**目录**

**[1. 引言 1](#_Toc31653)**

[1.1 研究背景及意义 1](#_Toc11459)

[1.2 本课题主要工作 2](#_Toc22408)

**[2. 系统分析 2](#_Toc3985)**

[2.1 可行性分析 2](#_Toc9818)

[2.2 系统体系结构 3](#_Toc11649)

[2.3 功能需求分析 3](#_Toc23795)

[2.4 业务流程分析 3](#_Toc18841)

[2.5 系统数据E-R关系图简介 5](#_Toc30117)

**[3. 系统总体设计 6](#_Toc12989)**

[3.1 软件设计原则 6](#_Toc12850)

[3.2 系统整体的设计要求 6](#_Toc24653)

[3.3 系统模块设计 7](#_Toc7802)

[3.4 数据架构设计 8](#_Toc24740)

[3.5 用户界面设计 17](#_Toc24964)

[3.6 物理配置方案设计 17](#_Toc4217)

**[4. 系统实现 17](#_Toc6089)**

[4.1 技术选型与开发设计 18](#_Toc2684)

[4.2 开发环境介绍 18](#_Toc2288)

[4.3 模块详细实现 19](#_Toc30879)

**[5. 系统测试与维护 38](#_Toc23646)**

[5.1 系统测试 38](#_Toc31983)

[5.2 系统维护 40](#_Toc19265)

**[6. 结束语 41](#_Toc21993)**

**[【参考文献】 41](#_Toc1030)**

**[致谢 42](#_Toc17062)**

基于Web的个人信息融合系统的设计与实现

计算机与软件学院计算机科学与技术专业 冯德兵

准考证号：030516100877

【摘要】 个人信息管理是将个人信息用信息化技术进行管理，以帮助个人提升[工作效率](https://wiki.mbalib.com/wiki/%E5%B7%A5%E4%BD%9C%E6%95%88%E7%8E%87" \o "工作效率)，融合个人所有的[信息资源](https://wiki.mbalib.com/wiki/%E4%BF%A1%E6%81%AF%E8%B5%84%E6%BA%90" \o "信息资源)，提高个人的竞争力。它是一种新的管理理念和方法，能将个人拥有的各种资料、随手可得的信息变成更有[价值](https://wiki.mbalib.com/wiki/%E4%BB%B7%E5%80%BC" \o "价值)的知识，最终有利于自己的工作和生活，通过个人信息融合，人们可以养成良好的[学习习惯](https://wiki.mbalib.com/wiki/%E5%AD%A6%E4%B9%A0%E4%B9%A0%E6%83%AF" \o "学习习惯)，增强[信息素养](https://wiki.mbalib.com/wiki/%E4%BF%A1%E6%81%AF%E7%B4%A0%E5%85%BB" \o "信息素养)，完善自己的专业知识体系，提高自己的能力和竞争力，为实现个人价值和[可持续发展](https://wiki.mbalib.com/wiki/%E5%8F%AF%E6%8C%81%E7%BB%AD%E5%8F%91%E5%B1%95" \o "可持续发展)打下坚实的基础。本文主要阐述了利用Lavaral技术在Web上针对个人信息融合系统的设计与实现，本系统融合个人或家庭所有的信息（包括家庭与生活、工作与学习以及个人或者家庭健康信息等）到一个系统里面，只要使用个人信息融合系统就能实现查询、管理与分析等功能。使用本系统能够在信息时代高效率地利用个人信息来提高生活、工作和学习质量与效率。

**【关键词】** 个人信息管理，个人信息融合，Web，Laravel

1. 引言
   1. **研究背景及意义**

随着新时代通信IT技术的快速发展，围绕个人相关的应用数据和信息呈现爆炸性的增长，面对如此多的个人信息数据，如何高效处理和利用数据，如何迅速寻找到需要用到的信息，如何整合分散在各处的信息，所有这些问题都直接影响到我们的工作学习和生活质量。

个人信息管理是将个人信息用信息化技术进行管理，以帮助个人提升工作效率，整合个人或家庭的信息数据，提高个人的竞争力。它是一种创新的管理思路和技巧，能将个人或家庭拥有的各种资料、容易获取的信息变成更有价值的数据，以使其更加有效率地服务于个人的工作和生活，通过个人信息管理，人们可以培养更好的学习习惯，增强信息素养，完善自己的专业信息与技能体系，提高自己的能力和竞争力，为实现个人价值和可持续发展打下坚实的基础。

随着移动互联网和5G等现代信息技术的快速发展把人类带入了海量应用时代，在我们生活、工作和学习的过程中产生各种与个人息息相关的数据，在面对如此多的各个方面的个人信息数据时，如何科学高效地管理个人信息和将这些来自各个平台与应用的个人信息进行科学地有机地整合在一起，来提升我们的生活质量和学习工作的效率。正是撰写此文的意义所在。

国内外发展状况：

1. 发展比较迅速和宽泛，主要集中个人信息管理系统，缺乏针对各个方面的深入研究与应用。

目前个人信息管理（PIM）在国内外发展的比较迅速和宽泛，研究热点主要涵盖了ＰＩＭ的基本概念、信息整合和增强、信息检索、信息安全、语义桌面、个人信息空间、个人信息管理工具、个人信息管理系统、移动环境下的个人信息管理等 ９个主要方面。市场上存在一些有特色的个人信息管理软件。虽然近年来针对个人信息管理的全球研讨会在不断地进入讨论和创新，但是针对各个子领域的应用研究比较缺乏。因此深入研究个人信息管理的各子领域研究的大有可为。

二、近年来个人信息管理呈现出结合芯片与移动设备发展的趋势。

随着信息技术特别移动通信技术和云计算的发展，以智能手机、可穿戴设备以及物联网设备等为主的通信设备的崛起，这将对我们的生活和学习带来巨大的影响。现在人们越来越依赖各种移动与嵌入式设备，如何在移动环境下实现个人信息的管理，并能够同其他系统实现数据同步与共享已经成了未来发展的一个新趋势。

* 1. **本课题主要工作**

本系统主要实现了常用的个人信息包括财务收支、个人知识库、目标与任务、日程提醒、通讯信息、邮件数据、评论信息和账号信息等的数据管理与展示。并且支持自定义模块对数据进行动态化管理，实现了根据模块和关键字来进行搜索，实现根据近期数据进行分析给出建议。基本满足个人或家庭对信息管理的需求，后续逐步将实现支持不同平台。

1. 系统分析

2.1 可行性分析

可行性分析是为了明确要做的系统的需求与目的，对系统开发需求背景、必要性和意义进行详细的调研分析，并根据需求提出拟开发系统的初始方案与计划。可行性分析从必要性和可行性方面入手，为系统开发项目的决策提供必要的依据。下面进行分析：

1. 必要性： 随着信息和应用的爆炸性增长，市场上软件系统已经难以满足个人需求，因此分析整理用户的痛点，通过改进和融合各个功能需求来逐步实现个人日益增长的信息管理需求。
2. 可行性通过以下步骤来分析：
3. 分析需求的出发点和目标

本系统旨在通过完善和整合个人信息管理的痛点需求，来实现提高个人或家庭的工作、生活和学习质量与效率的目标。

1. 确定用户常用的信息需求

在分析设计本系统的时候，花了大量的精力去分析当代人的个人信息管理的痛点需求，在收集和征求许多同事朋友的需求与想法、查询网上个人信息管理软件功能以及参考图书馆查询大量的资料，分析了对比各个相关的系统功能，完善和整合痛点需求来实现系统目标。

1. 草拟系统的初始方案

在经过上述步骤后基础上，画出顶层的需求功能流程图以及标识各个功能的之间的关系。撰写出初始方案。要一台基础的基础配置服务器即可，低成本快速提高个人信息使用和整合效率。

因此，设计实现个人信息融合系统的想法是可行的。

2.2 系统体系结构

本系统主要采用浏览器/服务器结构，本系统有游客和管理员两个角色，管理员主要通过数据库中的用户表对用户进行管理，防止非法入侵系统等，然后就是对个人信息的数据库进行管理与维护和使用待办事项、收支管理、邮件管理以及其他自定义管理，个人空间模块即展示公开的个人信息数据。结构如下：

游客

http

MySQLDB

服务器

数据库服务器

管理员

http

2.3 功能需求分析

经过上一步骤的分析后，现在需要对系统的具体需求做一下简单地分析。需求分析本质是对系统的功能需求整合的详细描述，用统一的规范的格式化的术语来表述出来。可以从系统需求分析、目标和系统范围界定和功能模块分析以及划分等方面来分析。

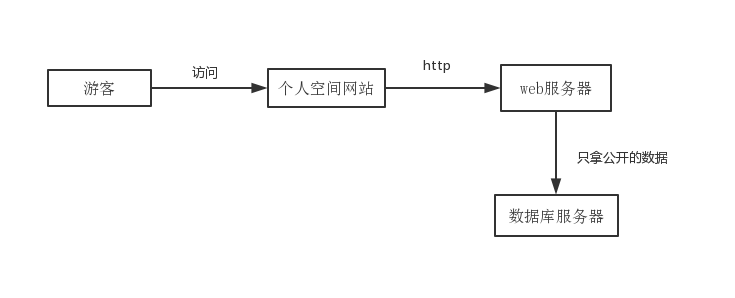
用上述三个步骤分析得到本阶段的成果，个人信息管理系统处理应满足以下需求：

1. 满足个人信息管理痛点需求包括工作、生活和学习。
2. 查询数据灵活方便，支持按照模块、关键字搜索。
3. 管理数据操作容易灵活，支持自定义模块管理。
4. 支持个人知识库管理。
5. 支持根据近期数据使用习惯来智能给出建议。
6. 超前美观的视觉界面设计与实现。
7. 界面交互操作简单。
8. 支持动态切换界面皮肤。

2.4 业务流程分析

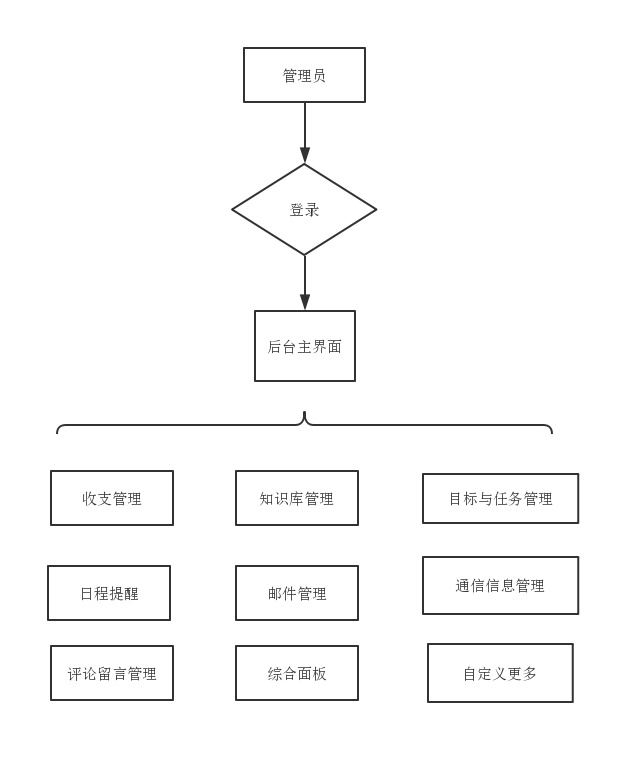
经过较为详实地需求分析，现来对业务做一个基本的梳理。把系统分为公开信息展示内容和后台个人信息数据管理系统；把公开的这部分内容称之为个人空间，公开展示内容不需要登录账号，称之为访客，主要展示个人的分享的知识学习、经验总结和生活场景等内容，把它称之为访客。后台个人管理系统需要账号登录，可以管理系统内所有的数据内容包括个人空间数据、个人隐私数据和数据分析内容以及获取其他平台的个人数据与第三方信息。

公开展示的个人空间，游客可以浏览一些关于个人的知识学习、生活旅行、经验总结和评论留言以及其他信息。如下毕设图1



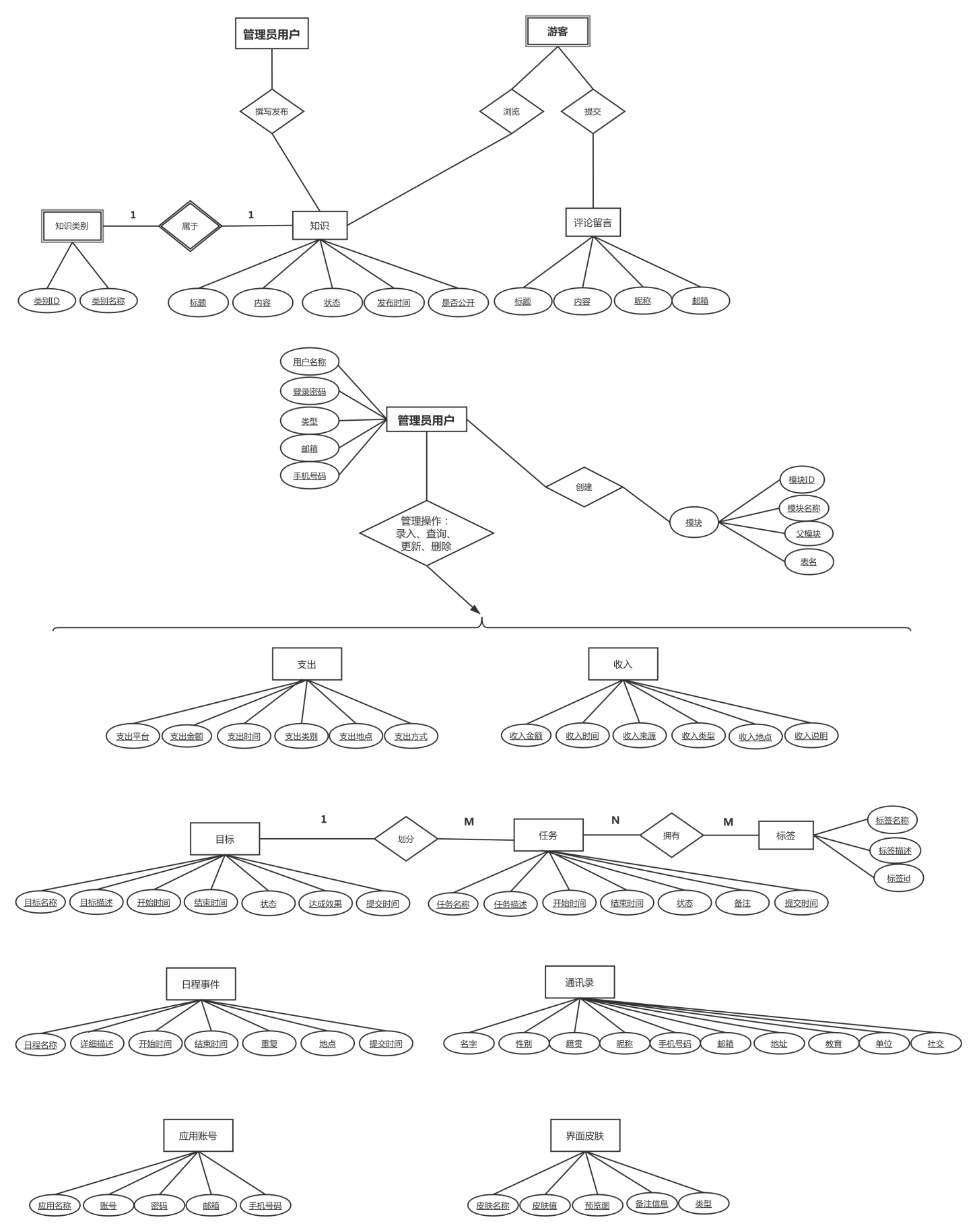
毕设图 1

在后台个人管理系统，管理员在该系统中可以对个人空间数据、个人隐私数据和数据分析内容以及获取其他平台的个人数据与第三方信息进行管理和分析甚至还可以管理个人空间的界面风格。如下毕设图2



毕设图 2

2.5 系统数据E-R关系图简介



毕设图 3

1. 系统总体设计

3.1 软件设计原则

系统设计是一个逐步迭代的过程，一般是先顶层后局部；然后再对局部进行细化；期间也会伴随·着一些数据设计和接口设计。为了开发出性价比高的软件，软件设计一般遵循以下几个原则：

1. 抽象

抽象指的是提取物体的最原始最本质的属性和行为，忽略其他干扰因素。采用分模块抽象办法方便控制系统开发的复杂程度，有利于程序的可读性和可维护性。

1. 封装

封装指的隐藏实现隐藏起来，对于不需要的信息不提供访问，以便是程序方法尽可能地简单和可复用。

1. 模块化

模块化指的是将逻辑上相对独立的单元单独分开来实现，模块内尽量实现高内聚低耦合。

1. 一致性

一致性指的将整个软件系统上的各个模块都要使用一致的概念、符合和术语；程序内部接口尽量保持一致。

3.2 系统整体的设计要求

系统设计为个人或者家庭用户使用，以个人或者家庭信息数据为基础，系统设计时需要考虑各个需求的特性，需要采用多模块架构来实现。根据系统分析的结果，全方位地明确了个人信息融合系统的最终需求，现根据需求进行系统的设计。总的来说，围绕高可扩展性、易维护性和交互使用简洁和避免数据冗余以及方便分阶段实施的原则来设计。

* 后台管理系统，承载诸多功能数据的管理与分析功能，而且还要支持高度自定义数据管理，良好的扩展性，易于操作显得非常重要。为了方便个人获取核心日常信息，后台主页面板基本要把常用信息管理放在显眼位置；然后将不同信息分类分模块进行管理，每个模块互相独立进行相应地增删改查操作。为了综合利用各类数据以及使用习惯，使用自定义的算法分析经常定期出现的事件，进而给用户提出功能使用建议。为了方便使用某项数据信息或者功能，在首页设计搜索，支持按模块或者关键字搜索相应的信息或者功能高亮显示。
* 系统采用B/S模式实现，系统采用了PHP的Laravel技术以及开放性的RESTFUL风格接口技术，以便适应各系统开发的需求。
* 在追求功能丰富、使用简介、易于扩展的同时，使用参考国外优秀的视觉设计，来美化界面，并使用界面交互大气又不失简洁，同时具有响应式兼容各种PC和移动设备。
* 个人空间展示区域，设计展示聚合优秀的个人内容、漂亮简洁的界面和优秀的响应式加载能力。
* 系统采用前后端分离的方式开发，支持自定义服务器接口，Web前端单独部署。
* 为了数据的安全性着想，应对数据进行加密传输，支持HTTPS，不公开的数据在没有登录的情况下不允许访问。

3.3 系统模块设计

根据总体设计的成果，把各个功能需求设计成单独的模块，现在针对各个模块进行详细的陈述。如下毕设图4：

****

毕设图 4

**（1）后台管理：**

后台管理的功能就分为各个功能的数据管理和综合信息查询分析，下面根据系统模块设计来详细地说 明各个模块的内容。

* **管理员登录验证：**

登录后台需要管理员账号才能登陆进行数据管理。该模块的功能是对管理员身份进行登录校验。输 入登录admin账号和密码后，系统将判断账号和密码是否正确匹配，如果通过验证则状态则跳转后 台主页，反之则提示错误。

* **综合面板模块：**

为后台综合信息展示模块，管理员登录后首先跳转至该模块，该模块会抓取管理员感兴趣的新闻博 客文章，个人最近一个月的收支账单，个人待办事项，快捷信息查询功能，模块快捷导航功能以及 后台和个人空间的界面管理，为了丰富功能，后期还会迭代增加数据分析与建议功能，即根据个人 近一段时间的使用，给出一些信息建议，例如账单情况，近期哪些类型的消费较多，给予提示；如 重要邮件提醒，紧急事项他。

* **收支管理模块：**

该模块记录个人账户的所有的收入和支出情况。收入可以包括工资收入、股票分红收入、租金收入 以及社会福利收入等，此功能针对收入进行按照收入类型、数量时间等进行管理。支出管理功能主 要用来管理个人的各种支出，包括生活日常支出、交通旅行支出、人情支出和教育学习等，对这些 支出信息进行按类型、时间、地点和金额进行管理。

* **知识模块：**

该模块主要用来记录个人学习、经验总结、优秀文章和电子文档以及音视频知识。用来对个人知识 进行高效率的分门别类整理，以期能够帮助提升个人知识高效利用。此外，还可以把一些可以优秀 的文章与人分享，标记成公开状态，用于个人空间的展示。

* **目标与任务管理模块：**

该模块用来管理个人的目标与任务。设立此模块的目的并不仅仅是管理每日待做的事项，更重要的 是用来管理个人的中长期目标，把中长期目标拆成一个个，把中长期目标拆成一个个明确的、具体的、可量化、有截止时间并且可以实现的任务，以此来跟进与实现个人的中长期目标。

* **日程管理模块：**

该模块用来管理将每天的工作和事务安排在日程表中，并做一个有效的记录，方便管理日常的工作和事务，达到工作备忘的目的。同时也具有对日常任务进行自我提醒的作用。

* **通用信息模块：**

通用信息模块主要用来管理通讯录和地址信息以及其他通用信息管理。

* **邮件管理模块：**

该模块用来对日常学习或工作的收发邮件进行管理，支持跨设备跨平台同步信息。

* **评论留言管理模块：**

该模块用来管理个人空间的评论留言，过滤掉各种广告、非法违规的留言评论。

* **账号管理模块：**

该模块用来管理个人和亲属的各种互联网应用的账号与密码。

* **自定义模块管理模块：**

该模块用来扩展一些个人自定义的信息管理，可以自主设计一些需要管理的项，来进行各种管理与应用操作。例如用来管理不动产、债务投资信息和需要帮助亲属管理一些信息。

**（2）个人空间**

个人空间用来分享一些个人的学习上的收获、技术上的总结、产品上的思考，以及对社会生活、科技发展等方面的一些看法。

3.4 数据架构设计

1. **数据库设计**

数据库是本系统最为核心的部分。数据库的设计得是否合理、是否方便扩展、是否易于维护和维护成本高低是关系着系统的是否可用和性能好坏的重要因素。因此本系统的数据库设计参照理论标准的遵循的原则来设计，主要步骤如下：

* 根据需求设计

保证数据符合用户需求，保证数据处理功能符合用户的需求，保证数据的安全性和完整性。

* 根据需求的概念来设计字段结构

对于用户的需求进行梳理、规整与抽象以符合数据库系统中的概念模型。

* 逻辑结构设计

逻辑结构设计是在概念设计上的基础上，将概念结构E-R图转换为某个数据库管理系统所支持的数据模型，并对其进行优化。

* 物理结构设计

给数据选择合理的计算机存储的物理结构，以保证数据存储时间和空间以及维护的最优化。

* 数据库的实施与维护

根据上述的原则建立数据库结构，并写入数据进行测试和运行的过程。

1. **数据表设计**

数据表的设计是数据库设计的具体内容。根据模块设计阶段对需求的分析，可以得到以下本系统所用到的信息包括用户表、收入与支出表、知识表、目标与任务表、日程表、通讯表、邮件表、留言表、账号管理表、模块菜单表。以下就是详细的表结构设计：

**库的名称：db\_person\_union**

1. **用户表：tb\_user**

表 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| u\_id | int | 4 | 无 | 否 | 是 | 用户ID |
| username | varchar | 50 | 无 | 否 | 否 | 用户名称 |
| password | char | 16 | 无 | 否 | 否 | 登录密码 |
| type | int | 1 | 无 | 否 | 否 | 类型：1超管 2 普通 |
| email | varchar | 50 | 无 | 否 | 否 | 邮箱 |
| phone | int | 13 | 无 | 否 | 否 | 手机号码 |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

1. **收支表**
2. 收入表：tb\_income

表 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| id | int | 4 | 无 | 否 | 是 | 唯一ID |
| amount | varchar | 50 | 无 | 否 | 否 | 收入金额 |
| source | varchar | 50 | 无 | 否 | 否 | 收入来源 |
| summary | varchar | 200 | 无 | 否 | 否 | 收入说明 |
| category | int | 2 | 1 | 否 | 否 | 收入类型 |
| place | varchar | 50 | 无 | 否 | 否 | 收入地点 |
| income\_time | int | 13 | 无 | 否 | 否 | 收入时间 |
| remark | varchar | 200 | 无 | 否 | 否 | 备注 |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

1. 支出表：tb\_expend

表 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| id | int | 4 | 无 | 否 | 是 | 唯一ID |
| amount | varchar | 50 | 0 | 否 | 否 | 支出金额 |
| category | int | 2 | 1 | 否 | 否 | 支出类型 |
| description | varchar | 200 | 无 | 否 | 否 | 支出说明 |
| trade\_location | varchar | 50 | 无 | 否 | 否 | 支出地点 |
| platform | int | 2 | 1 | 否 | 否 | 1天猫  2京东  3 亚马逊  4、拼多多  5、实体  6、其他 |
| channel | tinyint | 2 | 1 | 否 |  | 1. 支付宝 2. 微信 3. 银行卡 4. 现金 5. 其他 |
| expend\_time | int | 13 | 无 | 否 | 否 | 支出时间 |
| remark | varchar | 200 |  | 否 | 否 | 备注 |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

1. **知识表**
2. 知识表：tb\_knowledge

表 4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| id | int | 4 | 无 | 否 | 是 | 唯一ID |
| title | varchar | 200 | 无 | 否 | 否 | 标题 |
| html\_content | longText |  | 无 | 否 | 否 | 转义内容 |
| origin\_content | longText |  | 无 | 否 | 否 | 原始内容 |
| status | int | 1 | 1 | 否 | 否 | 1.草稿  2.已发表  3.推荐  4.删除 |
| t\_id | int | 2 | 1 | 否 | 否 | 类别ID |
| origin | varchar | 200 | 无 | 否 | 否 | 来源 |
| is\_share | int | 1 | 1 | 否 | 否 | 是否公开共享  1 否  2 是 |
| published\_at | int | 13 | 无 | 否 | 否 | 发布时间 |
| remark | varchar | 200 |  | 否 | 否 | 备注 |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

1. 知识类别：tb\_knowledge\_type

表 5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| id | int | 13 | 无 | 否 | 是 | 唯一ID |
| name | varchar | 50 | 无 | 否 | 否 | 类别名称 |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

1. **目标与任务表**
2. 目标表：tb\_goals

表 6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| id | int | 13 | 无 | 否 | 是 | 唯一ID |
| name | varchar | 250 | 无 | 否 | 否 | 目标名称 |
| description | text |  | 无 | 否 | 否 | 目标描述 |
| start\_time | int | 14 | 无 | 否 | 否 | 开始时间 |
| end\_time | int | 14 | 无 | 否 | 否 | 结束时间 |
| status | int | 1 | 1 | 否 | 否 | 目标完成状态  1 未完成  2 已完成  3 挂起  4 废弃  5 其他 |
| effect | varchar | 300 | 1 | 否 | 否 | 目标实现的效果描述 |
| published\_at | int | 13 | 无 | 否 | 否 | 提交时间 |
| remark | varchar | 200 | 无 | 否 | 否 | 备注 |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

B.任务表：Tb\_tasks

表 7

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| id | int | 13 | 无 | 否 | 是 | 唯一ID |
| g\_id | int | 13 | 无 | 是 | 否 | 关联目标Id |
| task\_name | varchar | 250 | 无 | 否 | 否 | 任务名称 |
| task\_desc | text |  | 无 | 否 | 否 | 任务描述 |
| start\_time | int | 14 | 无 | 否 | 否 | 开始时间 |
| end\_time | int | 14 | 无 | 否 | 否 | 结束时间 |
| status | int | 1 | 1 | 否 | 否 | 任务完成状态  1 未完成  2 已完成  3 挂起  4 废弃  5 其他 |
| label\_id | int | 13 | 无 | 否 | 否 | 关联任务标签 |
| published\_at | int | 13 | 无 | 否 | 否 | 提交时间 |
| remark | varchar | 200 | 无 | 否 | 否 | 备注 |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

1. 标签表设计tb\_label

表 8

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| id | int | 13 | 无 | 否 | 是 | 唯一ID |
| name | varchar | 50 | 无 | 否 | 否 | 标签名字 |
| desc | varchar | 100 | 无 | 是 | 否 | 标签补充描述 |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

1. 任务与标签关联表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| t\_id | int | 13 | 无 | 否 | 是 | 任务ID |
| l\_id | int | 13 | 无 | 否 | 是 | 标签ID |

表 9

1. **日程表：tb\_schedule**

表10

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| id | int | 13 | 无 | 否 | 是 | 唯一ID |
| name | varchar | 50 | 无 | 否 | 否 | 日程名称 |
| desc | varchar | 200 | 无 | 是 | 否 | 日程描述 |
| start\_time | int | 13 | 无 | 否 | 否 | 日程开始时间 |
| end\_time | int | 13 | 无 | 否 | 否 | 日程结束时间 |
| repeat | int | 1 | 0 | 否 | 否 | 是否重复  0 不重复  1 每天重复  2 每周重复  3 每月重复  4 每季度重复  5 每年重复 |
| place | varchar | 50 | 无 | 否 | 否 | 日程进行的地点 |
| remark | varchar | 100 | 无 | 否 | 否 | 备注信息 |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

1. **通讯录表：tb\_address\_book**

表 10

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| id | int | 13 | 无 | 否 | 是 | 唯一ID |
| name | varchar | 50 | 无 | 否 | 否 | 名字 |
| sex | int | 1 | 0 | 否 | 否 | 性别  0 男  1 女 |
| birth\_date | int | 13 | 无 | 是 | 否 | 生日 |
| birthplace | varchar | 80 | 无 | 否 | 否 | 籍贯 |
| nickname | int | 1 | 0 | 否 | 否 | 昵称 |
| avatar | varchar | 100 | 无 | 是 | 否 | 头像 |
| email | varchar | 50 | 无 | 是 | 否 | 邮箱 |
| phone | varchar | 100 | 无 | 否 | 否 | 备注信息 |
| address | varchar | 200 | 无 | 是 | 否 | 通信地址 |
| education | varchar | 50 | 无 | 是 | 否 | 毕业院校或者学历 |
| employer | varchar | 50 | 无 | 是 | 否 | 工作单位 |
| wechat | varchar | 30 | 无 | 是 | 否 | 微信号 |
| weibo | varchar | 30 | 无 | 是 | 否 | 微博 |
| qq | int | 13 | 无 | 是 | 否 | QQ |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

1. **留言表：tb\_comments**

表 11

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| id | int | 13 | 无 | 否 | 是 | 唯一ID |
| content | varchar | 400 | 无 | 否 | 否 | 留言内容 |
| nickname | varchar | 50 | 无 | 否 | 否 | 浏览昵称 |
| email | varchar | 40 | 无 | 否 | 否 | 邮箱 |
| a\_id | int | 13 | 无 | 否 | 否 | 关联知识Id |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

1. **账号表：tb\_apps**

表 12

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| id | int | 13 | 无 | 否 | 是 | 唯一ID |
| appname | varchar | 400 | 无 | 否 | 否 | 应用名称 |
| username | varchar | 50 | 无 | 否 | 否 | 登录账号 |
| password | varchar | 40 | 无 | 否 | 否 | 登录密码 |
| email | varchar | 40 | 无 | 是 | 否 | 电子邮箱 |
| phone | int | 13 | 无 | 是 | 否 | 手机号码 |
| remark | varchar | 200 | 无 | 是 | 否 | 备注信息 |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

1. **邮件表：tb\_mails**

表 13

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| id | int | 13 | 无 | 否 | 是 | 唯一ID |
| mail\_name | varchar | 40 | 无 | 否 | 否 | 邮箱名称 |
| password | char | 32 | 无 | 否 | 否 | 邮箱密码 |
| title | varchar | 50 | 无 | 否 | 否 | 邮件标题 |
| summary | varchar | 40 | 无 | 否 | 否 | 邮件概要 |
| content | varchar | 40 | 无 | 是 | 否 | 邮件内容 |
| receive\_time | int | 13 | 无 | 是 | 否 | 收件时间 |
| sender | varchar | 200 | 无 | 是 | 否 | 发件人名称 |
| send\_address | varcahr | 80 | 无 | 否 | 否 | 发件地址 |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

1. **模块管理表：tb\_modules**

表 14

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| id | int | 13 | 无 | 否 | 是 | 唯一ID |
| module\_name | varchar | 40 | 无 | 否 | 否 | 模块名称 |
| order | int | 1 | 无 | 否 | 否 | 模块排序 |
| f\_id | int | 13 | 0 | 否 | 否 | 父模块 |
| remark | varcahr | 80 | 无 | 否 | 否 | 备注信息 |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

1. **界面皮肤表：tb\_skins**

表 15

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **长度** | **默认值** | **是否为空** | **是否为主键** | **字段描述** |
| id | int | 13 | 无 | 否 | 是 | 唯一ID |
| skins\_name | varchar | 40 | 无 | 否 | 否 | 皮肤名称 |
| value | int | 1 | 无 | 否 | 否 | 皮肤值 |
| img\_url | int | 13 | 无 | 否 | 否 | 预览图 |
| remark | varcahr | 80 | 无 | 否 | 否 | 备注信息 |
| inserted\_at | int | 13 | 无 | 否 | 否 | 插入时间 |
| updated\_at | int | 13 | 无 | 否 | 否 | 更新时间 |
| deleted\_at | int | 13 | 无 | 是 | 否 | 软删除 |

3.5 用户界面设计

系统用户界面设计对于用户使用和系统安全性来说十分重要，一个好的用户界面可以给用户带来更好的使用体验和更舒心的操作享受。选择用户界面时参照下面几个准则：

1. 界面美观简洁友好
2. 适配PC和移动设备，
3. 交互操作简单明了
4. 提示信息简洁明确

为了减少时间和开发成本，快速实现上述原则的用户界面，选择了业界成熟的开源的UI框架来实现，参考了众多UI框架，选择了iview的设计作为本系统的后台管理的界面框架和选择了Bootstrap作为个人空间的界面框架。

3.6 物理配置方案设计

个人信息融合系统是建立在信息技术的平台上，包括计算机软硬件选择、网络带宽的设计、数据库管理系统的选择等。因此，选择一个合理的物理配置设计方案对系统开发来说是非常重要的。在设计物理配置时从系统吞吐量、系统的响应时间、系统的处理方式和管理方式等方面来考虑。

由于本系统设计为个人或者家庭用户部署使用，从上述几个方面来说，选择云服务厂商提供的基本的配置如双核CPU、10M带宽和100G硬盘的配置即可满足一个家庭相当长一段时间的使用。

1. 系统实现

经过前期的分析与设计，现在到了系统实现阶段。本系统是基于Web技术实现的，Web天然的跨平台属性支持windows、linux、mac等系统，设计兼容PC与移动的浏览器。

这里介绍一下开发本系统用到的一些技术，主要包含了以下基础技术：

HTML：它的定义为超文本标记语言，用于描述、定义We网页的内容，它包括很多用来描述网页中的内容的标签例如布局标签、表单标签和文本标签等。

CSS：即层叠样式表，用于控制和描述Web页面内容的样式与布局例如文字颜色、背景和动画等。

JavaScript 是一种运行在在浏览器中的解释型脚本编程语言。 它主要用来给Web网页添加交互效果和其他动态功能，近年社区推出了可以运行在服务器上JavaScript框架Node.js，从而大大丰富了JavaScript的应用场景。

HTTP：即超文本传输协议，用于传输网页中的 HTML和其他超媒体文档，以及请求服务器接口数据等功能。

PHP：官方释义是超文本预处理器，是一种运行在服务器上脚本编程语言，由于其学习容易，上手简单，能够支持跨平台部署运行，用来进行中小型的Web开发和接口相当快速简洁，性价比也非常高。

Laravel是一个比较流行的PHP Web开发框架，因其设计非常优秀、易扩展、易维护且社区生态非常丰富，代码富有表现力，目前是世界上最受欢迎的PHP开发框架。

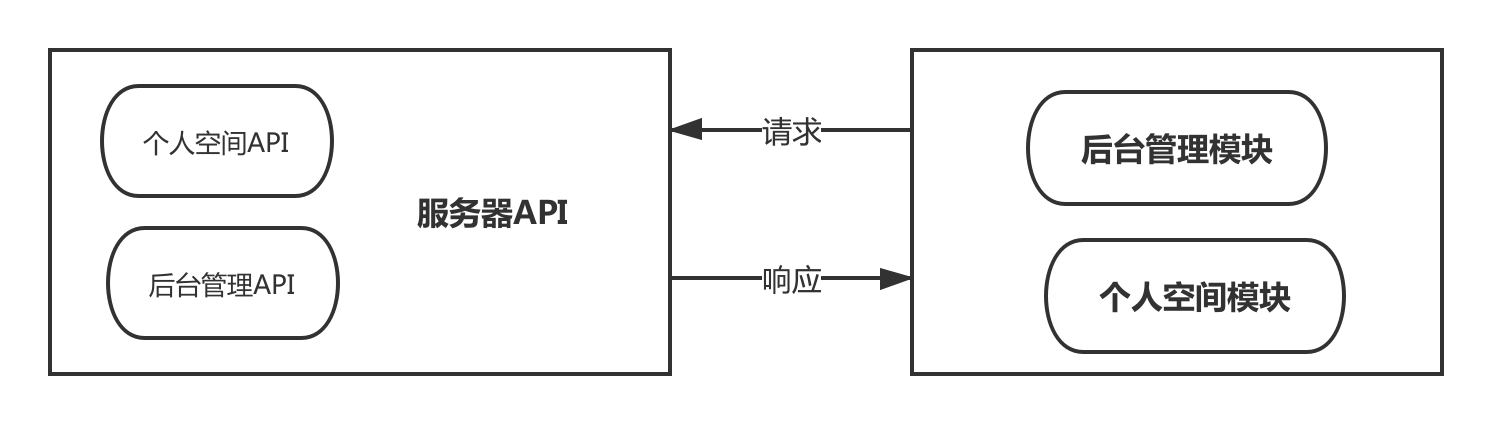
* 1. **技术选型与开发设计**

技术选型即开发系统的程序语言、框架和开发架构的设计与选择，本系统实现基于几个原则：

1. 开发迅速
2. 易维护、易扩展
3. 部署和运行稳定、灵活
4. 性能满足个人或及家庭使用

基于上述几个原则，在参考了常见的几种Web开发技术组合，选择基于PHP语言最流行的Laravel框架来开发本系统。相比Java等语言，PHP上手简单，开发快速，社区生态成熟，部署方便快捷，不需要高昂的运行和维护成本。

本系统采用前后端分离开发，把本系统分为服务器端和浏览器端分别进行开发，服务器端API包括后台管理API和个人空间API；浏览器端分为后台管理网站和个人空间网站。此外，部署的时候支持前后端独立部署，支持前端配置自定义的API接口，支持独立修改服务器接口。如下图所示



毕设图 5

* 1. **开发环境介绍**

本系统是在windows7操作系统下进行开发的，理论上支持window、Linux、mac系统上运行。开发语言使用的是PHP程序语言，服务器API开发工具使用的是使用Phpstorm开发，web前端部分选择的Webstorm开发，前端部分开发调试是在Chrome浏览器上进行的。

Web服务器：Apache 2.4

操作系统：Windows 2000/NT/XP/ME/Win7

开发工具：Phpstorm + Webstorm

数据库：MySQL5.7.2

PHP版本： 7.2.2

Laravel版本： 6.0

版本管理： Git

CPU：200MHz以上的处理器

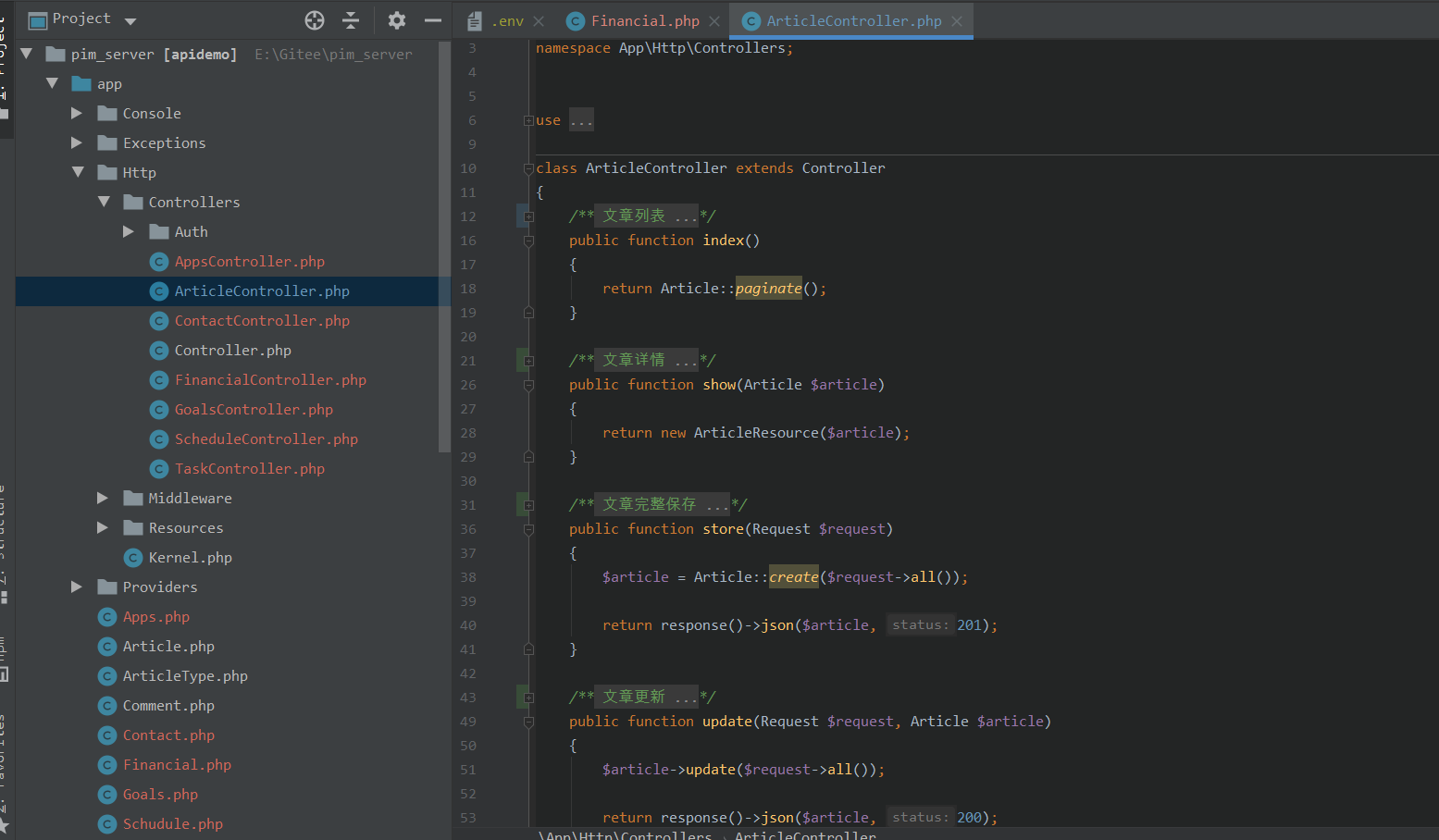
内存：512MB以上

PHP组合开发套装：laragon

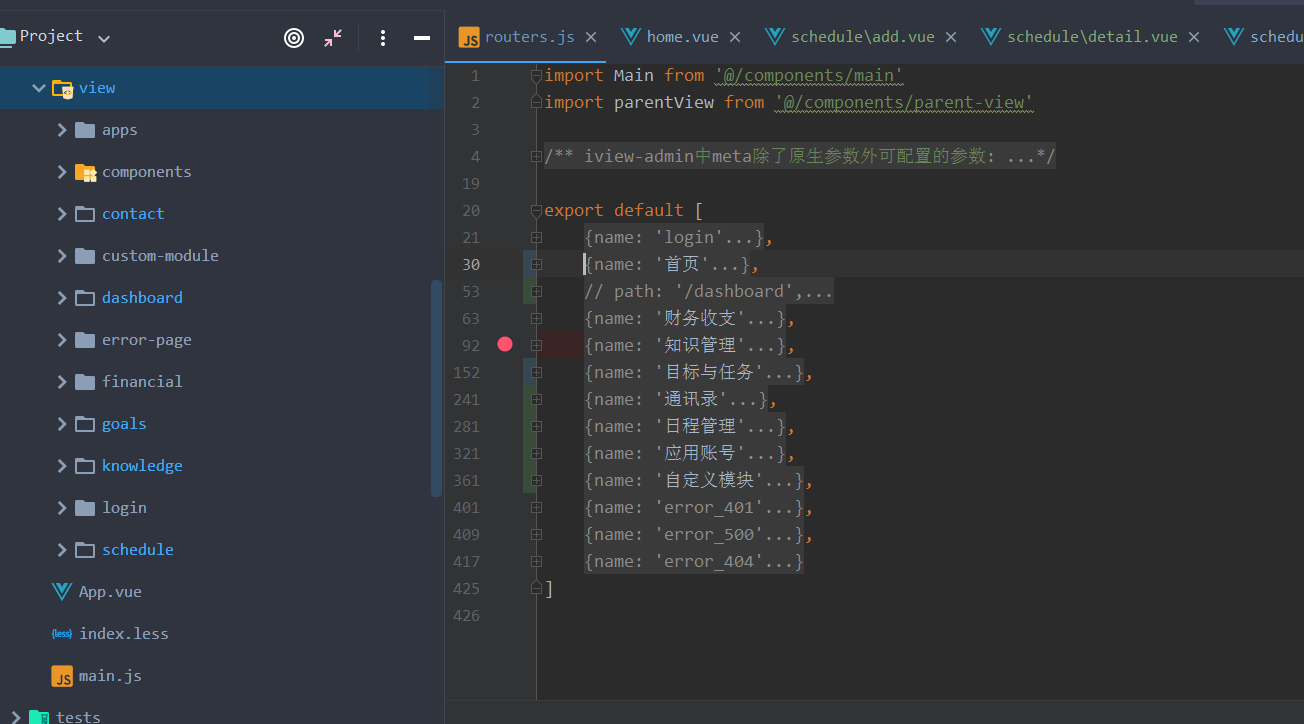
浏览器兼容：谷歌浏览器、火狐浏览器、QQ浏览器、360极速浏览器、微软Edge浏览器和IE9以上版本以上，以及webkit内核的浏览器，PC和移动端均能使用。

* 1. **模块详细实现**

首先从服务器接口说起，接口使用RESTful API风格，其表示一种应用之间网络通信的架构风格，依赖于无状态的协议HTTP进行交互。服务器接口实现使用数据模型 -> 路由 -> 控制器的编写方式，然后编写前端界面。先看一下项目结构，使用Laravel6.0创建服务器项目和Vue-cli创建的前端项目，



毕设图 6



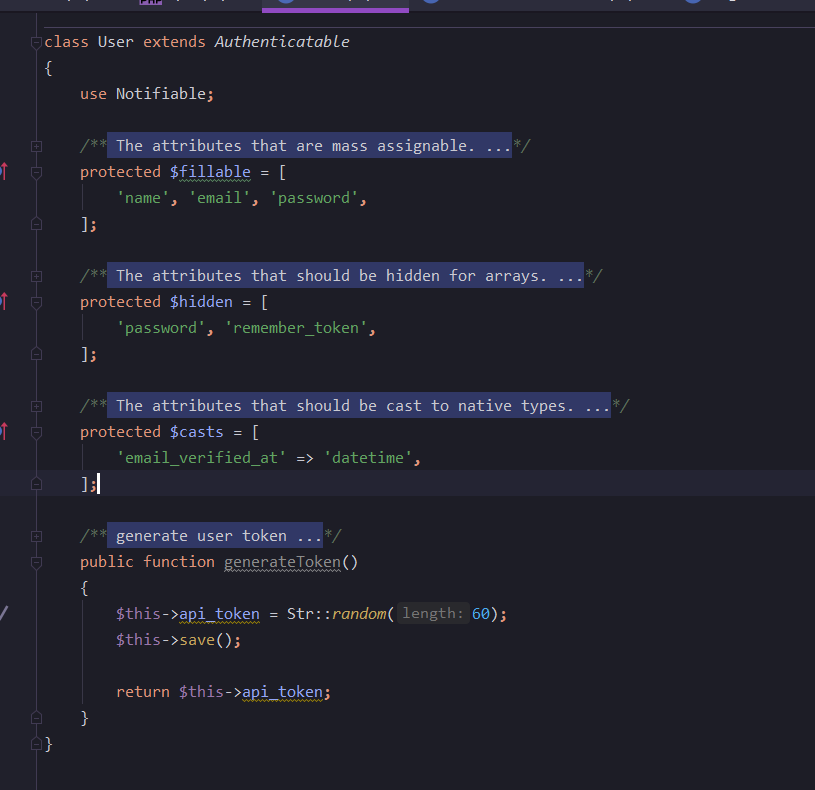
毕设图 7

* + 1. **后台管理**

1. **注册管理员用户与登录验证**

用户模型主要用来管理用户表中字段插入、更新与请求token验证，这里为了简化客户端与服务器认证的信任过程，通过在用户表中新增api\_token的字段，在新增管理员时就新增，一直有效，除非在数据库中删除该用户。

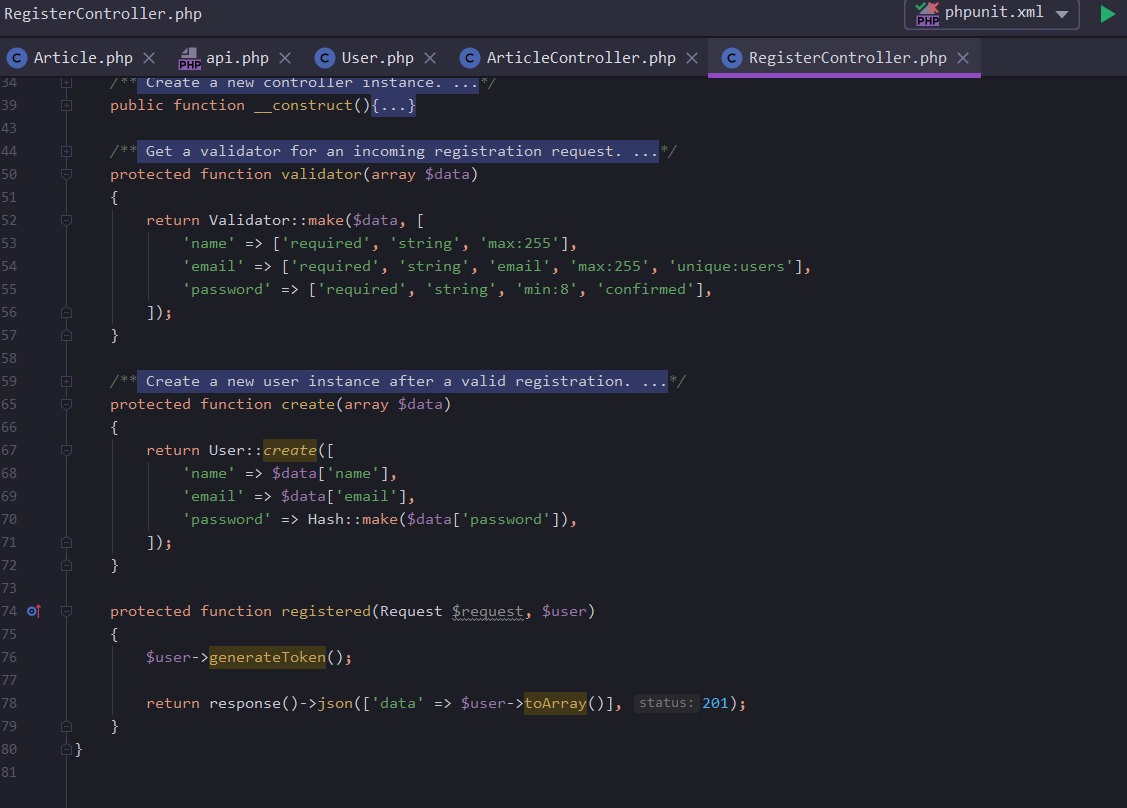
实现代码如下：



毕设图 8

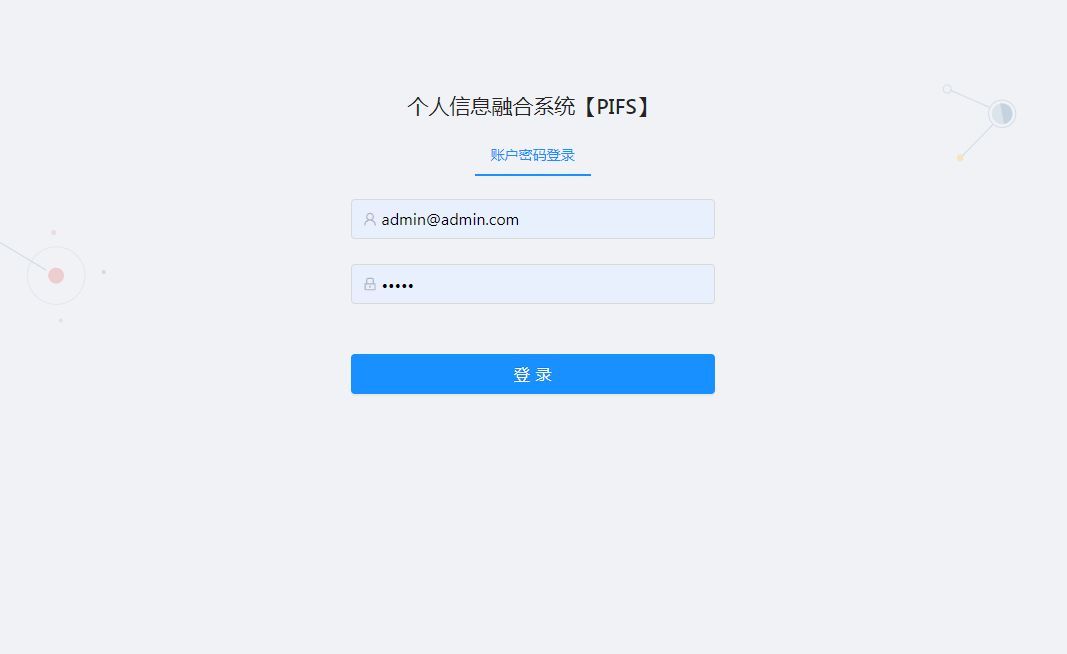
注册管理员用户控制器功能用来验证客户端请求、创建用户和验证请求是否合法。

代码如下



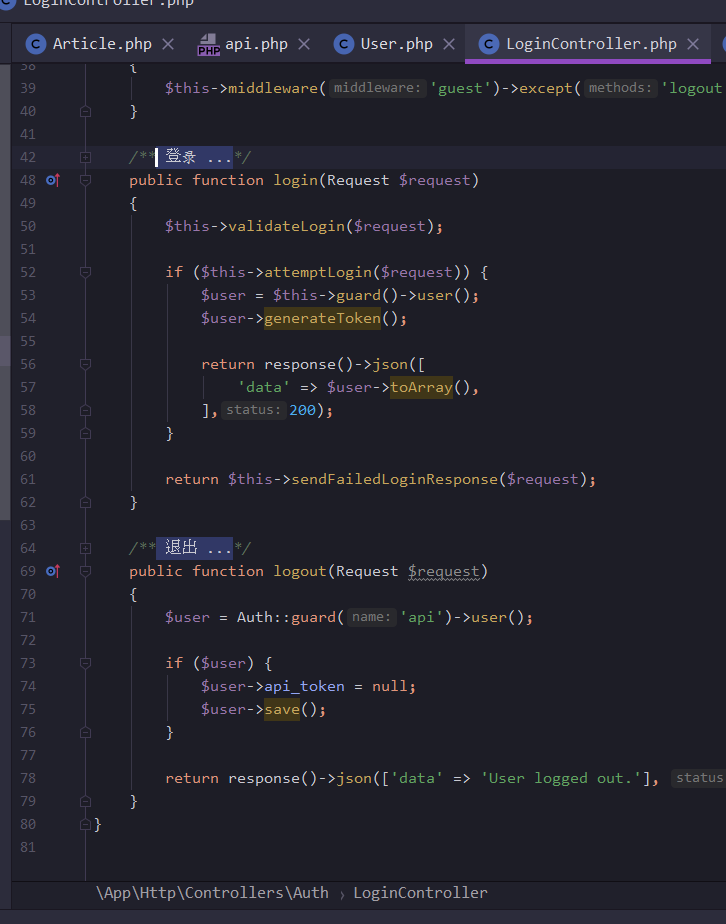
毕设图 9

前端登录界面



毕设图 10

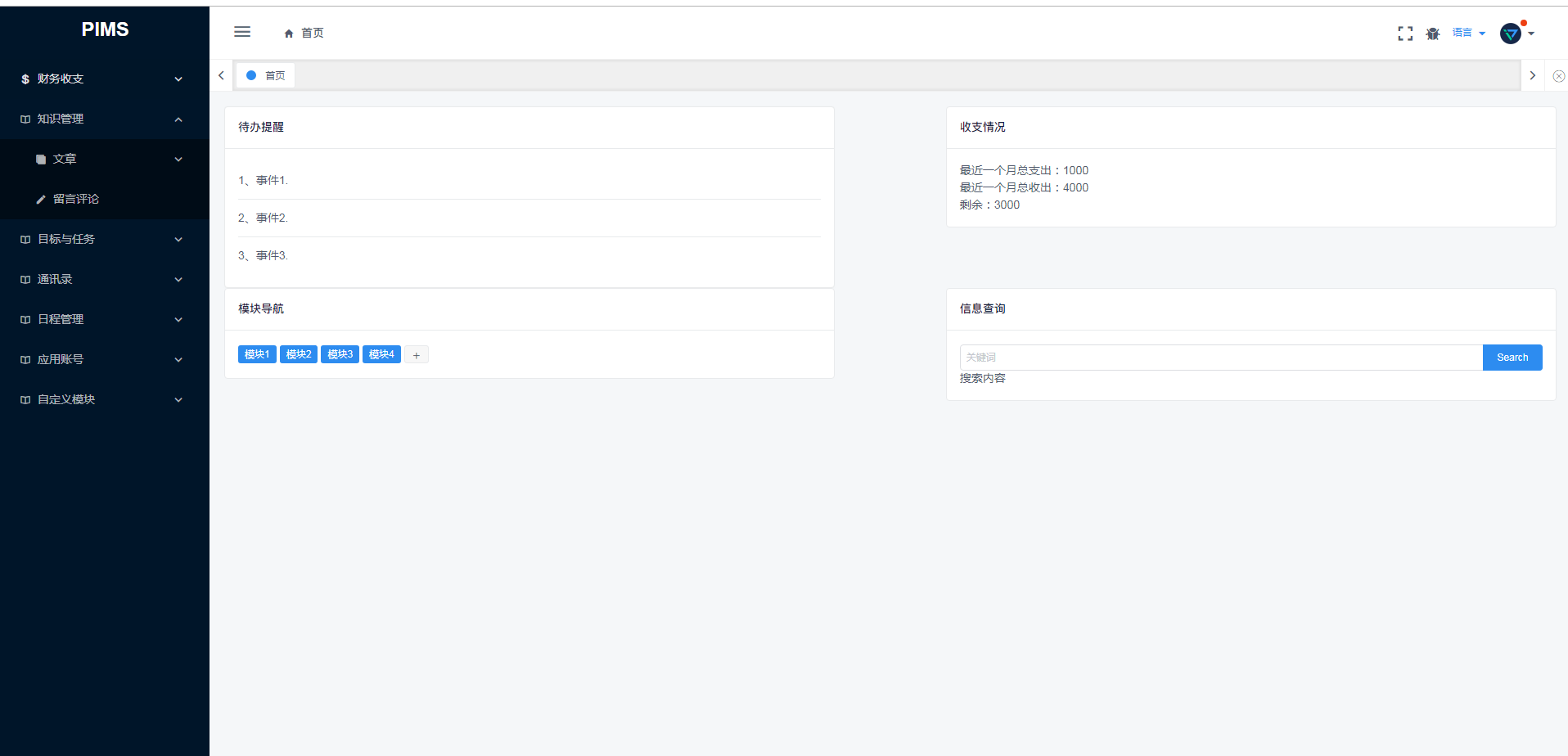
服务器接口代码实现如下：



毕设图 11

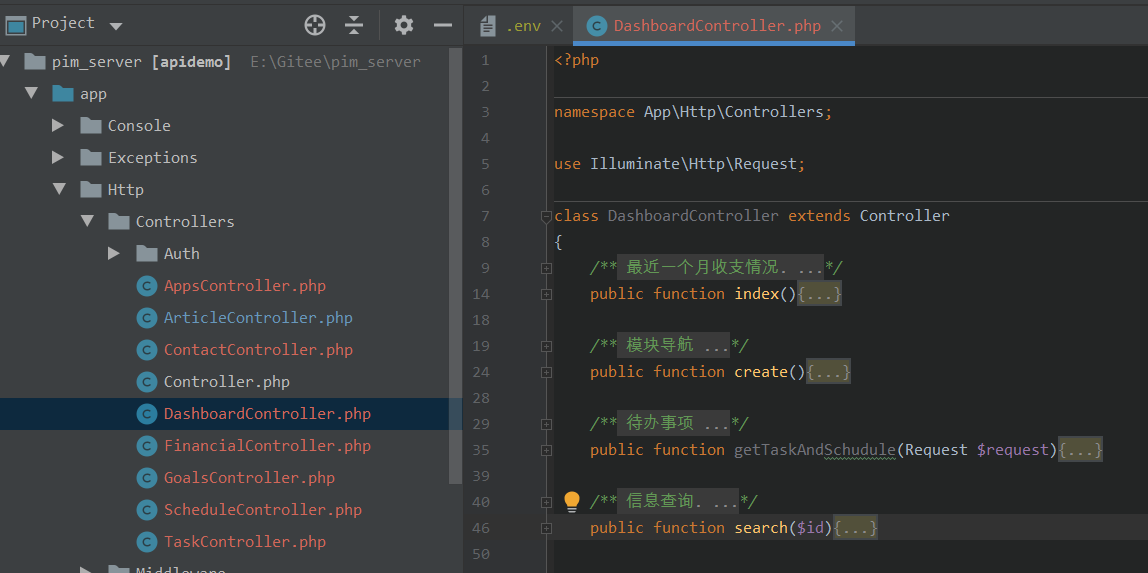
1. **综合面板**

综合面板主要展示快捷信息如下图：



毕设图 12

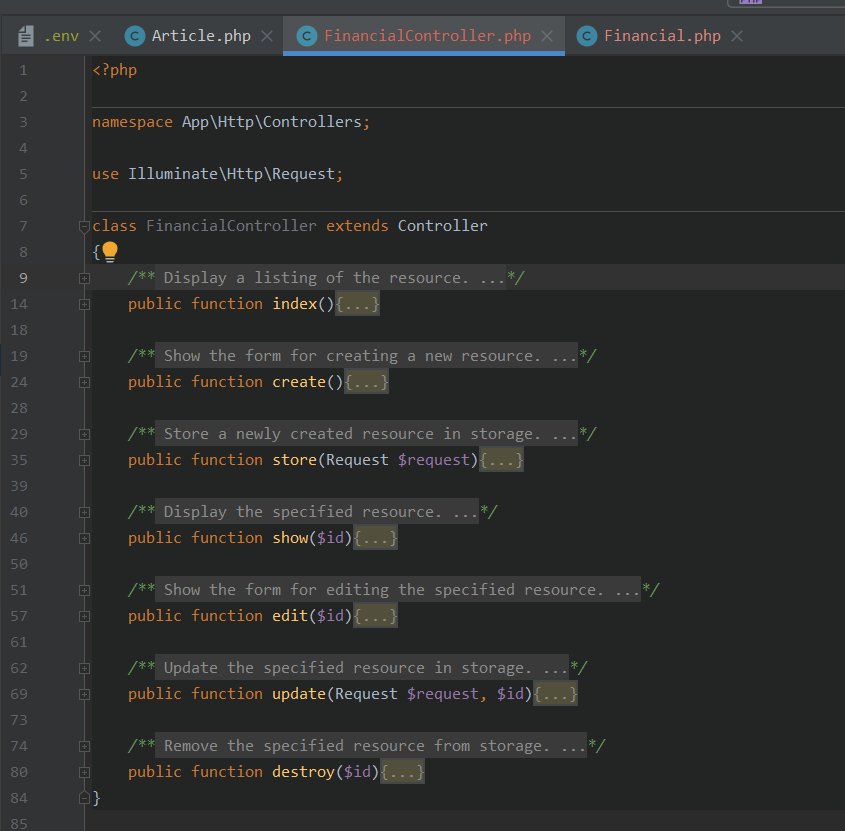
服务器主要逻辑代码：



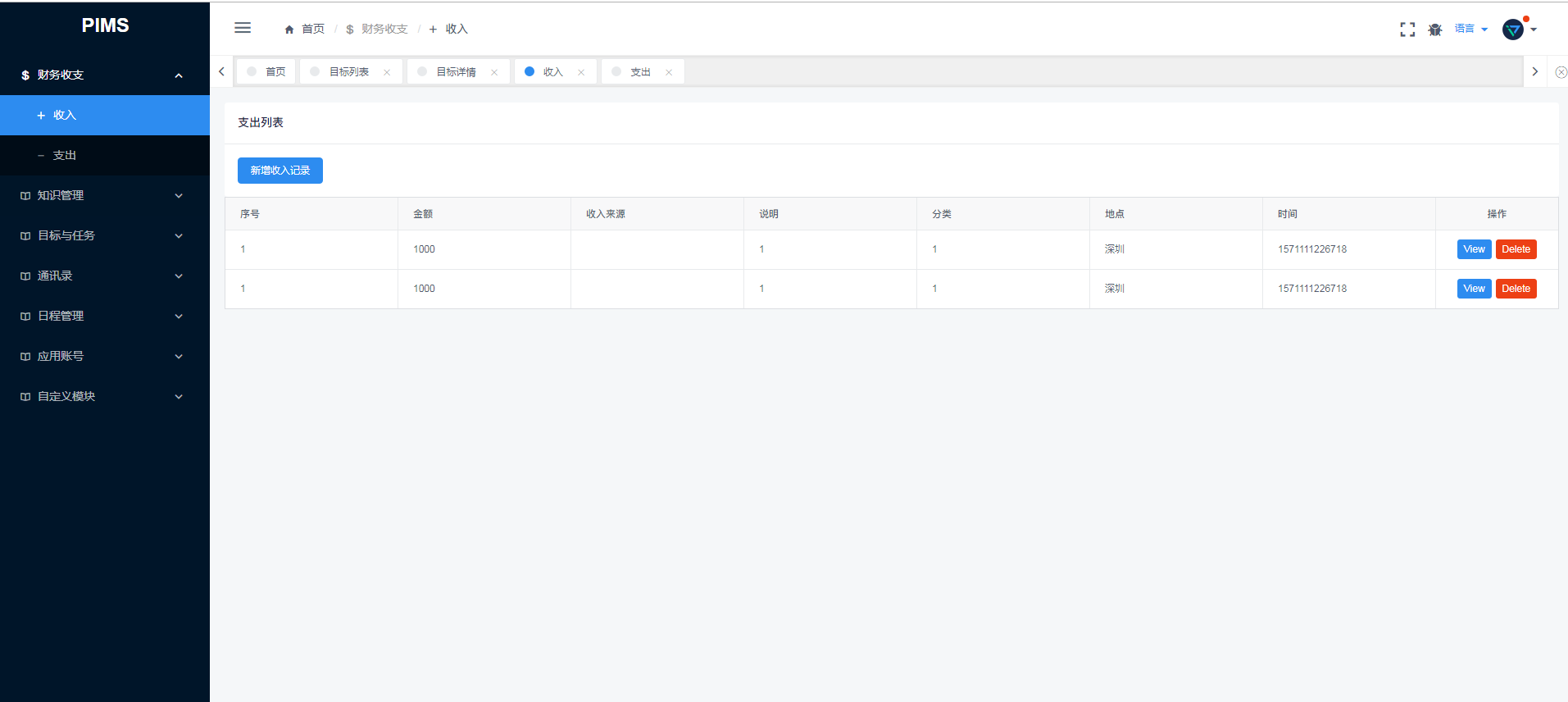
毕设图 13

1. **财务收支管理**

财务支出主要实现了个人或者家庭的收入与支出的管理。服务器代码和前端代码



毕设图 14

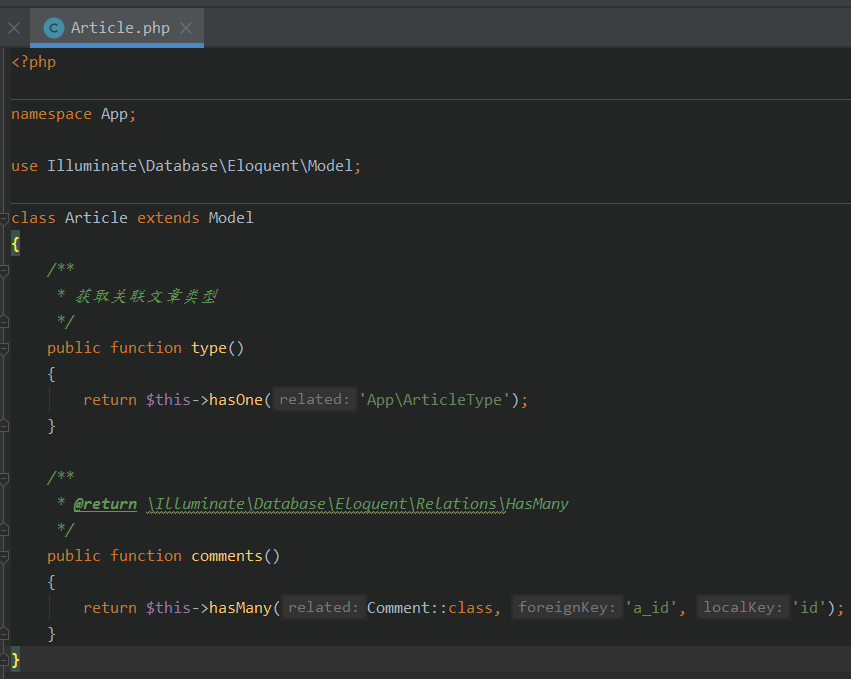


毕设图 15

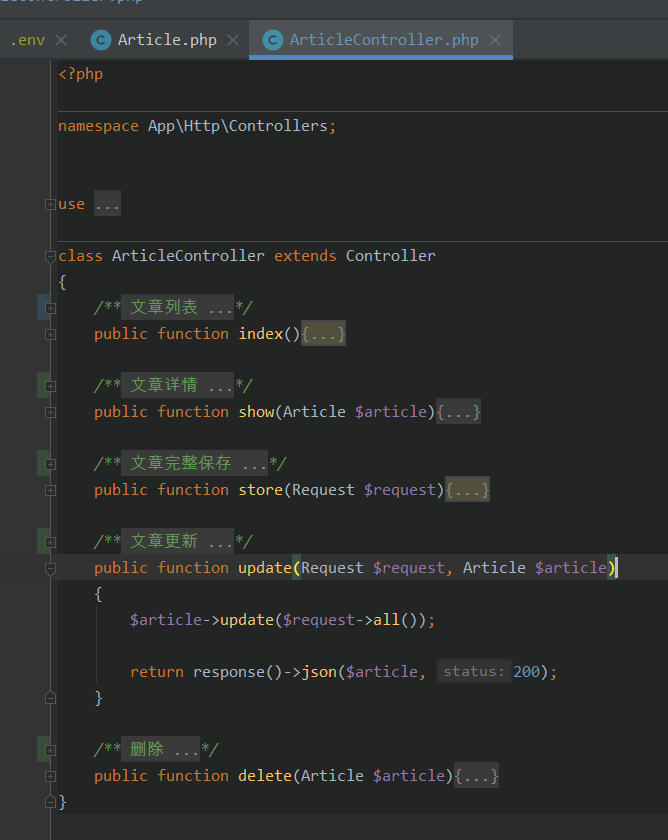
1. **知识管理**

知识管理用来管理文章和关联的评论。具体代码如下

Article模型拥有1:N个关联评论留言模型，属于一个唯一的文章类型。

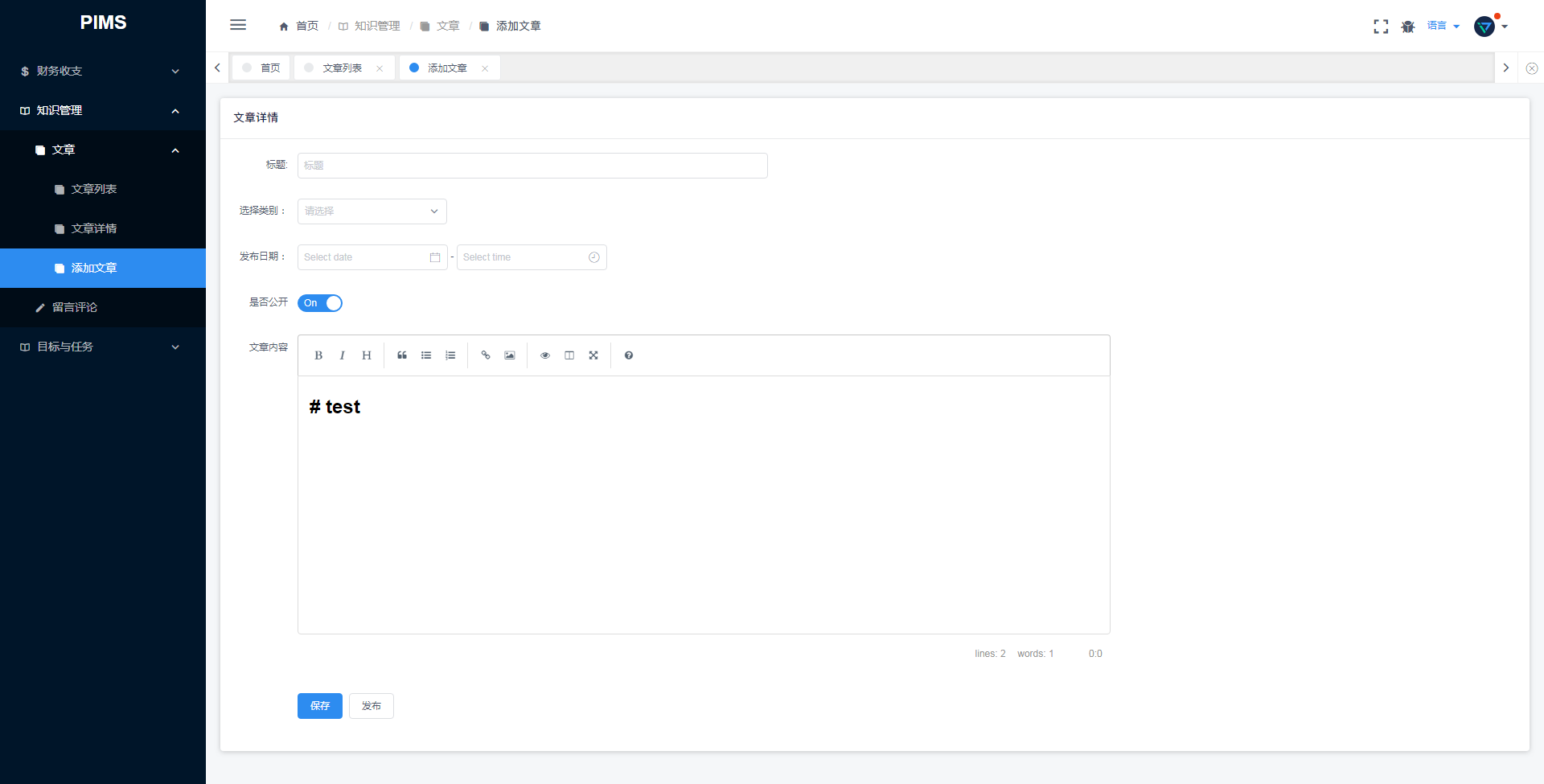
毕设图 16

控制器CURD代码实现

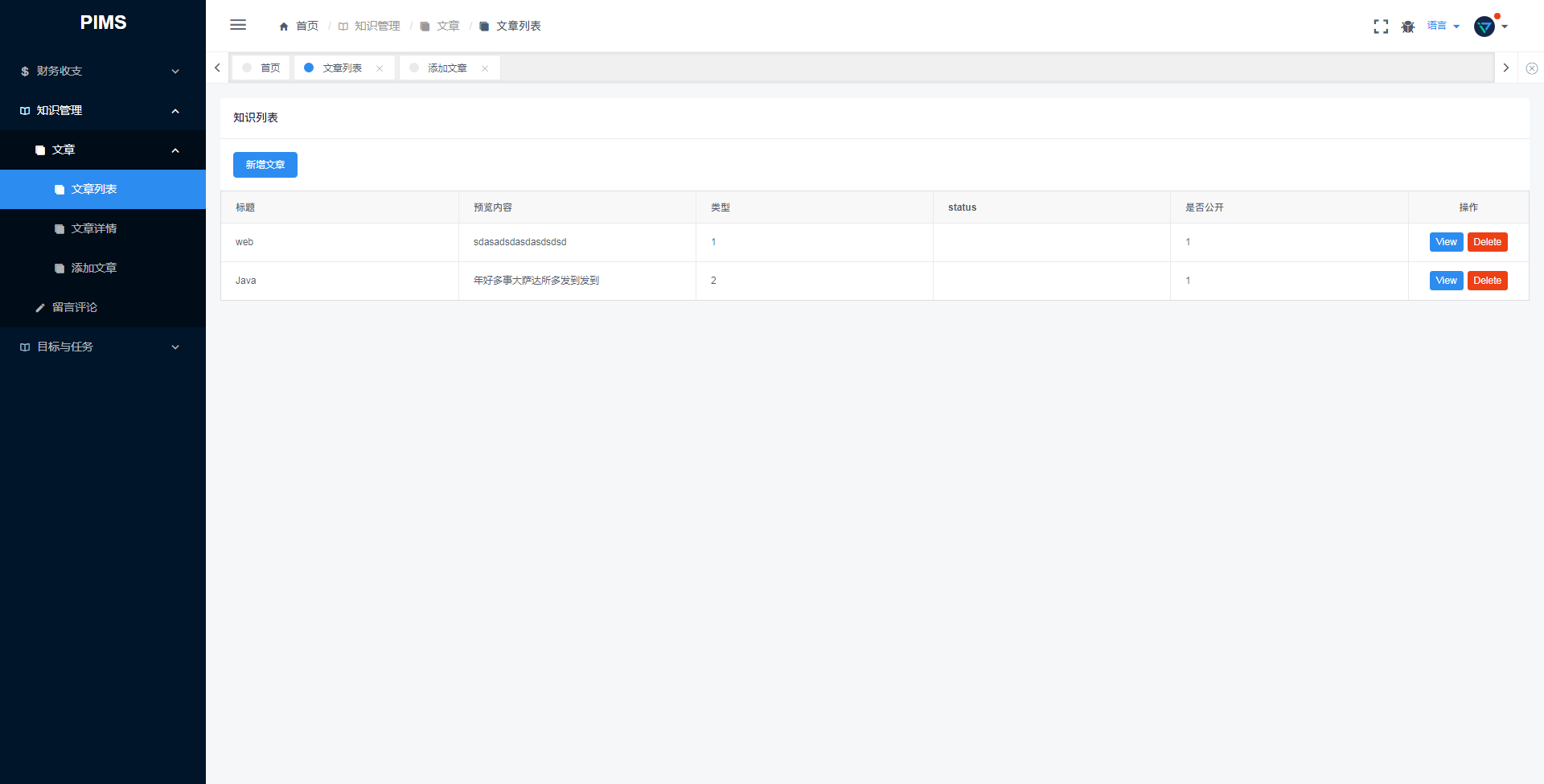


毕设图 17

前端新增、列表与更新界面



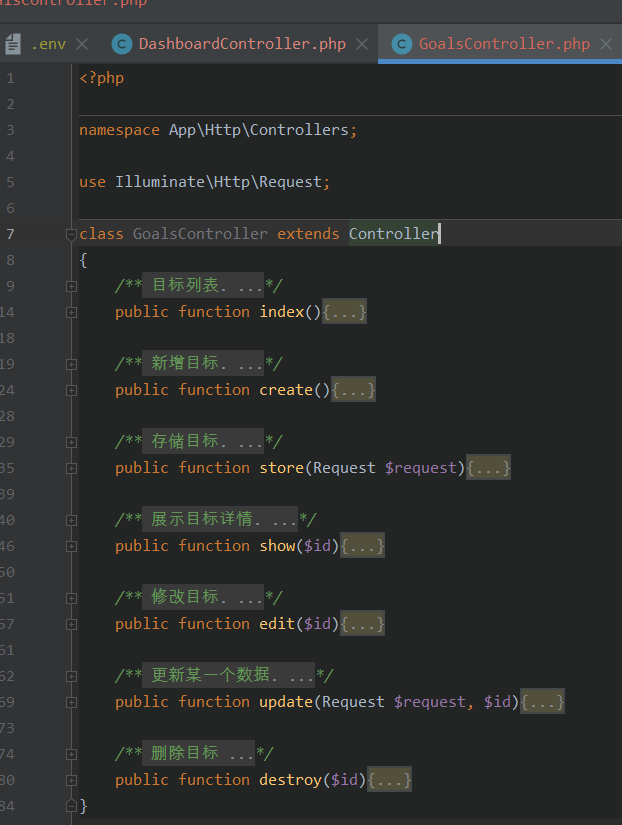
毕设图 18



毕设图 19

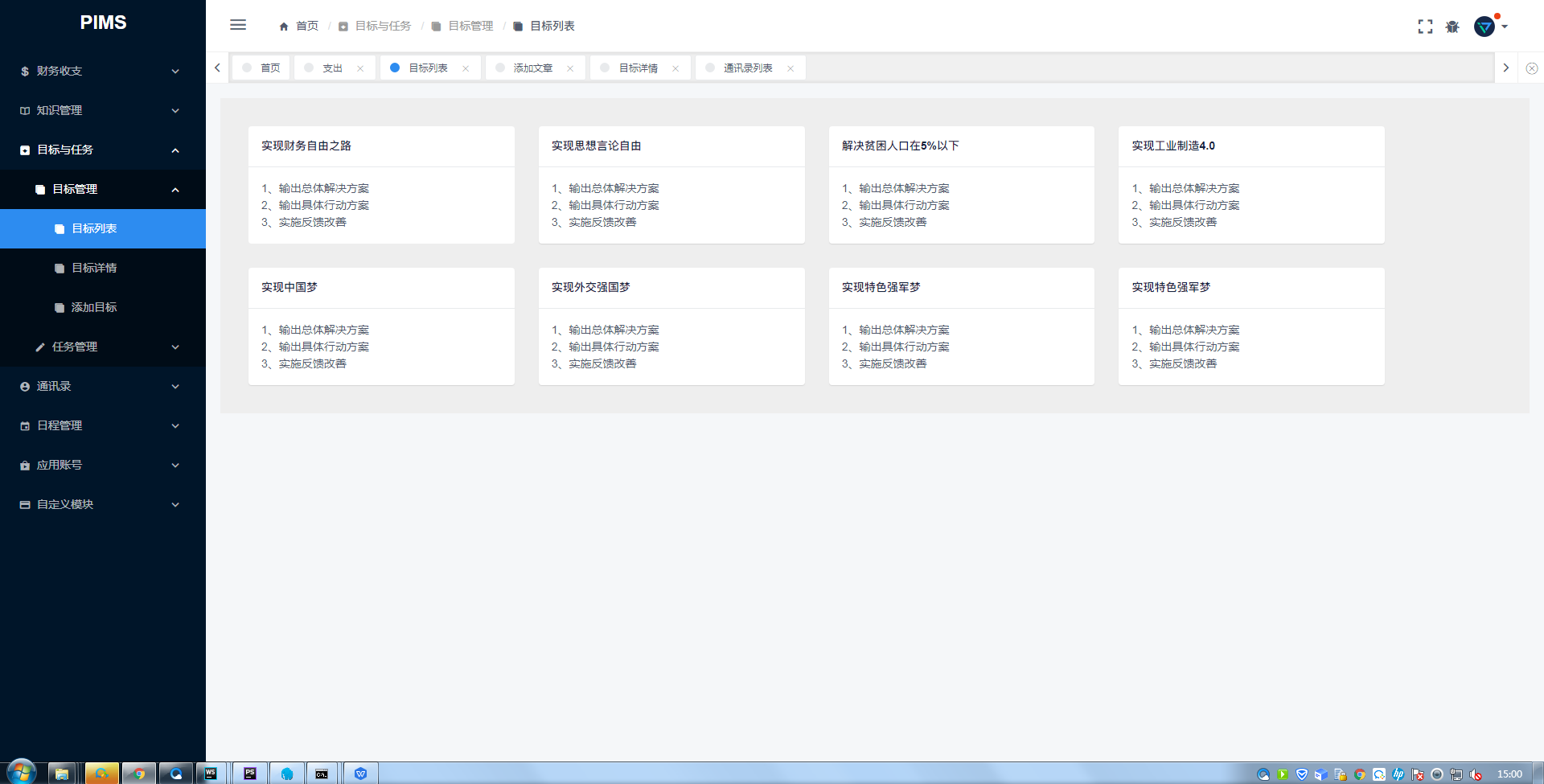
1. **目标与任务管理**

目标与任务模块实现了目标与其关联的任务管理的。服务器代码和前端界面



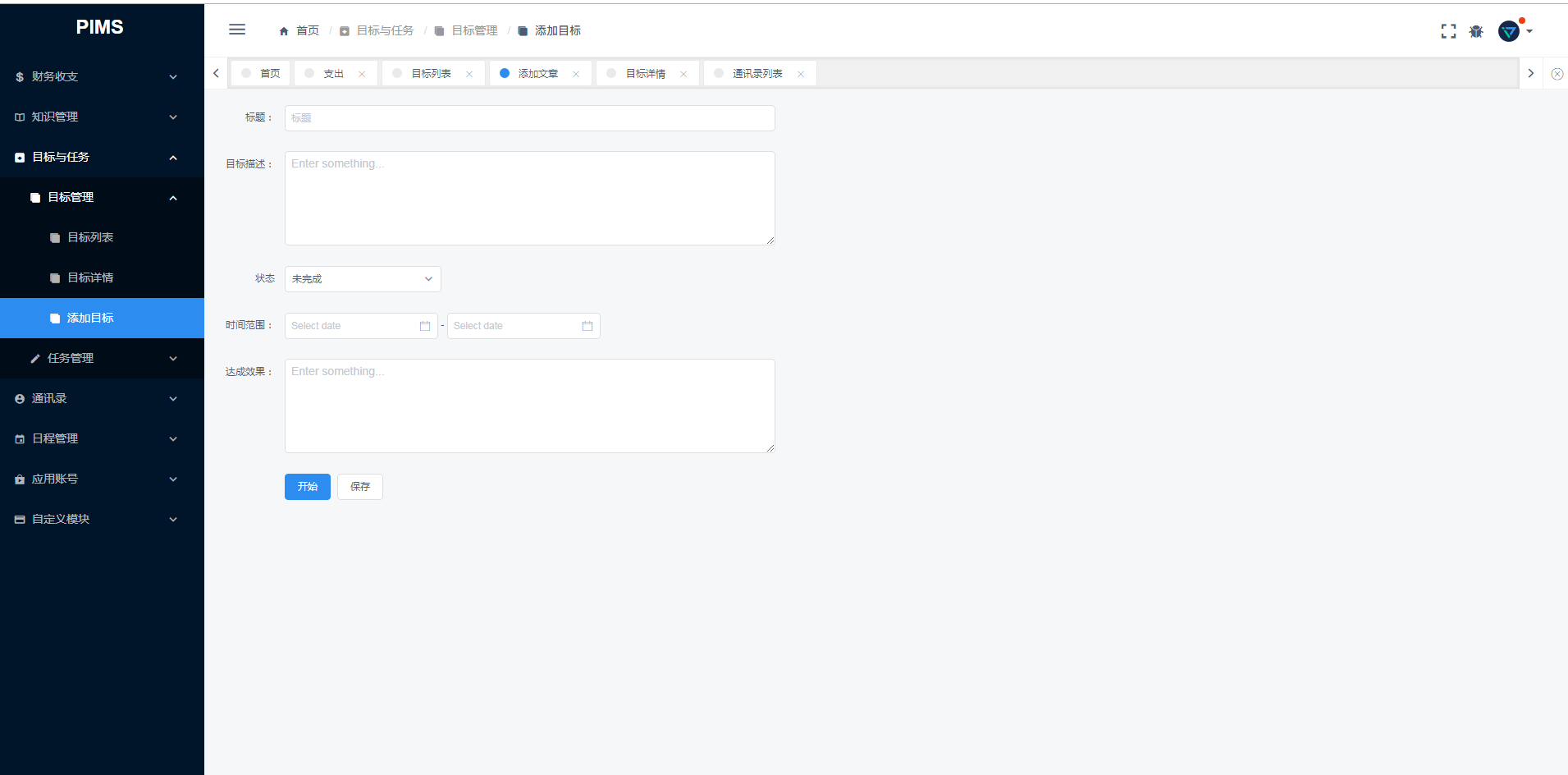
毕设图 20

目标列表：



毕设图 21

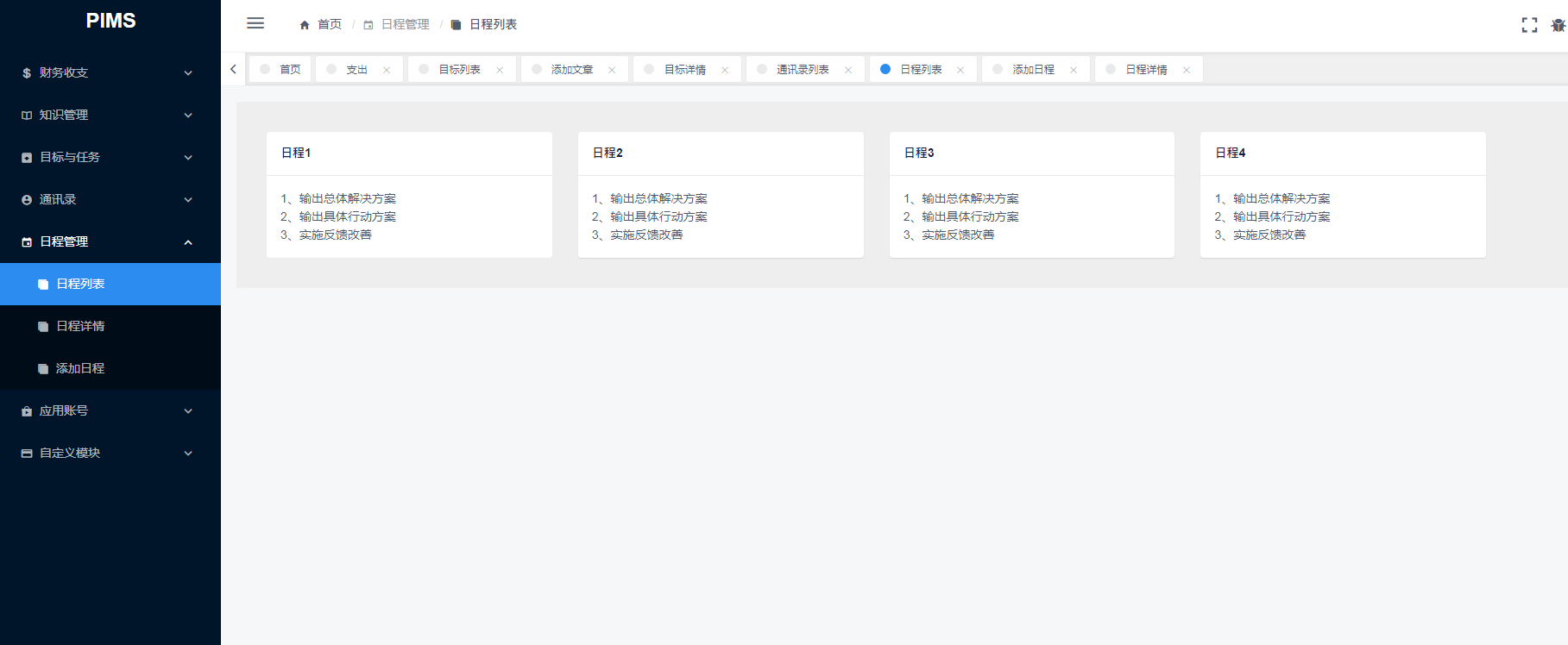
添加目标：



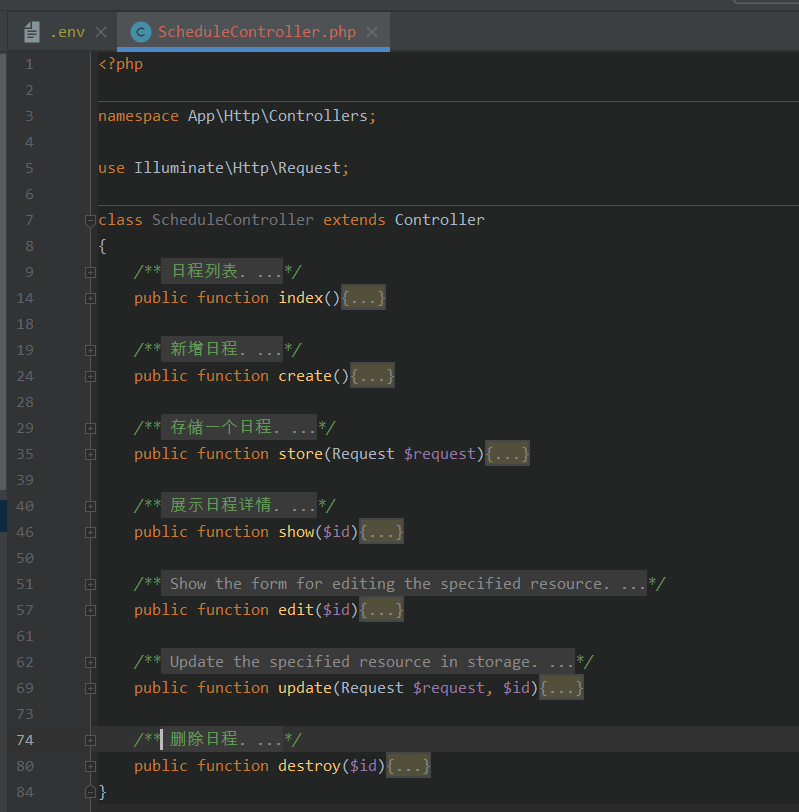
毕设图 22

1. **日程管理**

日程管理功能主要实现了对日程的增删改查操作。服务器代码和前端界面如下：



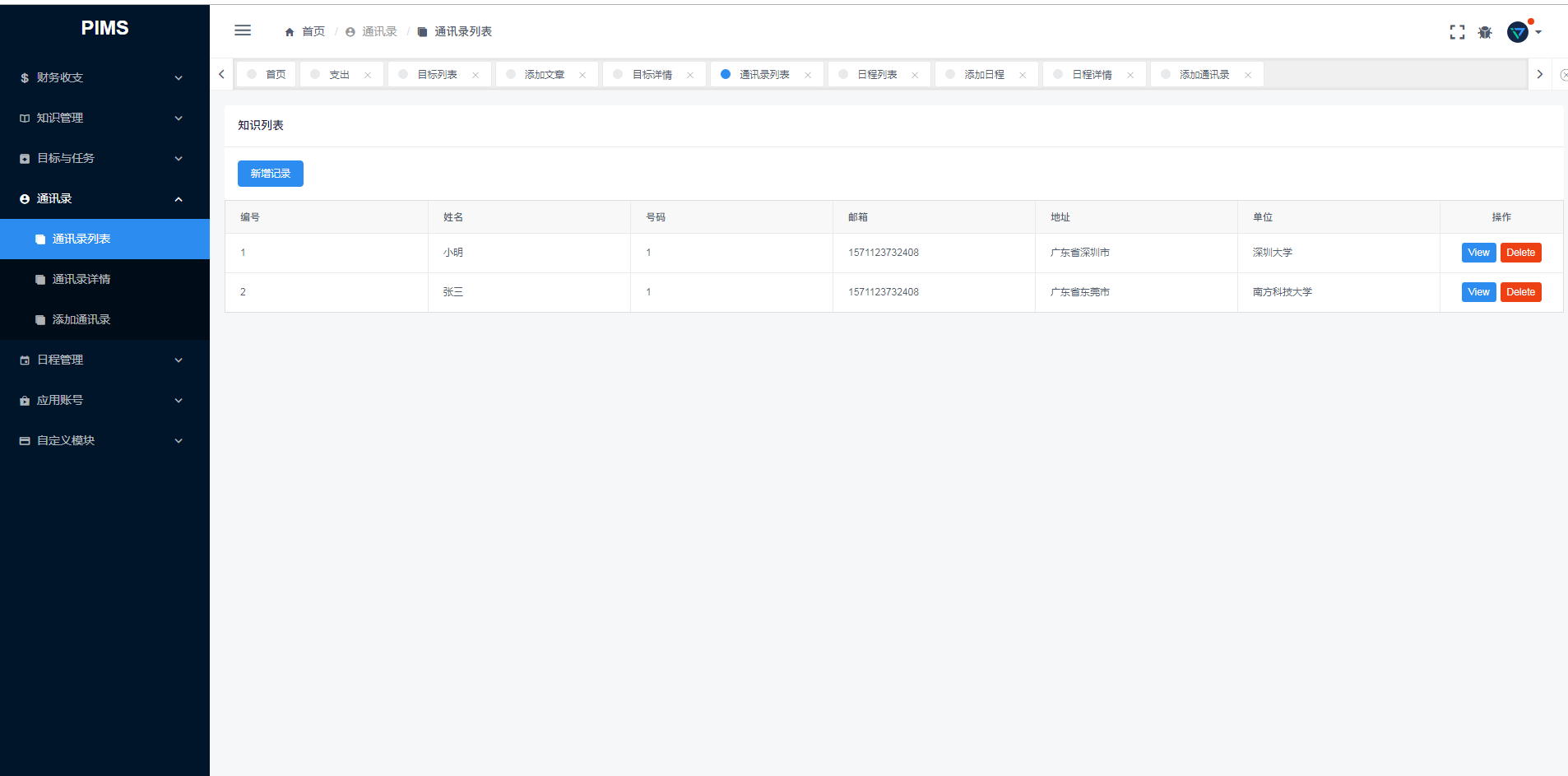
毕设图 23



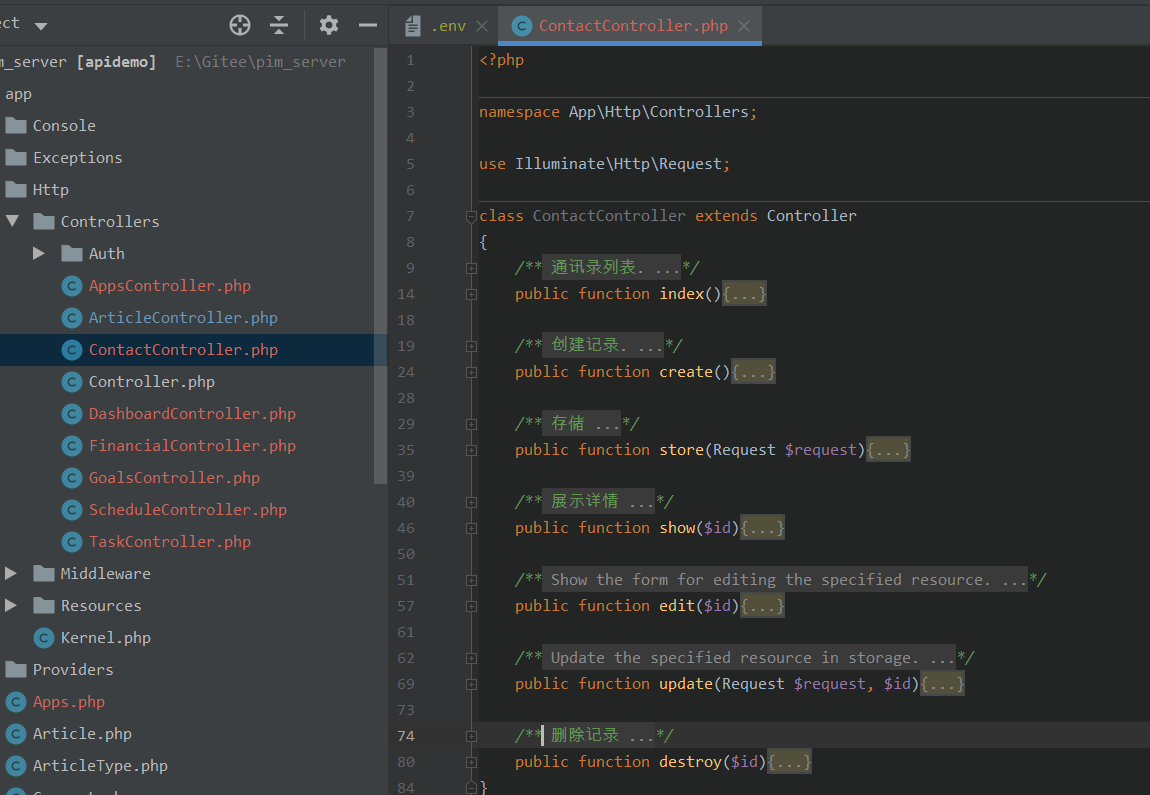
毕设图 24

1. **通讯录管理**

通讯录管理主要实现了个人或者家庭的亲朋好友的通讯和备忘信息。服务器代码和前端界面如下：



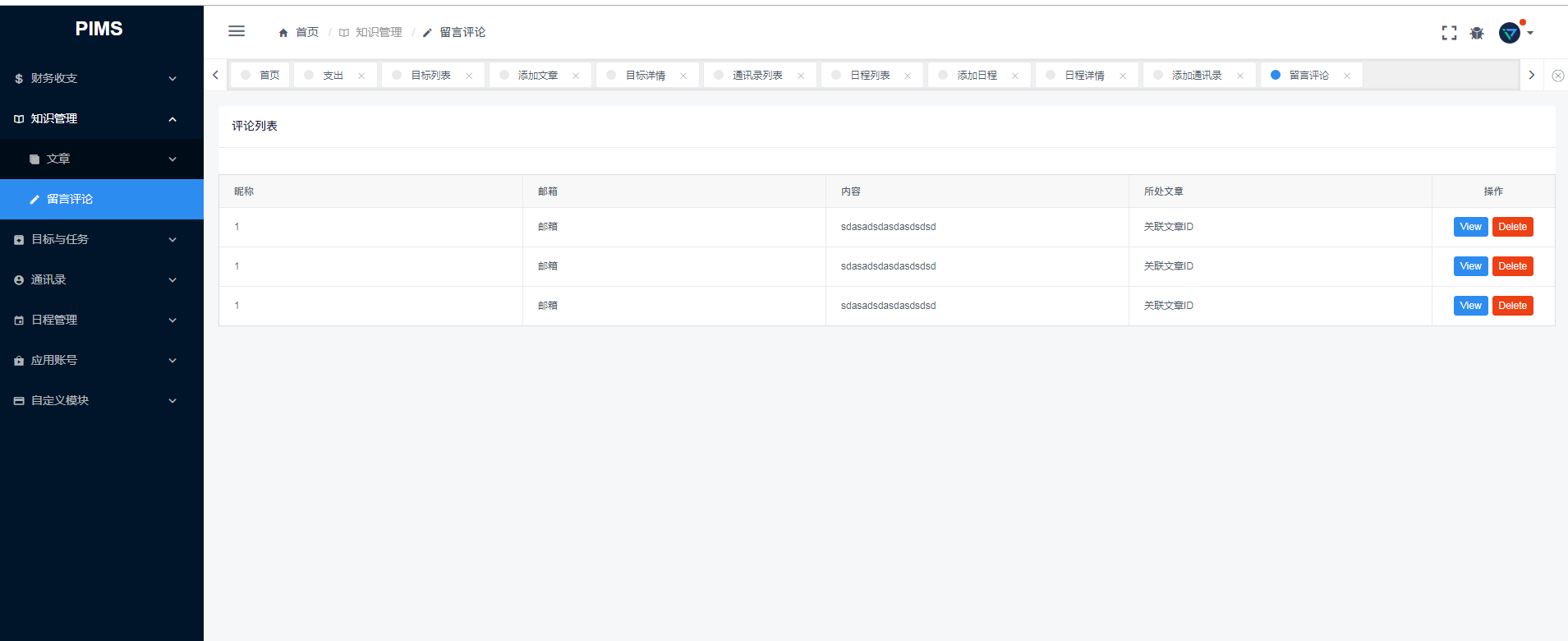
毕设图 25



毕设图 26

1. **评论留言管理**

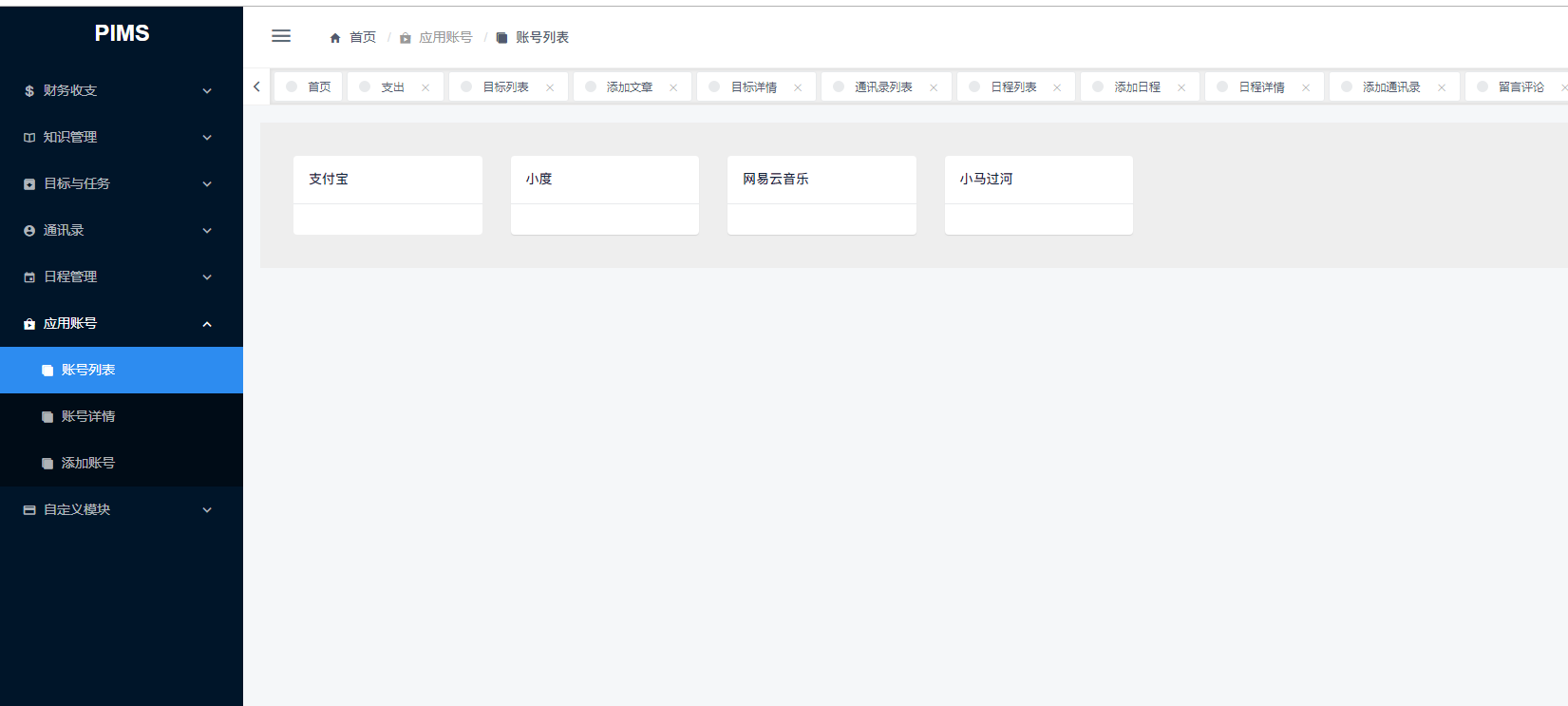
评论留言管理功能主要实现了对文章的评论与留言管理功能。前端界面如下



毕设图 27

1. **账号管理**

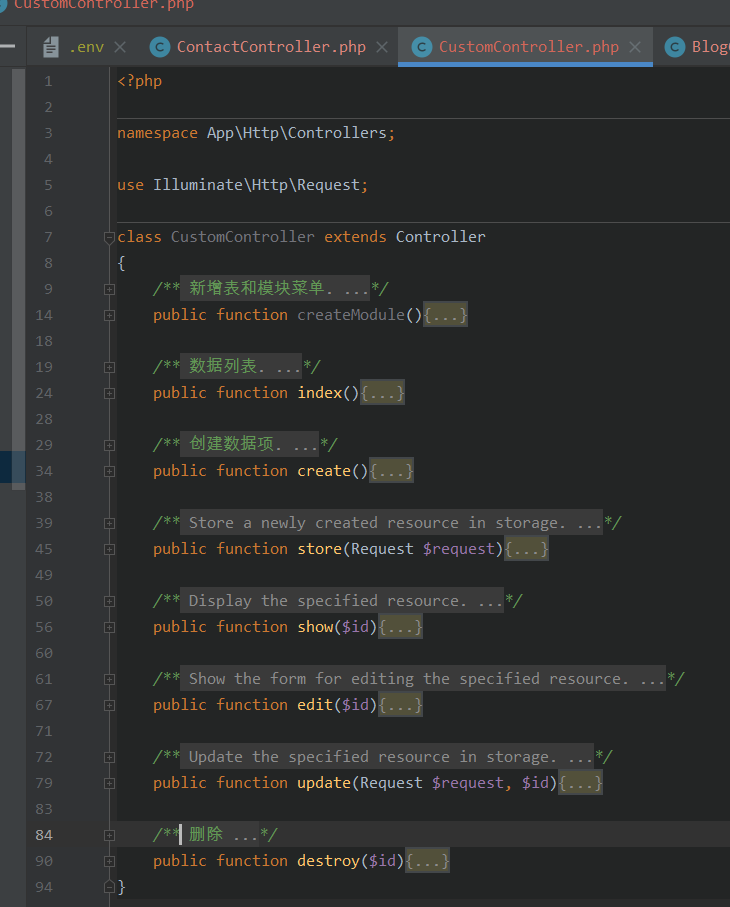
账号管理实现了个人各种应用软件的用户名和密码的管理以防止遗忘。界面如下



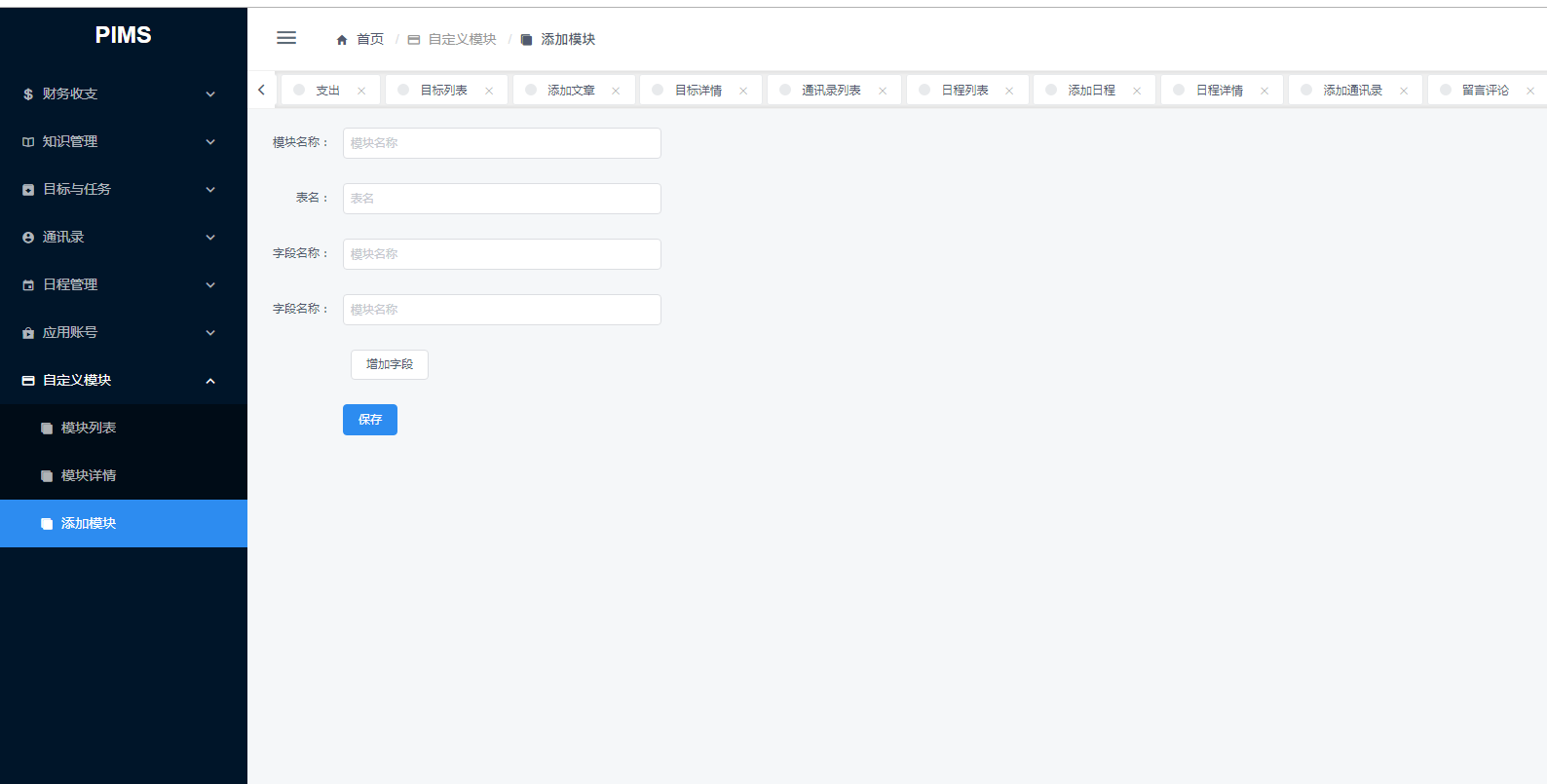
毕设图 28

1. **自定义模块管理**

自定义模块的功能实现了新增自定义字段的表，然后对其进行简单的增删改查功能。实现的思路是第一现在模块表中插入一项纪录，表示新增一个菜单模块，第二个根据用户提交的数据字段和表名动态创建数据表，第三步实现对其进行增删改查的操作。界面和服务器代码逻辑如下：



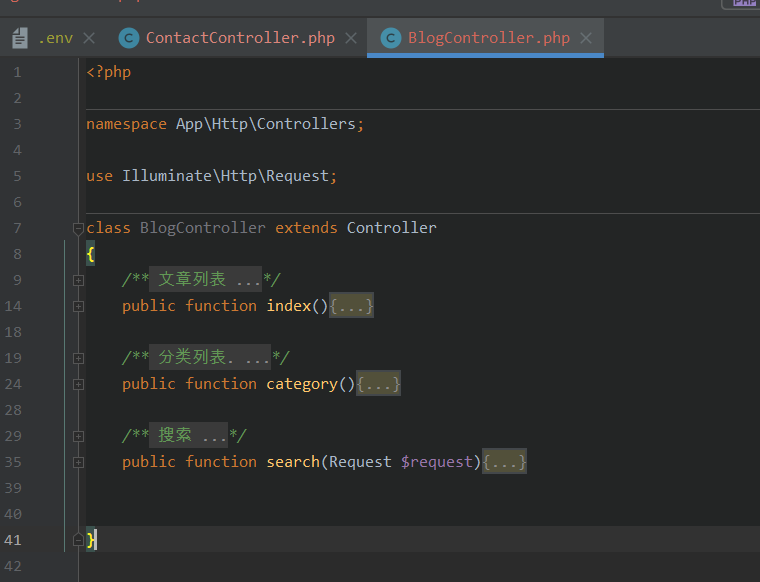
毕设图 29



毕设图 30

* + 1. **个人空间**

个人空间的功能主要分页获取公开的知识列表数据和游客提交留言评论功能。



毕设图 31

****

毕设图 32

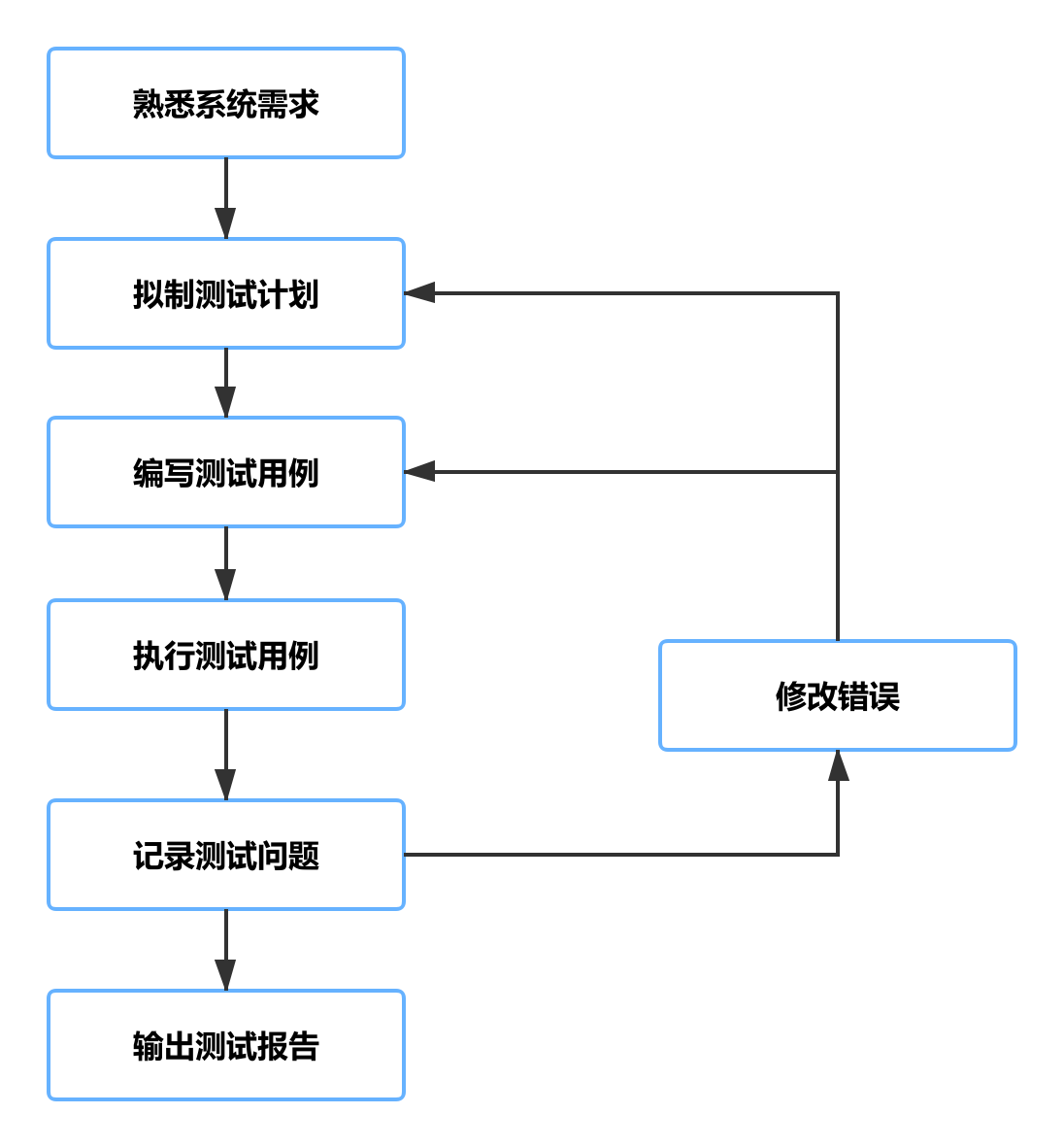
1. 系统测试与维护

5.1 系统测试

测试的目的是为了发现程序的错误而执行程序的过程，优秀的测试方案总是在寻找尚未可知的错误的测试方案。应该清楚地认识到没有一种测试方案能够保证发现程序中所有的问题，因此测试的目标应该是投入有限的测试资源尽可能地发现更多的错误，使收益最大化。

1. **测试基本流程**

为了保证的测试工作的有效性，软件测试遵循有效的基本流程。



毕设图 33

1. **测试方法与原则**
2. **黑盒测试法**

这是软件测试中比较重要的一种测试方法,又称为功能测试、数据驱动测试或基于规格说明的测试。基本定义是测试者不了解程序的内部逻辑,即把程序看作是一个黑盒,不需具备应用程序的代码、内部结构和编程语言的专门知识,只知道程序的输入、输出和系统的功能,这是从用户的角度针对软件界面、功能及外部结构进行测试,而不考虑程序内部逻辑结构。测试案例是应用系统应该做的功能,依照规范、规格或要求等设计。测试者选择有效输入和无效输入来验证是否正确的输出。常用的具体的黑盒测试方法有等价划分法、边界值分析和错误推测法等。此测试方法可适合大部分的软件测试,如集成测试以及系统测试。

1. **白盒测试法**

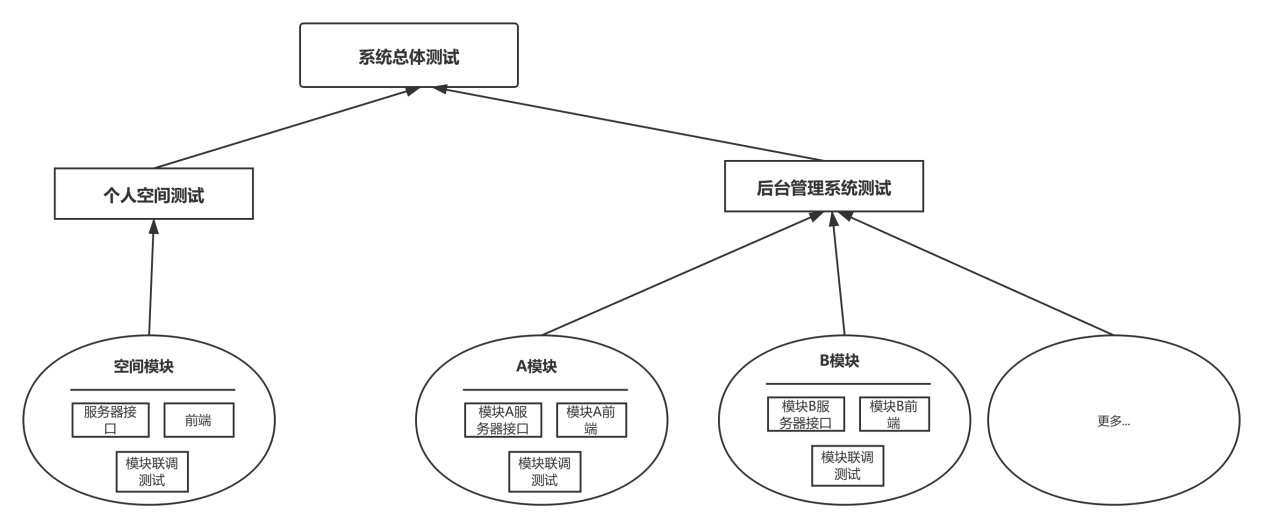
白盒测试又称透明盒测试、结构测试等,也是软件测试中常用的一种方法,也称结构测试、逻辑驱动测试或基于程序本身的测试。测试应用程序的内部结构或运作,而不是测试应用程序的功能(即黑盒测试)。在白盒测试时,以编程语言的角度来设计测试案例。测试者输入数据验证数据流在程序中的流动路径,并确定适当的输出,类似测试电路中的节点。测试者了解待测试程序的内部结构、算法等信息,这是从程序设计者的角度对程序进行的测试。白盒测试可以应用于单元测试、集成测试和系统的软件测试流程,可测试在集成过程中每一单元之间的路径,或者主系统跟子系统中的测试。尽管这种测试的方法可以发现许多的错误或问题,它可能无法检测未使用部分的规范。白盒测试考虑的测试用例主要针对程序内部逻辑的覆盖程度,按覆盖程度从低到高可分为语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖。

1. **测试原则**

由于功能逻辑的不同、时间紧迫程度和资源分布情况等原因，实际上测试会采用不同的测试方法，加上在一些测试原则的指导下，按照一定的步骤进行测试工作，通常重点遵循以下几个原则：

1. 明确预期的结果
2. 避免自己测试自己的程序
3. 避免进行破坏性测试。
4. 避免程序多余的功能
5. 保留测试数据，以便后续复测
6. **测试步骤**

测试稍微复杂的系统是一件比较复杂的工作，其所要求的创造力可能超过设计这个程序所要求的。由前面几个章节的步骤可知，系统由许多个模块组成。因此，为了提高测试效率，一般按照从分到总的方式进行测试工作。如下图示



毕设图 34

5.2 系统维护

系统开发和测试完后，面临着后续运行维护工作。正常维护的内容包括硬件设备的维护、系统软件的维护、应用软件系统维护、网络维护、数据维护和运行环境六部分内容。

本系统属于小型信息系统，服务器这一块一般是采用各大云厂商的云服务，自建机房成本过高，云服务提供商提供硬件、系统软件和网络维护这三个块维护支持。因此，接下来重点阐述数据维护和支持本系统运行的应用软件的维护。

1. **数据维护**

信息管理系统的重心就是数据，数据的维护是重中之重。数据维护工作主要负责包括数据库的安全、完整和一致性。数据库的查询、插入、更新和删除必须检查权限。同时还要维护数据库中的数据变化包括数据类型、长度等。另外还要定期备份数据，防止服务器硬件出现问题而导致的数据不可恢复的问题。

数据备份与恢复也是数据维护不可或缺的一部分，主要为了应付非正常原因导致的数据库文件损坏或者系统瘫痪等灾难的发生。应付系统灾难的基本流程包括计划、备份、模拟演练、组织、分析等步骤。数据备份管理的主要方法有数据备份、数据恢复、数据归档、归档恢复、在线备份、离线备份、增量备份、并行备份以及数据克隆。

1. **应用软件维护**

应用软件维护是比较复杂的工作，由于涉及软硬件等方面，要保证系统正常运行就必须要确保各种应用软件维护的及时性、正确。本系统牵涉应用软件的维护主要包括支持本系统运行的软件的升级与维护。包括Web服务器Apache,数据库软件MySQL,PHP各种扩展软件、SSL证书等。由于系统升级、扩展功能要求版本升级以及证书过期及时更新修复。避免因应用软件的问题而导致系统服务不正常。

1. 结束语

随着时代的发展，与个人相关的信息越来越多，个人信息融合的需求越来越强，使用个人信息融合系统的人也会逐渐增多，我相信设计开发本系统的意义会越来越明显。同时也会不断地迭代更新该系统，使其功能越来丰富，越来越贴近个人信息融合的需求。

经过本次的毕业设计，让我对信息管理系统的设计与开发的各个流程更加熟悉，对编程开发也有了深刻的认识，还积累了编程开发的经验。同时让我对信息管理系统相关的理论更加清晰，相信这些知识和禁言对我以后的工作和学习中将产生不可磨灭的影响。

由于时间和精力有限，以及设计经验和编程能力的欠缺，本系统尚有比较多的带改进的部分，还有一些功能由于本人的技术能力的原因并没有写入论文中，将在后续迭代中继续完善。

【参考文献】

1. 从国际个人信息管理专题研讨会（ISPIM）看当前个人信息管理研究的热点[C]. 占南、谢阳群、汪传雷
2. https://wiki.mbalib.com/wiki/%E4%B8%AA%E4%BA%BA%E4%BF%A1%E6%81%AF%E7%AE%A1%E7%90%86（个人信息管理.维基百科）
3. <https://baike.baidu.com/item/%E4%B8%AA%E4%BA%BA%E4%BF%A1%E6%81%AF%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/7122794?fromtitle=%E4%B8%AA%E4%BA%BA%E4%BF%A1%E6%81%AF%E7%AE%A1%E7%90%86&fromid=23721642&fr=aladdin>(个人信息管理.百度百科)
4. 信息系统开发与管理[M]. 刘世峰主编 机械工业出版社出版
5. 管理信息系统[M]. 杨一平马慧主编 经济科学出版社出版
6. 信息资源管理[M]. 武刚主编 机械工业出版社出版 118 - 119
7. 软件开发流程与项目管理[M] 邹承俊，张瑾主编 中国水利水电出版社出版
8. <https://developer.mozilla.org/zh-CN/docs/Web>（web开发技术）
9. <https://zh.wikipedia.org/wiki/PHP>（PHP维基百科）
10. 基于云平台的个人信息管理系统的设计与实现. 钟承硕士论文 华中科技大学 2014
11. [https://blog.csdn.net/weixin\_43951932/article/details/88970510#\_Toc533170026](https://blog.csdn.net/weixin_43951932/article/details/88970510" \l "_Toc533170026)（个人信息管理系统的设计与实现）
12. <https://learnku.com/laravel/wikis>（Laravel中国）

致谢

首先衷心地感谢庄华老师。给予了很多的指导与帮助。本次毕业设计，从选题到论文撰写，给予我很多宝贵的意见。庄老师和蔼可亲、博学多才，瑰玮倜傥，给我留下了深刻的印象；严谨的治学态度，从严从高的理论要求，无时无刻鞭策和鼓舞着我，潜移默化地影响着我。在此向庄华老师表示崇高的敬意和衷心的感谢。

感谢各位同学。从毕业设计到毕业论文的撰写，各位同学都给了我很大的帮助，在平时的讨论中给我提出了许多宝贵的意见和建议。同学们的思维敏捷、认真负责的工作学习态度都给了我很深的影响。在此，向各位同学表示衷心的感谢和诚挚的祝愿。

感谢四年以来各位老师们的倾情授课，让我们的知识和视野得到了很大的拓展，同时，提升了我们发现问题、解决问题的思维与能力，让我的思维有了质的飞跃。

感谢开源组织和社区，提供了诸多的开源框架和类库，给我提供了许多的思路上的引导和编码提示，让我在程序开发中得以快速实现功能需求。最后，感谢一直关心我的亲朋好友。

**Design and Implementation of Web-based Personal Information Fusion System**

【**Abstract**】 Personal information management is to manage personal information with information technology to help individuals improve work efficiency, integrate all personal information resources, and improve individual competitiveness. It is a new management concept and method. It can turn all kinds of materials owned by individuals and readily available information into more valuable knowledge, which will ultimately benefit their work and life. Through the integration of personal information, people can raise Be a good learning habit, enhance information literacy, improve your professional knowledge system, improve your ability and competitiveness, and lay a solid foundation for achieving personal value and sustainable development. This paper mainly describes the design and implementation of personal information fusion system on the Web using Lavaral technology. The system integrates personal or family-owned information (including family and life, work and study, and personal or family health information) into a system. As long as the personal information fusion system is used, functions such as query, management and analysis can be realized. Using this system can effectively use personal information in the information age to improve the quality and efficiency of life, work and learning.

【**Keywords**】personal information management, personal information fusion, Web, Laravel