Movie booking System

软件架构文档

版本 **<1.0>**

修订文档历史记录

日期	版本	说明	作者
2017/5/20	1.0	初始版本	XipengLin

目录

软件架构文档

1. 简介

1.1 目的

本文档将从架构方面对系统进行综合概述，其中会使用多种不同的架构视图来描述系统的各个方面。它用于记录并表述已对系统的架构方面作出的重要决策。

1.2 范围

本文档用于 Legendary 小组正在开发中的时光电影订票系统。时光电影订票系统是为用户提供电影订票业务的 Web 应用系统，将提供电影票销售、智能选座、智能匹配以及信息反馈等功能。

1.3 定义、首字母缩写词和缩略语

见系统术语表

1.4 参考资料

1. 系统术语表，1.0 版，legendary 小组
2. 系统前景文档，1.1 版，legendary 小组
3. 系统软件需求规约，1.0 版，legendary 小组
4. 系统软件开发计划，1.1 版，legendary 小组
5. 系统初始迭代计划，1.1 版，legendary 小组
6. 系统细化迭代计划，1.0 版，legendary 小组
7. 系统风险列表，1.0 版，legendary 小组
8. RUP 的软件架构文档模板

2. 架构表示方式

本文档将通过以下一系列视图来表示时光电影订票系统的软件架构：用例视图、逻辑视图、部署视图。本文档不包括进程视图和实施视图。这些视图都是通过 PowerDesigner 工具建立的 UML 模型。

3. 架构目标和约束

1. 系统在开发过程中有如下设计约束：开发语言为 Java，采用关系型数据库存放数据，采用基于 UML 的面向对象分析与设计方法进行开发，采用 B/S 架构。
2. 系统应支持 100 人以上同时访问服务器并支持 500 人以上同时访问数据库，服务器的响应时间不应该超过 5 秒。
3. 所有用户在保证网络连接的情况下可同时通过局域网和互联网访问系统。
4. 系统必须保证数据的安全访问，用户需要通过用户名和密码进行身份认证，同时对数据的访问要进行授权认证。

4. 用例视图

本章是对软件架构的用例视图的描述。由于系统的用例数量太多，因此本章只选了部分与架构设计相关的用例。对于其余的用例，可参考软件需求规约。

选取的用例包括：

- 注册申请
- 用户注册审核
- 用户角色管理
- 角色权限管理
- 电影信息管理
- 座位信息管理

4.1 主要用例

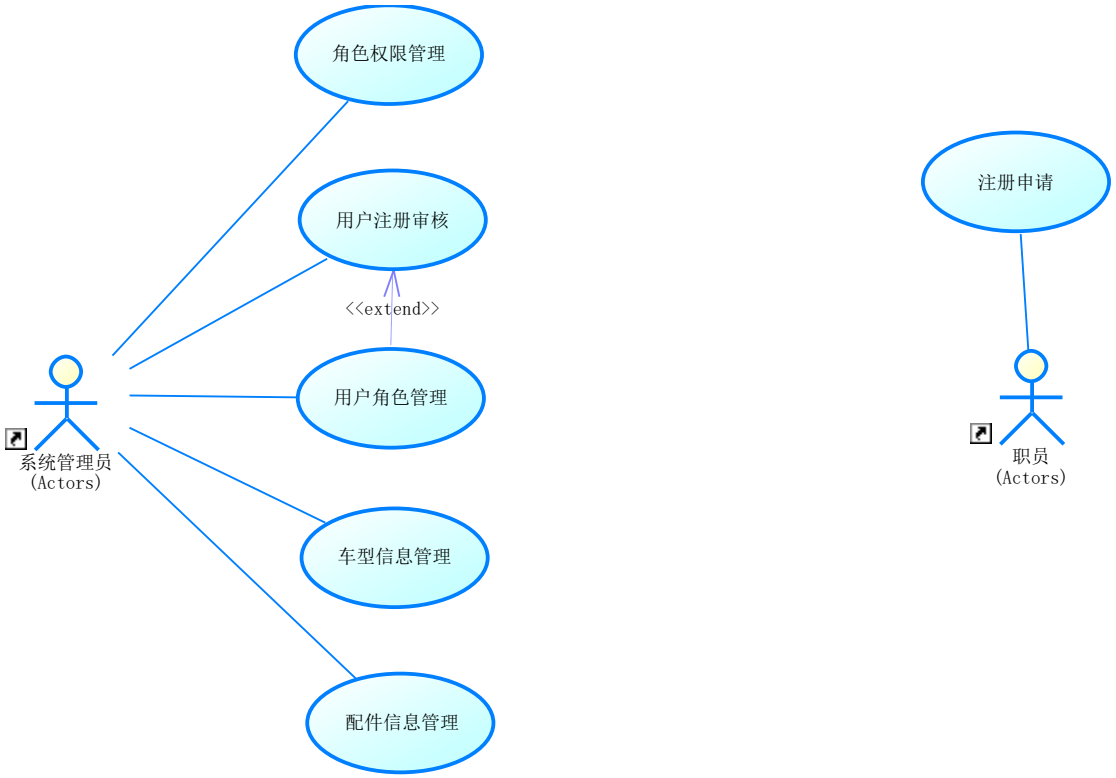

图 1 系统管理用例图

4.1.1 申请注册

简要说明：该用例允许影院员工提出注册申请，从而成为我们的系统用户。该用例的用户为大学生群体。

4.1.2 用户注册审核

简要说明：该用例允许系统管理员对系统的注册申请进行审核。决定其是否成为系统用户。该用例的主要参与者是系统管理员。

4.1.3 用户角色管理

简要说明：该用例允许系统管理员管理系统中所有用户的角色，它包括已分配角色用户的角色修改。该用例的主要参与者为系统管理员。

4.1.4 角色权限管理

简要说明：该用例允许系统管理员管理系统中各个职员角色的权限。它包括角色的创建，查询，删除。对各个角色进行权限的添加与移除。该用例的主要参与者是系统的管理员。

4.1.5 电影信息管理

简要说明：该用例允许系统管理员管理电影院参与销售的所有影票。它包括电影票信息的查询、创建、删除和修改。该用例的主要参与者是系统的管理员。

4.1.6 座位信息管理

简要说明：该用例允许系统管理员管理电影院参与座位分配与推荐的信息。它包括座位信息的查询、创建、删除和修改。该用例的主要参与者是系统的管理员。

5. 逻辑视图

本章是对软件架构的逻辑视图的描述。主要内容包括描述重要的类，类的分包，子系统以及子系统的分层等。另外还包括了一些重要用例的实现。

5.1 概述

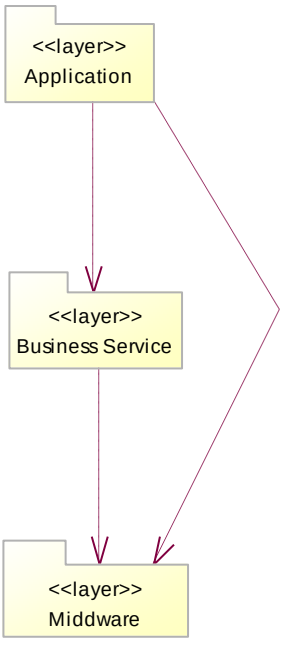

图 2 4In1 系统总体架构图

系统的逻辑视图主要由三层组成，分别是 Application 层、Business Service 层和 Middle 层。

Application 层主要由响应各种用户界面请求的动作类组成，它会调用 Business Service 层中的函数进行业务逻辑处理，同时根据结果显示不同的界面给用户。

Business Service 层主要完成实际的业务逻辑，同时包括与数据库的表对应的实体类，以及访问数据库的 DAO 类。

Middle 层为 SSH2 框架的函数库。

5.2 Application 层

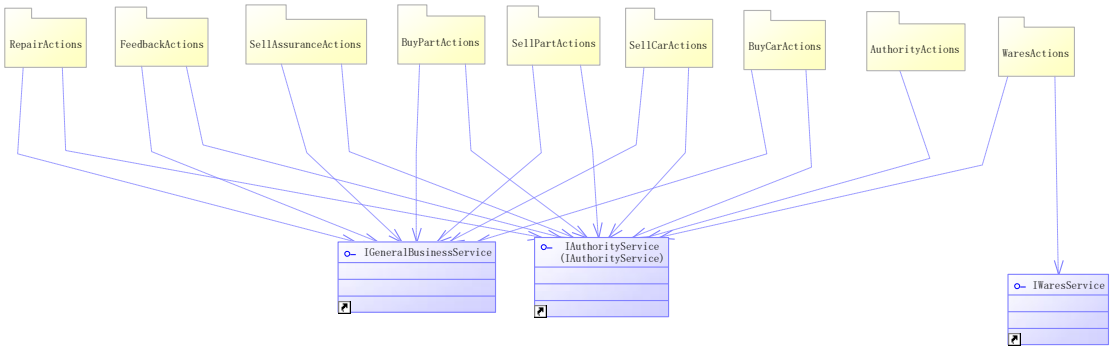

图 3 Application 层架构图

Application 层主要由 9 个子包组成，这 9 个子包分别是 RepairActions 包、FeedbackActions 包、SellAssuranceActions 包、BuyPartActions 包、SellPartActions 包、SellCarActions 包、BuyCarActions 包、AuthorityActions 包和 WaresActions 包。

5.3 Business Service 层

Business Service 层包括 Service 和 Model 两个包。

5.3.1 Service 包

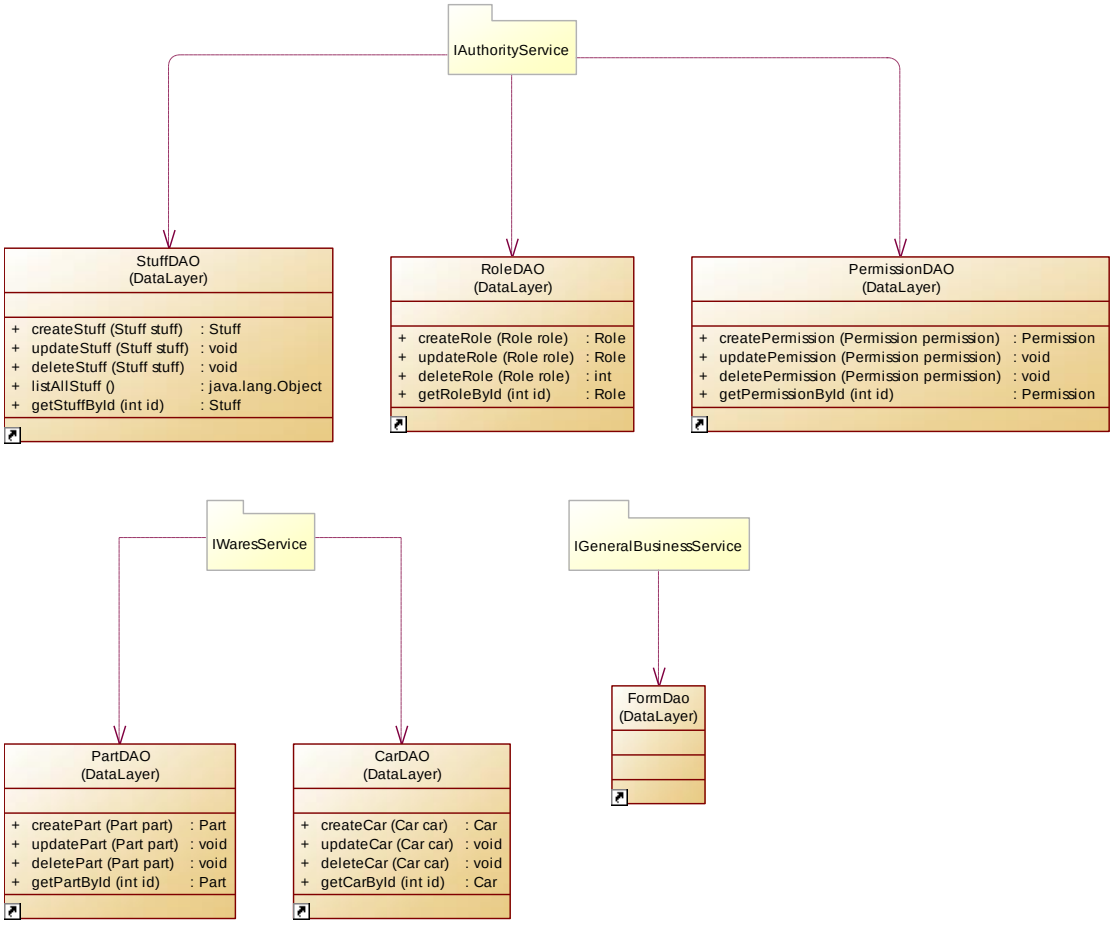

图 4 Service 包架构图

Service 包主要由三个包组成，IAuthorityService 包负责处理用户的身份认证以及角色和权限管理，IGeneralBusinessService 包负责整车销售、配件销售、采购、售后服务以及信息反馈过程中的表单处理，IWaresService 包负责车型信息、配件信息的管理以及库存信息。

5.3.2 Model 包

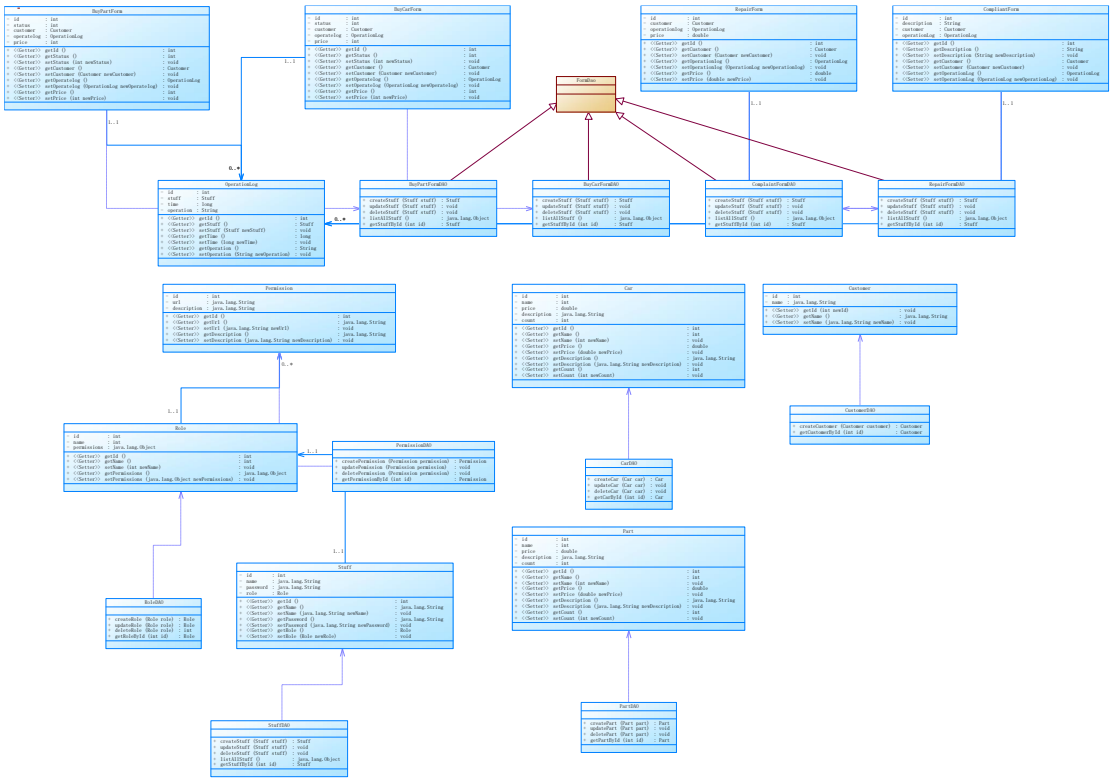

图 5 Model 包架构图

Model 包由 4 个表单的实体类以及对应的 4 个表单 DAO 类，与角色和身份相关的一些类：Permission 类、Role 类、Stuff 类、PermissionDAO 类、RoleDAO 类和 StuffDAO 类，Car 类、CarDAO 类、Part 类、PartDAO 类、Customer 类和 CustomerDAO 类等组成。

5.4 Middleware 层

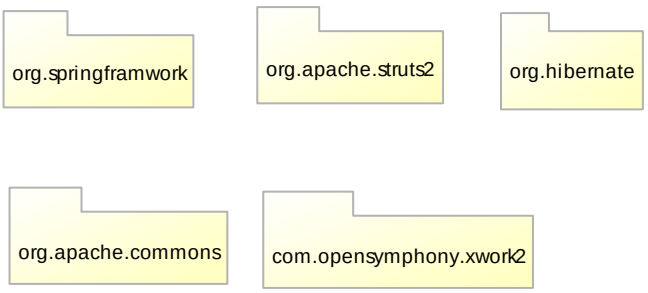

图 6 Middleware 层架构图

6. 部署视图

本章描述了部署和运行软件的物理网络（硬件）配置。

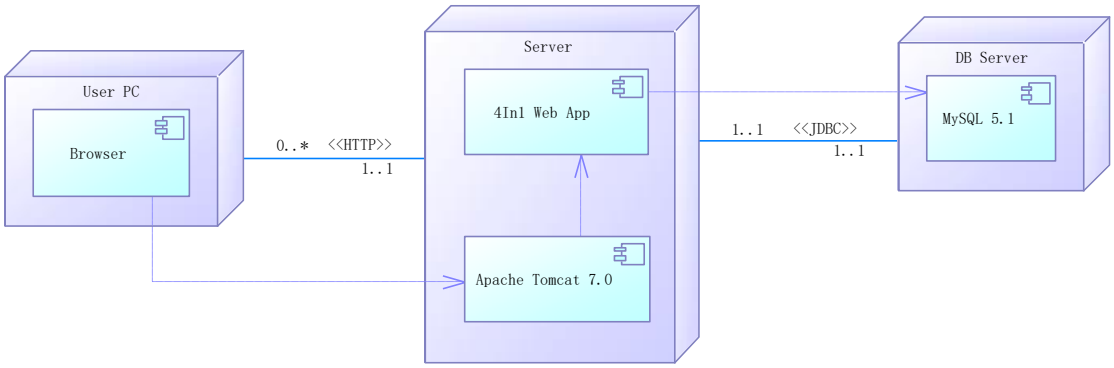

图 6 4In1 系统部署图

6.1 User Client

用户主要通过浏览器来访问系统，支持的浏览器包括 IE 6+和 Firefox 3.6+，客户端与服务端的连接可以是局域网或互联网。

6.2 Server

应用服务器运行系统，系统部署在 Tomcat7.0 容器上，它与数据服务器通过 jdbc 连接。

6.3 DB Server

数据服务器运行 mysql5.1 数据库。

7. 数据视图

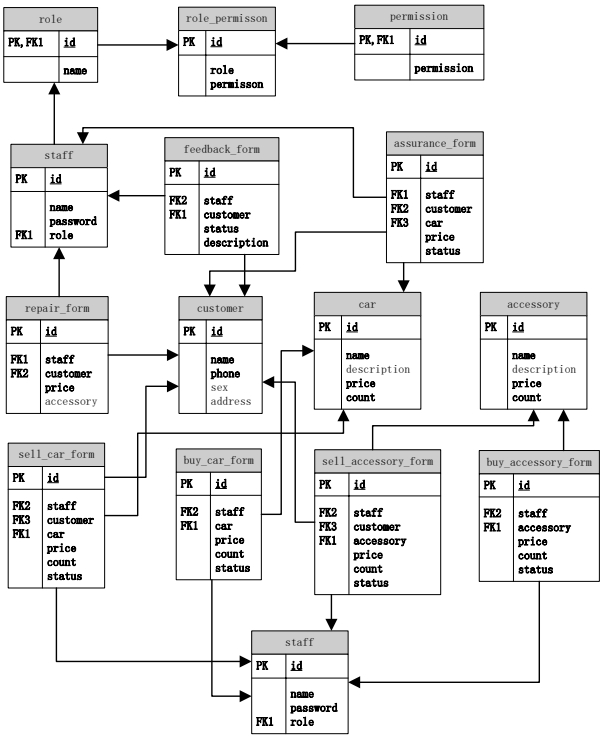

图 7 数据模型 ER 图

8. 大小和性能

本系统采用的软件架构可以很好的支持如下性能需求：

1. 系统应支持 100 人以上同时访问服务器并支持 500 人以上同时访问数据库。
2. 服务器的响应时间不应该超过 5 秒。

9. 质量

本系统采用的软件架构可以很好的支持系统质量方面的需求：

1. 系统应当方便所有用户的使用，对于有基础计算机水平的用户的培训时间应不超过 2 小时。
2. 系统应该提供在线的支持帮助。
3. 系统必须能够保证每天 24 小时不间断运行，可用率为 99%。
4. 合理的设计系统的结构以保证较高的可维护性，系统的模块应该可替换。
5. 系统应当正确处理发生的异常或者错误，并返回错误信息。