**基于大语言模型的**

**智能学习推荐管理系统**

**软**

**件**

**架**

**构**

**设**

**计**

**课程：软件工程实践**

**班级：公信21**

**成员：李欣桦、吴林昊、李傲、杜威、刘心彤**

**目录**

[1文档简介 1](#_Toc167386146)

[1.1文档目的 1](#_Toc167386147)

[1.2文档范围 1](#_Toc167386148)

[1.3定义、术语及缩略词 1](#_Toc167386149)

[1.4参考资料 1](#_Toc167386150)

[2架构描述方式 2](#_Toc167386151)

[2.1架构视图阅读指南 2](#_Toc167386152)

[2.2图表与模型阅读指南 2](#_Toc167386153)

[3架构设计目标 2](#_Toc167386154)

[3.1关键功能 2](#_Toc167386155)

[3.2关键质量属性 3](#_Toc167386156)

[3.3业务需求与约束因素 3](#_Toc167386157)

[4架构设计原则 4](#_Toc167386158)

[4.1设计原则概述 4](#_Toc167386159)

[4.2备选方案与被否原因 4](#_Toc167386160)

[4.3架构设计对后续工作的限制 5](#_Toc167386161)

[5逻辑架构视图 5](#_Toc167386162)

[5.1顶层逻辑架构视图 5](#_Toc167386163)

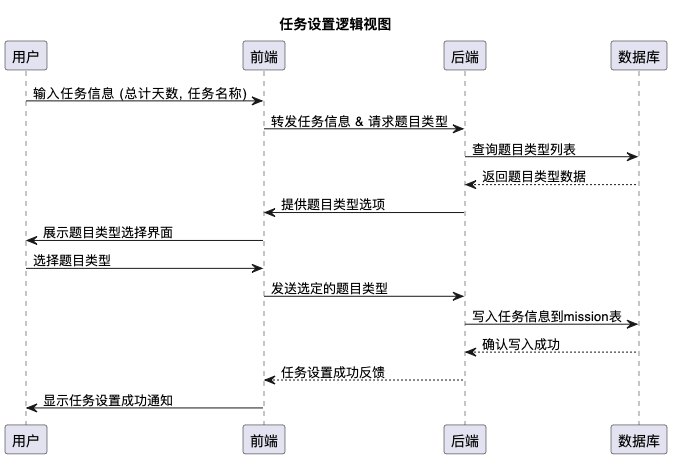
[5.1.1需求架构视图 5](#_Toc167386164)

[5.1.2数据流程视图 6](#_Toc167386165)

[5.2功能逻辑架构（分解）视图 7](#_Toc167386166)

[5.2.1用户登录逻辑视图 7](#_Toc167386167)

[5.2.2任务设置逻辑视图 8](#_Toc167386168)

[ 8](#_Toc167386169)

[5.2.3日常学习逻辑视图 9](#_Toc167386170)

[5.2.4学习测试逻辑视图 11](#_Toc167386171)

[5.3开发视图 12](#_Toc167386172)

[5.3.1模块结构视图 12](#_Toc167386173)

[5.3.2功能架构视图 13](#_Toc167386174)

[5.4运行视图 15](#_Toc167386175)

[5.5实现视图 16](#_Toc167386176)

[5.6物理架构视图 16](#_Toc167386177)

[6需求与架构之间的映射 16](#_Toc167386178)

[6.1核心功能 16](#_Toc167386179)

[6.1.1日常学习 16](#_Toc167386180)

[6.1.2任务设置 17](#_Toc167386181)

[6.1.3作业测试 17](#_Toc167386182)

[6.2辅助功能 18](#_Toc167386183)

[6.2.1用户创建与登录 18](#_Toc167386184)

[6.2.2数据分析 18](#_Toc167386185)

[6.3软件系统总体功能/对象结构 19](#_Toc167386186)

[6.3.1总体结构 19](#_Toc167386187)

[6.3.2软件子系统功能/对象结构 20](#_Toc167386188)

# 1文档简介

本节旨在概述软件架构设计文档的基本信息，为读者提供文档的整体框架、目的、覆盖范围以及在阅读过程中可能遇到的关键定义、术语和缩略词说明。此外，还将列出编写该文档时参考的重要资料，以确保设计的准确性和可追溯性。

## 1.1文档目的

本文档旨在详述基于大语言模型的智能学习推荐管理系统（LearnLark）的软件架构设计。它阐述了系统的技术栈、组件交互关系、数据流程、功能逻辑以及系统如何支撑关键目标，为高效、个性化学习体验服务。

## 1.2文档范围

本设计文档涵盖以下内容：

·系统的高层架构视图，展示主要组件及其交互方式。

·关键技术决策，包括所采用的编程语言、框架、数据库、中间件等。

·数据结构和接口设计，描述数据如何在系统内部流动及对外交互。

·安全性、性能、可扩展性和可维护性方面的考虑和策略。

·部署架构及运维要求。

·未解决的问题及未来可能的扩展方向。

## 1.3定义、术语及缩略词

本文档中未特别提及需要额外定义的术语，使用通用软件工程词汇。

## 1.4参考资料

[1] Martin Fowler. "Patterns of Enterprise Application Architecture." Addison-Wesley, 2002.

[2] Rozanski, Nick and Eoin Woods. "Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives." Addison-Wesley, 2012.

[3] "Microsoft Azure Architecture Center." Microsoft Corporation.

[4] OWASP Foundation. "OWASP Secure Software Development Lifecycle Project."

# 2架构描述方式

本部分详细阐述了如何理解和阅读架构设计中的各种视图和模型，以确保所有项目参与者能够准确把握系统架构的全貌及细节。

## 2.1架构视图阅读指南

架构视图是从不同角度对系统进行抽象和描述的窗口，帮助理解系统的关键方面。每个视图都针对特定的受众或关注点，例如开发者、运维团队或项目管理者。在阅读时，请注意以下几点：

·识别目标受众：了解每个视图旨在服务于哪类读者，这有助于你聚焦于与自己角色相关的信息。

·理解视角：每个视图代表了系统的一个特定视角，比如逻辑、开发、运行或物理视角，确保从正确的角度解读信息。

·关联视图：不同视图之间存在联系，理解它们之间的依赖和映射关系对于全面把握系统架构至关重要。

## 2.2图表与模型阅读指南

图表和模型是表达架构设计的主要手段，遵循以下原则可以帮助有效阅读：

·层次与分解：注意图表的层次结构，高级视图通常提供概览，而详细视图则深入到具体组件或流程。

·数据流向与交互：关注箭头指示的数据流向和组件间的交互模式，理解信息如何在系统内流转。

·关注点分离：区分功能性描述（如用户行为、服务调用）与非功能性关注点（如安全性、性能）。

# 3架构设计目标

## 3.1关键功能

软件的关键功能要求如下：

（1）个性化学习任务规划：用户进入系统后，可以根据自己的学习需求和目标，利用“任务制定”模块规划个性化的学习任务。用户可以设定学习目标、选择学习内容和制定学习计划。

（2）知识学习：在“知识学习”模块中，用户可以浏览平台提供的各类学习资源，包括课程、教材、视频等。根据个人兴趣和学习需求，选择合适的学习内容进行学习。

（3）知识测试：平台提供了“知识测试”模块，用户可以在此进行知识测试和练习，检验自己的学习成果，并及时发现和弥补学习中的不足之处。

（4）学习记录与分析：平台会记录用户的学习行为和学习数据，包括学习时间、学习内容、学习进度等。通过“学习记录”模块，用户可以查看自己的学习历史和学习情况，并进行数据可视化分析，了解自己的学习状况和趋势。

## 3.2关键质量属性

·性能：保证系统响应迅速，即使在高并发访问下也能维持良好的用户体验。

·可扩展性：随着用户量增长和学习资源的扩充，系统需易于扩展，支持平滑增加服务器资源或服务实例。

·安全性：确保用户数据安全，包括个人信息、学习记录等，采用加密技术保护数据传输和存储安全。

·可用性：实现高可用架构，减少系统故障时间，提供故障转移与恢复机制。

·易用性：界面友好，操作直观，降低用户上手难度，提升学习平台的吸引力。

## 3.3业务需求与约束因素

·业务需求：软件需支持多平台访问（Web、移动应用），适应不同场景下的学习需求。同时，应具备一定的社交学习功能，如学习小组、论坛交流，促进用户间互动与合作。

·约束因素：

·技术限制：考虑现有技术栈和团队能力，选择合适的技术方案，避免过度设计。

·成本预算：在保证功能和质量的前提下，合理控制开发与运维成本，避免不必要的开支。

·法律法规：遵守相关教育行业数据保护法规，确保用户隐私和版权合规。

·时间线：根据市场窗口期和竞争态势，设定合理的开发周期，确保产品按时上市。

# 4架构设计原则

本节阐述了在软件架构设计过程中的核心指导思想、评估过的备选方案及其排除理由，以及架构决策对未来开发与维护工作可能产生的影响。

## 4.1设计原则概述

设计原则旨在为整个架构提供一个坚固的理论基础，确保系统能够在复杂多变的环境中稳定运行，同时便于未来的扩展和维护。核心原则包括但不限于：

·模块化：系统应被划分为独立、可复用的组件，以便于单独开发、测试和部署，提高系统的灵活性和可维护性。

·松耦合：各模块之间保持最低程度的依赖关系，变更一个模块时，对其他模块的影响最小化。

·可扩展性：架构设计应易于水平或垂直扩展，以应对用户增长或负载增加的需求，不牺牲性能。

·安全性：从设计之初就融入安全机制，保护用户数据和系统免受攻击。

·响应式设计：确保系统能在各种设备和网络条件下提供良好用户体验，包括移动端优化和网络状态不佳时的降级处理。

·数据驱动：基于数据做出决策，利用收集到的用户行为和系统性能数据来持续优化服务。

## 4.2备选方案与被否原因

在架构设计初期，团队探讨了多个技术栈和架构模型，最终确定当前方案。其中，几个重要的备选方案及其排除理由包括：

·微服务架构 vs 单体架构：虽然微服务架构提供了高度的可扩展性和模块化，但考虑到项目初期规模较小且团队对微服务管理经验有限，决定采用单体架构以减少开发和运维的复杂度。未来随着系统发展，逐步向微服务架构过渡。

·云原生 vs 本地部署：尽管云原生架构提供了弹性伸缩、快速部署等优势，但由于初期预算限制和数据主权考虑，选择本地部署作为初始方案，但设计时保持对云迁移的兼容性。

·SQL数据库 vs NoSQL数据库：基于数据关系复杂度和事务处理需求，尽管NoSQL数据库在处理大规模非结构化数据上有优势，但本项目更多涉及结构化数据和关联查询，因此选择了关系型数据库（SQL）。

## 4.3架构设计对后续工作的限制

当前架构设计虽然在初期提供了快速开发和部署的优势，但也为后续工作带来了一定的局限性：

·扩展性挑战：单体架构可能随着功能增加而变得庞大，导致开发和部署周期延长，未来可能需要大量重构以适应更复杂的业务需求。

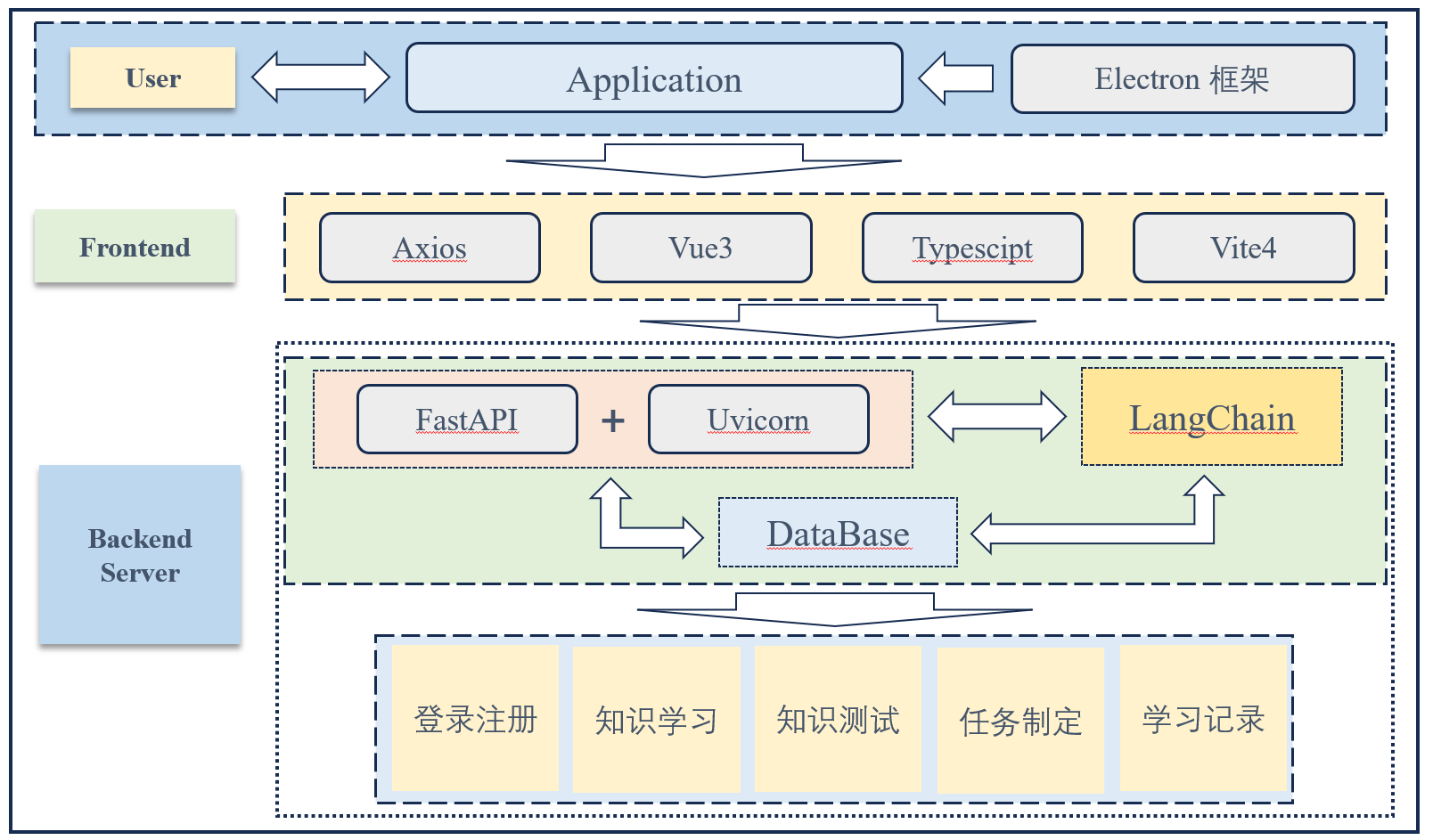
·技术栈锁定：选定的技术栈可能限制了引入新工具和技术的速度，特别是当这些新技术与现有架构不兼容时。

·运维复杂度：本地部署增加了硬件管理和运维的工作量，可能影响到系统的稳定性和响应速度，需要投入更多资源进行监控和维护。

# 5逻辑架构视图

## 5.1顶层逻辑架构视图

### 5.1.1需求架构视图

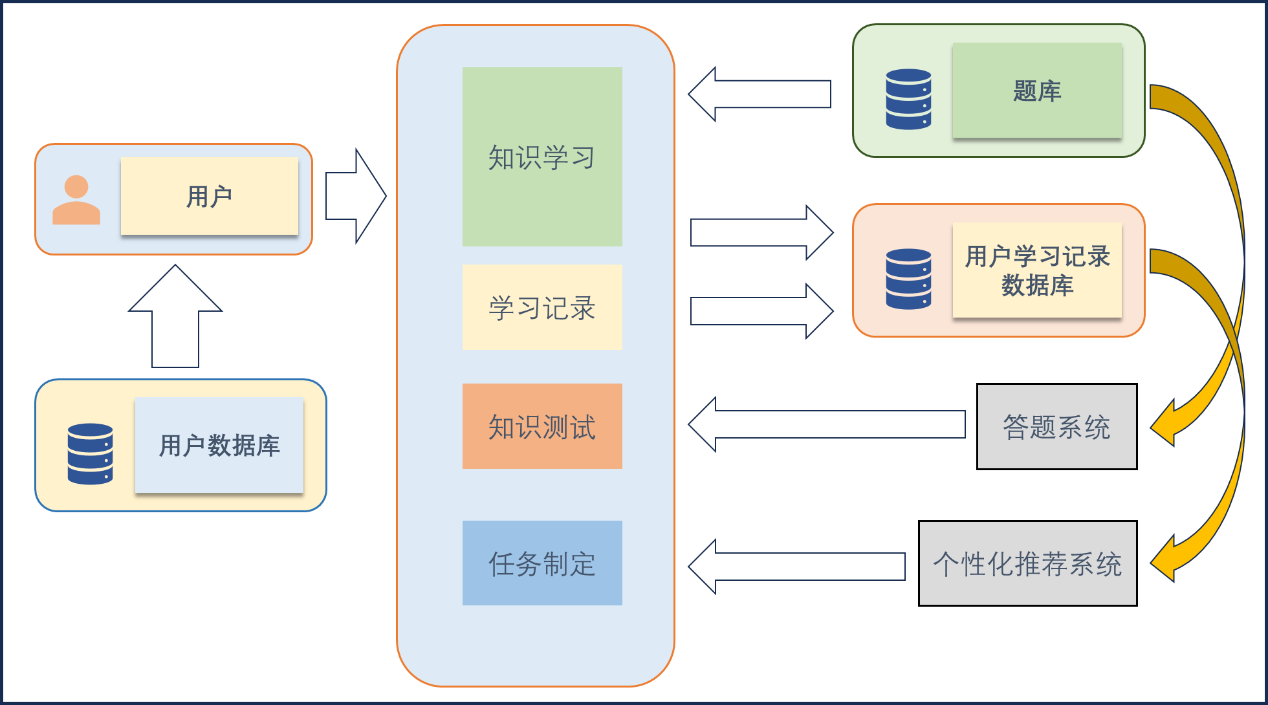


架构视图展示了该项目的技术堆栈和系统组件之间的交互关系。用户（User）通过应用（Application）与 Electron 框架进行交互。在前端部分（Frontend），应用基于 Vue3 和 Axios 构建，并使用 TypeScript 进行类型检查，而 Vite4 则作为构建工具。

在后端服务器（Backend Server）中，FastAPI 被用于快速构建 API 接口，Uvicorn 是一个高性能的 ASGI 服务器。这些接口与 LangChain 技术相结合，形成了系统的后台核心。数据库（Database）负责存储各种信息，如登录注册、知识学习、知识测试、任务制定以及学习记录等。

整个系统的设计旨在提供一个高效且个性化的学习体验，通过整合教育资源并追踪学习进度来优化学习过程。此外，该系统还利用 Electron 桌面应用程序框架实现了跨平台支持，使得用户能够在多种设备上访问和使用这个教育平台。

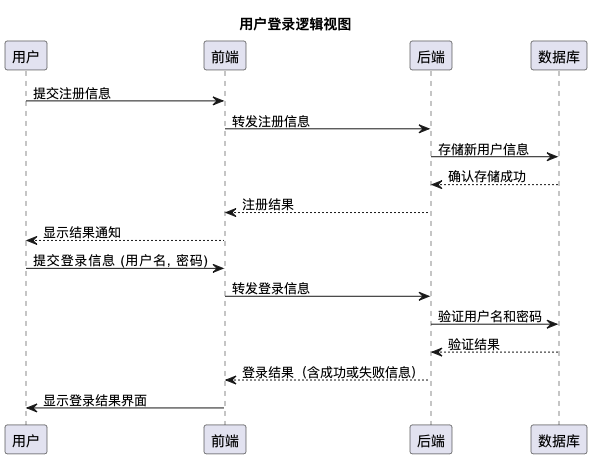
### 5.1.2数据流程视图



数据流程图展示了用户如何与系统互动以及数据如何在整个系统中流动。首先，用户（User）通过输入自己的信息进入系统，这些信息被保存在用户数据库（User Database）中。然后，用户可以开始学习知识，这部分内容被记录在学习记录（Learning Record）中。当用户完成一定的学习后，可以通过知识测试（Knowledge Test）检验自己的掌握程度。测试结果会被发送给答题系统（Answering System）进行评分，并根据用户的答案更新题库（Question Bank）。同时，这些测试结果也会被用来训练个性化推荐系统（Personalized Recommendation System），以便为每个用户提供更加精准的知识学习建议。最后，用户的学习记录会被定期备份到用户学习记录数据库（User Learning Record Database）中，以备后续查询或分析之用。

## 5.2功能逻辑架构（分解）视图

## 5.2.1用户登录逻辑视图



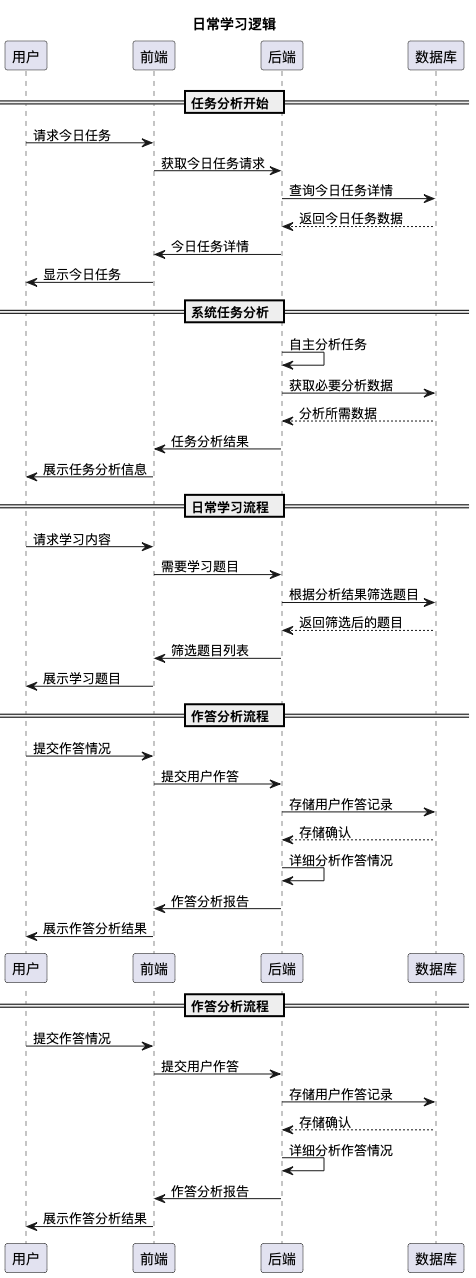
这张序列图描述的是用户登录注册的逻辑，涉及到用户、前端、后端和数据库四个角色的交互。首先，用户向前端提交注册信息，前端接收到这些信息后，将其转发给后端。后端负责将这些信息存储到数据库中，并确认存储成功。此时，前端会向用户展示注册结果。接下来，用户提交登录信息，包括用户名和密码。前端接收到这些信息后，同样将其转发给后端。后端负责验证用户名和密码是否匹配，得到验证结果。前端接收到验证结果后，会向用户展示登录结果。

### 5.2.2任务设置逻辑视图

### 1111.drawio

这张图描述的是任务设置的逻辑流程，涉及用户、前端、后端和数据库四个角色的交互。首先，用户输入任务信息，包括总计天数和任务名称。接着，前端接收到任务信息后，转发给后端。后端从数据库中查询题目类型列表，并返回给前端。前端根据后端返回的题目类型列表，提供给用户选择。用户选择题目类型后，前端将选定的题目类型发送给后端。后端将任务信息写入mission表，并确认写入成功。最后，前端接收到任务设置成功的反馈后，向用户展示任务设置成功的通知。

### 5.2.3日常学习逻辑视图

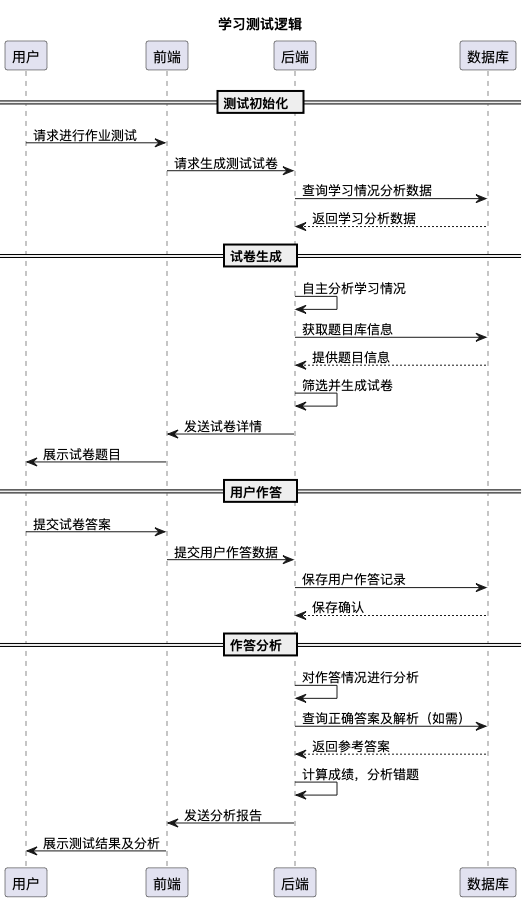


这张图描述了用户在日常学习中的逻辑流程，涉及用户、前端、后端和数据库四个角色的交互。首先，用户请求查看今日任务，前端接收到请求后转发给后端。后端从数据库中获取今日任务请求，并返回给前端。前端根据返回的数据，展示今日任务详情。

接着，系统开始任务分析流程。后端根据分析结果筛选题目，并返回给前端。前端根据返回的题目列表，展示需要学习的题目。然后，用户请求学习内容，前端接收到请求后转发给后端。后端根据分析结果筛选题目，并返回给前端。前端根据返回的题目列表，展示需要学习的题目。

最后，用户提交作答情况，前端接收到作答情况后，提交用户作答并存储用户作答记录。后端详细分析作答情况，并生成作答分析报告。前端接收到报告后，展示作答分析结果给用户。

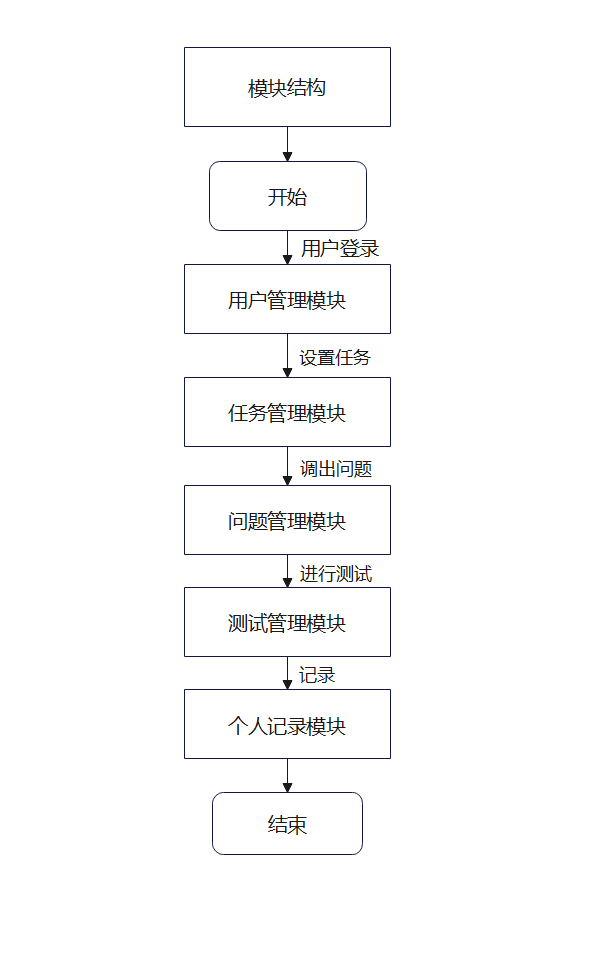
### 5.2.4学习测试逻辑视图



这张图描述了用户在学习测试逻辑流程中的交互，涉及用户、前端、后端和数据库四个角色的交互。首先，用户请求进行作业测试，前端接收到请求后转发给后端。后端从数据库中获取学习分析数据，并返回给前端。前端根据返回的数据，生成测试试卷。接着，用户提交试卷答案，前端接收到答案后，提交用户作答数据并保存用户作答记录。后端详细分析作答情况，并生成作答分析报告。最后，用户提交试卷答案，前端接收到答案后，提交用户作答数据并保存用户作答记录。后端详细分析作答情况，并生成作答分析报告。

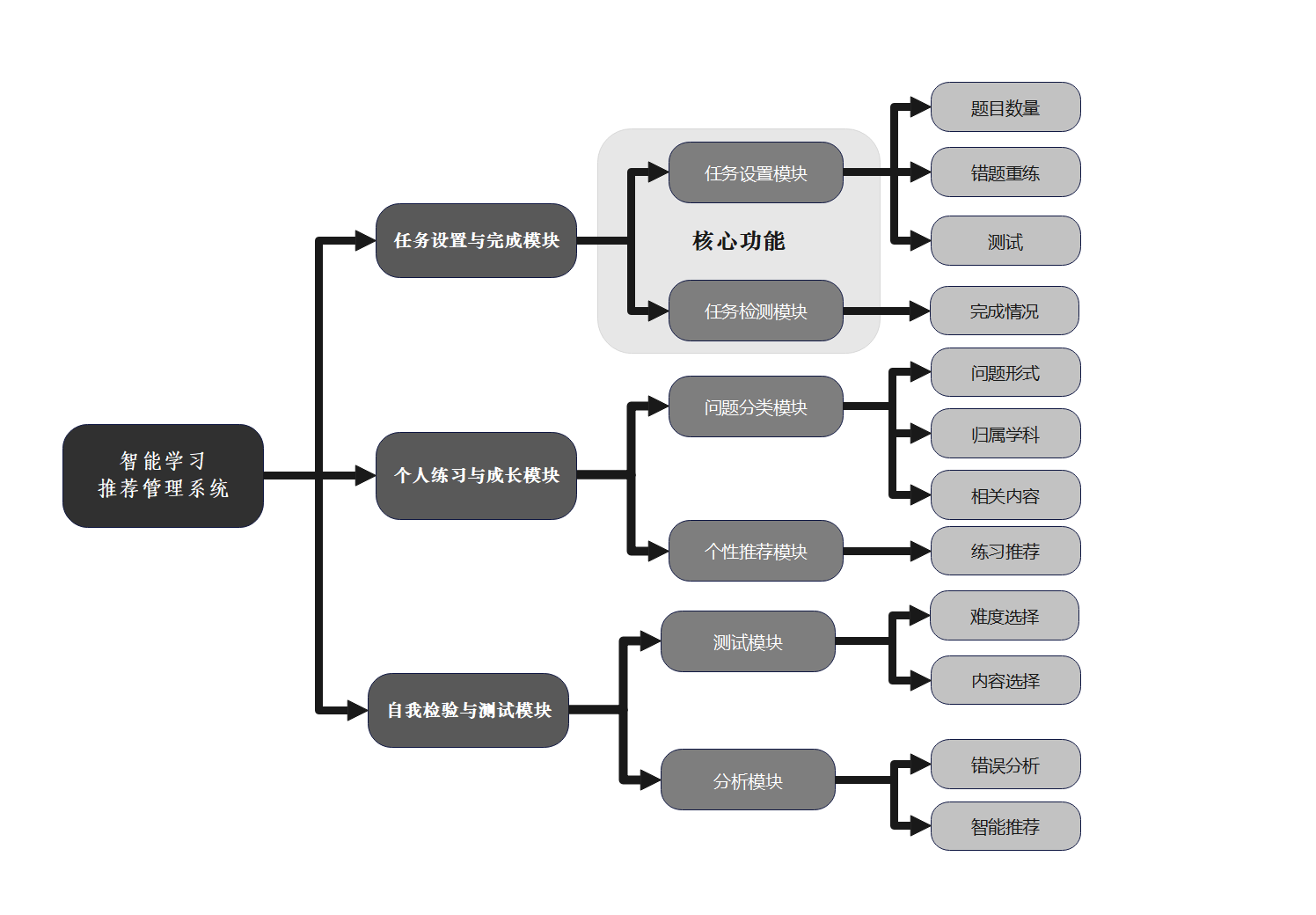
## 5.3开发视图

### 5.3.1模块结构视图



|  |  |  |  |
| --- | --- | --- | --- |
| **功能模块** | **功能** | **功能描述** | **优先级** |
| 用户管理模块 | 用户信息的管理 | 实现用户的登陆，并将后续各数据库的信息与用户进行捆绑，确保用户有登陆的唯一标识。 | 高 |
| 任务管理模块 | 任务的设置与完成 | 进行任务的设置，包含题目数量、纠错情况、等并最终进行完成与否的反馈。 | 中 |
| 问题管理模块 | 记录问题的相关分类与信息 | 将练习与测试的问题进行相关分类，除搜索相关内容外还可以根据题型、相关内容和题目错误率、难度等进行智能推荐 | 高 |
| 测试管理模块 | 测试的分类与推荐 | 根据练习错误情况进行智能推荐，选择恰当的题目进行测试。 | 中 |
| 个人记录模块 | 错题与测试的记录 | 记录错误情况以更好地进行反馈 | 高 |

### 5.3.2功能架构视图



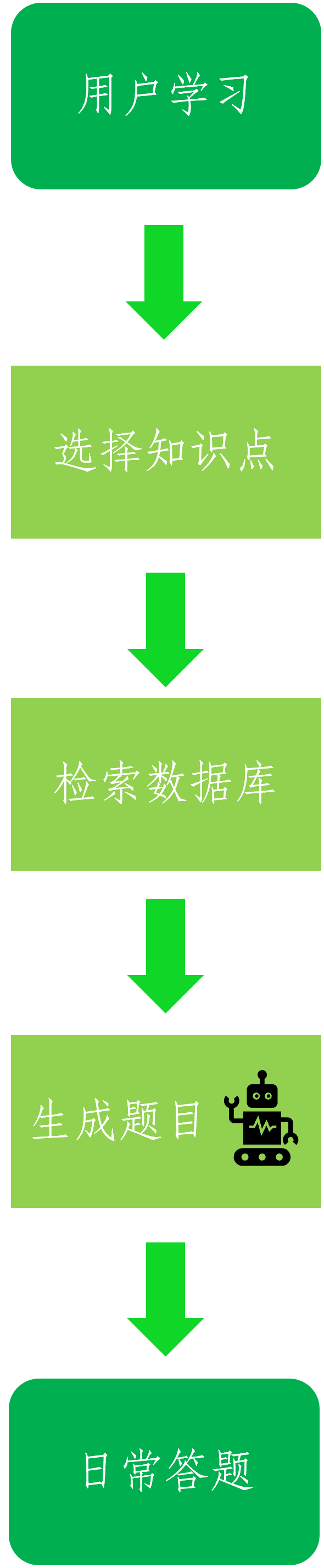
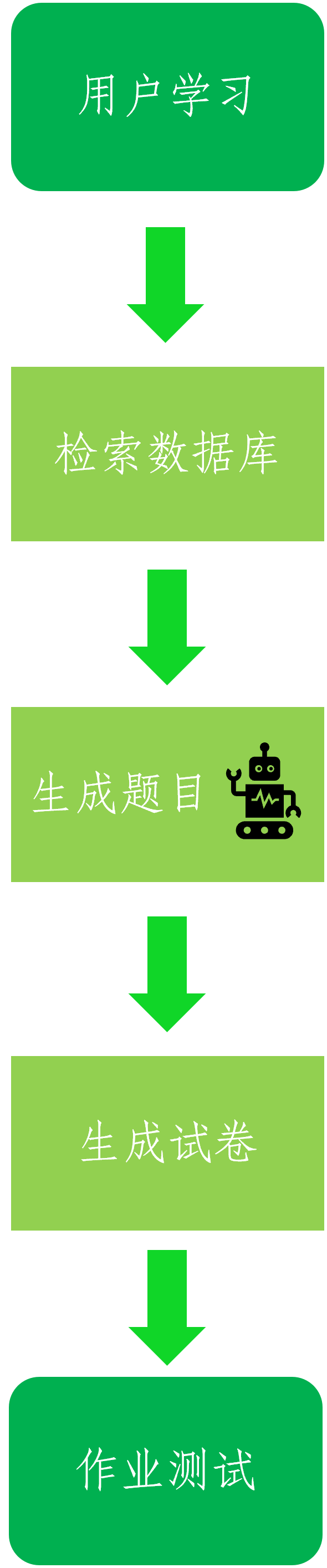
智能学习推荐管理系统分为三个主要模块，分别是任务设置与完成模块、个人练习与成长模块、自我检验与测试模块。

在任务设置与完成模块中，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再进行推荐，从而输出数据到界面显示给用户，并最终检测是否完成。

个人练习与成长模块是用户自主学习的过程，用户设置好任务后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，对题目进行筛选并进行巩固推荐。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。

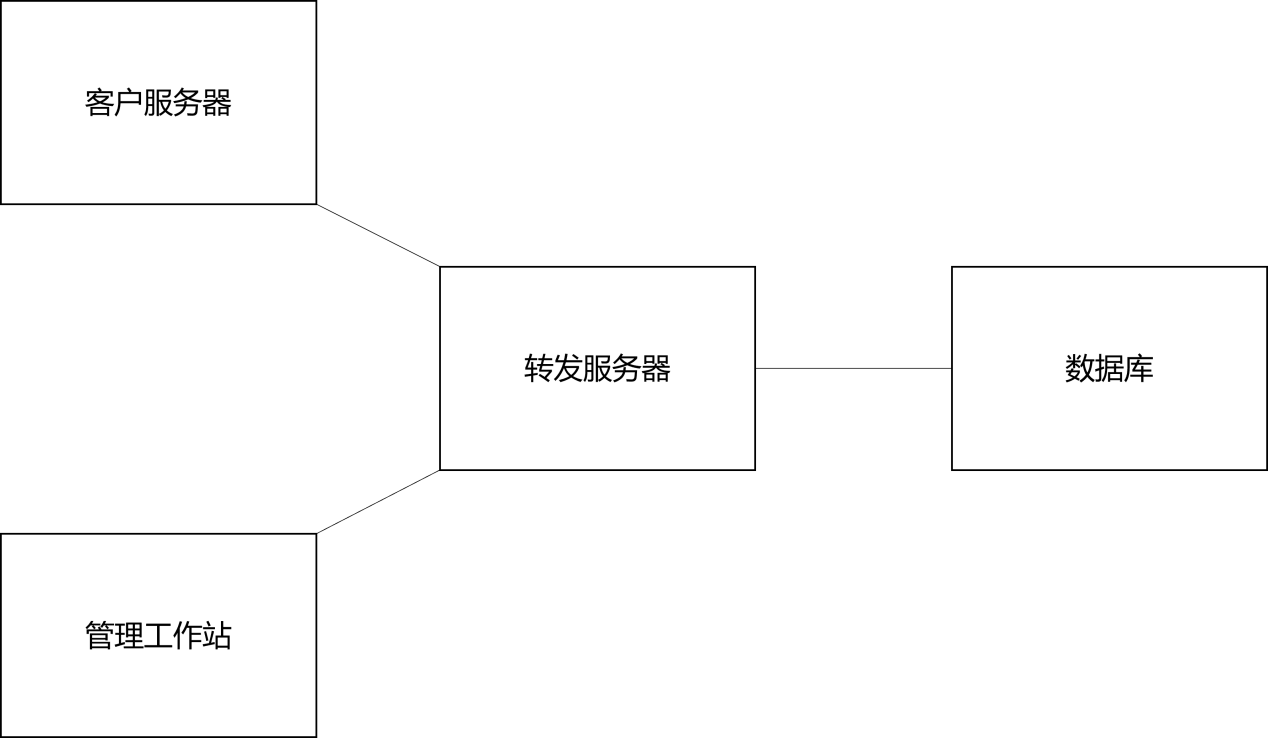
在自我检验与测试模块中，用户可以进行作业测试来判断自己的学习效果如何。首先学习情况分析系统从record数据库中获取分析数据。然后进行任务分析，得到相关的分析信息。作业测试系统结合用户的作答情况与自主选择对题目进行筛选并生成试卷。最后，作答分析系统获取用户的作答情况并进行详细的分析。

## 5.4运行视图

## 5.5实现视图

## 5.6物理架构视图



# 6需求与架构之间的映射

## 6.1核心功能

### 6.1.1日常学习

学生在个性化学习平台上，可以根据自己的学习需求和进度，自主选择感兴趣或需要巩固的知识点。平台提供了丰富多样的学习资源，包括视频教程、图文解析、动画演示等，以适应不同学生的学习偏好，确保每个学生都能以最适合自己的方式深入理解每一个知识点。

完成理论学习后，学生可立即进入配套的练习环节。练习题库针对每个知识点精心设计，覆盖了从基础到进阶的各类题目，旨在全方位检验学生对知识点的掌握程度。尤为值得一提的是，该平台采用了先进的自适应学习算法。在学生做练习时，系统会根据学生的实时答题情况动态调整题目的难度。例如，如果学生正确回答了一道中等难度的题目，接下来系统将会智能推送一道难度更高、更具挑战性的题目，以促进学生不断突破自我，达到学习效果的最大化。反之，若学生在某难题上遇到障碍，系统则会适时提供解题提示，并适当降低后续题目的难度，帮助学生重建信心，逐步攻克难关。

每道练习题的完成情况，包括答题时间、正确率、尝试次数等详细数据，都将被实时记录并存储到云端数据库中。这些数据不仅为学生个人提供了学习成效的直观反馈，还允许教师和家长通过数据分析功能，全面了解学生在各个知识点上的强弱点分布，从而为学生量身定制后续的学习计划和辅导策略。此外，大数据分析还能帮助教育研究者洞察学习模式，优化教学内容与方法，不断提升教学质量与效率，真正实现因材施教的目标。

### 6.1.2任务设置

学生在个性化学习旅程中扮演着积极主动的角色，他们可以根据自己的学习进度、兴趣偏好及目标设定，灵活地在智能学习平台上自行设计并管理一系列学习任务。这一自定义任务设置功能，宛如一位贴心的个人学习顾问，不仅赋予学生高度的自主权，还融入了智能化的辅助与监督机制，确保学习计划的有效执行。

通过直观易用的界面，学生可以轻松添加各类任务，无论是复习特定章节、完成在线课程、阅读专业文献，还是练习特定技能如编程、语言学习等。每项任务均可配置详细的描述、预期完成时间及优先级，帮助学生清晰规划每日、每周乃至每月的学习路径。更进一步，系统支持设置子任务分解复杂项目，让学生能逐步攻克难关，避免拖延，有效管理时间和精力。

一旦任务确立，智能提醒系统便开始发挥其核心作用。它依据学生设定的时间表，通过手机通知、邮件提醒或是平台内的消息推送，以温和而坚定的方式，按时督促学生关注即将到期的任务。这种即时反馈机制有助于培养良好的时间管理和自我约束习惯，确保学生不会遗漏任何自设的学习里程碑。

### 6.1.3作业测试

作业测试平台依托于一个强大的数据库，该数据库详细记录了每位学生针对每一个知识点的回答情况。这一过程不仅仅是数据的积累，更是一种深度学习的循环。平台采用先进的数据分析算法，对徐盛等每位学生的答题内容进行全面扫描，细致入微地分析他们的理解程度、问题解决策略以及潜在的知识盲区。

依据这些详实的数据分析结果，系统智能化地定制出一套套考试题目。这些题目不仅贴合教学大纲的要求，更能够精准反映出每位学生的学习现状与进步空间，实现了从“一刀切”到“量体裁衣”的教育测评转变。当学生完成作答后，系统即刻运用复杂的评估模型，对学生答案进行即时评分与反馈。这包括但不限于知识点掌握的正确率、解题思路的合理性以及可能存在的误解或混淆点。

尤为重要的是，每一次作答后的反馈不仅是对学生学习成效的直接展示，也是对教学内容适应性和教学方法有效性的一次检验。系统会汇总分析所有学生的作答结果，为教师提供可视化的班级学习概况，指明集体强项与共性弱点，从而助力教师调整教学策略，实施更加高效和针对性的教学干预。如此一来，作业测试不仅成为了学生自我提升的阶梯，也成为了优化教育资源配置、推动教育质量持续改进的重要工具。

## 6.2辅助功能

### 6.2.1用户创建与登录

在我们的系统设计中，用户能够轻松享受从账户创建到无缝登录的全过程，这一流畅体验旨在简化用户的入门步骤并确保安全无虞的访问环境。用户首先通过一个直观简洁的注册界面启动他们的旅程，这里要求输入基本信息，如电子邮件地址或者手机号码，设定一个强密码，并可选地完成验证步骤，比如短信验证码或邮箱确认链接，以此来加强账户的安全性并防止未经授权的访问企图。

完成注册后，新创建的用户资料会被安全地保存在加密的服务器上，确保个人信息的隐私和保护。紧接着，用户即刻可以利用刚刚设立的凭证进行登录。登录界面设计友好，支持多种认证方式，包括但不限于传统的用户名密码登录，以及更为现代的生物识别登录如指纹识别或面部识别（在支持的设备上），为用户提供便捷且多样化的选择。

为了提升用户体验，系统还融入了“记住我”选项，允许用户在信任的设备上保持登录状态，免去频繁输入登录信息的繁琐。同时，实施智能监测和二次验证技术，如在异常登录尝试时发送安全提示至绑定的邮箱或手机，进一步加固账户安全防线。

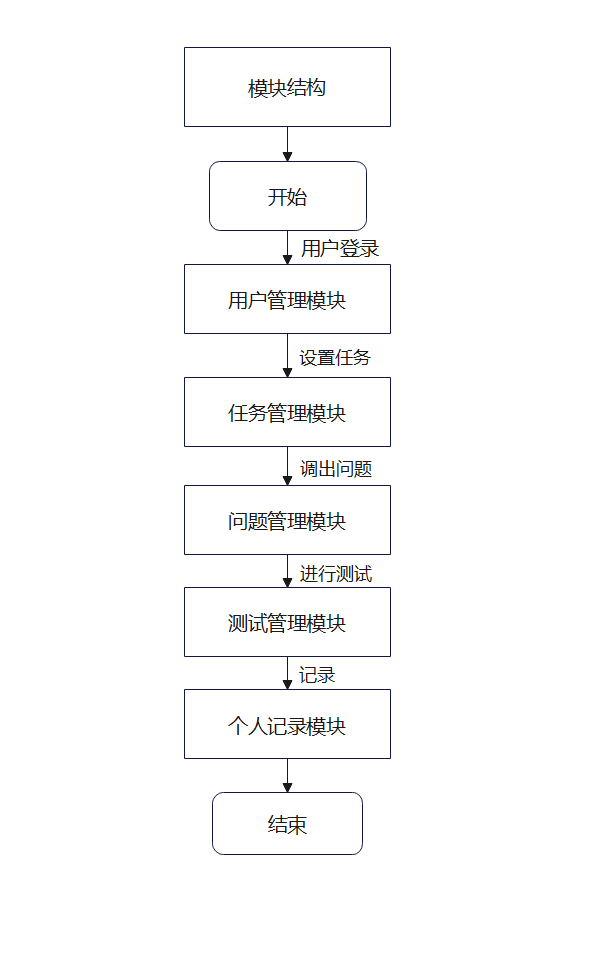
此外，考虑到用户可能遇到的登录问题，如忘记密码等情况，系统内建了高效的自助恢复流程。用户可以通过安全问题、备用邮箱或手机接收重置链接来迅速找回访问权限，整个过程既快捷又确保了账户的持续安全。

### 6.2.2数据分析

利用先进的数据分析技术，深入剖析学生的学习旅程，细致入微地追踪学生的学习成绩、作业提交情况、在线互动频次及质量，乃至他们在特定学科或技能点上的进展。通过时间序列分析，揭示学习趋势，明确进步与挑战的关键节点，为个性化教学策略的制定提供坚实基础。我们识别每位学生的独特学习模式，包括学生偏好的学习时段、资料类型及学习效率，确保教育资源精准对接其需求。更进一步，借助预测模型见学生未来的学习成效，及时预警潜在的学习障碍，并量身定制干预措施，如补充学习材料、一对一辅导安排，以维持并提升学习轨迹。

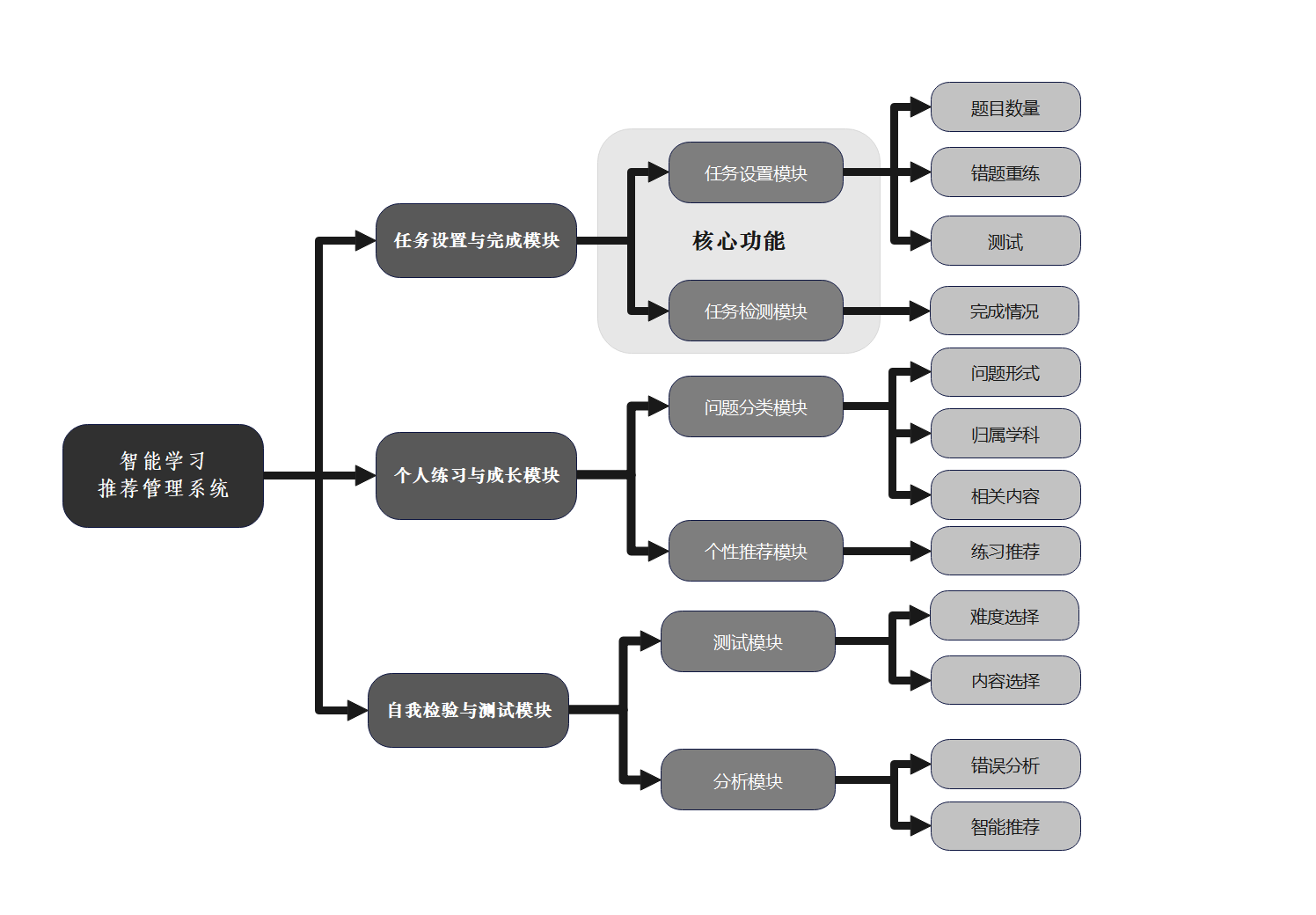
## 6.3软件系统总体功能/对象结构

### 6.3.1总体结构



用户首先需进行身份验证后登陆系统，为自己设置学习任务并进行完成。根据用户设定的学习目标，系统从题库中筛选出相关问题供用户练习和测试，并记录用户的学习成果。

### 6.3.2软件子系统功能/对象结构



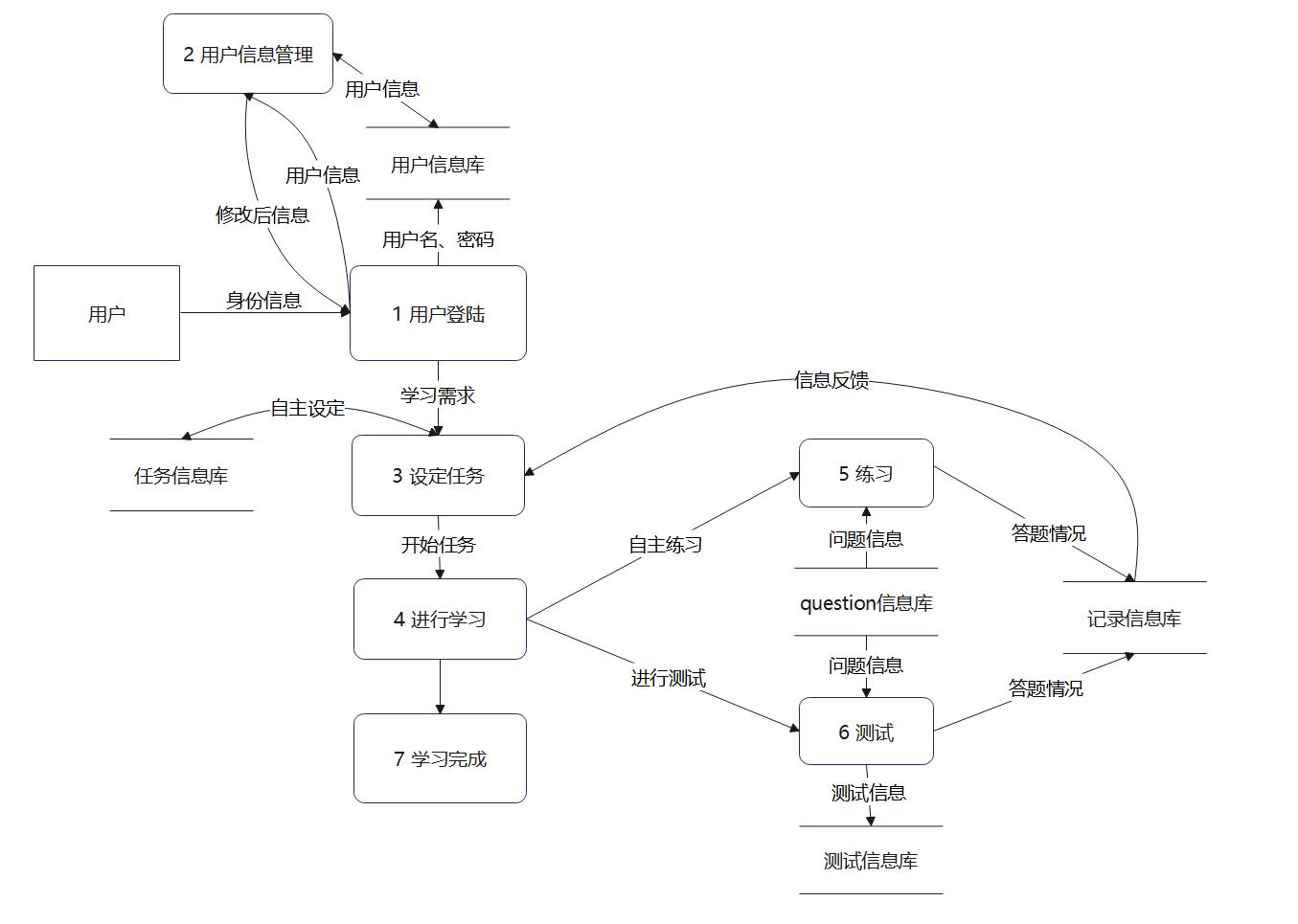
功能架构图

智能学习推荐管理系统分为三个主要模块，分别是任务设置与完成模块、个人练习与成长模块、自我检验与测试模块。

在任务设置与完成模块中，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再进行推荐，从而输出数据到界面显示给用户，并最终检测是否完成。

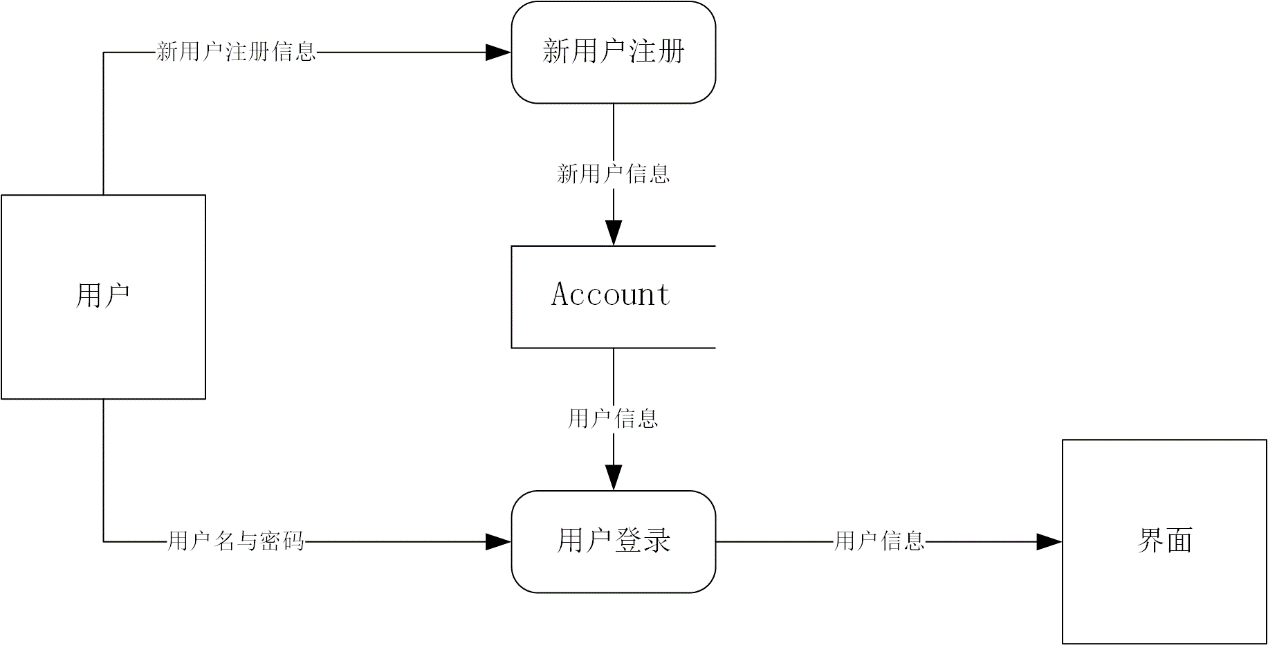
个人练习与成长模块是用户自主学习的过程，用户设置好任务后今日任务分析系统从mission数据库中获取当前的任务安排。然后系统自主进行任务分析，得到相关的分析信息。日常学习系统利用任务分析信息与题目信息，对题目进行筛选并进行巩固推荐。最后，作答分析系统获取用户的作答情况并进行详细的分析。

在自我检验与测试模块中，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。然后进行任务分析，得到相关的分析信息。作业测试系统结合用户的作答情况与自主选择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。



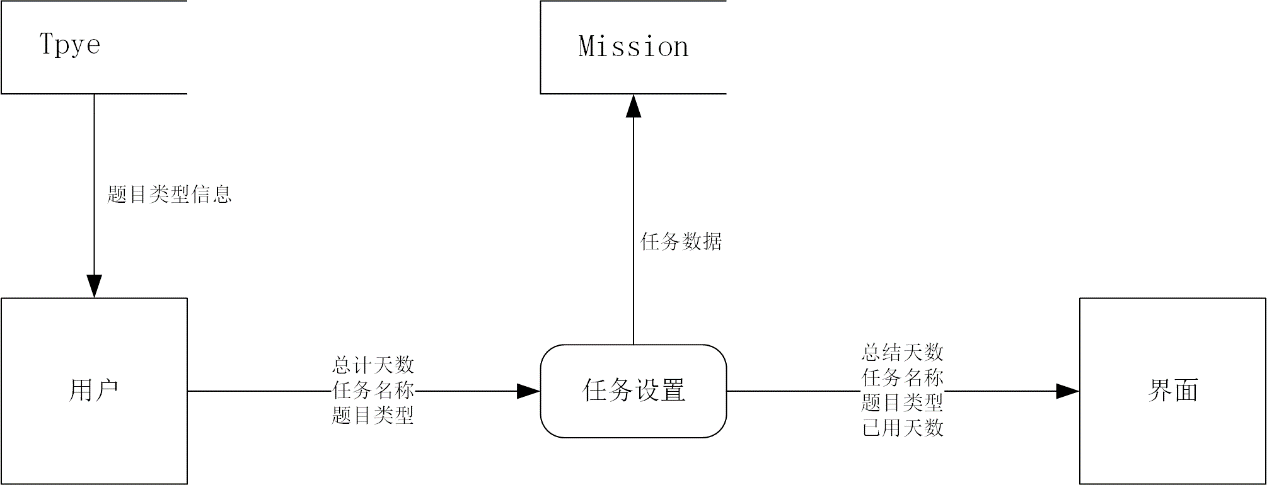
整体实现数据流图

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。



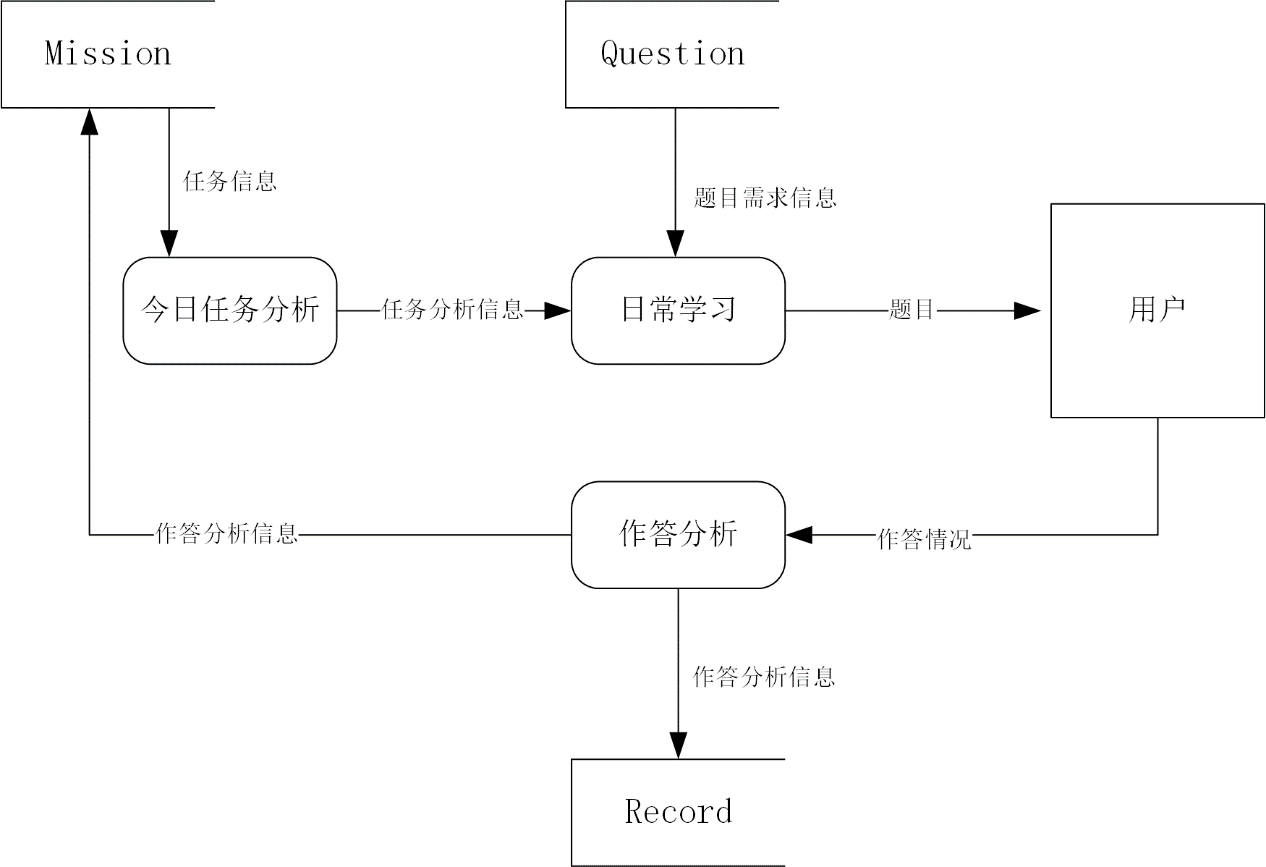
用户登陆

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。



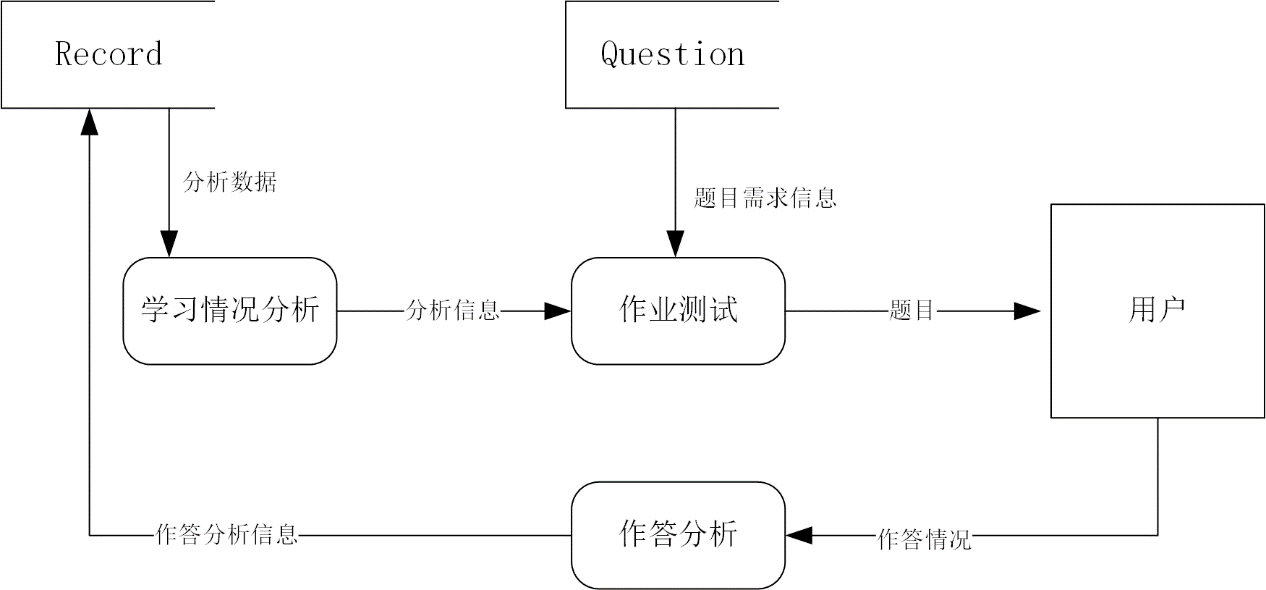
任务设置

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。



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。



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。