**Технологии:** react, Django, PostgreSQL.

**1. Общее описание системы:**

ECM-система предназначена для управления электронными документами в рамках строительных проектов.

Основные функции:

* Создание, хранение и версионирование документов (чертежи, сметы, акты).
* Группировка документов по проектам и категориям.
* Управление правами доступа.
* Логирование действий пользователей.
* Автоматическая генерация уникальных номеров документов в рамках проекта.

2. Схема базы данных (PostgreSQL):

Таблицы и их назначение:

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Поля (ключевые) | Ограничения | Описание |
| Departments | id (PK), name (UNIQUE) | Name уникально | Подразделения компании (например, "Проектный отдел"). |
| Roles | id (PK), name (UNIQUE) | Name уникально | Роли пользователей (например, "Инженер", "Архивариус"). |
| Users | id (PK), email (UNIQUE), role\_id (FK) | role\_id ссылается на roles.id | Пользователи. Привязаны к ролям и подразделениям. |
| Projects | id (PK), code (UNIQUE), status | status CHECK: active, completed, paused | Проекты/стройки. Поле code для уникальной идентификации (например, "PRJ-2023-001"). |
| Categories | id (PK), parent\_category\_id (FK) | Иерархия через parent\_category\_id → categories.id | categories.id Категории документов (например, "Чертежи → Архитектурные"). |
| Documents | id (PK), project\_id (FK), doc\_number | UNIQUE (project\_id, doc\_number) | Основная информация о документах. Номер (doc\_number) уникален в проекте. |
| DocumentsVersions | id (PK), document\_id (FK), version\_number | UNIQUE (project\_id, doc\_number) | История версий документов. |
| ActivityLog | id (PK), action\_type, details (JSONB) | action\_type CHECK: create, edit, approve, delete | Логирование действий. В details хранится контекст (например, "Утверждено пользователем X"). |
| Permissions | user\_id (PK, FK), document\_id (PK, FK) | Составной PK | Права доступа к документам (read, edit, manage). |

**3. Ограничения и правила:**

**Уникальные номера документов:**

В рамках одного проекта (project\_id) номер документа (doc\_number) должен быть уникальным.

Пример: DRW-001 в проекте PRJ-2023-001.

**Версионирование:**

Каждое изменение документа создает новую запись в document\_versions. Поле version в documents обновляется автоматически.

**Каскадное удаление:**

Удаление проекта → удаление всех связанных документов.

Удаление документа → удаление его версий и прав доступа.

**Права доступа:**

Роли (roles) определяют глобальные права (например, "Архивариус" может удалять документы).

Таблица permissions задает доступ к конкретным документам.

**4. Принцип работы системы:**

**Создание документа:**

Пользователь загружает файл (PDF, CAD и т.д.), указывает проект, категорию и тип (например, "Чертеж").

Система генерирует уникальный doc\_number (например, DRW-001).

Документ сохраняется во внешнем хранилище (S3/MinIO), путь записывается в file\_path.

**Редактирование и версионирование:**

При изменении документа создается новая запись в document\_versions.

Поле version в documents увеличивается на 1.

**Утверждение документа:**

При смене статуса на "approved" в activity\_log добавляется запись с action\_type = approve.

**Доступ:**

Проверка прав через permissions и roles. Например, только "Руководитель проекта" может утверждать документы.

**5. Пример использования:**

**Сценарий:**

Инженер создает чертеж для проекта "Строительство моста".

**Действия:**

Выбирает проект PRJ-2023-005, категорию "Чертежи/Несущие конструкции".

Загружает файл bridge\_design\_v1.dwg.

**Система:**

Присваивает номер DRW-001.

Сохраняет файл в хранилище (s3://.../PRJ-2023-005/DRW-001\_v1.dwg).

Записывает действие в activity\_log:

json

{

"action\_type": "create",

"details": {"author": "user\_123", "comment": "Первая версия чертежа"}

}

**Утверждение:**

Руководитель проекта меняет статус документа на "approved".

В activity\_log добавляется запись с action\_type = approve.

**6. Что важно учесть:**

Хранение файлов: Используйте внешнее хранилище (S3, MinIO) + шифрование при необходимости.

Индексы: Добавьте индексы на часто используемые поля (project\_id, status, timestamp).

Триггеры: Автоматизируйте генерацию doc\_number и обновление версий.

API: Реализуйте эндпоинты для CRUD операций, проверки прав и аудита.

**СТРАНИЦЫ:**

**2. Документы:**

Что будет:

* Таблица с документами: название, номер, проект, категория, статус, автор, дата создания.
* Фильтры: по проекту, статусу («черновик», «на согласовании», «утвержден»), категории.
* Поиск: по названию, номеру, ключевым словам.
* Действия: загрузка новой версии, утверждение, скачивание, удаление.
* Детальная страница документа: версии, комментарии, история изменений, права доступа.

**3. Проекты:**

Что будет:

* Список проектов: название, код, статус («активен», «завершен»), сроки.
* Карточка проекта: документы проекта, участники, сроки, прогресс.
* Фильтры: по статусу, дате начала.
* Действия: создание проекта, редактирование, архивация.

**4. Пользователи и права:**

Что будет:

* Список пользователей: имя, роль, подразделение, email.
* Ролевая модель: настройка прав для ролей («инженер», «руководитель», «архивариус»).
* Индивидуальные разрешения: доступ к документам/проектам для конкретных пользователей.

**5. Журнал действий:**

Что будет:

* Таблица логов: пользователь, действие («создал», «изменил», «утвердил»), документ, время.
* Фильтры: по типу действия, пользователю, дате.
* Экспорт: в CSV/Excel.