

**计算机与信息技术学院**

**课程设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 课程名称 | 软件系统设计 |  |
|  | 设计名称 | AI智能省钱助手 |  |
|  | 专业班级 | 计科2101 |  |
|  | 学 号 | 20851147 |  |
|  | 姓 名 | 卢梦豪 |  |
|  | 任课教师 | 田晋坤 |  |
|  | 日 期 | 2024-06-27 |  |

**《软件系统设计》课程考核说明**

**一、考试对象**

2021级计科2101-2106班

**二、考核方式**

课程设计报告 + 答辩。

**三、考试内容**

基于大语言模型（LLM）完成一个智能化系统，使用的技术包括但不限于Python、Langchain等。其中，大语言模型可以采用现有的第三方服务商提供的模型（如：通义千问、千帆等），也可以使用开源模型。

**四、技术要求**

1、要求使用Python、Langchain框架。

2、项目文件命名：学生姓名的拼音首字母-网站英文名称，名称与多个英文单词之间用“-”连接，如李明的手机销售网站命名为：lm-phone-seller

3、单文件组件命名要符合Pascal命名规范，变量、文件及文件夹命名要符合规范，尽量做到见名知意。

**五、提交物**

项目源码、设计说明书电子版、设计说明书纸质打印版、答辩纪录。

**六、期末成绩构成**

由项目成绩和答辩成绩组成，其中项目成绩60分，答辩成绩40分

项目成绩评分标准如下：

|  |  |
| --- | --- |
| **综合/单项类评分** | **分值** |
| **1 提交物** | **10** |
| 1.1 内容完整性 | 5 |
| 1.2 目录结构 | 5 |
| **2 代码** | **20** |
| 2.1 项目、代码命名规范 | 5 |
| 2.2 代码编写规范性 | 5 |
| 2.3 代码可读性 | 5 |
| 2.4 技术全面 | 5 |
| **3 程序演示** | **30** |
| 3.1 结果正确，功能完整 | 15 |
| 3.2 页面设计美观 | 15 |

**《软件系统设计》课程考核成绩**

|  |  |
| --- | --- |
| 项目成绩（60分） |  |
| 答辩成绩（40分） |  |
| 期末总成绩 |  |
| **答辩评语：**  提交物内容（完整、较完整、基本完整、不完整），目录结构（正确、较正确、基本正确、不正确）；项目、代码命名（符合规范、较符合规范、基本符合规范、不规范），代码注释（完整、较完整、基本完整、不完整），可读性（好、较好、尚可、差）；技术应用方面（全面、较全面、基本合格、有欠缺）；程序功能（完整、较完整、基本完整、不完整），页面（美观、较美观、设计尚可、不美观），演示过程（顺畅、较顺畅、基本顺畅、不顺畅），问题回答（正确、较正确、基本正确、错误较多），思路（清晰、较清晰、基本清晰、不清晰），概念（清楚、较清楚、基本清楚、不清楚）。 | |

**目 录**

[第1章 产品需求 1](#_Toc170397075)

[1.1 背景和目的 1](#_Toc170397076)

[1.1.1 产品背景 1](#_Toc170397077)

[1.1.2 产品目标 1](#_Toc170397078)

[1.2 用户需求和系统功能 2](#_Toc170397079)

[1.3 产品设计和页面 3](#_Toc170397080)

[1.3.1 登录窗口 3](#_Toc170397081)

[1.3.2 AI问答窗口 3](#_Toc170397082)

[1.4 项目资源和进度安排 4](#_Toc170397083)

[第2章 概要设计 5](#_Toc170397084)

[2.1 架构设计 5](#_Toc170397085)

[2.1.1 功能架构 5](#_Toc170397086)

[2.1.2 技术架构 5](#_Toc170397087)

[2.2 功能设计 6](#_Toc170397088)

[2.2.1 界面功能设计 6](#_Toc170397089)

[2.3 模型设计 7](#_Toc170397090)

[2.3.1 概念模型 7](#_Toc170397091)

[2.3.2 逻辑模型 8](#_Toc170397092)

[2.3.3 物理模型 9](#_Toc170397093)

[2.4 接口设计 9](#_Toc170397094)

[2.4.1 整体接口设计 9](#_Toc170397095)

[第3章 系统测试 12](#_Toc170397096)

[3.1 测试范围 12](#_Toc170397097)

[3.2 测试用例 13](#_Toc170397098)

[第4章 总结 15](#_Toc170397099)

[附件：答辩纪录 16](#_Toc170397100)

# 第1章 产品需求

## 1.1 背景和目的

### 1.1.1 产品背景

随着人工智能技术的不断发展和普及，智能助手在生活和工作中的应用越来越广泛。特别是在个人消费领域，消费者对于如何更加高效地管理自己的财务、寻找性价比更高的商品或服务的需求持续增长。

现代消费者越来越注重商品和服务的性价比，他们更倾向于通过智能工具来辅助自己做出更明智的消费决策。同时，消费者对于个性化、定制化的服务需求也在不断增加。

自然语言处理、数据挖掘等先进技术的发展，为AI智能省钱助手提供了强大的技术支持。这些技术使得AI能够更准确地理解消费者的需求，提供更精准、更个性化的服务。

### 1.1.2 产品目标

本产品的主要目标是设计并实现一个基于SpringBoot的考场管理系统，以解决当前考场管理过程中存在的效率低、资源浪费等问题。具体商业目标包括：

1. **提高消费者购物效率**

* AI智能省钱助手能够快速地帮助消费者搜索、比较不同商品或服务的价格、性能等信息，提供最优的购买建议，从而提高消费者的购物效率。

1. **降低消费者购物成本**

* 通过智能分析消费者的购物习惯和需求，AI智能省钱助手能够为消费者推荐性价比更高的商品或服务，帮助消费者节省购物成本。

1. **提供个性化服务**

* AI智能省钱助手能够根据消费者的个人喜好、需求等信息进行个性化定制，为消费者提供专属的服务体验。

1. **推动人工智能技术应用**

* 通过AI智能省钱助手项目的开发和应用，进一步推动人工智能技术在个人消费领域的应用和发展，促进相关产业的升级和转型。

综上所述，AI智能省钱助手项目的产品背景是基于市场需求增长、消费者行为变化和技术支持提升等多个方面的考虑，而产品目的是通过提供智能、高效、个性化的服务，帮助消费者提高购物效率、降低购物成本，同时推动人工智能技术的应用和发展。

## 1.2 用户需求和系统功能

用户作为主要参与者，与系统进行交互。首先输入他们考虑购买的产品信息。系统接收这些信息，并使用内置的人工智能模型来生成抑制购买的理由。用户可以查看这些理由，并评估它们以决定是否继续购买。系统还提供自然语言处理服务，以改善输出的表达方式，使其更加自然和易于理解。省钱助手系统用例说明表如图1-1所示。

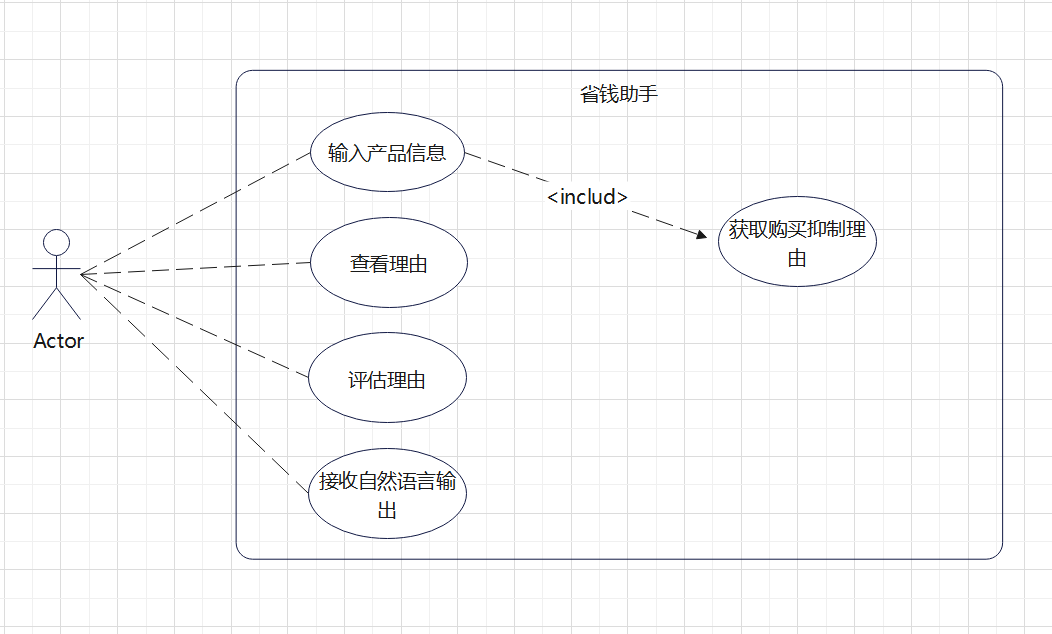


图1-1 系统用例图

## 1.3 产品设计和页面

### 1.3.1 登录窗口

用户在运行项目之后会跳转到前端登录界面，在登录界面用户可以输入用户名和密码进行登录，登录之后会跳转到具体功能界面。如下图1-2所示。

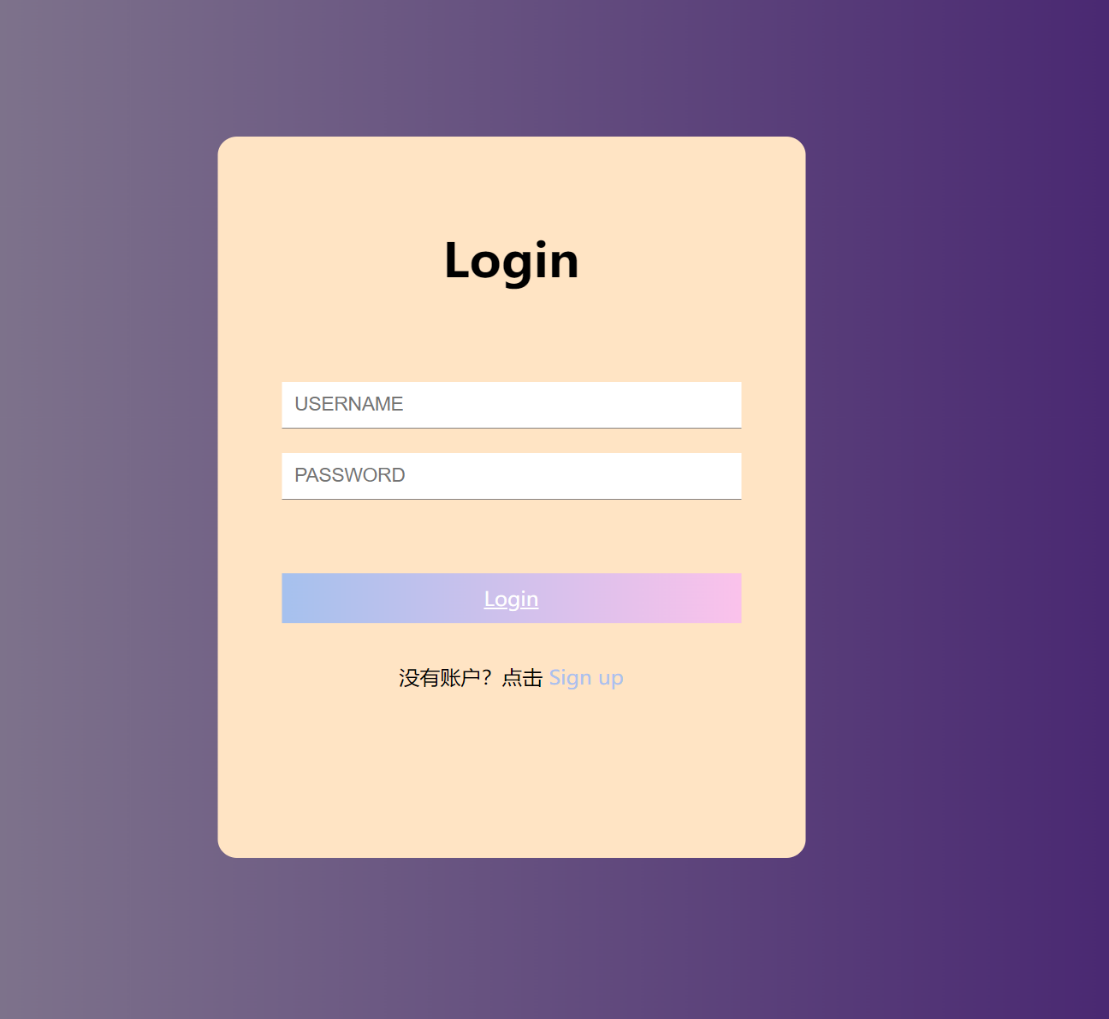


图1-2 用户登录界面

### 1.3.2 AI问答窗口

用户在登录界面进行登录之后会跳转到AI问答界面，在问答界面中用户可以在问答窗口中输入自己想要购买的物品，点击确定后等待几秒之后AI会在下方的回答窗口中显示出所给的出的回答。具体如下图1-3所示。



图1-3 用户管理界面展示图

## 1.4 项目资源和进度安排

本项目的主要任务包含：调用AI，使用langchain框架组件实现自定义对话。、前端登录跳转，界面设计，前后信息交互。如下图1-4所示。

参与本项目的人员包括：杨晨，卢梦豪。

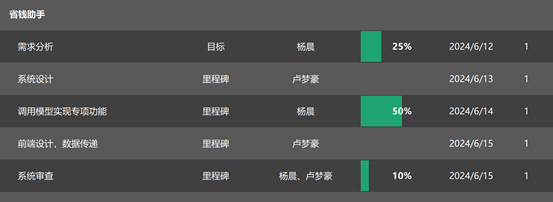


图1-4 进度甘特图

# 第2章 概要设计

## 2.1 架构设计

### 2.1.1 功能架构

省钱助手问答机器人项目的功能架构是一个集成了用户界面、后端逻辑和人工智能处理的系统。它通过一个简洁的用户界面接收用户输入的产品信息，然后通过后端将这些信息传递给AI模型，该模型生成并优化购买抑制理由。优化后的理由被反馈到用户界面供用户参考，以帮助他们重新考虑购买决策。整个过程旨在提供流畅、直观且互动性强的用户体验。而通过功能架构图，清晰的展现出系统要完成的任务。功能架构图具体设计如图2-1所示。

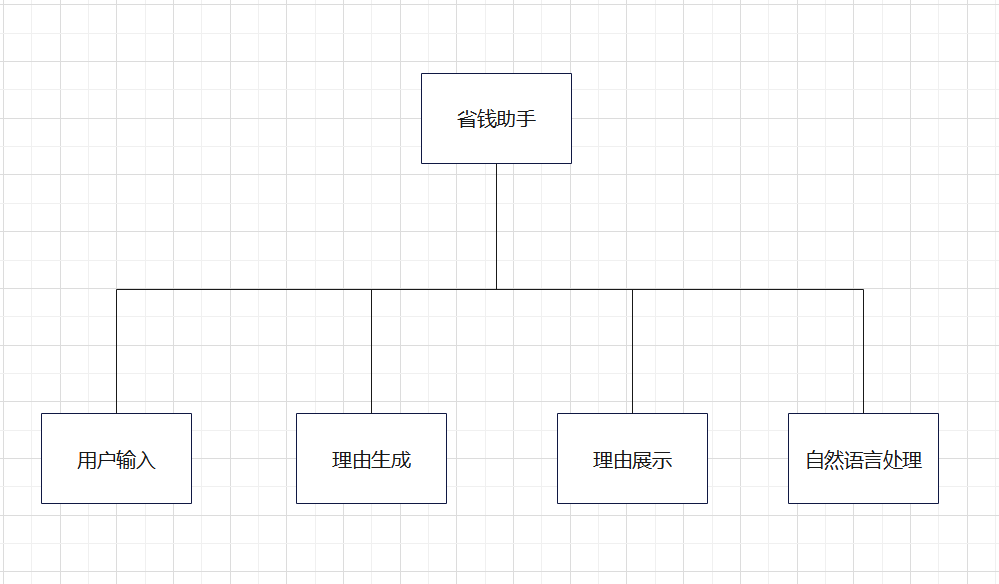


图2-1 功能架构设计图

### 2.1.2 技术架构

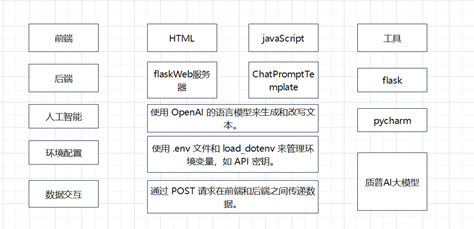


图2-2 技术架构设计图

## 2.2 功能设计

### 2.2.1 界面功能设计

在进行前端代码编写以及前端和后端所连接具体功能时，首先要先进行简单的需求分析来确定登录界面、页面跳转逻辑、界面美化以及前后端交互这些细节。

登录界面设计使用HTML和CSS构建登录表单，包括用户民、密码输入框、登录按钮等。使用JavaScript来编写处理登录按钮的点击事件，根据后端传来的响应来决定下一步的操作。后端中使用Python Web框架Flask来接收前端所发送的登录请求，定义路由和对应处理函数。在处理函数中后端进行对前端传输进来的登录信息进行匹配，验证成功后返回给前端。前端通过AJAX请求将登录表单数据发送到后端中，再继续进行接下来的操作。具体流程如下图2-3所示。

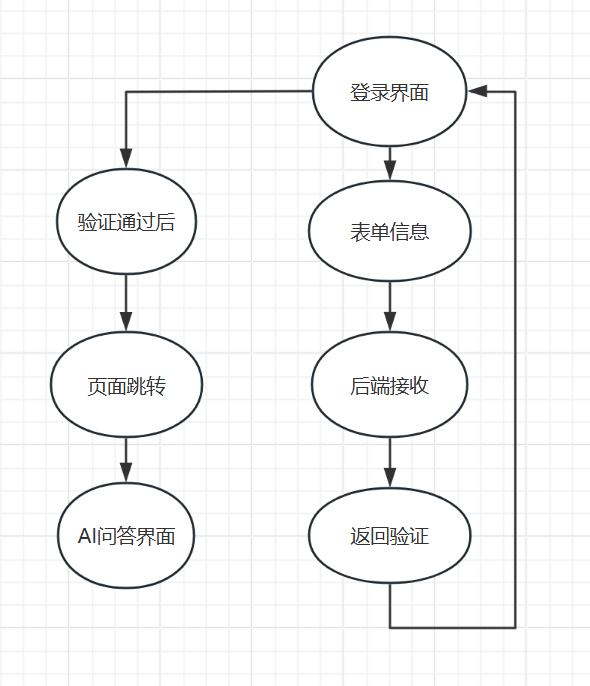


图2-3 用户角色分配活动图

## 2.3 模型设计

### 2.3.1 概念模型

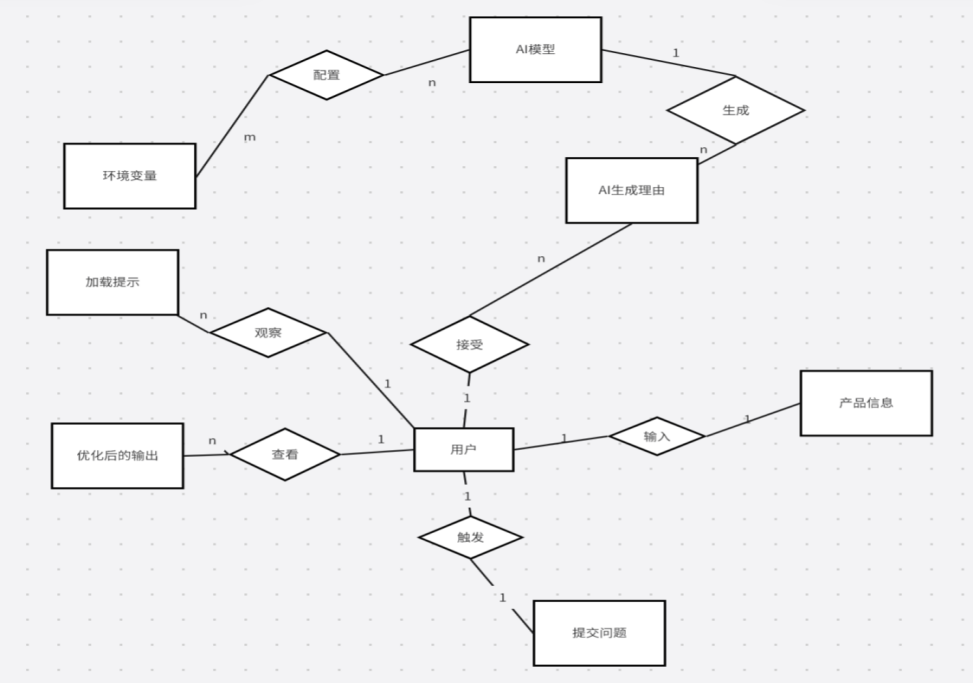


图2-4 E-R图

**关系和多重性：**

* 用户与产品信息：

关系：输入产品信息

多重性：一个用户可以输入多个产品信息，但每个产品信息只能由一个用户输入。

* 用户与提交查询：

关系：提交

多重性：一个用户可以提交多个查询。

* 用户与AI生成理由：

关系：接收理由

多重性：一个用户可以接收多个AI生成理由。

* 用户与优化后的输出：

关系：查看输出

多重性：一个用户可以查看多个优化后的输出。

* 提交查询与AI生成理由：

关系：生成

多重性：一个提交可以生成多个AI理由。

* AI生成理由与优化后的输出：

关系：优化

多重性：一个AI理由可以被优化为一个优化后的输出。

* 环境变量与AI模型：

关系：配置

多重性：多个环境变量可以配置一个AI模型。

* AI模型与AI生成理由：

关系：生成理由

多重性：一个AI模型可以生成多个AI理由。

### 2.3.2 逻辑模型

为省钱助手项目设计逻辑模型，需要考虑数据结构、系统组件以及它们之间的交互。以下是设计逻辑模型如下图2-5：

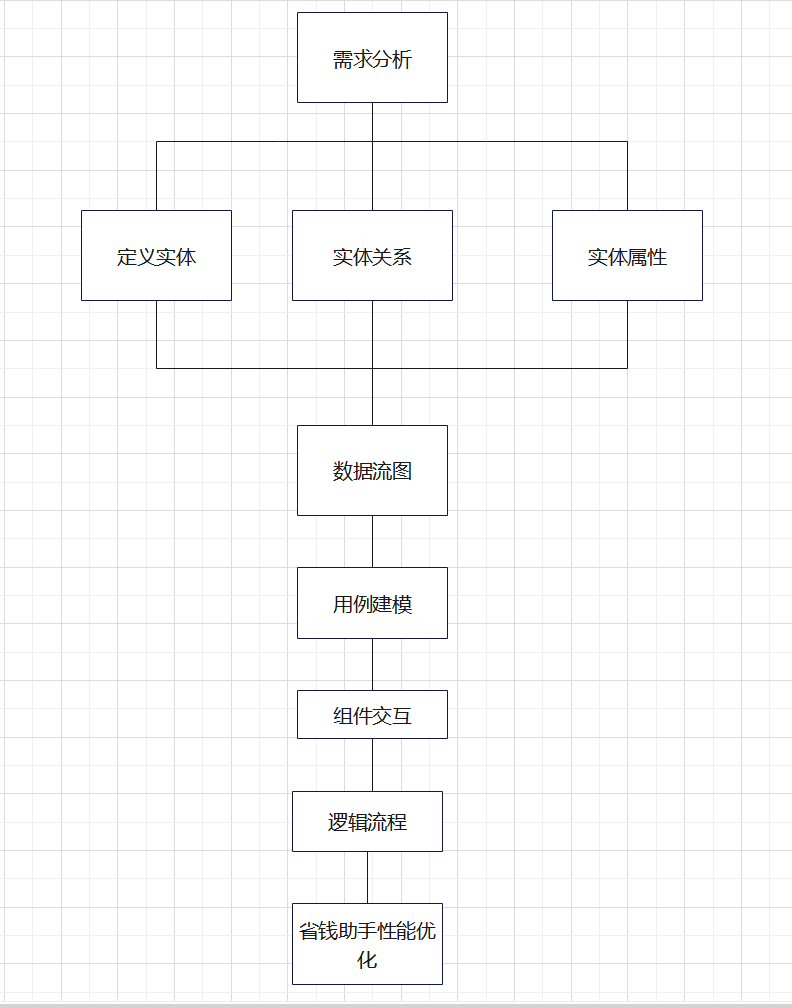


图2-5 逻辑模型图

### 2.3.3 物理模型

● 设计原则：用户体验优先、一致性、响应式。

● 表单元素：包含用户名、密码输入框，可能还有验证码连接。

● 提示信息：显示用户输入时的实时验证信息（如密码强度、格式错误等）。

● 跳转逻辑：登录成功跳转指定页面，失败显示错误信息。

● 界面美化：色彩选择、字体与排版、图标与图片。

● RESTful API：使用RESTful风格设计API，确保URL结构清晰、易于理解。

● HTTP方法：使用GET、POST、PUT、DELETE等HTTP方法来处理不同的请求。

● 响应处理：解析后端返回的JSON数据，并根据状态码和消息进行相应的处理。

● Web框架：使用如Flask Python Web框架来处理前端请求。

● 路由处理：定义路由函数来处理不同的URL请求。

● 业务逻辑处理：根据业务需求执行相应的数据库操作或其他逻辑处理。

● 响应返回：将处理结果转换为JSON格式并返回给前端。

## 2.4 接口设计

### 2.4.1 整体接口设计

由省钱助手项目的前端和后端代码，我们设计了一套详细的接口规范。接口设计通常包括请求的路径、支持的操作、请求和响应的数据格式等。以下是省钱助手项目可能的接口设计：

1. 首页接口

路径：/

方法：GET

描述：展示省钱助手的首页，包括用户输入界面。

请求参数：无

响应：成功：HTML页面，包含表单和加载提示。

失败：错误页面。

2. 提交产品信息接口

路径：/

方法：POST

描述：接收用户输入的产品名称，并请求AI生成抑制购买欲望的理由。

请求参数：product：用户输入的产品名称，类型为string。

响应：成功：HTML页面，包含生成的理由。

失败：错误信息。

3. AI理由生成接口

路径：/generate-reasons

方法：POST

描述：后端接口，接收产品名称并调用AI模型生成理由。

请求体：{

"product": "string"

}

响应：{

"success": true,

"content": "string" // AI生成的理由文本

}

或{

"success": false,

"error": "string" // 错误信息

}

4. 自然语言处理接口

路径：/nlp-process

方法：POST

描述：接收AI生成的原始理由文本，返回更自然、流畅的文本。

请求体：{

"text": "string"

}

响应：{

"success": true,

"content": "string" // 自然语言处理后的文本

}

或复制{

"success": false,

"error": "string" // 错误信息

}

5. 清理输入接口

路径：/clear-input

方法：POST

描述：清空用户输入的产品名称，通常在表单提交后调用。

请求参数：无

响应：成功：无内容或重定向。

失败：错误信息。

# 第3章 系统测试

## 3.1 测试范围

1. 功能测试

* 用户输入验证：测试输入字段是否接受有效和无效的输入。验证必填项是否在表单提交时进行检查。
* 表单提交：测试表单在正常和异常情况下的提交行为。验证提交后数据是否正确发送到服务器。
* AI理由生成：测试AI模型是否根据用户输入生成合理的购买抑制理由。检查生成理由的准确性和相关性。
* 自然语言处理：验证输出文本是否经过自然语言处理，提高可读性。
* 结果展示：测试结果页面是否正确显示生成的理由。验证在不同条件下（如无理由生成时）的用户界面表现。
* 用户界面交互：测试按钮、输入框等元素的交互是否符合预期。

2. 非功能测试

* 性能测试：测试应用在高负载下的表现，包括响应时间和服务器处理能力。
* 安全性测试：验证数据传输安全性，检查是否有适当的加密措施。
* 可用性测试：确保应用的易用性，包括导航、布局和用户指导。
* 兼容性测试：测试应用在不同浏览器和设备上的表现。
* 可访问性测试：确保应用符合可访问性标准，使所有用户都能使用。

3. 集成测试

* 测试前端和后端的集成是否无缝。验证外部API（如OpenAI服务）与应用的集成情况。

4. 系统测试

* 测试整个应用作为一个系统运行时的表现。验证所有组件作为一个整体协同工作的结果。

5. 回归测试

* 在每次更新后执行，确保新代码没有破坏现有功能。

6. 用户接受测试（UAT）

* 让实际用户测试应用，收集反馈以验证是否满足用户需求。

## 3.2 测试用例

基于上述测试范围，以下是省钱助手项目的测试用例：

1.用户登录页面登录测试用例：

用例ID：TIU\_01

目的：是否可以正常进行登录操作。

前提条件：有效用户正常

步骤：

打开登录页面。

在用户名和密码字段中输入有效用户的凭证。

点击“登录”按钮。

期望结果：

页面跳转到用户指定的页面（如首页或用户中心）。

后端返回登录成功的状态码（如200）。

2.无效用户名/密码测试用例

用例ID：TIU\_02

前置条件：无。

步骤：

打开登录页面。

在用户名和密码字段中输入无效的用户名和密码组合。

点击“登录”按钮。

期望结果：

页面停留在登录页面，并显示错误消息（如“用户名或密码错误”）。

后端返回登录失败的状态码（如401或自定义的错误状态码）。

3. 空字段提交

用例ID：TIU\_03

前置条件：无。

步骤：

打开登录页面。

不填写任何字段。

点击“登录”按钮。

期望结果：

页面停留在登录页面，并显示错误消息（如“请填写用户名和密码”）。

后端返回验证失败的状态码（如400）。

4. 安全性测试用例

用例ID：TSEC\_01

目的：验证数据传输的安全性。

前提条件：用户正在通过HTTPS提交表单。

测试步骤：

1. 提交包含敏感信息的表单。

2. 验证数据是否通过加密传输。

预期结果：所有数据应通过加密通道安全传输。

5. 性能测试用例

用例ID：TPERF\_01

目的：测试应用的响应时间。

前提条件：系统在正常运行状态。

测试步骤：

1. 测量提交表单到显示结果的总时间。

2. 在高并发条件下重复测试。

预期结果：系统应在预定时间内响应用户请求，即使在高负载下。

注意事项：

确保前端代码能够正确处理后端返回的各种状态码和响应数据。

在后端Flask应用中添加适当的错误处理和安全性措施（如防止SQL注入、使用HTTPS等）。

在测试用例中考虑各种边界条件和异常情况，以确保系统的健壮性和稳定性。

# 第4章 总结

经过本次《企业级应用开发》课程的学习，在开发省钱助手项目的过程中，我深刻体会到了从概念到实现的全过程。项目始于对用户需求的深入分析，明确了开发一个旨在帮助用户抑制冲动购买的智能助手的目标。

本项目旨在开发一个功能完善的前端登录系统，包括用户登录界面的设计、页面跳转功能的实现、前端页面的设计与美化，以及前端代码与后端Python Web Flask框架的信息交互。通过此项目，我们深入理解了前端与后端协同工作的机制，并掌握了相关技术的实际应用。在登录界面设计方面，我们采用了简洁明了的风格，确保用户能够快速理解并操作。界面包含用户名、密码输入框以及登录按钮。当用户成功登录后，前端页面需要跳转到指定的页面。通过表单的action属性进行页面跳转，可以在不刷新页面的情况下实现数据交互和页面跳转。在前端页面设计与美化方面，我们注重页面的整体布局和用户体验。采用了CSS框架（来快速搭建页面布局，并使用CSS样式和图片来美化页面。我们遵循了设计的一致性原则，确保页面风格统一、美观大方。同时，我们还考虑了页面的可访问性和响应式布局，使页面能够在不同设备和浏览器上正常显示。在前端代码与后端Flask信息交互方面，我们采用了HTTP请求和响应的方式来实现。前端通过发送HTTP请求将用户输入的数据发送到后端Flask应用；后端Flask应用接收到请求后，对请求数据进行处理，并将处理结果以HTTP响应的形式返回给前端。前端接收到响应后，根据响应数据更新页面内容或执行其他操作。

通过本项目的开发，我们深入了解了前端登录系统的开发流程和关键技术。在未来的工作中，我们将继续探索和学习新的技术和方法，不断提高自己的开发能力。同时，我们也希望能够将本项目中的经验和教训应用到以后其他的工作项目中去。

# 附件：答辩纪录

**《软件系统设计》课程答辩记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 专业班级 | 计科2101 | 姓名 | 卢梦豪 | 学号 | 21851147 |
| 答辩时间 | 2024-06-17 11:00-12:00 | | 答辩地点 | 思源楼424 | |
| 答辩题目 | 软件系统设计 | | | | |
| 答辩老师 | 田晋坤 | | | | |
| 提问及答辩记录：  1. 如何实现前端代码与后端Python Web Flask框架的信息交互？  答：实现前端代码与后端Python Web Flask框架的信息交互主要通过HTTP请求和响应来完成。前端可以使用JavaScript来发送HTTP请求（如GET、POST等），并将请求数据发送到后端Flask应用。后端Flask应用接收到请求后，会对请求数据进行处理，并将处理结果以HTTP响应的形式返回给前端。前端接收到响应后，可以根据响应数据更新页面内容或执行其他操作。在此过程中，还需要注意数据的安全性和完整性，可以采用HTTPS协议来加密传输的数据，并使用适当的验证和过滤机制来防止恶意攻击。  2.在用户成功登录后，如何实现页面跳转？  答：可以通过在服务器端设置session来保存用户的登录状态。当用户成功登录后，服务器会生成一个唯一的session ID，并将其返回给前端。前端在每次请求时，都将这个session ID作为请求头的一部分发送给服务器，服务器验证通过后，即可实现页面跳转。 | | | | | |