

教师文档管理系统设计说明书

文档历史（完成人：系统开发团队）

版本更新信息

时间	版本号	更新人	主要更新内容
2024-10-18	V1.0	开发团队	初始版本，完成核心功能设计
2024-10-18	V2.0	开发团队	增加备份恢复功能，完善系统架构

1 概述

1.1 编写目的

本文档的编写目的是：详细定义教师文档管理软件的总体功能；给出系统的结构设计和过程设计，作为程序编写的依据。

1.2 参考资料

a. 项目来源：基于Django框架的教师文档管理系统开发项目 b. 本文档中引用到的规范和资料：

- Django官方文档 (<https://docs.djangoproject.com/>)
- Python Web开发最佳实践
- MySQL数据库设计规范
- Bootstrap前端框架文档 c. 列出这些规范和资料的作者、编号、标题、发表日期、出版单位或资料来源：
- Django Software Foundation, Django Documentation, 2024
- Bootstrap Team, Bootstrap Documentation, 2024

1.3 术语和缩写词

术语/缩写词	中文译名	英文全称	定义
Django	Django框架	Django Web Framework	Python Web开发框架
MVC	模型-视图-控制器	Model-View-Controller	软件架构模式
ORM	对象关系映射	Object-Relational Mapping	数据库操作抽象层
API	应用程序接口	Application Programming Interface	软件组件间通信接口
URL	统一资源定位符	Uniform Resource Locator	网络资源地址标识

2 需求概述

教师文档管理系统是一个基于Web的文档管理平台，主要服务于教育机构的教师用户。系统采用“教师-管理员”双层角色设计，具有明确的身份校验流程和功能边界。

2.1 系统特性

- 用户管理：支持教师和管理员两种角色，具备完整的用户生命周期管理
- 文档管理：支持多种文件格式上传，提供版本控制和分类管理
- 分享功能：支持文档分享链接生成，具备访问控制和权限管理
- 系统管理：提供系统配置、数据备份恢复、操作日志等管理功能
- 安全控制：基于角色的权限控制，完整的操作审计日志

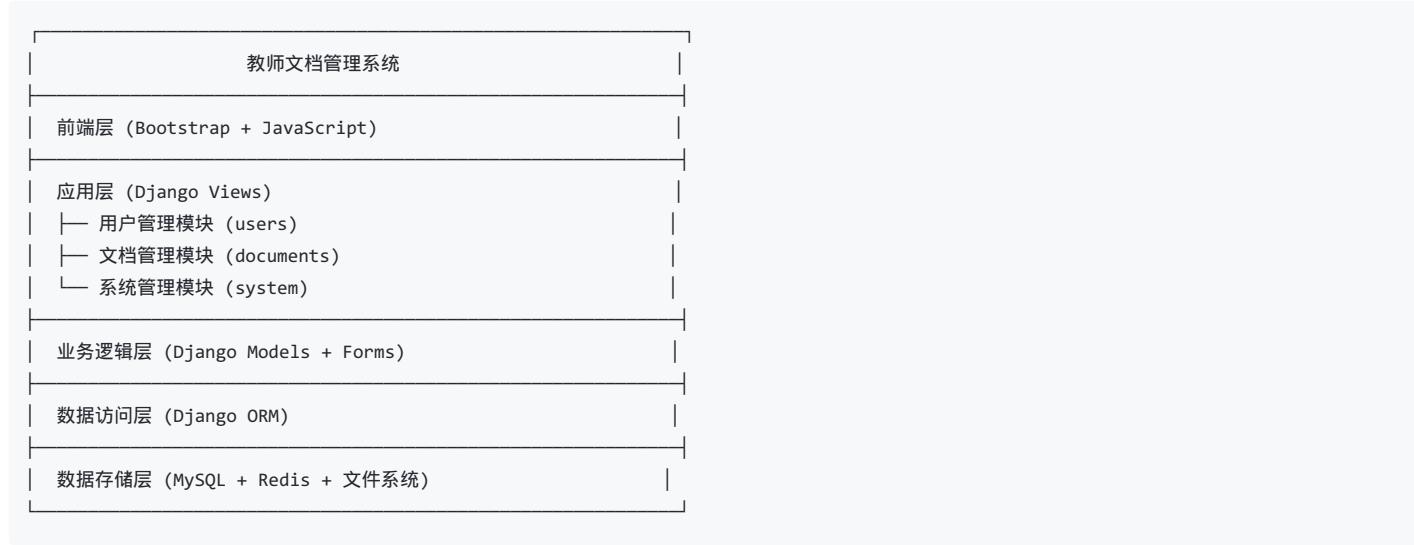
2.2 核心需求

- 用户认证系统：统一登录页面，支持工号和用户名登录，角色自动识别
- 文档管理：文档上传、分类、搜索、版本控制、权限控制
- 分享功能：生成限时分享链接，支持密码保护和下载次数限制
- 系统管理：用户管理、系统配置、数据备份恢复、操作日志

3 总体结构设计

3.1 对外接口设计

3.1.1 系统架构图



3.1.2 外部系统接口

- **数据库接口**：MySQL数据库连接，支持事务处理和连接池
- **缓存接口**：Redis缓存服务，提供会话存储和任务队列
- **文件存储接口**：本地文件系统存储，支持媒体文件管理
- **邮件接口**：SMTP邮件服务，支持通知和密码重置

3.2 内部结构设计

3.2.1 架构说明

系统采用分层架构设计：

1. 表示层 (Presentation Layer)
 - 技术：HTML5 + CSS3 + JavaScript + Bootstrap
 - 功能：用户界面展示，用户交互处理
 - 特点：响应式设计，支持多设备访问
2. 控制层 (Controller Layer)
 - 技术：Django Views + URL路由
 - 功能：请求处理，业务逻辑协调
 - 特点：基于类的视图，权限控制中间件
3. 业务逻辑层 (Business Logic Layer)
 - 技术：Django Models + Forms
 - 功能：业务规则实现，数据验证
 - 特点：ORM映射，表单验证
4. 数据访问层 (Data Access Layer)
 - 技术：Django ORM + MySQL
 - 功能：数据持久化，查询优化
 - 特点：数据库抽象，查询优化

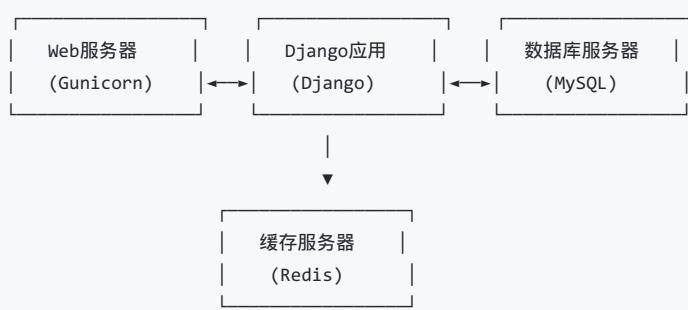
3.2.2 包图设计

```

teacher_doc_system/
├── teacher_doc_system/          # 项目配置包
│   ├── settings.py              # 项目设置
│   ├── urls.py                 # 主URL配置
│   ├── wsgi.py                  # WSGI配置
│   └── celery.py                # Celery配置
└── users/                      # 用户管理包
    ├── models.py                # 用户模型
    ├── views.py                 # 用户视图
    ├── forms.py                 # 用户表单
    ├── urls.py                  # 用户URL
    └── middleware.py            # 权限中间件
├── documents/                  # 文档管理包
    ├── models.py                # 文档模型
    ├── views.py                 # 文档视图
    ├── forms.py                 # 文档表单
    └── urls.py                  # 文档URL
└── system/                     # 系统管理包
    ├── models.py                # 系统模型
    ├── views.py                 # 系统视图
    ├── forms.py                 # 系统表单
    ├── urls.py                  # 系统URL
    ├── tasks.py                 # 异步任务
    └── utils.py                 # 工具函数
└── templates/                  # 模板包
    ├── base/                    # 基础模板
    ├── users/                   # 用户模板
    ├── documents/               # 文档模板
    └── system/                  # 系统模板

```

3.2.3 组件图



3.3 出错处理设计

3.3.1 错误分类

1. 系统错误：数据库连接失败、服务器内部错误
2. 业务错误：权限不足、数据验证失败
3. 用户错误：输入格式错误、文件类型不支持

3.3.2 错误处理策略

- 用户友好提示：所有错误都有中文提示信息
- 日志记录：系统错误自动记录到日志文件
- 错误恢复：提供重试机制和降级方案
- 安全考虑：敏感错误信息不暴露给用户

3.4 安全保密设计

3.4.1 身份认证

- 密码加密：使用Django内置的PBKDF2算法
- 会话管理：基于Cookie的会话控制
- 登录限制：支持登录失败次数限制

3.4.2 权限控制

- 角色权限：基于角色的访问控制（RBAC）
- 资源权限：细粒度的资源访问控制
- 操作审计：完整的用户操作日志记录

3.4.3 数据安全

- 数据加密：敏感数据加密存储
- 备份策略：定期数据备份和恢复测试
- 访问控制：文件访问权限控制

4 类的详细设计

4.1 CustomUser类

4.1.1 描述

用户模型类，继承自Django的AbstractUser，扩展了教师文档管理系统的用户属性，支持教师和管理员两种角色。

4.1.2 属性

属性名	类型	说明
role	CharField	用户角色（admin/teacher）
employee_id	CharField	工号，唯一标识
department	CharField	所属部门
avatar	ImageField	用户头像
last_login_ip	GenericIPAddressField	最后登录IP
storage_quota	BigIntegerField	存储配额（字节）
storage_used	BigIntegerField	已用存储（字节）
is_frozen	BooleanField	是否冻结
frozen_reason	TextField	冻结原因
frozen_at	DateTimeField	冻结时间
password_changed_at	DateTimeField	密码修改时间
must_change_password	BooleanField	必须修改密码

4.1.3 公有方法

方法名	方法说明
is_admin()	判断是否为管理员
is_teacher()	判断是否为教师
can_upload_file(size)	检查是否可以上传指定大小的文件
get_storage_usage_percentage()	获取存储使用百分比
get_full_name()	获取用户全名

4.1.4 私有方法

方法名	方法说明
_calculate_storage_usage()	计算存储使用量

方法名	_update_storage_quota()	方法说明	更新存储配额
-----	-------------------------	------	--------

4.2 Document类

4.2.1 描述

文档核心信息模型，管理文档的基本信息、文件存储、权限控制等。

4.2.2 属性

属性名	类型	说明
title	CharField	文档标题
category	ForeignKey	所属分类
description	TextField	文档描述
file	FileField	文档文件
file_size	BigIntegerField	文件大小(字节)
file_type	CharField	文件类型
file_hash	CharField	文件哈希值
author	ForeignKey	作者
status	CharField	状态(draft/review/published/archived)
is_public	BooleanField	是否公开
is_starred	BooleanField	是否星标
download_count	PositiveIntegerField	下载次数
view_count	PositiveIntegerField	查看次数

4.2.3 公有方法

方法名	方法说明
can_be_accessed_by(user)	检查用户是否可以访问文档
increment_view_count()	增加查看次数
increment_download_count()	增加下载次数
get_file_url()	获取文件访问URL
get_absolute_url()	获取文档详情URL

4.2.4 私有方法

方法名	方法说明
_generate_file_hash()	生成文件哈希值
_validate_file_type()	验证文件类型
_update_storage_usage()	更新存储使用量

4.3 ShareLink类

4.3.1 描述

分享链接模型，管理文档的分享链接生成、访问控制、过期管理等。

4.3.2 属性

属性名	类型	说明
document	ForeignKey	关联文档
token	CharField	分享令牌
password	CharField	访问密码
expires_at	DateTimeField	过期时间
max_downloads	PositiveIntegerField	最大下载次数
download_count	PositiveIntegerField	已下载次数
is_active	BooleanField	是否有效
created_by	ForeignKey	创建人

4.3.3 公有方法

方法名	方法说明
is_expired	检查是否已过期
is_available	检查是否可用
can_access(password)	检查是否可以访问
generate_token()	生成分享令牌
increment_download()	增加下载次数

4.3.4 私有方法

方法名	方法说明
_validate_access()	验证访问权限
_check_expiry()	检查过期状态

5 用例实现的详细设计

5.1 用户登录用例

5.1.1 功能说明

用户通过工号或用户名登录系统，系统验证用户身份并分配相应权限。

5.1.2 界面设计

- 登录页面：包含用户名/工号输入框、密码输入框、登录按钮
- 验证机制：实时验证输入格式，显示错误提示
- 安全特性：支持验证码、登录失败次数限制

5.1.3 参与类

- LoginView：处理登录请求
- CustomUser：用户模型
- LoginForm：登录表单
- AuthenticationMiddleware：认证中间件

5.1.4 交互设计

用户 → LoginView → CustomUser → AuthenticationMiddleware → 系统首页

5.2 文档上传用例

5.2.1 功能说明

教师用户上传文档文件，系统进行文件验证、存储配额检查、文件处理等操作。

5.2.2 界面设计

- 上传页面：文件选择器、文档信息表单、上传按钮
- 进度显示：上传进度条、文件大小显示
- 验证提示：文件类型验证、大小限制提示

5.2.3 参与类

- UploadDocumentView：处理上传请求
- Document：文档模型
- DocumentForm：文档表单
- CustomUser：用户模型

5.2.4 交互设计

用户 → UploadDocumentView → DocumentForm → Document → CustomUser → 文档列表

5.3 文档分享用例

5.3.1 功能说明

教师用户为文档创建分享链接，设置访问密码、过期时间、下载次数等参数。

5.3.2 界面设计

- 分享设置页面：密码设置、过期时间选择、下载次数限制
- 链接生成：显示生成的分享链接、二维码
- 链接管理：分享链接列表、状态管理

5.3.3 参与类

- CreateShareLinkView：创建分享链接
- ShareLink：分享链接模型
- ShareLinkForm：分享链接表单
- Document：文档模型

5.3.4 交互设计

用户 → CreateShareLinkView → ShareLinkForm → ShareLink → Document → 分享链接

5.4 系统备份用例

5.4.1 功能说明

管理员创建系统数据备份，包括数据库和媒体文件，支持备份恢复功能。

5.4.2 界面设计

- 备份创建页面：备份名称、描述、备份类型选择
- 备份列表页面：备份历史、状态显示、操作按钮
- 恢复确认页面：备份信息展示、恢复确认

5.4.3 参与类

- CreateBackupView：创建备份
- Backup：备份模型
- BackupForm：备份表单
- RestoreBackupView：恢复备份

5.4.4 交互设计

管理员 → CreateBackupView → BackupForm → Backup → 备份文件 → 备份列表

6 数据库设计

6.1 用户表 (users_customuser)

字段名	含义	类型(长度)	默认值	允许空	主键	外键	备注
id	用户ID	INT(11)	AUTO_INCREMENT	NO	√		主键
username	用户名	VARCHAR(150)		NO			唯一
employee_id	工号	VARCHAR(20)		NO			唯一
first_name	名	VARCHAR(150)		YES			
last_name	姓	VARCHAR(150)		YES			
email	邮箱	VARCHAR(254)		YES			
role	角色	VARCHAR(20)	'teacher'	NO			admin/teacher
department	部门	VARCHAR(100)		YES			
storage_quota	存储配额	BIGINT(20)	10737418240	NO			字节
storage_used	已用存储	BIGINT(20)	0	NO			字节
is_frozen	是否冻结	TINYINT(1)	0	NO			
frozen_reason	冻结原因	TEXT		YES			
created_at	创建时间	DATETIME(6)		NO			
updated_at	更新时间	DATETIME(6)		NO			

6.2 文档表 (documents_document)

字段名	含义	类型(长度)	默认值	允许空	主键	外键	备注
id	文档ID	INT(11)	AUTO_INCREMENT	NO	√		主键
title	标题	VARCHAR(255)		NO			
category_id	分类ID	INT(11)		YES		√	外键到 documents_documentcategory
description	描述	TEXT		YES			
file	文件路径	VARCHAR(100)		NO			
file_size	文件大小	BIGINT(20)		NO			字节
file_type	文件类型	VARCHAR(20)		NO			
file_hash	文件哈希	VARCHAR(64)		NO			唯一
author_id	作者ID	INT(11)		NO	√		外键到users_customuser

字段名	状态含义	类型(长度)	默认值	允许空	主键	外键	备注
is_public	是否公开	TINYINT(1)	0	NO			
is_starred	是否星标	TINYINT(1)	0	NO			
download_count	下载次数	INT(10) UNSIGNED	0	NO			
view_count	查看次数	INT(10) UNSIGNED	0	NO			
created_at	创建时间	DATETIME(6)		NO			
updated_at	更新时间	DATETIME(6)		NO			

6.3 分享链接表 (system_sharelink)

字段名	含义	类型(长度)	默认值	允许空	主键	外键	备注
id	链接ID	INT(11)	AUTO_INCREMENT	NO	√		主键
document_id	文档ID	INT(11)		YES		√	外键到documents_document
token	分享令牌	VARCHAR(64)		NO			唯一
password	访问密码	VARCHAR(128)		YES			
expires_at	过期时间	DATETIME(6)		NO			
max_downloads	最大下载次数	INT(10) UNSIGNED	0	NO			
download_count	已下载次数	INT(10) UNSIGNED	0	NO			
is_active	是否有效	TINYINT(1)	1	NO			
created_by_id	创建人ID	INT(11)		YES		√	外键到users_customuser
created_at	创建时间	DATETIME(6)		NO			

6.4 备份表 (system_backup)

字段名	含义	类型(长度)	默认值	允许空	主键	外键	备注
id	备份ID	INT(11)	AUTO_INCREMENT	NO	√		主键
name	备份名称	VARCHAR(200)		NO			
description	备份描述	TEXT		YES			
status	状态	VARCHAR(20)	'pending'	NO			pending/running/completed/failed
file_path	文件路径	VARCHAR(500)		YES			
file_size	文件大小	BIGINT(20)		YES			字节
created_by_id	创建人ID	INT(11)		YES		√	外键到users_customuser
created_at	创建时间	DATETIME(6)		NO			
completed_at	完成时间	DATETIME(6)		YES			
error_message	错误信息	TEXT		YES			

字段名	含义	类型(长度)	默认值	允许空	主键	外键	备注
7 跨职能业务流程图							

7.1 文档管理流程

```
教师用户 → 登录系统 → 选择文档分类 → 上传文档 → 填写文档信息 → 保存文档 → 文档列表
↓
管理员 → 审核文档 → 发布文档 → 文档公开可见
```

7.2 分享链接流程

```
教师用户 → 选择文档 → 创建分享链接 → 设置访问参数 → 生成分享链接 → 分享给他人
↓
外部用户 → 访问分享链接 → 输入密码 (如需要) → 下载文档
```

7.3 系统备份流程

```
管理员 → 创建备份 → 选择备份类型 → 执行备份任务 → 备份完成 → 备份列表
↓
管理员 → 选择备份 → 确认恢复 → 执行恢复任务 → 恢复完成
```

8 数据流图

8.1 系统数据流图

```
用户输入 → 前端界面 → Django视图 → 业务逻辑 → 数据库
↓
数据库 → 业务逻辑 → Django视图 → 前端界面 → 用户输出
```

8.2 文档上传数据流

```
文件选择 → 文件验证 → 存储配额检查 → 文件存储 → 数据库记录 → 用户反馈
```

8.3 分享链接数据流

```
分享请求 → 权限验证 → 链接生成 → 数据库存储 → 链接返回 → 用户使用
```

9 类图

9.1 核心类关系图

```
CustomUser (用户)
├── 1:N → Document (文档)
├── 1:N → ShareLink (分享链接)
└── 1:N → Backup (备份)

Document (文档)
├── N:1 → DocumentCategory (分类)
├── 1:N → DocumentVersion (版本)
├── 1:N → DocumentOperationLog (操作日志)
└── 1:N → ShareLink (分享链接)

ShareLink (分享链接)
├── N:1 → Document (文档)
└── N:1 → CustomUser (创建人)

Backup (备份)
└── N:1 → CustomUser (创建人)
```

9.2 系统架构类图

```
Django Framework
├── Models (数据模型)
│   ├── CustomUser
│   ├── Document
│   ├── ShareLink
│   └── Backup
├── Views (视图控制)
│   ├── UserViews
│   ├── DocumentViews
│   └── SystemViews
├── Forms (表单验证)
│   ├── UserForms
│   ├── DocumentForms
│   └── SystemForms
└── Templates (模板渲染)
    ├── UserTemplates
    ├── DocumentTemplates
    └── SystemTemplates
```

文档结束