

需求分析

参与开发者：张溪桐 汪李昀红 万沅昕

一、系统概述

1. 产品目标

开发一款高效、安全、易用的文本防伪软件，通过先进的文本识别与验证技术，快速判断文本是否由人完成。

2. 目标用户

学生，教师，作者，审稿人。

3. 产品范围

功能范围

核心功能：文本防伪验证：通过输入或扫描文本信息，快速验证产品真伪。

用户管理：支持用户注册、登录、个人信息管理及绑定。

二、总体设计

一、系统架构设计

AI 文本生成检测网站采用分层架构设计，主要分为用户界面层、业务逻辑层、数据存储层和接口层。这种分层架构有助于提高系统的可维护性、可扩展性和可测试性。

1. 用户界面层

功能：为用户提供全面的文本检测服务，支持多种设备和操作系统（如 iOS、Android、Web）。

实现方式：

移动端：使用微信小程序或原生应用开发，支持扫描、输入文本等功能。

Web 端：使用 HTML5、CSS3 和 JavaScript 框架（如 Vue.js 或 React.js）开发，提供企业用户管理后台。

特点：响应式设计，适配不同屏幕尺寸；简洁的用户交互流程，降低用户学习成本。

2. 业务逻辑层

功能：处理系统的业务逻辑，包括文本检测、用户管理等。

模块划分：

文本检测模块：接收用户输入或扫描的文本信息，进行分析和检测，返回检测结果。

用户管理模块：处理用户注册、登录、个人信息管理等功能，支持用户绑定文本。

数据统计与分析模块：收集用户行为数据，生成统计报告，支持企业决策。

技术选型：使用 Node.js 或 Python 作为后端开发语言，结合 Express.js 或 Flask 框架。

3. 数据存储层

功能：存储系统的数据，包括检测信息、用户信息等。

数据库选择：

关系型数据库：MySQL 用于存储结构化数据，如用户信息、检测信息等。

非关系型数据库：Redis 用于缓存高频访问数据，提升系统性能。

特点：数据加密存储，确保信息安全；分布式存储设计，支持高并发访问。

4. 接口层

功能：提供系统内部模块间通信的接口，以及与外部系统的接口。

实现方式：

内部接口：使用 RESTful API 设计，模块间通过 HTTP 请求通信。

外部接口：提供 API 文档，支持与其他系统（如 ERP、电商平台）集成。

特点：接口标准化，易于扩展和维护。

二、模块划分

1. 用户管理模块

功能：用户注册、登录、个人信息管理、权限设置。

2. 文本检测模块

功能：接收文本信息，进行分析和检测，返回检测结果。

3. 数据统计与分析模块

功能：收集用户行为数据，生成统计报告。

三、技术选型

1. 前端

移动端：微信小程序开发框架（WXML、WXSS、JavaScript）。

Web 端：Vue.js 或 React.js 框架，结合 HTML5、CSS3。

2. 后端

开发语言：Python。

框架：Flask（Python）。

数据库：MySQL（关系型数据库），Redis（缓存数据库）。

3. 接口

API 设计：RESTful API，支持 JSON 格式数据交互。

4. 安全

加密技术：SHA-256 用于数据传输加密，AES 用于数据存储加密。

身份验证：JWT（JSON Web Tokens）用于用户身份验证。

四、数据流程

1. 用户登录：

- 用户通过客户端输入账号和密码。
- 系统后端验证用户信息，生成 JWT 并返回给客户端。
- 客户端存储 JWT，用于后续请求的身份验证。

2. 文本检测：

- 用户通过客户端输入或扫描文本信息。
- 客户端将文本信息发送到后端的 `/api/detect/text` 接口。
- 后端进行文本分析和检测，返回检测结果到客户端。
- 客户端展示检测结果给用户。

3. 数据统计与分析：

- 客户端将用户行为数据（如检测次数、查询时间等）
- 后端收集并存储用户行为数据。
- 企业用户可以通过 Web 端后台查询统计报告。

五、系统特点

- 高效性：**采用先进的文本分析和检测算法，确保文本检测的高效性。
- 安全性：**数据加密存储和传输，采用 JWT 进行身份验证，确保用户数据和检测信息的安全。
- 易用性：**简洁的用户界面设计，支持多种设备，降低用户操作难度。
- 可扩展性：**分层架构设计，支持未来功能模块的扩展和技术升级。
- 兼容性：**支持多种操作系统和设备，确保用户在不同平台上都能使用。

三、功能需求

文本上传与粘贴功能：用户能够通过上传文档或直接粘贴文本的方式提交需要检测的内容。支持多种文档格式，如 TXT、DOC、PDF 等。

AI 生成概率评估：核心功能是对文本进行分析，判断其是由 AI 生成的概率，并以直观的方式（如百分比、颜色标记等）呈现结果。

多语言支持：能够检测多种语言的文本，满足全球用户的需求。

详细报告生成：除了给出 AI 生成的概率外，还能提供详细的分析报告，包括文本特征、可能的 AI 模型来源等信息，帮助用户深入理解检测结果。

语法与拼写检查：附加功能，用于检测文本中的语法、拼写和标点错误，并提出改进建议，提升文本质量。

文本改写建议：对于检测出的 AI 生成文本，提供改写建议或自动优化功能，使其更贴近人类写作风格。

内容总结：能够将长文本压缩为简短摘要，保留核心信息，方便用户快速获取文本主要内容。

剽窃检查：对比文本与在线资源，检测内容是否重复，确保文本的原创性。

批量检测：支持一次性上传多个文档或长篇幅文本进行检测，提高工作效率。

API 接口与集成能力：提供 API 接口，方便开发者将检测功能集成到其他应用程序中，实现更多定制化功能。

四、用户体验需求

简洁直观的界面：网站设计应简洁明了，操作流程简单易懂，降低用户的学习成本和使用难度。

快速检测：能够在短时间内完成检测并返回结果，提高用户体验。

隐私保护：严格保护用户的隐私，不存储用户的检测文本和结果，确保数据安全。

多平台支持：兼容多种操作系统和设备，包括 Windows、macOS、iOS、Android 等，确保用户在不同平台上都能使用。

五、技术需求

前端技术：采用响应式设计，使用 HTML5、CSS3 和 JavaScript 框架（如 Vue.js 或 React.js）开发，确保在不同设备上的良好显示和交互体验。

后端技术：使用 Python 或 Node.js 作为后端开发语言，结合 Express.js 或 Flask 框架，处理文本检测逻辑和业务流程。

机器学习与 AI 算法：核心检测功能依赖于先进的机器学习模型和 AI 算法，通过对大量人类写作和 AI 生成文本的学习，不断提高检测的准确率。

数据库：使用关系型数据库（如 MySQL）存储用户信息、检测结果等结构化数据，使用非关系型数据库（如 MongoDB）存储文本数据等非结构化数据。

六、安全需求

数据加密：对用户上传的文本和检测结果进行加密存储和传输，防止数据泄露和被非法访问。

身份验证：采用安全的身份验证机制，如 JWT（JSON Web Tokens），确保用户身份的真实性和合法性。