

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

2015-3-22

# 设计文档

CineplexSystem

Several thin, curved lines in dark blue and light gray originate from the bottom left corner and extend upwards and to the right.

管登荣

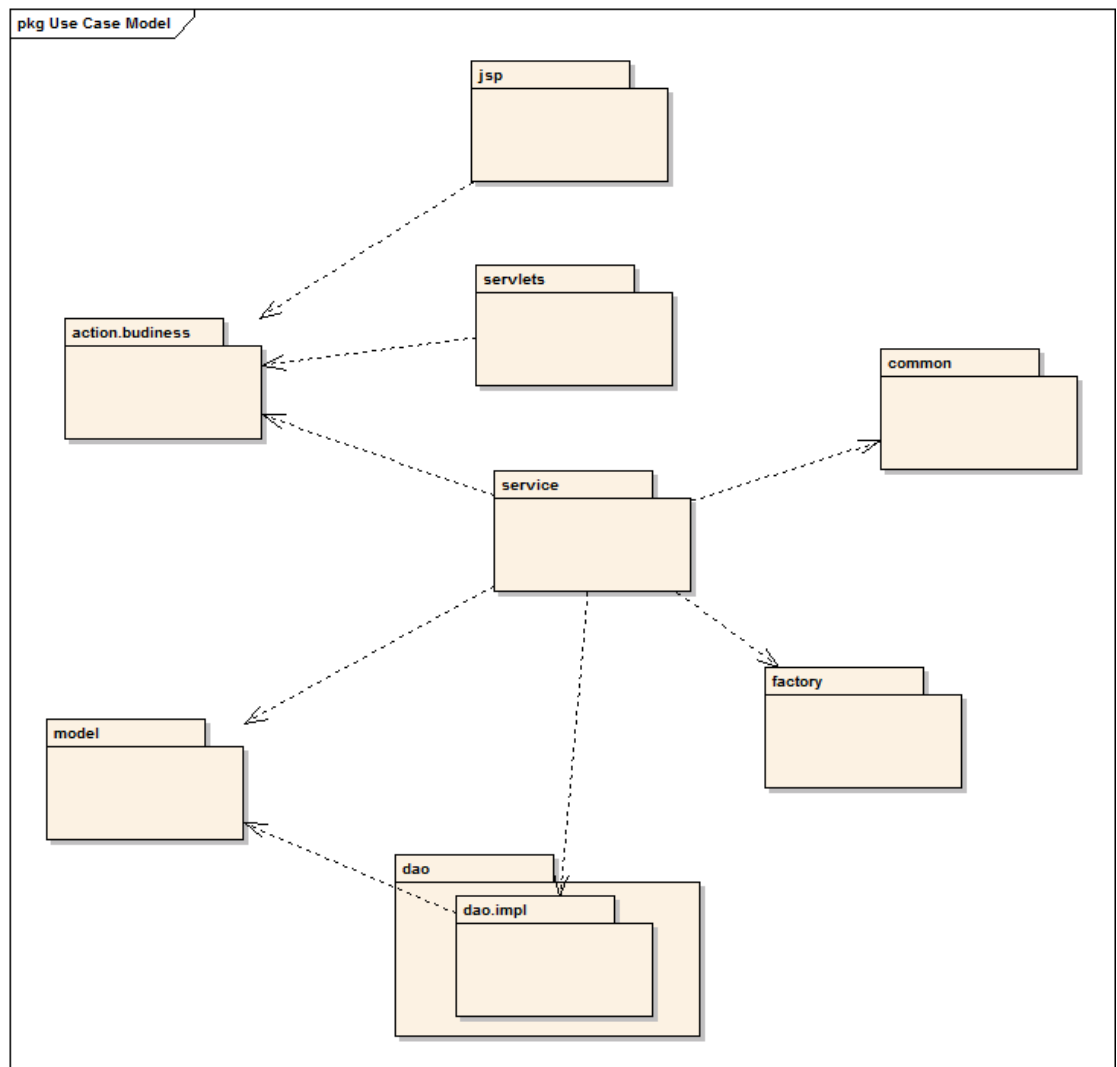
南京大学软件学院

架构设计.....	2
包图.....	2
类设计.....	3
1.common 包 .....	3
2.tag 包.....	3
3.servlets 包.....	3
4.Service 包 .....	3
5.model 包 .....	3
6.factory .....	3
7.enums 包.....	3
8.dao 包 .....	4
9.action.business 包.....	4
数据库设计: .....	5
ER 图: .....	5
数据表: .....	5
account: 银行账户表.....	5
activity: 发布的活动表.....	6
admin: 管理员, 服务员, 经理的账号表 .....	6
choice: 记录会员参加活动答题时的选择 .....	6
hall:记录所有的电影播放厅.....	6
member:记录会员 .....	7
movie:纪录电影.....	7
option:活动题目的答案选项 .....	7
plan: 放映计划表.....	7
purchase: 会员的购买记录.....	8
recharge:充值记录 .....	8

# 架构设计

采用 web 版 MVC 结构，servlets 包相当于 controller 层，jsp 中的页面相当于 view 层，model 层是具体业务逻辑以及数据库逻辑处理层，也就是 service 包（业务逻辑处理），dao 包（基本的底层数据库操作）。tag 包是自定义标签库的逻辑操作，model 包相当于 po，action.business 包相当于 vo，每次请求的大概处理流程为 servlet 接收到请求，调用相应的 service（业务逻辑处理，返回界面所需的封装好的数据（action.business）），service 调用具体的底层数据库操作(dao,返回封装好的 model/po 数据)，最后 servlet 将得到的数据通过绑定在 request 中重定向到 jsp 页面，jsp 页面显示 vo 中的数据。

## 包图



# 类设计

## 1.common 包

一些在不同项目中都通用的类，其中，**ACFilter** 类负责过滤请求，对没有访问权限的请求定位到对应的登录/注册页面；**ContextListener** 监听服务器的打开，并在打开的时候开启线程进行对会员的状态（费用不足，暂停，停止）进行检查；**EncodingFilter** 负责编码的检查，根据配置信息设置 **Request** 对象的编码信息；**HibernateUtil** 类负责 **sessionFactory** 的创建以及 **session** 的管理。

## 2.tag 包

自定义的标签库，其中，**NavSelectHandler** 类负责输出会员的导航条，由于会员的导航条会根据会员的状态发生变化，涉及后台的处理所以使用该类进行适当的逻辑处理；**HourSelectHandler** 类负责输出时间的小时选择框，**MinuteSelectHandler** 类负责输出时间的分钟选择框，**ActivateSelectHandler** 类负责根据会员的激活状态输出激活页面。

## 3.servlets 包

包含所有的 url 请求处理，每个类都是从请求中读取数据再调用 **service** 接口取得前台需要的封装好的数据，然后将数据封装到 **request** 请求中，**forward** 到目标页面。

## 4.Service 包

包含业务逻辑接口以及实现，通过调用 **dao** 中基本的数据库操作获取到需要处理的数据，然后根据业务要求进行逻辑处理并将处理结果封装好返回给 **servlet**

## 5.model 包

是与数据库中表对应的实体类。

## 6.factory

只有一个类 **DaoFactory**，负责将 **Dao** 实例返回给 **Service**

## 7.enums 包

有两个类，分别是 **MemberState**，用来表示会员状态（正常，欠费，暂停，停止），**MoviePlayState**，

用来表示电影的播放状态（尚未播放，正在放映，放映结束）。

## 8.dao 包

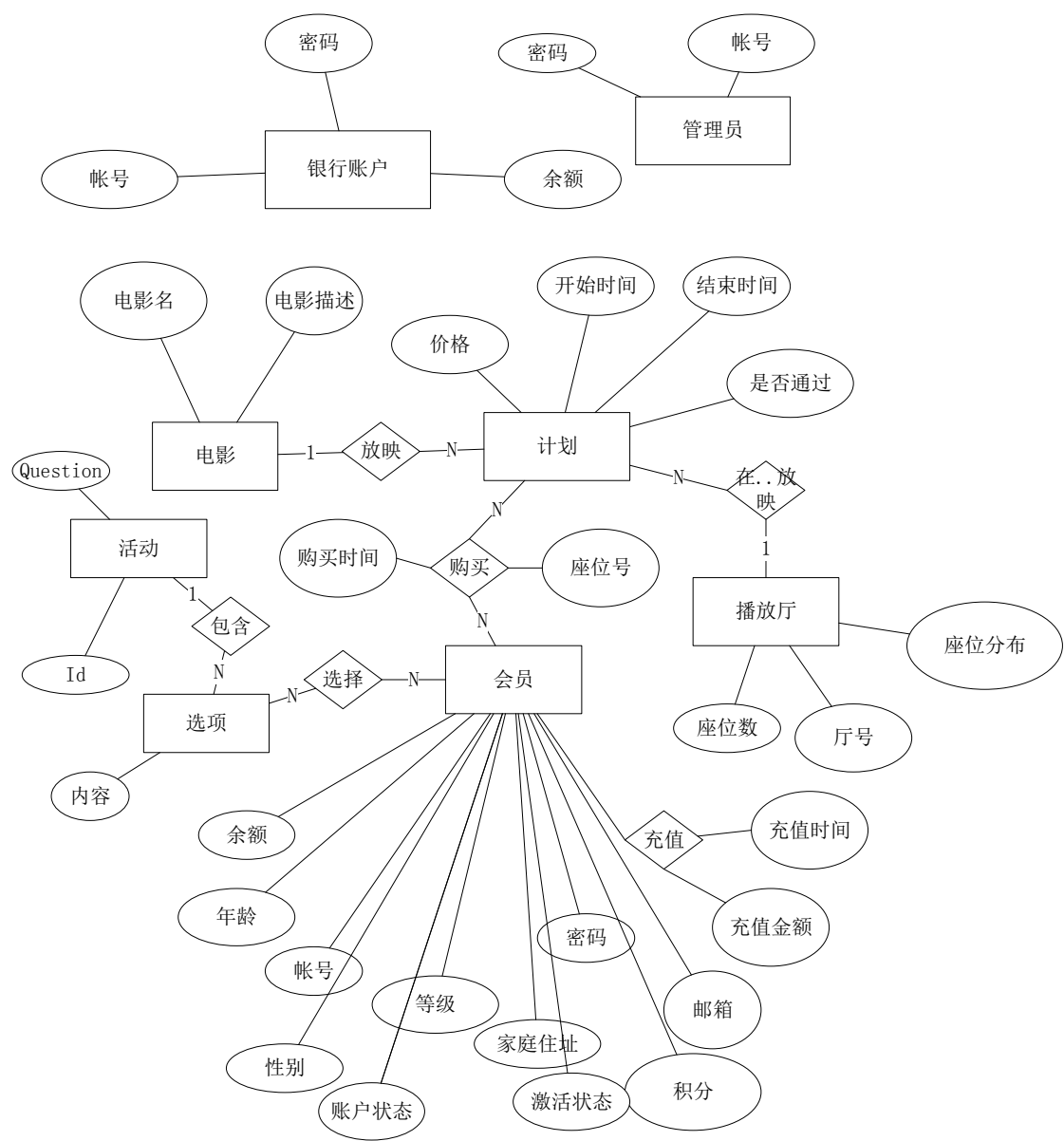
针对数据库表的操作，其中 BaseDao 中包含一些其他 Dao 可以通用的方法，比如 save, delect, update, load 方法。

## 9.action.business 包

是界面需要的封装好的数据类，相当于 vo。

# 数据库设计：

ER 图：



数据表：

account: 银行账户表

字段	解释	类型
id	账户名（主键）	varchar

password	密码	varchar
money	余额	double

### activity: 发布的活动表

字段	解释	类型
id	活动 id (自增, 主键)	int
planId	对应的放映计划 Id	int
question	问题内容	varchar

### admin: 管理员, 服务员, 经理的账号表

字段	解释	类型
id	账号 (服务员帐号以 'w' 开头, 经理帐号以 'm' 开头)	varchar
password	密码	varchar

### choice: 记录会员参加活动答题时的选择

字段	解释	类型
id	id (自增, 主键)	int
memberId	会员 Id	int
optionId	选项 Id	int

### hall: 记录所有的电影播放厅

字段	解释	类型
id	厅号 (主键)	int
seatNum	总的座位数量	int
seatPlan	座位分布 (即第一排多少座位, 第二排多少座位等, 中间用;隔开)	varchar

## member:记录会员

字段	解释	类型
id	账号（主键，自增，在匹配用户名时不够的地方补 0）	int
password	密码	varchar
money	账户余额	double
activated	是否激活（默认未激活‘0’,激活为‘1’）	char
point	积分（默认为 0）	int
level	等级(默认为 0)	char
email	邮箱	varchar
alive	账号是否存活(默认是‘1’,存活)	char
age	年龄	int
sex	性别（默认是 ‘男’）	char
home	家庭住址	varchar

## movie:纪录电影

字段	解释	类型
id	电影 Id（主键）	int
description	电影描述	varchar
name	电影名	varchar
endTime	下线时间	datetime

## option:活动题目的答案选项

字段	解释	类型
id	选项 Id（主键）	int
activityId	对应的活动题目 id	int
content	选项内容	varchar

## plan: 放映计划表

字段	解释	类型
id	计划 Id（主键）	int
hallId	放映厅 id	int



<b>movieId</b>	影片 Id	int
<b>startTime</b>	开始时间	datetime
<b>endTime</b>	结束时间	datetime
<b>price</b>	价格	double
<b>approved</b>	是否被经理同意	char

## **purchase:** 会员的购买记录

字段	解释	类型
<b>id</b>	记录 Id（主键）	int
<b>memberId</b>	会员 id	int
<b>planId</b>	购买的计划 Id	int
<b>seatId</b>	座位号 Id	int
<b>time</b>	购买时间	datetime

## **recharge:**充值记录

字段	解释	类型
<b>id</b>	记录 Id（主键）	int
<b>memberId</b>	会员 id	int
<b>money</b>	充值金额	double
<b>time</b>	购买时间	datetime