

山东工商学院

# 本科毕业论文(设计)



题 目：易生活客户服务系统

姓 名：姜山

学 院：计算机科学与技术学院

专 业：软件工程专业

班 级：2019 级 1 班

学 号：2019214251

指导教师：

职称：

2022 年 5 月 18 日

# 易生活客户服务系统

E-life Ticket Manager

姜山

Jiang Shan

2022 年 5 月 18 日

# 诚信声明

本人郑重声明：所呈交论文，是在导师指导下独立进行研究所取得的研究成果。论文除文中已经注明引用的内容外，不包含任何其他集体或个人已经发表或在网上发表的内容。

特此声明。

声明人：

年 月 日

## 指导教师意见

指导教师姓名：		职称：	
<b>指 导 参 考 项 目</b>			
论文选题	1	符合专业培养目标	
	2	具有学术价值	
	3	具有新颖性	
	4	难易程度适中	
文献资料	5	文献收集系统完备	
	6	文献理解准确得当	
	7	文献征引合乎规范	
	8	外文翻译准确通顺	
	9	观点归纳完整清晰	
论文写作	10	文题相符，立论正确	
	11	数据准确，逻辑严谨	
	12	条理清晰，论证充分	
	13	行文流畅，格式规范	
	14	篇幅适中，按期完成	
论文成果	15	文献综述完整，研究基础厚实	
	16	具有理论意义或实际价值	
	17	作者见解独到新颖	
	18	具有拓展、延伸性	
写作态度	19	积极同指导教师沟通	
指导成绩			
是否同意答辩（“□”内划“√”）		是□                      否□	
指导教师意见	指导教师签字： 年        月        日		

评阅人意见与成绩评定

评阅人姓名：		职称：	
评 价 项 目			
论 文 选 题  (10 分)	1	符合专业培养目标	
	2	具有学术价值	
	3	具有新颖性	
	4	难易程度适中	
文 献 资 料  (25 分)	5	文献收集系统完备	
	6	文献理解准确得当	
	7	文献征引合乎规范	
	8	外文翻译准确通顺	
	9	观点归纳完整清晰	
论 文 写 作  (40 分)	10	文题相符，立论正确	
	11	数据准确，逻辑严谨	
	12	条理清晰，论证充分	
	13	行文流畅，格式规范	
	14	篇幅适中，按期完成	
论 文 成 果  (25 分)	15	文献综述完整，研究基础厚实	
	16	具有理论意义或实际价值	
	17	作者见解独到新颖	
	18	具有拓展、延伸性	
评阅人最终给分			
评 阅 人 评 语	<div>评阅人签字： 年    月    日</div>		

答辩（评审）委员会意见

成绩 \_\_\_\_\_

鉴定意见：

主任（签章） \_\_\_\_\_

年      月      日

## 易生活客户服务系统

### 摘 要

在生产实践中，客户服务大多采用一对一的模式：在这种模式下，虽然用户的体验会有一定程度的上升，但是受限于指定客服人员的时间窗口，容易在一定程度上降低用户的体验。目前，许多企业的解决方案仍是扩大客户服务团队的规模来提高质量。但是对于小微企业以及只有数人的小型工作室而言是及其困难的。

鉴于此，我们提出了一种基于服务单客户服务模式，并使用 **Spring, Vue** 等技术制作了配套程序。经过一年的实践，这种模式可以支持一个不足百人的客户服务小组满足一万余名客户的服务需求，取得了巨大的成功。事实证明，该模式不仅可以提高小微企业的资源利用效率，还能够支持大型企业以及外包性质的客户服务。

**关键词：** 易生活客户服务系统；服务单；客户服务；服务模式；小微企业

## **E-life Ticket Manager**

### **Abstract**

In practice, customer service mostly adopts a one-to-one model which focus on user experience. However, due to staff's schedule, the time to provide services is greatly limited. For now, the solution for many corporations is still increasing the size of the customer service team to make the quality. It is extremely difficult for small, micro and even tiny commercial business with only a few people.

Because of the truth that tiny enterprise can hardly provide high quality customer service with limited human resource, we introduced a Ticket-driven customer service method to solve this problem and coded a program using Spring, Vue and other productions. During a practice lasts a year using a simple Python program, this method can support a customer service team of less than 100 people to meet the service needs of more than 10,000 customers. Facts have proved that this method can not only improve the resource utilization efficiency of small and micro enterprises, but also support large enterprises and outsourced customer service

**Keywords:** customer service; Ticket; micro company; Vue



## 目 录

引言 .....	1
第一章 绪论 .....	2
1.1 项目背景和意义 .....	2
1.1.1 项目背景 .....	2
1.1.2 项目意义 .....	2
1.2 服务单模式 .....	2
1.2.1 服务单 .....	3
1.2.2 服务流程 .....	4
第二章 需求分析 .....	5
2.1 总体描述 .....	5
2.1.1 产品概述 .....	5
2.1.2 系统目标 .....	5
2.1.3 定义及缩写词 .....	6
2.1.4 产品功能 .....	7
2.1.5 用户特征 .....	8
2.2 具体需求 .....	8
2.2.1 模板 .....	8
2.2.2 企业入驻申请 .....	9
2.2.3 .....	10
2.3 功能需求 .....	10
2.3.1 用例列表 .....	13
2.3.2 用例表 .....	15
2.4 数据库需求 .....	16
2.5 系统的质量属性 .....	17
2.5.1 可靠性 .....	17
2.5.2 可获得性 .....	17
2.5.3 保密性 .....	17
2.5.4 可维护性 .....	17
2.6 需求模型 .....	17
2.6.1 静态模型 .....	18
2.6.2 动态模型 .....	19
第三章 体系结构设计 .....	20
3.1 部署视图 .....	20
3.1.1 概述 .....	20
3.1.2 部署政策 .....	21
3.2 逻辑视图 .....	21
3.2.1 概述 .....	21
3.2.2 包或子系统 .....	22
3.3 进程视图 .....	24
3.3.1 概述 .....	25
3.3.2 过程间通信机制 .....	25
3.4 Use-Case 视图 .....	25

3.5 实现视图 .....	26
3.5.1 业务流程层 .....	27
3.5.2 业务逻辑层 .....	29
3.5.3 数据访问层 .....	32
3.6 数据视图 .....	33
3.6.1 逻辑设计 .....	33
3.6.2 数据访问机制 .....	35
第四章 详细设计 .....	36
4.1 用户交互层 .....	36
4.1.1 企业入驻审批 .....	36
第五章 测试报告 .....	39
<b>参考文献</b> .....	40
致谢 .....	41

## 引言

消费结构，不仅是消费领域的中心问题，也是社会再生产运行中一个极重要的问题。从社会再生产过程来看，衡量一个国家经济发展的好坏，衡量国民经济是良性循环还是恶性循环，关键在于是否合理解决了消费问题。研究消费结构的发展趋势和规律性，研究产业结构如何适应需求结构和消费结构的变化，对于促进国民经济协调、快速发展，具有重要的作用。因而研究消费结构的变化，不仅是经济理论的一个重要内容，而且也是政府调节经济、制定经济政策的重要依据。因此，加强对消费支出的研究有着十分重要的理论意义和极大的现实意义<sup>①</sup>

<sup>①</sup> 宏观经济学认为，GDP 等于消费加投资加政府购买及进出口，因此消费成为促进经济增长的重要一环。

# 第一章 绪论

## 1.1 项目背景和意义

### 1.1.1 项目背景

在生产实践中，许多企业的客户服务大多采用一对一的方式：一个客户服务人员对接多个客户，而且在当前客服无法回答时需要转接其他部门继续沟通交流。在这种模式下，不仅客户服务的时间窗口被大量压缩，转发客户服务申请时也要考虑对应部门当前是否有人在线。如此，一旦在短时间内大量人员提出请求，就可能造成单个客户服务人员工作量剧增，在一定程度上降低客户服务的质量。

目前，许多企业的解决方案仍是扩大客户服务团队的规模来提高质量；但是对于许多中小型企业，尤其是体量更小（通常仅由几人组成）的小型工作室以致个体工商户而言，由于没有专门的客户服务系统，他们通常采用传统的聊天软件（如 QQ、微信）来进行客户服务。在这种情况下，经常出现由于用户急于获取服务，向每个可以与用户对接的客户服务人员都提出服务请求的情况。这种行为产生了大量的资源浪费并大大降低、甚至毁灭了这些小微企业的客户服务质量，并且常常由于员工难以及时回答使得客户服务部门的员工往往需要承受更多来自用户的谩骂和侮辱；或在耐心解释后得到用户“已经解决”的回复，使得员工在工作中难以获得成就感，在极大程度上消磨了员工的精神、降低了员工工作的积极性和自我认同感，在恶性循环中一步步降低了服务质量。因此，一种能够替代传统聊天软件的客户服务系统逐渐成为了一种迫切的需求。

### 1.1.2 项目意义

在上文提到的研究背景下，开发一种能够解决中小型企业 and 小型工作室的客户服务问题的系统，不仅能够整合大量的小微企业、小型工作室甚至个体工商户的需求，降低整个系统的运营成本；并且能够在一定程度上解决当前小微企业的困境，解放生产力，优化资源安排、促进产业发展。如果能够按照需求整合电商平台、直播平台或其他功能，系统将能够进一步统合资源，提高工作效率。

## 1.2 服务单模式

易生活客户服务系统统筹各项资源的核心是一种经过重新架构设计的，颠覆现有客户服务模式的全新模式，我们将其称为基于服务单的客户服务模式，简称服务单模式。这种模式能够极大程度上优化客户服务人员的服务质量，提高用户体验，并且减少企业人力资源的消耗。

经过一年的实践，截至 2023 年 5 月 1 日，通过使用这种模式，一个不到 100 人的团

队在一个总人数 15827 人的社区中成功提供了总计 20064 次客户服务，基本能够达到及时响应、高效沟通的要求，得到了业务团队和社区成员的极佳评价。

在详细描述这种新的客户服务模式之前，需要先引入贯穿整个模式的概念——服务单。

### 1.2.1 服务单

服务单是易生活客户服务系统的核心，也是我们提出的一种全新的用于衡量客户服务工作的单位。在国际化界面中，我们将其称为 Ticket。由于这种全新提出的概念对用户不够友好，在实际的操作界面中往往将其封装成易于用户理解的友好界面。

服务单是由企业产品的用户提出的一个或多个服务申请组成的、由企业客服人员与其进行的一次或数次会话；服务单也是企业产品用户阐明自己的需求，企业客服人员对其进行服务的依据。在本模式中，服务单将作为处理客户请求的单位。

服务单可以是独立的，也可以是一组服务申请的一部分。可以为一张服务单定义子单：如果客户提出的申请过长、过多或过于繁琐，难以在一次会话中系统、简明地阐述，就需要定义一组服务单。一般来说，如果客户提出的需求可以在一张服务单中阐明，并且双方能够达成一致，则不应为用户发放子服务单。由多个服务申请组成的服务单仅在用户的请求过于繁琐、冗长且难以用篇幅较短的文字说明时才应出现。

在线上的服务单管理系统中，由类似聊天群组的形式对服务单进行管理。定义一个只能由指定客服人员和客户可见的聊天频道，我们可以将其视作一张线上的服务单。对于这种服务单来说，可以定义自一个频道创建开始到频道被关闭为止，作为一张服务单的生命周期。

一般来说，服务单具有以下几种状态：

#### 1) 申请并提交

用户为了从企业处获得期望的服务，向企业申请一张服务单。这张服务单可以是单独的，也可以是一张服务单的子服务单。

一般来说，企业应在接到申请后向用户发放服务单。在本系统中，企业可以通过设置单个用户最大服务单申请上限，来避免某些用户恶意申请多张服务单、进行信息轰炸等具有潜在危害的行为。这个上限一般由企业自定义，若企业未明确定义或尚未定义，则数量会被默认为两张。

用户在申请服务单后应当填写相关条目，包括期望获得的服务，现在的状况等；并在填写完成后提交到企业，以使企业进行有针对性地安排专业人员进行服务。

#### 2) 关闭

当用户提交服务单时意愿进行服务的问题已经被解决，或企业与用户双方达成一致后，用户不再需求企业的服务。此时可以关闭服务单，减少企业客户服务人员在工作时的

检索压力，释放一定的资源，并将服务单进行归档整理。

关闭服务单的操作可以由用户或企业员工执行，在关闭服务单后用户可以对本次服务进行评价。服务单关闭后，用户将不再能够检索到服务单；但是企业员工可以在对应的界面中检索已经关闭的服务单。

如果企业发现问题尚未被解决或需要后续通知用户，服务单可以被再次开启。再次开启的服务单将视同已经提交的服务单，用户能够正常检索并在服务单中发送消息，与客户服务人员进行沟通，并就重新发现的问题与企业达成一致。

### 3) 删除

如果已经归档的服务单数量过多，难以检索，企业员工可以选择将服务单删除来释放一些资源。一般来说，已经删除的服务单不能够再被检索到；除非用户或企业员工有备份服务单，他们不能够再获取已经删除的服务单的信息。

## 1.2.2 服务流程

本段将用故事的方式讲述在本模式中一次典型的客户服务流程：

我们假设用户“小光”在“海德林公司”购买了一台仍在保修期内的笔记本电脑。在正常使用过程中，电脑的硬盘出现了故障，不能继续读取数据。按照保修服务的规则，他可以联系公司的客户服务人员将故障件寄回，并换取一件良好的硬盘。

此时，小光需要登录易生活客户服务系统，在“海德林公司”的页面中点击“申请服务”的按钮。此时，系统会自动检查小光是否符合申请条件，并创建一个仅小光和公司客户服务人员能够看到的聊天室。在双方的系统中，这个聊天室被命名为“笔记本电脑硬盘故障”。

由于这个聊天室能够被任何属于海德林公司的客户服务人员检索到，很快一名经过良好培训的客户服务人员“μ”就看到了这个聊天室。μ查看了聊天室的标题，发现这是一个硬件方面的问题，不属于他的工作范围，于是他联系了属于公司硬件售后服务的部门，并且在聊天室中回复了小光，告知他问题已经收到；并且在等待硬件售后服务部门的员工回复时，μ和小光进行了充分友善的交流，使问题被详细地描述并暴露给各个员工。

由于μ联络了硬件服务部门，他们能够在系统中很快发现这个问题。部门下属的员工“小洛”在完成了对上一个用户的服务后立刻注意到了这个聊天室中有需要解决的问题。得益于μ已经和小光聊了不少，他仅仅浏览了一些聊天记录，就能够判断出小光的需求，并且根据公司规定的服务流程，将公司提前准备好的申请表格发送给了小光。在填写完所有的材料之后，线上的服务部分就结束了。接下来，只需要等待小光将快递单号发送给客户服务人员，并完成线下的资源交换。

## 第二章 需求分析

### 2.1 总体描述

#### 2.1.1 产品概述

在本产品中，系统由用户端和服务端两个端构成，根据不同的用户角色构建不同的页面，使用权限控制来控制不同用户的不同访问范围。

##### 1) 用户端

用户端主要负责显示界面的构建与渲染，接收服务端转发的数据，经过一定处理逻辑后将其显示出来。此外，对不同用户的不同角色会提供以下服务：

用户可以通过本产品对合作企业提出售后服务申请，并与企业的客服人员进行沟通交流、对不满意的企业进行投诉；并且能够查询并保存已有的服务记录。在经过用户与企业的双向认证之后，用户可以成为某个企业的员工。此时，用户的身份会转变成员工，拥有对应公司的服务单查看权限。

使用本软件的用户可以对运营公司提出申请，认证成为企业官方账号。经过认证的企业账号将拥有自己的企业频道，拥有企业频道中的所有权限。一般来说，一个企业频道将由多个分类组成，每个分类由多个聊天频道组成，并且拥有独立的权限设置。使用本软件的企业在申请入驻平台并通过审核后，可以认证某个用户，使其成为属于本公司的员工；能够更改员工所属部门、管理企业内部的员工数据和信息可见度；能够安排客服人员与客户沟通、对用户的请求进行回复、分类归档，并且能够查询以前的服务记录和员工或用户在服务过程中对系统执行的各种操作。此外，企业还能够收到用户对企业的投诉，并可以对各个客服部门的情况进行可视化分析。

开发人员和运维人员可以在登录后查看系统的运行日志、错误信息以及其他用户的操作记录。

##### 2) 服务端

服务端主要进行消息的转发，在用户在登录后，根据用户的权限，所属部门以及使用的功能，为用户提供对应的数据。

#### 2.1.2 系统目标

能够存储一定数量的服务单信息，并能够有效地对服务单数据进行操作和管理。

能够存储一定量使用者的信息，并能够有效地对使用者的信息进行操作和管理。

具有一定的安全性，为常见的攻击方式提供可靠的防护。

能够便利企业对客服人员的管理，用户与企业售后服务人员的沟通，并且能够一定程度上降低客户服务工作的成本。

### 2.1.3 定义及缩写词

表 2 - 1 缩写词对照表

缩写	含义
HTML	Hyper Text Mark-up Language
HTTP	Hyper Text Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
JDBC	Java Database Connectivity
SQL	Structured Query Language
UC	Use Case
PC	Personal Computer

#### 1) 服务单

服务单是由客户提出的一个或多个服务申请组成的、客服人员 and 用户进行的一次会话；也是客户阐明自己的需求，客服人员对其进行服务的依据。

#### 2) 用户

狭义上，用户在本文中指代需要企业提供售后服务，申请服务单、要求企业为其提供服务的用户。用户可以向服务提供商(通常是企业)申请服务。

广义上，所有注册了本产品账号的自然人或使用本产品接口的计算机软件都可以称为用户，用户根据实际情境的不同可以作为员工或企业用户存在，在一定条件下它们之间可以互相转换。

#### 3) 员工

如无特殊说明，在本文中指代使用本产品的企业人员，包括所有属于企业的人员(包括决策人员、管理人员和普通员工)。企业人员的认证通常是由企业发出的，经过用户确认后用户可以被认证为企业的员工，拥有企业内数据的可见权限。

员工主要负责解答用户的问题，并满足他们的合理需求。一个企业中可能有许多员工。这些员工可以有不同的部门、不同的权限，并且可以分别进行管理。

#### 4) 企业

如无特殊说明，在本文中指代通过认证入驻本产品、拥有企业频道、能够使用所有的企业功能的用户。

#### 5) 运营人员

如无特殊说明，在本文中指代全部或一部分实际部署使用本产品并进行盈利或非盈利



活动的自然人或法人。这类人员通常拥有一台或多台服务器，或持有使用它们的权限；能够部署本产品并稳定运行，为其他用户提供服务。一般来说，运营人员还需要负责对本产品所使用的软件进行维护及升级。

#### **6) 企业频道**

经过认证的企业会创建一个包含该企业所有信息（包括但不限于服务单、员工等）的企业频道，在一个企业频道下能够包含多个分类频道。

如果企业的管理用户（或拥有权限的企业员工）需要对企业进行管理，也需要在企业频道中进行对应操作。

#### **7) 分类频道**

分类频道主要用来对服务单进行分类管理，并且通过关联部门的形式进行权限的控制。一般来说，员工仅能够查看当前部门及下属部门的数据。

#### **8) 频道**

如无特殊说明，在本文中指代用户端中用于企业员工和用户进行沟通的文字聊天频道，一般以聊天室的形式存在。

#### **9) 分组**

如无特殊说明，在本文档中指代服务单的分组。

在单个服务单难以满足用户的需要时，往往需要提出一组服务单来描述用户的需求。此时，用户提出的一组服务单称为一个分组。如果制定了单个用户服务单申请数量上限，则一组服务单共同占用一个名额。

#### **10) 用户端**

用户端是一个使用 Vue 框架开发的网页应用，它为所有使用本产品的用户构建易于使用、用户友好的图形化界面。用户端负责接受服务器传来的数据，并将其进行渲染。用户可以在用户端方便地进行各种操作。

根据不同的用户角色，用户端会渲染不同的界面：普通用户会看到其已经申请的服务单和不同的企业；企业用户能够看到自己企业的信息并能够对其进行对应的设置；运营人员能够看到系统的后台数据日志。

#### **11) 服务端**

服务端为运行在独立服务器上的管理软件，负责数据的转发与筛选，和数据库通信以及保障通信安全。

### **2.1.4 产品功能**

本产品涉及的功能如下：

用户端注册、用户端登录、重置密码、服务单转发、搜索客服评价、查看评价详情、查看综合评价、添加问题类型、删除问题类型、搜索问题类型、修改问题类型、搜索投诉、

删除投诉、查看投诉、搜索入驻申请、同意入驻申请、驳回入驻申请、搜索客服、添加客服、注销客服、重置用户密码、切换客服状态、搜索企业、添加企业、注销企业、切换企业状态、申请服务单、关闭服务单、重开服务单、删除服务单、发送消息、服务评价、投诉服务、企业入驻申请、查看日志、添加员工角色、删除员工角色、修改员工角色、搜索员工角色、赋予员工角色、撤销员工角色、自定义回复模板、搜索回复模板、修改回复模板、删除回复模板。

将上述所有功能全部列出过于冗杂，本文将会挑选其中数个进行描述。

### 2.1.5 用户特征

表 2-2 用户群体

角色	职业	年龄	特征和需求
用户	不限	不限	有问题需要企业提供的客户服务
员工	企业的客服人员	不限	有工作经验
运营人员	企业的管理人员、法人或董事会成员等具有决策能力的人员、以及需要涉及到相关功能的企业员工，也可以是组织或个人。	不限	部署并使用本软件，通过其功能进行盈利或非盈利性质的活动；需要管理客服人员并及时得到反馈
管理员	软件的开发者和企业的技术人员，也可以是组织或个人	不限	有工作、编码经验，需要对软件进行维护

## 2.2 具体需求

### 2.2.1 模板

- 1) 窗口标题
- 2) 窗口标识符
- 3) 目的
- 4) 界面布局
- 5) 数据项描述
- 6) 数据项之间的关系
- 7) 窗口有效性
- 8) 与其它输入/输出的关系

- 9) 窗口布局
- 10) 反馈信息
- 11) 命令的执行方式

2.2.2 企业入驻申请

- 1) 窗口标题  
入驻申请
- 2) 窗口标识符  
EnterpriseImparting
- 3) 目的

运营人员需要在进驻前确认企业的信息，防止违反相关法律法规的企业使用本产品。故尚未进驻的企业为了使用本产品，需要进行经营资质、经营范围等的审查；在通过审查后，才能够使用本产品。

- 4) 界面布局

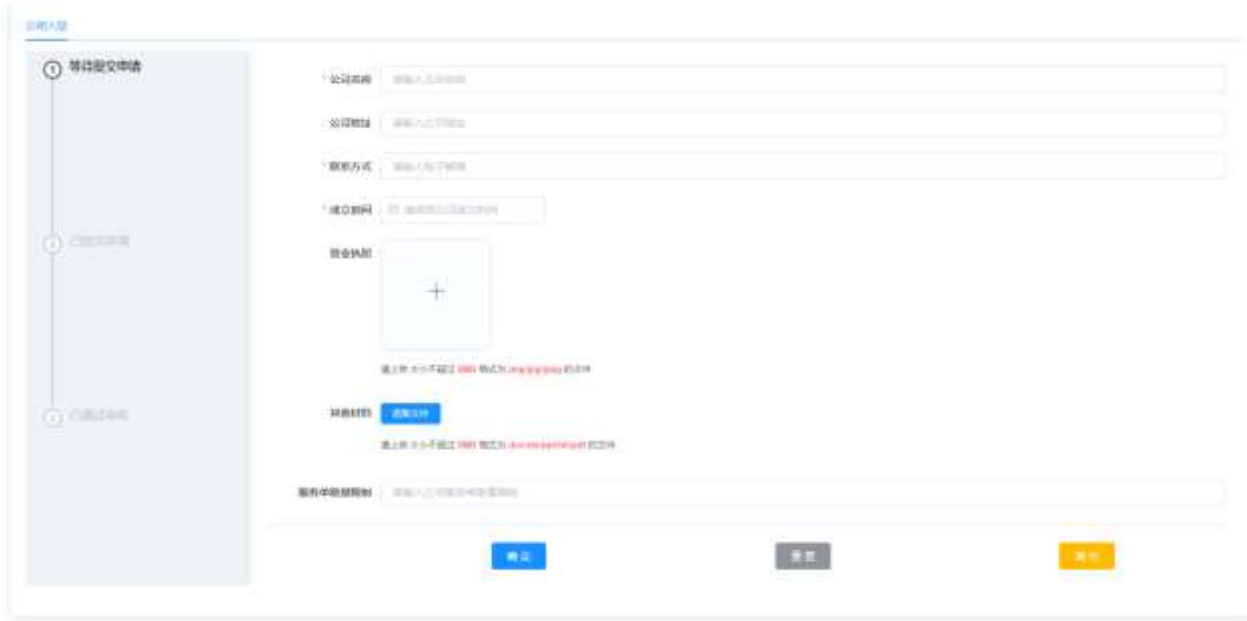


图 2 - 1 公司入驻申请界面

5) 数据项描述

表 2 - 3 企业入驻申请数据项

名称	类型	精度	有效范围	格式	长度限制	度量单位	缺省	是否可空	数据来源	备注
企业Id	字符串	NA	NA	^[0-9a-zA-Z]*\$	20 字	NA	无	否	系统输出	

企业名称	字符串	NA	NA	^[0-9a-zA-Z]*\$	20 字	NA	无	否	用户输入	
地址	字符串	NA	NA	^[u4e00-\u9fa5]*\$	40 字	NA	无	否	用户输入	
成立时间	日期	年	历史日期	^d*\$	NA	NA	无	否	用户输入	软件校验
联系方式	字符串	NA	NA	\w[-\w.+] * @ ([A-Za-z0-9] [-A-Za-z0-9] + \. ) + [A-Za-z] +	20 字	NA	无	否	用户输入	邮箱
营业执照	字符串	NA	NA	^[^s]+\$	100 字	NA	无	否	用户输入	图像路径
其他材料	字符串	NA	NA	^[^s]+\..pdf\$	100 字	NA	无	否	用户输入	存储路径

#### 6) 数据项之间的关系

无

#### 7) 窗口有效性

从用户点击“公司入驻”按钮开始，到用户离开公司入驻申请页面为止。

#### 8) 与其它输入/输出的关系

作为企业入驻审批、查询企业信息、修改企业信息的输入。

#### 9) 窗口布局

层叠

#### 10) 反馈信息

系统弹窗提示操作成功。

#### 11) 命令的执行方式

用户点击侧边栏的“企业入驻”按钮，填写完信息后点击“确定”按钮执行。

### 2.2.3

## 2.3 功能需求

由于本项目采用面向对象的方法进行易生活服务系统的开发，因此利用 Use-Case 图描述系统的功能需求。

本项目涉及的 Use-Case 图如下：

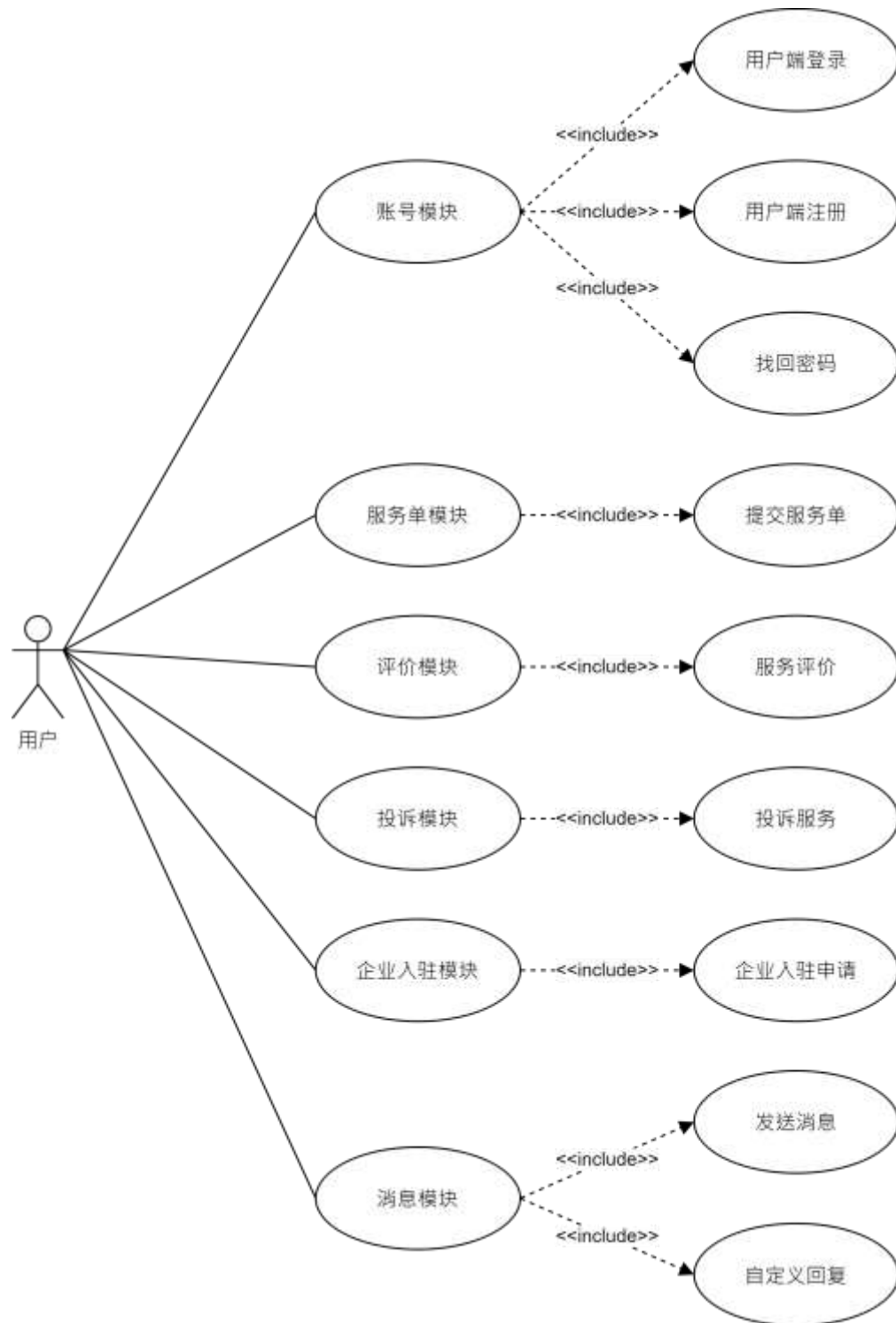


图 2 - 2 用户 Use-Case 用例图

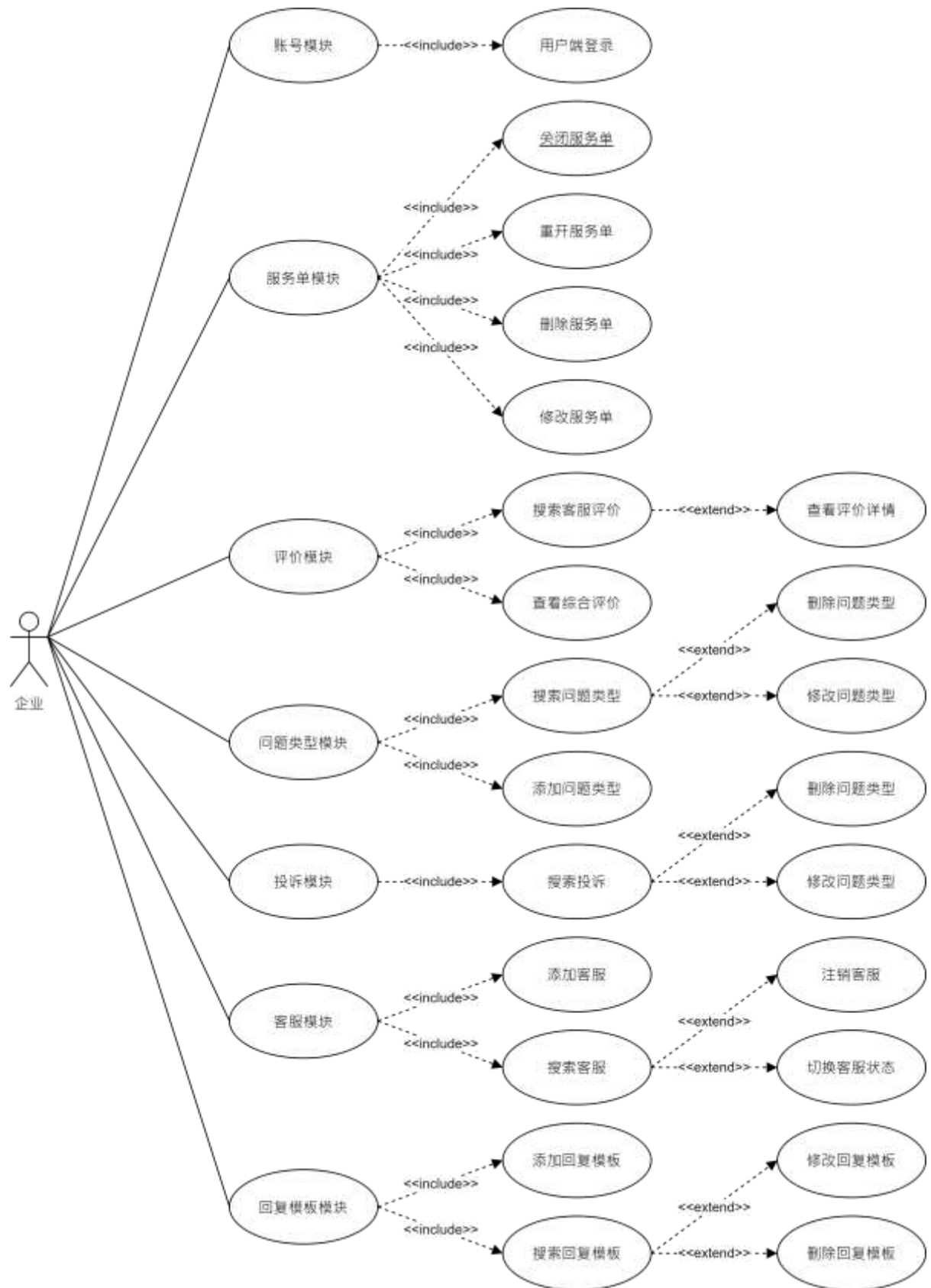


图 2-3 企业 Use-Case 用例图

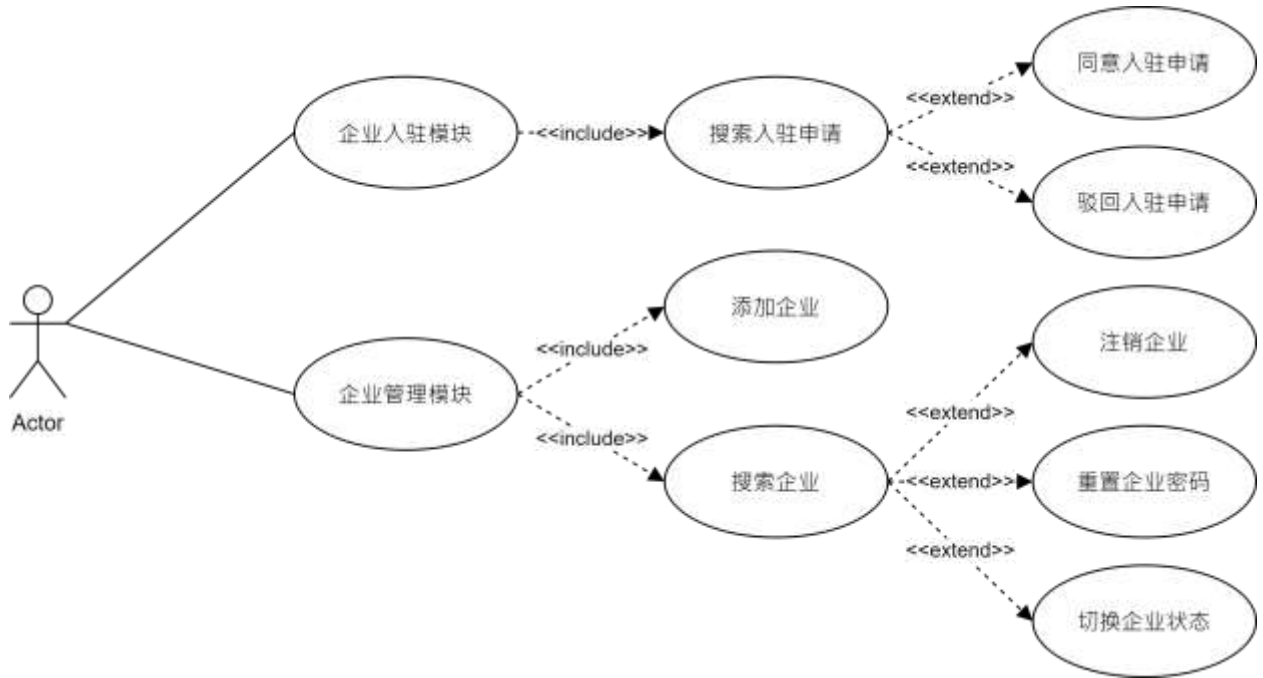


图 2-4 运营人员 Use-Case 用例图

### 2.3.1 用例列表

表 2-4 易生活客户服务系统用例列表描述了易生活客户服务系统中全部的用例。受限于篇幅，本文中仅选取其中数个用例进行详细描述。

表 2-4 易生活客户服务系统用例列表

序号	用例名	用例标识符
1	申请服务单	ApplyTicket
2	提交服务单	SubmitTicket
3	关闭服务单	CloseTicket
4	重开服务单	ReopenTicket
5	删除服务单	DeleteTicket
6	自定义回复	SetReply
7	服务单转发	TicketForward
8	服务评价	Feedback

9	投诉服务	Complaints
11	搜索客服评价	SearchServerEvaluation
12	查看评价详情	SeeEvaluationDetails
13	查看综合评价	SeeComprehensiveEvaluation
14	添加问题类型	AddQuestionType
15	删除问题类型	DeleteQuestionType
16	搜索问题类型	SearchQuestionType
17	修改问题类型	ModifyQuestionType
18	搜索投诉	SearchComplaint
19	查看投诉	SeeComplaint
20	删除投诉	DeleteComplaint
21	企业入驻申请	EnterpriseImparting
22	搜索入驻申请	SearchSettlementApplication
23	同意入驻申请	AgreeSettlementApplication
24	驳回入驻申请	RejectSettlementApplication
25	自定义角色	CompanyRoleManage
26	搜索客服	SearchServer
27	添加客服	AddServer
28	注销客服	CancelServer
29	重置客服密码	ResetServerPassword
30	切换客服状态	ExchangeServerStatus
31	搜索企业	SearchEnterprise
32	添加企业	AddEnterprise
33	注销企业	CancelEnterprise
34	重置企业密码	ResetEnterprisePassword



35	切换企业状态	ExchangeEnterpriseStatus
36	修改员工角色	ModifyCompanyRole
37	撤销员工角色	CancelCompanyRole
38	赋予员工角色	GrantCompanyRole
39	查看日志	ReviewLogs
40	客户端登录	ClientLogin
41	企业端登录	EnterpriseLogin

### 2.3.2 用例表

表 2 - 5 企业入驻审批用例表

用例标识符	EnterpriseImparting		
用例名称	企业入驻审批		
用例创建者	姜山	用例最后修改者	姜山
用例创建时间	2022 年 9 月 20 日	用例最后修改时间	2023 年 2 月 25 日
操作者	企业代表		
描述	企业对企业入驻申请进行审批		
前置条件	企业代表成功登录，系统处于用户端首页(UserInterface)		
后置条件	系统向企业信息表(companyInfoList)中写入一条新的数据，返回用户端首页(UserInterface)		
主事件流	企业代表	系统	
	1 企业代表点击用户端首页(UserInterface)的企业入驻申请按钮	2 系统进入企业入驻申请界面(EnterpriseImpartingView)。	



图 2-6 易生活客户服务系统数据实体关系图

## 2.5 系统的质量属性

本节中，主要从可靠性、可获得性、保密性、可维护性四个方面来描述系统的质量。

### 2.5.1 可靠性

易生活客户服务系统的可靠性为 99%，每年停机维护时间不超过 88 小时；运行时漏洞不超过一个。

### 2.5.2 可获得性

易生活客户服务系统在维护期间用户无法正常获得系统提供的相关服务，其他时间可正常使用。

### 2.5.3 保密性

易生活客户服务系统无论是客户、企业还是管理员想要获得系统服务必须用账号密码登录后才能获取。当密码发生变动时需要使用短信验证码修改或者线下联系管理员帮忙修改。

### 2.5.4 可维护性

如无特殊情况，服务端大型维护不超过 1 天，小型维护不超过 3 个小时。发生特殊情况将适当增加维护时间。

用户端程序允许热更新。

## 2.6 需求模型

### 2.6.1 静态模型

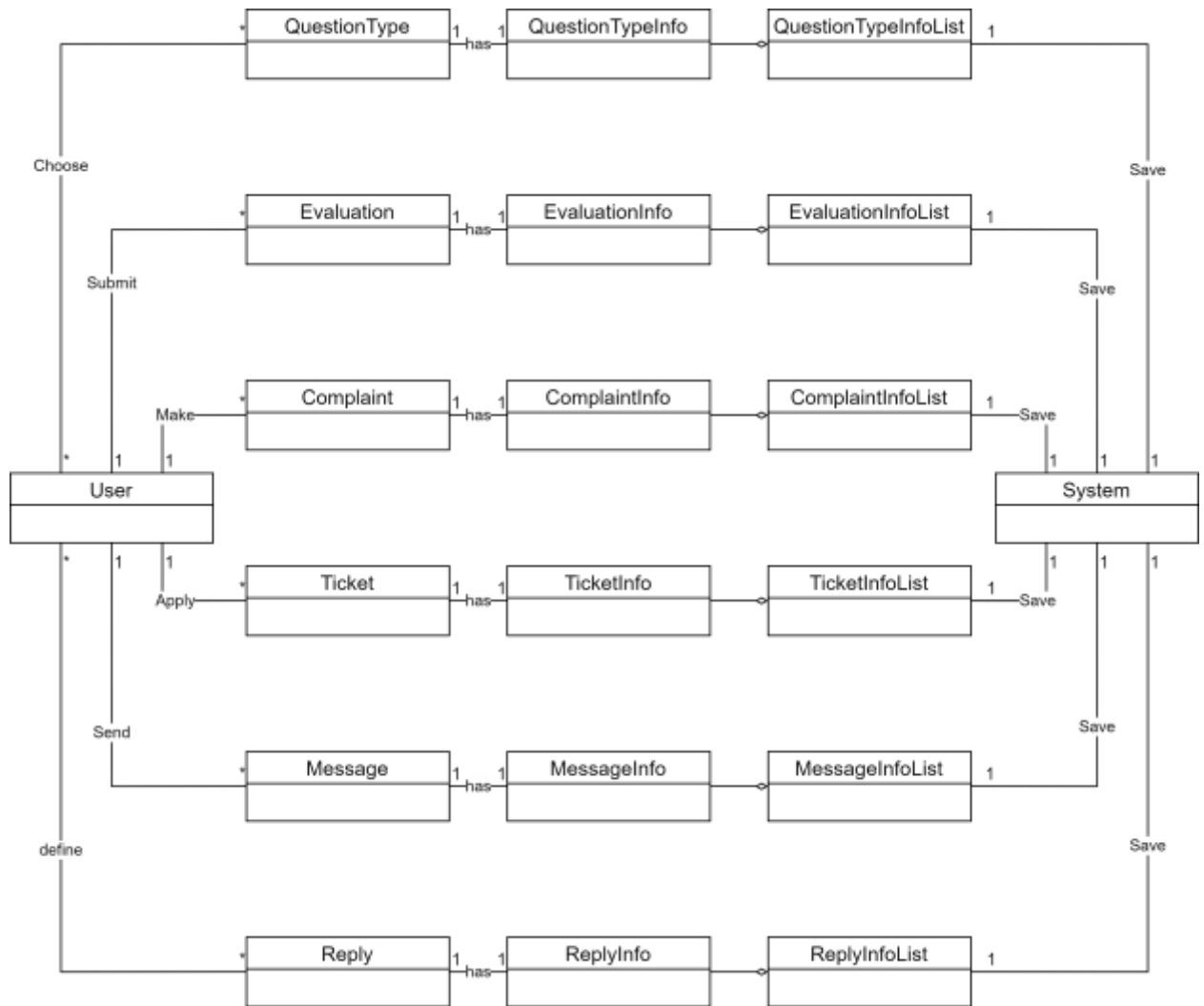


图 2-7 与用户有关的静态模型

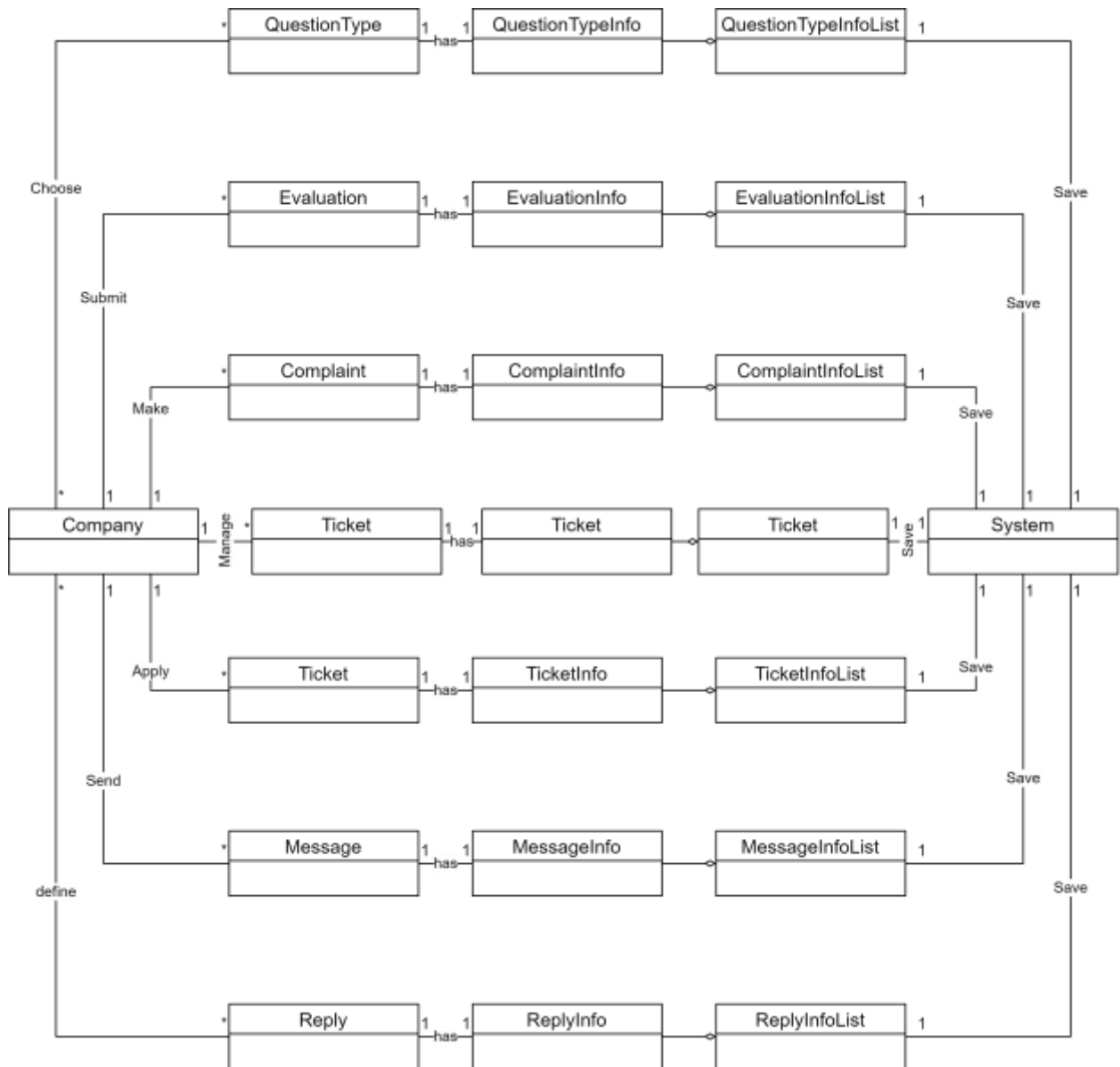


图 2-8 与企业有关的静态模型

## 2.6.2 动态模型

### 1) 企业入驻申请序列图

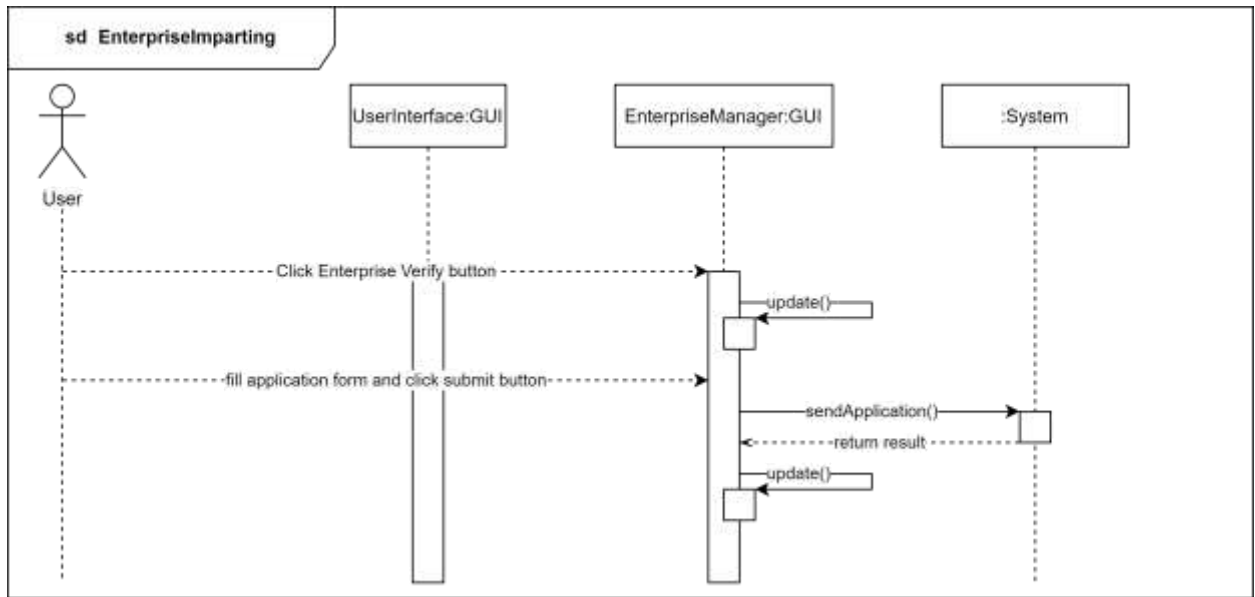


图 2 - 9 企业入驻申请序列图

注：

UserInterface: 用户端主界面

EnterpriseManager: 企业数据管理界面

System::sendApplication(Company company): Company

前置条件： 输入的数据参照 表 3.10 企业入驻申请数据项

后置条件： 系统向企业信息表(CompanyInfoList)中添加一条数据并返回。若操作成功，返回带有申请数据的数据集合；若操作失败，返回 Null。

## 第三章 体系结构设计

### 3.1 部署视图

#### 3.1.1 概述

易生活客户服务系统的部署视图见下图：

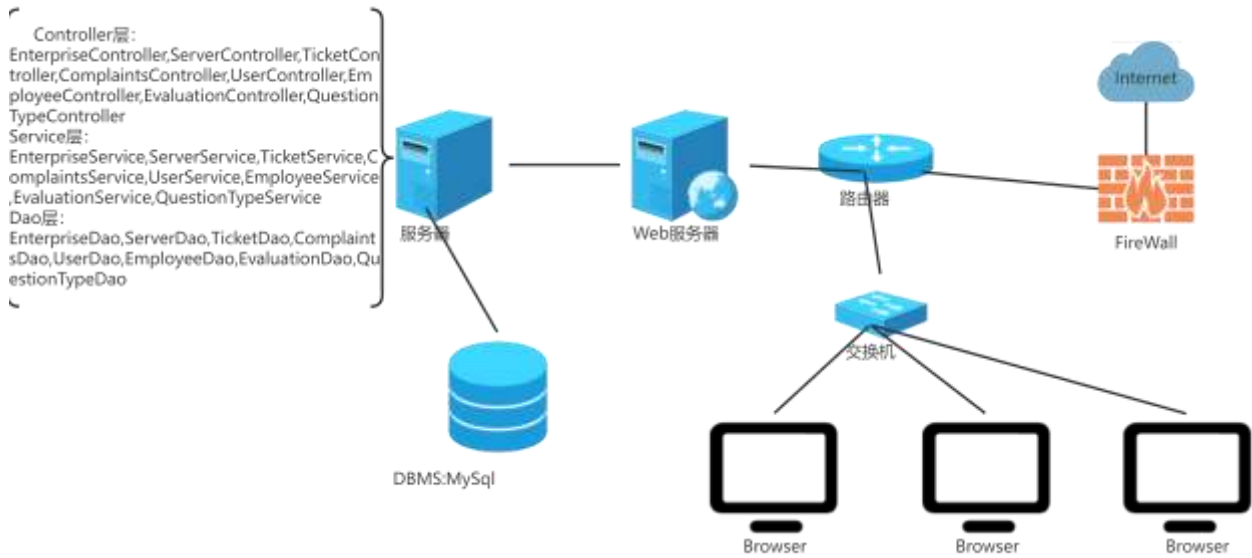


图 3-1 易生活客户服务系统部署图

### 3.1.2 部署政策

易生活客户服务系统的部署政策见下表：

表 3-1 易生活客户服务系统部署政策表

	用户端	服务端
浏览器名称	Google Chrome（谷歌浏览器）	Google Chrome（谷歌浏览器）
浏览器版本	86.0（64 位）版本以上	86.0（64 位）版本以上
应用操作系统	Windows10 及以上	Windows10 及以上
应用服务器软件	Apache Tomcat 9.0 及以上	Apache Tomcat 9.0 及以上
数据服务器操作系统	Windows 10 及以上	Windows 10 及以上
数据服务器数据库版本	SqlServer2017 及以上	SqlServer2017 及以上

## 3.2 逻辑视图

### 3.2.1 概述

易生活服务系统采用 B/S 模式，共分为七层。这七层分别为：

View（用户交互层）、Controller（控制层）、IService（业务服务接口层）、Service（业务服务层）、Mapper（数据访问映射层）、Dao（数据访问层）、Model（领域模型层）和 the object value class（值对象类包）。

易生活客户服务系统包图如下：

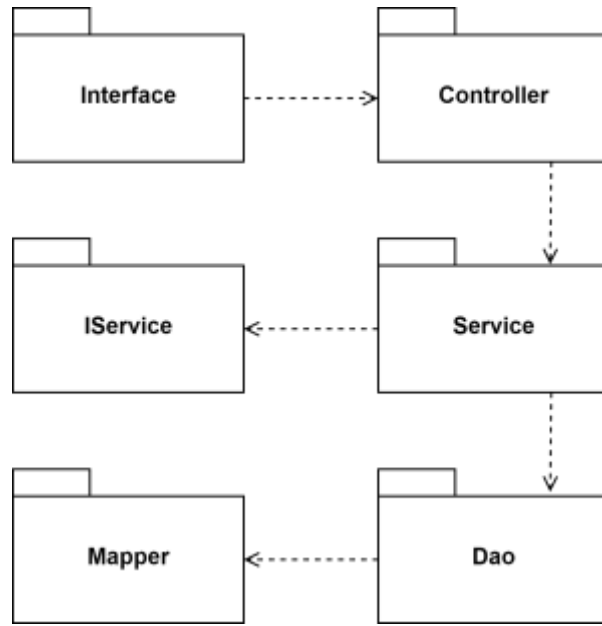


图 3 - 2 易生活客户服务系统包图

### 3.2.2 包或子系统

#### 1) 问题类型类关系图

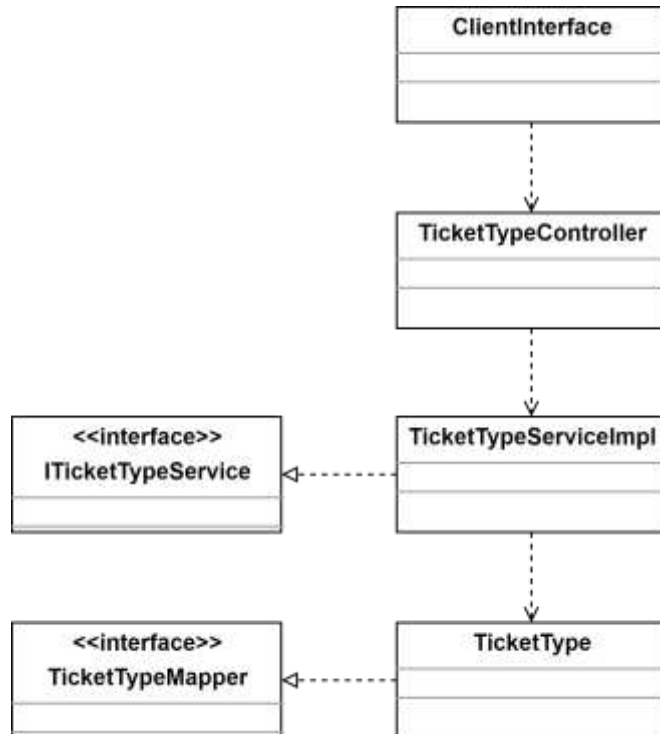


图 3 - 3 问题类型关系图

#### 2) 投诉类关系图



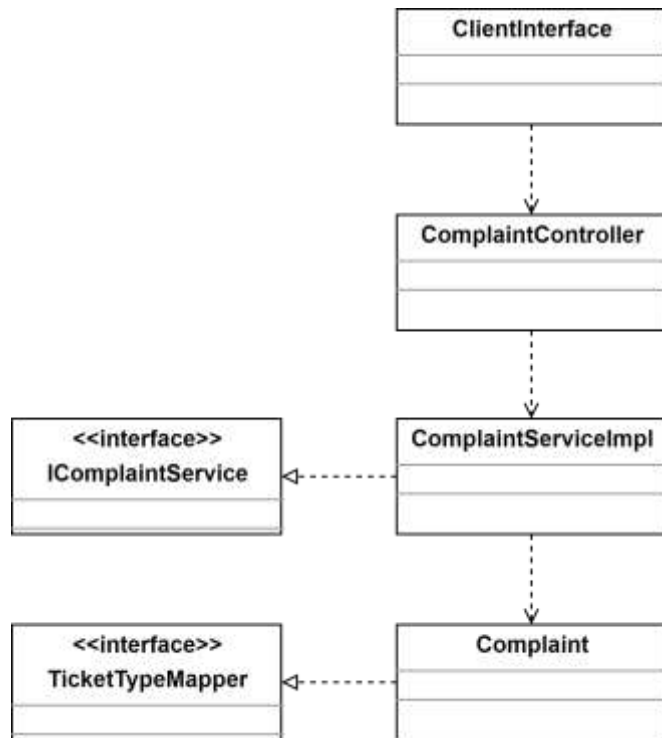


图 3 - 4 投诉关系图

### 3) 评价类关系图

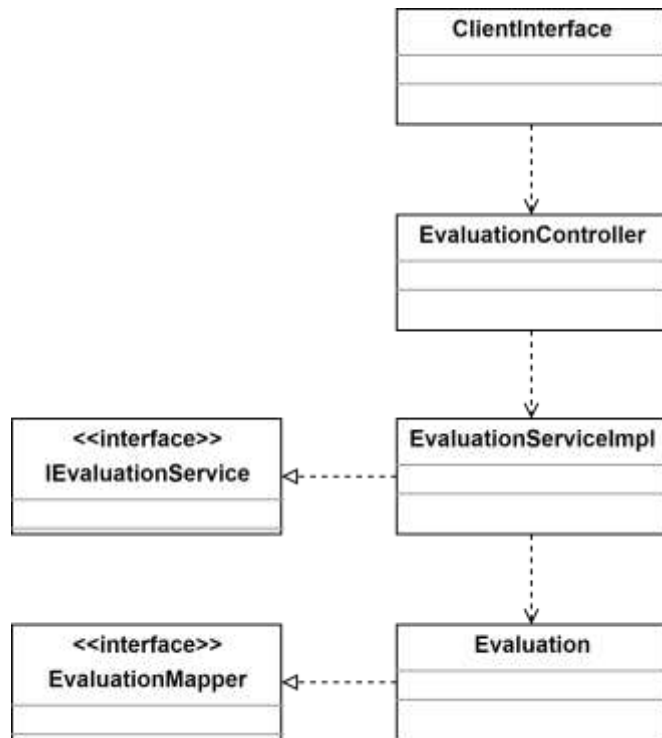


图 3 - 5 评价关系图

### 4) 用户类关系图

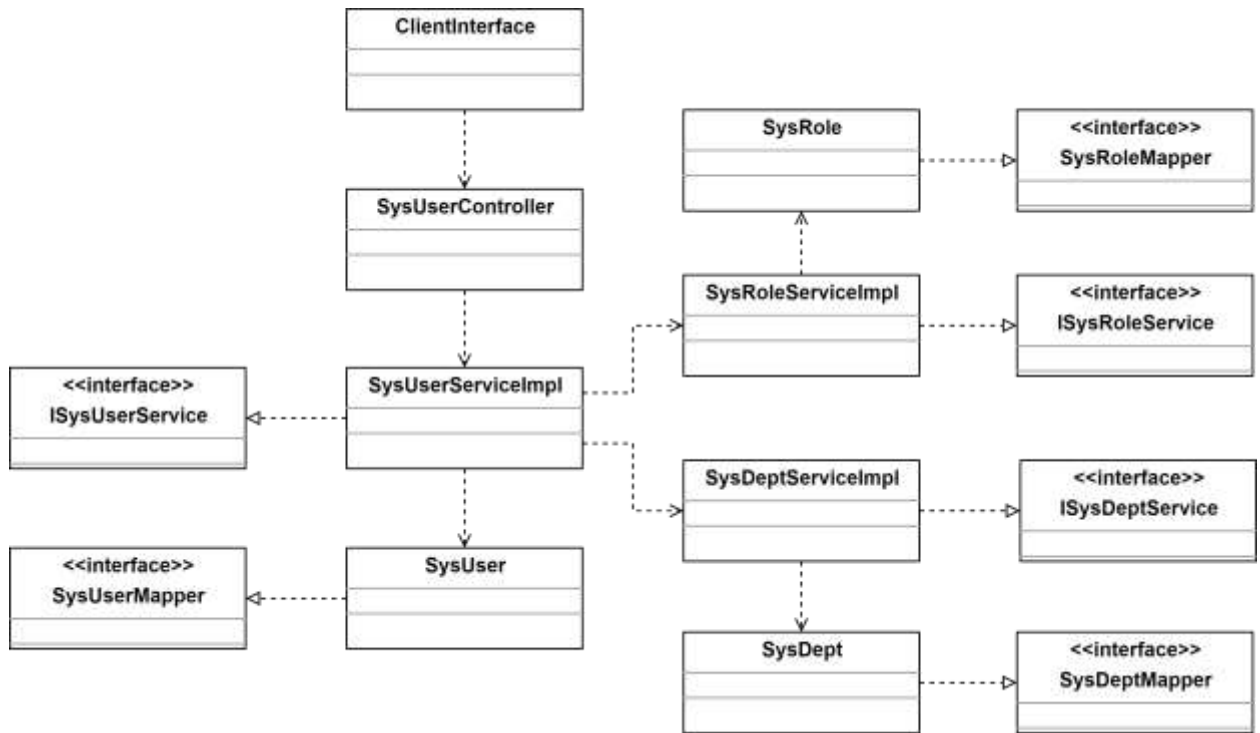


图 3 - 6 用户关系图

### 5) 公司类关系图

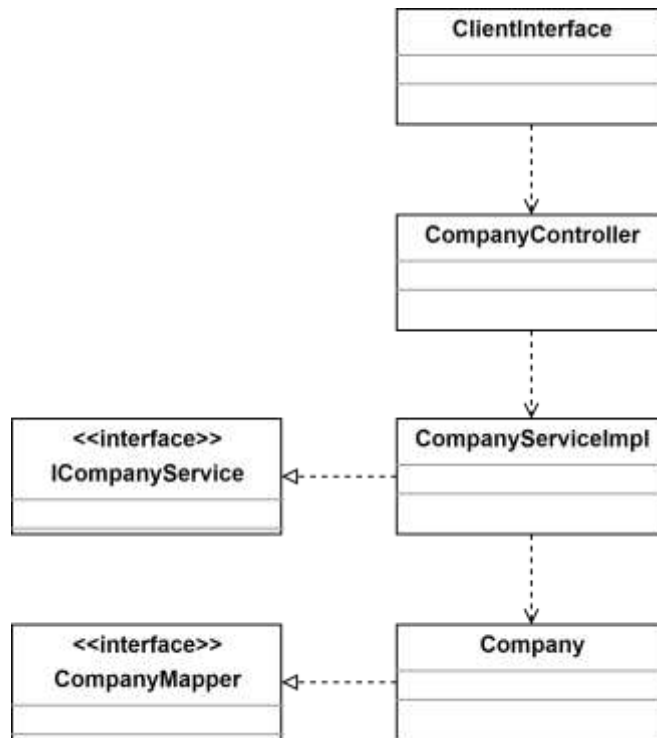


图 3 - 7 公司关系图

## 3.3 进程视图

### 3.3.1 概述

#### 1) 企业入驻申请的进程视图

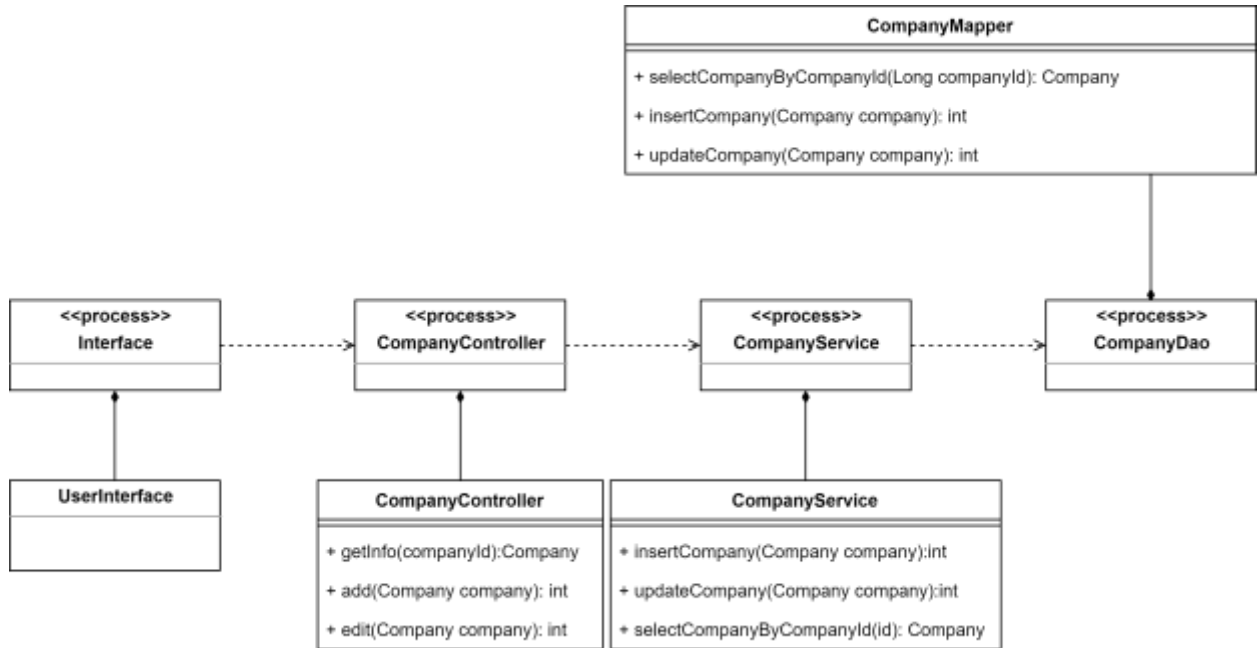


图 3 - 8 企业入驻申请进程视图

### 3.3.2 过程间通信机制

系统客户端与应用服务器间通过 JSON 数据传输格式传输数据,通过 URL 将客户端信息提交到应用服务器。应用服务器与数据库使用 JDBC 连接,通过 TDS 协议进行数据通信。应用服务器与设备之间使用 TCP 通信。

## 3.4 Use-Case 视图

#### 1) 企业入驻申请

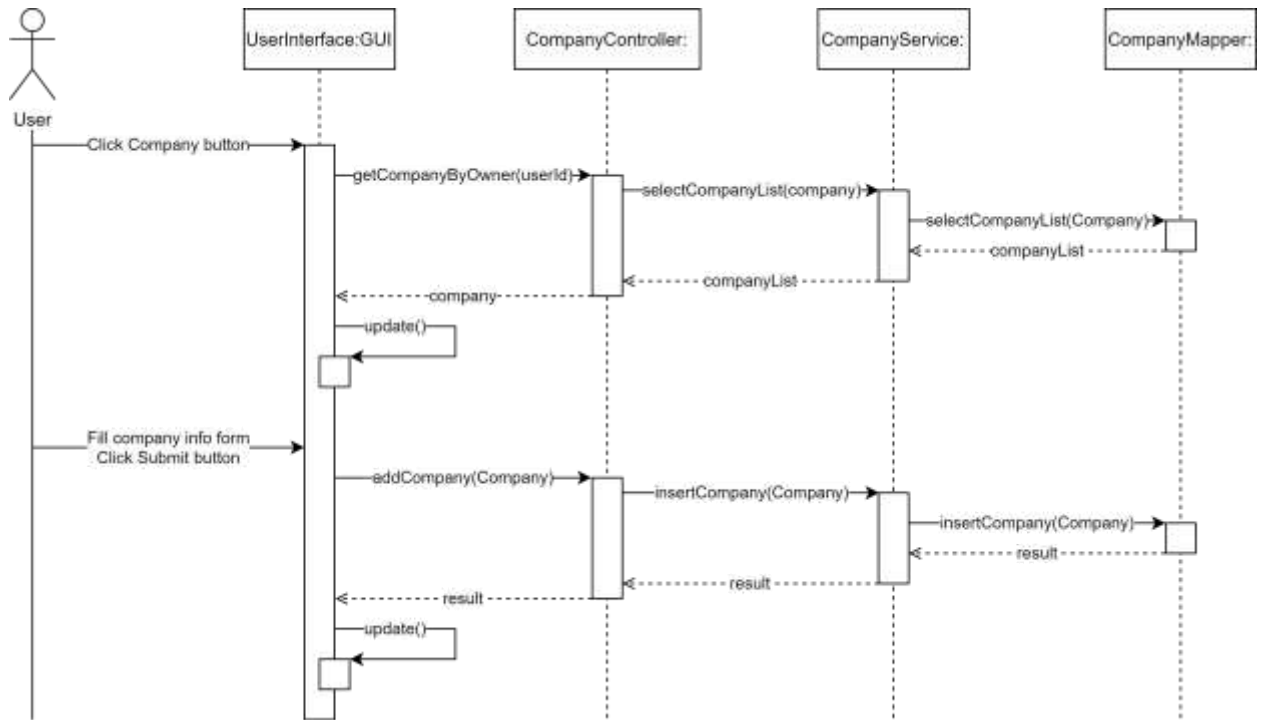


图 3 - 9 企业入驻申请 Use-Case 图

注:

UserInterface::update():void

CompanyController::getCompanyByOwner(Long userId):Company

CompanyService::selectCompanyList(Company company): List<Company>

CompanyMapper::selectCompanyList(Company company):List<Company>

CompanyController::addCompany(Company company):int

CompanyService::insersrtCompany(Company company):int

CompanyMapper::insersrtCompany(Company company):int

### 3.5 实现视图

### 3.5.1 业务流程层

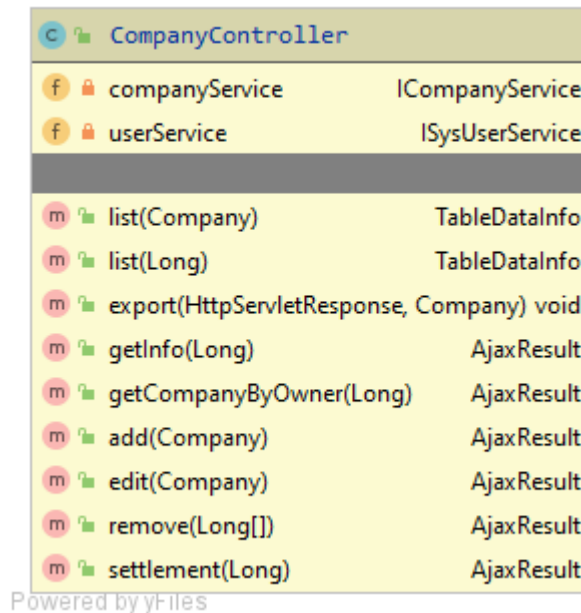


图 3 - 10 CompanyController 类

方法介绍及伪码:

`CompanyController::list(Company company): TableDataInfo`

前置条件: 查询条件(company)不为空

后置条件: 系统根据传入的约束条件从企业信息表(CompanyInfoList)中查询指定范围内符合条件的数据组成的集合, 将其封装成用于前后端交互时的 JSON 格式并返回。

`CompanyController::list(Company company): TableDataInfo {`

`startPage();`

`list ← companyService.selectCompanyList(company);`

`return getDataTable(list);`

`}`

`CompanyController::list(Long companyId): TableDataInfo`

前置条件: 公司 Id(companyId)不为空

后置条件: 系统根据传入的约束条件从用户信息表(UserInfoList)中查询所有符合条件的数据组成的列表, 将其封装成用于前后端交互时的 JSON 格式并返回。

`list(Long companyId): TableDataInfo {`

`List<SysUser> list ← userService.selectUserByCompany(companyId);`

`return getDataTable(list);`

```
}
```

CompanyController::getInfo(Long companyId): AjaxResult

前置条件：公司 Id(companyId)不为空

后置条件：系统从公司信息表(CompanyInfoList)中查询指定公司的数据，将其封装成用于前后端交互的 JSON 格式并返回。

CompanyController::getInfo(Long companyId): AjaxResult {

```
    return
```

```
    AjaxResult.success(companyService.selectCompanyByCompanyId(companyId));
```

```
}
```

CompanyController::getCompanyByOwner(Long userId): AjaxResult

前置条件：用户 Id(userId)不为空

后置条件：系统从公司信息表(CompanyInfoList)中查询指定公司的数据，将其封装成用于前后端交互的 JSON 格式并返回。

CompanyController::getCompanyByOwner(Long userId): AjaxResult {

```
    Company query ← new Company();
```

```
    query.setCompanyMaster(userId);
```

```
    List<Company> resList ← companyService.selectCompanyList(query);
```

```
    If (resList.size() > 0) {
```

```
        return AjaxResult.success(resList.get(0));
```

```
    } else {
```

```
        return AjaxResult.success();
```

```
    }
```

```
}
```

CompanyController::add(Company company): AjaxResult

前置条件：公司信息(company)不为空

后置条件：系统设置新公司的初始信息，向公司信息表(CompanyInfoList)中新增一条数据并返回操作结果。

CompanyController::add(Company company): AjaxResult {

```
    company.setCompanyMaster(getLoginUser().getUserId());
```

```
    company.setApplyDate(DateUtils.getNowDate());
```

```

        company.setStatus(0);
        return toAjax(companyService.insertCompany(company));
    }

```

**CompanyController::edit(Company company): AjaxResult**

前置条件：公司信息(company)不为空

后置条件：系统在公司信息表(CompanyInfoList)中修改指定公司的数据并返回操作结果。

```

CompanyController::edit(Company company): AjaxResult {
    return toAjax(companyService.updateCompany(company));
}

```

**CompanyController::remove(Long[] companyIds): AjaxResult**

前置条件：公司 Id 列表(companyIds)不为空

后置条件：系统从公司信息表(CompanyInfoList)中删除指定公司的数据并返回操作结果。

```

CompanyController::remove(Long[] companyIds): AjaxResult {
    return toAjax(companyService.deleteCompanyByCompanyIds(companyIds));
}

```

**CompanyController::settlement(Long companyId): AjaxResult**

前置条件：公司 Id(companyId)不为空

后置条件：系统从公司信息表(CompanyInfoList)中修改指定公司的状态为“已入驻”并返回操作结果。

```

CompanyController::settlement(Long companyId): AjaxResult {
    return toAjax(companyService.updateCompany(company));
}

```

### 3.5.2 业务逻辑层

#### 1) CompanyService

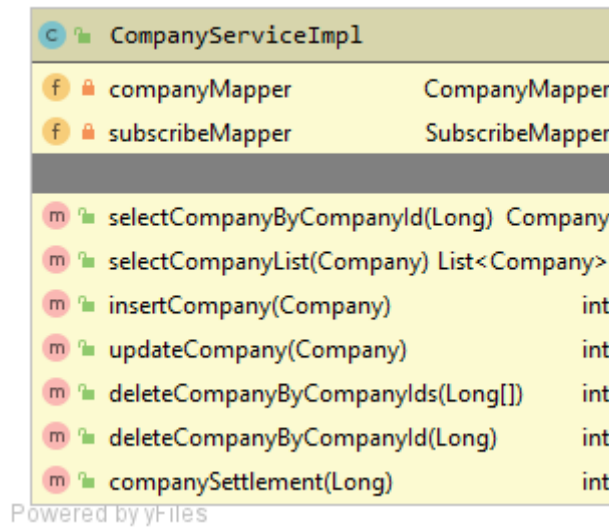


图 3 - 11 CompanyService 类

方法介绍及伪码:

CompanyService::selectCompanyByCompanyId(Long companyId): Company

前置条件: 公司 Id(companyId)不能为空。

后置条件: 系统从公司信息表(CompanyInfoList)中选择对应的公司信息并将其返回。

```
CompanyService::selectCompanyByCompanyId(Long companyId): Company{
    return companyMapper.selectCompanyByCompanyId(companyId);
}
```

CompanyService::selectCompanyList(Company company):List<Company>

前置条件: 公司信息(company)不能为空。

后置条件: 系统从公司信息表(CompanyInfoList)中选择对应的公司信息组成的列表, 并将其返回。

```
CompanyService::selectCompanyList(Company company):List<Company> {
    return companyMapper.selectCompanyList(company);
}
```

CompanyService::insertCompany(Company company): int

前置条件: 公司信息(company)不能为空。

后置条件: 系统向公司信息表(CompanyInfoList)中新增一条公司信息, 并将结果返回。

```
CompanyService::insertCompany(Company company): int {
    company.setCreateTime(DateUtils.getNowDate());
```



```
        return companyMapper.insertCompany(company);
    }
}
```

CompanyService::updateCompany(Company company) :int

前置条件：公司信息(company)不能为空。

后置条件：系统在公司信息表(CompanyInfoList)中修改一条公司信息，并将结果返回。

```
CompanyService::updateCompany(Company company) :int {
    company.setUpdateTime(DateUtils.getNowDate());
    return companyMapper.updateCompany(company);
}
```

CompanyService::deleteCompanyByCompanyIds(Long[] companyIds): int

前置条件：公司编号列表(companyIds)不能为空。

后置条件：系统在公司信息表(CompanyInfoList)中删除所有编号在传入参数中的数据，并将操作结果返回。

```
CompanyService::deleteCompanyByCompanyIds(Long[] companyIds): int {
    return companyMapper.deleteCompanyByCompanyIds(companyIds);
}
```

CompanyService::deleteCompanyByCompanyId(Long companyId): int

前置条件：公司编号(companyId)不为空。

后置条件：系统在公司信息表(CompanyInfoList)中删除一条数据，并返回操作结果。

```
CompanyService::deleteCompanyByCompanyId(Long companyId): int {
    return companyMapper.deleteCompanyByCompanyId(companyId);
}
```

CompanyService::companySettlement(Long companyId):int

前置条件：公司编号(companyId)不为空。

后置条件：系统从公司信息表(CompanyInfoList)中修改指定公司的状态为“已入驻”并返回操作结果。

```
CompanyService::companySettlement(Long companyId):int {
    Company company = companyMapper.selectCompanyByCompanyId(companyId);
    if(company.getStatus() == 0){
```

```

        company.setStatus(1);
        return companyMapper.updateCompany(company);
    }else {
        return 0;
    }
}

```

### 3.5.3 数据访问层

#### 1) Company

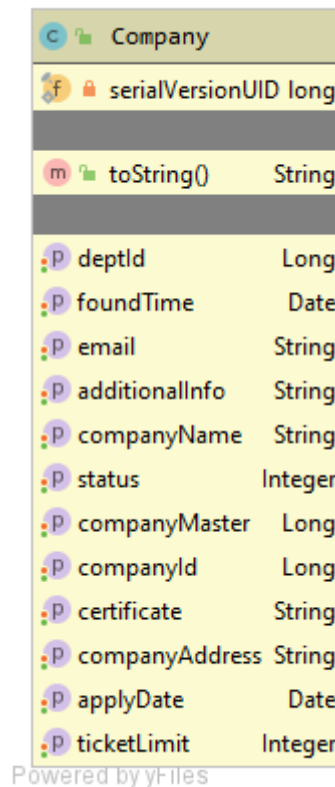


图 3 - 12 Company 类

方法介绍及伪码：

Company::toString():String

前置条件：无

后置条件：返回由当前数据格式化后的字符串。

Company::toString():String{

    return new ToStringBuilder(this, ToStringStyle.MULTI\_LINE\_STYLE)

```

.append("companyId", getCompanyId())
.append("deptId", getDeptId())
.append("companyName", getCompanyName())
.append("companyAddress", getCompanyAddress())
.append("email", getEmail())
.append("foundTime", getFoundTime())
.append("certificate", getCertificate())
.append("additionalInfo", getAdditionalInfo())
.append("applyDate", getApplyDate())
.append("companyMaster", getCompanyMaster())
.append("ticketLimit", getTicketLimit())
.append("state", getStatus())
.append("createBy", getCreateBy())
.append("createTime", getCreateTime())
.append("updateBy", getUpdateBy())
.append("updateTime", getUpdateTime())
.toString();
}

```

注：ToStringBuilder 为 Apache 公司开发的 Commons lang 项目中一个用于构建字符串的工具类。

## 3.6 数据视图

### 3.6.1 逻辑设计

物理模型设计图

#### 1) CompanyInfoList(企业信息表)

表 3 - 2 CompanyInfoList 物理模型

名称	数据类型	长度限制	缺省值	可缺省	可空
company_id	int	20	NA	否	否
company_name	varchar	50	NA	否	否
company_address	varchar	100	NA	否	是
email	varchar	50	NA	否	是

found_time	datetime	NA	当前时间	是	否
certificate	varchar	NA	NA	否	否
additional_info	varchar	NA	NA	否	是
apply_date	datetime	NA	当前时间	是	否
ticket_limit	int	4	2	是	否
status	int	4	0	是	否

## 2) UserInfoList(用户信息表)

表 3 - 3 UserInfoList 物理模型

名称	数据类型	长度限制	缺省值	可缺省	可空
user_id	int	20	NA	否	否
dept_id	int	20	NA	否	是
user_name	varchar	30	NA	否	是
nick_name	varchar	30	NA	否	是
user_type	varchar	2	00	是	是
email	varchar	50	空字符串	是	是
phonenummer	varchar	11	空字符串	是	是
sex	char	1	0	是	是
avatar	varchar	100	空字符串	是	是
password	varchar	100	空字符串	是	是
status	char	1	0	是	是
del_flag	char	1	0	是	是
login_ip	varchar	128	空字符串	是	是
login_date	datetime	NA	当前时间	是	是
create_by	varchar	64	空字符串	是	是
create_time	datetime	NA	当前时间	是	是

update_by	varchar	64	空字符串	是	是
update_time	datetime	NA	当前时间	是	是
remark	varchar	500	空字符串	是	是

### 3) TicketTypeInfoList

表 3 - 4 TicketTypeInfoList 物理模型

名称	数据类型	长度限制	缺省值	可缺省	可空
type_id	int	20	NA	否	否
dept_id	int	20	NA	否	是
type_company	int	20	NA	否	是
type_name	varchar	32	空字符串	是	是
description	varchar	64	空字符串	是	是

### 3.6.2 数据访问机制

客户、企业和管理员通过账号密码登录，账号密码需要通过 MySQL 匹配验证，根据不同角色权限进行数据操纵。

## 第四章 详细设计

### 4.1 用户交互层

#### 4.1.1 企业入驻审批

```
@RestController
@RequestMapping("/feature/company")
public class CompanyController extends BaseController {
    @Autowired
    private ICompanyService companyService;

    @Autowired
    private ISysUserService userService;

    /**
     * 新增公司管理
     */
    @PreAuthorize("@ss.hasPermi('feature:company:add')")
    @Log(title = "公司管理", businessType = BusinessType.INSERT)
    @PostMapping
    public AjaxResult add(@RequestBody Company company) {
        // 初始化公司数据
        // 公司所有者为当前登录用户
        company.setCompanyMaster(getLoginUser().getUserId());
        // 公司申请提交日期为当前日期
        company.setApplyDate(DateUtils.getNowDate());
        // 初始状态为等待审批
        company.setStatus(0);
        return toAjax(companyService.insertCompany(company));
    }
}

import store from "@store";
import {
```

```

    getCompanyByMaster,
    addCompany,
    updateCompany,
  } from "@/api/feature/company";
export default {
  name: "SettlementForm",
  data() {
    return {
      // 表单参数
      form: {},
      // 表单校验
      rules: {
        email: [
          { required: true, message: "电子邮箱不能为空", trigger: "blur" },
        ],
        companyName: [
          { required: true, message: "公司名称不能为空", trigger: "blur" },
        ],
        certificate: [
          { required: true, message: "请提交营业执照", trigger: "blur" },
        ],
        foundTime: [
          { required: true, message: "成立时间不能为空", trigger: "blur" },
        ],
      },
    };
  },
  props: {},
  methods: {
    queryCompany() {
      getCompanyByMaster(this.$store.state.user.userId).then((response) => {
        console.log(response);
        if (response.data) {

```

```

        this.form = response.data;
        this.resetForm("form");
    } else {
        this.reset();
    }
});
},
handleSubmit() {
    this.$refs["form"].validate((valid) => {
        if (valid) {
            if (this.form.companyId != null) {
                updateCompany(this.form).then((response) => {
                    this.$modal.msgSuccess("公司信息修改成功");
                    this.queryCompany();
                });
            } else {
                addCompany(this.form).then((response) => {
                    this.$modal.msgSuccess("申请提交成功");
                    this.queryCompany();
                });
            }
        }
    });
},
reset() {
    this.form = {
        companyId: null,
        companyName: null,
        companyAddress: null,
        email: null,
        foundTime: null,
        certificate: null,
        additionalInfo: null,
    }
}

```



```
        applyDate: null,
        companyMaster: null,
        ticketLimit: null,
        status: null,
        createBy: null,
        createTime: null,
        updateBy: null,
        updateTime: null,
    };
    this.resetForm("form");
},
},
created() {
    this.queryCompany();
},
};
```

## 第五章 测试报告

## 参考文献

- [1] 陈锦洪.陈畴镛.基于 AHP 的供应链合作伙伴选择[J].杭州电子工业学院学报, 2001, 21(3): 94-98.
- [2] 陈荣.物流供应链管理[M].大连: 东北财经大学出版社, 2001, 126-135.
- [3] 马士华.林勇.陈志祥.供应链管理[M].北京: 机械工业出版社, 2000, 125-153.
- [4] 王能民.孙林岩.汪应洛.绿色制造模式下的供应商选择[J].系统工程, 2001, 19(2): 37-41.
- [5] 赵美瑜.供应链战略管理及其评价研究[D]:[博士学位论文].西安: 西安交通大学, 2000.
- [6] 周新意.供应链管理中供应商的评价与选择分析[J].价值工程, 2002, (3): 16-19.
- [7] Filip Roodhooft .& Jozaf Konings. Vendor selection and evaluation: An activity based costing approach [J].European Journal of Operational Research, 1996, 96(1): 97-102.
- [8]Lee C. & Kasilingam R G. Vendor Selection Methods: A Survey of Industry Perceptions[R]. Department of Industrial Engineering, University of Arkansas. 1995.
- [9] Weber C A. Current & J R. Benton W C. Vendor Selection Criteria and Methods [J]. European Journal of Operational Research, 1991, 50(1):2-18.
- [10]Yahya S. & Kingsman B. Vendor rating for an entrepreneur development programme: a case study using the analytic hierarchy process method [J]. Journal of Operational Research Society, 1999, (50): 916-930.

## 致谢

在整个论文的写作过程中，我的指导老师某某某在定稿之前给予我多次指导，使论文得以顺利完成，在此表示衷心的感谢；同时，对在此过程中给予我无私帮助的某某某等老师表示诚挚的谢意。

**致谢语的具体写法，请根据自己的实际需要而定，以上仅供参考。**

山东工商学院  
本科生毕业论文（设计）开题报告  
（2023 届）

学生姓名 \_\_\_\_\_

院（部） \_\_\_\_\_

专 业 \_\_\_\_\_

指导教师（签名） \_\_\_\_\_

年 月 日

## 本科毕业论文开题报告表

学生姓名		所属学院		专业	
班级				学号	
指导教师	姓名		职称		
开 题 报 告 内 容	题目（中、英文）：				
	选题的目的和意义：				
	（可根据情况加页）				
	研究方向、思路和重点：				
	（可根据情况加页）				
	文献资料（包括与本课题相关的国内、外研究现状的资料）：				
（可根据情况加页）					
进度安排：					
指导教师 意见	<div style="text-align: right;">           指导教师签名：            年    月    日         </div>				
本科 毕业 论文 指导 小组 意见	<div style="text-align: right;">           年    月    日         </div>				

# 山东工商学院 本科生毕业论文指导记录

姓名:

年（班）级：

专业:

学 号:

题目：

日期	指导方式(面谈、电话、微信、QQ、E-mail 等)	指导内容	指导意见

(此表仅供参考, 可附页)

指导教师（签名）：\_\_\_\_\_