|  |  |
| --- | --- |
| 项目编号 | **ML.2018.HALP** |
| 文档编号 | **ML.2018.HALP.SD**  **S** |
| 密 级 | 内部保密 |



HALP

软件设计说明

版本：V1.0

软件 1404 班韩钊

## 拟 制：

审 核 ： 标准化： 会 签 ：

批 准 ：

文档修改记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修改内容描述 | 修改人 | 日期 | 备注 |
| V1.0 | 撰写文档 | 韩钊 | 2018/3/8 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

1. [引言 1](#_bookmark0)
   1. [编写目的 1](#_bookmark1)
   2. [项目概述 1](#_bookmark2)
   3. [术语定义 1](#_bookmark3)
   4. [缩写说明 1](#_bookmark4)
   5. [引用文档 2](#_bookmark5)
2. [软件设计决策 2](#_bookmark7)
   1. [设计目标 2](#_bookmark8)
   2. [设计原则 2](#_bookmark9)
   3. [设计约束 2](#_bookmark10)
      1. [遵循标准 2](#_bookmark11)
      2. [运行环境 2](#_bookmark12)
      3. [开发环境及工具 3](#_bookmark13)
      4. [技术限制 3](#_bookmark14)
      5. [其他 3](#_bookmark15)
3. [逻辑架构设计 3](#_bookmark16)
   1. [设计决策 3](#_bookmark17)
   2. [软件单元 4](#_bookmark18)
      1. [界面层 4](#_bookmark19)
      2. [业务逻辑层 5](#_bookmark20)
   3. [处理流程 6](#_bookmark21)
      1. [演出厅管理 (Studio) 6](#_bookmark22)
      2. [售票（sale） 6](#_bookmark23)
4. [人机界面设计 7](#_bookmark24)
5. [数据存储设计 8](#_bookmark25)
   1. [内部数据结构 8](#_bookmark26)
   2. [数据库 8](#_bookmark27)
      1. [数据库设计 8](#_bookmark28)
      2. [数据表定义 9](#_bookmark29)
   3. [数据文件 9](#_bookmark30)
6. [详细设计 10](#_bookmark31)
   1. [演出厅管理(Studio) 10](#_bookmark32)
      1. [功能描述 10](#_bookmark33)
      2. [处理流程 10](#_bookmark34)
      3. [内部数据 11](#_bookmark35)
      4. [异常与错误处理 11](#_bookmark36)
      5. [测试要点 11](#_bookmark37)
   2. [售票（sale） 11](#_bookmark38)
      1. [功能描述 11](#_bookmark39)
      2. [处理流程 12](#_bookmark40)
      3. [内部数据 12](#_bookmark41)
      4. [异常与错误处理 12](#_bookmark42)
      5. [测试要点 13](#_bookmark43)
7. [开发架构设计 13](#_bookmark44)
   1. [工程结构 13](#_bookmark45)
   2. [源代码文件 13](#_bookmark46)
   3. [系统组件 13](#_bookmark47)
8. [物理架构设计 14](#_bookmark48)
   1. [网络环境 14](#_bookmark49)
   2. [部署方案 15](#_bookmark50)

“星都国际剧院票务管理系统”软件设计说明

# 引言

## 编写目的

本文档用于说明星都国际剧院票务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，是星都国际剧院票务管理系统软件实现的基础。本文的预期读者包括：

* + - 开发人员 韩钊，张根，霍延洲，张泽煜，邱泽鸣
    - 测试人员 张泽煜
    - 项目管理人员 韩钊

## 项目概述

简要介绍项目的背景和总体目标……。本项目基本信息如下：

* + - 项目名称：星都国际剧院票务管理系统；
    - 项目编号：ML.2016.TTMS；
    - 投 资 方：星都国际大剧院；
    - 用 户：星都国际大剧院；
    - 开 发 方：魔驴工作室。

## 术语定义

无

## 缩写说明

无

## 引用文档

本文引用的文档及标准参见[表 1](#_bookmark6)。

表 1 引用文档

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序  号 | 文档编号 | 标题 | 版本号 | 修订日期 | 编制单位 |
| 1 | ML.2016.TTMS.SRS | 第三小组软件需求规格说  明书 | 1.0 | 2016/3/31 | 魔驴工作室 |

# 软件设计决策

## 设计目标

本系统实现星都国际剧院的整个人力管理和业务管理功能，包括人员管理， 售票模块，财务报表功能等，实现管理员和售票人员的不同权限操作，保证系统使用的流畅性和系统的健壮性。

## 设计原则

分层架构编写系统，每个模块单独书写，整个系统应用数据库存储数据，系统在使用过程中实时进行内存使用优化。

## 设计约束

* + 1. 遵循标准

系统界面简洁易懂，内容明确，操作性强，系统分层架构书写，每个模块都有 dao 层，model 层，server 层，view 层组成。

* + 1. 运行环境

本系统可在 Windows xp 及以上系统下运行。

* + 1. 开发环境及工具

本系统由 java 语言和 mysql 数据库开发，运行环境为 eclipse mar 2.0 版本。

* + 1. 技术限制

本软件存储要求 5G 以上，运行内存 2G 以上，系统数据存储在数据库中， 运行添加时可能会有些许延迟。

* + 1. 其他

本软件的登陆程序采用 md5 进行加密，对系统的安全性有更多的保障，本系统在运行的时候采用多线程操作，提高系统的运行速度，提高用户的体验感。

# 逻辑架构设计

本系统采用分层进行架构，分为 5 个层次进行编写，分为 view 层、service 层、DAO 层、IDAO 层、model 层。

## 设计决策

图 1 星都国际剧院软件逻辑架构

## 软件单元

* + 1. 界面层

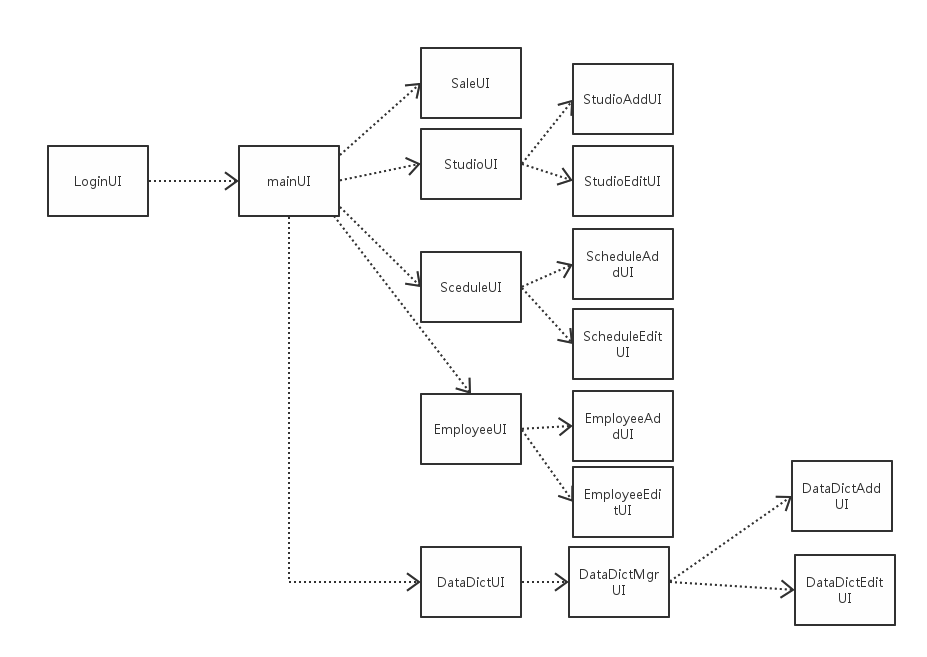


图 2 界面层类图

表 2 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 | DataDicUI | DataDictAddUI DataDictEditUI DataDictInformation  DataDictMgrUI | ，  ，  ， | 数据字典的界面显示 |  |
| 2 | EmployeeUI | EmployeeAddUI EmployeeEditUI EmployeeOptionPanel  EmployeeTableListener | ，  ，  ， | 员工管理的界面显示 |  |
| **3** | listener | MenuListener | | 监听事件 |  |
| **4** | Loading | Loading | | 登陆的数据处理 |  |
| **5** | LoginUI | LoginListener，LoginUI | | 登陆的界面显示 |  |
| **6** | SaleUI | SaleOptionPanelByPlay ，  SaleOptionPanelBySchedule，  SaleUI | | 售票界面的显示 |  |
| **7** | SceduleUI | ScheduleAddUI ScheduleEditUI ScheduleOptionPanel  ScheduleTableListener | ，  ，  ， | 演出计划的显示 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **8** | StudioUI | StudioAddUI，StudioEditUI， StudioOptionPanel ，  StudioTableListener | 演出厅的主界面显示 |  |

* + 1. 业务逻辑层

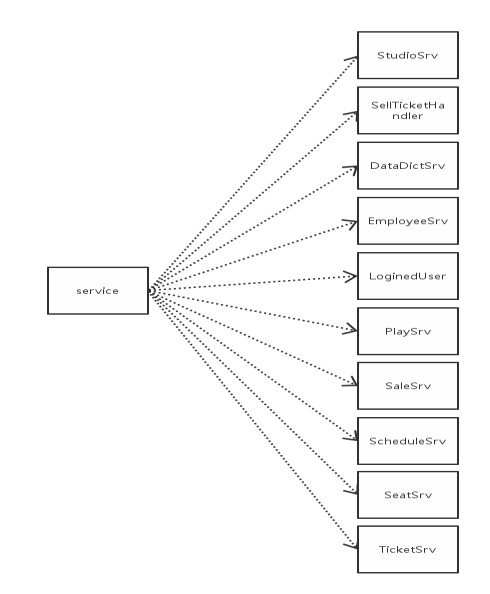


图 3 业务逻辑层类图

表 5 业务逻辑层软件单元构成

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 软件单元标  识符 | 软件单元(类)名称 | 功能说明 | 备注 |
| 1 | service | DataDictSrv | 数据字典 |  |
| 2 | service | EmployeeSrv | 员工管理 |  |
| **3** | service | LoginedUser | 用户登录 |  |
| **5** | service | PlaySrv | 剧目管理 |  |
| **6** | service | SaleSrv | 销售单 |  |
| **7** | service | ScheduleSrv | 演出计划 |  |
| **8** | service | SeatSrv | 座位管理 |  |
| **9** | service | SellTicketHandler | 售票管理 |  |
| **10** | service | StudioSrv | 演出厅管理 |  |

## 处理流程

* + 1. 演出厅管理 **(Studio)**

，

: 系统管理员



: SysMenuUI

1 : enterStudioMgr()

<<create>>

service层

2 : StudioMgrUI

4 <<create>> : StudioSrv

3 : setVisible(false)

7 : setVisible(true)

5 : stuList := FetchAll()

**ref** 提取所有演出厅 据

数

6 : showStudioList(stuList)

o()

8 : addStudi

**ref** 添加演出厅

**ref** 修改演出厅

)

9 : modStudio(

图 4 演出厅管理用例处理流程

* + 1. 售票（**sale**）

1 : selectPlay()

5 : selectSchedule()

9 : selectTicket()



: SysMenuUI

4 : showSchList(schList)

8 : showTickets(ticketList)



: TicketSe**l**UI

: 售票员

2 : schList := getSchedule(playID)

6 : ticketList := getTicket(schID)

: PlaySrv

: Se**l**TicketHandler

3 : schList := fetchSchedule(playID)



7 : ticketList := fetchTicket(schID)

: ScheduleSrv

: TicketSrv

: Sale

图 5 售票处理流程（1）

10 : beginNewSale()

9 : showPlayList(playList)



: 售票员

1 : enterTicketSale()

<<create>>

2

: TicketSellUI

: SysMenuUI

<<create>> 3

7 : playList := getAllPlays()

<<create>> 4

8 : playList := FetchAll()

: PlaySrv

5



11 : makeNewSale()

12 <<create>>



6

: SellTicketHandler

<<create>> <<create>>

: ScheduleSrv

: TicketSrv

: Sale

图 6 售票处理流程（2）

# 人机界面设计

右下部分显示表格内容，左边显示功能菜单栏，上方显示各功能模块的具体内容，每个菜单选项选择了之后，右边的界面会刷新一次，右上方对应为该模块的功能，比如演出厅有演出厅的添加，修改，删除和演出厅的座位管理功能，右下方显示的是该模块拥有的数据，整个界面保持风格统一，操作简洁。

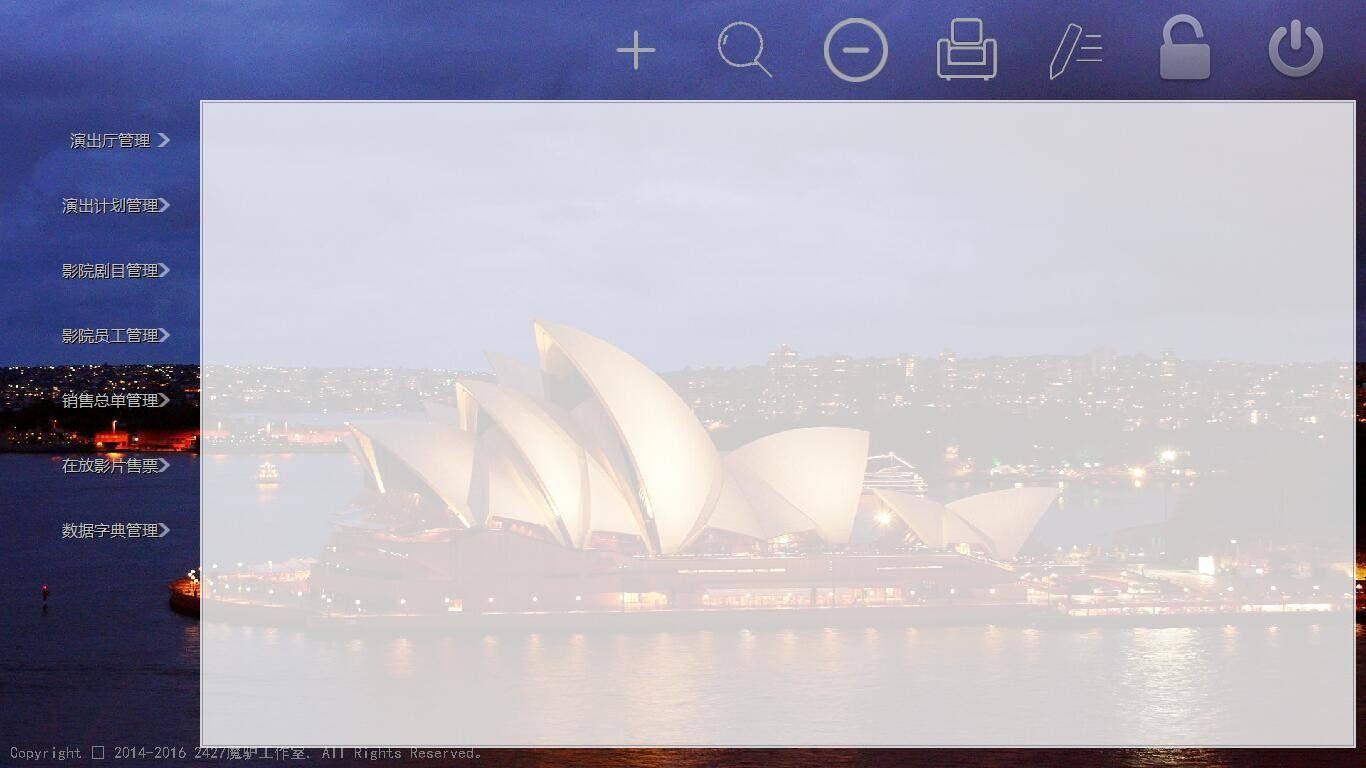


图 7 人机界面设计

# 数据存储设计

## 内部数据结构

用户的数据和权限存储在数据库中，数据通过界面层用户输入或点击传输到服务层，再由服务层传输到业务逻辑层，执行具体操作。

## 数据库

* + 1. 数据库设计

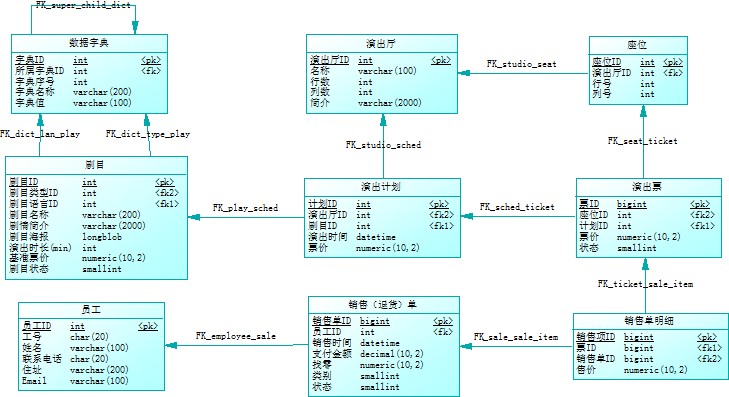


图 8 数据库设计图

* + 1. 数据表定义

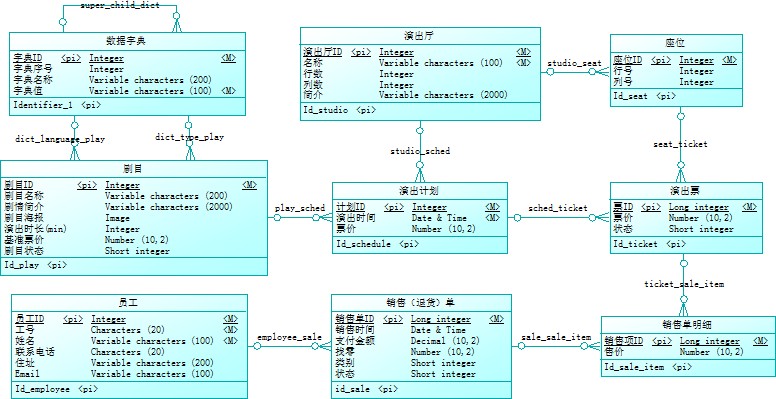


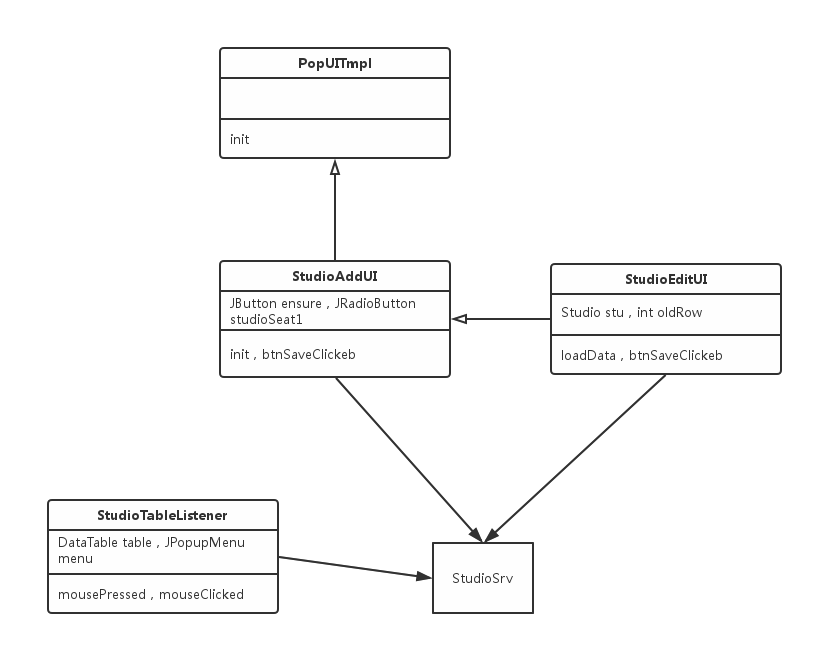
图 9 数据表设计图

## 数据文件

无数据文件。

# 详细设计

* 1. 演出厅管理**(Studio)**
     1. 功能描述



本演出厅管理模块有演出厅的增加，修改，删除，和演出厅的座位管理（如图 10）。

图 10 演出厅功能描述图

* + 1. 处理流程

演出厅 ID 由数据库自动生成，演出厅的名称，行数，列数，和简介在输入后，传输到业务逻辑层，根据业务逻辑层的具体方法进行处理，数据再传到 dao 层，进行数据库的信息更新操作。

演出厅的座位管理时，点击演出厅对应的座位，可变色，不用的图例代表不同的状态，选择好了后，点击保存按钮，座位数据由 view 层传到 service 层，再到 dao 层进行数据库的操作。

* + 1. 内部数据

演出厅 id 在创建演出厅时，由数据库自动生成，演出厅名称，行数，列数和简介由用户在创建的时候自行输入，没有具体要求（如图 11）



图 11 演出厅内部数据图

* + 1. 异常与错误处理

演出厅管理模块可能出现的错误是演出厅修改的时候，数据库的删除和数据库外键数据的修改。

* + 1. 测试要点

对演出厅管理系统的增加，修改，删除和整个模块做必要的黑盒测试。

* 1. 售票（**sale**）
     1. 功能描述

售票员提供剧目供选择，再提供演出场次，再提供对应的座位选择（如图

13）。

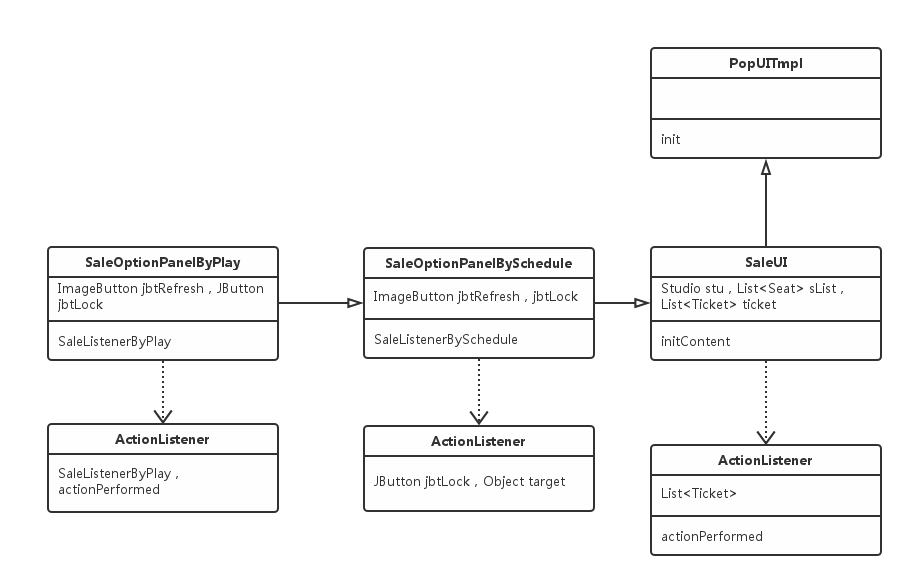


图 12 售票功能描述图

* + 1. 处理流程

售票员根据顾客所选的剧目，选择对应剧目，系统显示该剧目所有的场次供选择，再根据所选择的场次提供对应的座位信息选择。

* + 1. 内部数据

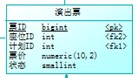
票 id 由数据库生成时自动生成，座位 id 和计划 id 和票价由系统自动获取， 票的状态根据用户的购买情况对应存储（如图 13）。

图 13 票的内部数据图

* + 1. 异常与错误处理

在选择完场次后，选择座位的时候，要给出对应场次的座位信息，不能提供错座位的信息，如果选择完剧目，对应的演出计划已经下线时，那么就不能够在选择该演出计划对应的座位的票。

* + 1. 测试要点

测试卖票功能能否正常使用，测试在选定座位时，对应的票是否会被锁定， 其他用户在此期间不能选择相同的票。

# 开发架构设计

## 工程结构

管理终端

查询终端

**DBMS**

售票终端

验票终端

图 14 项目工程结构图

## 源代码文件

Recouse 里有数据库的连接，service 是业务逻辑层。

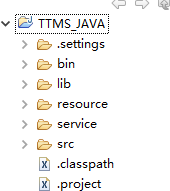


图 15 源代码结构文件图

## 系统组件

采用分层架构，dao 层为数据库操作，idao 层定义了 dan 层类的方法和接口，

model 层是类的定义，service 层是业务逻辑层，传递来自用户操作的信息，view 层是用户的人机交互图形界面，负责人的操作。

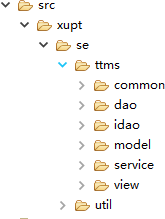


图 16 系统组件结构图

# 物理架构设计

## 网络环境

图 17 网络环境拓扑图

## 部署方案

TCP/IP JDBC

TCP/IP JDBC

TCP/IP JDBC

TCP/IP JDBC

售票处

售票终端

演出厅

验票终端

<<DBMS>>

数据库服务器

**(Oracler)**

管理部门

管理终端

剧院大厅

查询终端

图 18 项目部署方案图