

软件架构设计说明书

文档修改记录

| 版本号 | 版本描述 | 责任人 | 日期 | 修改内容 |
|------|------|-----|------------|----------------------|
| V1.0 | 初始编制 | 柯杰豪 | 2018/03/13 | 文档起草 |
| V2.0 | 修改 | 曾钊 | 2018/03/19 | 文档修改 |
| V3.0 | 修改 | 张坤伟 | 2018/03/19 | 文档修改 |
| V4.0 | 修改 | 张坤伟 | 2018/03/21 | 添加总体流程，功能描述， 系统用例 |
| V5.0 | 增加 | 陈捷维 | 2018/04/10 | 增加进程视图 |
| V6.0 | 增加 | 杜英铨 | 2018/04/17 | 增加类图 |
| V7.0 | 修改 | 陈捷维 | 2018/04/20 | 修改进程视图 |
| V8.0 | 修改 | 杜英铨 | 2018/04/24 | 修改类图 |

目录

| | | |
|----|-----------------|----|
| 一、 | 简介 | 27 |
| 1. | 编写目的 | 27 |
| 2. | 文档范围 | 28 |
| 3. | 定义 | 28 |
| 4. | 参考资料 | 28 |
| 二、 | 架构表示方式 | 29 |
| 三、 | 架构设计目标与约束 | 29 |
| 1. | 关键功能需求 | 30 |
| 2. | 关键质量需求 | 33 |
| 3. | 开发策略 | 33 |
| 四、 | 用例视图 | 34 |

| | |
|----------------------|----|
| 1. 用例视图 | 34 |
| 五、 设计框架..... | 35 |
| 1. 概述 | 35 |
| 2. 逻辑视图 | 36 |
| 六、 实施视图..... | 37 |
| 1. 概述 | 37 |
| 2. 实施模型视图(设计类图)..... | 38 |
| 七、 进程视图..... | 43 |
| 1. 概述 | 43 |
| 2. 角色进程视图 | 43 |
| 八、 部署视图..... | 46 |
| 1. 概述 | 46 |
| 2. 部署方案视图 | 47 |

一、简介

1. 编写目的

本文档全面与系统地表述了企业用户系统的构架,并通过使用多种视图来从不同角度描述本系统的各个主要方面,以满足企业用户系统的相关涉众(客户、设计人员等)对本系统的不同关注焦点和需求。本文档记录并表述了系统架构的设计人员对系统构架方面做出的重要决策。

项目经理将根据构架定义的构件结构制定项目的开发计划;程序设计员将据此进行各构件的详细设计;测试设计员按照构架设计系统的总体测试框架;另外构架文档还用于指导各构件的实施、集成及测试。

本文档的预期阅读人员为项目经理、程序设计人员、测试人员和其他有关的工作人员。

2. 文档范围

本软件架构说文档适合于企业用户采购和借阅系统的总体应用架构。

3. 定义

- a. SSM: 由 spring mybits springmvc 一起组成的 3 个开源框架，用于构建灵活、易于扩展的多层 Web 应用程序。
- b. Mysql: 一个小型关系型数据管理系统，开发者为瑞典 Mysql AB 公司，属于开源软件。
- c. JSP: JSP(Java Server Pages)是由 Sun Microsystems 公司倡导、许多公司参与一起建立的一种动态网页技术标准。
- d. Javascript: 为客户提供更流畅的浏览效果。
- e. jquery:优化前台界面显示,提升用户视觉体验,提高前台代码简洁度
- f. h-ui admin 工作人员管理前台界面显示插件
- g. Eclipse: 开发工具。
- h. github:团队合作开发管理工具
- i. tomcat:web 项目运行容器

4. 参考资料

- 1. 《Spring+springmvc+mybits 开发》，电子工业出版社。
- 2. 《文档写作指南》，清华大学出版社。
- 3. 《企业用户系统需求设计说明书》。
- 4. 《企业用户系统概要设计说明书》。

二、架构表示方式

本软件架构设计文档以一系列的视图（View）来表示系统的软件构架，主要包括用例视图、逻辑视图、进程视图、部署视图、实施视图等，每个视图拥有一个或多个模型（Model）。并围绕相关视图来描述系统的基本结构、组成机制与工作原理等。本软件架构设计文档还将系统的构架机制描述也放在了逻辑视图之下。本文档主要使用统一建模语言（UML）来充当相关模型的表达语言；主要图表（Diagram）引用自企业用户系统的 Rose Model。

三、架构设计目标与约束

描述构架设计最主要目标就是满足关键系统功能需求和质量约束，这些功能需求和质量要求对软件构架有重大的影响，并决定了构架的设计。本节同时还列

明影响构架的其他相关因素，如软件的复用策略、使用商业构件、设计与实施的策略等。

1. 关键功能需求

按照需求分析文档的规格要求，本企业用户系统的设计分成了以无登录一般功能、企业用户功能模块以及管理员功能模块为主的三大模块进行开发，而在此三大模块的基础上，又细分成了企业用户信息管理、员工信息管理等子功能模块。系统的开发主要是为了使企业用户获取信息,发布信息\管理人员的管理工作更加规范化、系统化和程序化，提高信息处理的速度和准确性，提高企业用户以及管理人员用户体验。其功能模块示意图如下所示：

- 系统自动管理模块

- 企业用户注册

对企业用户的注册进行验证记录进数据库,判断是否重复以及输入或者命名不合法,对注册成功的自动生成具有唯一性的 id,初始状态为 0,需要通过员工审核后激活。

- 管理员/企业用户登录

检查登录用户名密码是否正确;登陆成功后使用 session 记录登录状态,访问新页面检查 session 中记录的登录信息决定请求是否放行.阻止他人通过 url 直接访问系统资源

- 登出

安全登出,清除服务器以及用户浏览器中对应用户登录信息

- 日志功能

生成日志 id,记录员工操作行为/操作对象/操作日期,改功能使用 spring 的切面工程实现.

■ 获取其他网站资源

系统定期运行爬虫模块爬取网络数据并输进数据库中.

■ 权限识别

根据权限分配前台员工以及企业用户显示特定操作按钮,限定其操作;提交操作时判断员工或企业用户是否具有该操作权限,未授权则中断操作

● 管理员操作模块

■ 企业用户信息管理

◆ 增加企业用户

可直接帮助企业用户进行帐号注册,注册同时系统抽调数据库判断是否存在该企业名字,防止重名;

◆ 删除企业用户信息,同时删除级联连表的信息

◆ 审核:工作人员审核界面显示所有未审核企业用户信息,若审核通过,则该用户激活,未通过,则系统推送消息给企业用户个人空间.

◆ 查寻:可根据企业用户名/帐号/行业等字段进行模糊搜寻,或根据账号进行精确搜索

◆ 改:更改企业用户基本信息

■ 技术服务需求/供应发布管理

◆ 进行更改,查询并审核,可将其中不符合的发布由激活状态变为未激活,从而其他用户无法搜索该信息

● 企业用户操作模块

■ 注册

填写企业详细信息以及相关执照/公司图标上传到服务器保存等待审核.

■ 登录

登录错误进行提示,登录成功跳转用户空间

■ 退出登录

用户安全退出登录,服务器缓存清楚用户信息

■ 修改信息:修改/完善详细信息

■ 查询其他企业信息

根据企业名字/行业类型/创建日期等字段进行搜索

■ 查寻行业数据

根据行业类别/名称列出附带行业数据标题的 url,用户点击进入查看详细行业数据

■ 查询行业资讯

根据行业类别/名称列出附带行业数据标题的 url,用户点击进入查看详细行业资讯.

■ 下载:以上查询结束之后可进行数据/资讯下载到本地保存

■ 技术服务供应/需求发布

在编辑页面编辑之后进行发布;默认为激活状态,发布之后所有人可见,发布后可进行修改完善;若工作人员筛查不过关,则状态变为冻结,其他人搜索不可见,需要修改通过审核重新恢复;需求/供应配对后,状态改编为以配对,他人搜索不可见.

● 游客模块

■ 搜索

只可选择行业类别显示企业/行业信息,只提供概览页面,无法查看详细信息,不提供下载功能,限制用户前台复制功能.

2. 关键质量需求

由于此开发项目针对一般事业单位企业用户以及员工管理,使用频度较高,使用性要求比较高。为防止对信息资料和管理程序的恶意破坏,要求有较为可靠的安全性能。总之,要求稳定、安全、便捷,易于管理和操作。

- 查询速度: 不超过 10 秒;
- 其它所有交互功能反应速度: 不超过 3 秒;
- 可靠性: 平均故障间隔时间不低于 200 小时。

3. 开发策略

● 软件复用策略

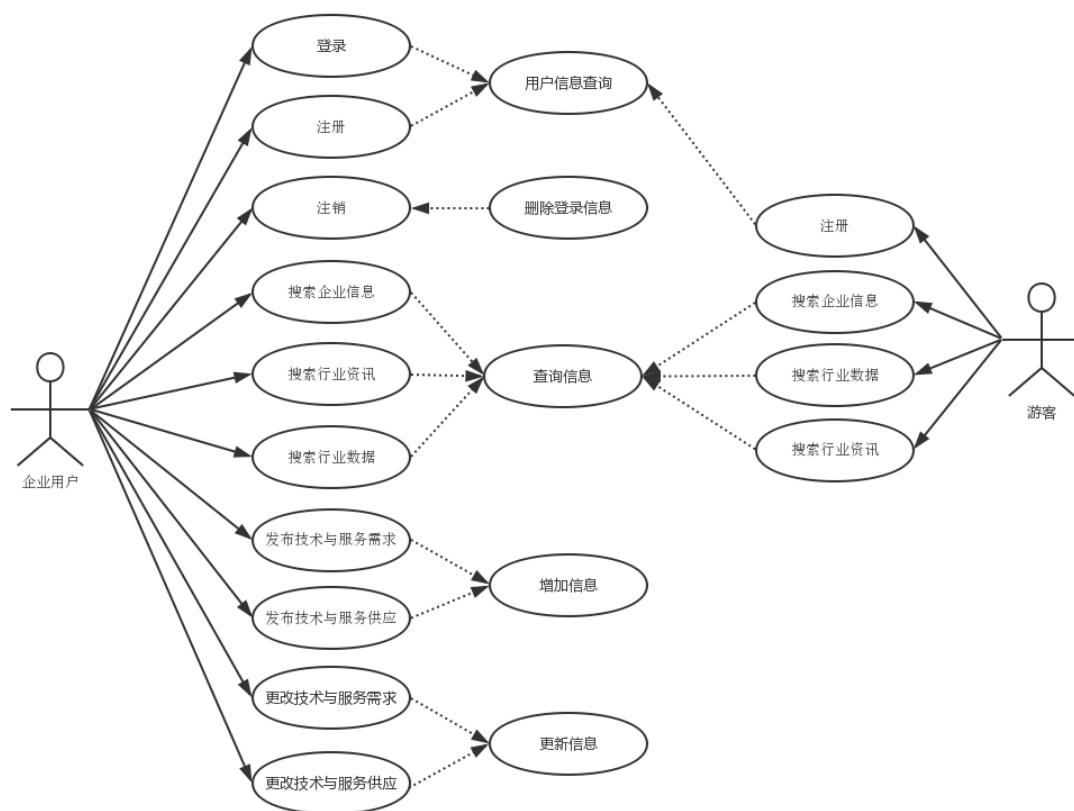
系统中重要基础构件应当具备较高的设计与构建质量,可以在产品中复用。

● 使用开源架构

本系统采用了由 spring、mybits 、springmvc 一起组成的 3 个开源框架,用于构建灵活、易于扩展的多层 Web 应用程序。Springmvc 是采用 Java Servlet/JavaServer Pages 技术,开发 Web 应用程序的开放源码的 framework。SpringMVC 实现了 MVC 设计模式的 WEB Framework。采用 SpringMVC 能开发出基于 MVC(Model-View-Controller)设计模式的应用构架。

四、用例视图

1. 用例视图



4-1-1 用例图

五、设计框架

1. 概述

本系统采用 SSM 框架，即 Spring、SpringMVC、MyBatis 三大框架的集合，属于标准的 MVC 模式，其将整个系统划分为四个层，即 view、controller、service 和 dao(mapper)四个层。

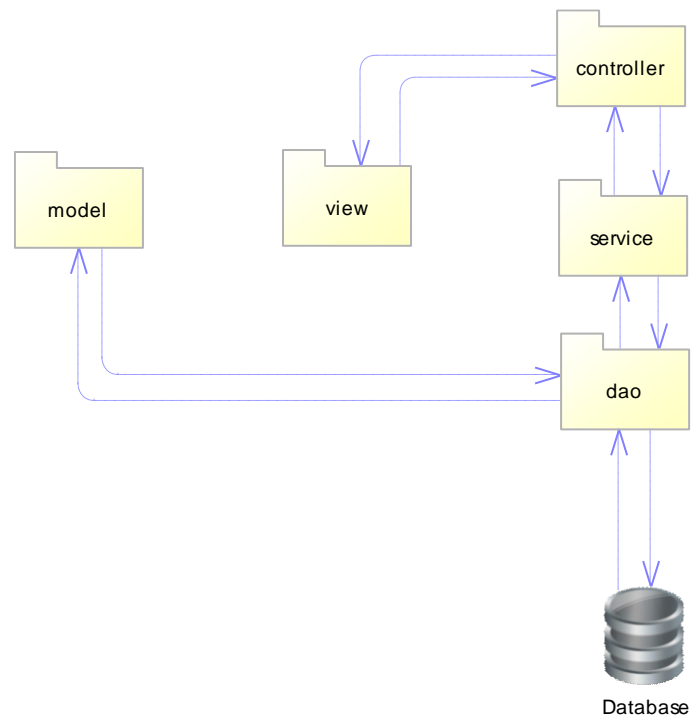
(1) SSM

- spring 实现业务对象管理
- spring MVC 负责请求的转发和视图管理
- mybatis 作为数据对象的持久化引擎

(2) 四个层级

- view，主要负责前台 jsp 页面的表示.
- controller，负责具体的业务模块流程的控制
- service，主要负责业务模块的逻辑应用设计
- dao(mapper)，主要是做数据持久层的工作，封装负责与数据库进行联络的一些任务

2. 逻辑视图



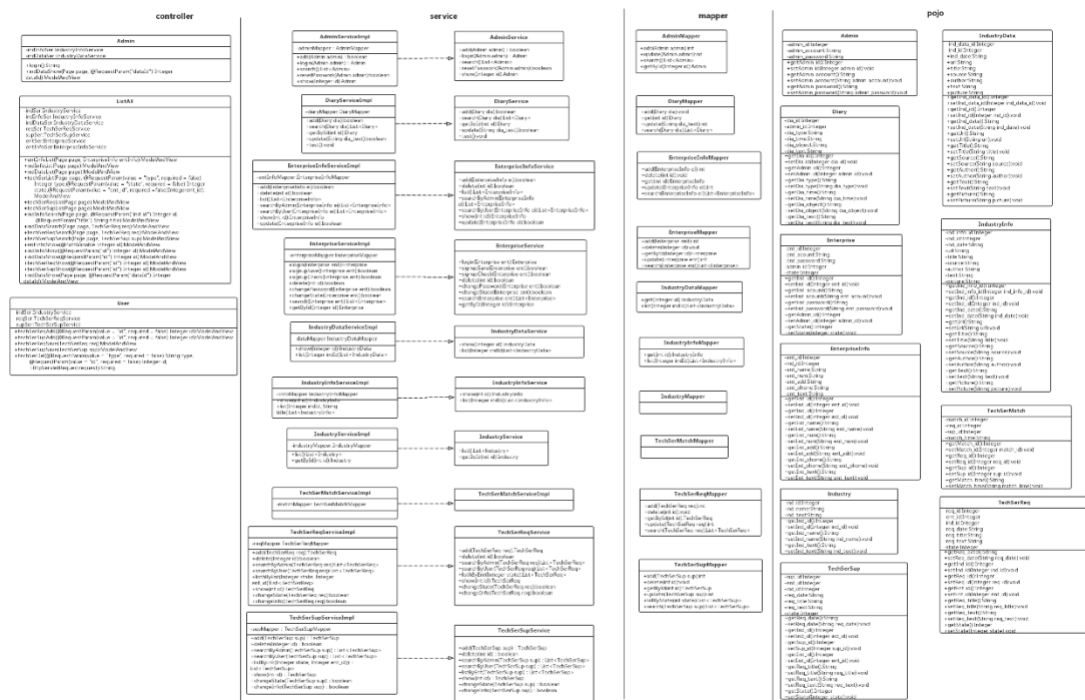
5-2-1 系统层次模型图

六、实施视图

1. 概述

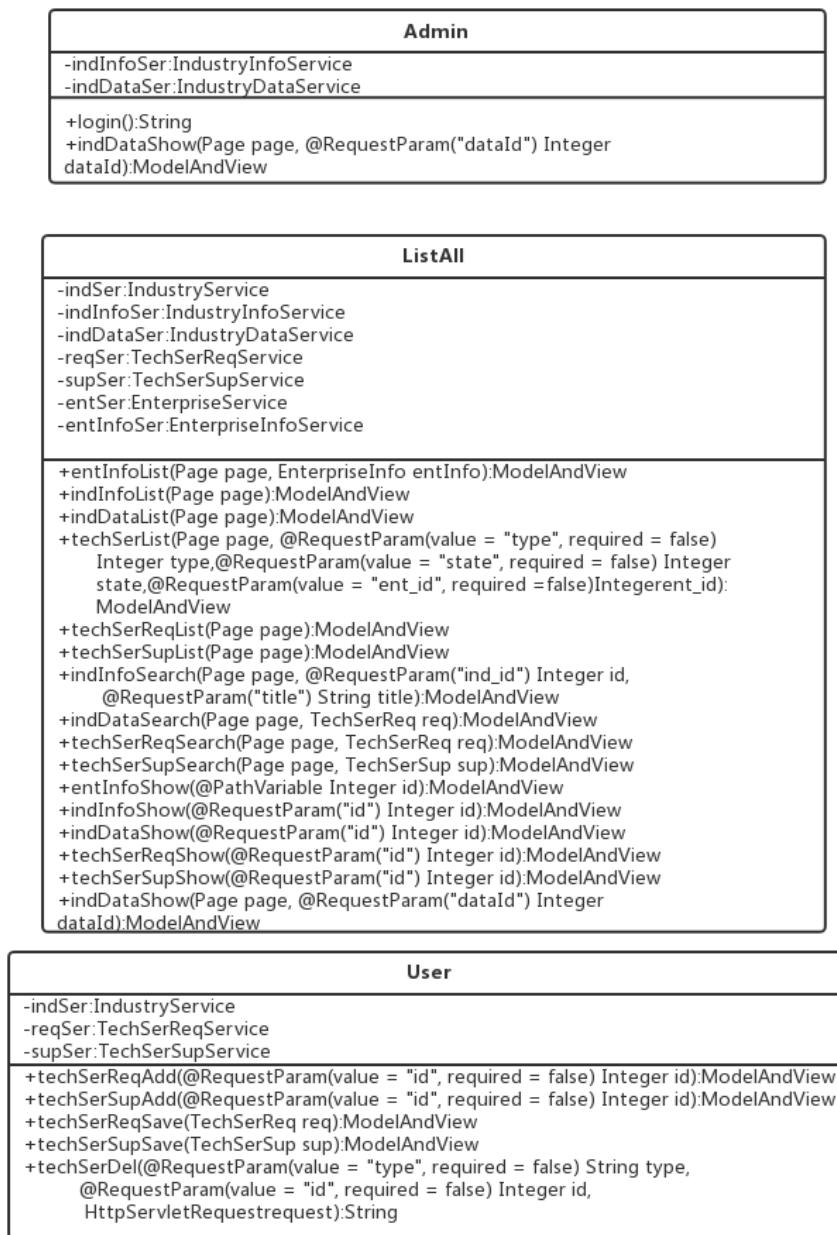
本部分从编译与构建的角度，描述系统实施构件的组织结构与依赖关系（主要是编译依赖）。模型包括实施子系统和构件结构，及其依赖关系。同时还表达了逻辑视图中各个包和类分配到实施视图中的子系统和构件的映射关系。

2. 实施模型视图(设计类图)

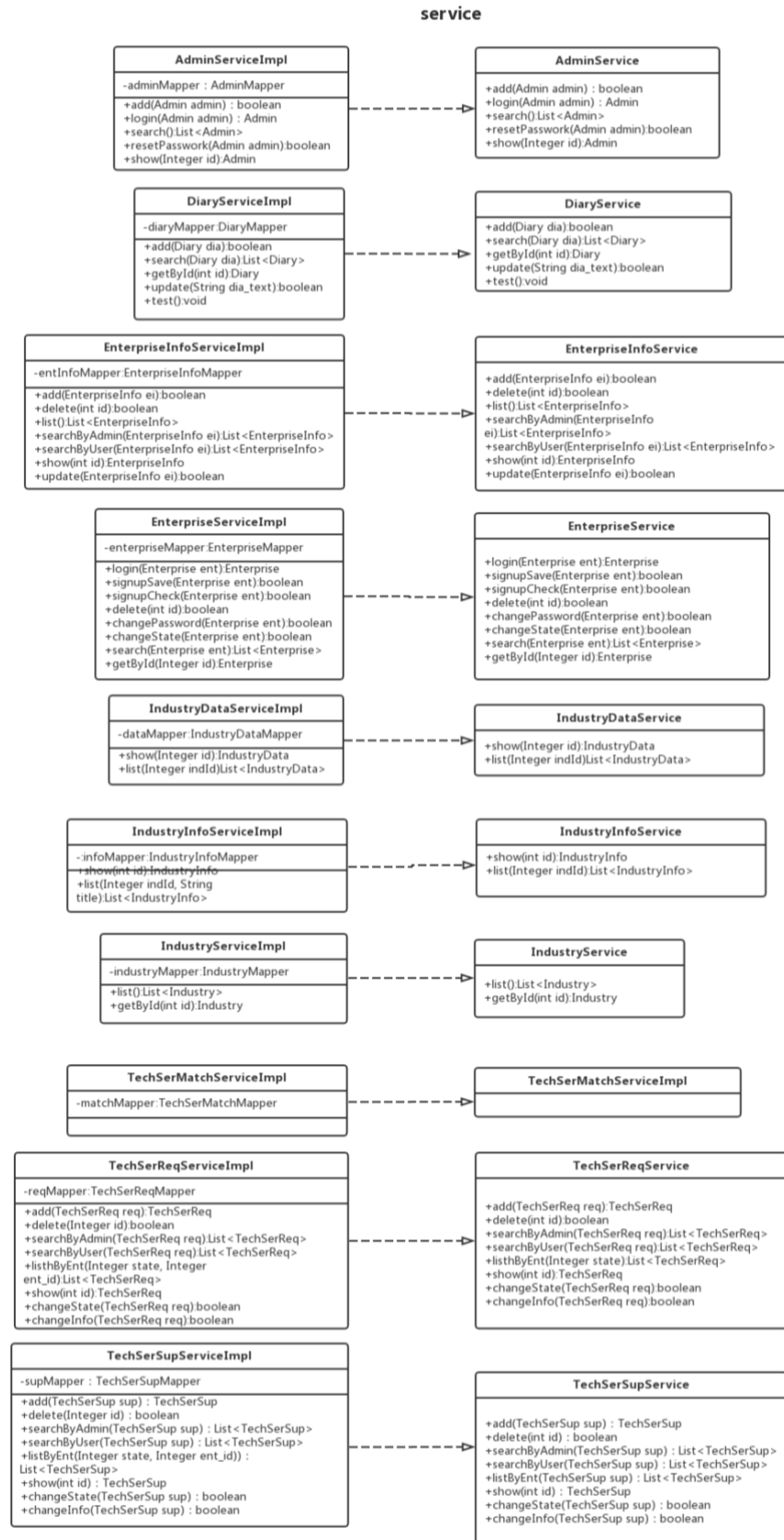


7-2-1 总类图

controller

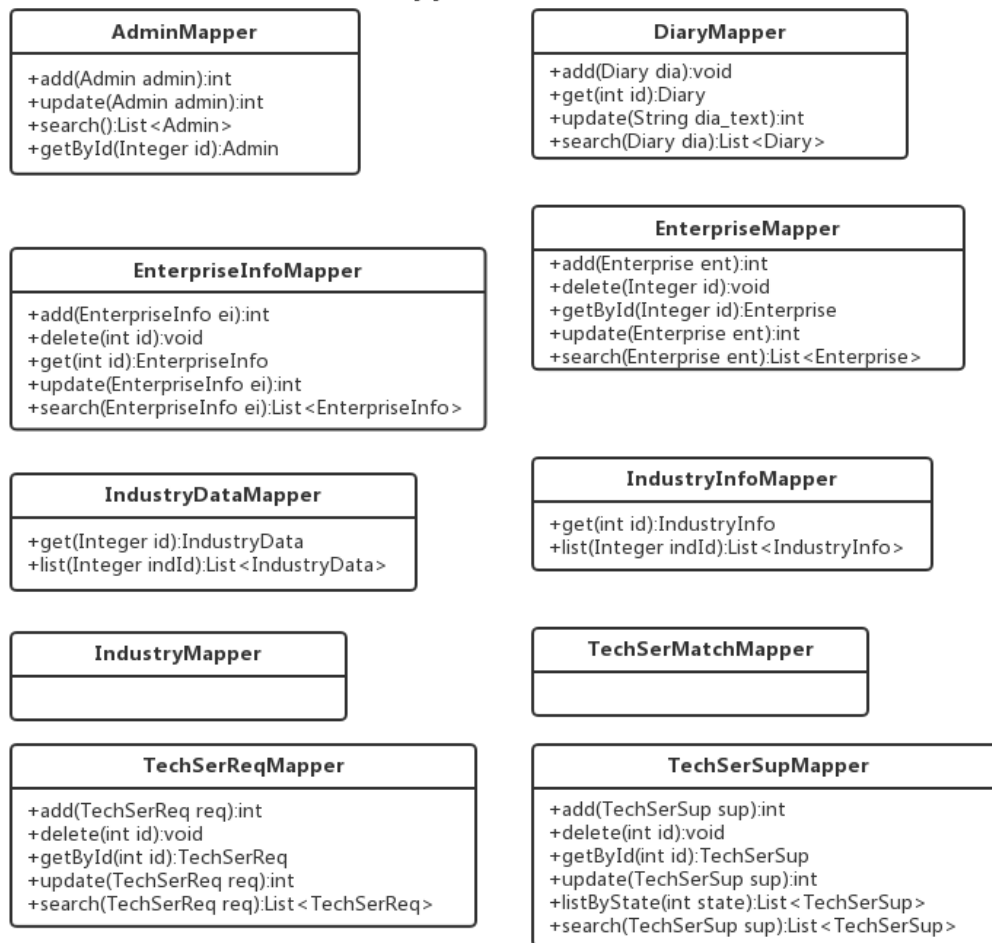


7-2-2 controller 层类图



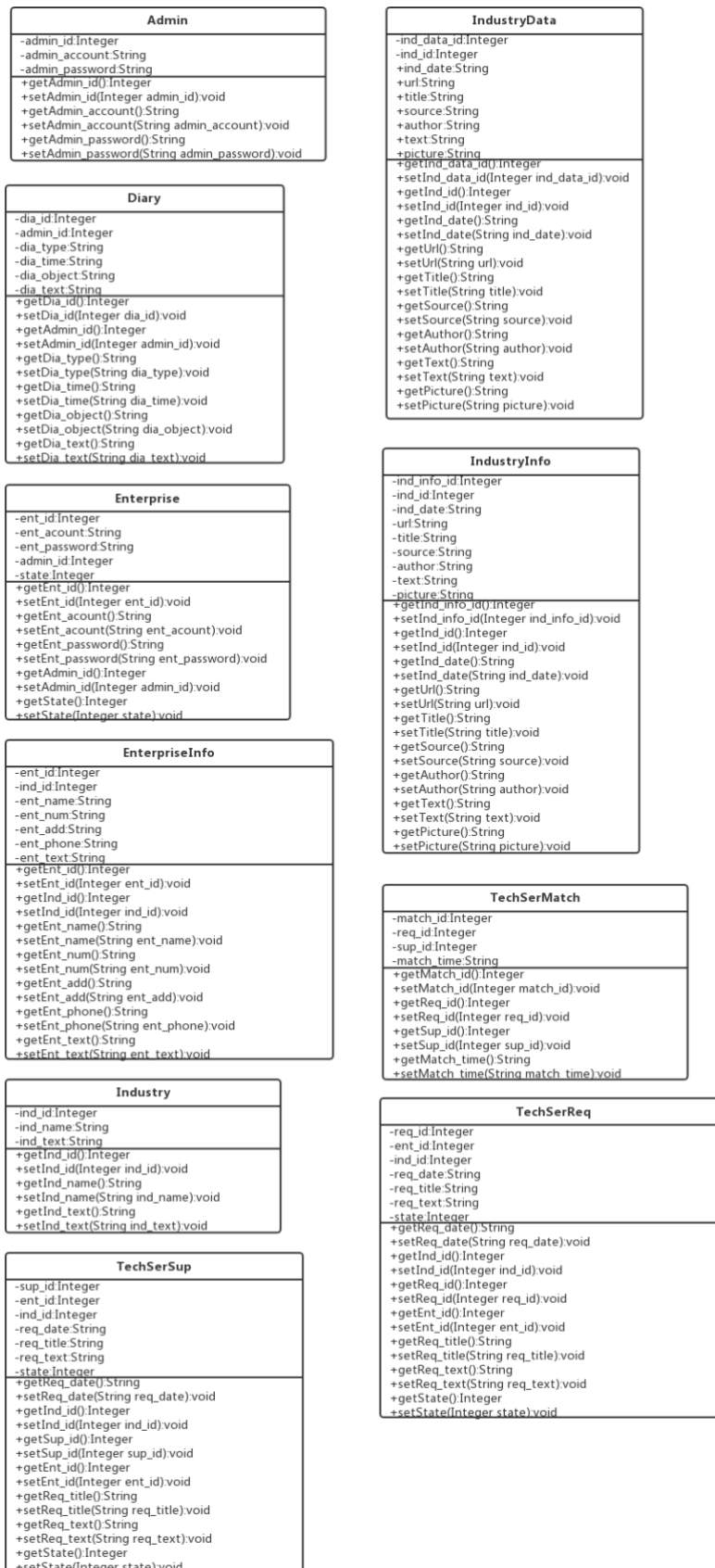
7-2-3 service 层类图

mapper



7-2-4 mapper 层类图

pojo



7-2-5 pojo 层类图

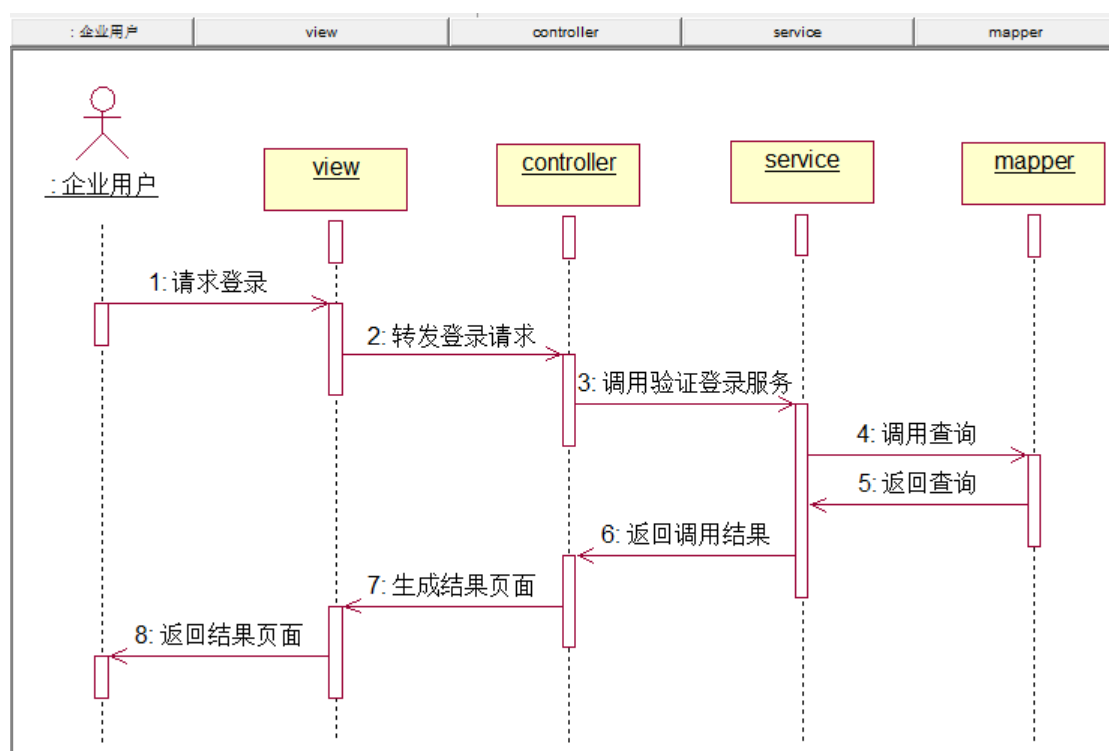
七、进程视图

1. 概述

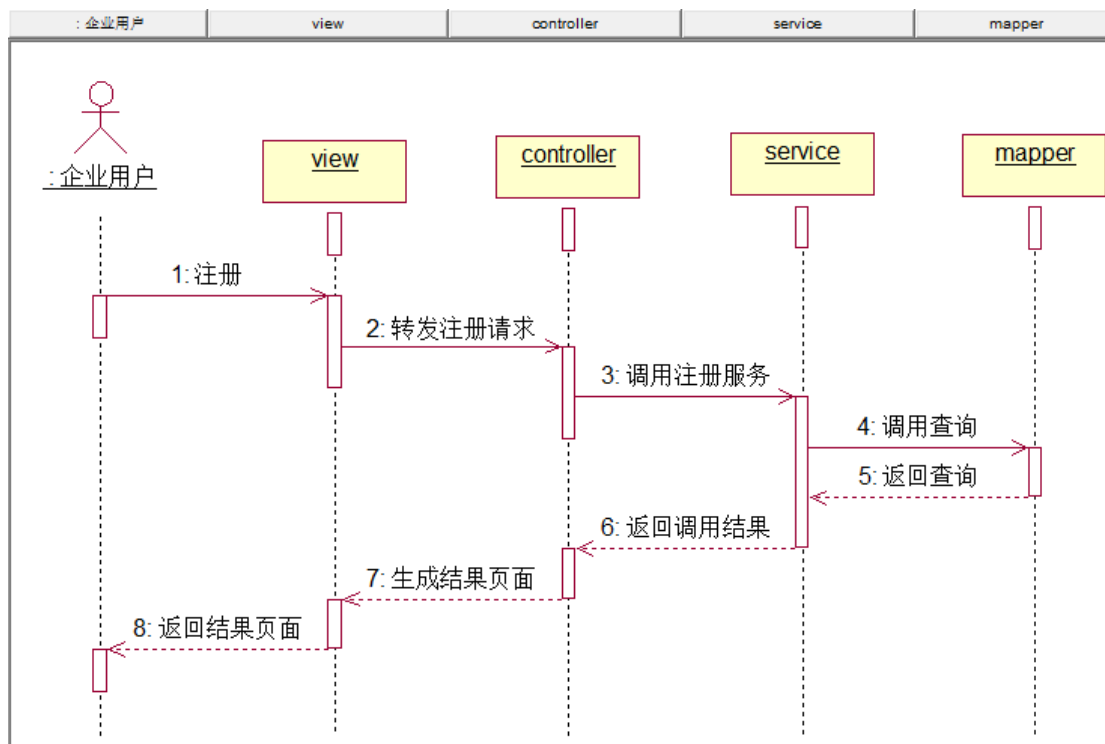
进程视图从系统运行时刻的角度，描述系统划分为进程、线程的结构，及其动态关系。模型主要说明不同系统角色之间的创建、交互和消息通讯关系等。

2. 角色进程视图

● 登录：

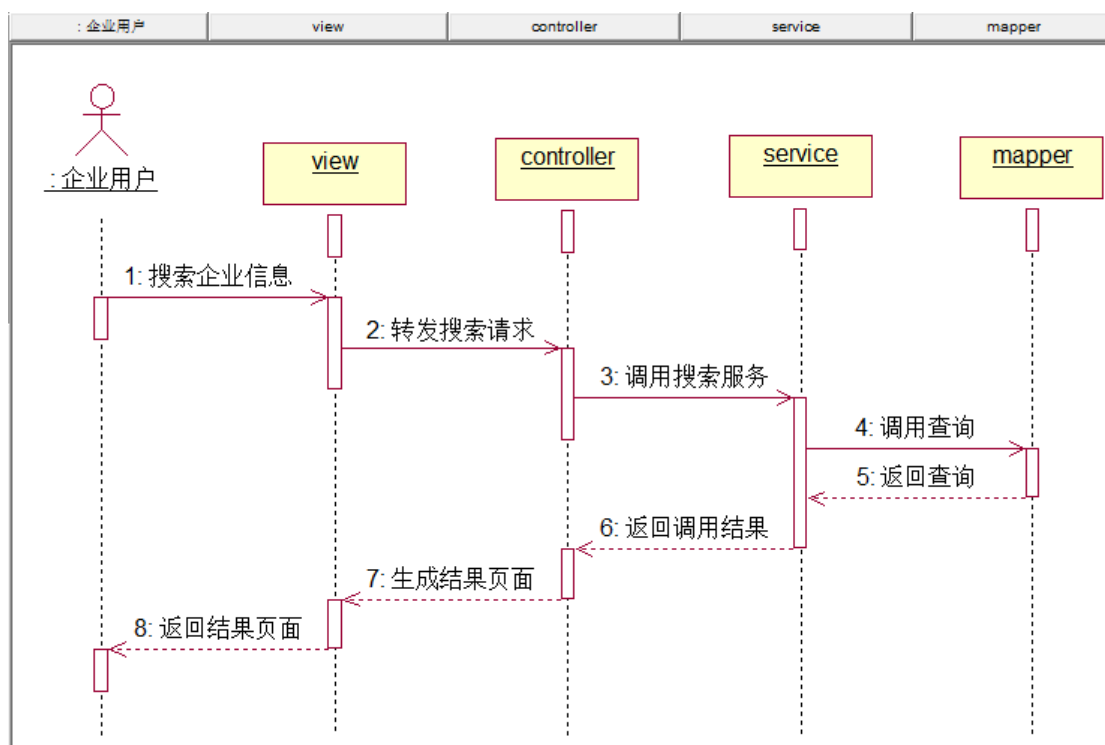


● 注册：



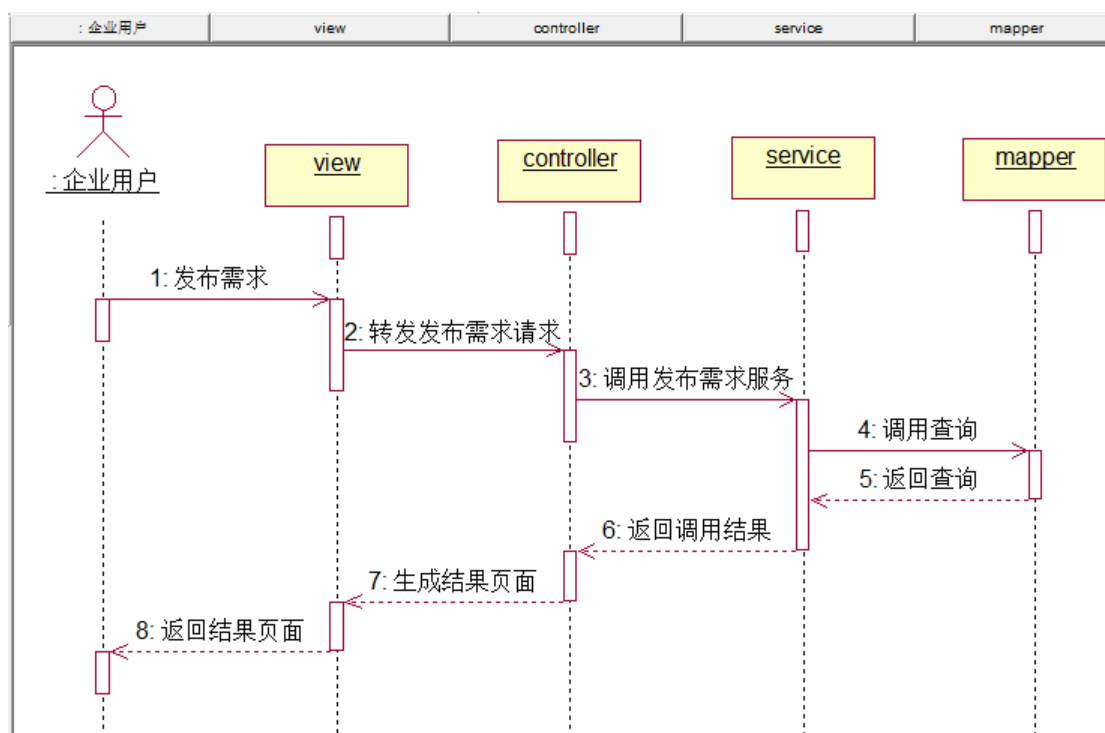
● 搜索企业信息：

S

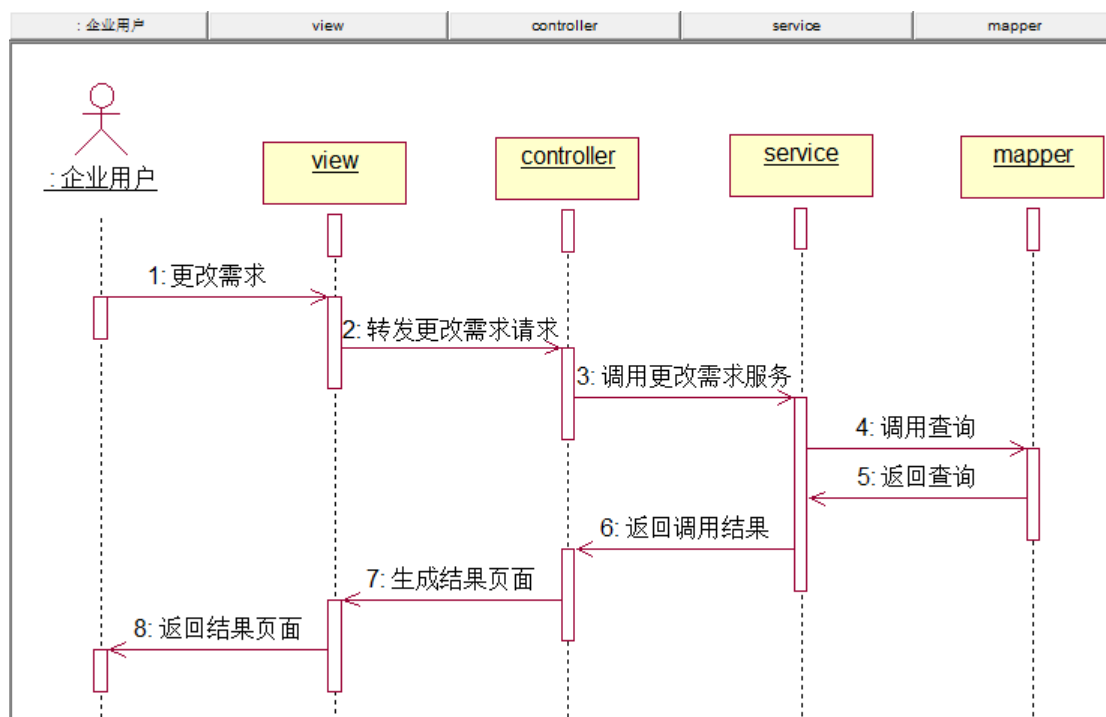


[键入公司名称]

- 发布技术与服务需求：



- 更改技术与服务需求：



八、部署视图

1. 概述

从系统软硬件物理配置的角度，描述系统的网络逻辑拓扑结构。模型包括各个物理节点的硬件与软件配置，网络的逻辑拓扑结构，节点间的交互和通讯关系等。同时还表达了进程视图中的各个进程具体分配到物理节点的映射关系。

2. 部署方案视图

