image201

**Iaecpedia**

**详细设计说明书**

|  |  |
| --- | --- |
| 参与人员： | 何徐昊 |
|  | 苏时元 |
|  | 修海锟 |
|  | 宋丹 |
|  | 刘润华 |

华南理工大学

二〇一六年九月

目录

目录 2

1 概述 3

1.1系统概述 3

1.2功能概述 4

1.3软件设计目标 4

1.4参考资料 4

1.4修订版本记录 5

修订历史记录 5

2 术语表 5

3 设计概述 6

3.1 系统的复用计划 6

3.2系统接口设计 6

3.2.1提供给用户的界面 6

3.2.2系统外部接口设计 16

3.2.3系统内部接口设计 16

3.3 对象模型设计 28

3.3 对象模型设计 28

3.4系统用例实现详细设计 32

3.4.1 动态模型设计 32

3.5系统非功能设计 50

3.5.1系统用户交互界面设计 50

3.5.2质量属性设计 51

性能 51

可用性 51

安全性 51

易用性 51

3.6 组件说明 52

3.6.1组件图 52

3.6.2 组件说明 52

3.7部署说明 52

3.7.1部署图 52

3.7.2部署方式 53

# 1 概述

## 1.1系统概述

在传统科普网站的科普模式中，通常是由科普信息的生产者(即相关领域的科学家、专家)通过一定的传播媒介(即网站)，向科普信息的消费者(即网民)发布、传播相关的科学技术知识。其模式运作的根本目的就是为了实现知识的线性传播。然而，随着强调学习、发现、探索过程观念的兴起，用户所希望获得的不仅仅是最后已经“成品”的知识，更渴望切实地体验全过程的“知识生产”流程。这种新模式下，每一个参与到百科建设过程中的用户既是信息的生产者，又是信息的消费者。不同的主体共同参与，改变了以往的单向度的科普传播模式，构建了一个网络化的公共空间。因此，我们项目组决定创建一个类似于互动百科那样的在线编纂型网络百科网站。用户可以实时在线与来自全球的互联网用户共同就感兴趣的内容进行创作、讨论、编辑和发布，形成一种多向交互式传播的模式。。

## 1.2功能概述

**互动百科(Iaecpedia)的功能如下：**

* 个人信息管理功能，用户注册登录注销管理个人信息等功能
* 词条管理功能。为用户提供发布词条、搜索词条、评论词条等功能
* 管理员功能。为管理员提供审核词条、删除词条等功能

## 1.3软件设计目标

Iacepedia系统能满足用户创作词条、浏览词条、查询词条、修改词条、讨论词条。并且努力做到可以容纳更多的用户的网页版。且主要内容由用户提供。

## 1.4参考资料

《Iacepedia 需求规格说明书》

《Iacepedia 架构设计说明书》

《Iacepedia 数据库设计说明书》

## 1.4修订版本记录

## 修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| **20161228** | **1.0** | **系统接口设计** | **刘润华、宋丹、苏时元、修海锟、** |
| **20161229** | **1.1** | **用例实现详细设计** | **刘润华、宋丹、苏时元、修海锟** |
| **20170101** | **1.2** | **系统非功能设计** | **何徐昊** |
| **20170104** | **2.0** | **最终修订** | **全体人员** |

# 2 术语表

详见需求文档。

# 3 设计概述

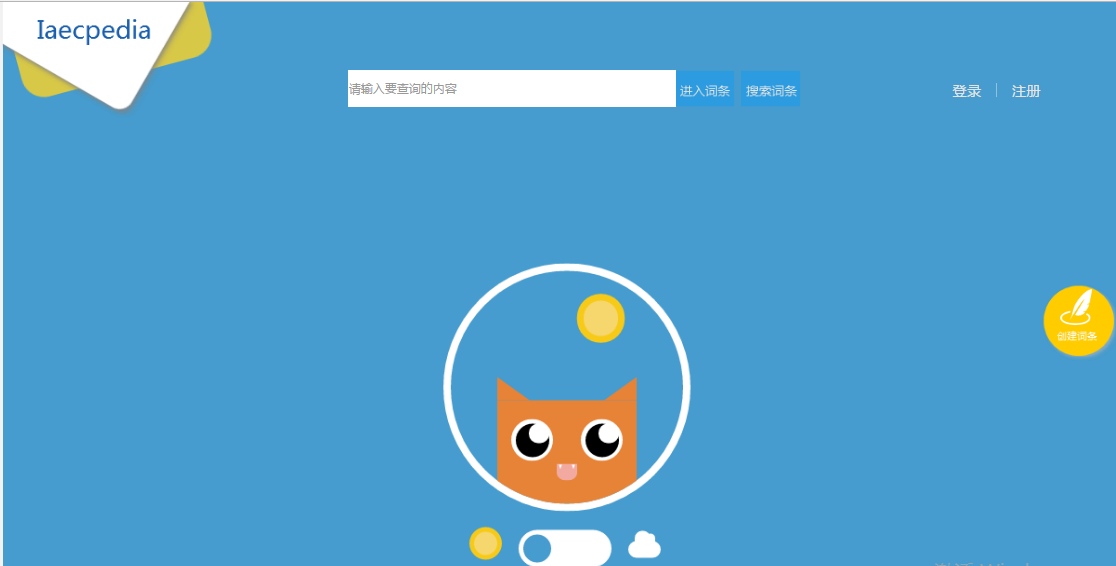
## 3.1 系统的复用计划

在我们的系统中，我们使用了SSM作为后端的基础架构，使用MySQL作为数据库，Redis作为缓存，更好的规划代码以及实现各模块的功能，在前端的实现中，我们采用了jQuery+CSS来百分百还原原型。

## 3.2系统接口设计

### 3.2.1提供给用户的界面

首页：



创建词条：







查看词条解释：



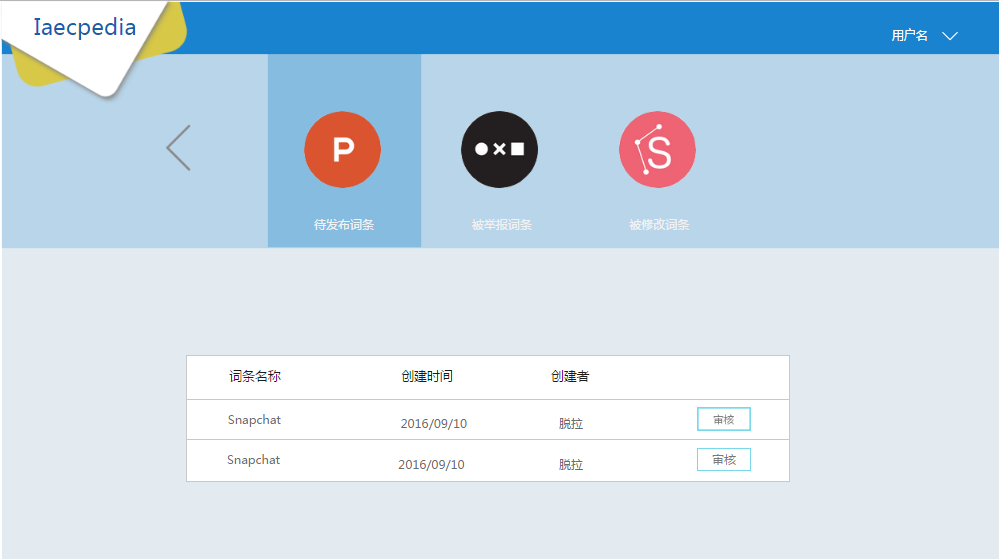
查看词条搜索列表：



个人主页：



管理员主页：



点击审核：



登录页面：



注册页面：



### 3.2.2系统外部接口设计

1、提供多个Dao接口，与数据库进行交互操作，例如增删查改。

2、提供多个Service接口，提供内部逻辑的转换与管理。

### 3.2.3系统内部接口设计

**硬件接口：**

1.支持一般X86、64位系统系列计算机。

2.一般性能的小型服务器。

**软件接口：**

1. 运行于Windows XP以上版本系统以及Linux系的系统，支持浏览器应用的系统。
2. 与MySQL数据库连接。
3. 于Redis缓存链接。

**系统接口：**

**1.用户注册**

1. 实现说明：

游客在本系统注册，成为普通用户。

1. 请求说明：

•请求方法 signup

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
| account | 用户名 | varchar | 是 |
| password | 密码 | varchar | 是 |
| username | 用户昵称 | varchar | 是 |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |
| data | 返回数据包装对象 | Map<String,Object> | 1 |  |
| info | 返回信息 | String | 2 | data |

**2.用户登录**

1. 实现说明

用户输入账号密码登录本系统。

1. 请求说明

•请求方法 login

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
| account | 用户名 | varchar | 是 |
| password | 密码 | varchar | 是 |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |
| data | 返回数据包装对象 | Map<String,Object> | 1 |  |
| info | 返回信息 | String | 2 | data |
| exp | 用户经验值 | int | 2 | data |
| iconaddr | 头像文件地址 | String | 2 | data |
| lastLoginTime | 上一次上线时间 | Date | 2 | data |
| level | 用户等级 | int | 2 | data |
| userName | 用户昵称 | String | 2 | data |

**3.用户注销**

1. 实现说明

用户退出本系统。

1. 请求说明

•请求方法 logout

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
|  |  |  |  |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |
| data | 返回数据包装对象 | Map<String,Object> | 1 |  |
| info | 返回信息 | String | 2 | data |

**4.个人主页**

1. 实现说明

普通用户进入个人主页，查看并管理自己发布的词条。

1. 请求说明

•请求方法 getPersonalHomePage

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
|  |  |  |  |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |
| data | 返回数据包装对象 | Map<String,Object> | 1 |  |
| info | 返回信息 | String | 2 | data |
| hasPassNum | 已通过版本数 | int | 2 | data |
| toPassNum | 已通过版本数 | int | 2 | data |
| hasNotPassNum | 未通过版本数 | int | 2 | data |
| hasPassList | 已通过版本列表 | List<Map<String,Object>> | 2 | data |
| toPassList | 待通过版本列表 | List<Map<String,Object>> | 2 | data |
| hasNotPassList | 未通过版本列表 | List<Map<String,Object>> | 2 | data |
| passRate | 通过率 | double | 2 | data |
| eid | 词条id | int | 3 | hasPassList |
| entryName | 词条名称 | String | 3 | hasPassList |
| createDate | 创建时间 | String | 3 | hasPassList |
| passDate | 通过时间 | String | 3 | hasPassList |
| modifyTimes | 被他人修改次数 | int | 3 | hasPassList |
| eid | 词条id | int | 3 | toPassList |
| entryName | 词条名称 | String | 3 | toPassList |
| createDate | 创建时间 | String | 3 | toPassList |
| eid | 词条id | int | 3 | hasNotPassList |
| entryName | 词条名称 | String | 3 | hasNotPassList |
| createDate | 创建时间 | String | 3 | hasNotPassList |
| refuseReason | 不通过原因 | String | 3 | hasNotPassList |

**5.创建词条**

1. 实现说明

用户创建一个新词条

1. 请求说明

•请求方法 createEntry

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
| entryName | 词条名称 | varchar | 是 |
| tag1 | 标签1 | varchar | 是 |
| tag2 | 标签2 | varchar | 是 |
| tag3 | 标签3 | varchar | 是 |
| tag4 | 标签4 | varchar | 是 |
| file | 图片文件 | varchar | 是 |
| detail | 词条描述 | varchar | 是 |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |
| data | 返回数据包装对象 | Map<String,Object> | 1 |  |
| info | 返回信息 | String | 2 | data |

**6.查找词条**

1. 实现说明

用户输入关键字来查找相关的词条。

1. 请求说明

•请求方法 queryEntry

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
| info | 查找关键字 | varchar | 是 |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |
| data | 返回数据包装对象 | Map<String,Object> | 1 |  |
| info | 返回信息 | String | 2 | data |
| listNum | 查询结果数 | int | 2 | data |
| list | 查询结果 | List<Map<String,String>> | 2 | data |

**7.进入词条**

1. 实现说明

用户进入一个词条，查看其详细内容。

1. 请求说明

•请求方法 enterEntry

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
| eid | 词条id | int | 是 |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |
| data | 返回数据包装对象 | Map<String,Object> | 1 |  |
| info | 返回信息 | String | 2 | data |
| entry | 词条对象 | Entry | 2 | data |
| labels | 词条标签 | List<Label> | 2 | data |
| comments | 词条评论 | List<Comment> | 2 | data |

**8.举报词条**

* 实现说明

用户输入一个理由来举报一个词条。

* 请求说明

•请求方法 submitReport

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
| eid | 词条id | int | 是 |
| report | 举报理由 | varchar | 是 |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |
| data | 返回数据包装对象 | Map<String,Object> | 1 |  |
| info | 返回信息 | String | 2 | data |

**9.删除词条**

1. 实现说明

普通用户删除一个自己发布的词条

1. 请求说明

•请求方法 deleteEntry

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
| eid | 词条id | int | 是 |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |

**10.点赞**

1. 实现说明

普通用户对一个词条进行点赞。

1. 请求说明

•请求方法praise

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
| eid | 词条id | int | 是 |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |
| data | 返回数据包装对象 | Map<String,Object> | 1 |  |
| info | 返回信息 | String | 2 | data |

**11.差评**

1. 实现说明

用户给一个词条差评。

1. 请求说明

•请求方法 badReview

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
| eid | 词条id | int | 是 |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |
| data | 返回数据包装对象 | Map<String,Object> | 1 |  |
| info | 返回信息 | String | 2 | data |

**12.评论**

* 实现说明

用户对一个词条发布评论。

* 请求说明

•请求方法 submitComment

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
| eid | 词条id | int | 是 |
| comment | 评论内容 | varchar | 是 |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |
| data | 返回数据包装对象 | Map<String,Object> | 1 |  |
| info | 返回信息 | String | 2 | data |

**13.查看待审核/已修改/被举报词条**

1. 实现说明

管理员查看并管理待审核的词条。

1. 请求说明

•请求方法 getAllData

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
|  |  |  |  |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |
| data | 返回数据包装对象 | Map<String,Object> | 1 |  |
| info | 返回信息 | String | 2 | data |
| userName | 管理员名字 | String | 2 | data |
| uncheckedEntryList | 待审核词条 | BaseEntryDataList | 2 | data |
| reportedEntryList | 被举报词条 | BaseEntryDataList | 2 | data |
| modifiedEntryList | 已修改词条 | BaseEntryDataList | 2 | data |

**14.审核待发布词条**

1. 实现说明

管理员管理待发布的词条，允许发布或者不通过。

1. 请求说明

•请求方法checkEntry

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
| eid | 词条id | int | 是 |
| allow | 是否审核通过 | bool | 是 |
| reason | 原因 | varchar | 是 |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |

**15.审核已修改词条**

1. 实现说明

管理员审核已经被修改过得的词条，允许发布或者不通过。

1. 请求说明

•请求方法checkModifiedEntry

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
| eid | 词条id | int | 是 |
| allow | 是否审核通过 | bool | 是 |
| reason | 原因 | varchar | 是 |

•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |

**16.审核被举报词条**

1. 实现说明

管理员管理被举报的词条，删除该词条或者驳回举报。

1. 请求说明

•请求方法checkReportedEntry

•请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 必须 |
| eid | 词条id | int | 是 |
| allow | 是否审核通过 | bool | 是 |
| reason | 原因 | varchar | 是 |

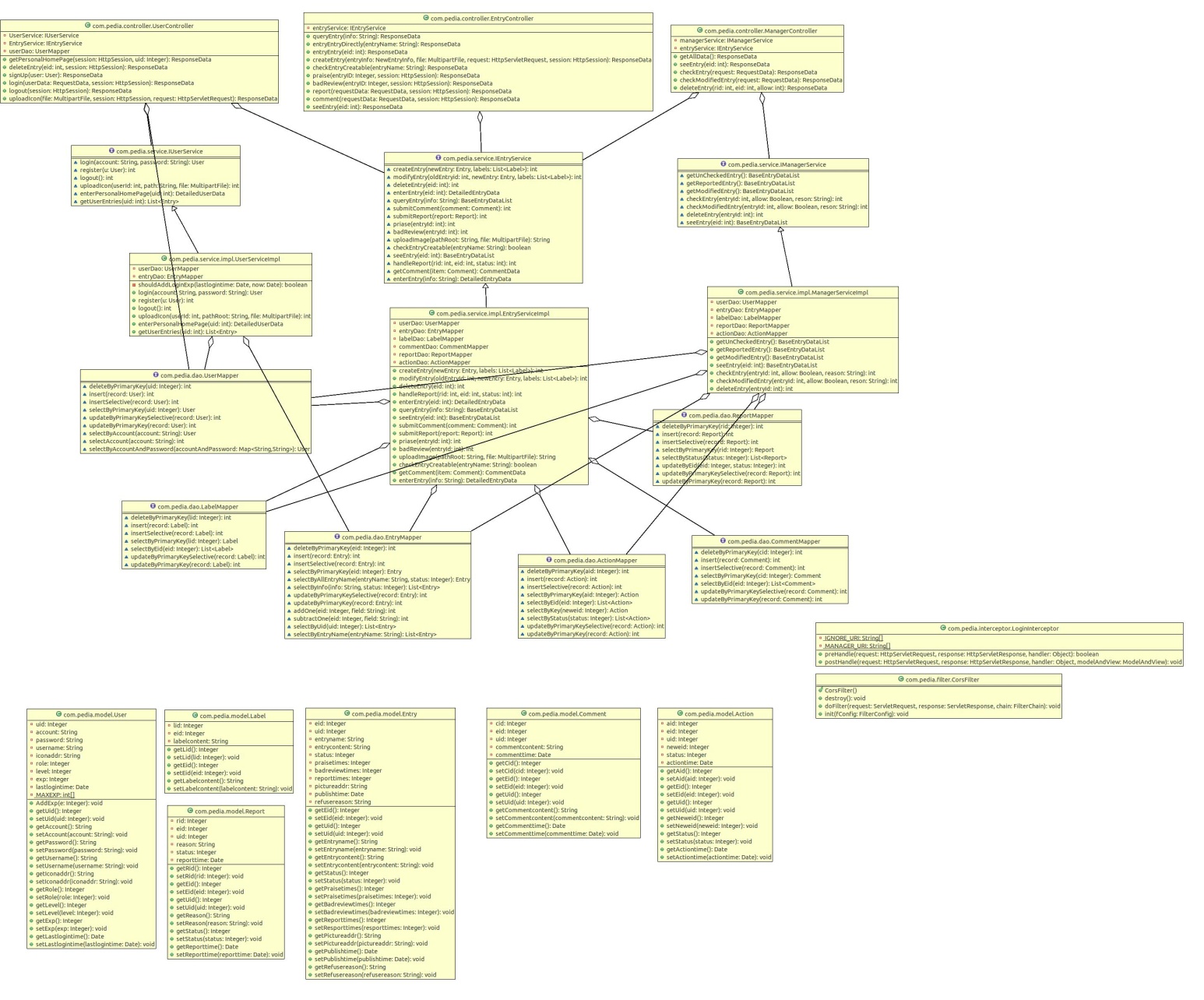
•返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 级别 | 父节点 |
| code | 返回值 | int | 1 |  |

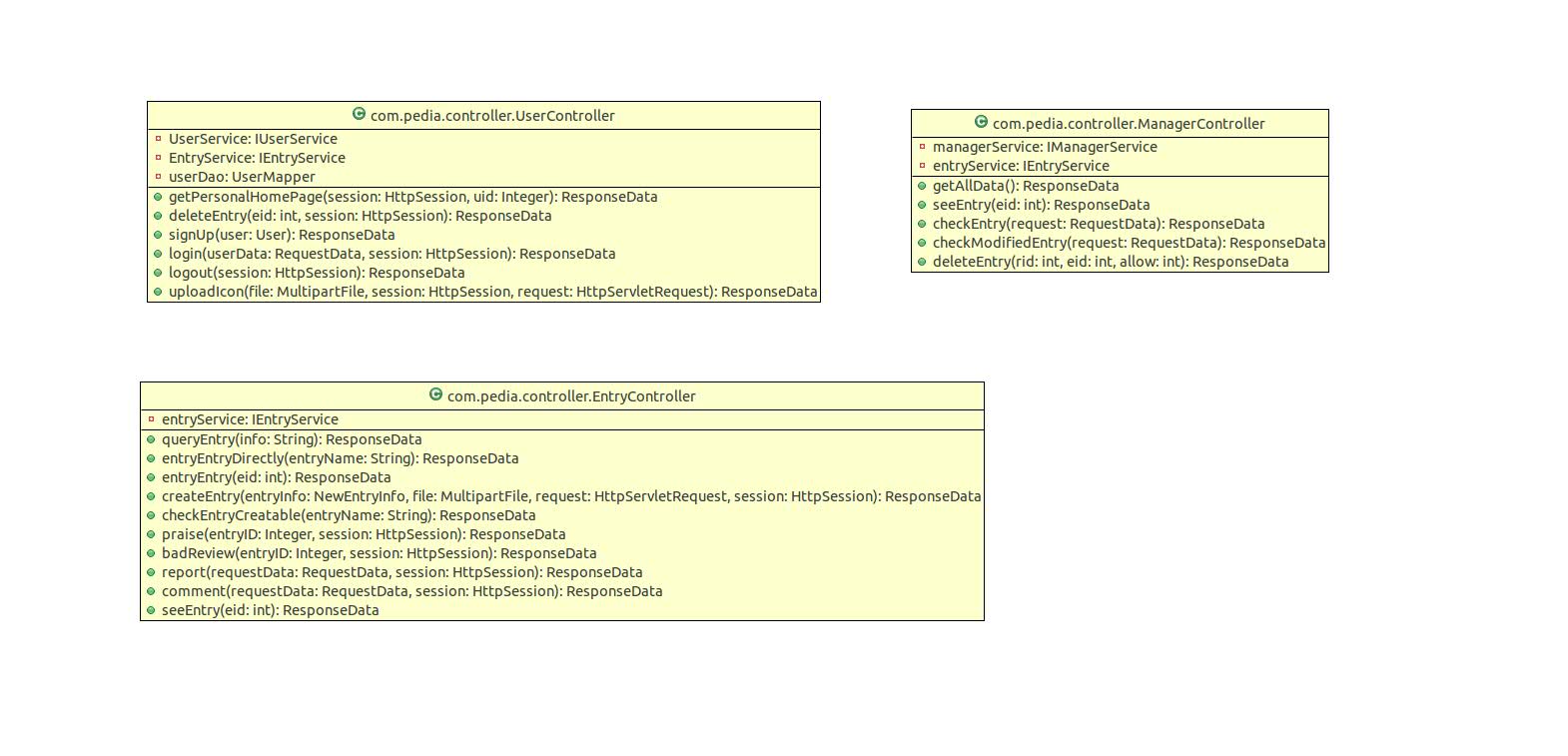
## 3.3 对象模型设计

## 3.3 对象模型设计

3.3.1 总体类图

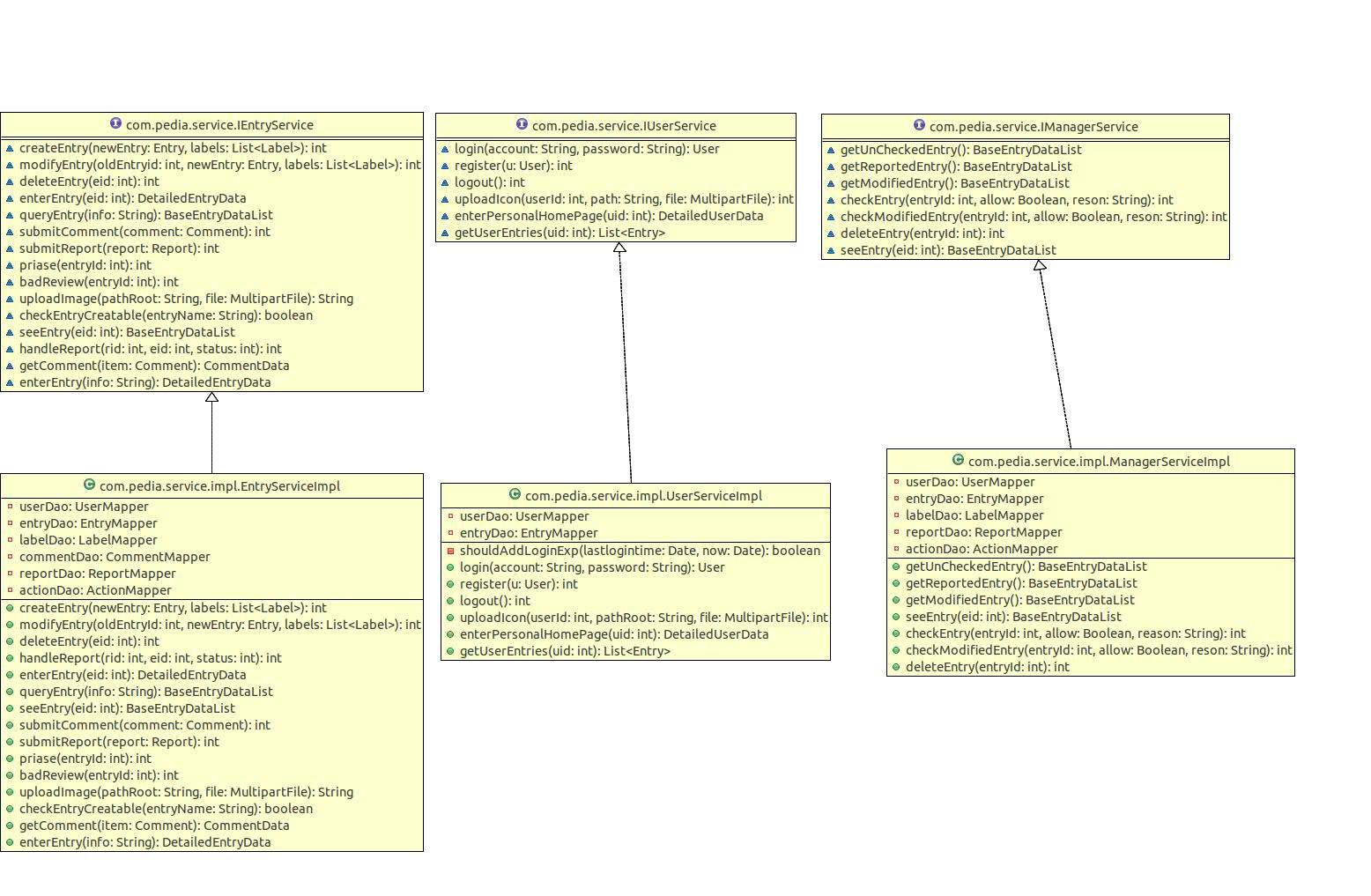


3.3.2 各个类图说明



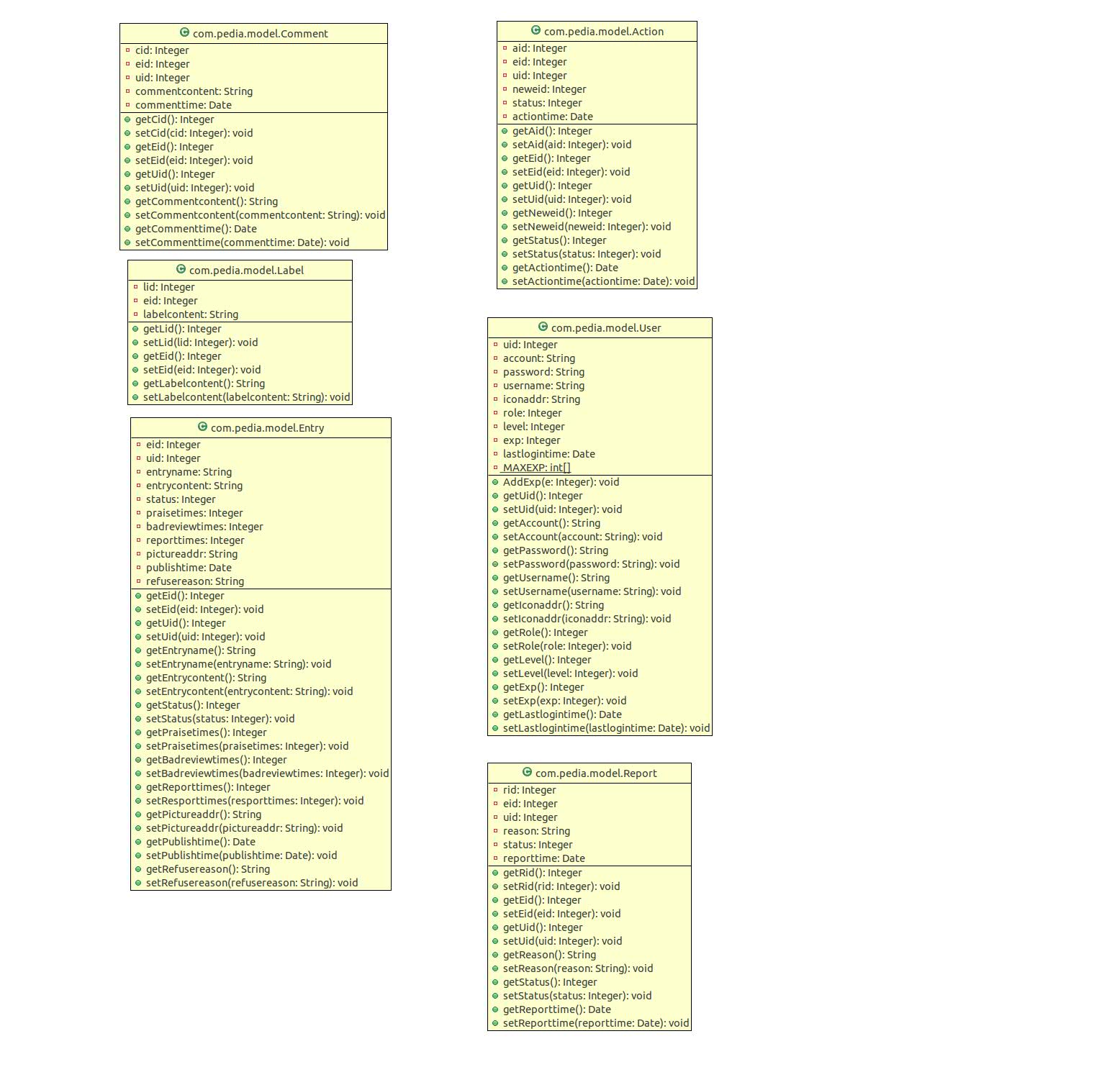
Controller：

基于 Springmvc 实现的控制器类，用于处理前端的请求。



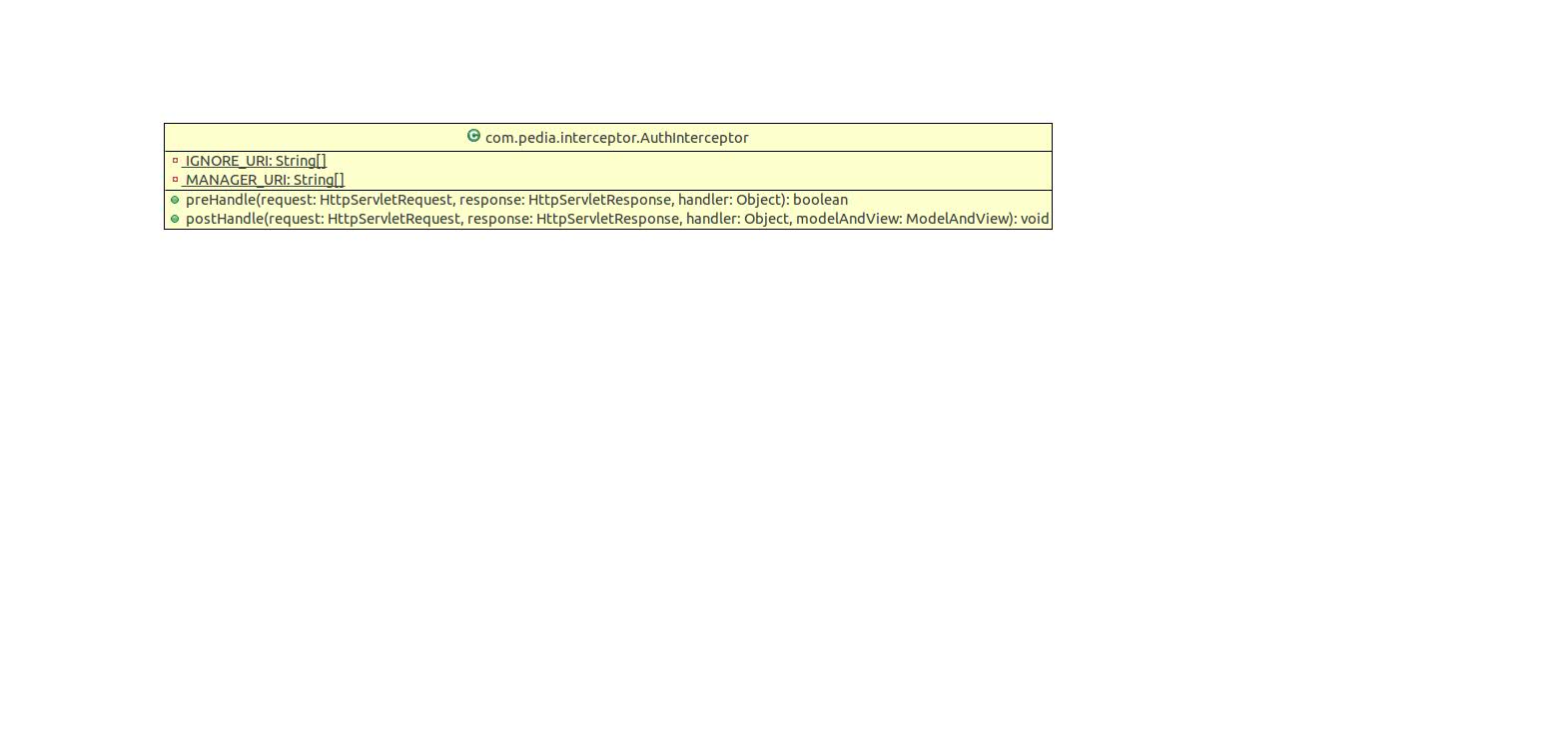
Service:

本系统的业务逻辑层，包含接口以及其实现类。

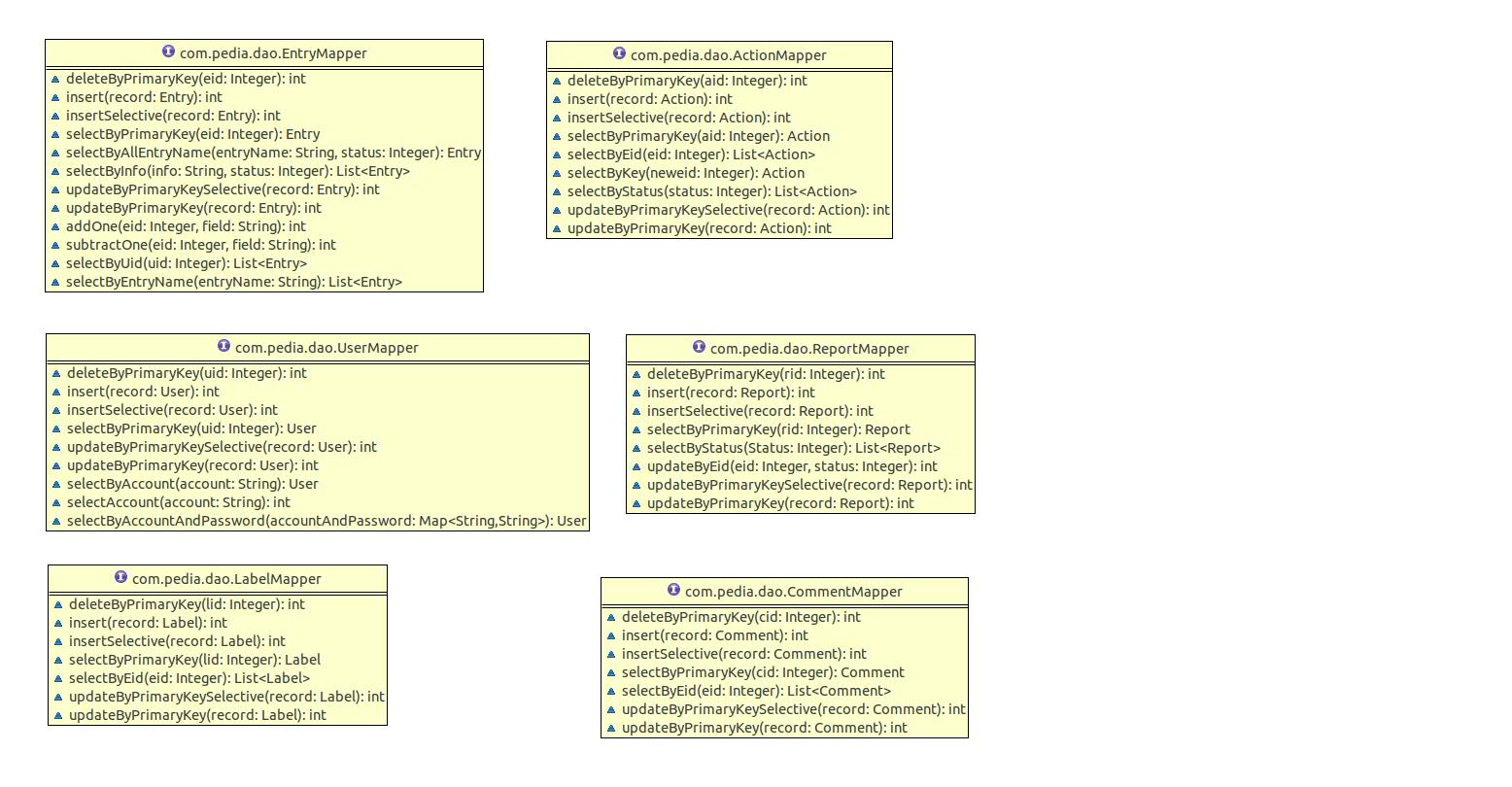


Model：

数据库的实体类，可以映射到数据表中的一列。

interceptor：

包含提供权限管理功能的类，用于允许授权访问和阻止非法访问。



Mapper：

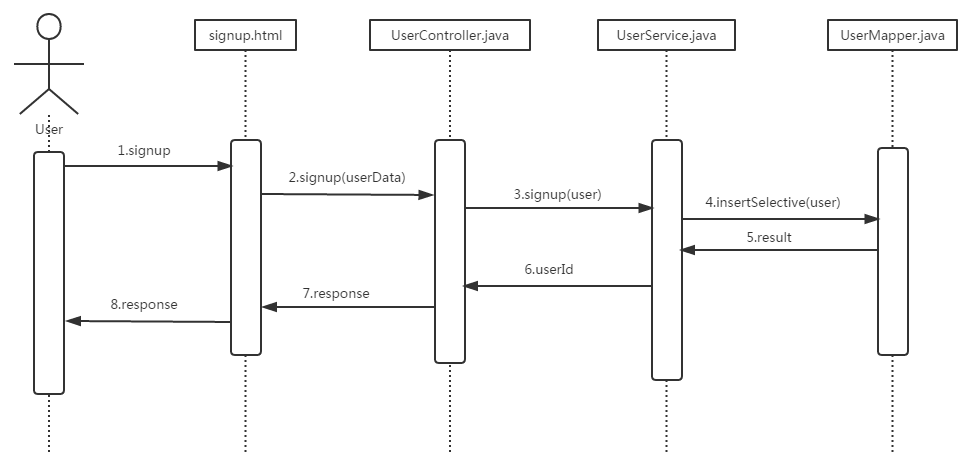
包含与数据库交互的操作类。

## 3.4系统用例实现详细设计

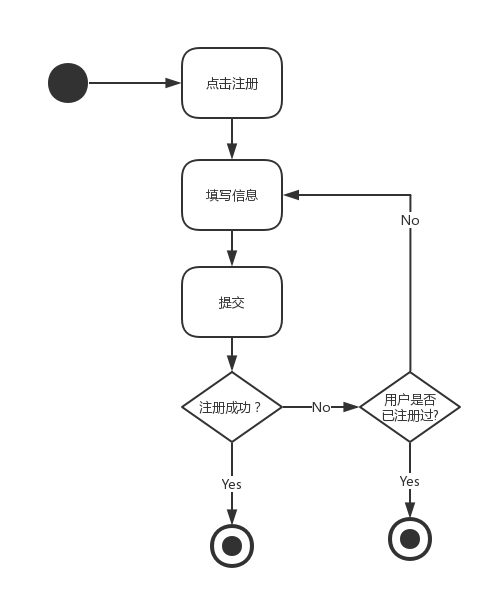
## 3.4.1 动态模型设计

1.注册 signup

顺序图：

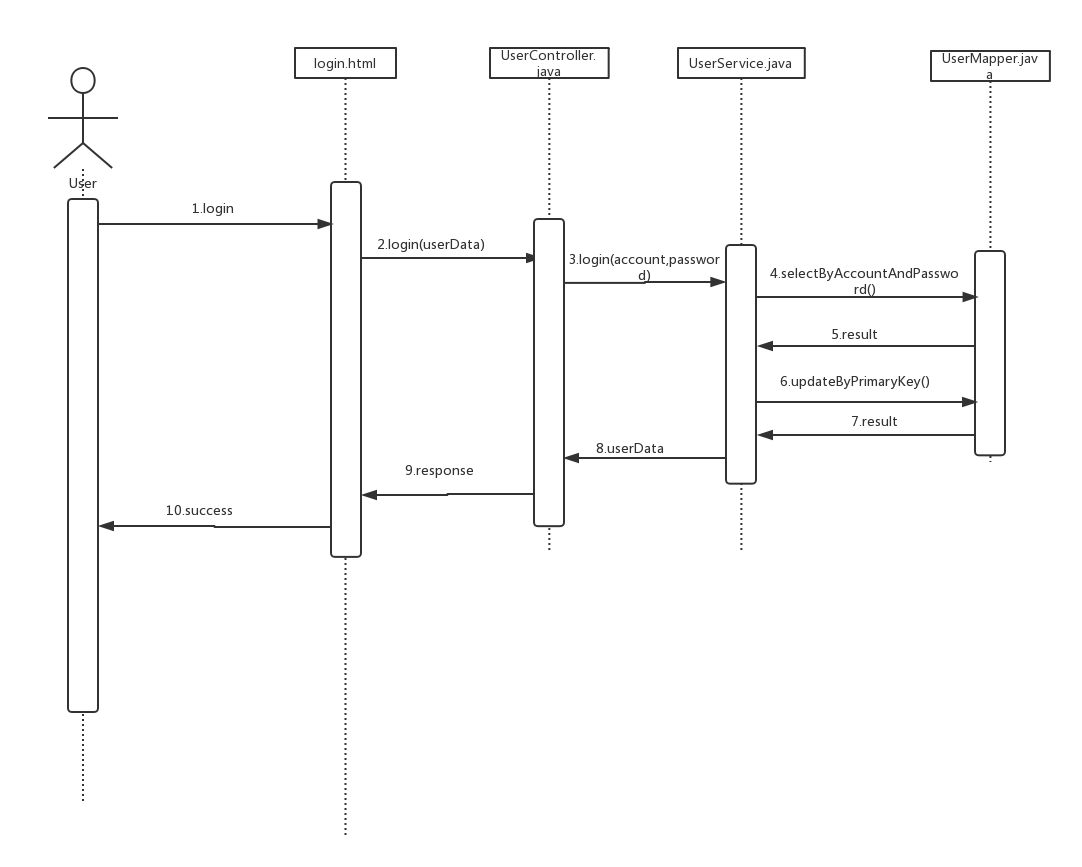


活动图：

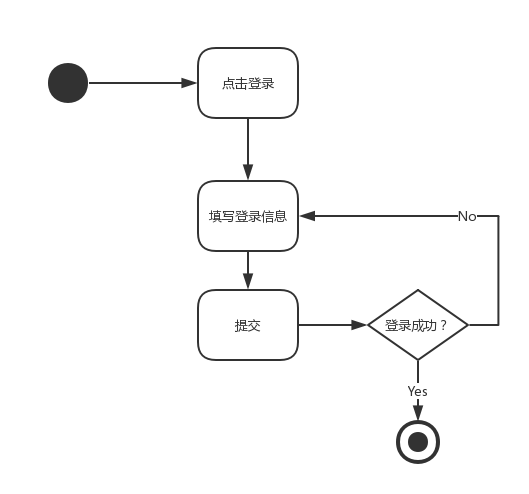


2.登录 login

顺序图：

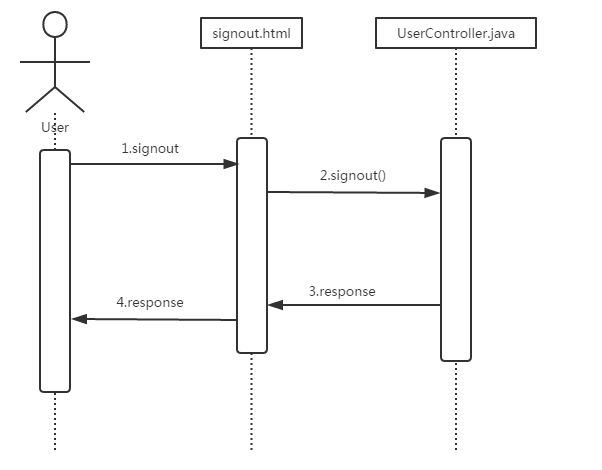


活动图：

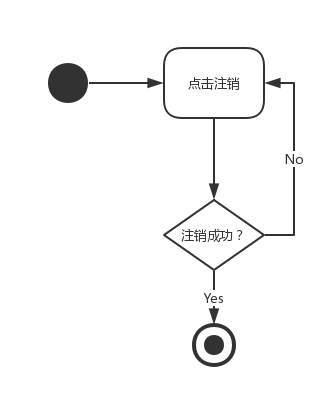


1. 注销 logout

顺序图：

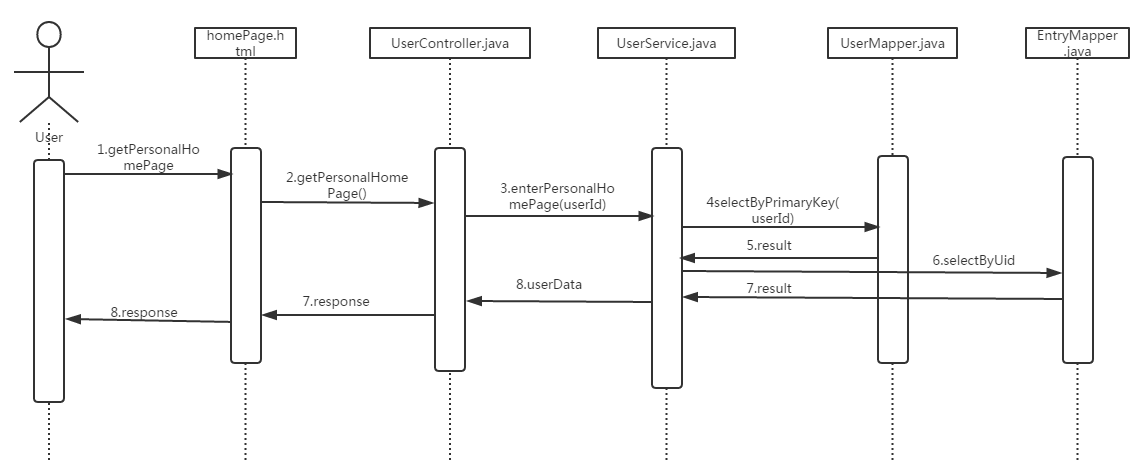


活动图：

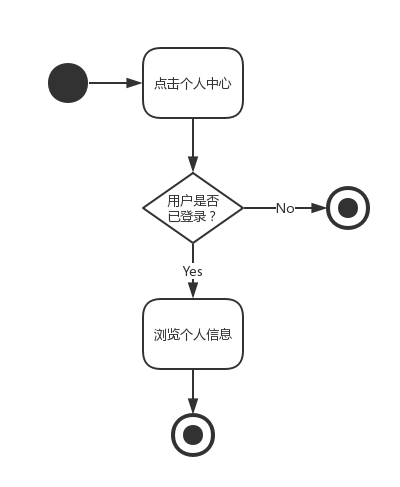


1. 进入个人主页 getPersonHomePage

顺序图：

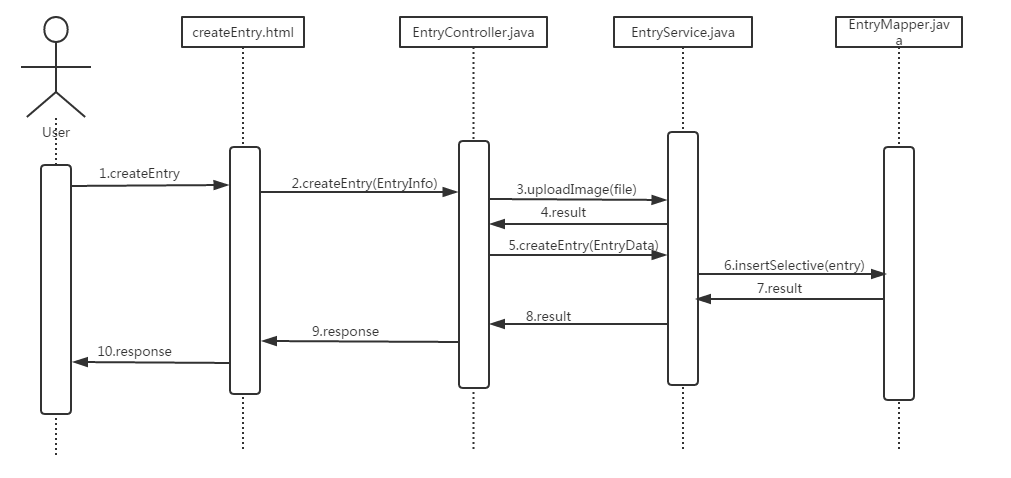


活动图：

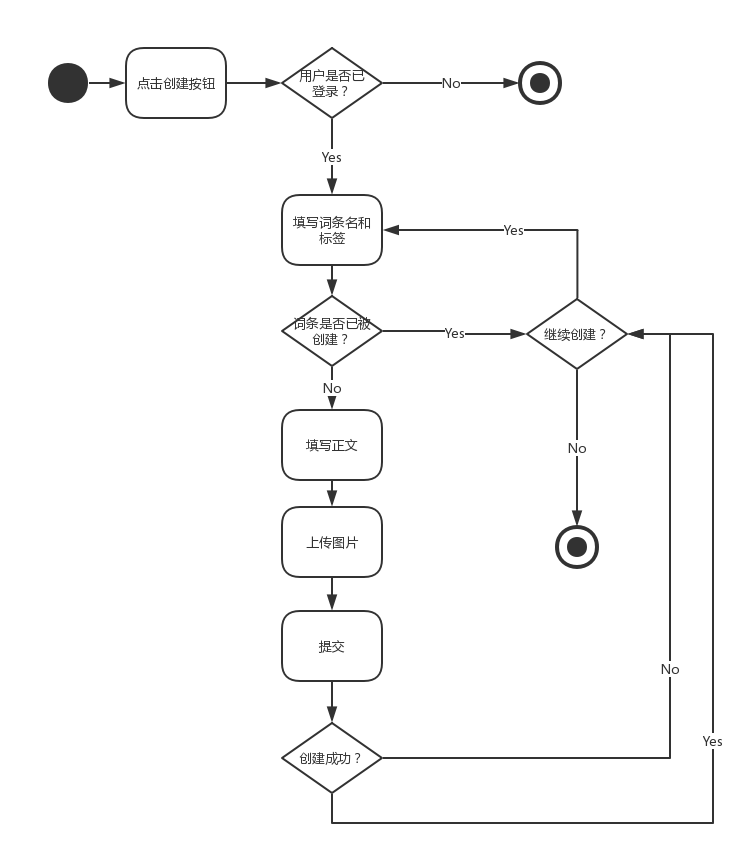


1. 创建词条 createEntry

顺序图：

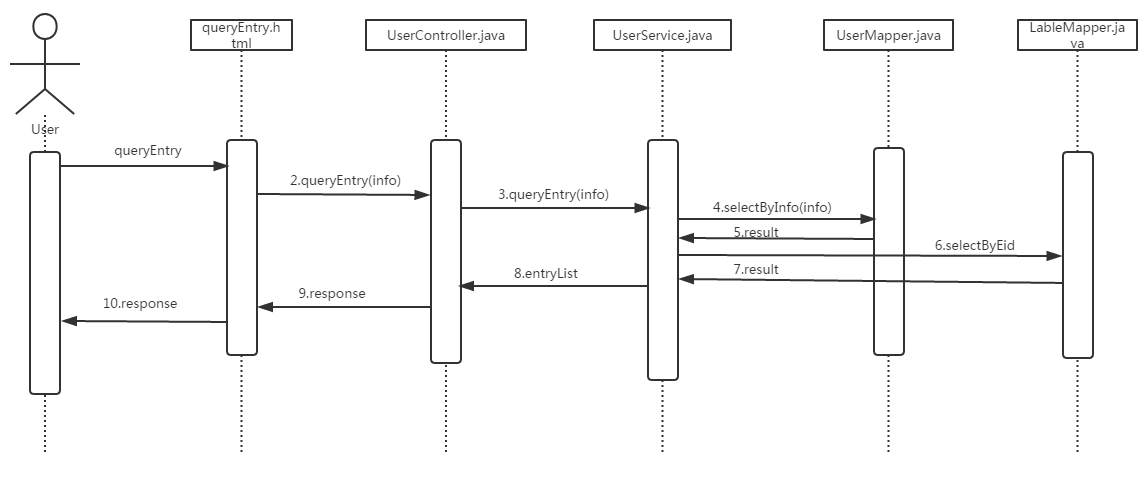


活动图：

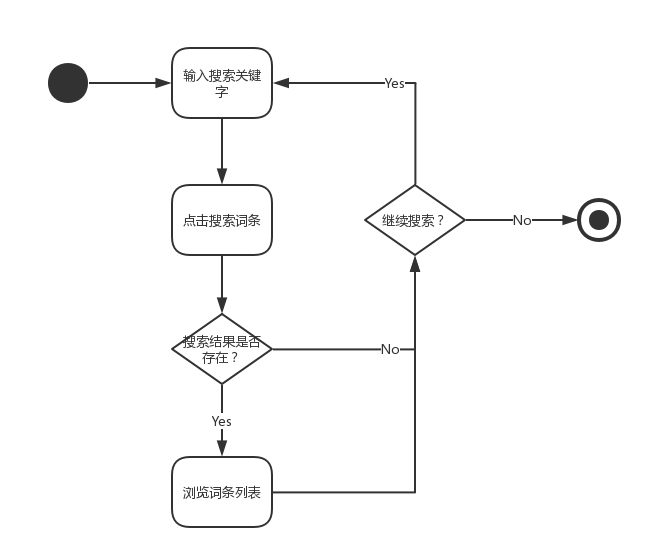


1. 查找词条 queryEntry

顺序图：

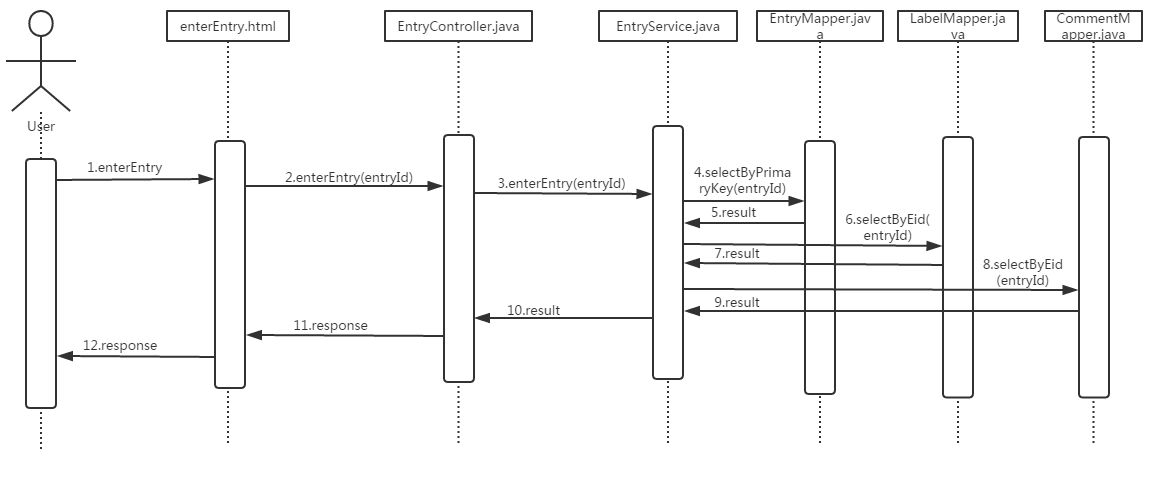


活动图：

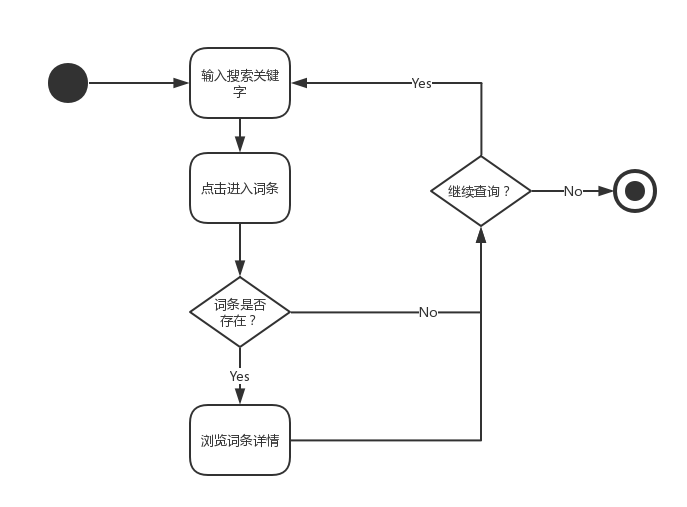


1. 进入词条 enterEntry

顺序图：

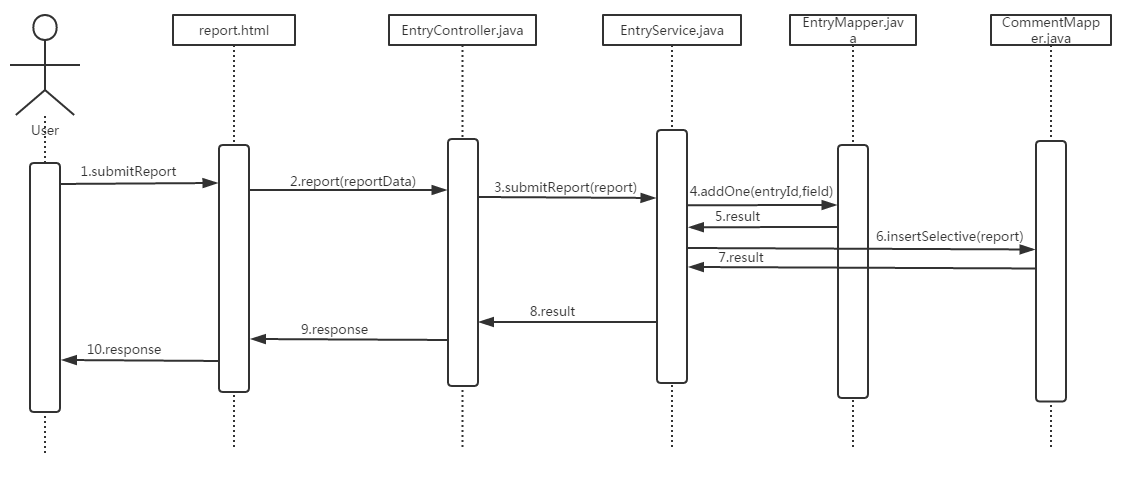


活动图：

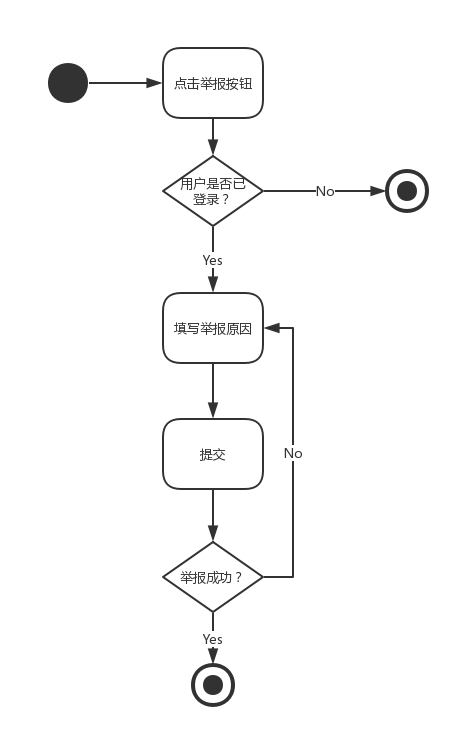


1. 举报词条 submitReport

顺序图：

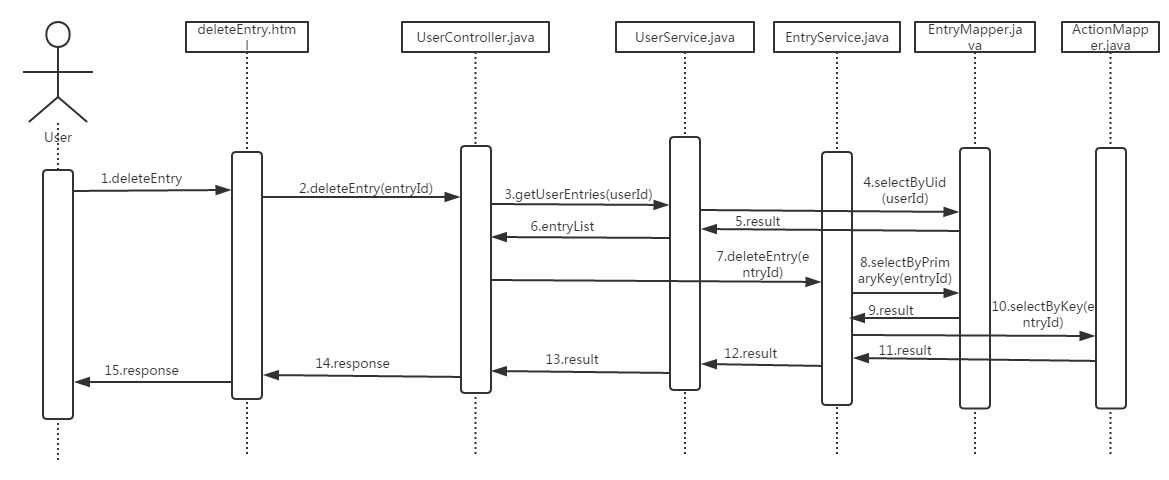


活动图：

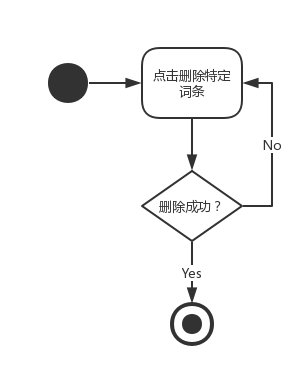


1. 删除词条 deleteEntry

顺序图：

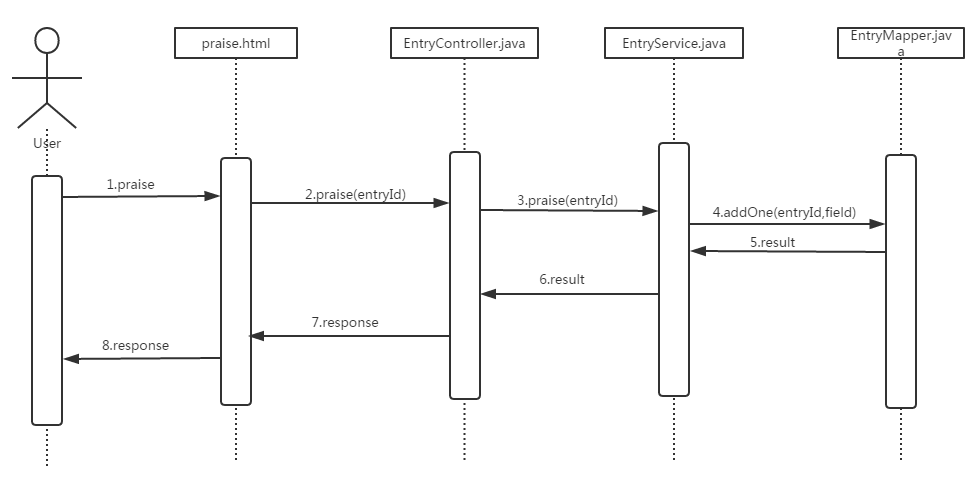


活动图：

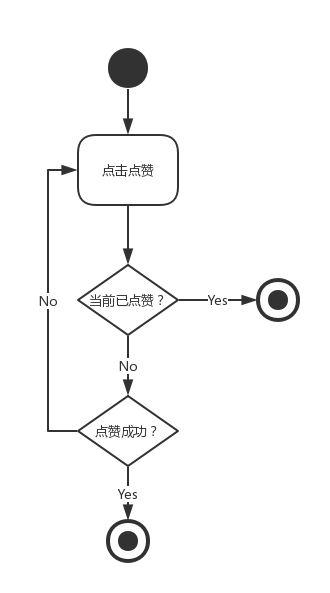


1. 点赞 praise

顺序图：

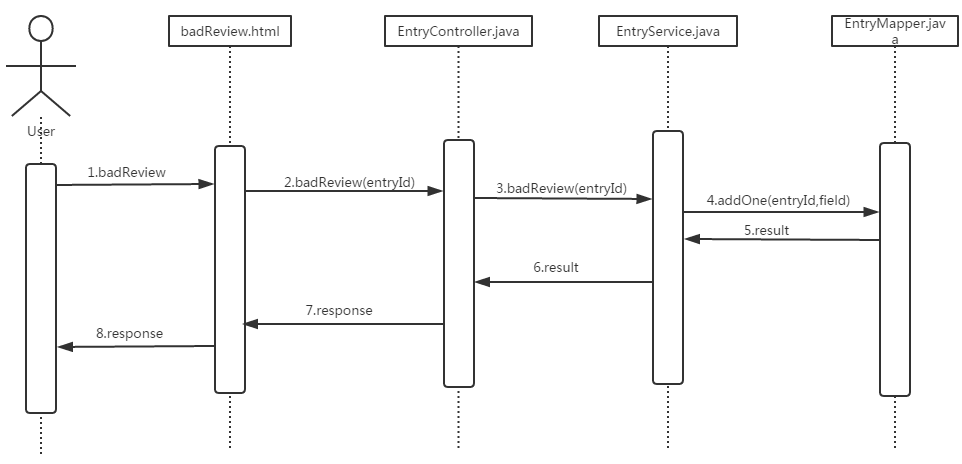


活动图：

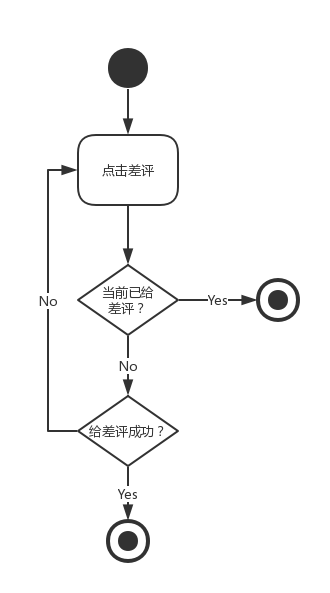


1. 差评 badReview

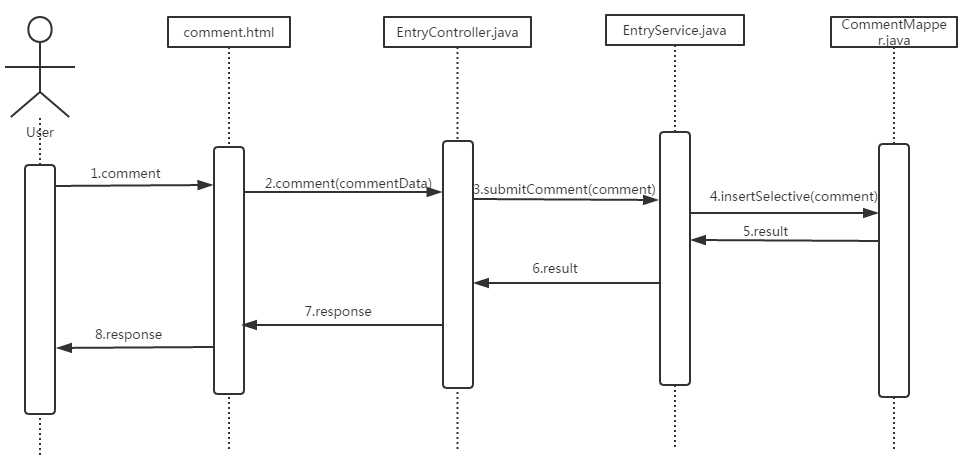
顺序图：



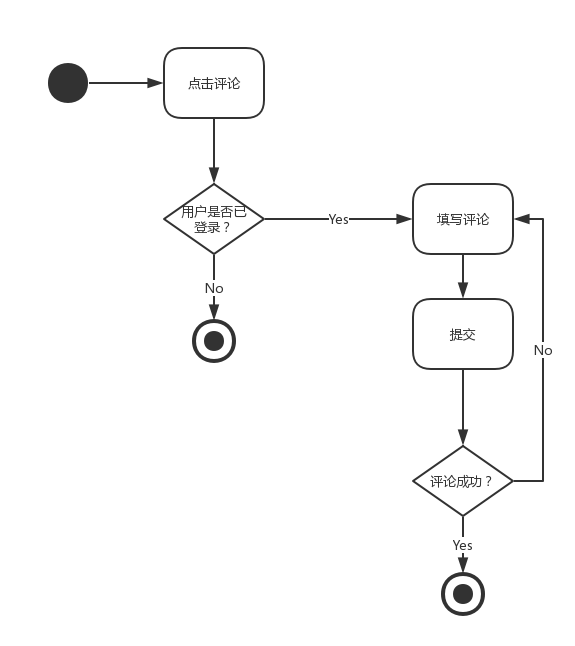
活动图：



1. 评论 submitComment

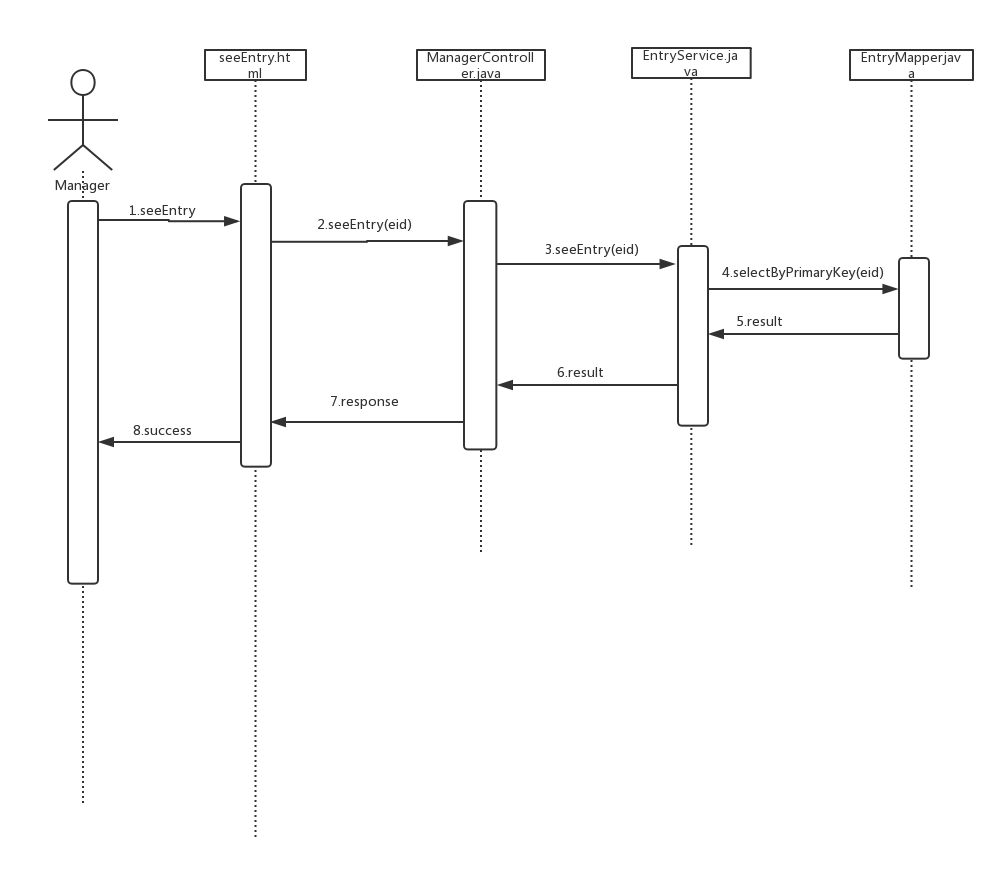
顺序图：

活动图：

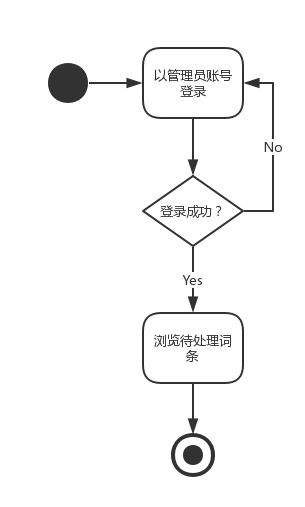


1. 查看待处理词条 getEntry

顺序图：

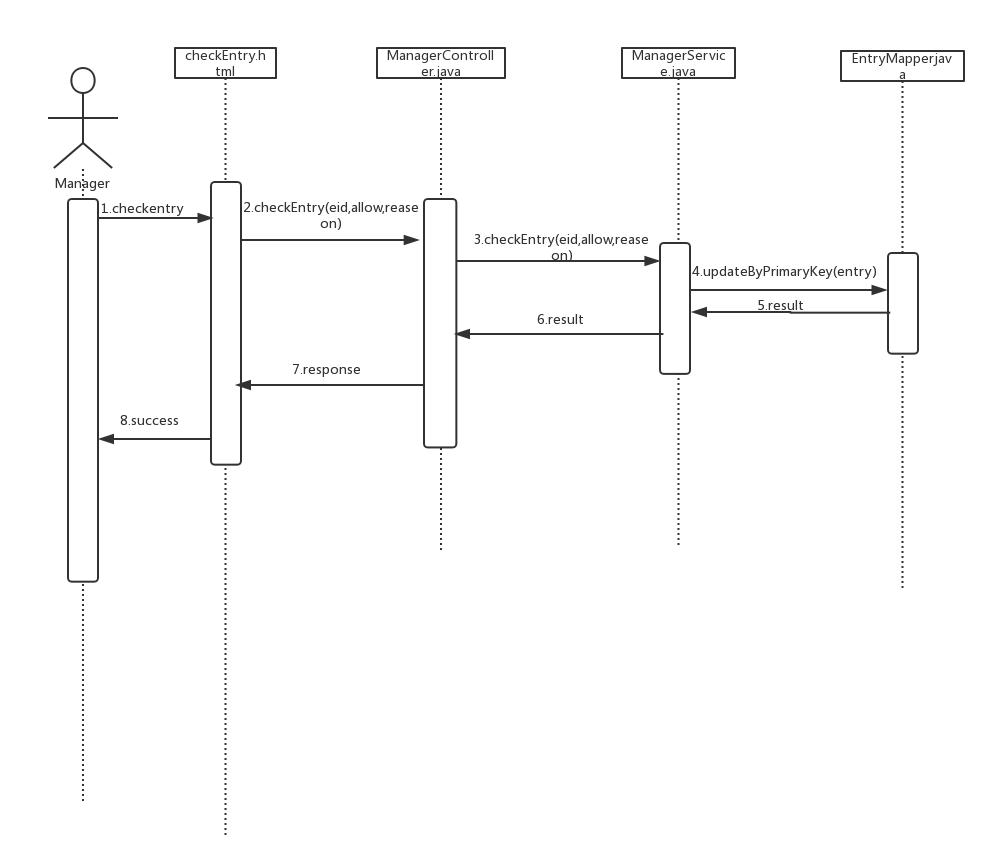


活动图：

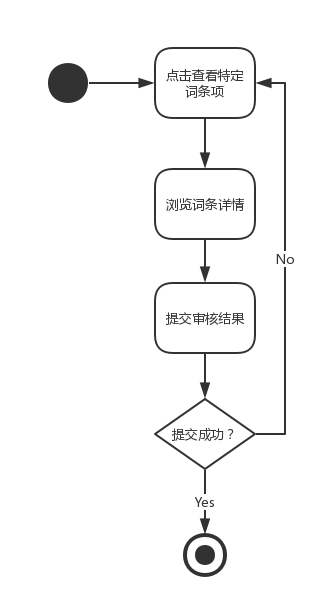


1. 审核待发布词条 checkEntry

顺序图：

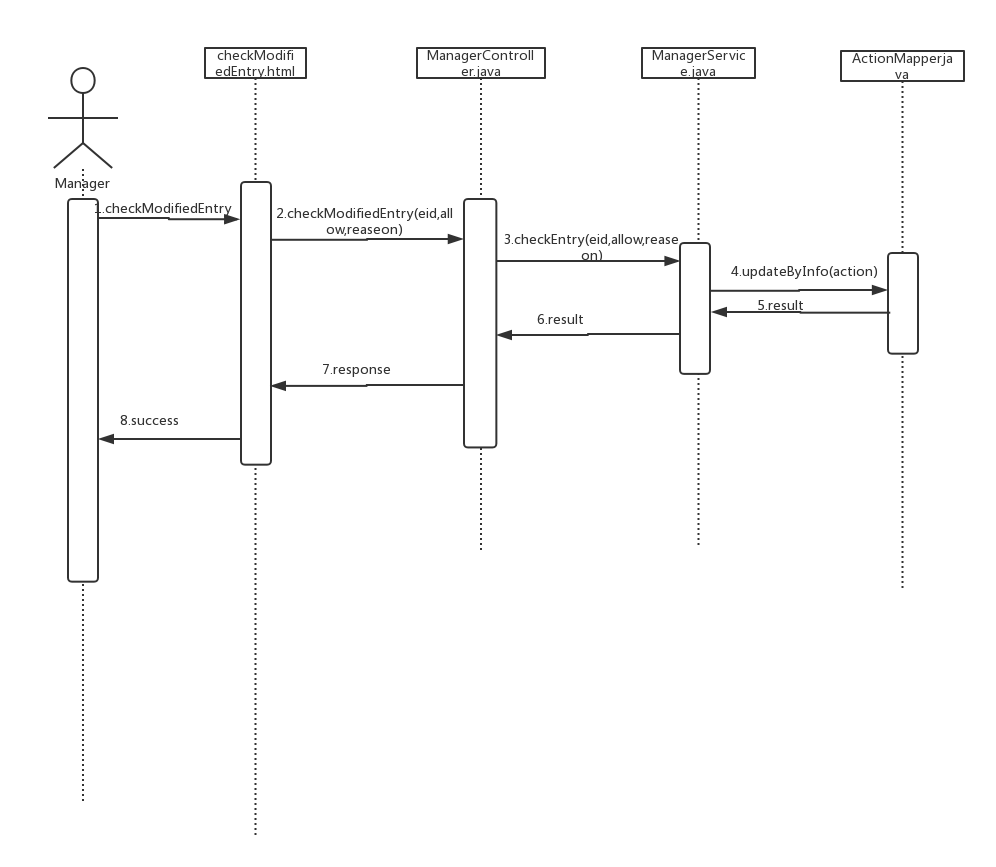


活动图：

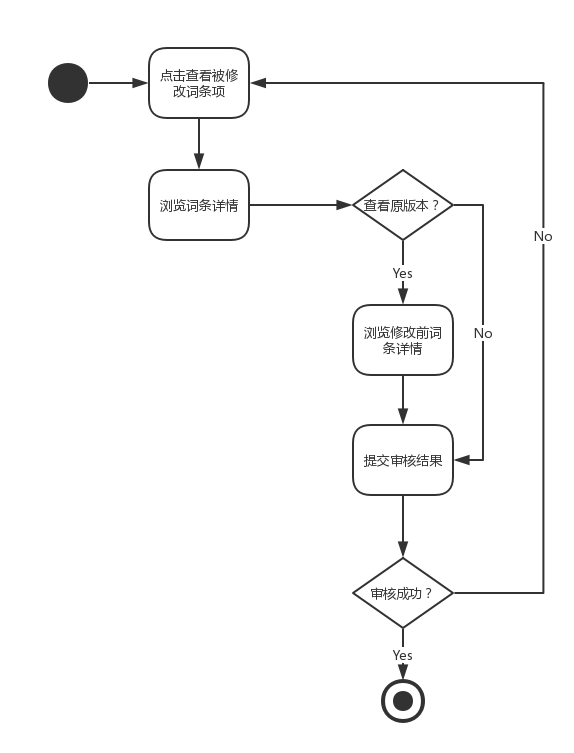


1. 审核待修改词条 checkModifiedEntry

顺序图：

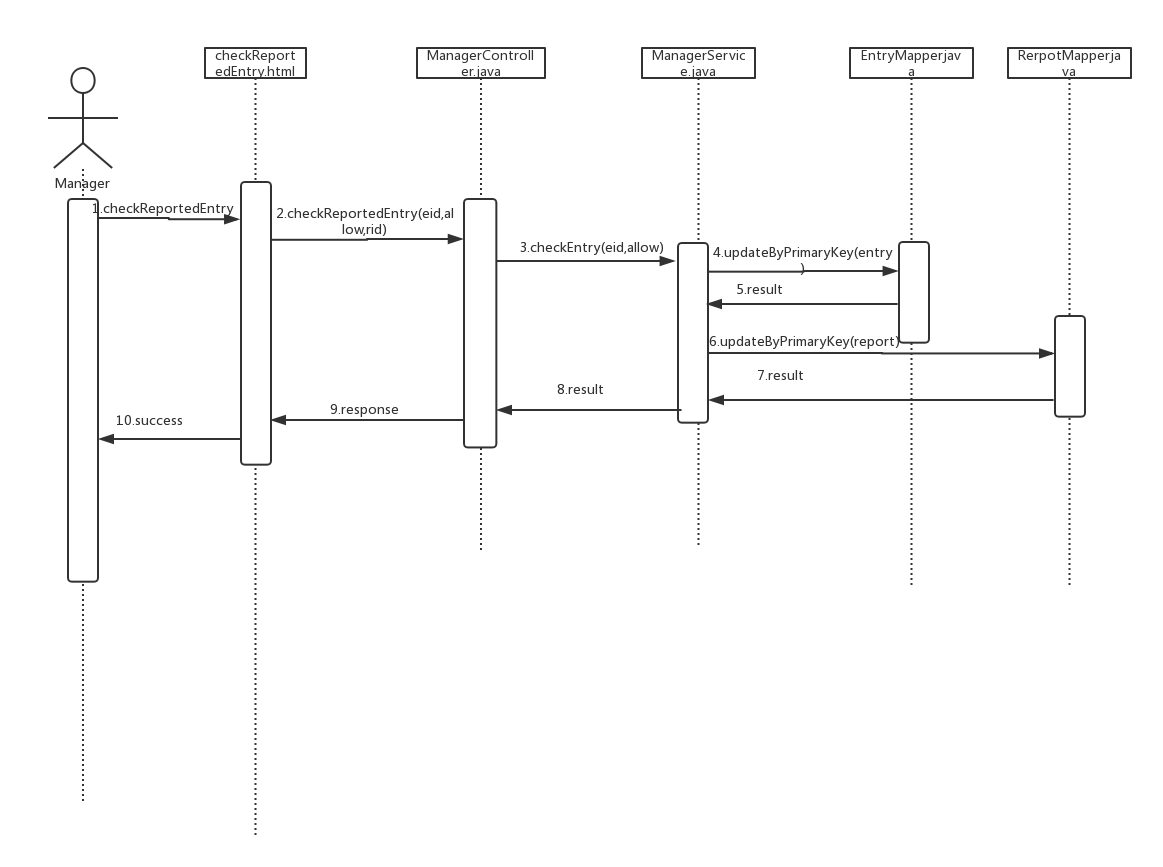


活动图：

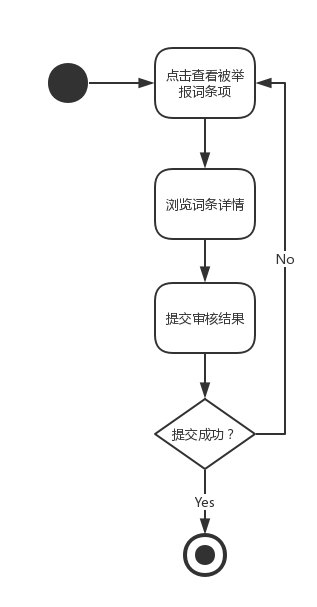


1. 审核被举报词条 checkReportedEntry

顺序图：



活动图：



## 3.5系统非功能设计

与功能性设计不同，系统的非功能性设计是指对系统在完成功能性任务时所要实现和遵循的一系列指标和约束的制定，如系统的界面、运行效率、安全性要求等等。

## 3.5.1系统用户交互界面设计

* + 界面元素：通常一个软件界面的元素包括界面主题颜色、字体颜色、字体大小、界面布局、界面交互方式、界面功能分布、界面输入输出模式等。其中，对于用户工作效率有显著影响的是：输入输出方式、交互方式、和功能分布。影响用户对系统友好性评价的元素有：颜色，字体大小和界面布局等，软件界面作为一个整体，其中一个元素不符合用户习惯、不满足用户要求都将大大降低用户对软件系统的认可度，设置影响用户的工作效率等。
  + 用户角色：通过分析用户特征，将每一些具有相同特征的用户归纳为集合，将用户集合定义为角色模型，同时赋予不同的优先级别，了解记录其界面需求。不同的用户角色在需求分析过程中的需求目标是不同的，侧重点也不同，依据用户角色不同的优先级别，平衡众多用户需求中矛盾，抽象出完整的GUI界面模型。
  + 帮助和提示原则：要对用户的操作命令做出反应，帮助用户处理问题。系统要设计有恢复出错的能力，在系统内部处理工作要有提示，尽量把主动权让给用户。

## 3.5.2质量属性设计

## 性能

充分关注业务需求和部署方式，对用户的动作能够及时做出反应。系统应支持100人以上同时访问服务器并支持40人以上同时访问数据库。

适当修改JVM参数以提高访问量。

利用缓存缩短读取数据的时间。

响应时间不超过3秒。

## 可用性

本系统采用数据库备份的方法确保数据的可靠存储，同时禁止数据的遗漏和重复，当系统出现故障和用户出现错误的操作后支持及时的恢复，用户在使用过程中遇到错误时可以立即定位问题，同时本系统在网络不稳定或者异常中断的时候会提供相应的容错措施。此外，通过功能的正确实现和硬件的合理配置，保证系统能顺利和准确的完成用户所要求的特定任务。

当系统出现故障时，可通过查看日志分析问题所在。

当缓存节点出现故障时，可立即通过哨兵机制切换至从节点。当数据库数据出现问题时，可通过冗余数据库恢复。

## 安全性

结合Session机制严格校验身份和限制登录者的操作权限，将其完成的操作限制在最小的范围内。管理人员进入系统前，必须首先通过系统的身份验证，然后系统根据登录的身份给予管理人员相应的权限。

## 易用性

用户界面中对于用户工作效率有显著影响的是：输入输出方式、交互方式、和功能分布等。本小组参照互动百科、百度百科，并结合实际情况调研了用户的实际需求，产品具有很高的用户友好度，与用户的交互十分紧密，力求在设计上满足用户的使用习惯，将主动权交给用户，操作简单。

增加词条处提供返回功能以供用户修改，提高用户主动性。

## 3.6 组件说明

### 3.6.1组件图

### 

### 3.6.2 组件说明

## 3.7部署说明

### 3.7.1部署图



### 3.7.2部署方式

数据库：用一台数据库服务器作为主数据库，并使用脚本定期保存数据库快照。

缓存：使用一个Redis作为主缓存节点，并配置两台从缓存节点进行主动冗余，使用哨兵机制，当主节点宕机时，令一个从节点成为主节点。

服务器基本信息

操作系统：Ubuntu 14.04 LTS

CPU:1核

内存：1.0GB

宽带：1Mbps

运行软件环境:

Jvm 1.8.0\_101

Tomcat V7.0

MySQL 5.5.49

Redis server v2.8.4