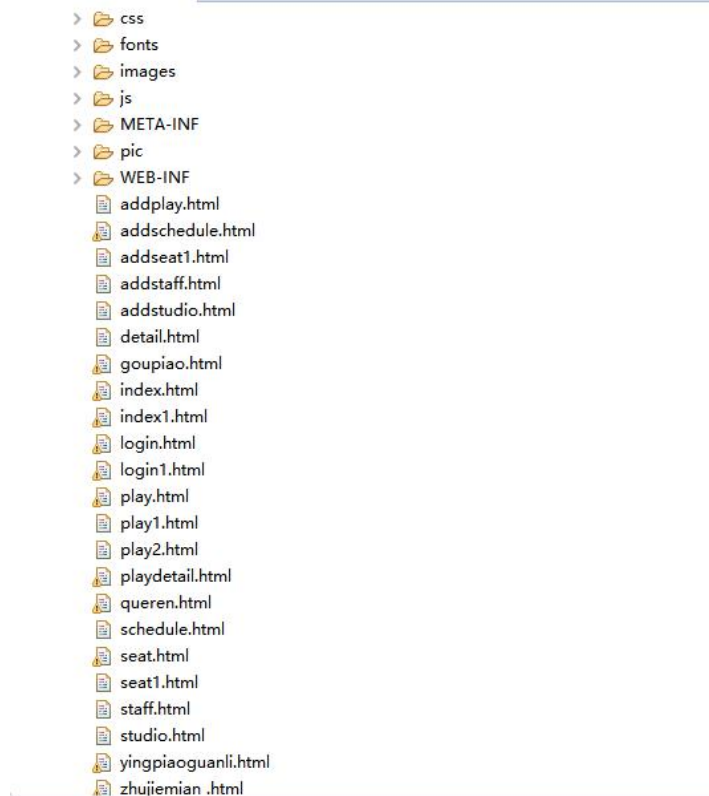
#### 6.7.4 测试要点

通过演出计划 id 查询电影票，锁定电影票，更新电影票状态一系列操作。

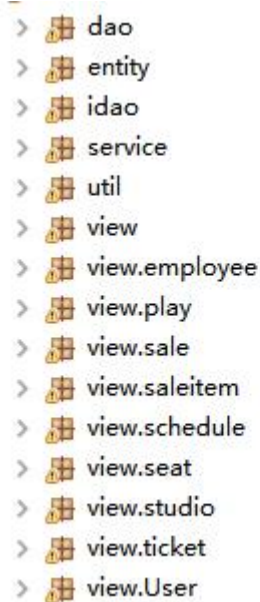
## 7. 开发架构设计

### 7.1 工程结构

前端:



后台:



## 7.2 源代码文件

前端:

Html、css 和.js

后台：

.java 业务流程

.mdf、.ldf 数据库 sql 语句配置文件

## 8. 物理架构设计

### 8.1 网络环境

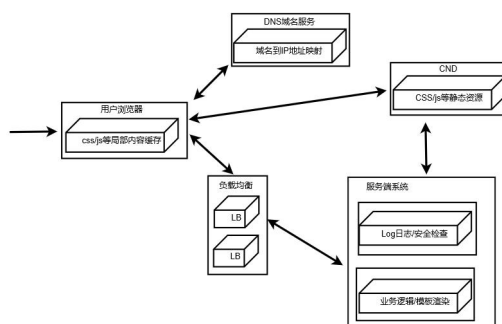


图 10 网络拓扑架构图

### 8.2 部署方案

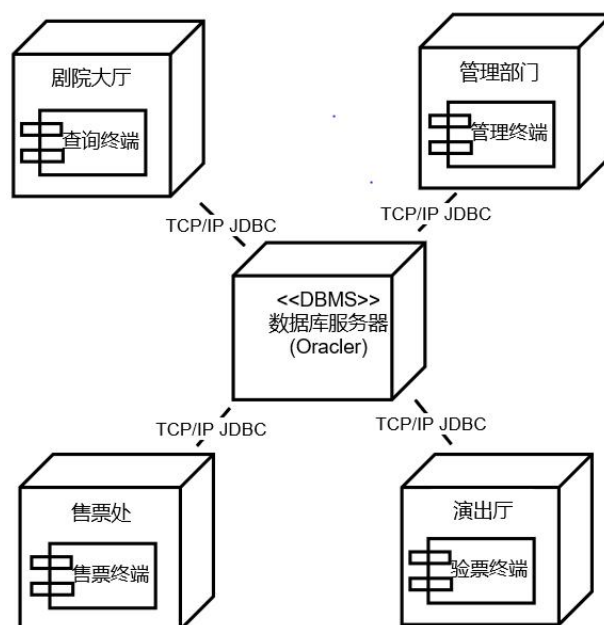


图 11 部署方案

