系统分析与设计

**科研信息共享平台**

**分析模型**

|  |  |  |
| --- | --- | --- |
| **项目组成员信息** | | |
| **小组名称** | **爆肝工程师的软工大组** | |
| **小组联系人** | **李昕航** | |
| **学号** | **姓名** | **本文档中主要承担的工作内容** |
| 16211063 | 李昕航 | 类图绘制，文档撰写 |
| 16211013 | 赵志浩 | 顺序图绘制，功能模块分析 |
| 16211042 | 郭镕昊 | 顺序图绘制，功能模块分析 |
| 16071072 | 洪振东 | 顺序图绘制，功能模块分析 |
| 15081070 | 张雨任 | 顺序图绘制，功能模块分析 |
| 16171086 | 曾宥崴 | 顺序图绘制，功能模块分析 |
| 15081013 | 霍晓亮 | 顺序图绘制，功能模块分析 |
| 16041124 | 王明鑫 | 顺序图绘制，功能模块分析 |
| 16211006 | 徐家兴 | 顺序图绘制，功能模块分析 |

2019-03

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| 01 | 2019.3.25 | 李昕航 | 李昕航 | 完成静态类分析模型 |
| 02 | 2019.3.26 | 李昕航 | 李昕航 | 完成顺序图和用例交互模型 |
| 03 | 2019.3.27 | 李昕航 | 李昕航 | 完成用例分析模型 |

目 录

[1 引言](#_Toc1445991984)

[1.1 目的](#_Toc1862325636)

[1.2 文档说明](#_Toc532809227)

[1.3 参考资料](#_Toc2065353846)

[2 静态类分析模型](#_Toc476419614)

[2.1 用例模型](#_Toc1365416482)

[2.2 核心业务类分析模型](#_Toc544561132)

[2.3 类详细描述](#_Toc2011125657)

[2.4 类关系描述](#_Toc1743797066)

[2.4.1 多重性关联关系与聚合关系说明](#_Toc1287957653)

[2.4.2 泛化关系说明](#_Toc69112211)

[3 用例交互模型](#_Toc1927760897)

[4 用例分析模型](#_Toc791613590)

[5 总结](#_Toc988413965)

# 分析模型

# 1 引言

## 1.1 目的

随着科学研究的发展，科研不再是少数人的专利，对于科研的需求在普通人群中越来越普及。为了更方便地为科研工作者们提供科研信息的获取途径以及学术交流的环境，我们旨在开发一个智能科研信息共享平台。本系统主要用于共享、管理和维护科技专家资源。平台以科技资源为中心，由专门的管理机构运行和维护，并在统一管理的基础上建立科技专家门户系统。用户可以在平台中搜索、查看、购买和下载科技资源，科技专家管理和维护自己的门户系统，管理员审核认证专家和资源，形成一个稳定、可持续的资源共享平台。

## 1.2 文档说明

文档内容为分析模型，是系统设计的必要环节，主要包含了资源共享平台中的业务抽象单元之间的逻辑关系，建立了系统结构的基本模型，以静态类分析和用例交互模型的形式描述以及更加细致地进行需求分析以及系统设计。

由于项目实际实现涉及到的信息传递过程较为复杂，因此本文档只作为架构设计和业务单元交互的参考，具体的实现细节在文档中没有过多的描述。

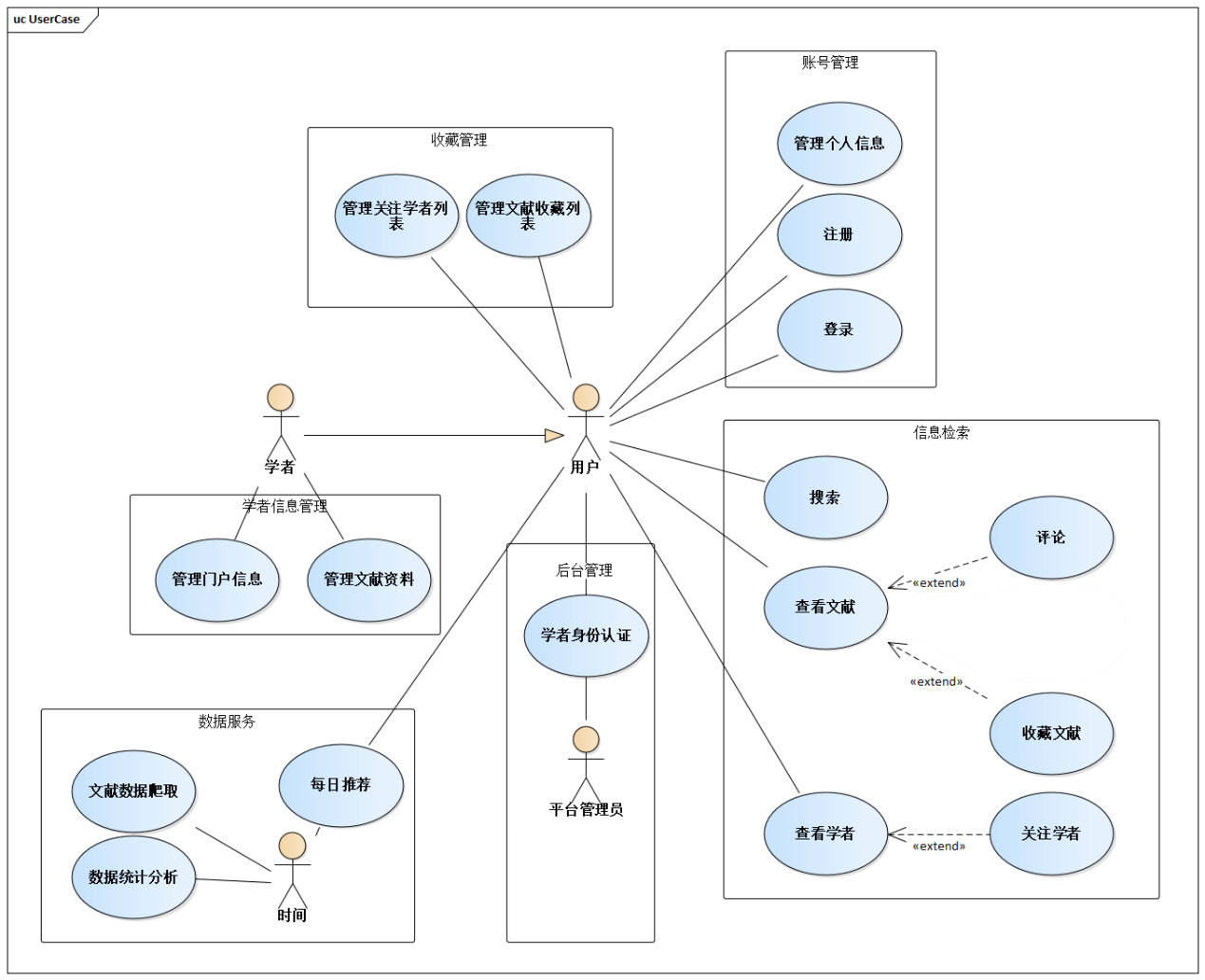
## 1.3 参考资料

|  |  |  |
| --- | --- | --- |
| 资料名称 | 作者 | 日期 |
| 科技专家资源共享平台-背景陈述 | 谭火彬 | 2019-2-26 |
| 需求调研报告 | 李昕航 | 2019-3-7 |
| 需求规格说明书 | 李昕航 | 2019-3-18 |

# 2 静态类分析模型

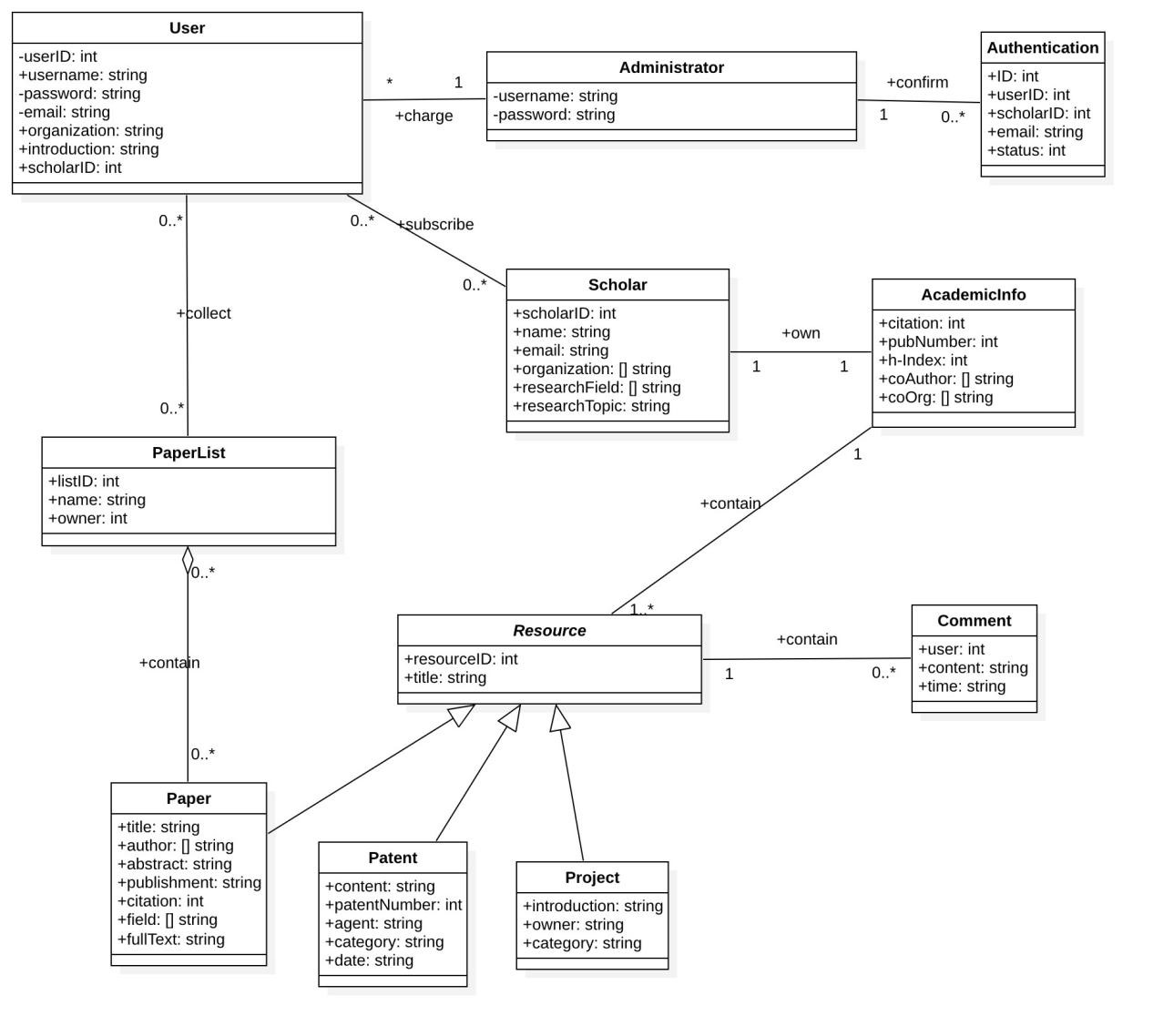
## 2.1 用例模型

静态类分析时首先就需要从用例中进行实体抽取，得到抽象的概念，同时用例模型还隐性地表示了抽象的概念之间的关系，因此用例模型对于静态类分析模型具有极强的指导作用。



**图2-1 用例图**

## 2.2 核心业务类分析模型



**图2-2 核心业务类图**

## 2.3 类详细描述

|  |  |  |
| --- | --- | --- |
| 类名 | 属性 | 功能说明 |
| User | userID: int  username: string  password: string  email: string  organization: string  introduction: string  scholarID: string | 用户实体类，包含唯一识别码用户ID，用户的用户名、密码、邮箱等个人信息。  用户类还包含关注的学者列表，收藏的文献列表，以及一个定时更新的每日推荐列表，有助于用户的便捷使用。  用户认证身份后还有绑定的学者ID，根据此属性可以判断用户的学者身份，给予相应的权限对学者门户信息进行管理。 |
| Scholar | scholarID: string  name: string  email: string  organization: [] string  researchField: [] string  researchTopic: string | 学者实体类，主要指抽象概念的“学者”，包含唯一识别码学者ID，姓名邮箱等基本信息。  除了学者的基本信息外，此实体类还包含一个绑定的学术信息对象，用以专门确定学者的学术信息。 |
| AcademicInfo | citation: int  pubNumber: int  h-Index: int  coAuthor: [] string  coOrg: [] string | 学术信息实体类，唯一所属于一名学者，列举了学者的引用量，发表数，学术资源列表，h-index等有关学术的信息。 |
| PaperList | listID: int  name: string  owner: int | 文献列表实体类，包含多个文献，增加了列表名和创建者，便于用户使用和管理权限确定。 |
| Resource | resourceID: int  title: string | 资源实体类，为抽象类，是文献、专利、项目的父类，包含有ID、标题和评论等共用信息。 |
| Paper | paperID: string  title: string  author: [] string  abstract: string  publishment: string  citation: int  field: [] string  fullText: string | 文献实体类，包含论文的基本信息。  额外增加了全文属性，用于用户的下载操作。 |
| Project | introduction: string  owner: string  category: string | 项目实体类，包含项目的简介，分类和所有人。  由于平台目的主要为展示成果而非传播资源，因此项目并没有内容实体，只包含摘要信息。 |
| Patent | content: string  patentNumber: int  agent: string  category: string  date: string | 专利实体类，包含专利号、专利类别、申请时间以及代理人。 |
| Comment | user: int  content: string  time: string | 评论实体类，包含评论用户、评论内容和评论时间信息。 |
| Administrator | username: string  password: string | 平台管理员实体类，只包含用户名和密码，不提供注册，为数据库直接内置账号。 |
| Authentication | ID: int  userID: int  scholarID: int  email: string  status: int | 认证申请实体类，包含申请人、认证对象以及申请状态信息。  认证需要进行多边操作，邮箱认证通过后还需要管理员认证才能够审核成功。 |

**表2-1 核心业务类属性描述**

## 2.4 类关系描述

### 2.4.1 多重性关联关系与聚合关系说明

**表2-2 多重性关联关系与聚合关系描述**

|  |  |  |
| --- | --- | --- |
| 编号 | 关系图示 | 关系描述 |
| 1 | Administrator  User  1  \*  charge | 平台管理员可以对用户信息进行管理，由于至少要有复数个用户管理才有意义，所以应为一对多关系 |
| 2 | Administrator  Authentication  1  0..\*  confirm | 平台管理员可以确认身份认证申请，为一对多关系 |
| 3 | User  Scholar  0..\*  0..\*  subscribe | 用户可以关注感兴趣的学者，用户与学者之间存在关注与被关注的关系，同一个学者可以被多个用户关注，为多对多关系 |
| 4 | Scholar  AcademicInfo  1  1  own | 学者除了拥有基础的个人信息外还有专门的学术信息，因为学术信息是隶属于特定的学者的，因此学者和学术信息为一对一关系 |
| 5 | User  PaperList  0..\*  0..\*  collect | 用户可以收藏并管理文献列表，由于业务场景中包含与他人的分享等功能，因此文献列表并非为私有，而是可以被共享的资源，用户与文献列表的收藏关系为多对多关系 |
| 6 | AcademicInfo  Resource  1  1..\*  contain | 学术信息包含多个资源，并且学术信息中的其他属性都根据资源进行计算 |
| 7 | Resource  Comment  1  0..\*  contain | 每个资源都可以被评论，评论本身无法独立存在，需要绑定到资源上，且资源可以被多次评论 |
| 8 | Paper  PaperList  0..\*  0..\*  contain | 文献列表为文献的集合，每个文献可以属于多个文献列表，文献列表也包含多个文献，文献和文献列表属于多对多的聚合关系 |

### 2.4.2 泛化关系说明

**表2-3 泛化关系描述**

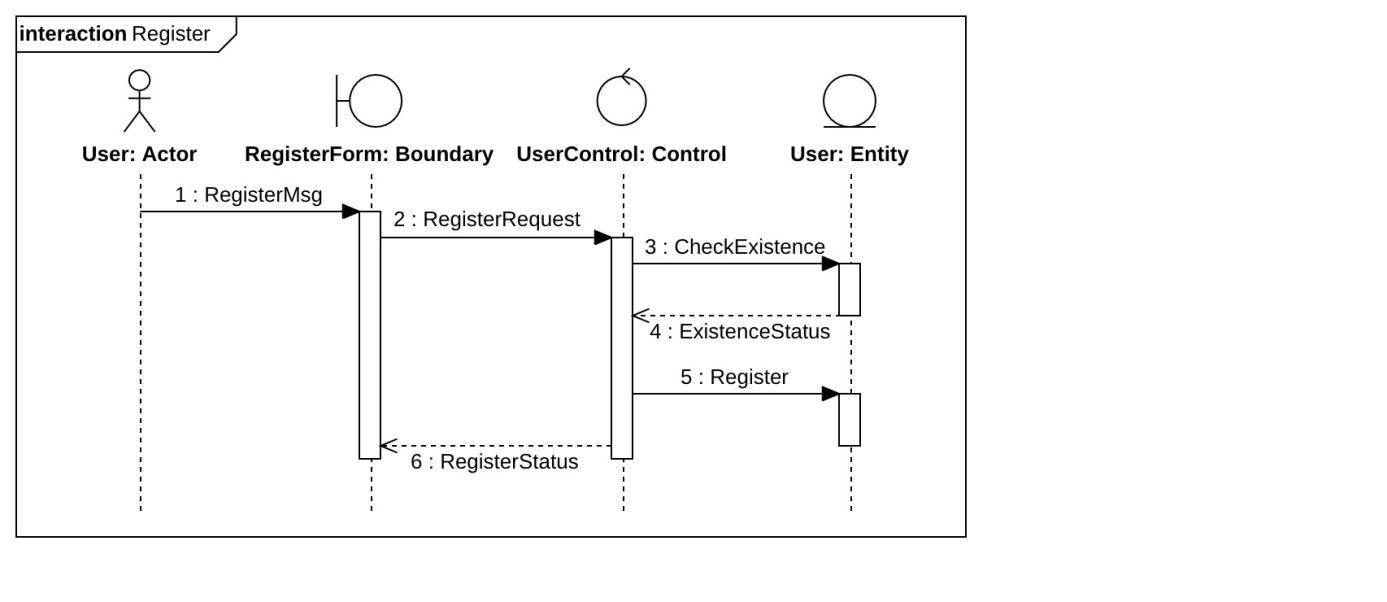
|  |  |  |
| --- | --- | --- |
| 编号 | 关系图示 | 关系描述 |
| 1 | Patent  Resource | 一共有论文，项目和专利三种资源，它们在系统中的行为是相同的，只是属性有一些不同。例如，论文的出版源、专利的专利号等属性。 |
| 2 | Project  Resource |
| 3 | Paper  Resource |

# 3 用例交互模型

静态类模型只表示了类的属性，方法以及类与类之间的关系，而对于动态的类间交互过程却没有体现，尤其是核心业务类主要表示实体类，而对于边界类和控制类都没有进行很好的表示，不能够显示出用户在使用平台时的信息传递和函数调用过程。

使用用例交互模型，主要以时序图为主，有助于更加微观更加细致地表现运行时信息在整个系统之中的传递过程，更能体现出各个类之间的主从和方法调用的关系。

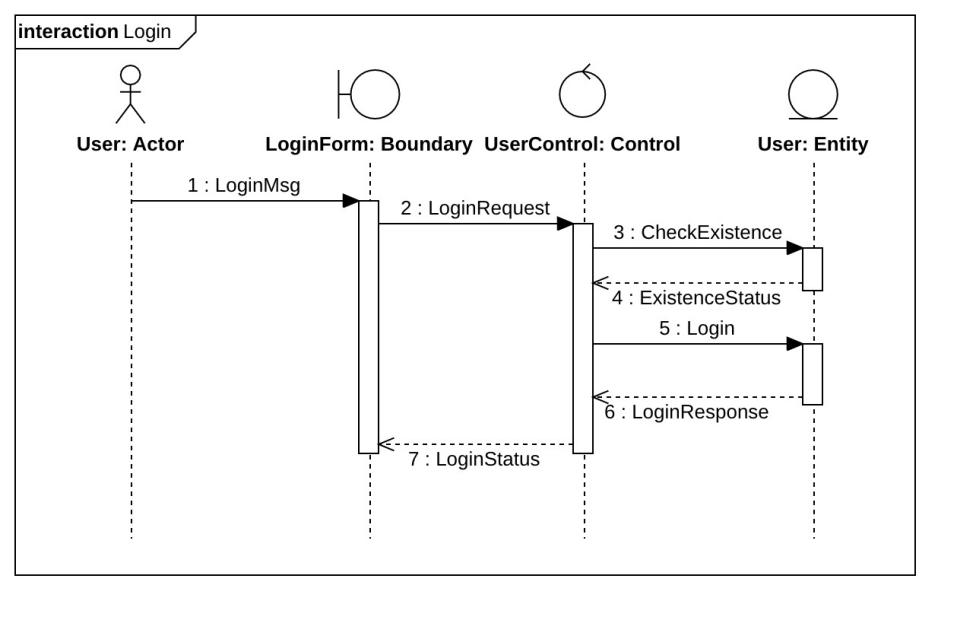
* 注册时序图：

****

**图3-1 注册时序图**

1. 用户输入注册信息到注册表单内
2. 注册表单向用户控制类发起注册请求
3. 用户控制类检索用户数据库，判断用户是否存在
4. 用户数据库返回检索的用户存在状态
5. 用户控制类创建新的用户实体对象，完成注册功能
6. 用户控制类将注册结果返回边界类显示给用户

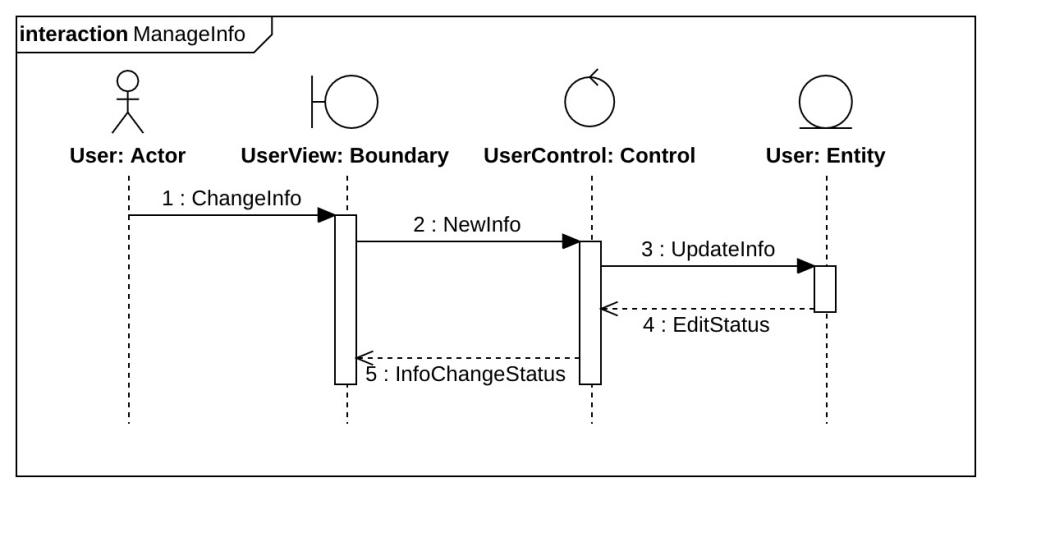
* 登录时序图：

****

**图3-2 登录时序图**

1. 用户输入登录信息到登录表单内
2. 登录表单边界类向用户控制类发起登录请求
3. 用户控制类检索用户数据库判断用户是否存在
4. 用户数据库返回用户存在状态
5. 用户控制类尝试登录该用户
6. 用户数据库返回用户登录状态
7. 用户控制类向边界类返回登录结果，根据结果进行相应页面的跳转

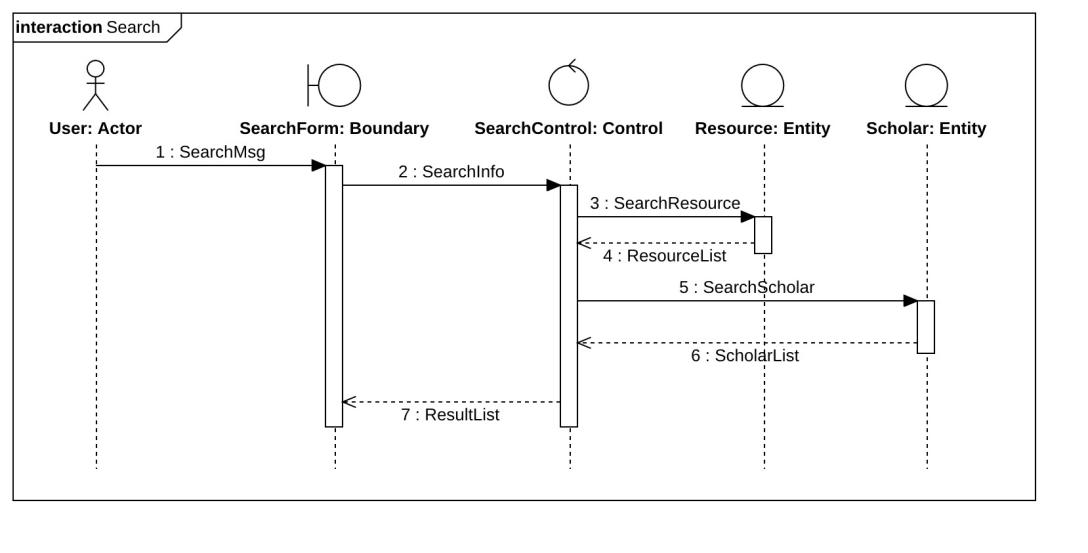
* 个人信息管理时序图：

****

**图3-3 个人信息管理时序图**

1. 用户修改个人信息并保存
2. 用户界面类将更新的信息发送给用户控制类
3. 用户控制类更新相应的用户的信息
4. 用户实体类返回更新状态
5. 用户控制类返回更新状态，用户界面更新信息

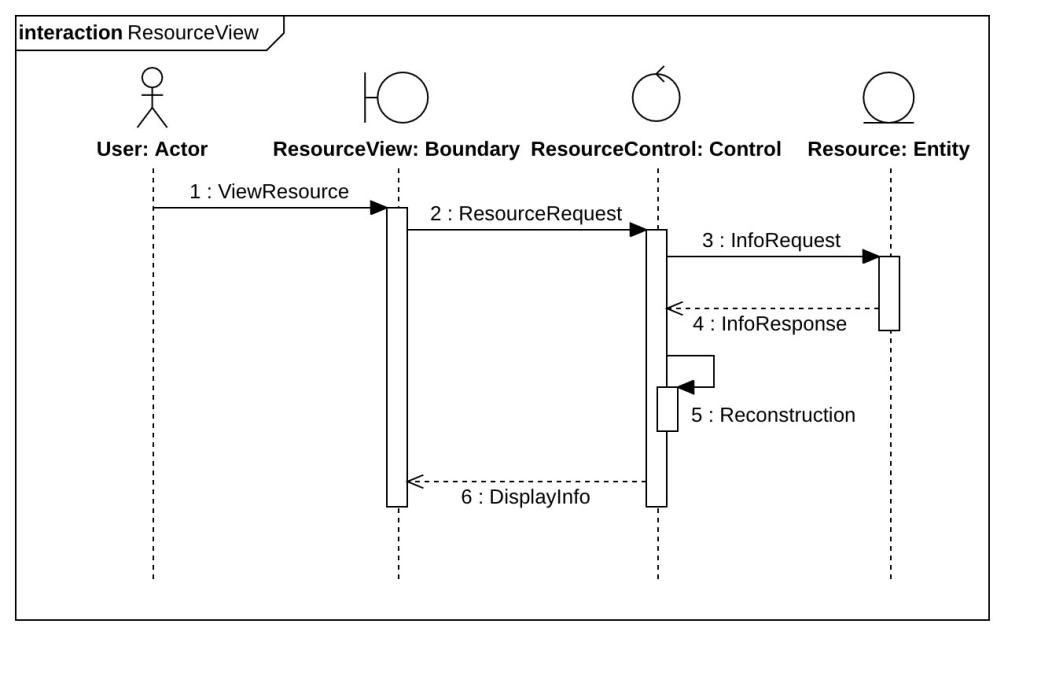
* 搜索时序图：

****

**图3-4 搜索时序图**

1. 用户在搜索框内选择搜索条目并输入搜索信息
2. 搜索表单发送搜索信息给搜索控制类
3. 搜索控制类根据搜索条目检索资源数据库和学者数据库
4. 资源数据库和学者数据库返回搜索结果列表
5. 搜索控制类返回搜索结果列表，边界类进行相应的处理并展示

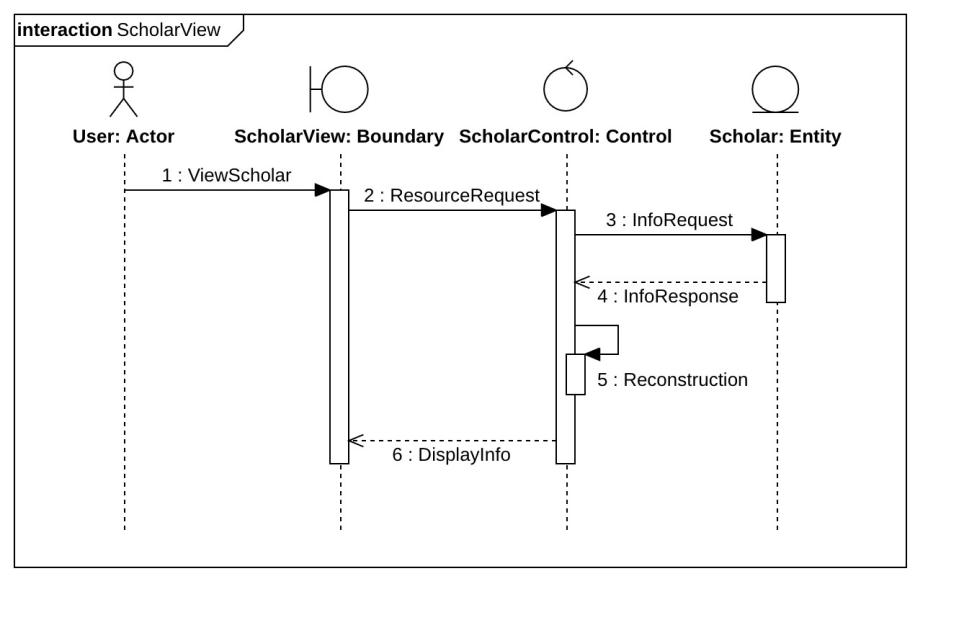
* 查看资源时序图：

****

**图3-5 查看资源时序图**

1. 用户点击条目查看资源
2. 资源边界类发送查看请求到资源控制类
3. 资源控制类检索资源数据库获取资源
4. 资源数据库返回对应的资源信息
5. 资源控制类对资源信息根据类型进行重构
6. 资源控制类返回重构后的可展示资源信息

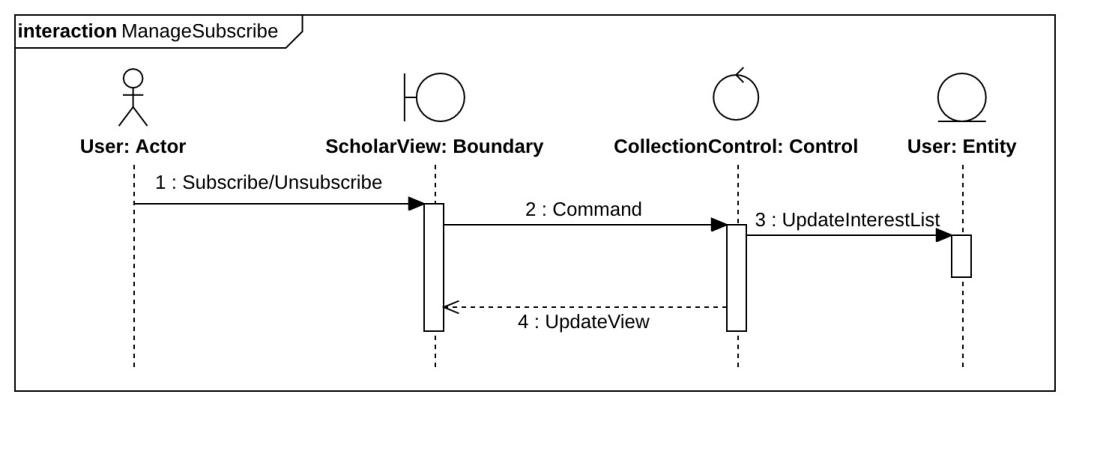
* 查看学者时序图：

****

**图3-6 查看学者时序图**

1. 用户点击条目查看学者
2. 学者界面边界类发送查看请求到学者控制类
3. 学者控制类检索学者数据库获取学者信息
4. 学者数据库返回相应的学者信息
5. 学者控制类对学者信息进行重构
6. 学者控制类返回重构后的数据到学者界面边界类进行展示

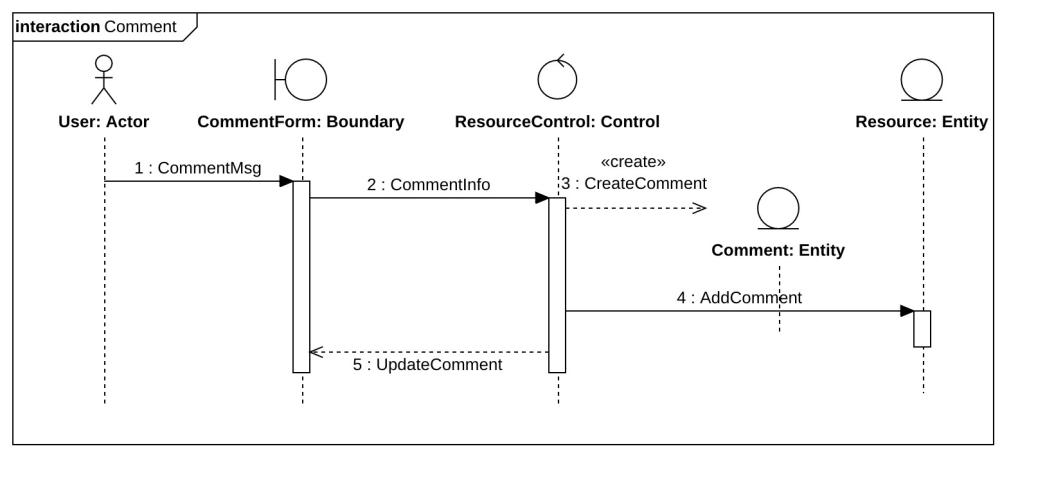
* 管理关注列表时序图：

****

**图3-7 管理关注列表时序图**

1. 用户在学者门户界面点击关注/取关
2. 学者界面边界类发送指令到收藏控制类
3. 收藏控制类根据用户信息，学者信息和关注状态更新用户数据库内关注列表
4. 收藏控制类发送信号到学者界面边界类更新页面

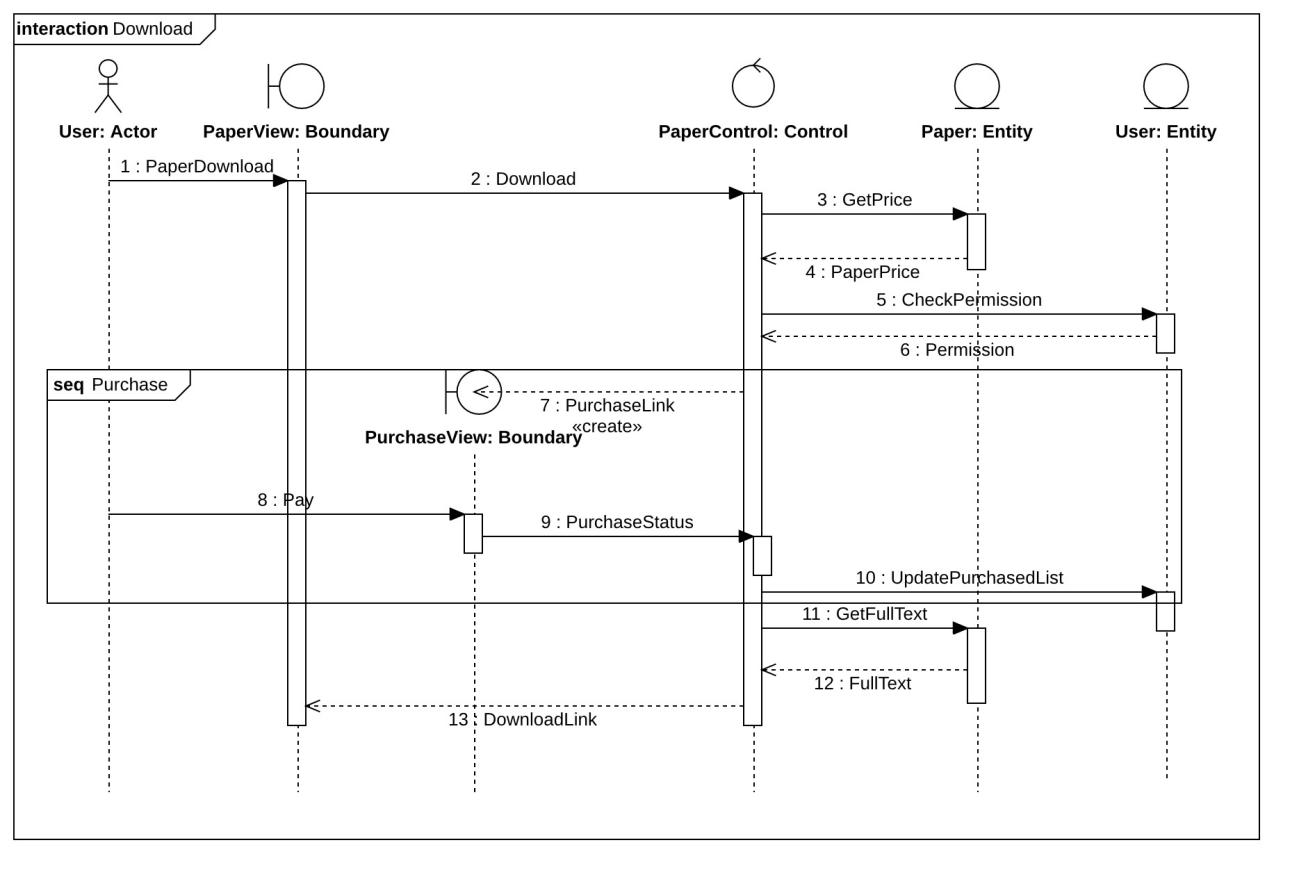
* 评论时序图：

****

**图3-8 评论时序图**

1. 用户在资源界面填写评论表单
2. 评论表单发送到资源控制类
3. 资源控制类根据表单信息创建评论对象
4. 资源控制类将评论对象添加到对应的资源评论列表内
5. 资源控制类返回信号到资源界面更新评论信息

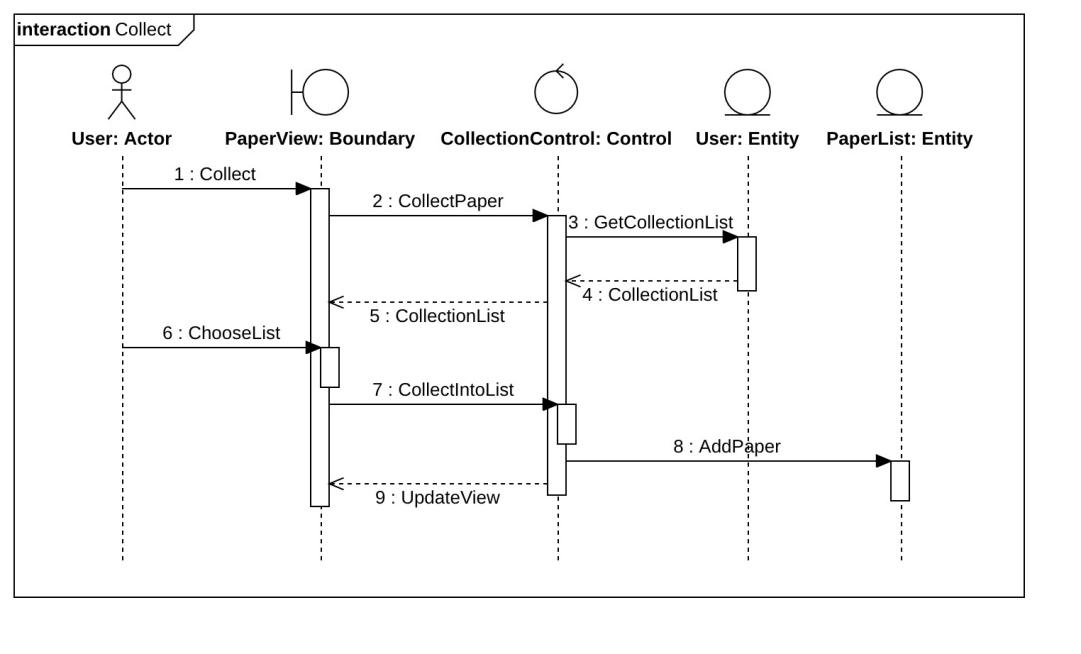
* 下载（购买）时序图：

****

**图3-9 下载（购买）时序图**

1. 用户在文献界面点击下载
2. 文献界面发送下载请求到文献控制类
3. 文献控制类查询文献数据库获取文献价格
4. 文献数据库返回文献价格
5. 文献控制类查询用户数据库判断用户是否已购买该文献
6. 用户数据库返回用户对该文献的拥有状态
7. 文献控制类跳转到第三方支付界面
8. 用户支付文献款项
9. 第三方界面返回支付状态到文献控制类
10. 文献控制类更新对应用户的文献购买列表
11. 文献控制类检索文献数据库获取全文链接
12. 文献数据库返回全文链接
13. 文献控制类返回全文链接并进行跳转

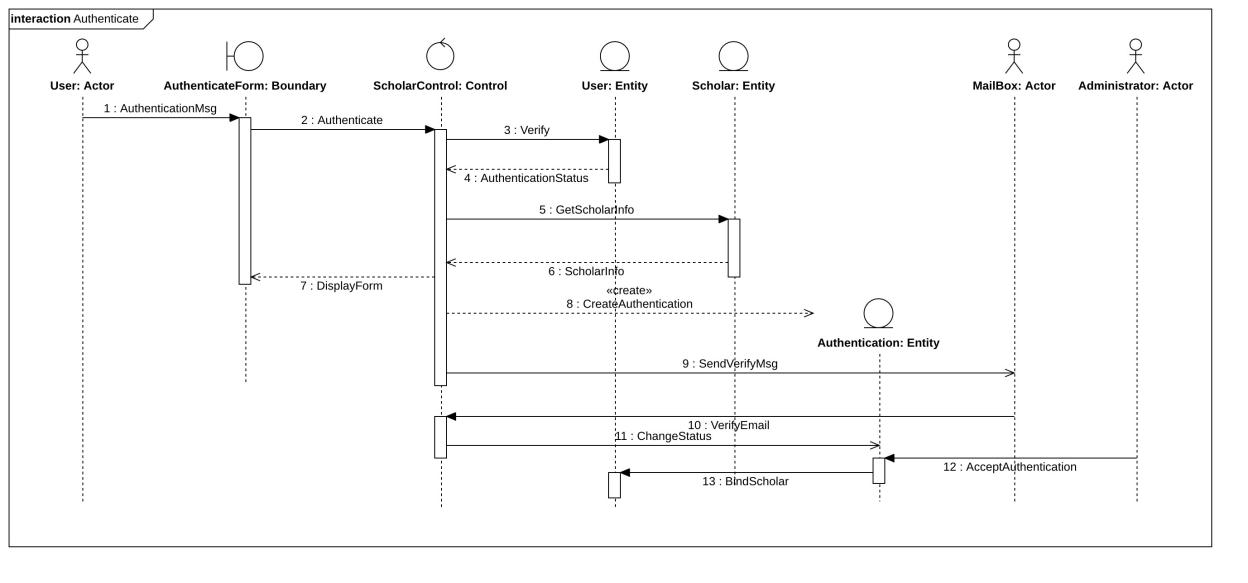
* 收藏时序图：

****

**图3-10 收藏时序图**

1. 用户在文献界面点击收藏
2. 文献界面发送收藏请求到收藏控制类
3. 收藏控制类查询用户数据库获取用户收藏列表
4. 用户数据库返回用户收藏列表
5. 收藏控制类返回收藏列表到文献界面
6. 用户选择收藏到的文献列表
7. 文献界面发送收藏请求到收藏控制类
8. 收藏控制类添加文献到该文献列表
9. 收藏控制类返回信号到文献界面更新信息

* 身份认证时序图：

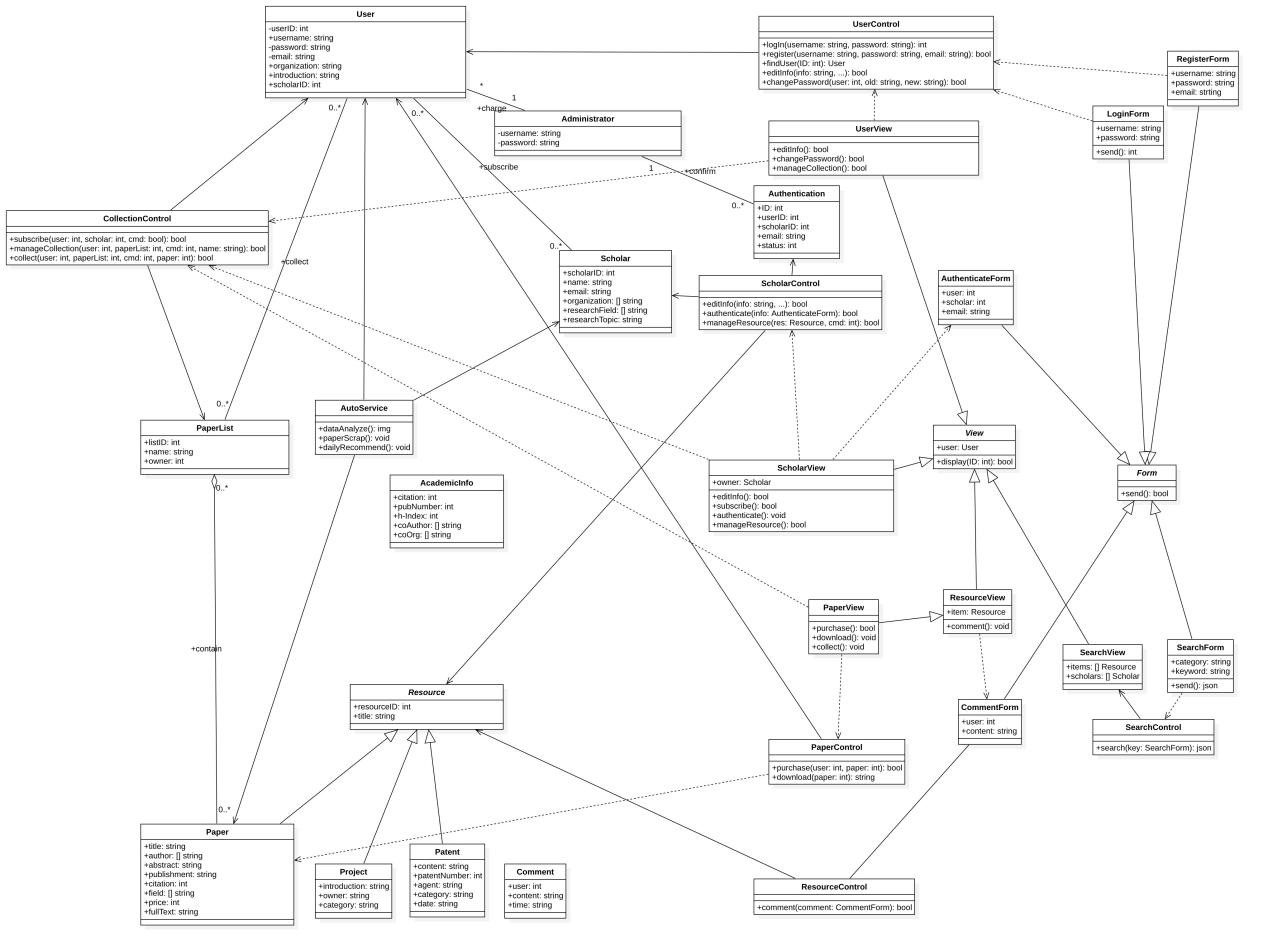


**图3-14 身份认证时序图**

1. 用户填写认证信息表格点击发送
2. 认证表格边界类发送到学者控制类
3. 学者控制类查询用户数据库确认用户认证情况
4. 学者控制类查询学者数据库获取学者相关信息
5. 学者控制类返回信息并显示
6. 学者控制类创建认证信息实体对象
7. 学者控制类发送确认信息到用户指定的邮箱
8. 用户在邮箱内点击确认邮件，调用学者控制类接口
9. 学者控制类修改认证信息状态
10. 平台管理员确认身份认证信息并修改认证信息状态
11. 认证信息绑定对应的用户并更新数据库内数据

# 4 用例分析模型

通过对用例交互模型的分析，可以进一步对静态类模型进行更新优化，依照用例交互模型中体现的信息传递过程和函数调用过程，使用专家模式，并进行整理，可以得到更加完善，涵盖整个系统业务功能和信息交互的用例分析模型类图。



**图4-1 分析类图**

分析类图中主要包含了边界类，控制类和实体类。其中边界类有用户在界面填写的用于发送信息的表单类，用于展示信息提供功能接口的界面类；控制类根据功能模块进行划分，每个控制类负责一个功能模块的全部方法，作为边界类和实体类之间信息传递的媒介，同时实现了需要经常更新的业务逻辑；实体类即需要进行数据持久化的类，为核心业务类，同时也作为最底层的逻辑实现。

|  |  |  |
| --- | --- | --- |
| **边界类** | **控制类** | **实体类** |
| *Form*  RegisterForm  LoginForm  SearchForm  CommentForm  AuthenticateForm  *View*  UserView  ScholarView  SearchView  ResourceView  PaperView | UserControl  ScholarControl  CollectionControl  SearchControl  ResourceControl  PaperControl  AutoService | User  Administrator  Authentication  Scholar  AcademicInfo  PaperList  Resource  Patent  Project  Paper  Comment |

# 5 总结

分析模型是通过对业务逻辑业务过程进行抽象整合，为架构设计提供参考的模型，由于分析模型的对象是系统的业务对象，所以进行分析模型建模首先需要对需求规格说明书中的用例进行分析，提取有效信息。其次，分析模型需要体现出各个不同的系统业务场景下，业务对象之间的信息交互，所以还需要从中提取出系统需要维护的业务对象，并根据业务需求分析总结不同的类的属性。与用例模型不同，分析模型中类间关系十分重要，从另一个角度反映了业务场景和业务需求，因此还需要通过分析模型中类之间的关系体现一些业务关系。

除了静态类分析外，对每个用例场景进行顺序图动态分析有助于更进一步地发掘系统中信息的传递过程，通过对顺序图中体现的MVC框架的整合，可以得到完整的用例分析模型，包括边界类、控制类以及静态类分析中的实体类。

虽然分析模型只是对业务场景和业务对象进行的抽象性分析，但是模型中的关系与信息传递过程都对后续的架构设计有着一定的指导作用。