**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение   
высшего образования

«Национальный исследовательский Томский политехнический университет»

|  |  |
| --- | --- |
| Школа | Инженерная школа информационных технологий и робототехники |
| Обеспечивающее подразделение | Отделение информационных технологий |
| Направление подготовки / специальность | 09.03.02 Информационные системы и технологии |
| Образовательная программа (направленность (профиль)) | Информационно-аналитические системы и технологии в бизнесе |

**ОТЧЕТ О ПРАКТИКЕ**

|  |  |
| --- | --- |
| Вид практики | Учебная практика |
| Тип практики | Технологическая (проектно-технологическая) практика |
| Место практики | г.Томск, Отделение информационных технологий, ТПУ |

|  |  |
| --- | --- |
| Выполнил обучающийся | Косинов Константин Анатольевич |
| Группа | 8И31 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись обучающегося)

Руководитель практикиТПУ:

Старший преподаватель ОИТ ИШИТР А. С. Гончаров

(степень, звание, должность)

Дата проверки «\_\_\_»\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Допустить / не допустить к защите

Подпись \_\_\_\_\_\_\_\_\_\_\_\_

Итоговая оценка по практике

(традиционная оценка, балл)

Томск 2025

Оглавление

[Введение](#Введение)

[Постановка цели и задач](#Постановкацелиизадач)

[Анализ предметной области](#Анализпредметнойобласти)

[Разработка требований](#Разработкатребований)

[Физическая схема базы данных](#Физическаясхемабазыданных)

[Диаграмма вариантов использования (use case) в нотации UML](#Диаграммавариантовиспользования)

[Описание бизнес-процессов в нотации BPMN](#Описаниебизнеспроцессов)

[Макет приложения](#Макетприложения)

Ход работы

**Введение**

Нередко в компаниях документы хранятся не централизованно, без администрирования, некоторые документы становятся неактуальными, иногда отсутствует маркировка документов по отделам, что затрудняет внутрикорпоративный поток документов. «Облачное хранилище документов», представляющее собой веб-приложение для облачного хранения документов, предназначено для решения этих проблем: документы хранятся централизованно, существует администрирование, уничтожение (удаление) неактуальных документов, а также маркировка документов по отделам.

**Постановка цели и задач.**

Цель: получение работоспособного веб-приложения для хранения документов компании

Задачи:

1. Проанализировать предметную область и получить макет веб-интерфейса пользователя и бизнес-логику
2. Разработать структуру базы данных для приложения.
3. Разработать серверную часть приложения с использованием выбранного стека технологий.
4. Разработать клиентскую часть веб-приложения с использованием выбранного стека технологий.

**Анализ предметной области.**

На данный момент большие компании имеют собственные хранилища (Data Warehouse, DWH) или пользуются услугами сервисов облачного хранения (reg.ru, VK Data Lakehouse и т.д.), но далеко не все компании могут позволить себе это. «Облачное хранилище документов» как раз рассчитано на такие компании, имея возможность предоставлять более доступные услуги, в связи с простотой обслуживания и импортозамещенностью.

**Разработка требований.**

Акторы:

1. Неавторизованный пользователь
2. Авторизованный пользователь (работник)
3. Администратор

Функциональные требования:

1. Система должна позволять пользователю загружать, хранить документы: .pdf, .docx, .xlsx -файлы.

Предусловия:

* Пользователь должен запустить приложение на своем устройстве.
* Пользователь должен быть зарегистрирован в системе.
* Пользователь должен быть авторизован в системе.

Основной поток:

1. Пользователь переходит на домашнюю страницу.
2. Пользователь нажимает кнопку «+».
3. Пользователь загружает файл документа в появившемся меню.
4. Пользователь указывает название и отделы.
5. Пользователь нажимает кнопку «добавить».

Постусловие:

1. Появление нового документа в хранилище.
2. Система должна позволять всем пользователям иметь доступ ко всем документам, хранящимся в системе.

Предусловия:

* Пользователь должен запустить приложение на своем устройстве.
* Пользователь должен быть зарегистрирован в системе.
* Пользователь должен быть авторизован в системе.

Основной поток:

1. Пользователь может с помощью поиска в верху и/или перехода на страницу отдела скачать любой документ.

Постусловие:

1. Скачивание документа пользователем.
2. Система должна позволять администраторам удалять документы, имеющиеся в системе.

Предусловия:

* Пользователь должен запустить приложение на своем устройстве.
* Пользователь должен быть зарегистрирован в системе как администратор.
* Пользователь должен быть авторизован в системе.

Основной поток:

1. Администратор переходит на домашнюю страницу.
2. Администратор может с помощью поиска и/или перехода на страницу отдела удалить любой документ.

Постусловие:

1. Удаление документа администратором из хранилища.
2. Система должна позволять пользователю, который загружает документы, определять их к каким-либо отделам (HR, Head, Finances и т.д.).

Предусловия:

* Пользователь должен запустить приложение на своем устройстве.
* Пользователь должен быть зарегистрирован в системе как администратор.
* Пользователь должен быть авторизован в системе.

Основной поток:

1. Пользователь переходит на домашнюю страницу.
2. Пользователь нажимает кнопку «+».
3. Пользователь загружает файл документа в появившемся меню.
4. Пользователь указывает название и отделы.
5. Пользователь нажимает кнопку «добавить».

Постусловие:

1. Наличие тэга отдела у документа.
2. Система должна позволять пользователю производить поиск по названию, отделу среди документов.

Предусловия:

* Пользователь должен запустить приложение на своем устройстве.
* Пользователь должен быть зарегистрирован в системе как администратор.
* Пользователь должен быть авторизован в системе.

Основной поток:

1. Пользователь с любой страницы может воспользоваться поиском по документам в верху страницы.
2. Система должна позволять пользователю запрашивать удаление файла у администратора.

Предусловия:

* Пользователь должен запустить приложение на своем устройстве.
* Пользователь должен быть зарегистрирован в системе как администратор.
* Пользователь должен быть авторизован в системе.

Основной поток:

1. С помощью поиска по документам и/или перехода на страницу отдела пользователь может нажать кнопку «запросить удаление» у нужного документа.

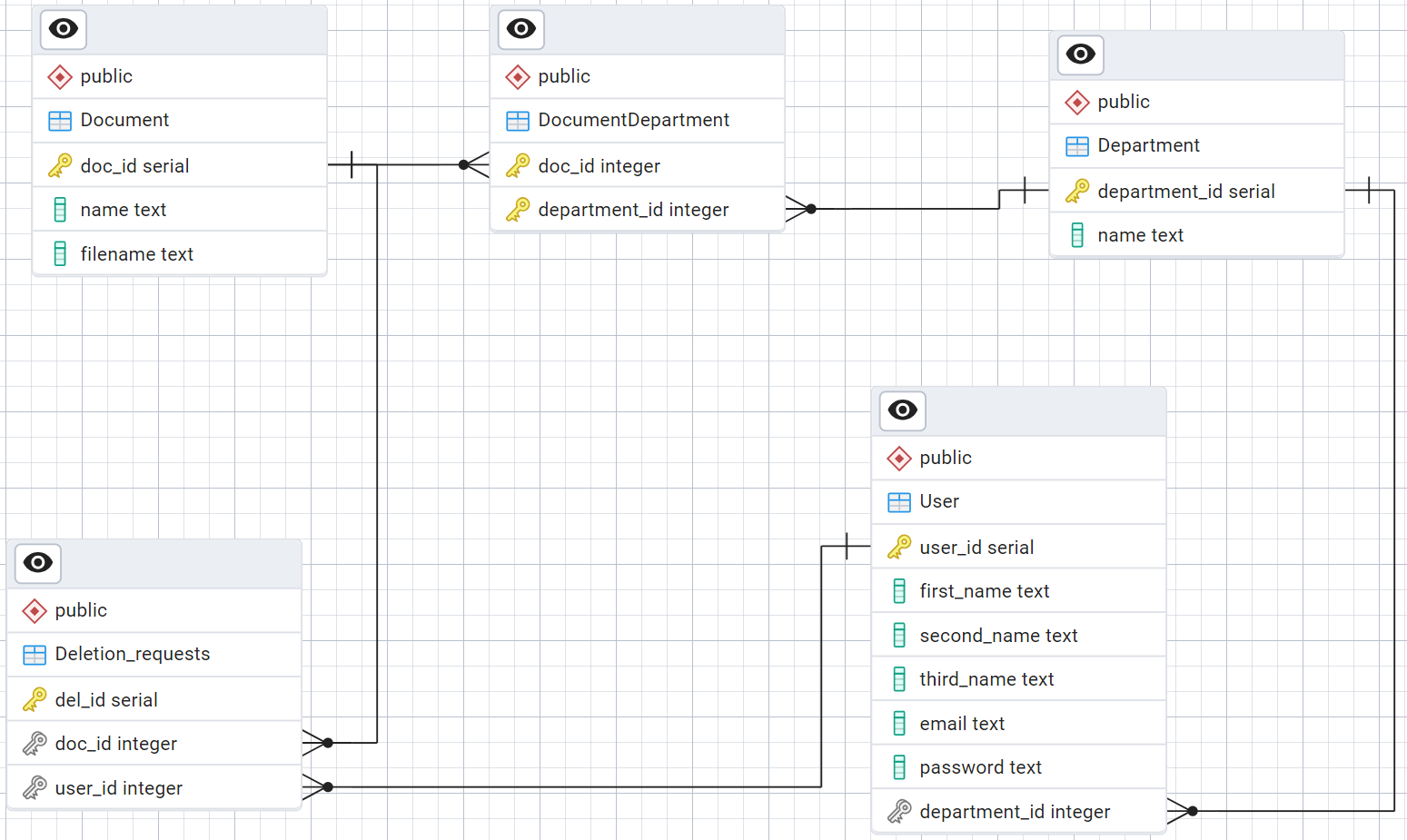
Постусловия:

1. Наличие запроса на удаление файла в хранилище.

Системные требования:

1. Бекэнд системы должен быть написан на Javascript.
2. Фронтэнд системы должен быть написан на Vue.
3. Фронтэнд должен иметь адаптивный дизайн.
4. Система должна хранить документы.
5. Каждый пользователь должен относиться к какому-либо отделу.

**Физическая схема базы данных.**



База данных с примерами:

BEGIN;

CREATE TABLE IF NOT EXISTS public.User

(

user\_id serial NOT NULL,

first\_name text NOT NULL,

second\_name text NOT NULL,

third\_name text,

email text NOT NULL,

password text NOT NULL,

department\_id integer,

PRIMARY KEY (user\_id)

);

insert into public.User values (default, 'Громова', 'Мария', 'Матвеевна', 'grom@corp.ru', '$2b$10$wrmUUUhh9wBj4Zce8osQOOOYXy8/m53f5iWYBBOhY4jAte41Qqouq', 4);

insert into public.User values (default, 'Попов', 'Евгений', 'Андреевич', 'head@corp.ru', '$2b$10$wrmUUUhh9wBj4Zce8osQOOoWURzJvdBPLC5/axdegSZyJHkdiF/lO', 2);

insert into public.User values (default, 'Ильин', 'Евгений', 'Васильевич', 'ilev@corp.ru', '$2b$10$wrmUUUhh9wBj4Zce8osQOO0WIJOa2YL.gIHlUyJQUzFgHk8chIbny', 1);

CREATE TABLE IF NOT EXISTS public.Department

(

department\_id serial NOT NULL,

name text NOT NULL,

PRIMARY KEY (department\_id)

);

insert into public.Department values (default, 'HR');

insert into public.Department values (default, 'Руководство');

insert into public.Department values (default, 'Финансы');

insert into public.Department values (default, 'Администратор');

CREATE TABLE IF NOT EXISTS public.Deletion\_requests

(

del\_id serial NOT NULL,

doc\_id integer NOT NULL,

user\_id integer NOT NULL,

PRIMARY KEY (del\_id)

);

insert into public.Deletion\_requests values (default, 2, 2);

insert into public.Deletion\_requests values (default, 16, 3);

insert into public.Deletion\_requests values (default, 11, 2);

CREATE TABLE IF NOT EXISTS public.Document

(

doc\_id serial NOT NULL,

name text NOT NULL,

filename text NOT NULL,

PRIMARY KEY (doc\_id)

);

insert into public.Document values (default, 'График отпусков 2024', 'Отпуска2024.docx');

insert into public.Document values (default, 'Счет на оплату №567', 'Счет567.docx');

insert into public.Document values (default, 'Стратегический план', 'План.docx');

insert into public.Document values (default, 'Годовой бюджет 2024', 'Бюджет2024.docx');

insert into public.Document values (default, 'Приказ о приеме на работу', 'Приказ о приеме на работу.docx');

insert into public.Document values (default, 'Акт сверки с контрагентом', 'Акт сверки.docx');

insert into public.Document values (default, 'Должностная инструкция', 'Должност инструкция.docx');

insert into public.Document values (default, 'Годовой отчет СовДиру', 'ОтчетСовДиру.docx');

insert into public.Document values (default, 'Бухгалтерский баланс', 'БухБаланс.docx');

insert into public.Document values (default, 'Положение об оплате труда', 'Оплата труда.docx');

insert into public.Document values (default, 'Протокол собрания директоров', 'Протокол СовДир.docx');

insert into public.Document values (default, 'Отчет о движении денег', 'Отчет о движ денег.docx');

insert into public.Document values (default, 'Штатное расписание', 'ШтатнРаспис.docx');

insert into public.Document values (default, 'Политика конфиденциальности', 'Политика конфиденциальности.docx');

insert into public.Document values (default, 'Налоговая декларация', 'НалогДеклар.docx');

insert into public.Document values (default, 'Заявление на отпуск', 'Заявление на отпуск.docx');

insert into public.Document values (default, 'Устав предприятия', 'Устав.docx');

insert into public.Document values (default, 'Счет-фактура №78-Ф', '78Ф.docx');

insert into public.Document values (default, 'План развития на 2024-2025', 'План20242025.docx');

insert into public.Document values (default, 'Акт о нарушении дисциплины', 'Акт о наруш дисциплины.docx');

insert into public.Document values (default, 'Трудовой договор ИТ-0001', 'ИТ0001.docx');

CREATE TABLE IF NOT EXISTS public.DocumentDepartment

(

doc\_id integer NOT NULL,

department\_id integer NOT NULL,

PRIMARY KEY (doc\_id, department\_id)

);

insert into public.DocumentDepartment values (1, 1);

insert into public.DocumentDepartment values (2, 2);

insert into public.DocumentDepartment values (2, 3);

insert into public.DocumentDepartment values (3, 2);

insert into public.DocumentDepartment values (4, 3);

insert into public.DocumentDepartment values (5, 1);

insert into public.DocumentDepartment values (6, 2);

insert into public.DocumentDepartment values (7, 1);

insert into public.DocumentDepartment values (8, 1);

insert into public.DocumentDepartment values (8, 3);

insert into public.DocumentDepartment values (9, 3);

insert into public.DocumentDepartment values (10, 1);

insert into public.DocumentDepartment values (11, 2);

insert into public.DocumentDepartment values (12, 3);

insert into public.DocumentDepartment values (13, 1);

insert into public.DocumentDepartment values (14, 1);

insert into public.DocumentDepartment values (14, 2);

insert into public.DocumentDepartment values (14, 3);

insert into public.DocumentDepartment values (15, 3);

insert into public.DocumentDepartment values (16, 1);

insert into public.DocumentDepartment values (17, 1);

insert into public.DocumentDepartment values (17, 2);

insert into public.DocumentDepartment values (17, 3);

insert into public.DocumentDepartment values (18, 3);

insert into public.DocumentDepartment values (19, 2);

insert into public.DocumentDepartment values (20, 1);

insert into public.DocumentDepartment values (21, 1);

ALTER TABLE IF EXISTS public.User

ADD CONSTRAINT Department\_User\_department\_id FOREIGN KEY (department\_id)

REFERENCES public.Department (department\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public.Deletion\_requests

ADD CONSTRAINT Document\_Deletion\_requests\_doc\_id FOREIGN KEY (doc\_id)

REFERENCES public.Document (doc\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public.Deletion\_requests

ADD CONSTRAINT User\_Deletion\_requests\_user\_id FOREIGN KEY (user\_id)

REFERENCES public.User (user\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public.DocumentDepartment

ADD CONSTRAINT Document\_DocumentDepartment\_doc\_id FOREIGN KEY (doc\_id)

REFERENCES public.Document (doc\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public.DocumentDepartment

ADD CONSTRAINT Document\_DocumentDepartment\_department\_id FOREIGN KEY (department\_id)

REFERENCES public.Department (department\_id) MATCH SIMPLE

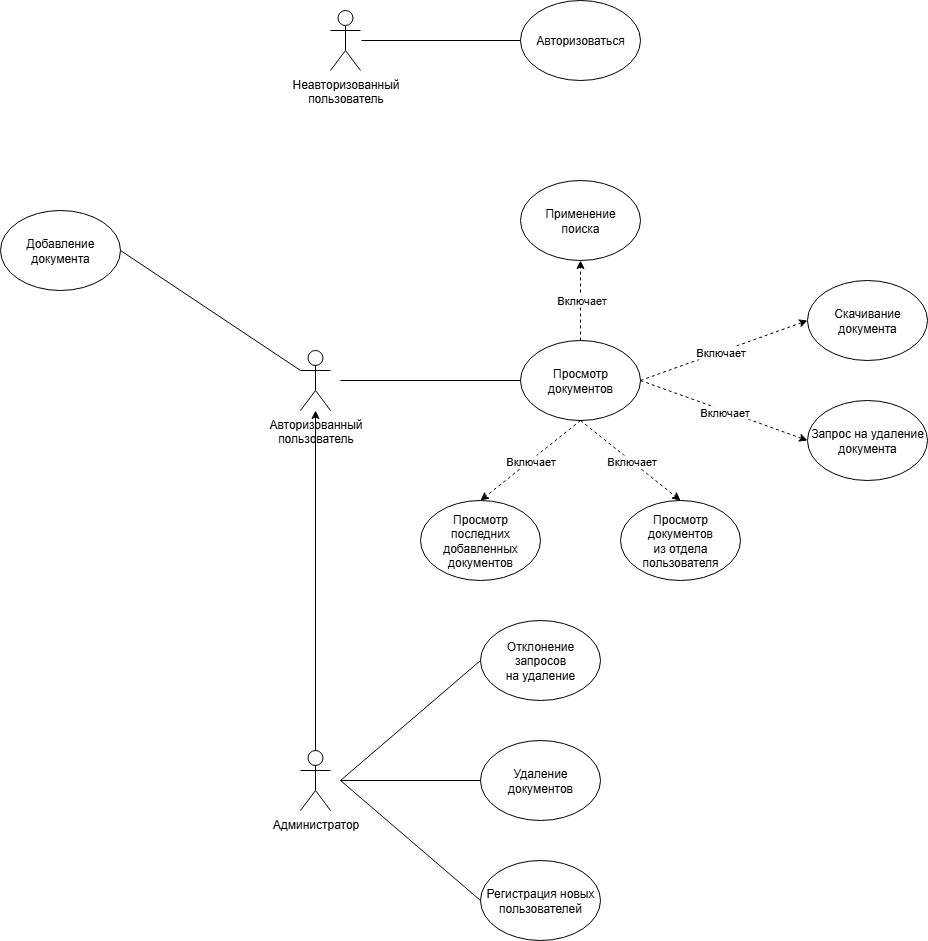
ON UPDATE NO ACTION

ON DELETE NO ACTION

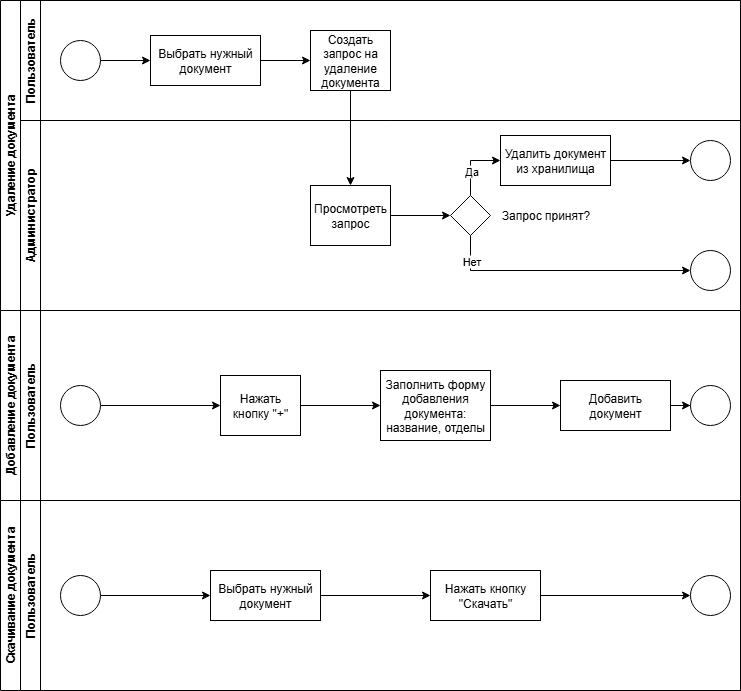
NOT VALID;

END;

**Диаграмма вариантов использования (use case) в нотации UML.**



**Описание бизнес-процессов в нотации BPMN.**



**Макет приложения.**

В качестве макета приложения представляется проект Figma, доступный по ссылке <https://www.figma.com/design/Ml3oANePFSzq14dS6U898e/cloud?node-id=0-1&t=49odxVmBgUp39BzA-1>

**Разработка приложения.**

Для просмотра приложения представляется проект на GitHub по ссылке <https://github.com/RenderAlien/Cloud>

Заключение

Список источников