系统设计说明书

1. 引言

1.1 目的

本系统设计说明书旨在为开发团队提供关于 **AetherNet 校园互助平台** 小程序项目的详细设计，包括功能模块、架构设计、数据库设计、交互逻辑及技术实现方案。此文档将为开发团队的编码工作提供明确的指导，并确保平台各项功能能够顺利实现。

1.2 范围

该平台项目初步将实现为一个 **校园互助小程序**，后续可根据需求扩展为 APP 版本。项目包括以下功能：

- 学生间的二手交易、代拿服务、拼车等校园服务；
- 内容审核与智能匹配机制；
- 用户信息管理、帖子发布与互动等社交功能。

本说明书将详细描述系统架构、功能模块设计、数据流与交互逻辑，并对每个模块的开发需求进行说明。

1.3 定义与缩写

- **AetherNet**：指校园互助平台的名称。
- **MVP**：最小可行产品，指该系统的初始版本，将支持二手交易、代拿服务、拼车等功能。
- **API**：应用程序接口，提供前后端的交互接口，确保数据传输的顺畅。
- **UI**：用户界面，指系统中所有与用户交互的界面元素。
- **AI 智能体**：指平台中用于内容审核（SafetyBot）、任务匹配（MatchBot）、智能问答（CampusQA）等模块的人工智能系统。

2.系统设计概述

2.1 系统目标与功能概述

AetherNet 系统的目标是通过简单、便捷的功能帮助学生在校园内进行二手交易、代拿服务、拼车等日常事务的处理。系统将通过智能化算法进行任务自动分配和内容审核，确保平台功能的高效性和内容的合规性。系统设计中包含以下模块：

- **用户管理模块**：负责学生注册、登录、个人资料管理等功能。

- **二手交易模块**：用户可发布二手商品信息，进行商品浏览、购买、交易等操作。
- **代拿服务模块**：学生可发布代拿服务请求，其他用户可选择接单。
- **智能审核模块**：使用 AI 技术进行内容审核，防止恶意、低俗内容的传播。
- **任务智能匹配模块**：系统根据用户需求与兴趣，智能匹配任务和服务。
- **社交模块**：用户可以评论、点赞、分享帖子，参与平台互动。

2.2 系统架构设计

AetherNet 平台的架构采用前后端分离的架构设计，确保系统的高效性与可扩展性。系统架构分为以下几个层次：

- **前端层**：使用 Taro 框架开发微信小程序，确保系统的跨平台兼容性；后期可扩展为 App 版本，使用 React Native 或 Flutter。
- **后端层**：使用 FastAPI 框架搭建后端服务，处理用户请求，提供数据处理、任务分配、内容审核等服务。
- **数据库层**：使用 MySQL 进行数据存储，Redis 用于缓存，Kafka 用于异步消息处理。
- **AI 智能体层**：包括 SafetyBot（内容审核）、MatchBot（任务匹配）、CampusQA（智能问答）等模块，负责任务分配、内容审核和推荐推送。

2.3 技术栈与工具

- **前端技术**：Taro、React Native、Flutter
- **后端技术**：FastAPI、MySQL、Redis、Kafka
- **AI 技术**：Python、TensorFlow、自然语言处理（NLP）
- **数据库**：MySQL
- **开发工具**：VSCode、Git、Docker、Swagger（API 文档）

3.功能模块设计

3.1 用户管理模块

- **功能**：实现学生的注册、登录、个人信息管理等功能。
- **设计**：
  - 用户通过微信账号或学号、邮箱进行注册，注册后进行身份验证。
  - 用户登录后可修改个人资料，查看发布记录、任务等。

- 支持用户角色管理，确保学生与管理员的权限分离。

3.2 二手交易模块

- **功能**：支持学生发布、浏览、购买二手商品。
- **设计**：
  - 用户可发布带有标题、内容、标签和图片的二手商品信息。
  - 用户可浏览商品，按照类别（如书籍、家电、衣物等）进行筛选。
  - 支持商品购买、留言、交流等功能。

3.3 代拿服务模块

- **功能**：学生可发布代拿服务需求，其他学生可选择接单。
- **设计**：
  - 用户可发布代拿请求，包含任务描述、酬劳、接单期限等信息。
  - 系统智能匹配合适的接单学生，并通知接单者。
  - 支持接单者评价任务，完成后任务状态更新。

3.4 内容审核模块（AI 智能体）

- **功能**：通过 AI 智能体对用户发布的内容进行审核，防止低俗或违法内容的传播。
- **设计**：
  - SafetyBot 会自动分析帖子内容，判断是否符合平台的发布规定。
  - 内容如果未通过审核，系统会返回审核失败原因，并要求用户修改。
  - 系统会记录每次审核结果，并允许管理员复审。

3.5 AI 推荐与任务匹配模块

- **功能**：根据用户兴趣、历史行为推荐相关任务和帖子。
- **设计**：
  - MatchBot 会分析用户行为与兴趣标签，智能推荐任务与内容。
  - 任务匹配的结果根据任务的类型、时间、紧急程度等因素进行排序。

3.6 标签生成与热门话题模块

- **功能**：根据用户发布的帖子自动生成标签，并根据标签热度生成热门话题。
- **设计**：

- 当用户发布帖子时，系统通过自然语言处理自动识别帖子中的关键词，生成相关标签。
- 当某个标签的帖子数量达到一定数量，系统会自动生成该标签的热门话题，并在平台首页展示。
- 用户可点击热门话题查看相关内容。

4.系统数据模型

4.1 数据库表设计

以下是平台的关键数据库表及其字段设计：

- **用户表（Users）**
  - 用户 ID（user_id）
  - 用户名（username）
  - 密码（password）
  - 邮箱（email）
  - 角色（role）
  - 注册时间（created_at）
- **帖子表（Posts）**
  - 帖子 ID（post_id）
  - 标题（title）
  - 内容（content）
  - 用户 ID（user_id，外键）
  - 发布时间（created_at）
  - 审核状态（status）
- **任务表（Tasks）**
  - 任务 ID（task_id）
  - 标题（title）
  - 描述（description）
  - 发布者 ID（user_id，外键）
  - 接单者 ID（assignee_id，外键）
  - 任务状态（status）
- **评论表（Comments）**

    - 评论 ID（comment_id）
    - 帖子 ID（post_id，外键）
    - 用户 ID（user_id，外键）
    - 评论内容（content）
    - 发布时间（created_at）
- **标签表（Tags）**
    - 标签 ID（tag_id）
    - 标签名称（tag_name）
    - 帖子 ID（post_id，外键）
- **审核日志表（ModerationLogs）**
    - 审核 ID（log_id）
    - 帖子 ID（post_id，外键）
    - 审核结果（decision）
    - 风险级别（risk_level）
    - 审核时间（created_at）

5.技术架构

5.1 系统架构

平台采用分布式架构，前端与后端分离，前端通过 Taro 框架开发，后端使用 FastAPI 与 MySQL 进行数据管理。系统支持 Web 端与小程序端的访问，未来支持移动端 App。

5.2 部署架构

平台将部署在阿里云或 AWS 云服务器上，使用 Docker 进行容器化部署，确保系统的可扩展性与高可用性。

6.安全性设计

6.1 数据加密

所有敏感数据（如密码、用户信息）均采用 AES 加密存储，确保数据安全。

6.2 用户认证

采用双重认证机制（如邮箱验证码、短信验证码）加强安全性。

7.测试计划

7.1 单元测试

进行单元测试，确保每个模块的功能正确性。

7.2 集成测试

进行模块间集成测试，确保各模块间的接口和数据流畅通。

7.3 性能测试

进行压力测试与负载测试，确保平台能在高并发情况下稳定运行。

8.风险管理

8.1 潜在风险

- 任务匹配不精准：AI 算法可能导致任务匹配的准确性低，影响用户体验。
- 内容审核不及时：自动化内容审核可能出现误判或漏判，影响平台健康发展。

8.2 对策

- 优化任务匹配算法，增加数据训练与模型优化。
- 增强内容审核机制，提供人工复核选项，确保审核质量。

结语

通过本系统设计说明书，我们为 AetherNet 校园互助平台的开发团队提供了清晰、详细的技术路线和功能设计。系统设计的各个层次与模块的描述将确保项目顺利开发并达到预期目标，为校园社区提供便捷、安全的服务平台。