

Орлов Артём Николаевич

обучающийся, БГТУ «ВОЕНМЕХ»

им Д.Ф. Устинова, Санкт-Петербург,

Россия

ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ КАДРОВОГО ДОКУМЕНТООБОРОТА

В статье представлены результаты проектирования веб-приложения для автоматизации смешанного кадрового документооборота промышленного предприятия. Цель состоит в формировании согласованного комплекса проектных решений, включающего требования, пользовательские роли, архитектуру, структуру базы данных, сценарии взаимодействия и пользовательский интерфейс. На основе анализа предметной области сформированы функциональные и нефункциональные требования, определены четыре категории пользователей: сотрудник, сотрудник кадрового отдела, руководитель предприятия и администратор. Предложена клиент-серверная архитектура с React-клиентом, серверной частью на FastAPI и централизованной базой PostgreSQL. Разработана реляционная модель, связывающая пользователей, сотрудников, роли, кадровые формы, заполненные поля, приказы, файлы, запросы на регистрацию и журнал действий.

Ключевые слова: проектирование; веб-приложение; кадровый документооборот; FastAPI; React; PostgreSQL; UML; база данных; пользовательский интерфейс

Введение

После анализа предметной области возникает задача перевода выявленных проблем в формализованные требования и проектные решения. Для кадрового документооборота ООО «ВЛК-ОЙЛ» система должна поддерживать подготовку документов, их проверку и согласование, а также сохранять возможность экспорта

итогового документа для печати и подписания. Цель статьи – разработать согласованный проект веб-приложения, обеспечивающего автоматизацию основных этапов смешанного кадрового документооборота.

Материалы и методы

Исходными материалами являлись результаты анализа кадрового процесса предприятия, перечень форм документов, требования к их заполнению и перечень ролей пользователей. Для описания взаимодействий применялось UML-моделирование, для структуры хранения – реляционное моделирование и ER-представление. Архитектура рассматривалась как клиент-серверная, с разделением интерфейса, серверной логики и слоя данных.

Пользовательские роли

Для представления взаимодействия пользователей с разрабатываемой программой построена диаграмма вариантов использования. В системе выделяются четыре категории пользователей: сотрудник, сотрудник кадрового отдела, руководитель предприятия и администратор.

Сотрудник проходит авторизацию, просматривает личные данные, выбирает форму кадрового документа, заполняет её поля, при необходимости прикрепляет файл, отправляет форму в кадровый отдел и отслеживает статус документа.

Сотрудник кадрового отдела просматривает заполненные формы сотрудников, проверяет корректность внесённых данных, при необходимости возвращает форму на доработку, а при отсутствии ошибок формирует приказ и направляет его руководителю предприятия. Кроме того, сотрудник кадрового отдела может создавать запрос на регистрацию нового пользователя.

Руководитель предприятия просматривает готовые приказы, утверждает или отклоняет их, а также может экспортировать приказ для последующей печати.

Администратор обрабатывает запросы на регистрацию пользователей, создаёт учётные записи, назначает роли, управляет справочниками и просматривает

журнал действий. Для работы с журналом действий предусмотрены поиск и фильтрация записей.



Рисунок 1 – Диаграмма вариантов использования программы

Архитектура программного продукта

База данных выделяется как самостоятельный компонент хранения информации. В ней сохраняются сведения о пользователях, сотрудниках, ролях, кадровых формах, приказах, статусах документов и действиях пользователей. Прямое взаимодействие клиентской части с базой данных не предусматривается: все обращения к данным выполняются через серверную часть.

часть – операции обработки данных, а база данных – сведения, необходимые для работы программы.

Взаимодействие компонентов строится следующим образом: клиентская часть передаёт запросы серверной части, серверная часть обрабатывает их, при необходимости обращается к базе данных и возвращает результат выполнения операции. Такое разделение компонентов позволяет отделить интерфейс программы от логики обработки данных и структуры их хранения.

UML-диаграмма компонентов программы представлена на рисунке 6.



Рисунок 6 – UML-диаграмма компонентов программы

Проектирование базы данных

База данных должна хранить пользователей, сотрудников, роли, кадровые формы, заполненные поля, приказы, файлы, запросы на регистрацию и журнал действий. Основные таблицы соответствуют сущностям проектной модели.

Таблица 1 – основные таблицы

Таблица	Назначение
roles	Роли пользователей
employees	Сведения о сотрудниках
users	Учётные записи
document_types	Типы кадровых документов
document_requests	Кадровые документы
request_fields	Заполненные поля
orders	Приказы
attached_files	Прикреплённые файлы
registration_requests	Запросы на регистрацию

action logs	Журнал действий
-------------	-----------------

Ключевыми связями являются связь роли с пользователями, сотрудника с учётной записью, типа документа с заявками, сотрудника с документами, документа с заполненными полями, документа с приказом и документа с файлами. Один пользователь может иметь несколько записей в журнале действий.

Проектирование пользовательского интерфейса

Пользовательский интерфейс программы проектируется с учётом ролевой модели доступа. После авторизации пользователю отображаются только те разделы, которые соответствуют его роли и выполняемым функциям.

Основные разделы интерфейса представлены в таблице 5.

Таблица 5 – Основные разделы пользовательского интерфейса

Раздел интерфейса	Назначение
Страница авторизации	Вход пользователя в систему по логину и паролю
Личный кабинет сотрудника	Просмотр личных данных и работа с собственными кадровыми документами
Формы кадровых документов	Выбор формы, заполнение полей и отправка формы в кадровый отдел
Панель кадрового отдела	Проверка заполненных форм, возврат на доработку и формирование приказов
Панель руководителя	Просмотр, утверждение или отклонение готовых приказов
Панель администратора	Обработка запросов на регистрацию

Результаты и обсуждение

В результате проектирования сформирован набор взаимосвязанных решений: требования, ролевая модель, архитектура, сценарии, структура базы данных и интерфейс. Каждая пользовательская операция сопоставлена с ролью и серверной функцией. React отвечает за взаимодействие с пользователем, FastAPI – за обработку запросов и бизнес-правила, PostgreSQL – за хранение данных.

Заключение

Разработан проект веб-приложения для автоматизации смешанного кадрового документооборота ООО «ВЛК-ОЙЛ». Сформированы требования и определены четыре категории пользователей. Предложена клиент-серверная архитектура с React, FastAPI и PostgreSQL, спроектирована структура базы данных и интерфейса. Полученные решения создают основу для реализации и последующего функционального тестирования.

Список использованных источников

1. Лаврищева Е.М. Программная инженерия. Парадигмы, технологии и CASE-средства. – М.: Юрайт, 2022. – 280 с.
2. Петрова И.Р. и др. Методология объектно-ориентированного моделирования. Язык UML. – Казань: Казанский университет, 2018. – 79 с.
3. Куклина И.Г., Сафонов К.А. Методы и средства проектирования информационных систем. – Нижний Новгород: ННГАСУ, 2020. – 84 с.
4. Илющечкин В.М. Основы использования и проектирования баз данных. – М.: Юрайт, 2019. – 213 с.
5. FastAPI Documentation. URL: <https://fastapi.tiangolo.com/> (дата обращения: 01.04.2026).
6. React Documentation. URL: <https://react.dev/> (дата обращения: 06.04.2026).
7. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/> (дата обращения: 15.04.2026).