**系统设计**

**参与开发者：张溪桐 汪李昀虹 万沅昕**

1. **系统概述**
2. 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。

1. 目标用户

学生，教师，作者，审稿人。

1. 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。

1. **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：

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。

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。

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：

外部接口：提供API文档，支持与其他系统集成。

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

1. 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。

2. 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。

3. 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。

4. 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。

5. 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。

1. **功能设计**

**用户管理模块：**用户注册、登录、个人信息管理、权限设置。

**文本检测模块：**接收文本，分析检测，返回结果。

**数据统计与分析模块：**收集用户行为数据，生成统计报告。

1. **接口设计**

**外部接口设计：**提供RESTful API文档，支持与其他系统集成。

**内部接口设计：**模块间通过HTTP请求通信，标准化接口。

1. **性能指标设计**

**响应时间：**每千字检测1 - 3秒内完成。

**资源利用率：**高负载下CPU使用率≤80%，单次检测内存≤512MB。

**准确性**：AI生成文本检测准确率≥90%。