智能有声书

软件架构文档

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <日/月/年> | <x.x> | <详细信息> | <姓名> |
| 2019/07/04 | 1.1 |  | 徐轶喆 |
|  |  |  |  |
|  |  |  |  |

**目录**

第1级 1

第2级 2

第2级 3

第1级 4

第2级 5

第2级 6

**软件架构文档**

# 简介

## 目的

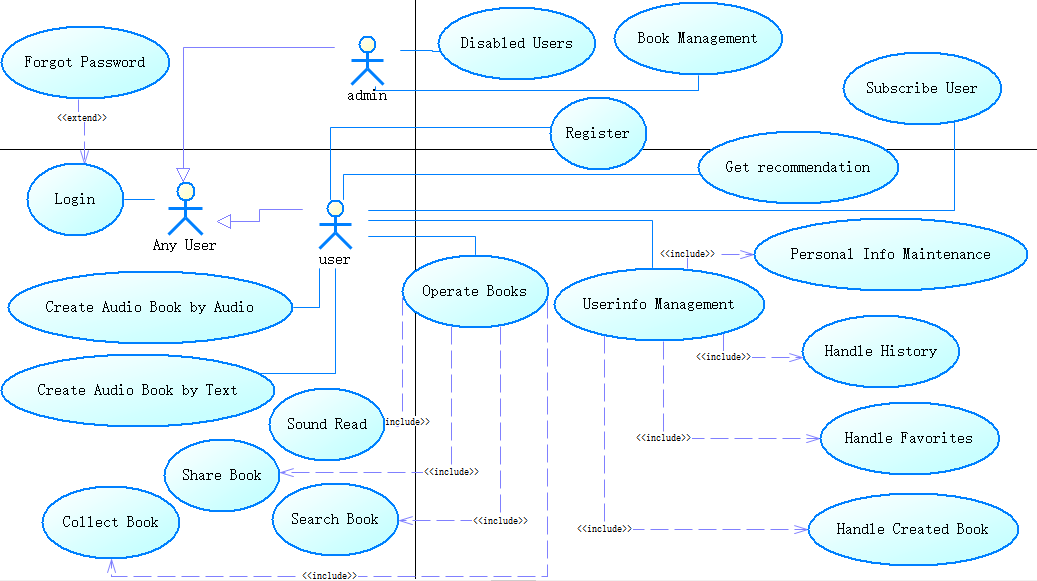
本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

架构设计定义了项目软件的全貌，记录了最重要的设计决策，并成为项目软件随后的详细设计与实现工作的战略指导原则。而在本文档中，我们主要使用视图来描述和归档软件的高低层次、表示软件架构的某个方面、展示软件系统的某些特性。

本文一共列举了个视图，分别用于展示小组项目在架构设计当中对架构设计的不同方面采用的不同策略。我们的项目设计从分析需求、设计用例，到设计具体逻辑架构，再到分进程、开发架构的方向进行设计，最后设计出项目软件的物理架构，生成具体的部署与实现视图。

## 参考资料

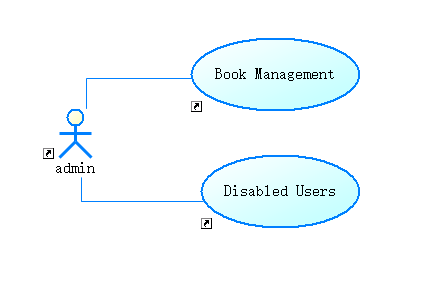
# 用例视图



完整的用例图 图(1)

我们小组将只能有声书的用例模型分为了6个包，每个包都有一张具体用例图，分别代表了不同的场景。下面的内容将会一一列举这些场景和用例。

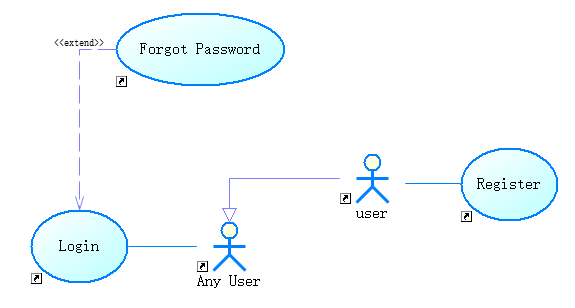
**场景1: Admin Management**



包Admin Management用例图 图(2)

Admin Management用例描述了管理员用户想要禁用、解禁一位普通用户，或管理有声书的场景和需求。

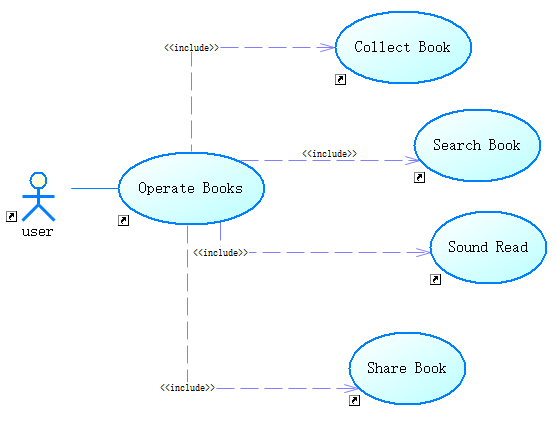
**场景2: User Recognition**



包 User Recognition用例图 图(3)

User Recognition用例描述了任意用户想要登录有声书系统(包括普通用户系统和管理员系统)，或普通用户想要注册普通用户系统账号的场景和需求。

**场景3: User Book Management**

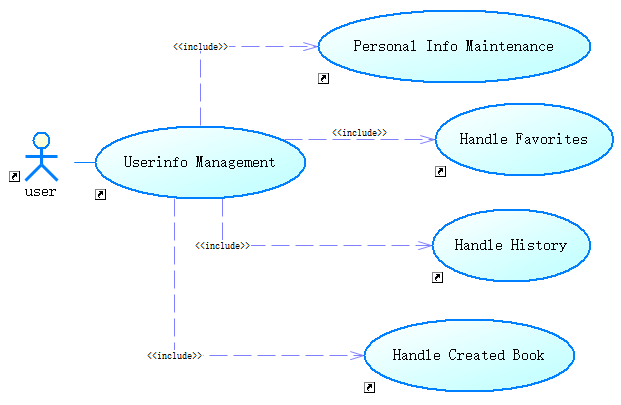


包 User Book Management用例图 图(4)

User Book Management用例描述了普通用户想要对有声书进行操作的场景和需求。

普通用户对有声书的操作包括了4个子用例，分别是收藏有声书、搜索有声书、有声书阅读、分享有声书。

**场景4: User Information Management**

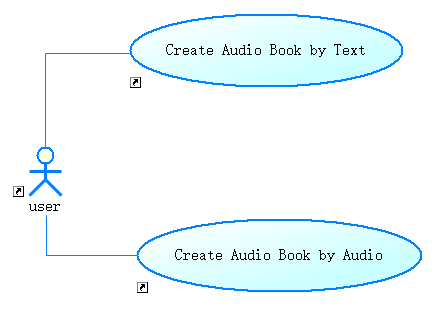


包 User Information Management用例图 图(5)

User Information Management用例描述了普通用户想要管理个人信息的场景和需求。

普通用户管理个人信息包括4个子用例，分别是修改个人信息、管理收藏夹、管理阅读历史记录、管理创作的有声书。

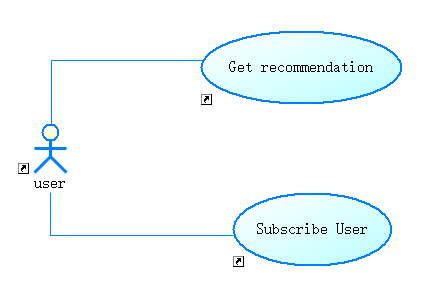
**场景5: User Book Creation**



包 User Book Creation用例图 图(6)

User Book Creation用例描述了普通用户想要通过文本文件、或音频文件创作有声书的场景和需求。

**场景6: User Transaction**



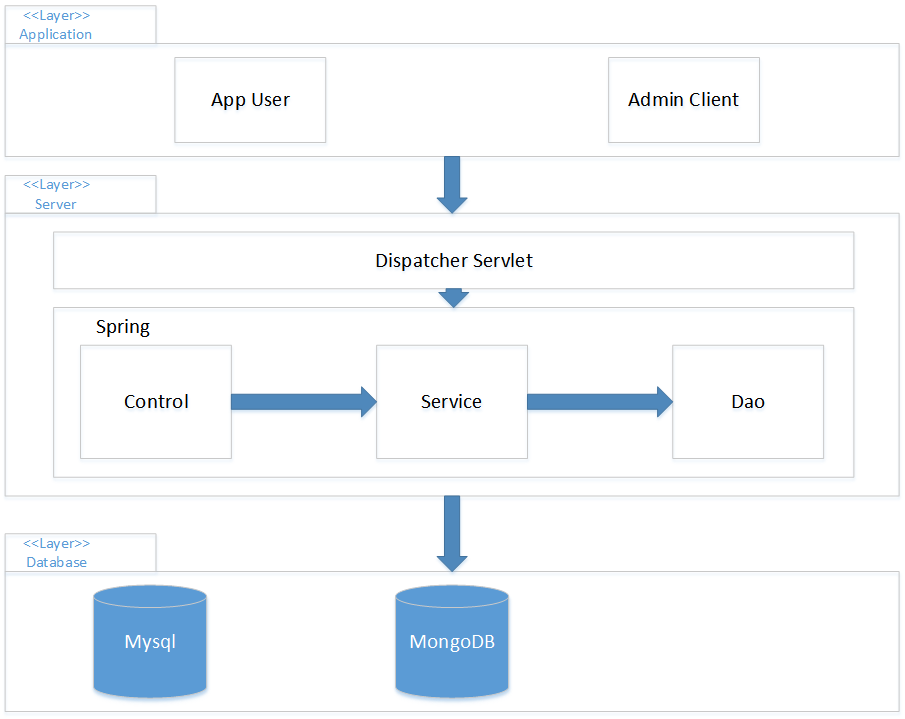
包 User Transaction用例图 图(7)

User Transaction用例描述了普通用户获取推荐有声书、关注其他用户的场景和需求。

# 逻辑视图

## 概述

逻辑架构设计方面我们采用三层架构风格。



逻辑架构图 图(8)

表示层包含普通用户App移动端和管理员系统客户端，向服务层发送请求。

服务层通过Dispatcher Servlet包响应客户端发送的请求，传递给Spring类包，经由控制类调用相应的服务类，再由服务类调用数据访问与缓存类，最终实现对数据层的访问。

## 在构架方面具有重要意义的设计包

* + 1. **Control Package**

Control包中包含了各种用于接收对应不同实体操作请求的控制类。下面将对于其中每个类的具体功能进行描述。

**User Controller:**

一个UserService类私有变量，注解为自动装配，用于调用Service包中UserService类，即对应于用户服务类的各个接口。

setRHeader函数，用于设置响应头。

handlelogin函数，用于接收普通用户的登录请求，并调用普通用户的登录服务。

handleregister函数，用于接收普通用户注册的请求，并调用普通用户注册服务。

handleactivation函数，用于接收普通用户邮箱验证激活账号的请求，并调用普通用户账号激活服务。

askcollector函数，用于接收普通用户查看收藏夹的请求，并调用普通用户查看收藏夹服务。

askhistory函数，用于接收普通用户查看阅读历史记录的请求，并调用普通用户查看阅读历史记录服务。

askcomment函数，用于接收普通用户查看评论记录的请求，并调用普通用户查看评论记录服务。

askworks函数，用于接收普通用户查看创建的有声书的请求，并调用普通用户查看创建的有声书服务。

* + 1. **Service Package**

Service包中包含了各种对应各个实体类的不同的服务类，用于调用并处理数据，返回给控制层响应body。下面将对于其中每个类的具体功能进行描述。

**User Service:**

一个UserDao私有变量，注解为自动装配，用于调用Dao包中UserDao类，即普通用户数据缓存类的各个接口。

logincheck函数，调用查询用户数据的数据访问层接口，返回信息是否通过查询。

register函数，调用插入注册用户数据的数据访问层接口，返回是否成功。

useractive函数，调用设置用户激活状态的数据访问层接口，返回是否成功。

askcollector函数，调用查询用户收藏夹的数据访问层接口，返回书籍列表。

askhistory函数，调用查询用户阅读历史的数据访问层接口，返回书籍列表。

askcomment函数，调用查询用户评论的数据访问层接口，返回评论实体列表。

askworks函数，调用查询用户创建有声书的数据访问层接口，返回书籍列表。

askuser函数，调用查询用户具体信息的数据访问层接口，返回用户具体信息。

setImg函数，调用插入用户头像的数据访问层接口，返回是否成功。

* + 1. **Dao Package**

Dao包中包含了各种对应各个实体类的数据访问和存储逻辑，并将结果返回给服务层。对数据库访问使用Mybatis框架，在包中，一个实体类对应一个Dao interface和一个XML配置文件。在interface定义了各种需要的实例，通过MAPPER XML自动生成。

# 进程视图

# 部署视图

# 实现视图