



哈尔滨工业大学(威海)  
Harbin Institute of Technology at Weihai

# 综合课程设计报告

2016 级

设计题目： 博客网站的设计与开发

院 系： 计算机科学与技术学院

角色	学号	姓名	组内分数	小组成绩
组长	161110218	孙浩		
组员	161110220	任正凯		
组员	161110233	杜辰阳		
组员	161110231	陈裕		
组员	161110219	朱耀栋		
组员	161110228	刘泽		

教师评语：

---

---

---

哈尔滨工业大学(威海)

二零一九年六月

## 哈尔滨工业大学（威海）综合课程设计任务书

姓 名： 孙浩

学 号： 161110218

院（系）： 计算机科学与技术暨软件学院 专 业： 软件工程

任务起止日期： 2019 年 5 月 27 日 至 2019 年 6 月 14 日

课程设计题目：

总体题目为基于 Vue 和 Spring Boot 框架实现的博客网站。并对此网站进行设计与开发。

系统可行性分析及主要功能：

### 1. 题目背景与设计意义：

在当今互联网飞速发展的时代，博客已经成为人们生活中不可缺少的一部分。博客已经成为人们在互联网上分享知识，展现自己个性与看发的平台。

而在当下，大多数的博客网站仍然是采用多页的结构，设计美观性较差，同时也会造成页面结构较为冗余且用户体验不好。相反，采用单网页、用户体验较好的博客网站较少。故而我们可以使用自己的技术来填补这一方面的空白。因此我们决定采用单网页的技术来打造一个博客网站，来使得界面更加简洁美观，用户体验较好。

### 2. 需要解决的问题：

在我们的这个设计中，我们需要解决的问题有：

- 1) 采用 Vue 框架来实现一个较为简洁、界面美观的网页端的应用程序。
- 2) 采用 Vue 对网页上的逻辑进行处理，来实现一个用户体验较好的 SPA 应用程序。
- 3) 采用 Spring Boot 框架来实现对后端逻辑的开发与数据库开发，向前端提供相应的接口。
- 4) 使用 Axios 或者其他工具来实现前后端接口的对接，完成相应的功能。

### 3. 主要功能：

- 1) 用户相关的功能：注册、登陆、修改资料、修改密码、注销账户等。
- 2) 博客的功能：发表博客、删除博客、查看博客、修改博客、评论博客、增加阅读量等操作。
- 3) 网页的其他的功能：推荐博客、搜索博客、查看作者主页等操作。
- 4) 后台管理的功能：从后台对博客进行管理，如删除博客、关闭用户等。

工作量：

1. 前端界面的设计：尽量美观、简介，不要过于冗余。
2. 前端界面的实现：使用 Vue 和其他工具实现前端的开发，尽量模块化，层次调用分明。
3. 后端接口的设计：根据前端要提供的功能，在后端进行接口设计与开发，通过接口向前端提供数据。
4. 后端服务层的设计与实现：根据要实现的功能，对后端的服务层进行设计，并对其进行实现，实现主要的业务逻辑操作。
5. 数据库的设计与实现：分析数据之间的关系，通过 ER 图来进行数据库表的分析，并实现该数据库，使用 JPA 或者 Hibernate 实现向逻辑层提供数据。
6. 前后端接口的对接与测试工作。

工作计划安排：

1. 2019.5.27 与 5.28：完成题目的确定与开题。
2. 2019.6.05 之前：课程设计中期检查之前完成用户模块的开发。
3. 2019.6.13 之前：初步验收之前完成博客模块与其他模块的开发，并将代码及所有文档提交至 SVN。
4. 2019.6.14 之前：完成所有的测试工作并部署，并完成验收。
5. 2019.6.17：提交打印文档。

同组设计者及分工：

1. 孙浩（1611102）：用户相关的后端的开发。
2. 刘泽（1611102）：博客增删改查相关的后端开发。
3. 杜辰阳（1611102）：其他操作（分类，搜索，评论）等的增删改查的后端的开发。
4. 任正凯（1611102）：总体前端界面的设计与实现。
5. 陈裕（1611102）：素材提供与设计，搜索前端界面的设计与实现。
6. 朱耀栋（1611102）：用户模块前端界面的设计与实现。

## 摘 要

### 项目背景：

在当今互联网飞速发展的时代，博客已经成为人们生活中不可缺少的一部分。博客已经成为人们在互联网上分享知识，展现自己个性与看发的平台。

而在当下，大多数的博客网站仍然是采用多页的结构，设计美观性较差，同时也会造成页面结构较为冗余且用户体验不好。相反，采用单网页、用户体验较好的博客网站较少。故而我们可以使用自己的技术来填补这一方面的空白。因此我们决定采用单网页的技术来打造一个博客网站，来使得界面更加简洁美观，用户体验较好。

### 项目意义：

在当今这个多页 IT 博客网站占主导地位的时代，大多数的单页网站是程序员自己用来分享自身看法与经验的地方。而并没有一个统一的网站为他们提供一个分享自身看法经验的单网页社区。于是我们采用单网页的技术来打造一个博客网站，可以供 IT 人员来分享自己的生活经历或编程经验，来填补网络这一个方面的空白。

### 项目主要完成功能：

- 1) 用户相关的功能：注册、登陆、修改资料、修改密码、注销账户等。
- 2) 博客的功能：发表博客、删除博客、查看博客、修改博客、评论博客、增加阅读量等操作。
- 3) 网页的其他的功能：推荐博客、搜索博客、查看作者主页等操作。
- 4) 后台管理的功能：从后台对博客进行管理，如删除博客、关闭用户等。

**关键词：**博客；网站；B/S 架构；Spring MVC；Hibernate；Spring Boot；Vue.js。

# 目 录

摘 要 .....	III
第 1 章 绪论 .....	5
1.1 项目来源 .....	5
1.2 项目开发的目的是意义 .....	5
1.3 国内外相关领域开发及应用现状分析 .....	5
1.4 本文主要研发内容 .....	5
1.5 本文的组织结构 .....	5
第 2 章 系统需求分析 .....	7
2.1 博客网站的业务流程 .....	7
2.2 功能性需求 .....	8
2.2.1 平台管理员需求 .....	8
2.2.2 已注册用户需求 .....	8
2.2.3 游客用户需求 .....	8
2.2.4 平台需求 .....	8
2.3 非功能性需求 .....	9
2.3.1 测试环境 .....	9
2.3.2 性能需求 .....	9
2.3.3 可靠性需求 .....	9
2.3.4 易用性需求 .....	9
2.3.5 安全性需求 .....	9
2.3.6 兼容性需求 .....	10
2.3.7 可移植性需求 .....	10
2.4 系统用例描述 .....	10
2.4.1 平台管理员用例描述 .....	10
2.4.2 已注册用户用例描述 .....	11
2.4.3 游客的用例描述 .....	12
2.5 领域模型 .....	12
2.6 系统状态变化 .....	13
2.6.1 用户账户的状态变化 .....	13
2.6.2 博客的状态变化 .....	13
2.7 系统核心业务流程 .....	14
2.8 系统功能模块划分 .....	15
2.9 本章小结 .....	15
第 3 章 系统概要设计 .....	16
3.1 数据库实体关系 .....	16
3.2 系统架构 .....	16
3.3 系统实体类 .....	18
3.4 工程结构 .....	18
3.5 用户界面框架 .....	19
3.5.1 ***界面布局 .....	19
3.5.2 平台管理员和***界面布局 .....	20
3.6 本章小结 .....	20
第 4 章 系统详细设计 .....	21
4.1 数据库详细设计 .....	21

4.2	平台管理员模块详细设计 .....	22
4.2.1	类图及描述 .....	22
4.2.2	关键流程设计 .....	22
4.3	***模块详细设计 .....	22
4.3.1	类图及描述 .....	22
4.3.2	关键流程设计 .....	22
4.4	***人模块详细设计 .....	23
4.4.1	类图及描述 .....	23
4.4.2	关键流程设计 .....	23
4.5	平台定时模块详细设计 .....	23
4.5.1	类图及描述 .....	23
4.5.2	关键流程设计 .....	24
4.6	本章小结 .....	24
第 5 章	系统实现 .....	25
5.1	平台管理员模块实现 .....	25
5.2	***模块实现 .....	25
5.3	***人模块实现 .....	25
5.4	平台定时模块实现 .....	25
5.4.1	订单检查定时原理 .....	25
5.4.2	项目检查定时原理 .....	25
5.5	本章小结 .....	26
第 6 章	系统测试 .....	27
6.1	JUnit 测试 .....	27
6.2	白盒测试 .....	28
6.3	黑盒测试 .....	29
6.4	本章小结 .....	30
课设总结	.....	31

## 第1章 绪论

本章从项目来源，项目开发的目的和意义，国内外相关领域及应用现状，主要研究内容和组织结构这几个方面对本文的课题和论文的书写做了一个简要的介绍和说明。

### 1.1 项目来源

该项目的主要灵感来源于 CSDN 与网易云音乐（以及其他单页应用程序）。我们既想设计出一个 IT 类的博客网站，又想采用 Vue 等框架所编写的单页应用程序所带来的便利性与良好的用户体验。因此，我们决定开发一个基于单页应用程序的博客网站。

### 1.2 项目开发的目的和意义

大多数的博客网站仍然是采用多页的结构，设计美观性较差，同时也会造成页面结构较为冗余且用户体验不好。相反，采用单网页、用户体验较好的博客网站较少。

在当今这个多页 IT 博客网站占主导地位的时代，大多数的单页网站是程序员自己用来分享自身看法与经验的地方。而并没有一个统一的网站为他们提供一个分享自身看法经验的单网页社区。于是我们采用单网页的技术来打造一个博客网站，可以供 IT 人员来分享自己的生活经历或编程经验，来填补网络这一个方面的空白。提升 IT 人员来浏览博客时的用户体验。

### 1.3 国内外相关领域开发及应用现状分析

在国内外，与 IT 相关的博客网站数不胜数，如国内的 CSDN、CNBLOG 等，国外的也有 StackOverflow 等。但是这些网站都是一些典型的传统的多页应用程序，会带来用户体验上的降低，同时也会影响界面的美观。

同时，单页的应用程序也不在少数，如 163Music 等。但是这些网站都严重的偏向娱乐化，很少与 IT 类相关联。

故而综上所述，使用单网页开发技术开发 IT 类博客网站的在国内和外国确实都在少数，这是一份互联网上的空白。

### 1.4 本文主要研发内容

本文是该项目的一份设计文档，从国内外的现状分析、业务流程到可行性分析。本文也涵盖了对整个系统的需求分析、概要设计、数据库设计、系统架构设计以及详细设计。在整个系统设计完成后，还详细的阐述了该系统的测试情况

### 1.5 本文的组织结构

本文共分六个章节，各章节主要内容安排如下：

第 1 章中，介绍了项目的来源，项目开发的目的和意义，国内外相关领域开发及应用现状，以及本文的主要研究内容和组织结构。

第 2 章中，主要围绕博客业务流程，进行了功能性需求分析和非功能性需求分析，构

建了用例模型和领域模型。

第 3 章中，从数据库实体关系，网络拓扑结构，系统架构，系统实体类，工程架构和用户界面等方面对单页 IT 博客网站进行了概要设计。

第 4 章中，主要完成了博客网站的详细设计工作并对实现效果进行了展示。首先根据数据库实体关系图设计出了数据字典，接着分模块对平台进行了类图设计和关键流程设计。

第 5 章中，主要分模块展示了该博客网站的实现效果。

第 6 章中，使用了 JUnit 单元测试，黑盒测试和白盒测试三种测试手段对前端、后端、博客网站进行了测试工作。

文章的最后部分，对该博客网站的设计与实现所做的工作进行了总结，同时也对平台的发展方向作了展望。



## 第2章 系统需求分析

本章主要围绕博客网站的业务流程使用了面向对象需求分析的方法，对博客网站进行了详细的需求分析，既有功能性需求又有非功能性需求。

### 2.1 博客网站的业务流程

博客网站业务流程如图 2.1-1 所示，包括以下环节：

- (1) 已注册用户相关流程：如修改资料、修改密码、注销账户等。
- (2) 博客相关的流程：如写博客、删除博客、修改博客等，该功能需要用户的个人信息。
- (3) 游客的相关流程：该功能模块不涉及用户的个人信息，包括查看博客、阅读量的修改、搜索博客等。
- (4) 数据操作流程：网站根据用户不同的操作来对已经存储的数据库或其他数据进行修改，如增加博客、更新用户资料等。
- (5) 游客的操作流程：网站不使用用户数据而进行的数据操作，如搜索数据、增加访问量、增加阅读量等功能。
- (6) 管理员的监察流程：管理员可以实时监控用户上传的博客、发表的评论等，此操作需要通过管理员专属的页面。
- (7) 管理员的推荐流程：管理员通过后台可以筛选出较为热门、阅读量较多的博客或者用户，来推荐给用户。
- (8) 管理员的强制操作流程：管理员发现有违法、违规的操作时，可以强制关闭用户或者强制删除博客。

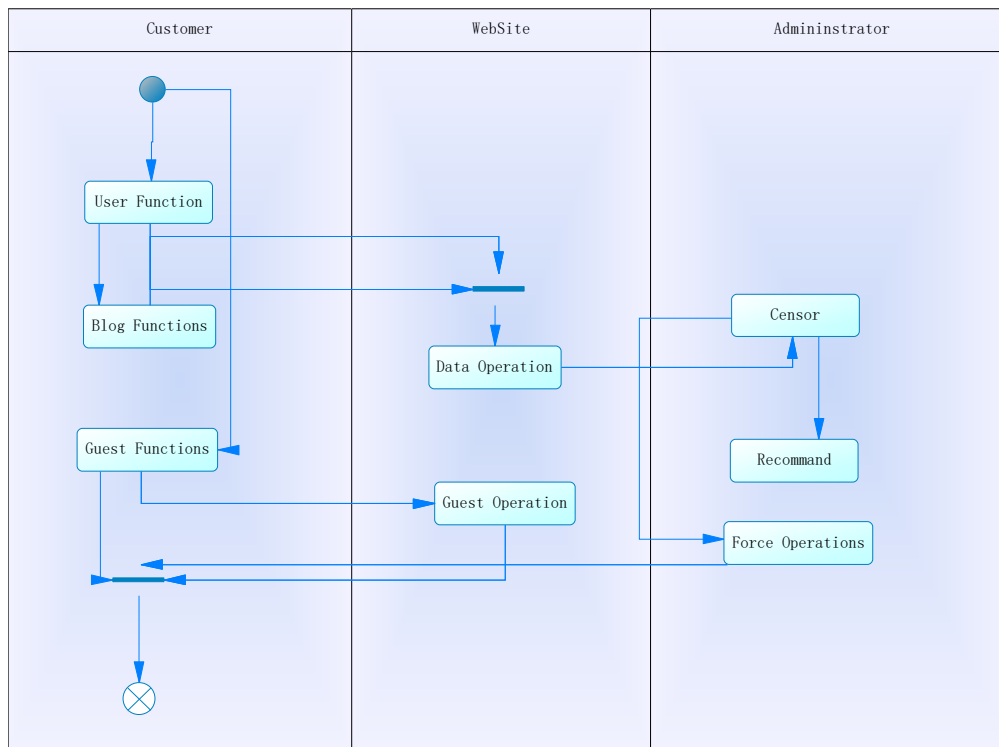


图 2.1-1 业务流程图

## 2.2 功能性需求

根据 IT 类博客网站的基本业务场景，可以将该网站的用户角色分为已注册用户、游客、管理员，不同的角色有不同的功能需求，另外，为了配合三种角色完成整个博客网站的业务流程，网站平台自身也有自动触发的定时任务。

### 2.2.1 平台管理员需求

平台管理员是 IT 博客网站的维护人员和担保人员。

平台管理员可以随时查看近期上传的所有博客的内容。同时管理员可以随时查看所有用户的资料。除此之外，管理员还有各种操作的特权。

平台管理员可以对于不合规定的博客进行删除操作。同时管理员可以通过后台直接关闭某个违规用户的账户。不仅如此，管理员还可以通过后台来直接对用户资料进行管理，如用户的密码等。

### 2.2.2 已注册用户需求

已注册用户是该网站所有博客的内容提供者。

已注册用户的需求描述如下：

- (1) 用户可以注册账户，同时对自己的账户进行注销等操作。
- (2) 用户可以修改自己的账户资料，如密码、性别、电话等。
- (3) 用户可以发表博客，并对自己的博客进行修改。
- (4) 用户可以删除自己的博客。
- (5) 用户可以评论某一个博客，在这篇博客下方留下自己的评论信息。

### 2.2.3 游客用户需求

游客是本网站群体最大的用户类别。

普通游客的需求如下：

- (1) 查看博客，所有的游客可以查看所有的博客与评论。
- (2) 搜索博客的功能，所有的游客可以在本站搜索他想要的博客内容，并查阅搜索出来的结果。
- (3) 注册的功能，游客可以通过我们的注册功能注册成为已注册的用户，进而使用已注册的用户的功能。

### 2.2.4 平台需求

该平台除了给三种角色提供服务之外，还需要处理一些定时的任务，辅助完成整体的业务流程。

- (1) 该平台可以在每次用户阅读博客的时候增加阅读量。
- (2) 该平台可以定时更新推荐的博文，给用户推送最新最热的博文。

## 2.3 非功能性需求

### 2.3.1 测试环境

硬件环境如下：

- (1)CPU：英特尔 i7 四核
- (2)内存：16GB
- (3)硬盘：SSD 256G，硬盘 1T
- (4)网络：校园局域网 100Mbps

软件环境如下：

- (1)操作系统：Windows 10 家庭版 64 位
- (2)JDK：JDK 1.8 64 位
- (3)服务器：Tomcat 8.0
- (4)数据库：MYSQL 8.0.16

### 2.3.2 性能需求

该博客网站在上述运行环境下需要满足以下性能要求：

- (1)支持 30 个用户同时并行操作。
- (2)所有的用户请求应在 3 秒内做出响应。

### 2.3.3 可靠性需求

- (1) Mysql 数据库操作应开启事务处理，以保证在故障出现可以进行事务回滚。
- (2) 浮点数操作精确到小数点后两位四舍五入，统一标准。
- (3) 400 与 500 异常以及其他服务器内部错误需要进行捕获，并进行相应的提示。
- (4) 404 错误需要将用户重定向到错误页。

### 2.3.4 易用性需求

该 IT 博客网站应提供友好美观的扁平滑用户图形界面，采用单网页的设计方式，简单易上手的交互性，尽可能的方便用户进行操作，以满足游客、已注册用户和平台管理员各个使用层面的需求。

### 2.3.5 安全性需求

- (1) 该平台的用户密码和支付密码需要进行 MD5 加密处理，数据库中存储密码的密文形式。
- (2) 用户登录时需根据角色进行权限限定，平台管理员、普通游客和已注册用户三种角色有不同的操作权限。
- (3) 平台管理员通过不同的地址登陆进入平台管理员界面。

- (4) 普通已注册用户登录后进入普通用户界面。
- (5) 未登录的游客操作仅限浏览与搜索博客信息，无法修改任何信息。
- (6) 已登录的已注册用户能够进入个人中心对个人信息进行修改等。

### 2.3.6 兼容性需求

该 IT 博客网站是 B/S 架构的单网页程序，需要考虑到兼容常用的浏览器，具体的说明如下：

- (1)兼容 IE 8 及其以上的 IE 系列浏览器和其他 IE 内核的浏览器，IE 8 以下的浏览器不进行支持。
- (2)兼容 Firefox 及 Firefox 内核的浏览器。
- (3)兼容 Chrome 及 Chrome 内核的浏览器。
- (4)兼容 Safari 浏览器。
- (5)兼容 Opera 浏览器。

### 2.3.7 可移植性需求

- (1) 内测试阶段项目的部署测试在本机进行，公测和商用阶段项目都要部署到服务器上。
- (2) Web 服务和数据库服务器要分开。
- (3) 采用的运行环境为 Linux 系统+Tomcat。

## 2.4 系统用例描述

### 2.4.1 平台管理员用例描述

如图 2.4-1 为平台管理员的需求用例图，平台管理员的需求组成了平台管理员模块。

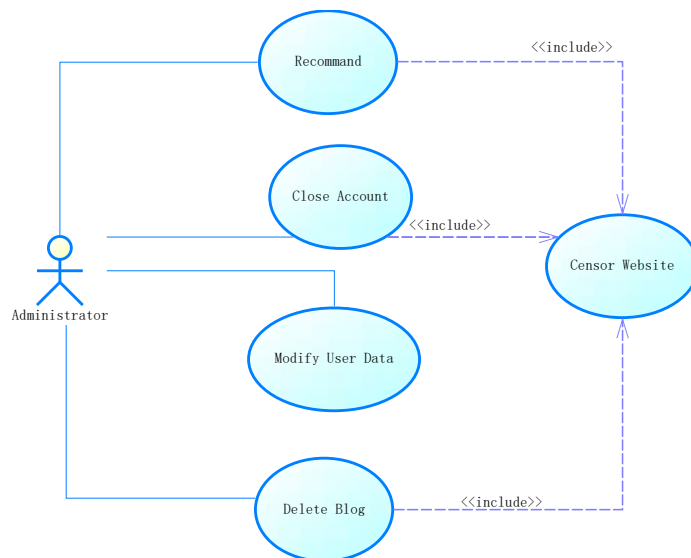


图 2.4-1 平台管理员用例图

平台管理员需求的用例描述如表 2.4-1 所示。

用例名称	参与者	说明
Recommend	管理员	将热门博客推荐到主页。
Close Account	管理员	关闭不和规定的用户。
Modify User Data	管理员	修改用户资料，可以在用户申诉时使用。
Delete Blog	管理员	删除不合规定的博客。
Censor Website	管理员	管理员随时监察新发布的博客等信息。

表 2.4-1 平台管理员用例说明

## 2.4.2 已注册用户用例描述

如图 2.4-2 为已注册用户的需求用例图，已注册用户的需求组成了主要的模块。

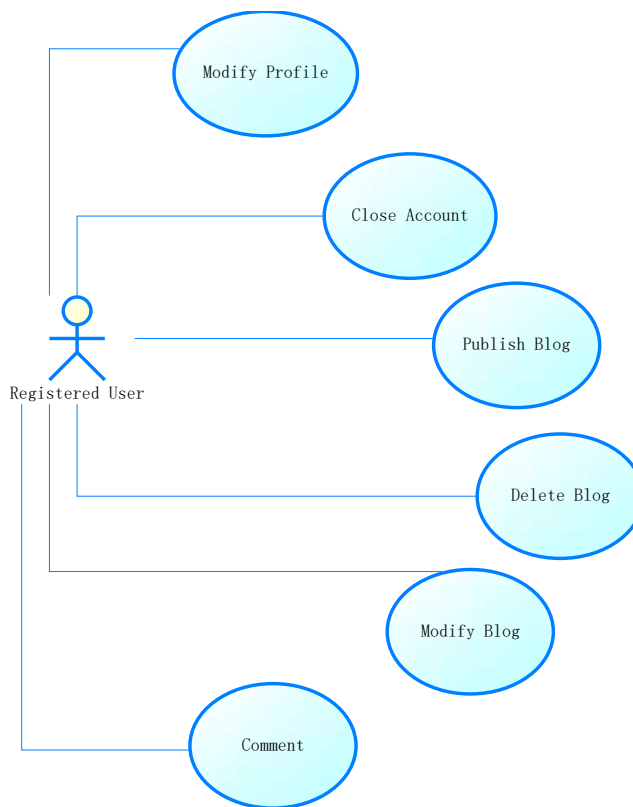


图 2.4-2 已注册用户用例图

已注册用户需求的用例描述如表 2.4-2 所示。

用例名称	参与者	说明
Modify Profile	已注册用户	用户通过该网站修改自己的资料。
Close Account	已注册用户	用户可以通过网站关闭自己的账户。
Publish Blog	已注册用户	用户可以发布自己的博客。
Delete Blog	已注册用户	用户可以删除自己已经发布的博客。

Modify Blog	已注册用户	用户可以修改自己以及发布的博客。
Comment	已注册用户	用户可以在博客下面进行评论。

表 2.4-2 已注册用户用例说明

### 2.4.3 游客的用例描述

如图 2.4-3 为游客用户的需求用例图，游客用户的需求组成了游客的模块。

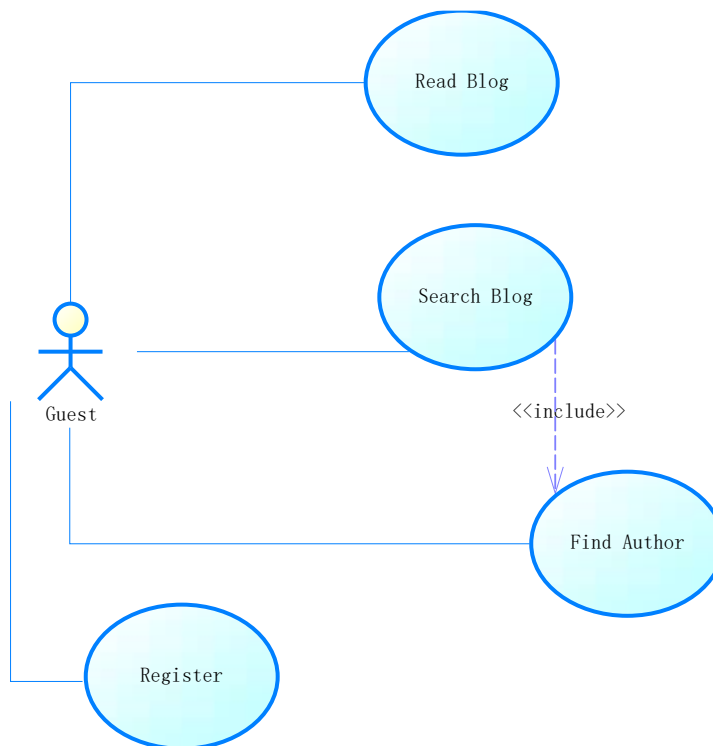


图 2.4-3 游客用户的用例图

游客用户的需求的用例描述如表 2.4-3 所示。

用例名称	参与者	说明
Read Blog	游客	游客可以正常地浏览所有博客。
Search Blog	游客	游客可以搜索自己感兴趣的博客。
Find Author	游客	游客可以查看作者的主页。
Register	游客	游客可以进行注册，并成为正式用户。

表 2.4-3 游客的用例说明

## 2.5 领域模型

经过详细的用例分析可知，实体的主要属性和相互之间的关系如图 2.5-1 错误!未找到引用源。所示。

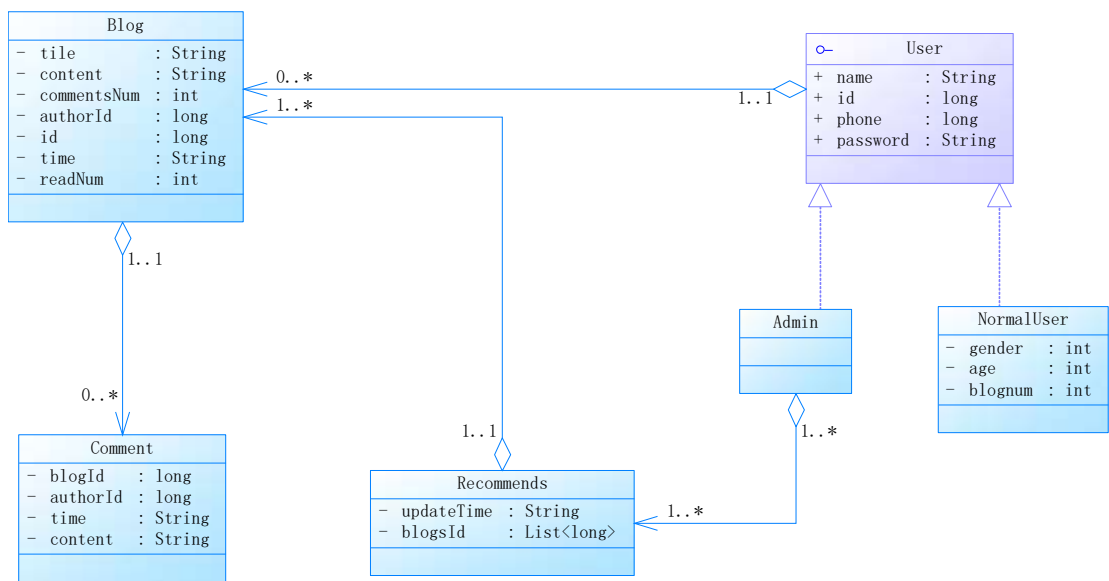


图 2.5-1 领域模型

## 2.6 系统状态变化

整个 IT 博客网站的状态变化主要是用户状态和博客状态的变化。

### 2.6.1 用户账户的状态变化

如图 2.6-1 所示为用户账户改变的状态图。

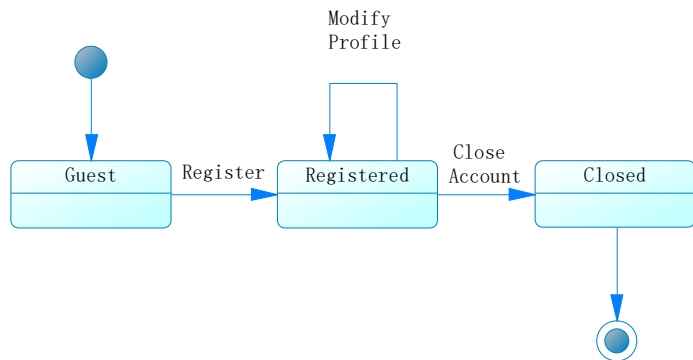


图 2.6-1 用户账户的状态图

### 2.6.2 博客的状态变化

如图 2.6-2 所示为用户发表的博文的状态转换图。

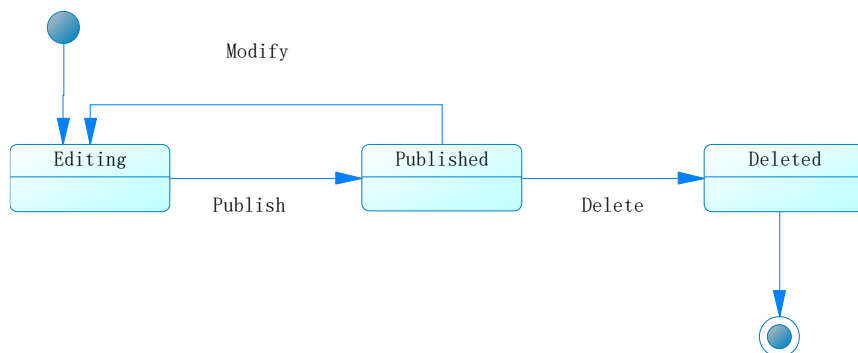


图 2.6-2 博客的状态图

## 2.7 系统核心业务流程

该系统的核心业务流程为用户发表/修改/删除博客的业务流程。

如图 2.7-1 所示为该系统的核心业务流程图。

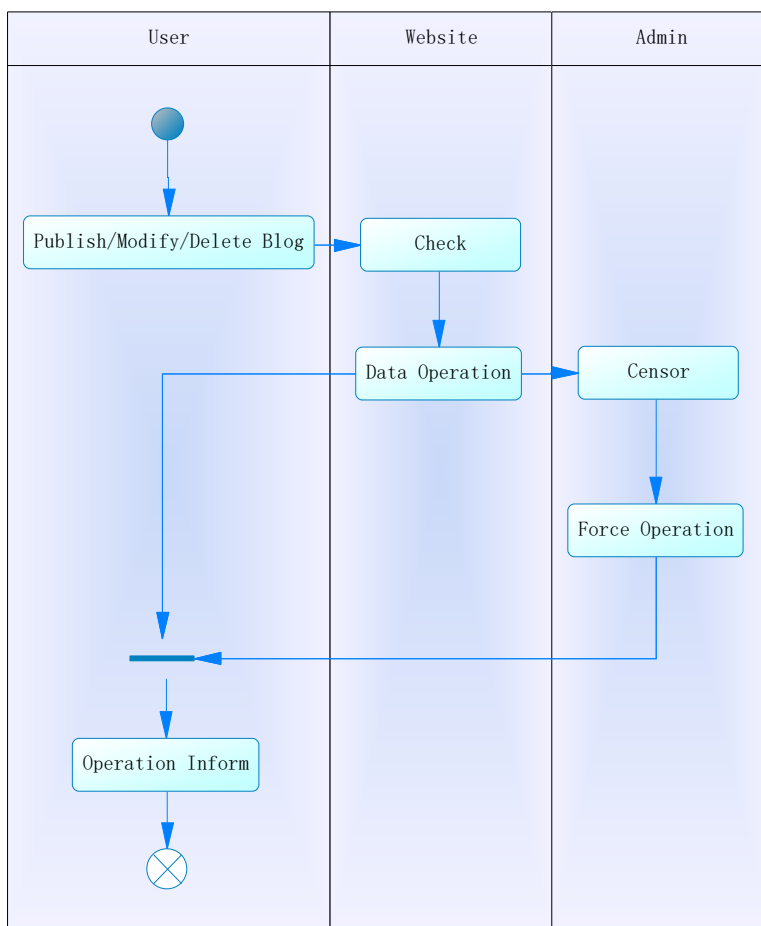


图 2.7-1 核心业务流程图



## 2.8 系统功能模块划分

如图 2.8-1 所示为该 IT 博客系统的功能模块图。

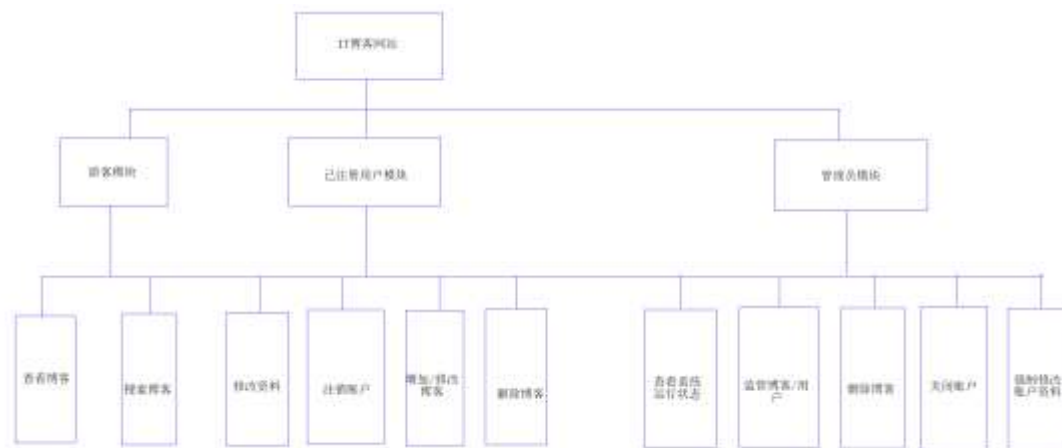


图 2.8-1 功能模块图

## 2.9 本章小结

本章围绕 IT 博客网站的核心业务流程对整个系统进行了详细的需求分析，建立了用例模型和领域模型，分析了系统状态的变化，同时还进行了系统功能模块的划分，为下一章的概要设计打下了基础。

## 第3章 系统概要设计

本章将根据需求分析的成果进行概要设计，主要从数据库实体关系，网络网络拓扑结构，系统架构，实体类，工程结构和用户界面这些方面进行总体设计。

### 3.1 数据库实体关系

如图 3.1-1 所示为\*\*\*的 ER 图，

图 3.1-1 实体关系图

### 3.2 系统架构

\*\*\*采用 B/S 架构进行开发，服务器端采用 Spring MVC 框架，同时和 Hibernate 框架结合，实现了模型层、控制层、业务逻辑层和数据持久层的分离。Spring MVC 框架是一种基于 Java 的请求驱动类型的轻量级 Web 框架，使用了 MVC 架构模式的思想，将 Web 层进行职责解耦。Spring MVC 框架强大的注解支持极大简化了开发过程，利用 Spring MVC 框架对控制层，服务层，数据访问层和模型层使用了注解的类进行扫描注入，免去了实例化对象的麻烦。

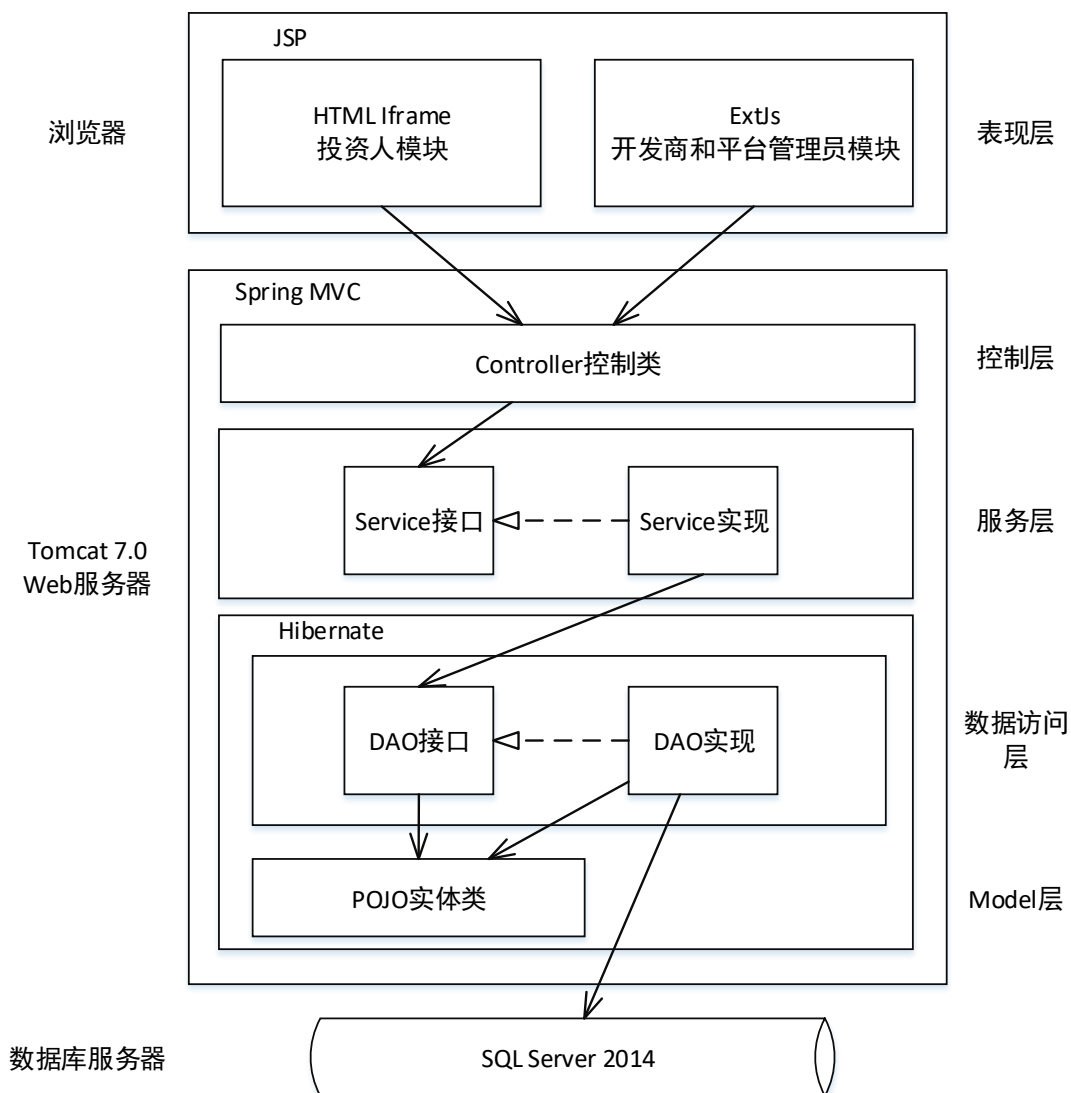


图 3.2-1 系统架构图

如图 3.2-1 所示，为\*\*\*的架构设计，\*\*\*分为五层：

- 1)表现层：整体上采用 JSP 页面进行显示，\*\*\*人模块采用 HTML 中的 Iframe 进行页面布局，平台管理员和\*\*\*模块在 jsp 页面中嵌入 Extjs 实现页面布局。
- 2)控制层：接收表现层传来的用户请求，调用服务层完成业务逻辑，控制页面跳转和视图显示。
- 3)服务层：调用数据访问层完成\*\*\*的业务逻辑。
- 4)数据访问层：利用 Hibernate 框架对 JDBC 数据库访问操作的进行封装。
- 5)Model 层：利用 Hibernate 框架的 ORM 技术产生数据库二维表对应的实体类。

### 3.3 系统实体类

\*\*\*实体类  
(1)。

图 3.3-1 实体类图

### 3.4 工程结构

如图 3.4-1 所示为\*\*\*的工程结构，根据具体的职能能对工程文件进行分类存放，相应的类都放到相应的包中，包与包之间的关系如图 3.4-2 所示。

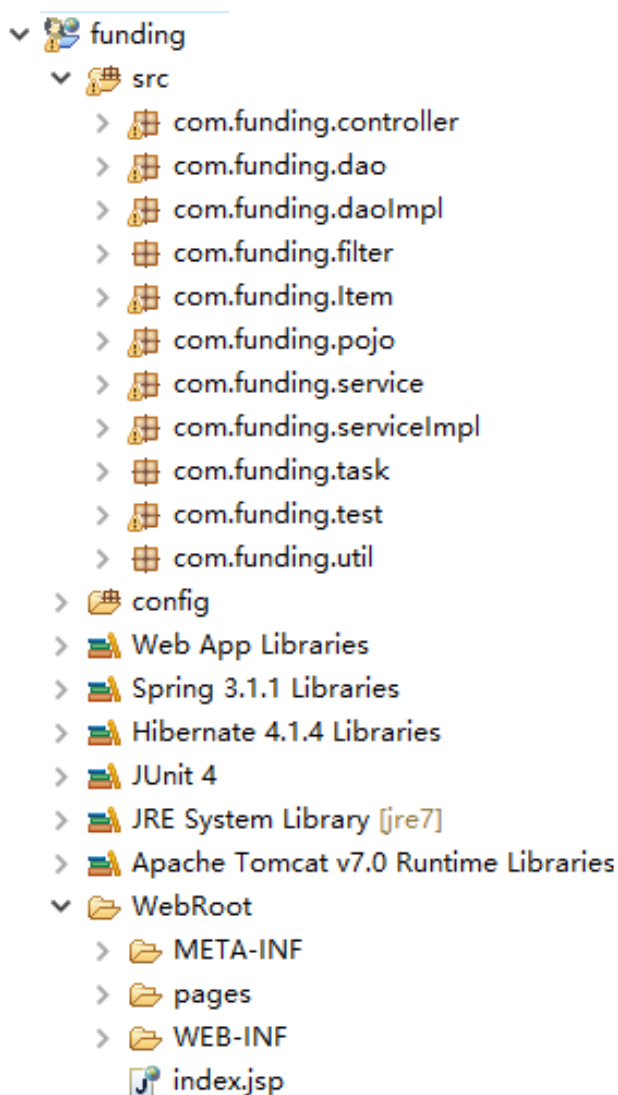


图 3.4-1 工程结构图

包结构的具体说明如下：

- (1)controller: 该包中的类为控制类，负责接收用户请求，调用服务层，完成业务逻辑，并控制页面跳转和视图显示。
- (2)dao: 该包中为数据访问层接口。
- (3)daoImpl: 该包中为数据访问层实现，利用 Hibernate 框架对 JDBC 数据库操作，进行封装处理。
- (4)filter: 该包中为自定义的过滤器，对指定的 url 进行过滤处理，以满足某种需求。
- (5)item: 该包中的类为列表项，用来配合页面列表显示。
- (6)pojo: 该包中的类为实体类，是 Hibernate 框架 ORM 技术对数据库二维表的映射。
- (7)service: 该包中为服务层类的接口。
- (8)serviceImpl: 该包中为服务层类的实现，负责调用数据访问层来完成业务逻辑。
- (9)task: 该包中为定义的定时任务类，定义了定时任务要做的业务处理。
- (10)test: 该包中为 JUnit 测试工具类，来辅助完成测试工作。
- (11)util: 该包中为工具类，提供日期时间转化，数据类型格式化等功能。

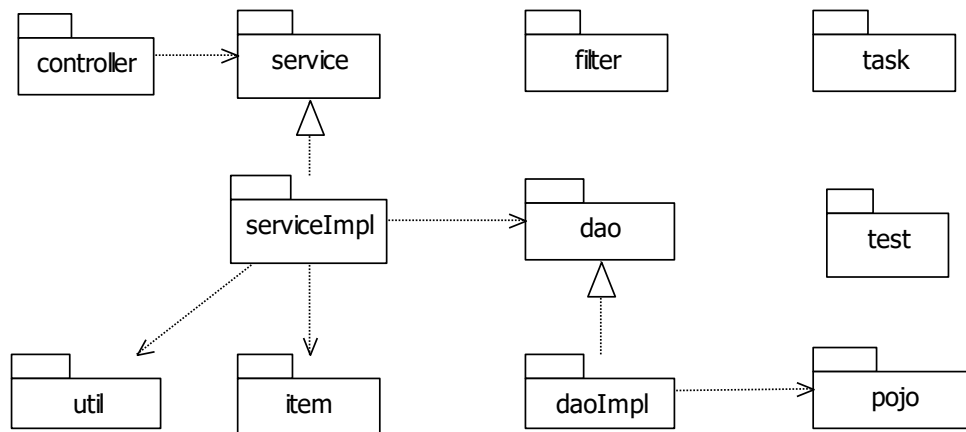


图 3.4-2 包图

## 3.5 用户界面框架

### 3.5.1 \*\*\*界面布局

\*\*人界面布局的框架如图 3.5-1 所示，该部分用 JSP 页面中嵌入 Iframe 实现界面切换，访问不同的链接，只有页面主体内容发生变化，其中用户状

态来显示用户登录信息，导航栏用来显示页面所属位置并提供导航，搜索框让用户随时都可以输入关键字对项目进行搜索。

LOGO	搜索框	用户状态
导航栏		
页面主体内容 (Iframe)		
友情链接 网站备案		

图 3.5-1 \*\*\*人界面布局

### 3.5.2 平台管理员和\*\*\*界面布局

模块根据所选的导航菜单不同，显示不同的内容。

LOGO	用户状态
树形 导 航 菜 单	子模块

图 3.5-2 平台管理员和\*\*\*界面布局

## 3.6 本章小结

本章在需求分析的基础上从数据库实体关系，网络拓扑结构，系统框架，系统实体类，工程结构和用户界面这些方面对\*\*\*进行了概要设计，理清了系统整体脉络，为下一章的详细设计做好了准备。

## 第4章 系统详细设计

本章的主要工作是在概要设计的基础上进行详细设计，首先对数据库进行详细设计，接着从类图，关键流程设计两个方面对\*\*\*分模块进行了详细设计。

### 4.1 数据库详细设计

实体关系图中的实体和实体之间的关系进行具体化就得到了数据库的详细设计，具体的数据字典如表 4.1-1 到错误!未找到引用源。所示。

表 4.1-1 Users(用户)表

字段名	含义	类型	长度	是否为空	备注
user_id	用户 id	int		否	主键
username	用户名	varchar	50	否	数字, 字母, 下划线组成, 首字母不能为数字
password	密码	varchar	50	否	MD5 加密密文, 明文密码不安全
sex	性别	int		是	0—未知 1—男 2—女
phone	手机号	varchar	11	否	手机号注册
portrait	头像	varchar	255	是	默认 default.png, 也支持 jpg, gif, bmp 等常用图片格式。
address	地址	varchar	255	是	省市两级
email	电子邮箱	varchar	255	是	标准电子邮箱格式
realname	真实姓名	varchar	50	是	
id_card	身份证号	varchar	18	是	
role	角色	int		否	0—平台管理员 1—***人 2—***
nickname	昵称	varchar	100	是	
motto	座右铭	varchar	255	是	
is_identified	是否认证	int		否	0—未认证 1—已认证

## 4.2 平台管理员模块详细设计

### 4.2.1 类图及描述

图 4.2-1 平台管理员模块类图

### 4.2.2 关键流程设计

图 4.2-2 修改\*\*\*时序图

图 4.2-3 新项目审核时序图

## 4.3 \*\*\*模块详细设计

### 4.3.1 类图及描述

\*\*\*模块的主要功能有类提供了数据库的增删改查操作。

图 4.3-1 \*\*\*模块类图

### 4.3.2 关键流程设计

处理流程如图 4.3-2 所示，

图 4.3-2 发起项目时序图



图 4.3-3 申请资金时序图

## 4.4 \*\*\*人模块详细设计

### 4.4.1 类图及描述

\*\*\*人模块是平台最重要的模块，也是

图 4.4-1 \*\*\*人模块类图

### 4.4.2 关键流程设计

注册流程如图 4.4-2 所示，\*\*\*允许\*\*\*人采用手机号进行注册。

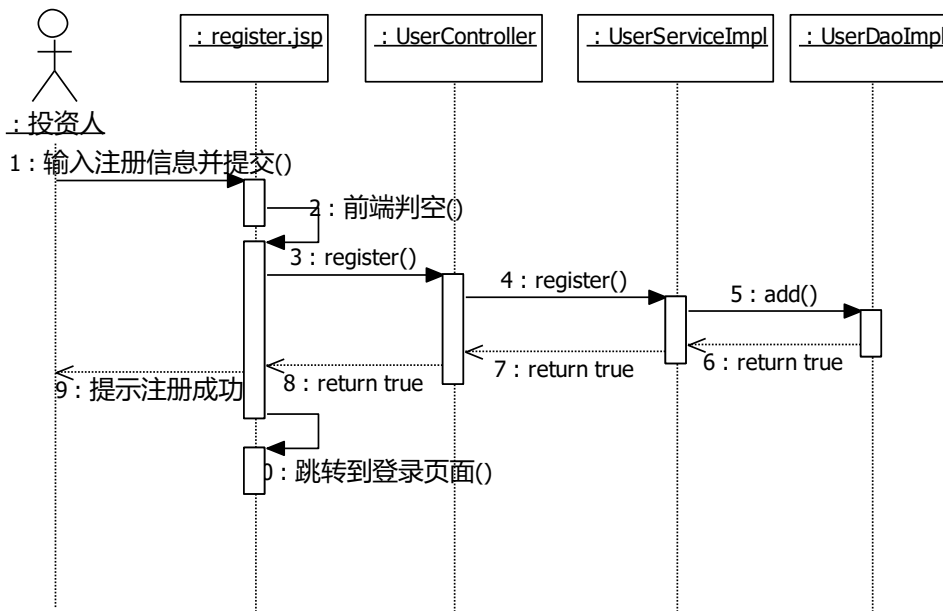


图 4.4-2 注册时序图

登录流程如错误!未找到引用源。所示，

图 4.4-3 认购项目时序图

## 4.5 平台定时模块详细设计

### 4.5.1 类图及描述

平台定时模块主要完成平台所需的定时任务，包的具体实现。

图 4.5-1 平台定时模块类图

### 4.5.2 关键流程设计

图 4.5-2 订单逾期检查时序图

图 4.5-3 项目检查时序图

## 4.6 本章小结

本章对\*\*\*进行了详细设计，数据库方面由实体关系图导出了数据库所有表，并完成了数据字典的设计，另外分模块进行了模块内部的详细设计，从类图，类的描述和关键流程设计三个方面完成了设计工作。

## 第5章 系统实现

本章分模块对\*\*\*的功能实现情况进行了展示，同时也简单描述了操作的具体流程。

### 5.1 平台管理员模块实现

平台管理员是\*\*\*的三个角色中权限最高的角色，主要负责基本信息的维护和审核关键业务信息。

图 5.1-1 审核管理效果图

### 5.2 \*\*\*模块实现

\*\*\*模块实现效果如错误!未找到引用源。所示，

图 5.2-1 发起项目效果图

图 5.2-2 资金申请效果图

### 5.3 \*\*\*人模块实现

\*\*\*人首页和登录实现效果如错误!未找到引用源。所示，

图 5.3-1 项目详情效果图

另外，需要说明的是只有登录的\*\*\*人才能进入个人中心。

### 5.4 平台定时模块实现

#### 5.4.1 订单检查定时原理

#### 5.4.2 项目检查定时原理

项目检查定时任务有固定的触发时间，每天只需要执行一次即可，但是比较耗时，不会受到平台并发性的影响，但是要考虑计时过程中的效率，考虑使用 Quartz 框架。Quartz 是一个完全由 Java 编写的开源企业级作业调度框架，能并发运行多个作业，使用简单高效，Spring 框架对 Quartz 提供了很好的支持，只需定义好 Job 类，在 Spring 配置文件中引入相应的 Job 对象，指

定触发的时间的表达式，指定好调用的处理方法，Quartz 就可以成功嵌入 Spring Web 工程，随 Web 工程的启动而一同启动，开始计时，并完成定时任务。

## 5.5 本章小结

本章分模块展示了\*\*\*的实现效果，简单描述了主要模块页面之间的切换方式和操作的一些流程，另外，还说明了平台定时模块的实现原理。

## 第6章 系统测试

本章将完成\*\*\*的测试工作，主要采用 JUnit 工具进行单元测试，功能方面进行黑盒测试，主要业务流程进行白盒测试，考虑到论文篇幅的限制和测试工作的重复性，三种测试分别用一个例子来说明测试的方法和结果。

### 6.1 JUnit 测试

JUnit 是一个 Java 语言的单元级测试工具，MyEclipse 内置了对 JUnit4 的支持。测试使用 JUnit4 版本进行测试，JUnit4 引入了注解来执行编写的测试代码。

表 6.1-1\*\*\*1 中待测试方法

	方法声明	方法说明
	public boolean add(Project p)	添加 Project，Project 表 id 自增
	public boolean delete(int id)	删除 Project，id 应该为已有 id
	方法声明	方法说明
	public void testAdd()	测试 add()
	public void testDelete()	测试 delete()

JUnit 提供的一系列 assert 断言方法，可以对结果进行判断，看执行结果是否与期望值一致，若断言正确，则通过测试；若断言不正确，则显示为失败。针对类设计的测试用例如表 6.1-2 所示，测试结果如图 6.1-1 所示。

表 6.1-2 ProjectDaoImpl 类 JUnit 测试用例

		输入参数	期 输 出	
		p 不为 null	tru e	
		p 为 null	fal se	

		id 为 Project 表中已有 id	true	
--	--	----------------------	------	--

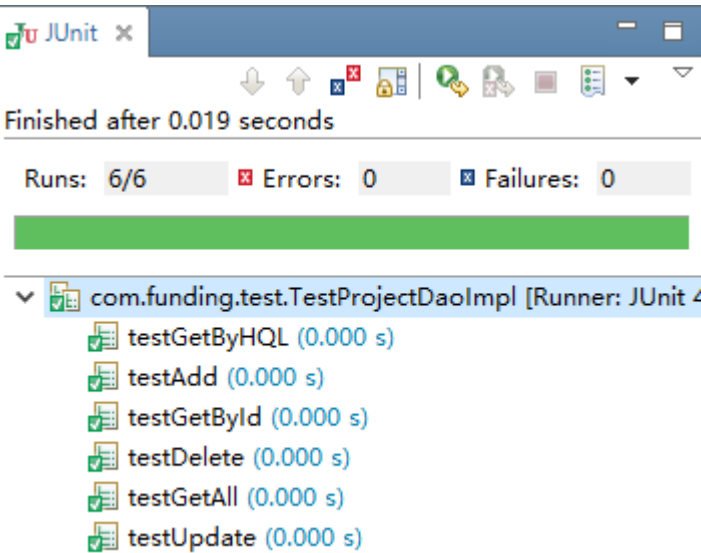


图 6.1-1 类 JUnit 测试结果图

## 6.2 白盒测试

白盒测试主要针对两个方面进行测试，一方面是流程图的各个判定框，另一方面是程序执行路径。考虑到测试的工作量，程序执行路径测试采用了JUnit 进行测试来提高效率，所以这里的白盒测试主要对\*\*\*的关键操作流程进行测试。

。

表 6.2-1 认购支付白盒测试测试用例

测试路径	覆盖节点	输入数据	预期输出	实际输出
abcdefgh	1,2,3,4, 5,6,7	*****	认购成功	认购成功

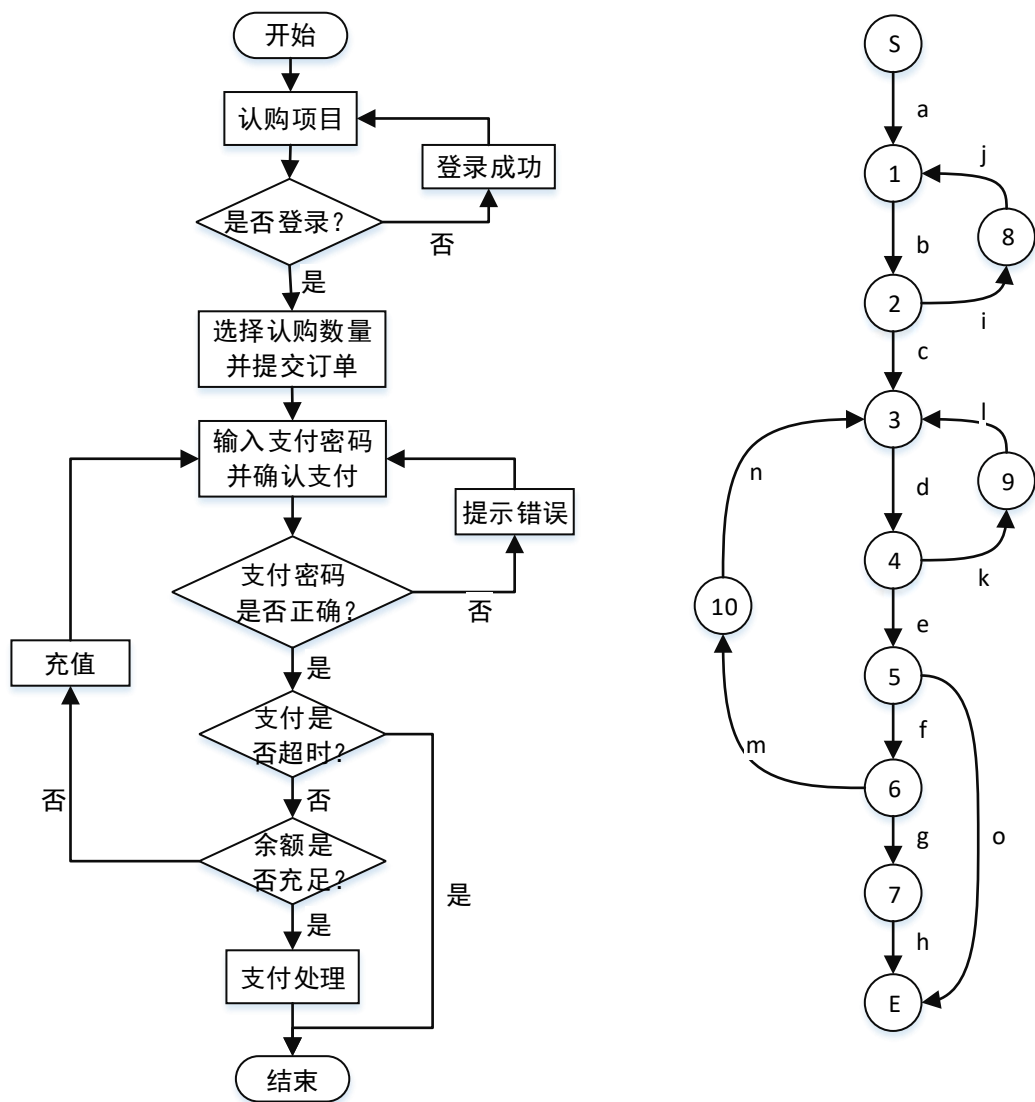


图 6.2-1 认购支付流程图和程序图

### 6.3 黑盒测试

黑盒测试是根据被测试程序功能来进行测试，这里采用划分等价类和边界值分析的方式设计黑盒测试的测试用例，来保证平台功能的准确性。

以\*\*\*人登录，注册和修改个人信息部分的关键输入信息为例来进行黑盒测试，首先要划分等价类，表 6.3-1 就是密码，手机号，真实姓名，身份证号，邮箱和头像等输入数据的有效等价类和无效等价类划分结果。

表 6.3-1 等价类划分结果

输入数据	有效等价类	无效等价类
------	-------	-------

手机号	(1)11 位数字	(2)空 (3)小于 11 位 (4)大于 11 位 (5)含非数字字符
密码	(6)1~20 位任意字符	(2)空 (7)大于 20 位字符

根据有效等价类和无效等价类的划分结果，结合边界值分析的方法，针对以上这些主要输入数据设计黑盒测试用例，并进行测试。黑盒测试用例和测试结果如下：

(1)手机号测试用例设计和实际测试结果如表 6.3-2 所示。

表 6.3-2 手机号黑盒测试用例

用例编号	输入数据	期望结果	测试范围	通过与否
1	13173307182	输入有效	(1)	√
2	空	输入无效	(2)	

(2)密码测试用例设计和实际测试结果如表 6.3-3 所示。

表 6.3-3 密码黑盒测试用例

用例编号	输入数据	期望结果	测试范围	通过与否
1	123456	输入有效	(6)	√
2	a	输入有效	(6)	√
3	12345678901234567890	输入有效	(6)	√
4	空	输入无效	(2)	
5	123456789012345678901	输入无效	(7)	

## 6.4 本章小结

本章从 JUnit 测试，白盒测试和黑盒测试三个方面完成了\*\*\*的测试工作，保证了平台功能的准确性和可靠性。



## 课设总结

本文紧紧围绕\*\*\*核心业务流程，从需求分析，概要设计，详细设计，实现和测试几个方面，对\*\*\*进行了设计与实现。\*\*\*是一个基于 Spring MVC 框架和 Hibernate 框架的 B/S 架构的网站，平台的用户分为平台管理员，\*\*\*和\*\*\*人三种角色，三种角色互相配合，完成了项目发起，项目审核，项目筹资，项目放资和项目退款等一系列\*\*\*业务中的关键环节，让\*\*\*业务能够有条不紊地在该平台上运转起来。

.....。