



ANALYSIS MODEL

# 分析模型

校园招聘平台

摘要

本文档建立了校园招聘平台项目的分析模型。文档包括架构分析、分析模型（包括类图和时序图）以及对项目用例模型的改进等。

吴谓 | 冉启迪 | 李林洲 | 何子硕

# 目录

## Table of contents

目录 .....1

简介 .....2

    摘要 .....2

    项目目标 .....2

    项目介绍 .....3

体系结构分析 .....4

分析模型 .....7

    简历子系统 .....7

        提交简历 .....8

        查看简历 .....9

    论坛子系统 .....10

        发表帖子 .....11

        管理论坛 .....12

        回复帖子 .....13

    招聘信息子系统 .....14

        发布招聘信息 .....15

        查看招聘信息 .....16

    注册登录子系统 .....17

        注册 .....18

        登录 .....18

关于文档 .....20

    用例模型更新 .....20

    参考文献 .....20

    小组成员贡献 .....21

图表索引 .....22

# 简介

## Brief introduction

### 摘要

#### Abstract

本文档建立了校园招聘平台项目的分析模型。文档包括架构分析、分析模型（包括类图和时序图）以及对项目用例模型的改进等。本文档供项目的开发者、管理者及使用者阅读，以便对该项目的开发和使用获得更加具体和深入的认识。

### 项目目标

#### Project objectives

本项目的目标是开发一个致力于校园招聘信息发布，增进学生与学生、学生与用人单位之间沟通交流的校内平台，旨在为学生收集招聘与实习信息，并为企业提供学生反馈，在学生和用人单位之间搭起信息交互的桥梁。同时本平台拟设立论坛模块，为学生之间的信息交流和经验分享创造机会。

大学的职责与使命之一是为社会与国家输送人才，而找工作也是毕业生们得以脱离大学，在社会上安身立命之基础，因而学生的就业问题一直以来是学生和校方所关心的重要问题，其中，校招无疑是学生就业的最重要的渠道。然而我们小组发现，虽然有众多企业与用人单位会在每年的招聘季前往同济大学开展宣讲会、招聘会等活动，但由于缺少统一的信息发布平台，用人单位的宣传方式也五花八门，包括但不限于：微信公众号推送、QQ 群聊发布信息、校园内贴海报等等。繁多冗杂的宣发渠道往往不利于学生和校方集中获取信息，于企业而言一来是增加宣发成本，二来也可能失去了发掘人才的机会。因此，本项目投入使用之后，可以有效地方便大学生找工作和应聘，也能帮助用人单位招聘时更加方便地找到需要的人才。

## 项目介绍

### Project introduction

本系统中主要有三种角色类型，分别是：学生、用人单位、管理员；系统分为四个子系统：注册/登录子系统、招聘信息子系统、论坛子系统、简历子系统。

在注册/登录系统中，学生、用人单位以及管理员可以注册不同类型的账号并登录到系统中。

在招聘信息子系统中，用人单位在经过资历认证后可以发布各种类型的招聘信息，包括校招信息、实习信息以及其他类型的信息，也可以对已发布的招聘信息进行管理，包括修改或删除等操作。管理员会对招聘信息进行审核，决定是否予以发布。学生可以查看用人单位发布的招聘信息。

在论坛子系统中，学生可以发布帖子，帖子在经过管理员的审核之后会发布到论坛中供所有人可见；也可以在已发布的帖子下进行跟帖评论。管理员会对学生提交的帖子进行审核；对于已发布的帖子，可以予以删除。

在简历子系统中，学生可以向用人单位投递简历，用人单位可以查看学生投递的简历。

# 体系结构分析

## Architectural analysis

本系统从功能模块出发，将系统分为 4 个子系统：注册/登录子系统、招聘信息子系统、简历管理子系统、论坛子系统。

为了降低前后端的耦合性，提高开发效率的同时更加便于后期维护、优化，本系统将采用前后端分离技术，通过约定的 API 交互接口进行通信。前端主要为 Web 网页及 Android、IOS 端的应用程序页面，后端则装载于云端服务器、负责处理前端需求并传回数据。前后端分离技术，使得在开发期间实现前后端完全独立的进行并行开发及测试，在投入运行后也分离进行部署，能够极大的提高开发效率，增强代码的可维护性，以便应对多变复杂的应用需求发展。

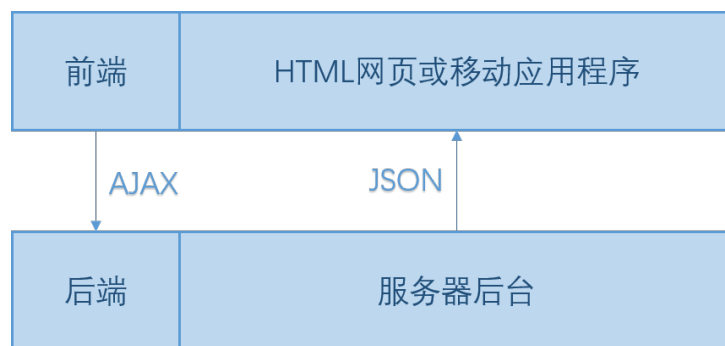


图1 前后端分离图示

本系统采用的是 MVVM 模式（Model-View-ViewModel），是在经典的 MVC 模式基础上迭代升级而来的一种新兴架构模式。MVVM 模式与 MVC 模式最大的不同在于有一个 ViewModel 层，相当于 View 层与 Model 层之间的中转站，负责转换 Model 中的数据使其更加容易管理和使用，同时与 View 视图层进行双向数据绑定，能够高效的在 View 和 Model 之间实现数据的传输。View 层就是用户实际看到的界面，主要用 HTML 来构建。Model 层则是部署在服务器端的数据库和相关的业务逻辑。而 ViewModel 是由前端开发人员来进行开发维护的一个视图数据层，在 ViewModel 层，需要对前端从后端获取的 Model 数据进行处理，做二次封装，使其更加符合 View 层所使用的预期数据。

ViewModel 层不仅存储了页面的状态数据，同时也封装了 View 层的页面交互数据，这样的结构使得 ViewModel 层能够完整的描述 View 层，从而实现双向数据绑定。ViewModel 层的存在，使得 View 完全不需要展现 Model 层的数据，而直接使用 ViewModel 层的数据，实现了 View 层和 Model 层的完全解耦，即前后端分离。

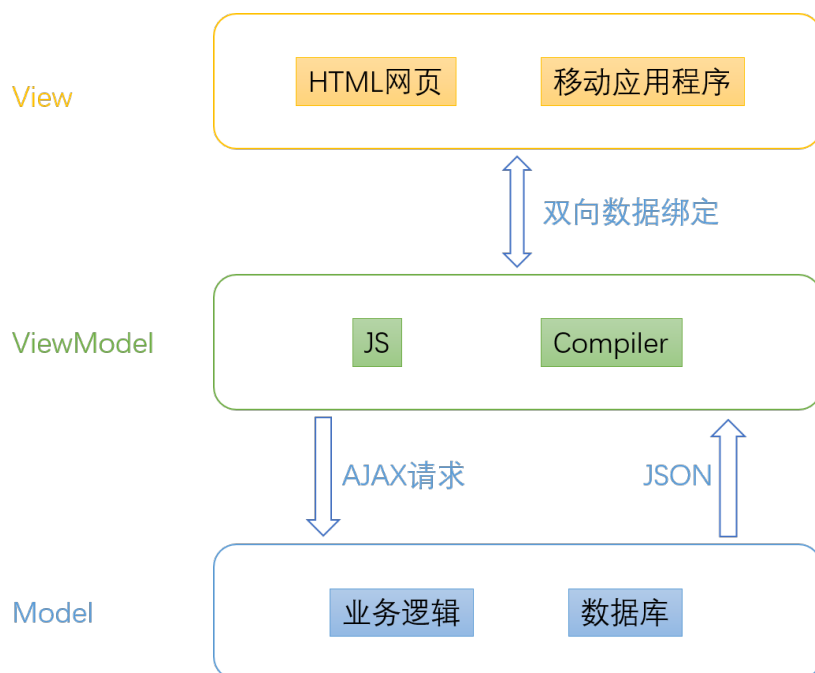


图2MVVM 模式组成图

从分层体系结构上看，本系统是经典的四层结构：应用程序接口层、业务逻辑层、中间件层、系统软件层，各层的具体含义如下：

**应用程序接口层（Application Interface）**：用户提供此层所提供的各种 API 接口与系统进行交互，下层结构从本层获取用户请求或指令。

**业务逻辑层（Business Logic）**：负责各系统的各类业务服务逻辑管理，接受到上层的请求后根据相应的逻辑从下层结构调取相应数据进行处理。

**中间件层（Middleware）**：主要对数据库的访问进行管理，上层结构必须通过中间件层从数据库获取数据。

系统软件层（System Software）：即系统底层的逻辑结构，主要为平台的数据库系统。

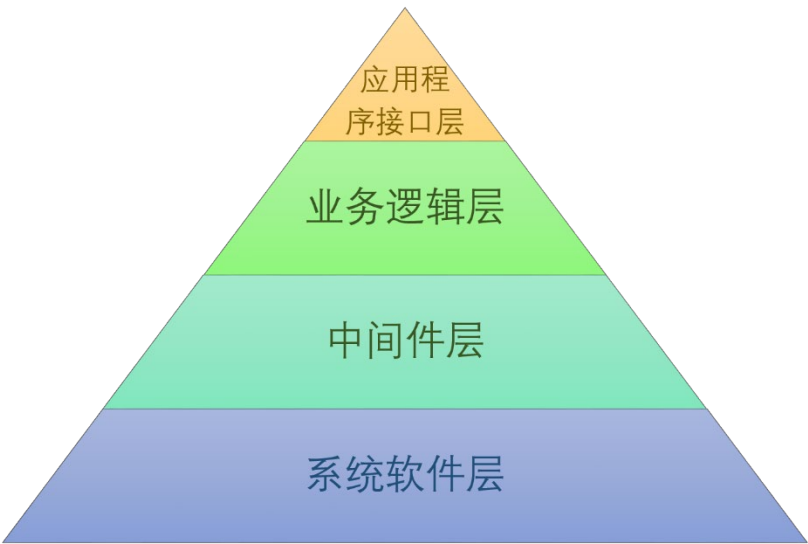


图3 四层体系结构

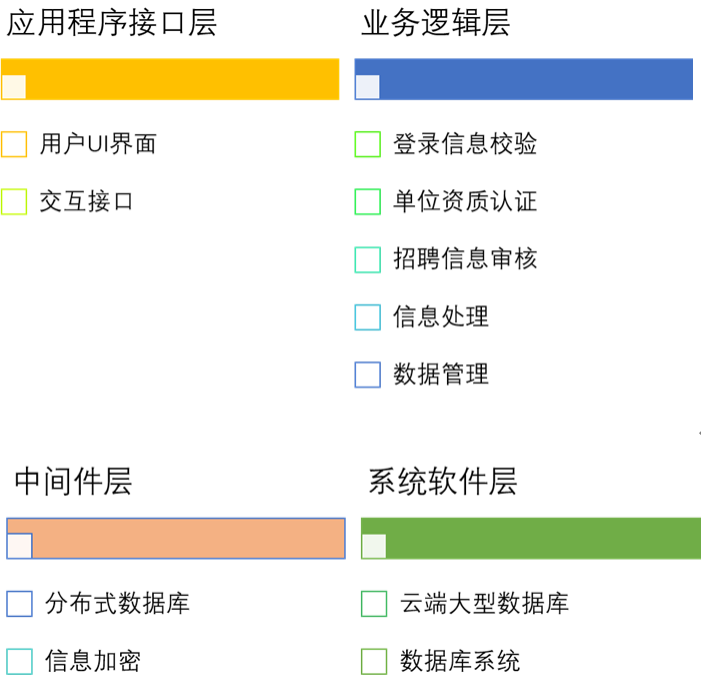


图4 各层结构的组成

# 分析模型

Analysis model

## 简历子系统

Resume subsystem

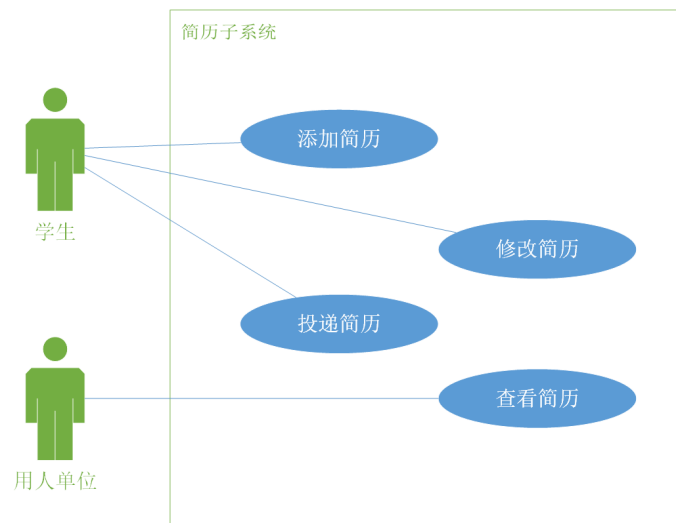


图5 简历子系统用例图

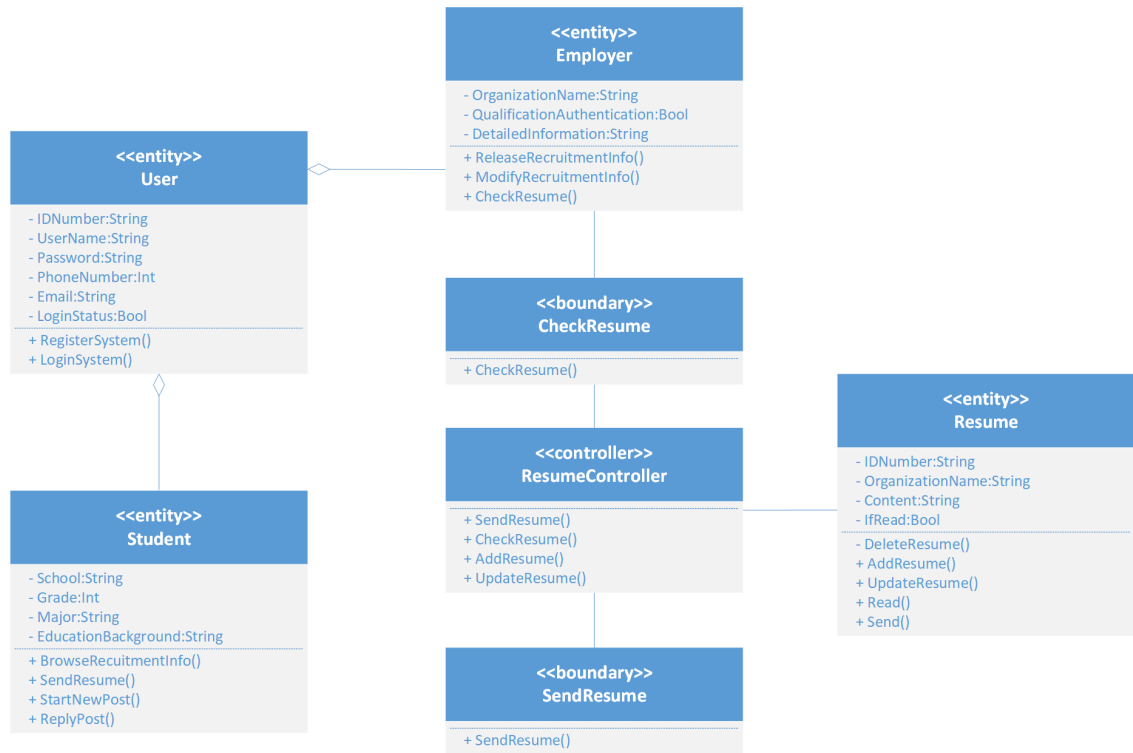


图6 简历子系统类图



该子系统的主要功能是“学生用户提交简历”和“用人单位查看简历”，即涵盖了原用例图中的“提交简历”与“查看简历”两个用例。前者又包含了如下四个类：学生用户类(Student)，提交简历边界类(SendResumeBoundary)、简历控制类(ResumeController)以及简历实体类(Resume)。后者则涉及用人单位类(Employer)，查看简历类边界类(CheckResumeBoundary)，简历控制类和简历实体类。

### 提交简历

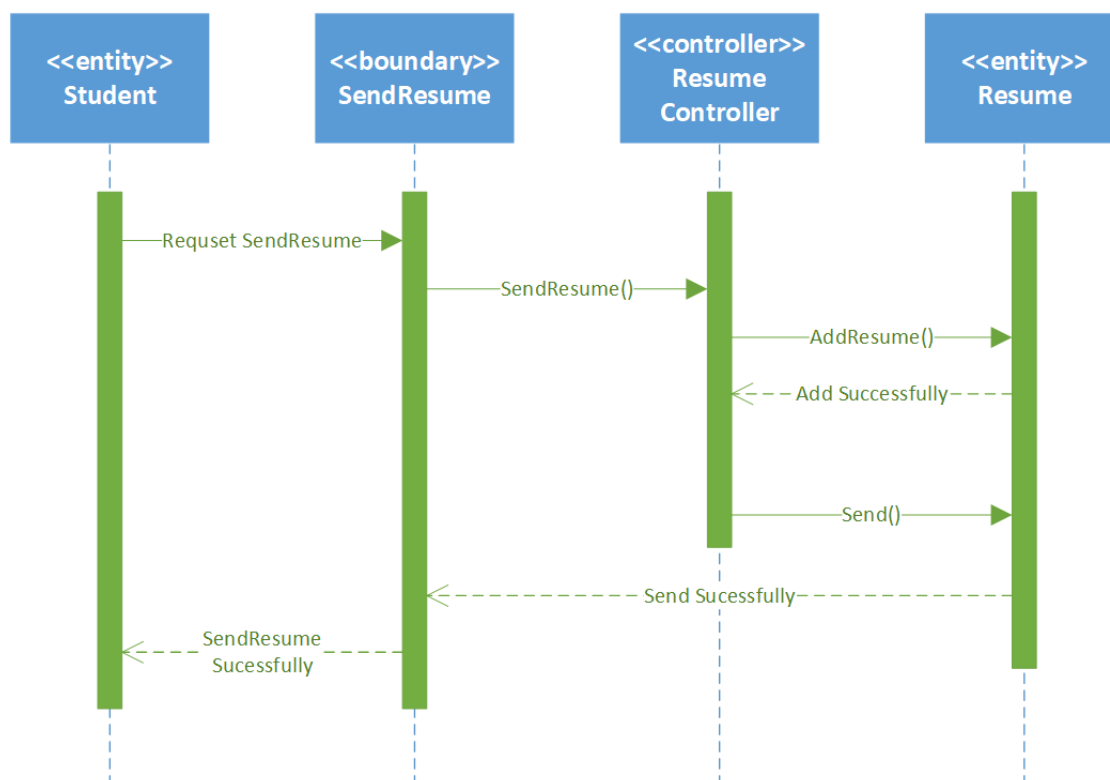


图7 提交简历序列图

当学生用户需要提交简历时，向投递简历的边界类递交简历内容，该类将简历内容的相关参数传递给简历控制类。控制类将该信息整理送入简历实体类，提交简历完成。

## 查看简历

当用人单位需要查看简历时，通过查看简历边界类发出申请，此边界类再与简历控制类交互，传递该用人单位的具体编号，并由简历控制类根据编号从简历实体类中调出相应简历，并送回给用人单位，查看简历完成。

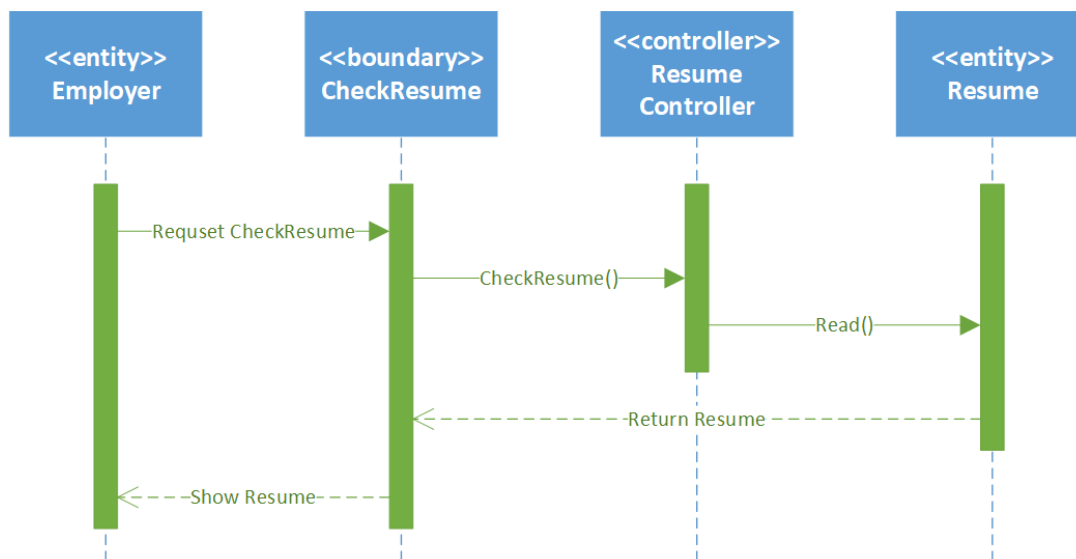


图8 查看简历序列图

# 论坛子系统

Forum subsystem

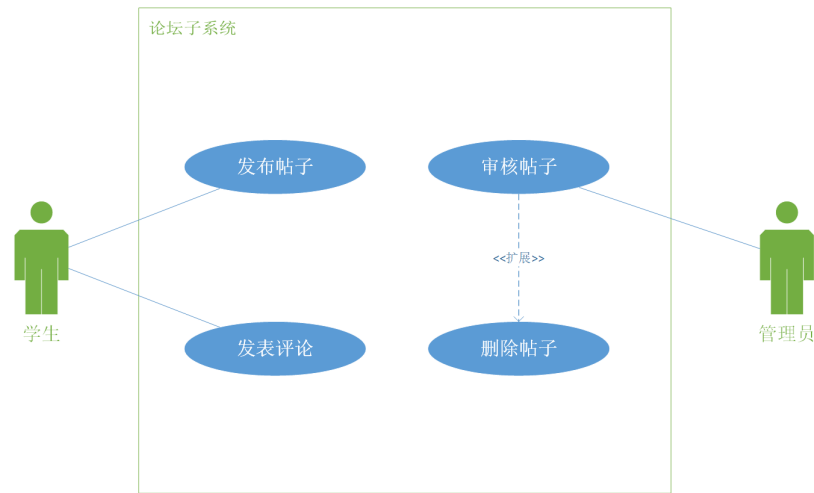


图9 论坛子系统用例图

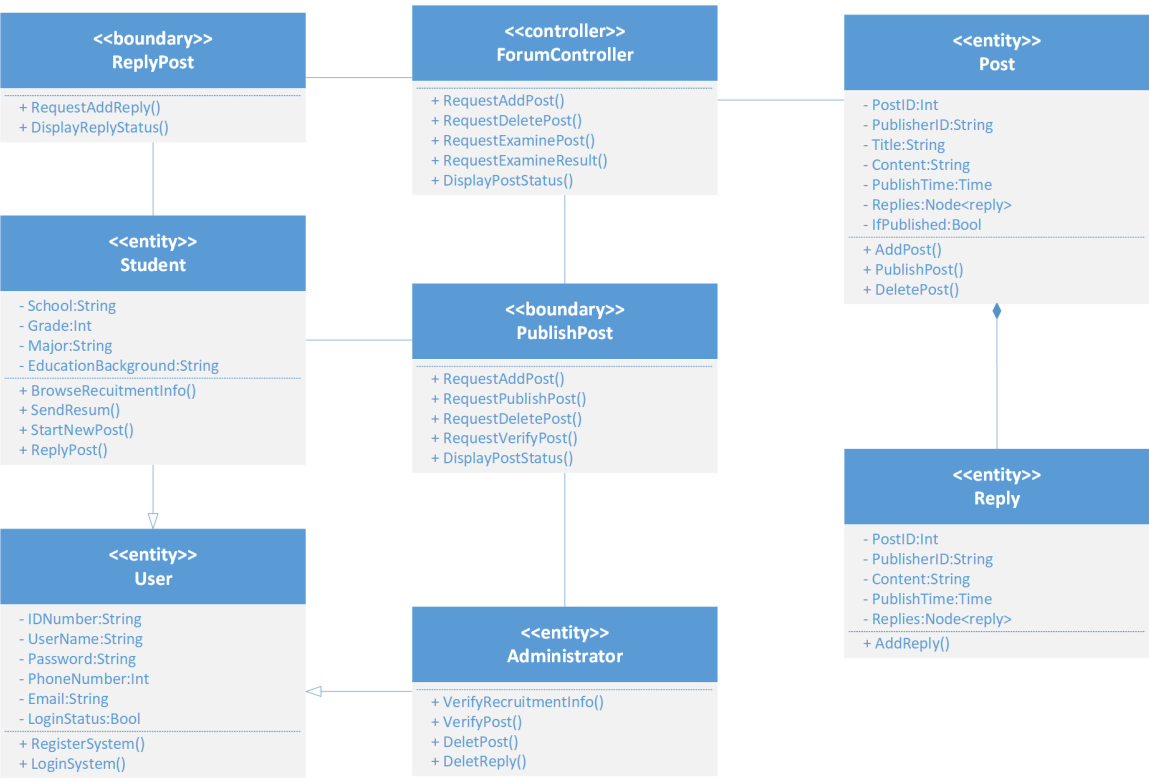


图10 论坛子系统类图

论坛子系统主要实现了学生在论坛中发表帖子、发表评论和管理员审核帖子、删除帖子的功能。论坛子系统中有一个控制类：ForumController，两个边界类：

ReplyPost、PushlishPost，五个实体类：Student、Administrator、User、Reply、Post，其中 Post 与 Reply 之间是组合关系，User 和 Student、Administrator 之间是泛化关系。

### 发表帖子

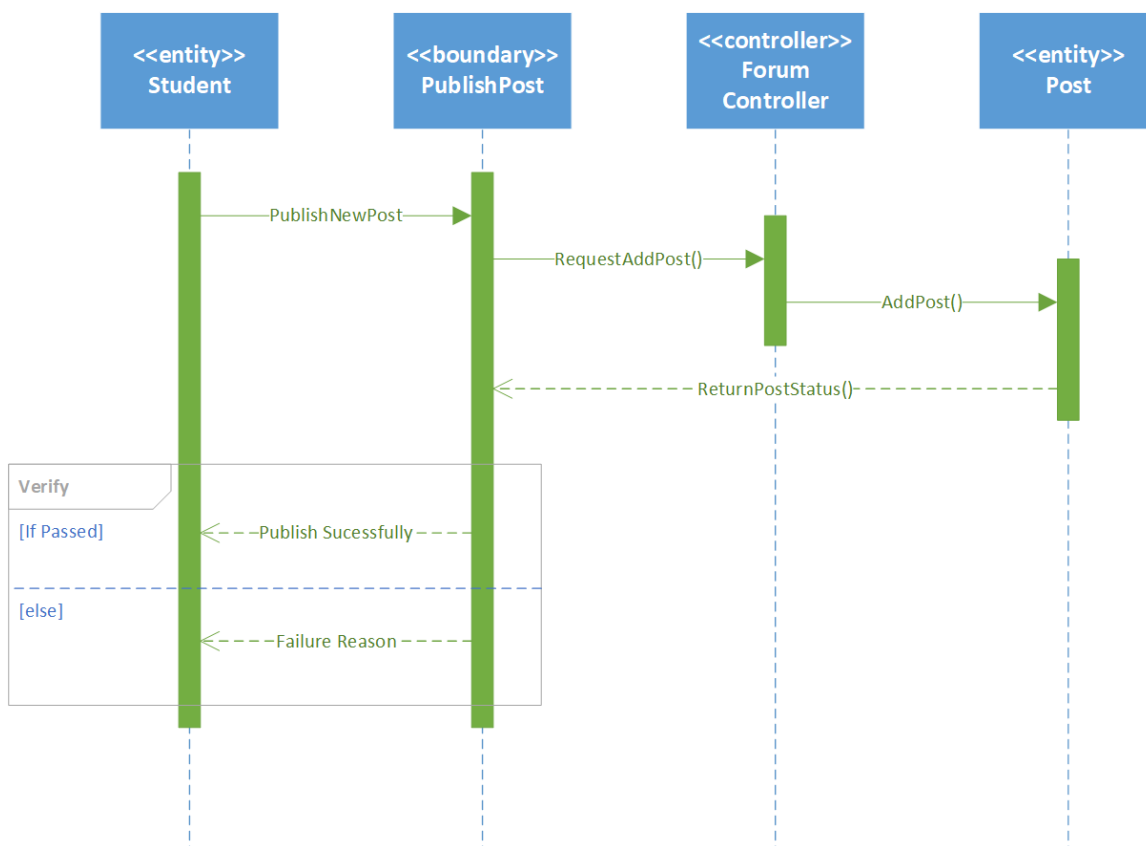


图11 发布帖子序列图

学生首先向发表帖子边界类（PublishPost）发出发表帖子（PublishNewPost）的请求，然后帖子边界类向论坛控制类（ForumController）发出添加帖子的请求（RequestAddPost），论坛控制类对帖子数据库（Post）进行添加帖子的操作（AddPost），帖子数据库返回帖子状态（ReturnPostStatus）给发表帖子边界类。若帖子状态为发表成功，则边界类向学生展示发表成功（Publish Successfully）的信息；若发表失败，则向学生展示失败原因（Failure Reason）。

管理论坛

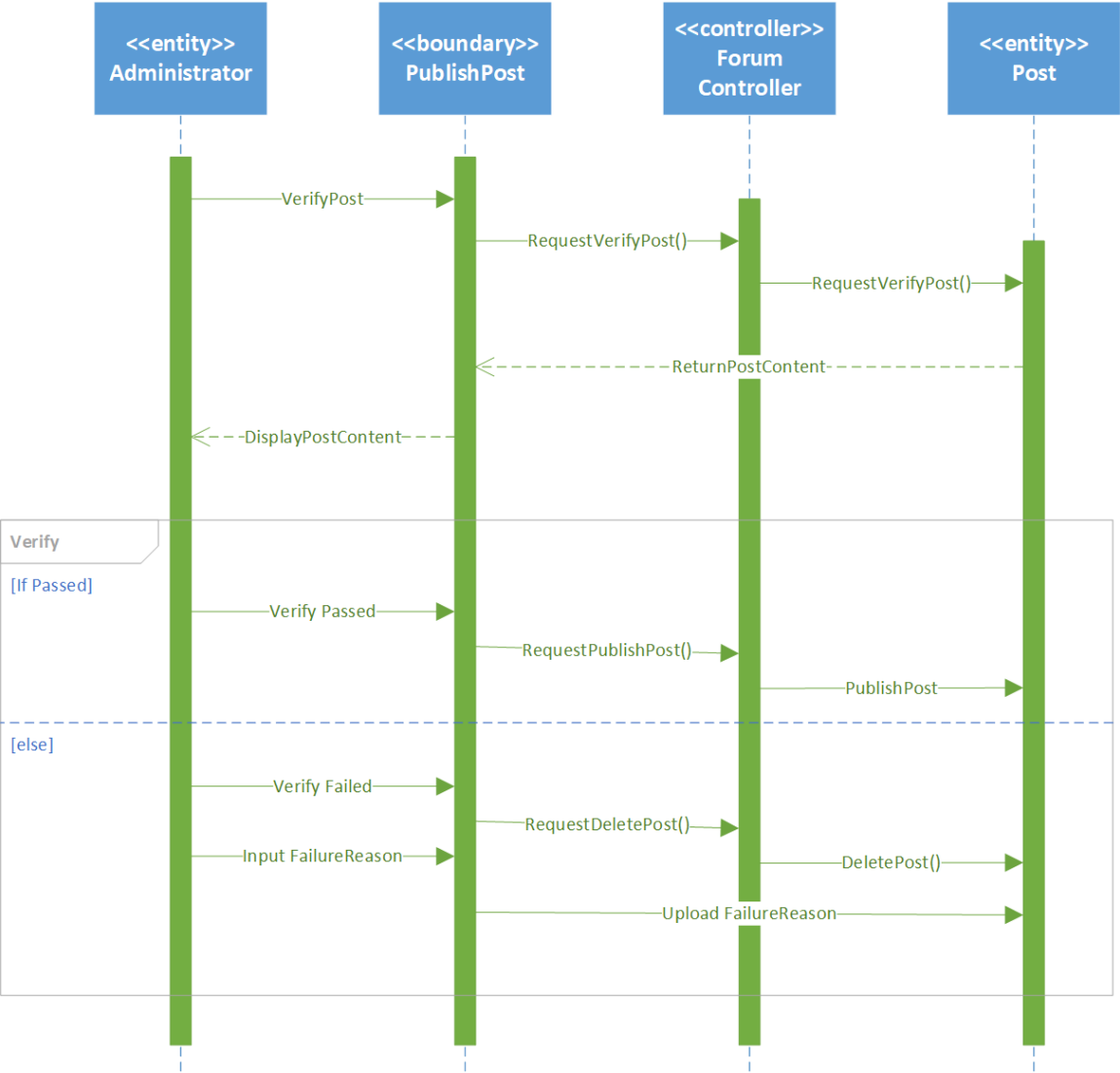


图12 管理论坛序列图

管理员首先向帖子边界类（PublishPost）发出审核帖子（VerifyPost）的请求，然后帖子边界类向论坛控制类（ForumController）发出审核帖子的请求（RequestVerifyPost），论坛控制类对帖子数据库（Post）发出审核帖子的请求（RequestVerifyPost），帖子数据库返回帖子内容（ReturnPostContent）给发表帖子边界类。边界类将帖子内容展示给管理员（DisplayPostContent）。管理员选择审核是否通过，若通过，则边界类向控制类发出发布帖子的请求（RequestPublishPost），控制类对帖子数据库进行发表帖子（PublishPost）操作。

若审核不通过，则输入原因（InputFailureReason），边界类向控制类发出删除帖子的请求（RequestDeletePost），控制类对帖子数据库进行删除帖子（DeletePost）的操作。另外，边界类将失败原因上传到帖子数据库中（UploadFailureReason）。

### 回复帖子

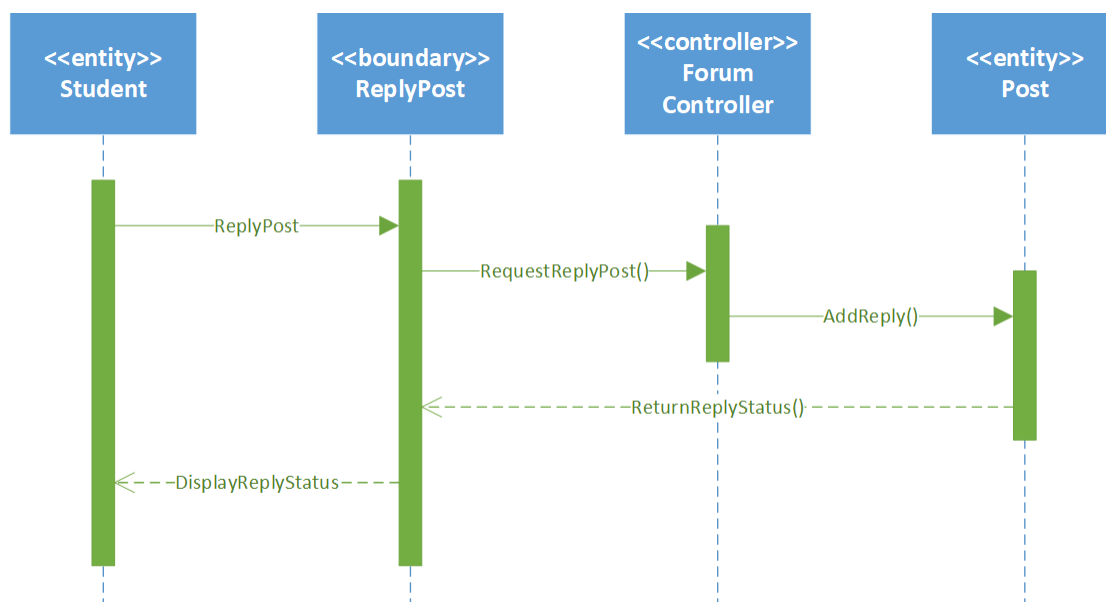


图13 回复帖子序列图

学生首先向发表回复边界类（PublishReply）发出发表回复（ReplyPost）的请求，然后边界类向论坛控制类（ForumController）发出回复帖子的请求（RequestReplyPost），论坛控制类对回复数据库（Reply）进行添加回复的操作（AddReply），回复数据库返回回复状态（ReturnReplyStatus）给边界类。最后边界类向学生展示回复状态（DisplayReplyStatus）。

## 招聘信息子系统

### RecruitmentInfo subsystem

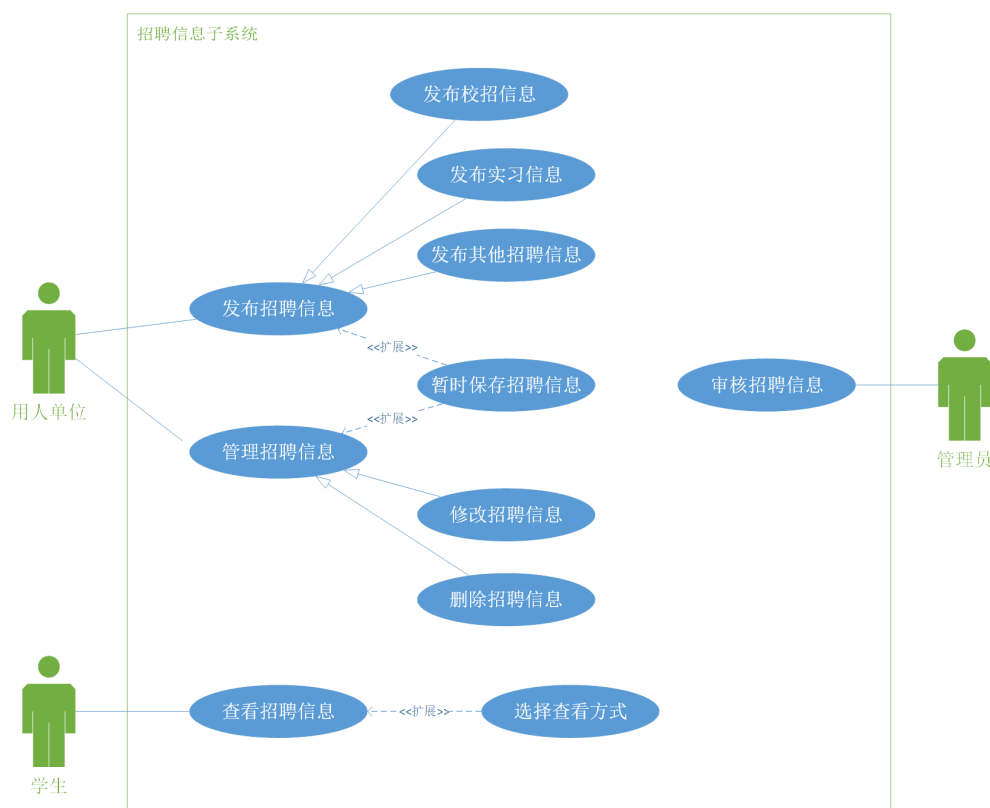


图14 招聘信息子系统用例图

招聘信息子系统的主要功能是用人单位用户发布和管理招聘信息，学生用户查看招聘信息。子系统涉及的实体类有招聘信息类（RecruitmentInfo）、用人单位类（Employer）、学生类（Student）和管理员类（Administrator），其中招聘信息类特化为3个类，分别是校园招聘信息（ScoolRecuitmentInfo）、实习信息（IntershipRecuitmentInfo）和其他招聘信息（OtherRecuitmentInfo）。学生可以通过边界类 BrowseRecuitmentInfo 浏览招聘信息。用人单位可以通过 ModifyRecuitmentInfo 边界类修改招聘信息，也可以通过 ReleaseRecuitmentInfo 边界类进行添加和提交招聘信息。招聘信息提交后，由管理员通过 VerifyRecuitmentInfo 边界类进行审核招聘信息。子系统的核心是招聘信息控制类（RecuitmentController），它控制所有边界类对子系统的修改以及招聘信息对不同用户的展示方式。

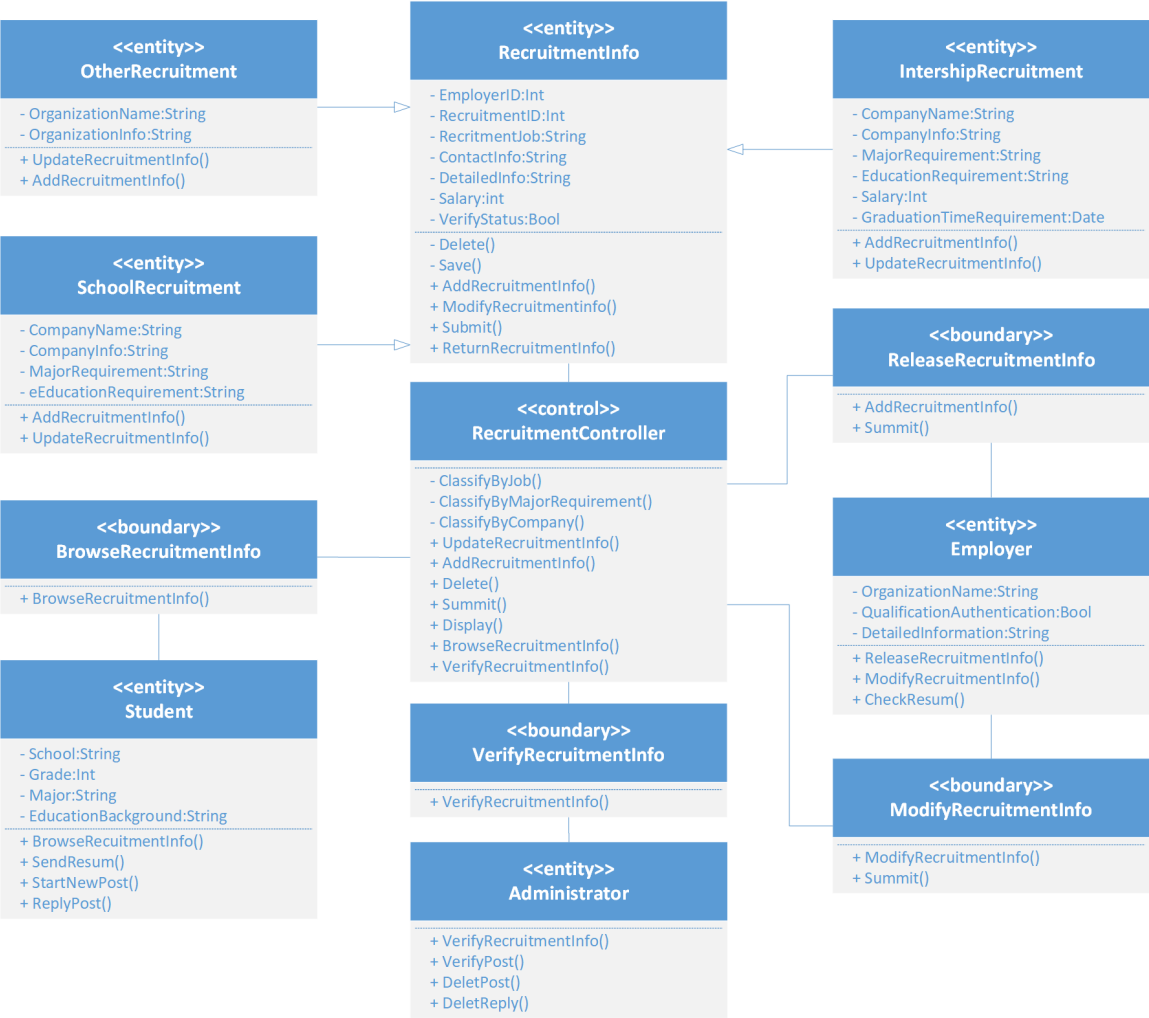


图15 招聘信息子系统类图

发布招聘信息

首先由 **Employer** 类输入招聘信息至 **ReleaseRecruitmentInfo** 边界类，再由边界类将招聘信息添加至 **RecruitmentInfo** 类，并暂时保存。然后提交至 **Administrator** 进行审核，审核结果返回审核结果，审核成功则向用户返回成功信息并发布，审核不通过则返回失败原因。



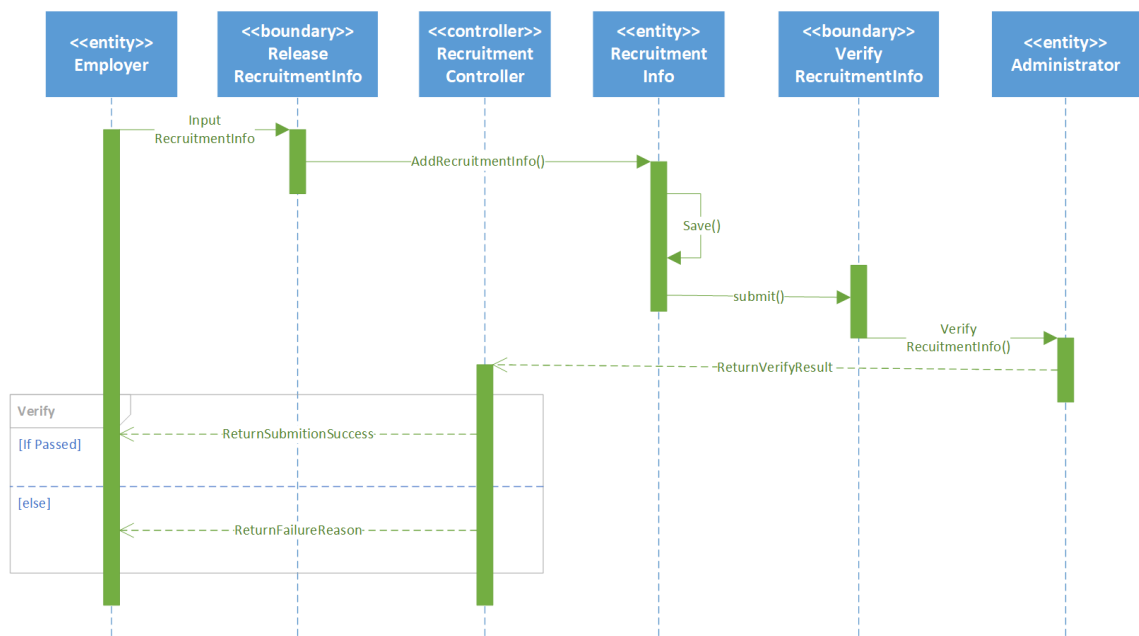


图16 发布招聘信息序列图

查看招聘信息

首先由 Student 类选择浏览模式到 BrowseRecruitment 边界类，然后控制类按浏览模式对 RecruitmentInfo 进行分类，最后返回招聘信息到 Student 类。

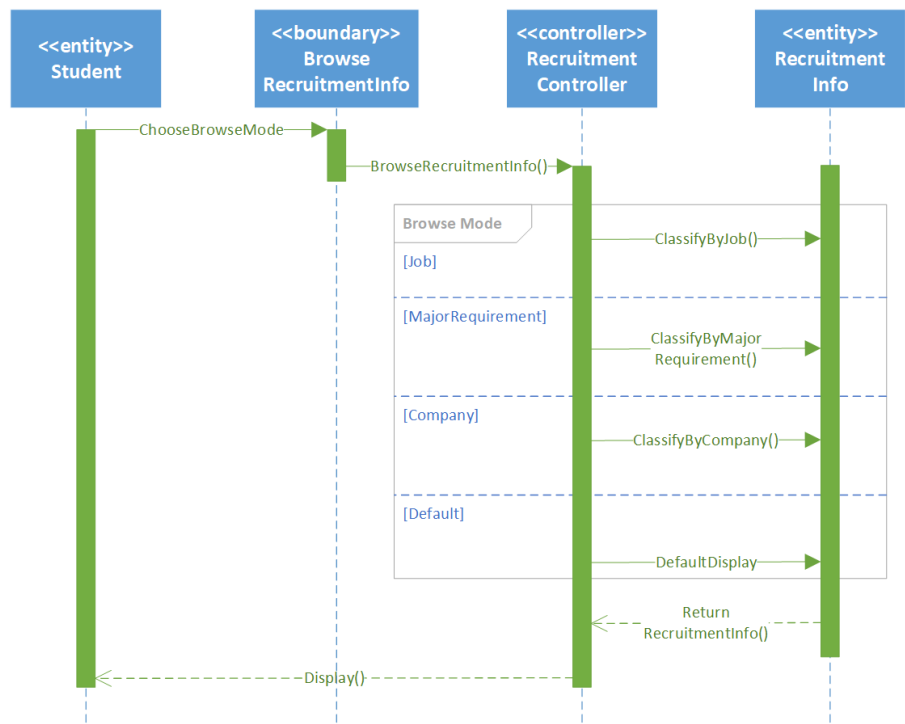


图17 查看招聘信息序列图

# 注册登录子系统

Sign up & login subsystem

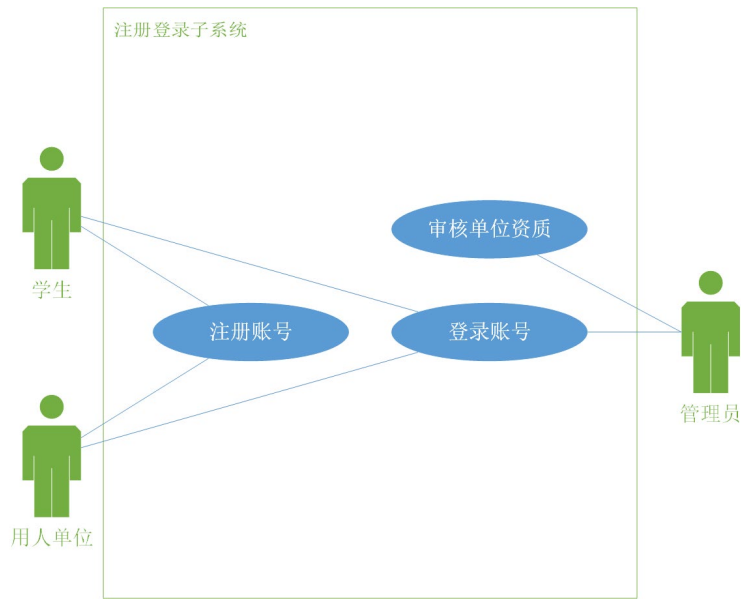


图18 注册登录子系统用例图

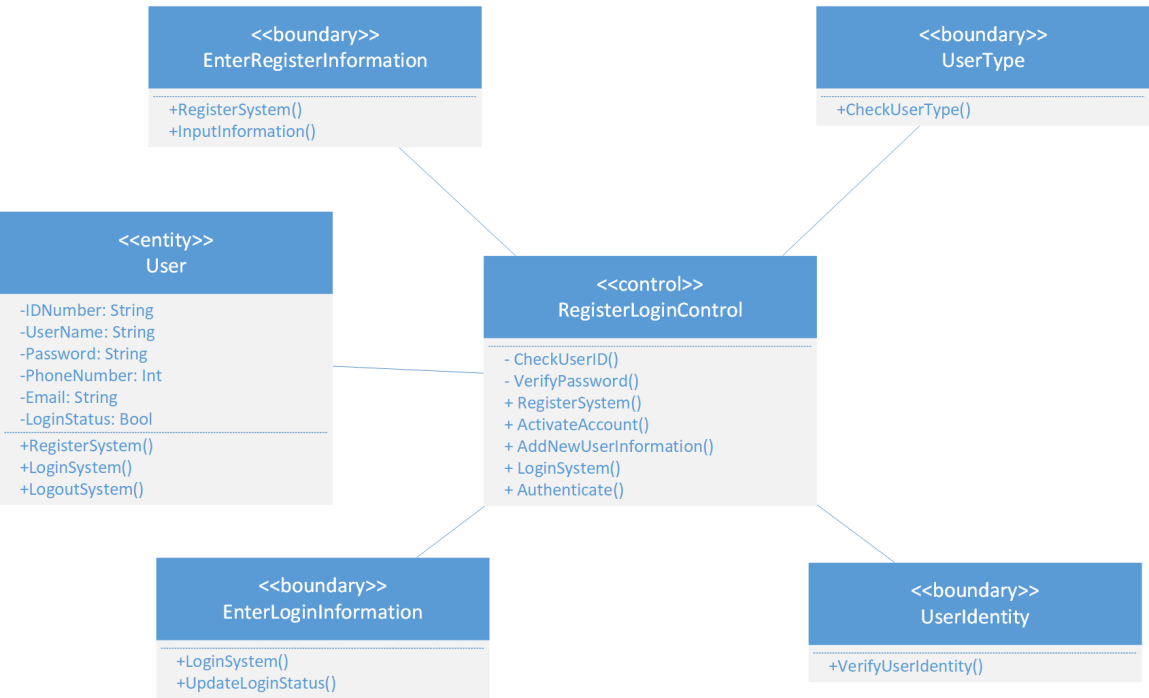


图19 注册登录子系统类图

注册/登录子系统负责用户注册和登录平台账号，该子系统涉及的实体类有用户实体类（User）以存储用户的相关信息，控制类为注册登录控制类

（RegisterControl）用于控制用户的注册和登录功能，边界类包括注册信息录入类（EnterRegisterInformation）、登录信息录入类（EnterLoginInformation）、用户类型类（UserType）、用户身份类（UserIdentity）。

## 注册

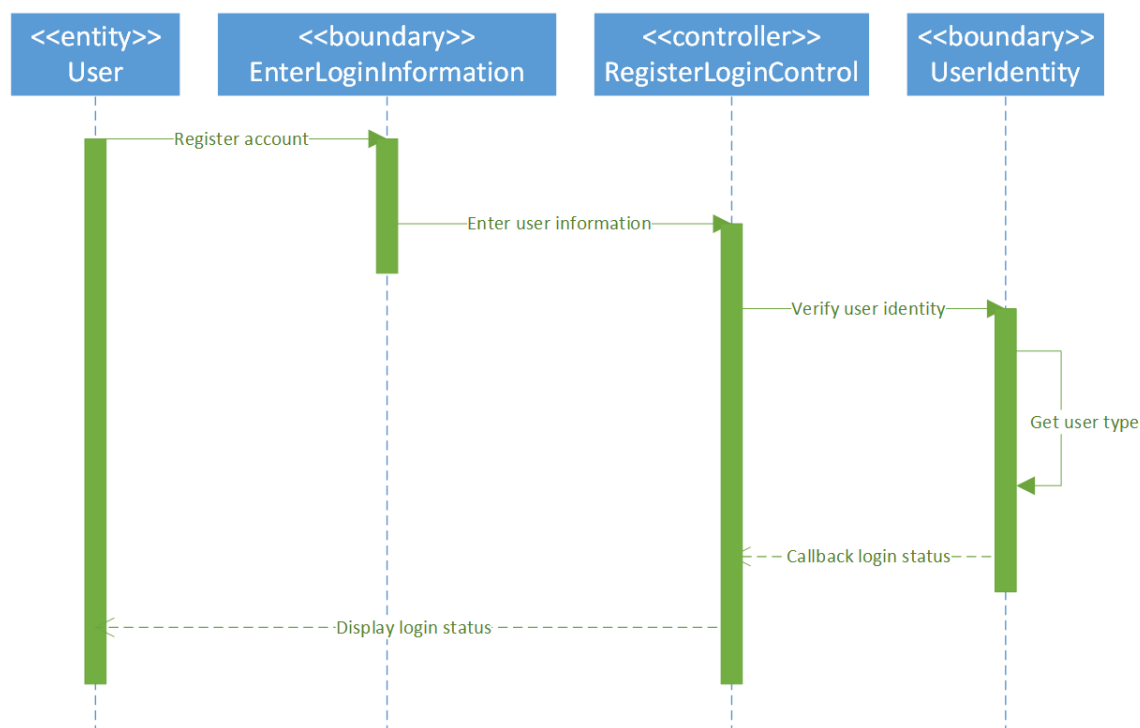


图20 注册序列图

在注册过程中，用户向注册信息录入类发送注册信息录入请求，录入所需信息后，然后向注册登录控制类发送注册请求，注册登录控制类将用户录入的信息传递给用户类型类，由用户类型类识别用户类型后反馈给注册登录控制类，由注册登录控制类授予用户登录权限并将用户的信息录入数据库，最终反馈注册结果给用户。

## 登录

在登录过程中，用户向登录信息录入类发生登录信息录入请求，录入用户名/ID/电话号码、密码后，向注册登录控制类发送登录请求，注册登录控制类将用户

录入的用户名/ID/电话号码、密码信息传递给用户身份类，由用户身份类从数据库中尝试查询用户信息，并验证 ID 密码是否匹配，将结果反馈给注册登录控制类，由注册登录控制类改变用户登录状态并将登录结果反馈给用户。

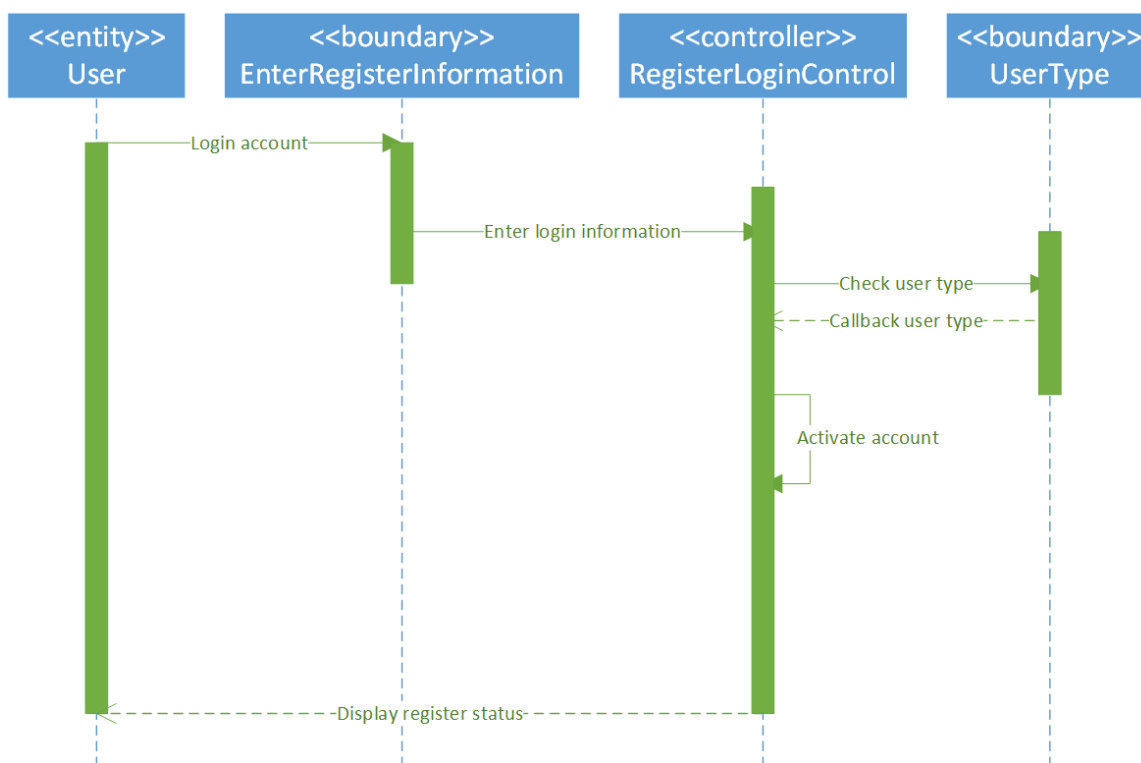


图21 登录序列图

# 关于文档

## About the document

## 用例模型更新

### Updated use case model

在之前的需求分析模型中，我们已经完成了对软件需求的分析，通过用例图和用例规约初步建立了用例模型，明确了系统的主要功能与组成。自上一次报告以来，我们对系统的需求进行了进一步的深入分析，针对用例模型中存在的一些瑕疵进行了改进。

将模型细分成为 4 个子系统，即注册/登录子系统、招聘信息子系统、论坛子系统、简历子系统，以方便类图、时序图等分析模型的构建。

## 参考文献

### References

[1]胡荷芬, 曹德胜. UML 系统建模基础教程. 清华大学出版社, 2021

本书详细介绍 UML 系统建模的思想和具体方法，内容包括面向对象设计、UML 通用知识点概述、Rational 统一过程、Rational Rose 的安装和作用、使用 Rose 设计 UML、用例图、类图与对象图、序列图、协作图、活动图、包图、构件图和部署图、状态图，后以典型案例详解 UML 各种技术的综合应用。通过阅读本书，我们小组对 UML 建模建立了更为清晰的概念体系，同时掌握了 UML 建模的基本方法。

[2]郑伟. 基于 UML 的车站联锁软件建模研究[J]. 科学技术创新, 2021(08): 73-74.

该论文在研究联锁逻辑关系的基础上, 采用 UML 语言及建模工具实现对联锁软件系统的建模研究, 分析了信号设备的基本状态变化及在进路处理中的状态变化, 完成了室外三大件（信号机、道岔、轨道区段）和进路之间的静态结构建模及数据结构分析, 建立了进路处理过程的动态模型。从中，我们团队进一步了解了 UML 建模在实际场景和工程实践中的应用方法与步骤。

小组成员贡献

Contributions of team members

李林洲	冉启迪	何子硕	吴渭
绘制子系统用例图、文档排版	体系结构分析	文档简介	参考文献

分析模型			
李林洲	冉启迪	何子硕	吴渭
招聘信息子系统	注册登录子系统	论坛子系统	简历系统

1953910 李林洲 25%

1953290 冉启迪 25%

1951714 何子硕 25%

1951574 吴谓 25%

# 图表索引

Diagram index

图 1 前后端分离图示.....4

图 2MVVM 模式组成图.....5

图 3 四层体系结构.....6

图 4 各层结构的组成.....6

图 5 简历子系统用例图.....7

图 6 简历子系统类图.....7

图 7 提交简历序列图.....8

图 8 查看简历序列图.....9

图 9 论坛子系统用例图..... 10

图 10 论坛子系统类图 ..... 10

图 11 发布帖子序列图 ..... 11

图 12 管理论坛序列图 ..... 12

图 13 回复帖子序列图..... 13

图 14 招聘信息子系统用例图 ..... 14

图 15 招聘信息子系统类图 ..... 15

图 16 发布招聘信息序列图 ..... 16

图 17 查看招聘信息序列图 ..... 16

图 18 注册登录子系统用例图 ..... 17

图 19 注册登录子系统类图 ..... 17

图 20 注册序列图..... 18

图 21 登录序列图..... 19