需求分析报告：面向学情分析的教育数据管理平台

1. **概述**

随着教育信息化的不断发展，各类教育软件在教学活动中的应用日益广泛。为了有效利用这些软件产生的海量数据，需要建立一个能够进行高效精准学情分析的教育数据管理平台。本平台将支持学情数据的集成、分析和反馈，以增强教育决策的质量和效率。

1. **角色定义**
2. 教务管理人员（云端用户）:

负责维护和管理学校教学数据的集成、审核与分析。

负责制定分析指标和推送分析结果到相关用户。

1. 教师（设备端用户）:

使用平台查看和分析所教授课程的学情数据。

使用平台查看学生在课程上的学习情况。

提出教学过程中对数据分析的具体需求。

1. 学生（设备端用户）:

访问个人学习情况报告和课程相关的学情分析。

利用分析结果进行自我学习调整和改进。

1. 系统管理员:

负责系统的技术维护、用户权限管理和数据安全。

1. **主要用例——参考用例图**
2. 数据采集与集成

主要参与者: 教务管理人员

用例描述: 教务管理人员从多个数据源采集教育过程数据，并进行清洗、标准化处理后集成到平台数据库中。（如选课情况、课堂参与度、考试成绩等）。数据通过自动化工具进行清洗、标准化和集成。

1. 生成与推送分析指标

主要参与者: 教务管理人员

用例描述: 教务管理人员利用集成的数据生成教学和学习相关的分析指标，并根据需求推送这些指标到教师或学生端。

1. 查看课程信息

主要参与者: 教师、学生

用例描述: 教师和学生可以查看相关课程的详细信息，包括课程内容、时间表等。学生可以查看自己在各课程的学习情况，如学习时长、作业完成情况等。

1. 查看课程指标

主要参与者: 教师

用例描述: 教师可以查看其教授课程的各种教学指标，如学生参与度、成绩分布等。

1. 查看学习情况报告

主要参与者: 学生

用例描述: 学生可以查看自己的学习情况报告，包括学习时长、作业完成情况等。

根据分析建议，学生可以优化自己的学习计划。

1. 提出指标需求

主要参与者: 教师

用例描述: 根据分析结果，教师可以在系统中提出新的数据分析指标需求，以便更好地监控和分析学情。

1. 系统维护与更新

主要参与者: 系统管理员

用例描述: 系统管理员负责系统的维护、更新和用户权限管理等工作。

1. 数据安全与备份

主要参与者: 系统管理员

用例描述: 系统管理员确保数据的安全性，执行定期数据备份和恢复操作。

1. 用例间的关系

「查看课程信息」和「查看课程指标」用例包含在「生成与推送分析指标」用例中，因为教师查看的指标是从生成的数据中提取的。

「查看学习情况报告」用例被「生成与推送分析指标」用例包含，学生接收的报告基于推送的分析指标。

「提出指标需求」用例扩展「查看课程指标」和「查看学习情况报告」用例，因为教师和学生基于查看结果可能需要更深入的分析。

1. **核心算法和智能化处理**
2. 智能指标分析（核心要点）：

根据不同用户的指标需求特点（学生、家长、教师、学院管理者等）对收集到的学情指标采取不同的指标分析策略（如对学生更注重分析学生的知识掌握情况，薄弱项等），同时也可以尝试根据用户指定的指标需求进行定制化的分析并分发给该用户。

1. 个性化推荐系统：

对教师和学生实施个性化的学情分析结果推荐，根据用户行为和偏好优化推送内容。

使用协同过滤和内容基推荐算法来实现精准定向的信息推送。

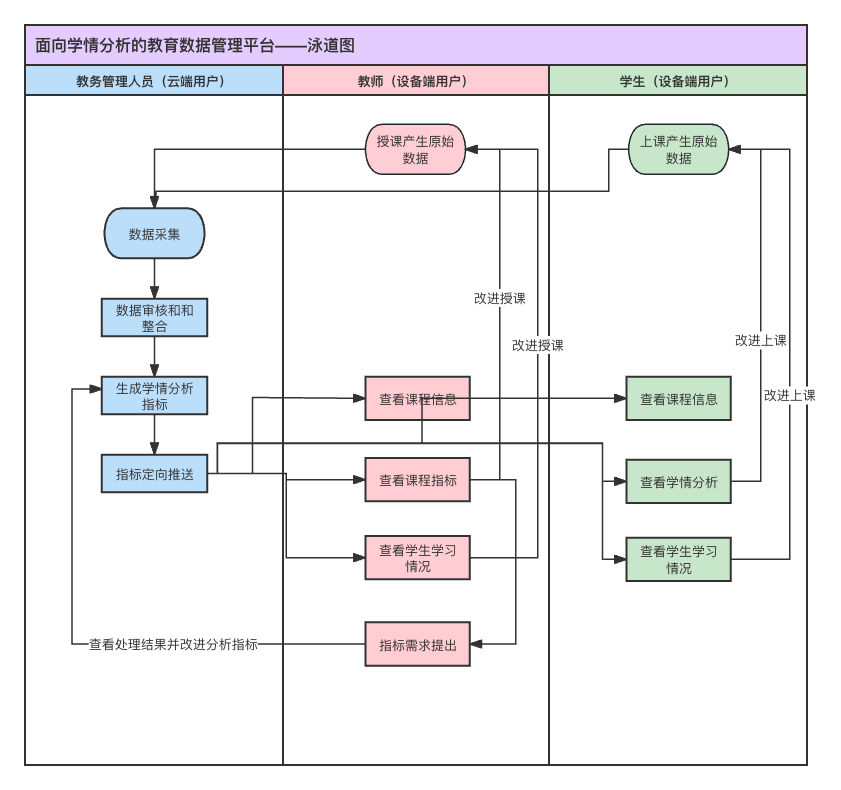
1. 智能化报告生成:

自动化生成可视化报告，包括图表和关键指标摘要，使得非技术用户也能轻松理解复杂数据。

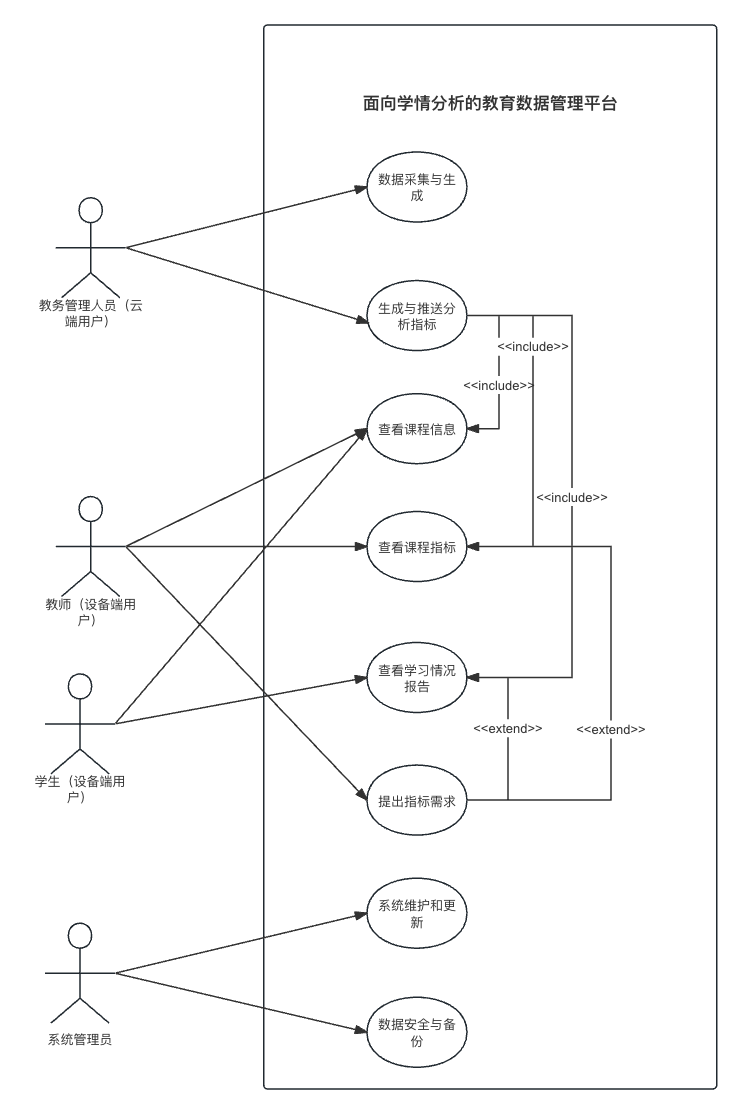
报告生成工具可支持自定义布局和展示方式，满足不同用户的具体需求。

**附录**

1. **带泳道的流程图**

****

1. **用例图**

****

软件架构文档：面向学情分析的教育数据管理平台

1. **核心算法和智能化处理**
2. 智能指标分析（核心要点）：

根据不同用户的指标需求特点（学生、家长、教师、学院管理者等）对收集到的学情指标采取不同的指标分析策略（如对学生更注重分析学生的知识掌握情况，薄弱项等），同时也可以尝试根据用户指定的指标需求进行定制化的分析并分发给该用户。

1. 个性化推荐系统：

对教师和学生实施个性化的学情分析结果推荐，根据用户行为和偏好优化推送内容。

使用协同过滤和内容基推荐算法来实现精准定向的信息推送。

1. 智能化报告生成:

自动化生成可视化报告，包括图表和关键指标摘要，使得非技术用户也能轻松理解复杂数据。

报告生成工具可支持自定义布局和展示方式，满足不同用户的具体需求。

1. **技术架构图描述**
2. 用户界面层（外部网络）

React: 用于构建用户界面的JavaScript库。

Antd: Ant Design，一个React UI库。

与应用层通过AJAX和HTTPS进行通信。

1. 应用层

Web 应用服务器（Tomcat）: 承载和运行Web应用，处理用户请求。

RMP 低代码平台API: 用于数据的获取和更改。

API 控制器: 处理HTTP请求并调用RMP平台的API进行业务处理。

前端控制器: 处理用户界面的交互逻辑。

1. 服务层: 实现所有业务逻辑，如数据分析、推荐算法、报告生成等。

认证和授权模块（JWT）: 负责用户认证和授权。

服务模块:

ETL: 数据提取、转换和加载服务。

Data Analysis: 数据分析服务。

Recommendation: 个性化推荐服务。

Report Generation: 报告生成服务。

1. 数据层

RMP 数据存储: RMP低代码平台基于的数据存储方式，用于存储结构化和非结构化数据。

包含以下资源对象：

Student，teacher，announcement，course，class\_course\_detail，course\_review，student\_course\_data，courseTeacher

1. 基础设施层

操作系统: 提供系统运行的基础环境（如Linux）。

硬件服务器: 提供计算和存储资源。

网络设备: 确保系统网络的正常运行，提供稳定的网络连接。

1. **技术架构各层功能详细描述**
2. 用户界面层

提供用户界面，供教务管理人员、教师和学生使用。

使用React和Antd构建用户界面，确保用户体验的友好性。

通过AJAX和HTTPS确保与应用层的安全通信。

1. 应用层

Web 应用服务器（如Tomcat）承载和运行Web应用，处理来自用户界面层的请求。

RMP 低代码平台API负责数据的获取和更改。

API 控制器处理HTTP请求并调用RMP平台的API进行业务处理。

前端控制器处理用户界面的交互逻辑。

1. 服务层：实现所有业务逻辑，包括数据分析、推荐算法和报告生成等。

认证和授权模块（JWT）确保系统的安全性，管理用户权限。

服务模块:

ETL负责数据的提取、转换和加载，确保数据质量和一致性。

Data Analysis利用机器学习模型进行学情分析，提供深度数据洞察。

Recommendation实现个性化推荐，满足教师和学生的个性化需求。

Report Generation自动生成可视化报告，使用户轻松理解数据。

1. 数据层

RMP 数据存储用于存储结构化和非结构化数据，支持RMP低代码平台的数据需求。

包含以下资源对象：

Student，teacher，announcement，course，class\_course\_detail，course\_review，student\_course\_data，courseTeacher。

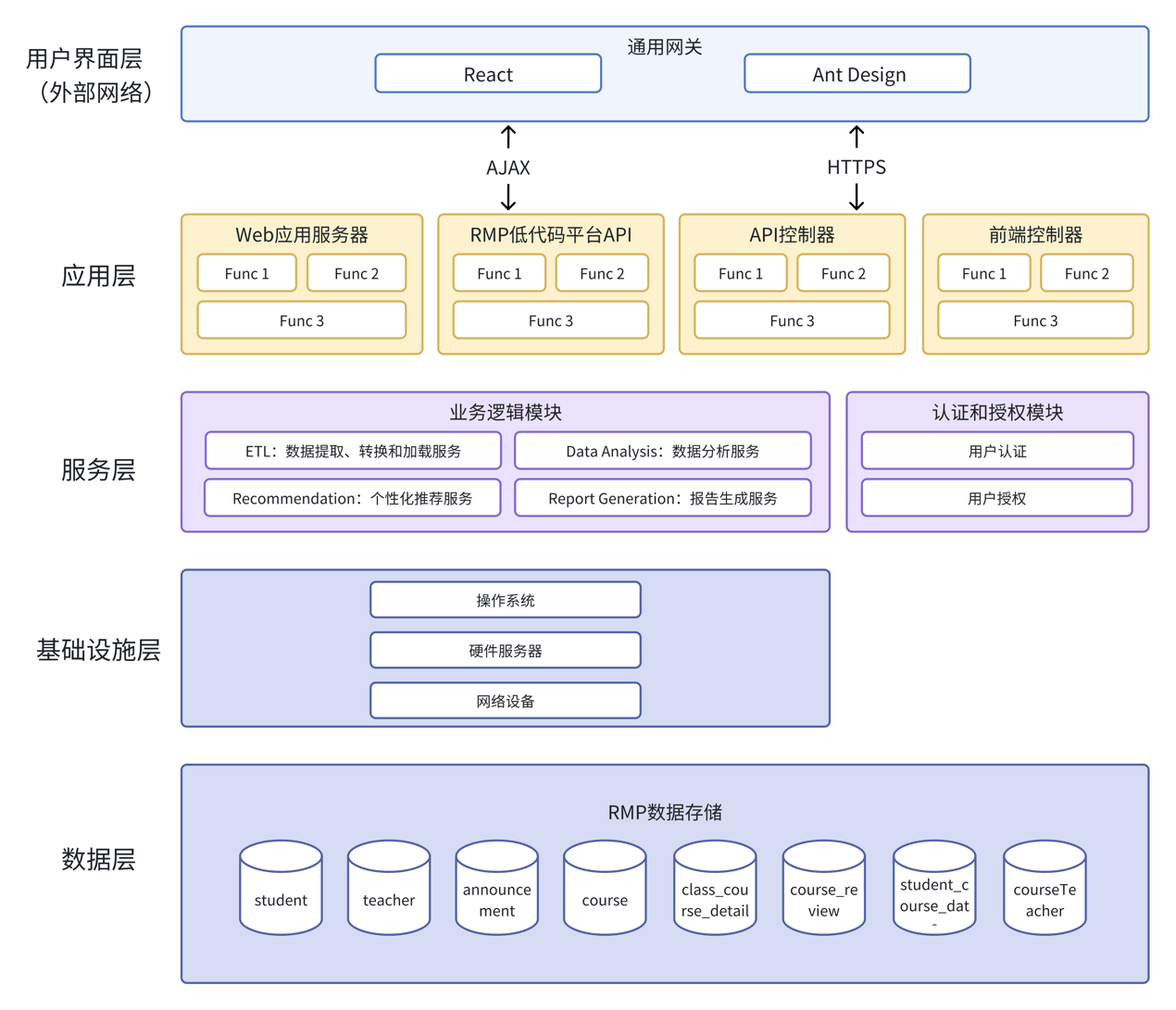
包含Student，Class和Teacher三个资源对象。

1. 基础设施层

操作系统提供系统运行的基础环境。

硬件服务器提供计算和存储资源，支持系统的运行。

网络设备确保系统网络的正常运行，提供稳定的网络连接。



实现内容文档：面向学情分析的教育数据管理平台

**1. 用户界面层**

1.1 技术栈

* React: 用于构建用户界面的JavaScript库。
* Ant Design (Antd): 一套基于React的UI组件库，用于快速构建用户界面。
* AJAX: 使用Axios库进行异步请求处理。
* HTTPS: 确保数据传输的安全性。

1.2 实现步骤

1. 项目初始化: 使用create-react-app初始化React项目。
2. 安装依赖
3. 创建页面和组件:
   * 登录页面
   * 仪表板页面
   * 学情分析页面
   * 课程信息页面
4. 组件开发:
   * 使用Ant Design组件构建表单、表格和图表。
   * 使用Axios与RMP API进行数据交互。
5. 状态管理: 使用React的useState和useEffect管理组件状态和生命周期。
6. 路由配置: 使用react-router-dom配置应用路由。

1.3 内容展示

**2. 应用层**

2.1 技术栈

* Spring Boot: 用于构建Java后端应用。
* Tomcat: 作为Web应用服务器。
* RMP低代码平台API: 用于数据的获取和更改。

2.2 实现步骤

1. 项目初始化: 使用Spring Initializr生成Spring Boot项目。
2. 配置依赖:
3. 实现API控制器:
   * 处理HTTP请求并调用RMP平台的API。
   * 使用RestTemplate或WebClient进行RMP API调用。
4. 前端控制器: 处理用户界面的交互逻辑。
   1. 内容展示

**3. 服务层**

3.1 技术栈

* Spring Boot: 用于构建服务模块。
* JWT: 用于用户认证和授权。
* ETL: 数据提取、转换和加载服务。
* 机器学习: 用于数据分析。
* 推荐系统: 实现个性化推荐。
* 报告生成: 自动化生成可视化报告。

3.2 实现步骤

1. 认证和授权:
   * 使用Spring Security和JWT进行认证和授权。
2. ETL:
   * 使用Spring Batch实现数据提取、转换和加载。
3. 数据分析:
   * 集成机器学习库，如TensorFlow或Weka，进行数据分析。
4. 推荐系统:
   * 实现协同过滤和内容基推荐算法。
5. 报告生成:
   * 使用JasperReports或其他报告生成工具。

3.3 内容展示

**4. 数据层**

4.1 技术栈

* RMP 数据存储: 用于存储结构化和非结构化数据。

4.2 实现步骤

1. 配置数据存储:
   * 根据RMP平台的配置文档，设置数据存储。
2. 定义资源对象:
   * 定义Student, Class, Teacher三个资源对象。

4.3 内容展示

**5. 基础设施层**

5.1 技术栈

* 操作系统: Linux。
* 硬件服务器: 提供计算和存储资源。
* 网络设备: 确保系统网络的正常运行。

5.2 实现步骤

1. 操作系统安装:
   * 安装和配置Linux操作系统。
2. 硬件配置:
   * 部署服务器硬件，配置存储和计算资源。
3. 网络配置:
   * 配置路由器、交换机等网络设备，确保网络连接稳定。

**组员分工**

|  |  |  |
| --- | --- | --- |
| 成员 | 分工情况 | 贡献比例 |
| 陈华昭 | 前端核心功能构建 | 25% |
| 蔡卓强 | 指标分析，GPT接入功能实现 | 25% |
| 杨征 | 数据收集，数据模型设计 | 25% |
| 康力烜 | 需求分析，架构设计与实现 | 25% |