**基于大语言模型的**

**智能学习推荐管理系统**

**需**

**求**

**规**

**格**

**说**

**明**

**课程：软件工程实践**

**班级：公信21**

**成员：李欣桦；李傲；吴林昊；杜威；刘心彤**

**目录**

[1引言 1](#_Toc25274)

[1.1编写目的 1](#_Toc11180)

[1.2背景 1](#_Toc30177)

[1.3术语和缩略词 1](#_Toc31890)

[1.4参考资料 1](#_Toc31549)

[2任务概述 2](#_Toc16116)

[2.1项目概述 2](#_Toc3286)

[2.1.1项目来源及背景 2](#_Toc3979)

[2.1.2项目目标 3](#_Toc23566)

[2.1.3系统功能概述 3](#_Toc27541)

[2.2用户特点 3](#_Toc17664)

[2.3假定和约束 4](#_Toc11055)

[3功能需求 4](#_Toc16897)

[3.1功能划分 4](#_Toc32607)

[3.1.1系统功能组成 4](#_Toc26617)

[3.1.2功能编号和优先级 5](#_Toc6524)

[3.2功能描述 6](#_Toc6829)

[3.2.1用户登录 6](#_Toc3805)

[3.2.2任务设置 7](#_Toc24626)

[3.2.3日常学习 7](#_Toc25331)

[3.2.4作业测试 9](#_Toc13600)

[4数据需求 10](#_Toc13112)

[4.1静态数据 10](#_Toc3907)

[4.2动态数据 10](#_Toc7069)

[4.3数据字典 11](#_Toc27346)

[4.4数据库描述 13](#_Toc24409)

[5性能需求 15](#_Toc24191)

[5.1数据精度 15](#_Toc332)

[5.2时间特性 15](#_Toc21720)

[5.3灵活性 15](#_Toc4174)

[6运行需求 15](#_Toc15820)

[6.1用户界面 15](#_Toc20259)

[6.2软件接口 16](#_Toc24922)

[6.3硬件接口 16](#_Toc23791)

[7其他需求 17](#_Toc1202)

[7.1验收标准 17](#_Toc27224)

[7.2质量标准 18](#_Toc23879)

# 1引言

## 1.1编写目的

该架构文档对基于大语言模型的智能学习推荐管理系统整体结构进行描述，LearnLark的整体架构主要分为用户数据模块、日常学习模块、测试模块三个部分。其中日常学习模块、测试模块主要涉及大语言模型、Langchain等技术，在整个架构中占据中枢位置；用户数据模块主要包含可视化系统与数据分析整理系统。

## 1.2背景

文档的组织结构如下：

第一部分 引言

本部分主要概述了文档内容组织结构，使读者能够对文档内容进行整体了解，并快速找到自己感兴趣的内容。同时，提供了提供了架构元素的索引，同时包括了术语表、缩略语表。

第二部分 任务概述、功能需求

本部分主要介绍了软件架构，向读者提供系统概览，建立开发的相关上下文和目标。分析需求与架构所考虑的约束和影响，并介绍了架构中所使用的主要设计方法，

第三部分 数据需求、性能需求、运行需求

描述系统数据和性能属性需求、运行需求与架构之间的映射关系，包括架构评估和验证等。

第四部分 附录

## 1.3术语和缩略词

本文档未涉及需要额外说明的术语与缩略词。

## 1.4参考资料

（1）国家标准：

中华人民共和国国家质量监督检验检疫总局, 中国国家标准化管理委员会. (2016). GB/T 16260.1-2016 软件工程 质量要求 第1部分：质量模型[S]. 北京: 中国标准出版社.

这份标准提供了软件产品质量模型，对于明确需求规格说明中的质量要求非常关键。

（2）软件工程经典书籍

Pressman, R. S. (2021). 软件工程：实践者的研究方法[M]. 北京: 清华大学出版社.

本书是软件工程领域的经典教材，对需求分析、规格说明编写等方面有深入的论述和指导。

（3）项目管理标准：

项目管理协会(PMI). (2015). PMBOK指南[R]. 北京: 电子工业出版社.

尽管主要聚焦于项目管理，但其中关于需求管理和文档编制的内容对编写需求规格说明同样适用。

（4）信息技术文献：

ISO/IEC. (2011). ISO/IEC 25010:2011 系统和软件工程 系统和软件质量要求和评价(SQuaRE) 第1部分：系统和软件质量模型[S]. 国际标准化组织/国际电工委员会.

这个标准定义了系统和软件的质量模型，有助于在需求规格说明中建立质量基准。

（5）学术论文：

Boehm, B. W., & Turner, R. (2015). Balancing agility and discipline: A guide for the perplexed[M/OL]. Boston, MA: Addison-Wesley.

论述了如何在软件开发的敏捷性和规范性之间取得平衡，对理解如何灵活且严谨地编写需求规格说明有重要启示。

# 2任务概述

## 2.1项目概述

### 2.1.1项目来源及背景

LearnLark是一款针对学生的答题软件，大家可以在这里进行答题、测试，对自己的学习数据进行分析与复盘。当今市场上大部分的答题软件都不具备大语言模型的调用能力，同时其不仅仅是为用户提供答题服务，还掺杂了一些诸如推送广告、推销等的附带功能，所以我们推出LearnLark，主打答题学习，无其他附带功能。

### 2.1.2项目目标

LearnLark 的主要目标是创建一个纯净、高效的在线学习环境，专注于提升学生的学习成效与自我检测能力。具体目标包括：

* 提升学习效率：通过智能化的题型推荐系统，根据用户的学习进度和掌握程度动态调整难度，确保每位用户都能在适合自己的挑战水平下学习，从而最大化学习效率。
* 个性化学习体验：利用大语言模型的智能分析，为每位用户生成个性化的学习路径和复习计划，满足不同学习风格和需求。
* 数据驱动进步：收集并分析用户的学习数据，提供详尽的学习报告，帮助用户识别强项和弱点，实现精准复盘和策略调整。
* 无干扰环境：保持平台的纯净性，不包含任何广告或无关的推销内容，确保用户能够全神贯注于学习过程。
* 易用性与可访问性：设计简洁直观的用户界面，确保所有年龄段的学生都能轻松上手，同时考虑特殊需求群体的使用便利性。

### 2.1.3系统功能概述

* 知识学习：采取反复答题的形式，类似背单词软件。答题分多次，第一次调用题库里的题，根据对错，使用大模型生成或者推荐类似的更难/更简单的题，三次都答对后通过，记录到数据库中。
* 知识测试：一次性答题，表单。使用的题目可以是由模型根据用户答题情况推荐的题库中的题目。答完后可以显示结果，并记录在数据库中。
* 任务制定：用户自主选择，可以选择学习类型、学习量，根据科学预置模型制定。
* 学习记录：个人界面，呈现个人学习记录分析。

## 2.2用户特点

LearnLark 的目标用户主要是学生群体，涵盖从小学到高等教育各个阶段，具有以下特点：

* 学习需求多样化：不同年龄、不同学科、不同学习阶段的学生有着不同的学习需求和目标。
* 追求高效与便捷：现代学生生活节奏快，倾向于使用能快速获取知识、节省时间的学习工具。
* 偏好个性化服务：期待定制化的学习方案，以适应个人的学习习惯和能力水平。
* 社交化学习倾向：喜欢通过社交媒体和网络平台与他人互动交流，共同进步。
* 对新技术接受度高：作为数字原住民，乐于尝试并依赖于技术辅助学习。

## 2.3假定和约束

1. 人力和时间的约束：本软件开发过程中需要考虑到人力和时间的约束，相较于一般软件的开发团队来说人员较少时间较短。

(2) 技术发展的约束：计算机技术和发展的日新月异，将会给信息处理带来更多手段，同时也会带来更加丰富的信息表达形式，例如日新月异的更新的大语言模型等等，这就要求软件在设计时要考虑技术变化的可能性，为可能的变化预留一定的处理能力。

(3) 数据安全与隐私保护：在设计与实施过程中，严格遵守相关法律法规，确保用户数据的安全性和隐私保护，这是软件开发不可妥协的前提。

(4) 跨平台兼容性：考虑到用户可能使用多种设备（如手机、平板、电脑）进行学习，系统需保证在不同操作系统和浏览器上的良好兼容性和用户体验。

(5) 可持续发展：设计时需考虑系统的扩展性和维护性，确保未来能够随着用户基数的增长和技术的进步进行有效升级和优化。

# 3功能需求

## 3.1功能划分

### 3.1.1系统功能组成

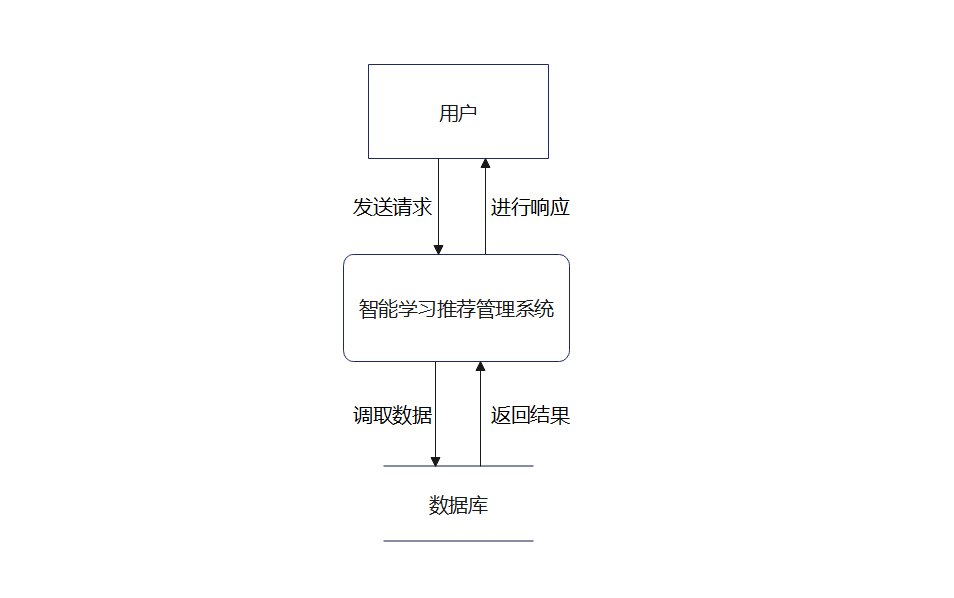


图 1 智能学习推荐管理系统顶层数据流图

如图1所示，用户在登陆后可以向智能学习推荐管理系统发送相关的请求，并能够接收智能学习推荐管理系统给出的响应。智能学习推荐管理系统可以向数据库发起请求调取数据并获得结果，再向用户进行反馈。

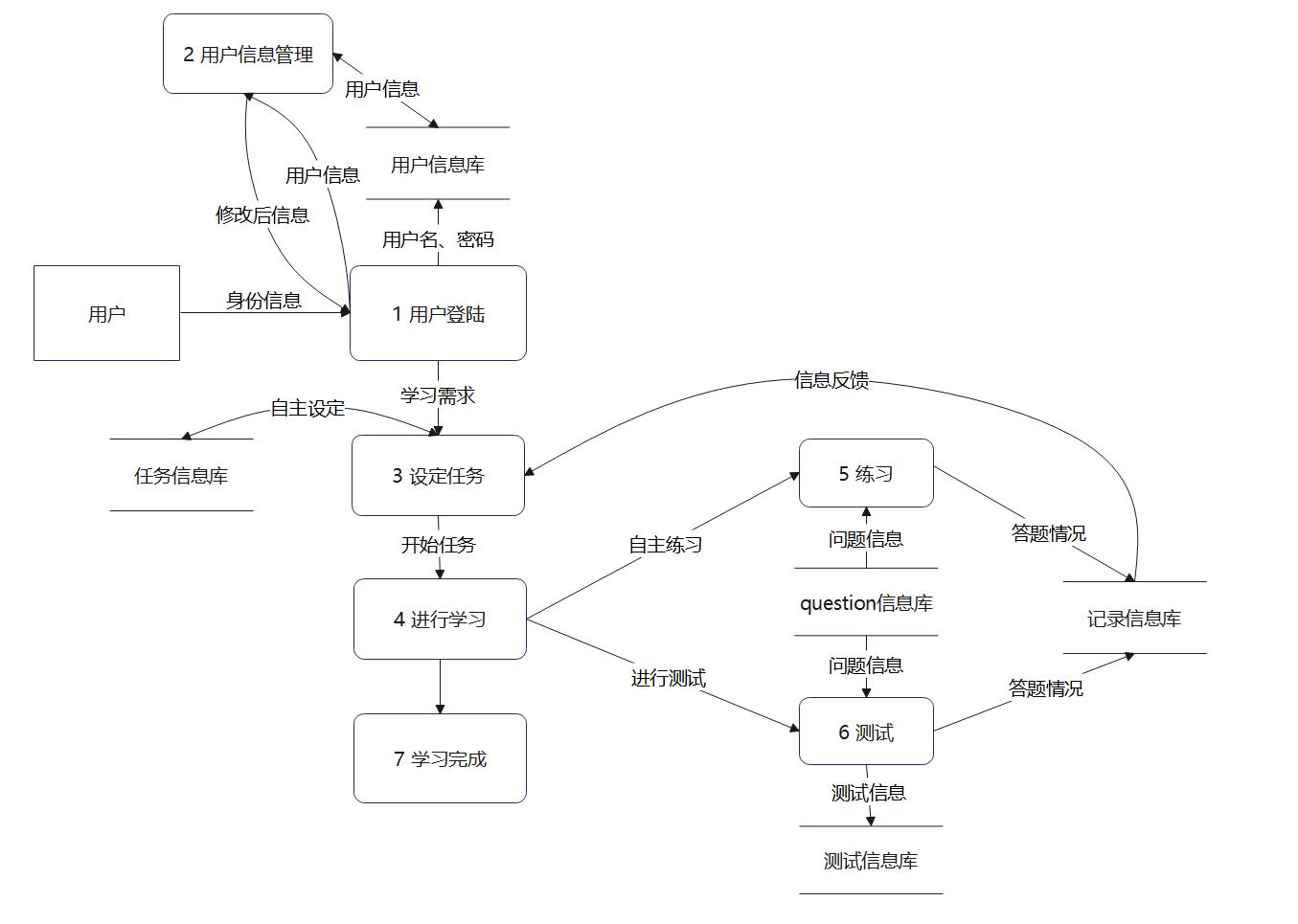


图 2 智能学习推荐管理系统0层数据流图

如图2所示，用户可以通过提交身份信息向用户登陆事务发送请求。用户登陆事务从用户信息库中读取相应的用户信息进行匹配判断登录结果，同时用户可以在登陆时向用户信息管理事务发送请求，修改后的信息可以同步到用户信息库。登陆完成后可以进行任务的设定并与任务信息库进行同步，下次登陆可以直接选取任务或创建新的任务。任务设定完成后可以进行学习，用户可以进行练习操作或测试操作。在自主练习时，练习事务从question信息库调取匹配的问题送给用户，并将答题情况存入记录数据库。在进行测试时，测试事务从question信息库选取题目组合出相应测试送给用户，将测试信息与测试分数存入测试信息库并将答题情况存入记录信息库。记录信息库可以向设定任务事务提供反馈作为用户设置学习任务的参考。在设定的学习任务全部完成后，系统进行学习完成提示。

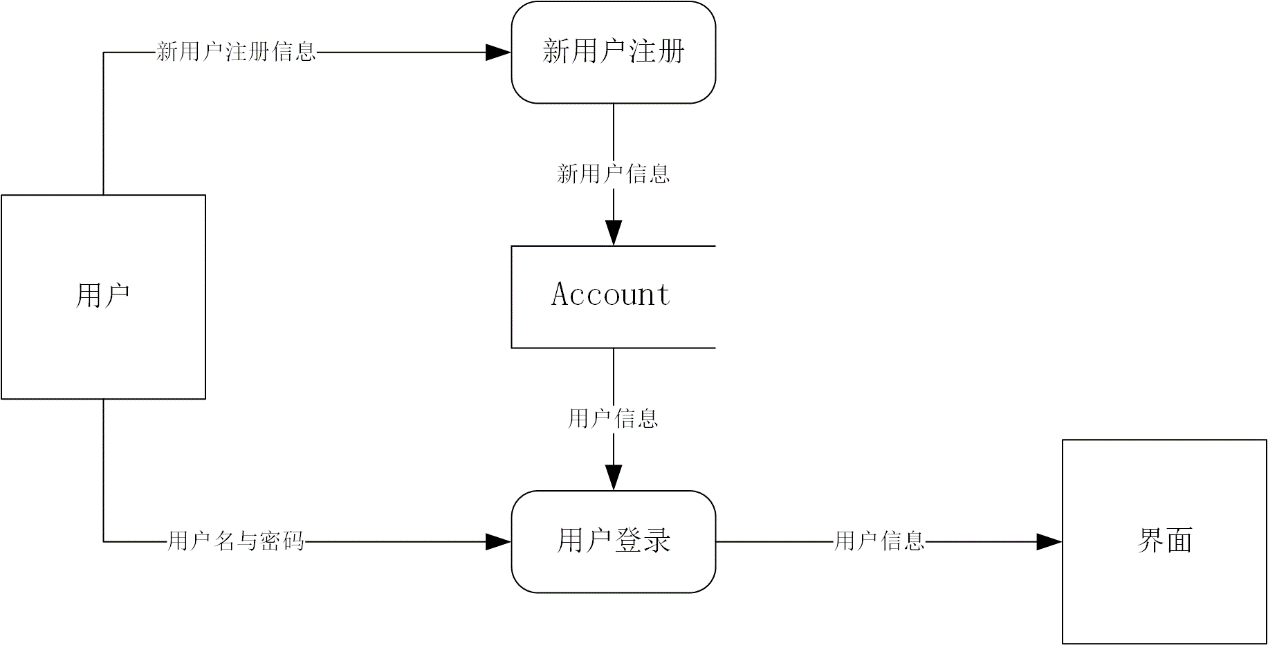
### 3.1.2功能编号和优先级

|  |  |  |
| --- | --- | --- |
| **功能编号** | **功能** | **优先级** |
| 1 | 用户登录 | 高 |
| 2 | 用户信息管理 | 中 |
| 3 | 设定任务 | 中 |
| 4 | 进行学习 | 高 |
| 5 | 练习 | 高 |
| 6 | 测试 | 高 |
| 7 | 学习完成 | 中 |

## 3.2功能描述

### 3.2.1用户登录

用户登录可以分为注册和登录两个过程。在注册过程中，用户提供新用户注册信息，该信息传输至注册事务。注册事务根据提供的信息获取新用户的详细信息，并将其存入account数据库。新用户注册后需要重新登录。而在登录过程中，用户提供用户名和密码，这些信息传输至登录事务。登录事务将提供的用户名和密码与account数据库中的信息进行匹配，同时输出用户信息到界面。

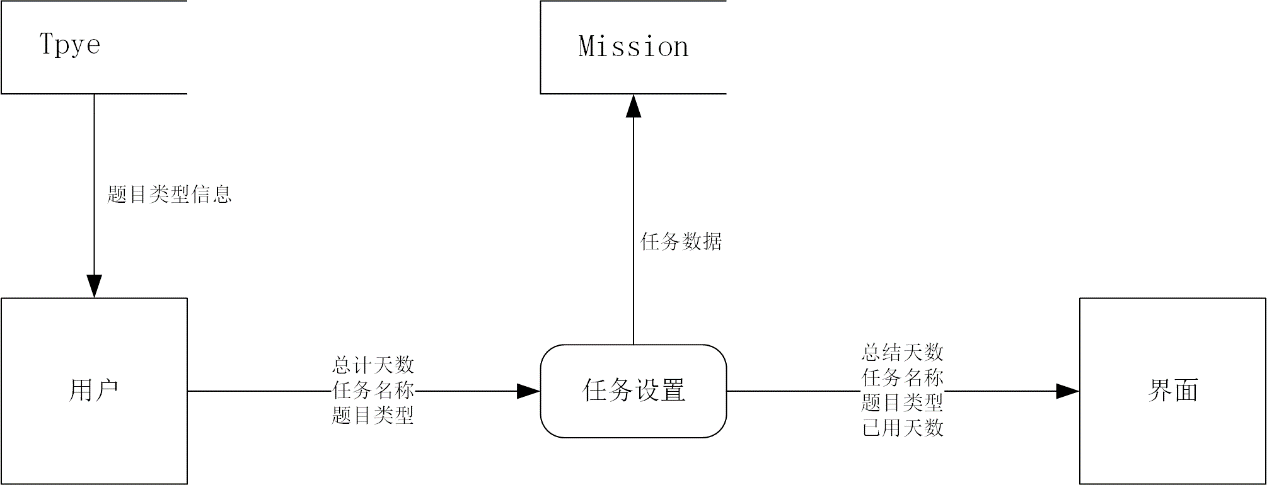


|  |  |  |
| --- | --- | --- |
| **加工名称：1.1 登录** | **使用单位：所有用户** |  |
| **输入部分I** | **处理描述P界面** | **输出部分O** |
| 用户名、密码 | 用户利用自己的用户名与密码进行登录 | 用户信息 |

|  |  |  |
| --- | --- | --- |
| **加工名称：1.2 加工** | **使用单位：所有用户** |  |
| **输入部分I** | **处理描述P界面** | **输出部分O** |
| 用户名、密码、邮箱 | 用户输入用户名、邮箱、密码进行注册 | 用户信息 |

### 3.2.2任务设置

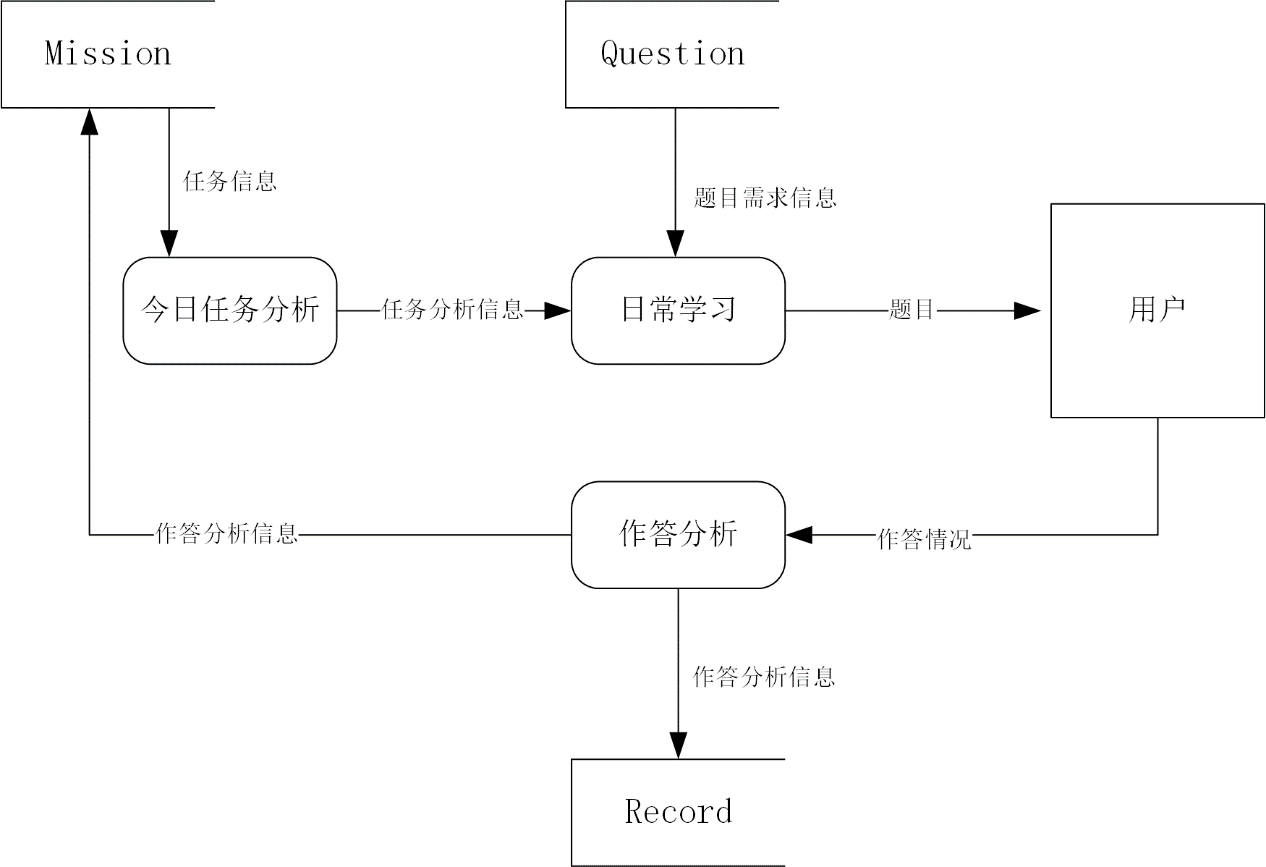
用户可自行设置任务，输入总计天数，任务名称，并从type数据库中获取题目类型，然后任务设置会分析其中的需求，从而输出数据到界面显示给用户，同时写入mission数据库。



|  |  |  |
| --- | --- | --- |
| **加工名称：2.1 任务设置** | **使用单位：所有用户** |  |
| **输入部分I** | **处理描述P界面** | **输出部分O** |
| 总结天数  任务名称  题目类型 | 用户输入自己的任务安排，系统自动设置任务并进行记录与提醒 | 任务数据与安排 |

### 3.2.3日常学习

用户设置好任务后，首先，今日任务分析系统从mission数据库中获取当前的任务安排。然后，系统自主进行任务分析，得到相关的分析信息。日常学习系统利用任务分析信息与题目信息，对题目进行筛选。最后，作答分析系统获取用户的作答情况，并进行详细的分析。



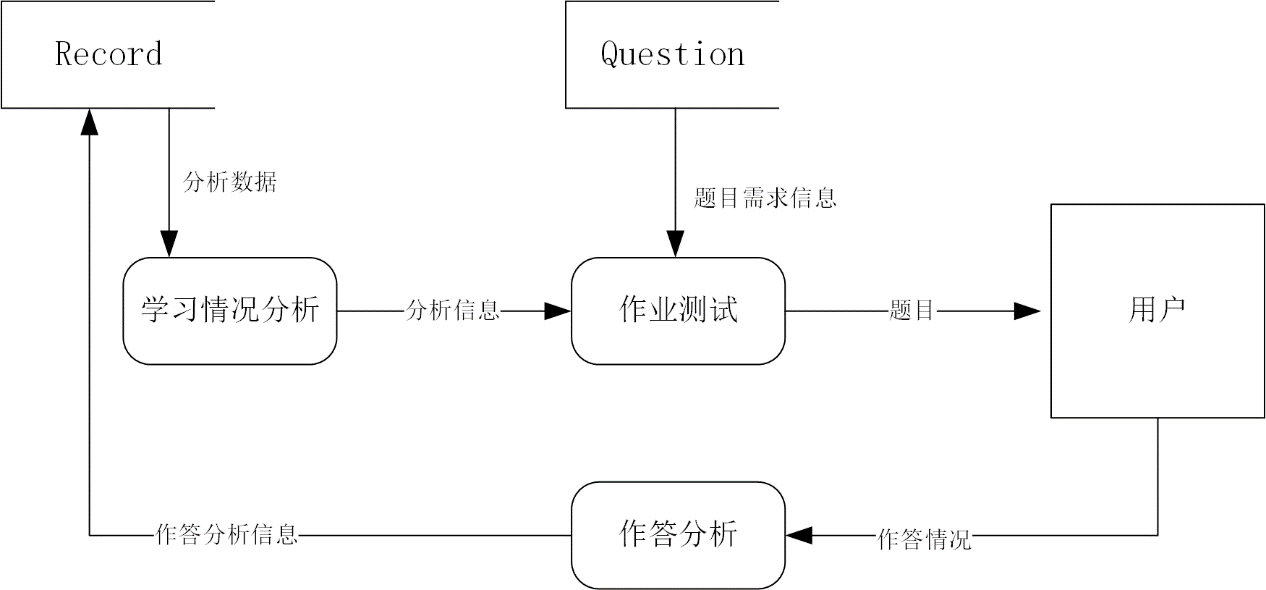
|  |  |  |
| --- | --- | --- |
| **加工名称：3.1 今日任务分析** | **使用单位：所有用户** |  |
| **输入部分I** | **处理描述P界面** | **输出部分O** |
| 任务信息 | 今日任务分析系统从mission数据库中获取当前的任务安排，然后自主分析并得到分析信息 | 任务分析信息 |

|  |  |  |
| --- | --- | --- |
| **加工名称：3.2 日常学习** | **使用单位：所有用户** |  |
| **输入部分I** | **处理描述P界面** | **输出部分O** |
| 任务分析信息  题目信息 | 日常学习系统获取任务分析信息与题目信息并筛选题目 | 题目 |

|  |  |  |
| --- | --- | --- |
| **加工名称：3.3 作答分析** | **使用单位：所有用户** |  |
| **输入部分I** | **处理描述P界面** | **输出部分O** |
| 作答情况  题目信息 | 作答分析系统获取用户作答情况并进行分析 | 作答分析信息 |

### 3.2.4作业测试

用户可以进行作业测试来判断自己的学习效果如何。首先，学习情况分析系统从record数据库中获取分析数据。然后，系统自主进行任务分析，得到相关的分析信息。作业测试系统利用分析信息与题目信息，对题目进行筛选并生成试卷。最后，作答分析系统获取用户的作答情况，并进行详细的分析。



|  |  |  |
| --- | --- | --- |
| **加工名称：4.1 学习情况分析** | **使用单位：所有用户** |  |
| **输入部分I** | **处理描述P界面** | **输出部分O** |
| 分析数据 | 系统从record数据库中获取用户学习情况，然后自主分析并得到分析信息 | 分析信息 |

|  |  |  |
| --- | --- | --- |
| **加工名称：4.2 作业测试** | **使用单位：所有用户** |  |
| **输入部分I** | **处理描述P界面** | **输出部分O** |
| 分析信息  题目信息 | 系统获取任务分析信息与题目信息并筛选题目，生成一张试卷提供给用户测试 | 题目 |

|  |  |  |
| --- | --- | --- |
| **加工名称：4.3 作答分析** | **使用单位：所有用户** |  |
| **输入部分I** | **处理描述P界面** | **输出部分O** |
| 作答情况  题目信息 | 作答分析系统获取用户作答情况并进行分析 | 作答分析信息 |

# 4数据需求

## 4.1静态数据

静态数据是系统中相对固定的数据，主要用于提供基础信息和支持系统功能的运行。课程信息和学习资源数据是平台核心内容，通过丰富多样的资源和合理分类的课程，可以满足用户的学习需求。学生信息和教师信息数据用于个性化推荐和学习管理，帮助系统更好地理解用户的学习背景和需求。静态数据的具体信息如下：

1. **课程信息数据**
   * 包括课程名称、课程描述、课程封面图等。
   * 每门课程应具有标签或分类信息，以便于个性化推荐算法的使用。
   * 课程信息应包含适用年级范围、教学目标、学习难度等属性。
2. **学习资源数据**
   * 视频、文档、练习题等多种类型的学习资源。
   * 每种学习资源应有相关的课程信息和标签，方便系统推荐和用户检索。
   * 学习资源应包含标题、描述、作者、时长（对于视频）、难度等属性。
3. **学生信息数据**
   * 包括学生姓名、年龄、性别、学校信息等基本信息。
   * 学生的学习历史记录，包括已学课程、学习时长、学习成绩等。
4. **教师信息数据**
   * 包括教师姓名、所教授科目、教学经验等基本信息。
   * 教师的教学计划、课程内容、学生成绩等相关信息。
5. **学习社区数据**
   * 学习社区中的帖子、评论等数据。
   * 每个帖子应包含标题、内容、作者、发布时间等基本信息。

## 4.2动态数据

动态数据反映了用户在平台上的实时行为和学习状态，是个性化推荐和学习管理的重要依据。学习进度数据和学习成绩数据可以帮助系统跟踪用户的学习情况，生成学习报告和个性化建议。用户行为数据则可用于分析用户兴趣和行为习惯，优化推荐算法和用户体验。推荐结果数据反馈了系统推荐效果，可用于评估算法的准确性和用户满意度。具体如下：

1. **学习进度数据**
   * 记录学生在每门课程中的学习进度，包括已学习内容、学习时长、学习状态等。
   * 每次学习行为的时间戳，以便于生成学习时间分布等报告。
2. **学习成绩数据**
   * 记录学生在每门课程中的学习成绩，包括每次作业、考试的得分情况。
   * 每次成绩记录应包含时间戳、相关学习任务/考试的信息等。
3. **用户行为数据**
   * 记录用户在平台上的行为，如浏览课程、观看视频、参与社区讨论等。
   * 用户行为数据应包括行为类型、目标对象（如课程、帖子）、时间戳等信息。
4. **推荐结果数据**
   * 记录系统对每位学生的个性化推荐结果，包括推荐的课程、学习资源等。
   * 每次推荐结果应包含推荐时间、推荐内容、推荐理由等信息。

## 4.3数据字典

本项目的数据字典主要包括数据流条目和数据存储条目两部分。

数据流条目描述了系统中涉及的各种数据流动情况，包括数据的输入和输出，以及数据在系统内部的传递。每个数据流条目都提供了对数据流的名称、描述以及相关的具体内容。数据存储条目则描述了系统中用于存储数据的各种数据存储方式，包括数据库、文件系统等。每个数据存储条目都提供了对存储方式的名称、描述以及存储内容的具体信息。

1. **数据流条目**

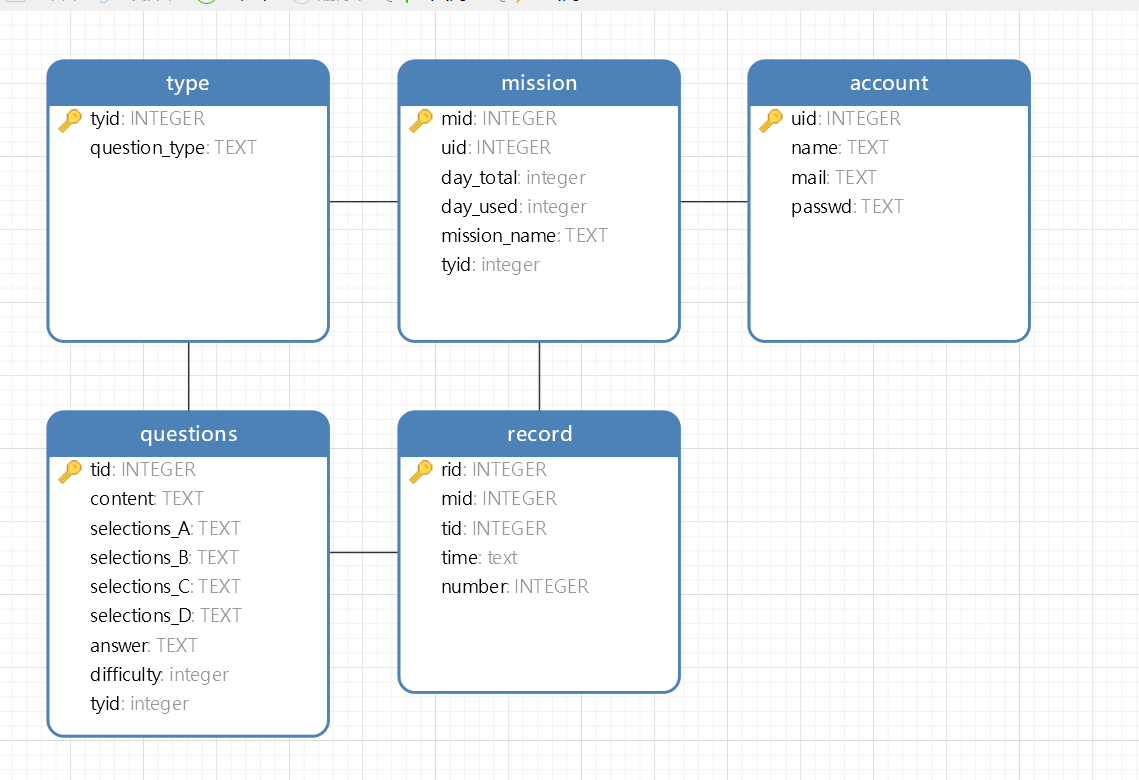
|  |  |
| --- | --- |
| **数据流名称** | **描述** |
| 用户注册信息输入流 | 用户在注册时输入的个人信息，包括姓名、年龄、性别、联系方式等。 |
| 用户登录信息输入流 | 用户在登录时输入的账号和密码信息。 |
| 个性化课程推荐结果流 | 系统根据用户的学习需求、兴趣爱好和学习历史记录生成的个性化课程推荐结果流，包括推荐的课程列表及推荐理由。 |
| 学习进度更新流 | 用户学习过程中的学习进度更新信息，包括学习的课程、学习时间、学习状态等。 |
| 学习成绩更新流 | 用户完成课程评估任务后的学习成绩更新信息，包括课程名称、评估得分、评估时间等。 |
| 用户行为记录流 | 记录用户在平台上的各类行为，如浏览课程、观看视频、参与社区讨论等，用于个性化推荐和用户行为分析。 |
| 学习任务管理信息输入流 | 教师或家长为学生设置的学习任务和计划的信息输入流，包括任务名称、截止日期、任务内容等。 |
| 学生学习报告输出流 | 系统根据学生的学习进度和成绩生成的学习报告信息输出流，包括学习总结、建议改进的地方等。 |
| 系统错误信息输出流 | 系统在处理过程中产生的错误信息输出流，用于提示用户或记录系统日志，包括错误代码、错误描述、解决建议等。 |

1. **数据存储条目**

|  |  |
| --- | --- |
| **数据存储名称** | **描述** |
| 用户信息存储 | 存储用户的个人信息，包括用户名、密码（加密存储）、姓名、年龄、性别、联系方式等。 |
| 课程信息存储 | 存储平台中所有课程的基本信息，包括课程名称、描述、封面图、标签、适用年级范围、难度等。 |
| 学习资源存储 | 存储平台中各类学习资源的信息，包括视频、文档、练习题等，每个资源包含标题、描述、作者、类型、难度等。 |
| 学生学习进度存储 | 存储学生在各门课程中的学习进度信息，包括学习时间、学习状态、已学内容等。 |
| 学生学习成绩存储 | 存储学生在各门课程中的学习成绩信息，包括评估任务名称、得分、评估时间等。 |
| 用户行为记录存储 | 存储用户在平台上的各类行为记录信息，如浏览课程、观看视频、参与社区讨论等，用于个性化推荐和用户行为分析。 |
| 学习任务管理信息存储 | 存储教师或家长为学生设置的学习任务和计划的信息，包括任务名称、截止日期、任务内容等。 |
| 学生学习报告存储 | 存储系统生成的学生学习报告信息，包括学习总结、建议改进的地方等。 |
| 系统日志存储 | 存储系统运行过程中产生的日志信息，包括错误日志、警告日志、调试日志等，用于系统运行状态监控和故障排查。 |

## 4.4数据库描述

我们项目的数据库的ER图如图所示：



其中包括多个实体：

1. **type实体：**
   * tid: 用于唯一标识类型记录的整数ID。
   * question\_type: 描述问题类型的文本字段。
2. **mission实体：**
   * mid: 用于唯一标识任务记录的整数ID。
   * uid: 关联到用户的整数ID。
   * day\_total: 任务总天数的整数值。
   * day\_used: 已使用的天数的整数值。
   * mission\_name: 任务名称的文本字段。
   * tyid: 关联到类型记录的整数ID。
3. **account实体：**
   * uid: 用于唯一标识用户记录的整数ID。
   * name: 用户名的文本字段。
   * mail: 用户邮箱地址的文本字段。
   * passwd: 用户密码的文本字段。
4. **questions实体：**
   * tid: 用于唯一标识问题记录的整数ID。
   * content: 问题内容的文本字段。
   * selections\_A: 选项A的文本字段。
   * selections\_B: 选项B的文本字段。
   * selections\_C: 选项C的文本字段。
   * selections\_D: 选项D的文本字段。
   * answer: 正确答案的文本字段。
   * difficulty: 难度等级的整数值。
   * tyid: 关联到类型记录的整数ID。
5. **record实体：**
   * rid: 用于唯一标识记录记录的整数ID。
   * mid: 关联到任务记录的整数ID。
   * tid: 关联到问题记录的整数ID。
   * time: 记录时间的文本字段。
   * number: 记录数量的整数值。

这些实体之间的关系如下：

（1）mission 实体与 account 实体之间存在关联，表示任务由特定的用户创建或参与。

（2）questions 实体与 type 实体之间存在关联，表示问题属于特定的问题类型。

（3）record 实体同时与 mission 和 questions 实体存在关联，表示记录了用户在特定任务中回答问题的情况。

# 25性能需求

## 5.1数据精度

|  |  |  |
| --- | --- | --- |
| **字段** | **精度** | **备注** |
| 用户名 | char型 | 邮箱 |
| 密码 | char型 | 8-32位 |
| 昵称 | char型 | 2-20位 |
| 用户是否存在 | map型 | 前端传过来含有用户名和密码的json对象，后端接受到之后在数据库中匹配，返回是否匹配的信息给前端 |

## 5.2时间特性

（1）响应时间：用户任意操作后1秒内系统给予反馈信息。

（2）负载检测和调整时间：由系统运行状态来决定，尽可能短以避免影响系统性能和稳定性

（3）数据更新处理时间：由系统运行状态来决定

（4）系统启动时间：由系统运行状态来决定

（5）数据转换和传送时间：能够在20秒内完成

（6）定时任务执行时间：实时进行数据备份，并清理三个月前的数据以保持高效

## 5.3灵活性

当需求发生某些变化时，该软件的基本操作、数据结构、运行环境等等基本不会发生变化，只是对系统的数据库的文件和记录进行处理，就可以满足需求系统。具有良好的模块化和可扩展性设计，能够灵活地增加、修改或替换系统的功能模块。系统可以根据需求扩展新的功能模块，或者集成第三方服务和资源，以满足不断变化的用户需求和业务场景。

# 6运行需求

## 6.1用户界面

（1）注册：用户可以填写用户名、邮箱地址、密码和确认密码等信息完成注册。点击提交按钮后，系统会返回注册成功或失败的消息。

（2）登录：用户可以填写用户名和密码信息完成登录。如果用户没有账号，可以点击注册链接进行注册。

（3）个人中心：登录后，用户可以进入个人中心，其中包含以下功能。首先是修改个人信息，用户可以点击修改信息按钮来更新个人信息，包括用户名、邮箱地址、密码等。其次是查看学习记录，用户可以查看自己学习过的内容和已经学习的内容。还有学习中的内容，显示用户当前正在学习的内容。最后是已学习的内容，显示用户已经学习完成的内容。

（4）任务设置：在任务设置环节，系统引导用户通过两个简单步骤定制学习计划：先是输入打算连续学习的天数，比如“30天”。随后，从提供的学科或主题列表中选择一项，如“编程”。基于用户的选择，系统展示多种题型选项，涵盖基础、进阶、实践等不同难度和侧重点，用户选取最感兴趣或需要加强的题型，由此确定学习方向和内容，启动个性化学习之旅。

（5）学习：日常学习系统智能化地根据每位用户的进度、能力水平以及学习偏好，定制化推送练习题目。

（6）测试：用户可以通过以下方式进行测试。系统提供一系列测试题目供用户答题，用户完成答题后系统给出测试结果和评价。用户可以查看自己的测试成绩和答题情况，并对测试结果进行分析和总结。

## 6.2软件接口

操作系统：支持 Windows 10、Windows 11 和 Android 系统。

软件设备：Node.js 、Python

## 6.3硬件接口

（1）内存：1GB 以上

（2）磁盘空间：1GB 以上

（3）CPU 需求频率：1.2GHz 以上

（4）硬盘空间：5GB 以上

# 7其他需求

## 7.1验收标准

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **测试功能** | **测试项** | **输入/操作** | **检验点** | **预期结果** | **验收** |
| 用户注册 | 注册 | 填写注册页面的用户名、邮箱、密码等信息，点击提交按钮进行注册 | 注册信息的格式检查，注册成功后系统是否返回注册成功的消息 | 注册信息符合格式要求，注册成功后系统返回注册成功的消息；如果输入信息格式不正确，系统应提示相应的错误信息。 |  |
| 用户登录 | 登陆 | 填写登录页面的用户名和密码，点击登录按钮进行登录 | 用户名和密码的验证，登录成功后是否跳转到个人中心页面 | 输入正确的用户名和密码后，登录成功并跳转到个人中心页面；如果用户名或密码错误，系统应提示登录失败。 |  |
| 用户信息查看、修改 | 用户中心 | 在个人中心页面进行相应的操作，如修改个人信息、查看学习记录等 | 修改信息是否成功保存，学习记录和学习内容的展示是否准确 | 修改个人信息后，信息成功保存并显示在个人中心页面；学习记录、学习中的内容和已学习的内容都应准确显示。 |  |
| 设定学习目标并学习 | 学习 | 在设定目标界面输入学习天数、学习名称、题目类型，在日常学习中获得题目并进行学习 | 目标的设定能够正确设置在数据库中，能够根据模型推送个性化的日常学习题目 | 用户能够个性化的设置自己的学习目标，并在日常学习中获取自己想要的学习内容 |  |
| 答题、检测学习成果 | 测试 | 在测试页面进行答题，提交答案后系统给出测试结果 | 答题过程是否流畅，测试结果的准确性 | 用户能够顺利进行答题，系统根据答题情况给出准确的测试结果。 |  |

## 7.2质量标准

1. 可用性：软件能够在需要时可靠地提供服务，用户能够方便地访问和使用软件，系统无故障地运行并持续可用。
2. 可靠性：软件能够在各种环境和条件下保持稳定性和可靠性，不会频繁出现故障或崩溃，并且能够正确地处理异常情况。
3. 可维护性：软件易于理解、修改和维护，具有良好的代码结构和文档，使开发人员能够轻松地进行软件维护和更新。
4. 安全性：软件能够保护用户的数据和系统免受未经授权的访问、攻击和恶意行为的影响，确保用户信息的保密性、完整性和可用性。
5. 性能：软件能够在合理的时间内响应用户请求，并且能够处理大量数据或用户同时访问的情况，保持良好的性能表现。
6. 可扩展性：软件能够在需要时进行扩展和升级，以应对不断变化的需求和增长的用户量，保持系统的灵活性和可扩展性。
7. 可测试性：软件设计良好，易于进行测试和验证，能够及早发现和修复潜在的问题，确保软件的质量和稳定性。
8. 易用性：软件界面友好、操作简单，用户能够轻松地学习和使用软件，提高用户满意度和使用体验。