

体系结构设计描述文档-I

申请重判期间

软件体系结构文档模板

文档修改历史

修改人员	日期	修改原因	版本号
------	----	------	-----

目录

[TOC]

1.引言

1.1 编制目的

本报告详细完成对连锁影院管理系统的概要设计，达到指导详细设计和开发的目的，同时实现和测试人员以及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

1.2 词汇表

词汇名称	词汇含义	备注
NJU	南京大学	无
_ui	表示某展示层	...
_bl	表示某业务层	...
_data	表示某数据层	...

1.3 参考资料

- 1.IEEE std 1471-2000
- 2.丁二玉，刘钦.计算与软件工程（卷二）[M]机械工业出版社，2012：134—182

2.产品概述

☐ 是否有文档标题

填写备注

☐ 是否有目录

目录TOC被吃了。

☐ 是否有编号

填写备注

☐ 编号是否逐级标示

填写备注

☐ 变更记录是否明确

没有变更记录

☐ 是否选择了适合的体系结构风格表达了系统的高层抽象

图被吃了，看不出来

☐ 是否每一个用例都有响应的逻辑模块来实现

填写备注

☐ 是否加入登录注册权限管理等系统功能

填写备注

☐ 是否定义了逻辑层接口？

填写备注

☐ 是否定义了数据层接口？

填写备注

☐ 逻辑层是否负责系统业务上的逻辑计算？

填写备注

☐ 数据层是否负责提供数据服务

参考NJU连锁院线用例文档和软件需求规格说明中队产品的概括描述

3.逻辑视图

连锁院线管理系统中，选择了分层体系结构风格，将系统分为3层（展示层、业务逻辑层、数据层）能够很好地示意整个高层抽象。展示层包含web界面地实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如图3-1和图3-2所示

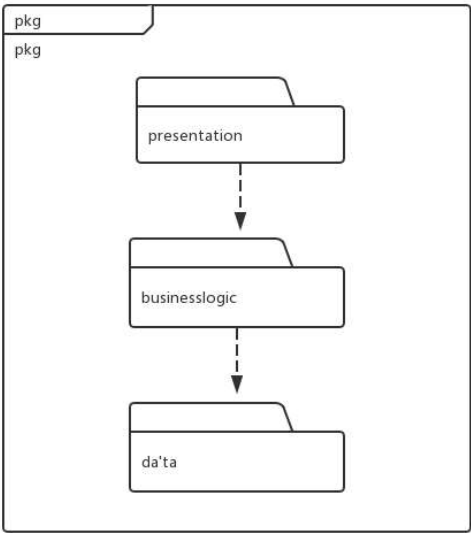


图3-1 参照体系结构风格的包图表达逻辑视角

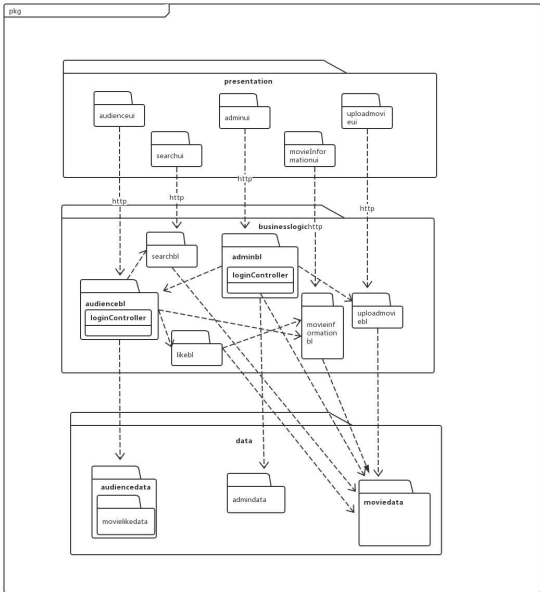


图3-2 软件体系结构逻辑设计方案

填写备注

☐ 是否用过VO和PO在各层的接口传递数据？

填写备注

☐ 是否合理的依赖第三方的库？

填写备注

☐ 是否说明了部署的环境？

填写备注

☐ 是否说明了其它依赖的包？

填写备注

☐ 是否说明了系统安装的包？

填写备注

☐ 是否说明了部署的节点？

填写备注

☐ 是否说明了网络通讯方式？

填写备注

☐ 是否说明了数据持久化的定义？

填写备注

☐ 是否违反了共同封闭原则CCP？

填写备注

☐ 是否违反了共同重用原则CRP？

填写备注

☐ 是否违反了重用发布等价原则REP？

填写备注

☐ 是否违反了无环依赖原则ADP？

没有图

4.组合视图

4.1 开发包图

连锁院线管理系统的最终开发包设计如表4-1所示

开发包开发包	依赖的其他开发包
mainui	audienceui、adminui、vo、movieInformationui、searchui、uploadmovieui
audienceui	audienceservice、界面类库包、vo
audienceservice	
audiencebl	audienceservice、audiencedataservice、po、searchbl、likebl、movieInformationbl
audiencedata	databaseutility、po、audiencedataservice、likedataservice、Mysql
adminui	adminservice、界面类库包、vo
adminservice	
adminbl	adminservice、admindataservice、po、uploadmoviebl
admindata	databaseutility、po、admindataservice、Mysql
uploadmovieui	uploadmovieservice、界面类库包、vo
uploadmovieservice	
uploadmoviebl	uploadmovieservice、moviedataservice、po、
moviedata	databaseutility、po、moviedataservice、Mysql
searchui	searchservice、界面类库包、vo
searchservice	
searchbl	audienceservice、audiencedataservice、po、searchbl、likebl、movieInformationbl

☐ 是否违反了稳定依赖原则SDP?

填写备注

☐ 是否违反了稳定抽象原则?

填写备注

☐ 是否能够为后续开发提供一个足够的视角?

填写备注

☐ 是否所有的非功能属性都得到了满足?

填写备注

☐ 是否所有的项目约束都得到了满足?

填写备注

☐ 不同的体系结构设计视角的依赖是否一致?

填写备注

☐ 体系结构设计是否能应对可能发生的变更?

填写备注

SEEC

软件工程与计算II-2019

作业

☐ 代码作业

☐ 文档作业

☐ 评审作业

小组

我的小组

开发包开发包	依赖的其他开发包
movieinformationui	movieinformationservice 、界面类库包、 vo
movieinformationservice	
movieinformationbl	moviedataservice、 po、
vo	
po	
界面类库包	javax.swing, java.awt, java.2D, java.3D
java RMI	java.rmi, <a href="#">java.net</a>
JDBC	
databaseutility	JDBC
audiencedataservice	Java RMI、 po
admindataservice	Java RMI、 po
moviedataservice	Java RMI、 po

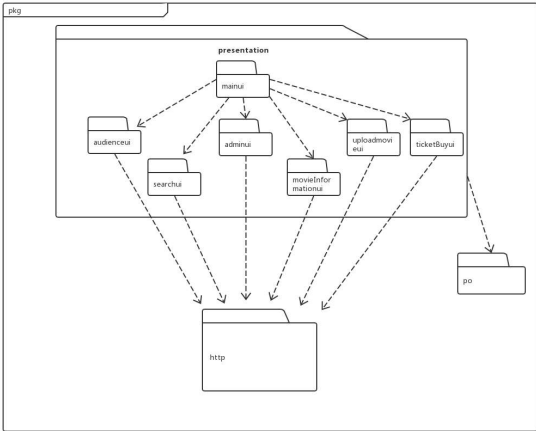


图4-1 连锁院线管理系统web端开发包图

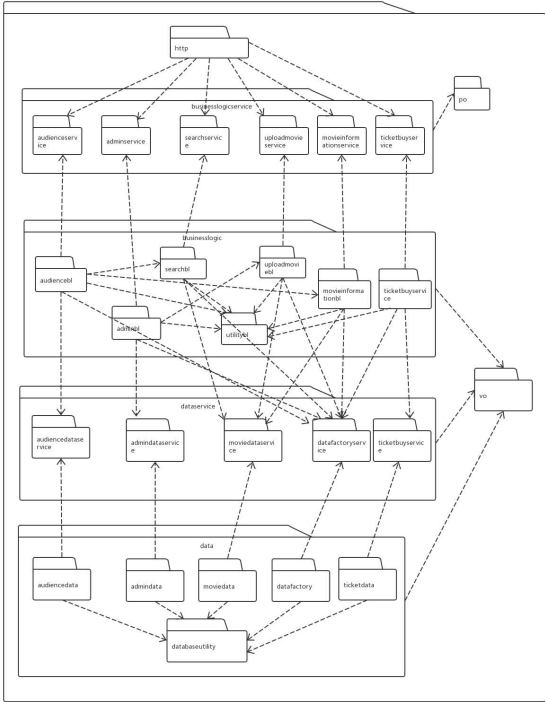
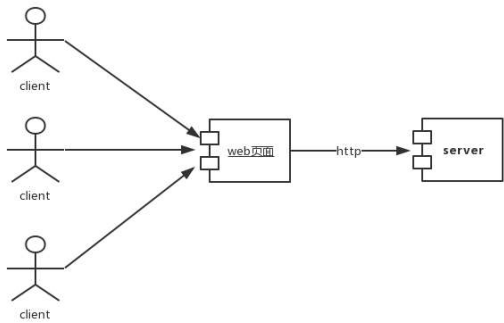


图4-2 连锁院线管理系统服务器端开发包图

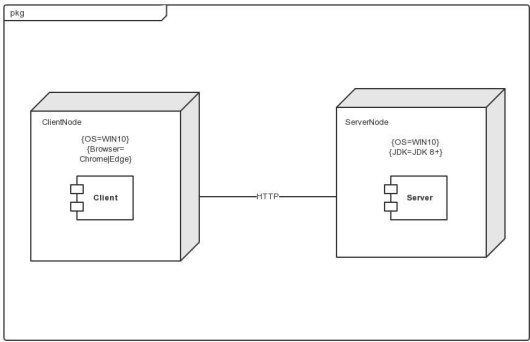
4.2 运行时进程

在连锁院线管理系统中，会有多个客户使用一个web端和一个服务器端相连。其进程如图4-3所示。结合部署图，web页面在客户机器上展示，服务器端进程在服务器端机器上运行



4.3 物理部署

CINEMA系统中，客户经由浏览器通过URL访问，服务器端构件部署在服务器上，包括所有后端代码以及数据库。由于是web项目，客户直接通过浏览器访问网页，因此客户端不需要独立部署。部署图如图1所示。



5.接口视角

5.1模块的职责

客户端模块和服务器模块视图分布如图7和图8所示。客户端各层和服务端各层的职责分别如表2和表3所示。



图5-1



图5-2

表2 客户端各层的职责

层	职责
启动模块	负责初始化网络通信机制，启动用户界面
用户界面层	基于web的影院用户界面
业务逻辑层	对于由用户端传来的数据及操作进行业务处理
客户端网络模块	利用http请求向服务器获取数据

表3 服务器端各层的职责

层	职责
启动模块	负责初始化网络通信机制，启动用户界面
数据层	负责数据的持久化及数据访问接口
服务器端网络模块	对服务端发来的http请求处理回复

每一层只是使用下方直接接触的层，层与层之间仅仅通过接口的调用来完成。层之间调用的接口如表4所示。

表4 层之间调用的接口

接口	服务调用方	服务提供方
LoginBLService MemberBLService TicketBLService MovieBLService PromotionBLService SaleBLService ArrangeBLService	客户端展示层	客户端业务逻辑层

接口	服务调用方	服务提供方
UserDataService	客户端业务逻辑层	服务器端数据层
MemberDataService		
TicketDataService		
MovieDataService		
PromotionDataService		
SaleDataService		
ArrangeDataService		
DatabaseFactory		

5.2用户界面层的分解

根据需求，系统存在以下界面。其中影院员工和用户访问的电影列表界面及电影详情界面可复用，未登录时亦可访问该界面。

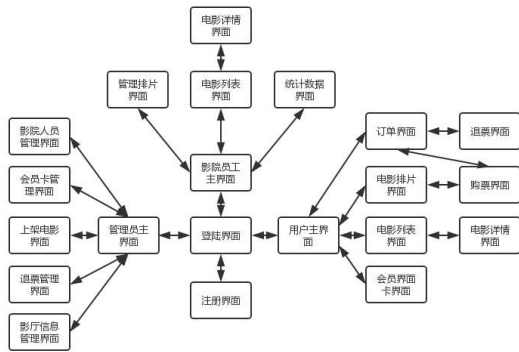


图5-3

5.2.1用户界面层主要组件的构成和职责

组件	职责
MainFrame	作为容器用于显示子组件
MovieDetail	用于展示电影详情
MovieList	显示电影列表
OrderTicket	用于显示选择电影的上座情况并提供选座功能
BillHistory	用于显示账单历史
DataAnalysis	用于展示选择电影的统计分析数据
PutAwayMovie	用于输入信息上架电影



组件	职责
UserCenter	用于管理用户信息和会员卡信息

5.2.2用户界面层组件所依赖的主要服务

用户界面视图组件本身不需要提供接口，视图逻辑由视图本身提供。只需依赖后端服务器提供接口进行数据处理。下表提供本阶段所需的主要接口

服务名	服务
LoginService	登陆界面业务逻辑接口
RegisterService	注册界面业务逻辑接口
MovieinfoService.list	提供电影列表展示信息
MovieInfoService.detail	提供单个电影详细信息
MovieinfoService.putAw ay	上架电影
OrderTicketService.Info	提供单个电影某个场次选座情况
OrderTicketService.order	对单个电影某个场次进行选择购票
CinemaService.*Blservice	对排片信息进行相应的增删改查操作

5.2.3

用户界面利用Vue框架及iview、element等ui组件库实现5.3业务逻辑层分解

业务逻辑层包括多个针对页面的业务逻辑处理对象。例如，audience对象负责处理观众登陆界面的业务逻辑；admin对象负责处理管理员登陆界面的业务逻辑。业务逻辑层的设计图如5.3-1所示。

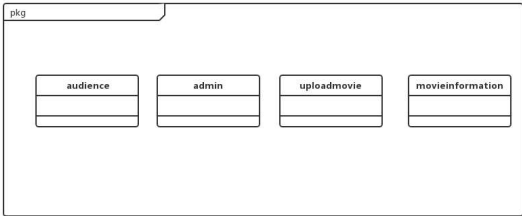


图5-4

###5.3 业务逻辑层分解

5.3.1 职责

业务逻辑层模块的指责如表所示

表1 业务逻辑层模块的职责

模 块	职 责
audiencebl	负责实现对应与观众登陆 并进行操作所需要的服务
adminbl	负责实现对应与管理员登 陆并进行操作所需要的服 务
ticketbl	负责实现票务相关的服务
movieinformationbl	负责实现电影模块界面所 需要的服务
movieLayOutbl	负责实现排片界面所需要 的服务

5.3.2接口规范

audiencebl模块的接口和规范如表所示

提供的服务 (供接口)		
	语法	public ResultMessage Audiencelogin( String id,String password);
audience.login	前置条件	password 符合 输入规则
	后置条件	查找是否存在相 应的audience用 户，根据输入的 password返回 结果
需要的服务 (需接口)		
服务名	服务	
DatabaseFacory.getAudie nceDatabase	得到audience数据库的服 务的印用	
audienceDataservice.inser t(UserPo po)	在audience数据库插入PO 对象	
movieDataservice.like	修改movie数据库的like人 数	
movieDataservice.search	查找movie数据库中的电 影	

adminbl模块的接口和规范如表所示

提供的服务 (供接口)		
----------------	--	--

提供的服务 (供接口)		
	语法	public ResultMessage adminlogin(String id,String password);
admin.login	前置条件	password 符合输入规则
	后置条件	查找是否存在相应的admin用户，根据输入的password返回结果

需要的服务 (需接口)	
服务名	服务
DatabaseFacory.getAudienceDatabase	得到admin数据库的服务的印用
adminDatasevice.insert(UserPo po)	在admin数据库插入PO对象
movieDatasevice.upload	向movie数据库中添加电影
movieDatasevice.search	查找movie数据库中的电影
movieDatasevice.change	修改movie数据库中的电影信息

movieinformationbl模块的接口和规范如表所示

提供的服务 (供接口)		
	语法	public ResultMessage movieBrowse(String movieName);
movie.browse	前置条件	movieName 符合输入规则
	后置条件	查找是否存在相应的movie，根据输入的movieName返回结果
需要的服务 (需接口)		

需要的服务		(需接口)
服务名		服务
DatabaseFacory.getMovieDatabase		得到 movie数据库的服务的印用
movieDataservice.change		修改movie数据库中的电影信息
movieDataservice.upload		将数据添加至movie数据库

movieLayoutbl模块的接口和规范如表所示

提供的服务			(供接口)
语法			public ResultMessage movielayout(String[] layoutInformation);
movielayout.change	前置条件	操作人为管理员身份	
	后置条件	将结果添加至对应数据库，如时间冲突予以提示	

需要的服务		(需接口)
服务名		服务
movielayoutDataservice.change		将排片添加至 movielayout数据库

ticketbl模块的接口和规范如表所示

提供的服务			(供接口)
语法			public ResultMessage ticket.buy(String time,String money,String place);
ticket.buy	前置条件	用户已登陆	
	后置条件	查找是否存在对应场次电影且所选座位为空	

需要的服务		(需接口)
-------	--	-------

需要的服务	(需接口)
服务名	服务
DatabaseFacory.getAudienceDatabase	得到audience数据库的服务的印用
ticketDataService.buy	在ticket数据库中寻找对应数据进行修改

5.4 数据层分解

数据层主要给业务逻辑层提供数据访问服务，包括对于持久化数据的增、删、改、查。User业务逻辑需要的服务由UserDataService接口提供；MovieInformation业务逻辑需要的服务由MovieDataService接口提供;Ticket业务逻辑需要的服务由TicketDataService接口提供；ScheduleArrange业务逻辑需要的服务由SADDataService接口提供；VIP业务逻辑需要的服务由VIPDataService接口提供。由于持久化数据的保存可能存在多种形式：序列化文件、数据库等，所示抽象了数据服务。数据层模块的描述具体如图所示。

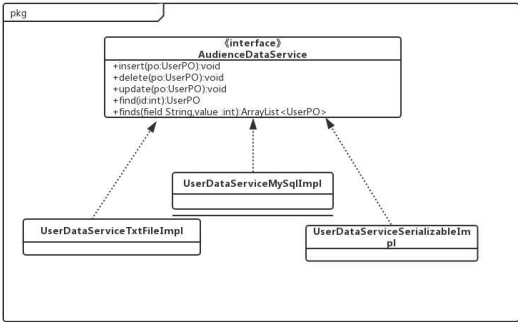


图5-5

5.4.1 职责

数据层模块的指责如表所示。

模块	职责
UserDataService	持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
UserDataServiceSerializableFileImpl	基于序列化文件的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
UserDataServiceMySQLImpl	基于MySQL数据库的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务

模块	职责
MovieDataService	持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
MovieDataServiceSerializableFileImpl	基于序列化文件的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
MovieDataServiceMySQLImpl	基于MySQL数据库的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
TicketDataService	持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
TicketServiceSerializableFileImpl	基于序列化文件的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
TicketServiceMySQLImpl	基于MySQL数据库的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
SCDataService	持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
SCServiceSerializableFileImpl	基于序列化文件的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
SCServiceMySQLImpl	基于MySQL数据库的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
VIPDataService	持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
VIPServiceSerializableFileImpl	基于序列化文件的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
VIPServiceMySQLImpl	基于MySQL数据库的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务

5.4.2接口规范

数据层模块的接口规范如表所示

提供的服务（供接口）		
UserDataService.find	语法	public UserPO find( 
	前置条件	无
	后置条件	按ID进行查找返回机
UserDataService.insert	语法	public void insert(U 
	前置条件	同样ID的po在mapp
	后置条件	在数据库中增加一个
UserDataService.delete	语法	public void delete(L 
	前置条件	在数据库中存在相同
	后置条件	删除一个po
UserDataService.update	语法	public void update( 
	前置条件	在数据库中存在相同
	后置条件	更新一个po
UserDataService.init	语法	public void init() thr 
	前置条件	无
	后置条件	初始化持久化数据库
UserDataService.finish	语法	public void finish() t 
	前置条件	无
	后置条件	结束持久化数据库的
提供的服务（供接口）		
MovieDataService.find	语法	public MoviePO find 
	前置条件	无
	后置条件	按ID进行查找返回机
MovieDataService.insert	语法	public void insert(M 
	前置条件	同样ID的po在mapp
	后置条件	在数据库中增加一个

MovieDataService.delete	语法	
	前置条件	同样ID的po在map中
	后置条件	在数据库中增加一个po
	语法	public void delete(MoviePo po)
	前置条件	在数据库中存在相同ID的po
	后置条件	删除一个po
	语法	public void update(MoviePo po)
	前置条件	在数据库中存在相同ID的po
	后置条件	更新一个po
MovieDataService.init	语法	public void init() throws IOException
	前置条件	无
	后置条件	初始化持久化数据库
MovieDataService.finish	语法	public void finish() throws IOException
	前置条件	无
	后置条件	结束持久化数据库的初始化
提供的服务（供接口）		
TicketDataService.find	语法	public TicketPO find(String id)
	前置条件	无
	后置条件	按ID进行查找返回机票
TicketDataService.insert	语法	public void insert(TicketPO po)
	前置条件	同样ID的po在map中
	后置条件	在数据库中增加一个po
TicketDataService.delete	语法	public void delete(TicketPO po)
	前置条件	在数据库中存在相同ID的po



TicketDataService.update	后置条件	删除一个po
	语法	public void update( 
	前置条件	在数据库中存在相同
TicketDataService.init	后置条件	更新一个po
	语法	public void init() thr
	前置条件	无
TicketDataService.finish	后置条件	初始化持久化数据库
	语法	public void finish() t
	前置条件	无
提供的服务（供接口）	后置条件	结束持久化数据库的
	语法	
	前置条件	
SADataService.find	后置条件	按ID进行查找返回机
	语法	public SAPO find(lo
	前置条件	无
SADataService.insert	后置条件	在数据库中增加一个
	语法	public void insert(S
	前置条件	同样ID的po在mapp
SADataService.delete	后置条件	删除一个po
	语法	public void delete(S
	前置条件	在数据库中存在相同
SADataService.update	后置条件	更新一个po
	语法	public void update( 
	前置条件	在数据库中存在相同
SADataService.init	后置条件	
	语法	public void init() thr
	前置条件	

提供的服务（供接口）	SADataService.finish	前置条件	无
		后置条件	初始化持久化数据库
		语法	public void finish() 1
		前置条件	无
		后置条件	结束持久化数据库的
	VIPDataService.find	语法	public VIPPO find(lo
		前置条件	无
		后置条件	按ID进行查找返回机
	VIPDataService.insert	语法	public void insert(V
		前置条件	同样ID的po在mapp
		后置条件	在数据库中增加一个
	VIPDataService.delete	语法	public void delete(V
		前置条件	在数据库中存在相同
		后置条件	删除一个po
	VIPDataService.update	语法	public void update(
		前置条件	在数据库中存在相同
		后置条件	更新一个po
	VIPDataService.init	语法	public void init() thr
		前置条件	无
		后置条件	初始化持久化数据库
	VIPDataService.finish	语法	public void finish() 1
		前置条件	无
		后置条	结束持久化数据库的

AdminDataService与MovieDataService同上，略去

## 6.信息视角

系统的PO类就是相对应的相关的实体类，在此只做简单的介绍。

- UserPO类包含用户的用户名、密码属性。
- MoviePO类包含电影的名字、导演、演员、上映时间和想看人数。
- AdminPO类包含管理员的用户名、密码属性。

数据库表

- 数据库中包含Audience表，Movie表，Admin表