



**Институт интеллектуальных и кибернетических  
систем**

**Кафедра «Компьютерные системы и технологии»**

Пояснительная записка  
к проекту по курсу «Программирование  
сетевых приложений»  
на тему «Электронный документооборот для машиностроения»

Студент гр.	Б20-523 Овсянников А.Ю. / _____ /
Руководитель	Хисамутдинов Марат / _____ /
	Григорьев Михаил / _____ /

Москва, 2023г.

## АННОТАЦИЯ

Пояснительная записка 30 с., 26 рис., 8 табл., 12 ист.

Объектом разработки является приложение для электронного облачного документооборота для предприятий промышленности.

Целью работы является разработка серверной и клиентской части приложения, с помощью которого возможно хранение документов на сервере с разграничением прав среди авторизованных пользователей, а также их изменение.

Было произведено ручное тестирование разработанного приложения.

Серверная часть приложения реализована на языке программирования C# с использованием фреймворка ASP.NET Core MVC. В качестве клиентской части выступает веб-браузер. Для работы с базой данных была выбрана СУБД PostgreSQL.

# Содержание

Содержание.....	3
Введение.....	4
1 Теоретическая часть.....	5
1.1 Краткий анализ существующих решений.....	5
1.2 Постановка задачи.....	5
1.3 Схема приложения.....	7
1.4 Формализация данных.....	7
2 Практическая часть.....	8
2.1 Используемые технологии.....	8
2.2 Описание таблиц базы данных.....	8
2.3 Уровень доступа к данным.....	9
2.4 Уровень бизнес-логики.....	10
2.5 Уровень взаимодействия.....	11
2.5.1 Контроллер домашней страницы.....	12
2.5.2 Контроллер авторизации.....	12
2.5.3 Контроллер регистрации.....	13
2.5.4 Контроллер документов.....	13
2.5.5 Контроллер отделов.....	14
2.5.6 Контроллер профиля.....	14
2.5.7 Контроллер запросов.....	15
2.6 Аутентификация пользователей.....	16
2.7 Веб-приложение.....	17
2.7.1 Интерфейс веб-приложения.....	17
2.7.2 Взаимодействие с сервером.....	25
3 Тестирование.....	26
Заключение.....	27
Список используемых источников.....	29

## **Введение**

«Электронный облачный документооборот для машиностроения» - программа, упрощающая работу промышленных предприятий при работе с большим количеством бумажных документов и чертежей.

Большой проблемой на многих предприятиях машиностроения является чрезмерная бюрократия [1], из-за которой требуется хранить большое количество документов и чертежей, что может замедлить работу всего предприятия, из-за того, что всякий раз, когда необходимо получить нужный документ, либо чертёж нужной детали, нужно получить её физически. Немаловажным фактором может являться потеря бумаг ввиду человеческого фактора.

Решением данных проблем может являться программа «Электронный облачный документооборот для машиностроения», которая имеет возможность создавать и редактировать документы, являющейся обёрткой над документом формата .pdf. Вся информация будет храниться на сервере. Пользователи будут иметь разные права в зависимости от их роли и состоять в разных группах (отделах). Роли определяют возможности пользователя в системе.

В процессе работы была поставлена задача выполнения проекта, который заключается в создании веб приложения в качестве клиентской части и API для этого приложения в качестве серверной части.

Данное приложение предназначено в первую очередь для промышленных компаний, но также может использоваться и для любых компаний, которые хотят пройти цифровизацию. Главным преимуществом является использование в качестве клиента веб браузера, поэтому технические требования, предъявляемые к пользователям, минимальны.

# **1 Теоретическая часть**

## **1.1 Краткий анализ существующих решений**

В настоящее время существуют сервисы для электронного документооборота, но большинство из них также требуют установку клиентской части отдельно, например, как Союз-PLM [2], используемый на предприятии ПАО НПО «Алмаз» [3].

Текущее приложение в последствии может быть использовано как основа для дальнейшей разработки электронного документооборота с использованием веб браузера как клиента.

Из преимуществ использования веб-браузера как клиента можно выделить:

- Легкодоступность – веб браузер возможно установить на любой современный компьютер;
- Использование современных фреймворков, так как веб разработка одна из самых быстроразвивающихся направлений в it.

## **1.2 Постановка задачи**

Были выделены следующие требования к функционалу приложения:

- 1) Хранение на сервере технических документов – обёртки над основным файлом формата .pdf. Технический документ дополнительно к файлу хранит: описание, название, историю изменений с датами, кто разрабатывает документ, отдел разработки и статус документа: в разработке, на подтверждении, готово. Также хранятся все предыдущие версии этого документа с датами обновлений.
- 2) В программе реализована система ролей. Всего существует 3 роли, каждая из которых может совершать определенные действия над другими пользователями, группами и техническими документами:

1. Администратор:

- Группы: создать группу, прикрепить пользователя к группе после регистрации и отменить запрос на присоединение пользователей,

независимо от их группы. При этом может выбрать роль новому пользователю – либо начальник отдела, либо работник. Группа представляет из себя набор пользователей, которые объединены для совместной разработки конкретного продукта.

- Технические документы: возможность просматривать любой технический документ, независимо от отдела. Возможность подтверждать готовность ввода в эксплуатацию любого документа. Возможность добавлять технические документы и их новые версии.

## 2. Начальник отдела:

- Группы: прикрепление нового пользователя в отдел, и отмена запроса. Может выдать только роль работника.
- Технический документ: возможность просматривать документы своего отдела. Возможность изменять их статус. Возможность добавлять технические документы и их новые версии.

## 3. Работник:

- Группы: нет возможности изменять состав группы.
- Технические документы: возможность изменять и добавлять новые файлы в технический документ. Возможность добавлять технические документы и их новые версии.

3) Присутствует система регистрации пользователей, в которой они указывают следующие данные: логин, пароль, ФИО, и выбирают из списка отдел, к которому они хотят присоединиться. При регистрации, администратор, начальник или заместитель начальника выбранного отдела должны подтвердить добавление пользователя.

4) Присутствует система авторизации и аутентификации. Для входа в систему пользователь должен предоставить логин и пароль.

5) Присутствует функция поиска по документам.

### 1.3 Схема приложения

Клиент-серверное приложение состоит из нескольких связанных частей. Схема взаимодействия между частями приложения представлена на рисунке 1.



Рисунок 1. Схема взаимодействия между частями приложения

Пользователь через веб браузер может совершать действия, которые будут обрабатываться сервером и сохраняться в базе данных.

### 1.4 Формализация данных

База данных должна хранить данные о следующих сущностях, представленных в таблице 1.

Таблица 1. Сущности базы данных

Сущности	Сведения
Пользователь	Id, логин, фамилия, имя, почта, хэш пароля, роль, id отдела, подтвержденный ли аккаунт, дата создания
Отдел	Id, название отдела
Технический документ	Id, название технического документа, id автора, дата создания, описание, статус, номер последней версии.
Файл документа	Id, путь в файловой системе сервера, версия, id технического документа, к которому прикреплен файл, дата добавления.
Запросы на добавление	Id, id пользователя, который хочет добавиться в систему, id отдела, к которому он желает присоединиться

## 2 Практическая часть

### 2.1 Используемые технологии

В качестве языка программирования для разработки серверной части приложения был выбран язык программирования C#, поскольку он поддерживает парадигму объектно-ориентированного программирования, а также поддерживает возможность работы с фреймворками, такими как ASP .NET Core MVC [4]. Данный фреймворк упрощает разработку, используя такой паттерн как MVC – Model-View-Controller [5]. Также язык C# упрощает разработку под Windows-окружение.

В качестве базы данных используется PostgreSQL [6], как самая распространённая, надёжная и быстрая СУБД с открытым исходным кодом.

Для разработки пользовательского интерфейса было принято решение ограничиться html-страницами на основе Razor Pages [7]. Данный страницы позволяют использовать html-страницы с использованием C# кода.

### 2.2 Описание таблиц базы данных

Чтобы организовать хранение информации о пользователях, отделах, технических документов, файлов документов и запросов используется база данных PostgreSQL. Структура базы данных представлена ниже.

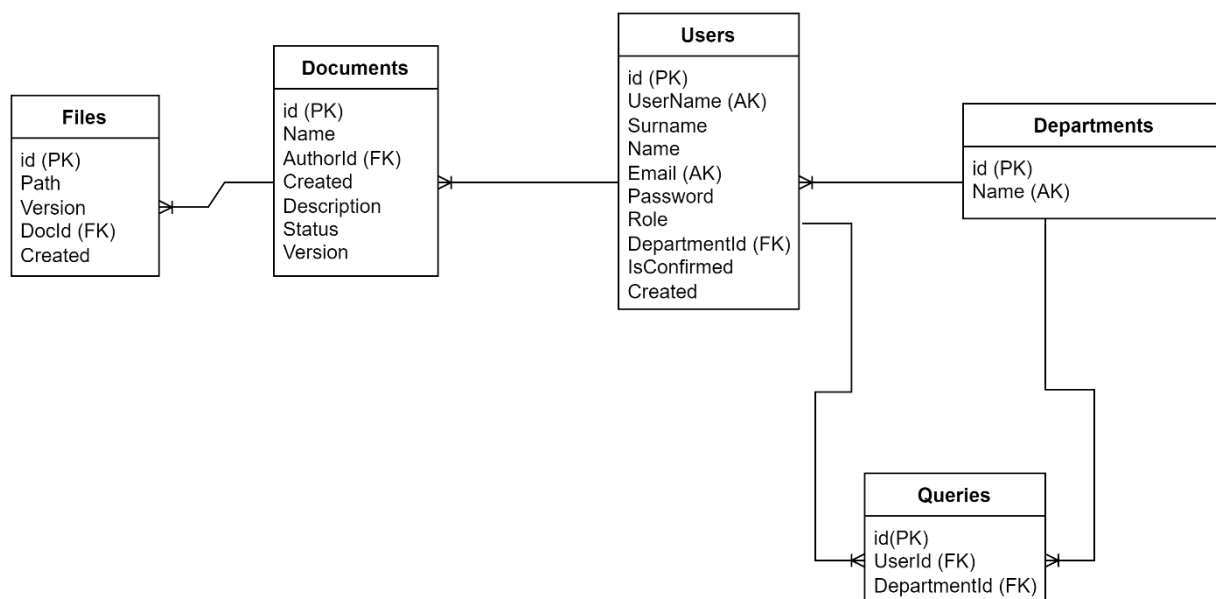


Рисунок 2. Структура базы данных



Сущность Users – в ней хранятся id, логин, фамилия, имя, почта, хешированный пароль, роль, id отдела, подтверждённый ли аккаунт, дата создания.

Сущность Departments – в ней хранятся id и название отдела.

Сущность Documents – в ней хранится id, название документа, id автора, дата создания, описание, статус и версия.

Сущность Files – в ней хранится id, путь до файла на сервере, версия, id документа и дата создания.

Сущность Queries – в ней хранится id запроса, id аккаунта, который хочет присоединиться к системе, и id отдела, к которому хочет присоединиться пользователь.

### **2.3 Уровень доступа к данным**

Data Access Object (Объект доступа к данным) – структурный паттерн, который позволяет изолировать бизнес-логику от уровня базы данных [8]. Это помогает изменять функциональность сервиса, не затрагивая схемы в БД – лишь изменять уровень доступа к данным.

Чтобы создавать объекты доступа к данным было решено использоватьстроенную библиотеку объектно-реляционного отображения для C# EntityFramework [9], которая позволяет конвертировать C# классы, в сущности, для реляционной базы данных и наоборот. [10]

Для обращения к базе данных объектов из бизнес-логики, были созданы следующие интерфейсы:

- 1) IUserRepository – интерфейс, для обращения к базе данных пользователей.
- 2) IDepartmentRepository – интерфейс, для обращения к базе данных отделов.
- 3) IQueryRepository – интерфейс, для обращения к базе данных запросов.
- 4) IDocumentRepository – интерфейс, для обращения к базе данных документов.

5) IRepository – интерфейс, для обращения к базе данных файлов документов.

Каждый из этих интерфейсов задает для классов реализацию через методы добавления, изменения, чтения и удаления из базы.

При запуске сервера, происходит связывание вышеперечисленных интерфейсов с реализацией конкретного класса. Код представлен на рисунке 3.

```
builder.Services.AddTransient<IUserRepository, UserRepository>();  
builder.Services.AddTransient<IDepartmentRepository, DepartmentRepository>();  
builder.Services.AddTransient<IClaimService, ClaimService>();  
builder.Services.AddTransient<IAuthService, AuthService>();  
builder.Services.AddTransient<IQueryRepository, QueryRepository>();  
builder.Services.AddTransient<IPasswordHasher, PasswordHasher>();  
builder.Services.AddTransient<IQueryService, QueryService>();  
builder.Services.AddTransient<IDocumentRepository, DocumentRepository>();  
builder.Services.AddTransient<IFileRepository, FileRepository>();  
builder.Services.AddTransient<IFileService, FileService>();
```

Рисунок 3. Связывание интерфейса с конкретным классом.

Теперь при вызове метода интерфейса, будет вызываться метод конкретного класса.

## 2.4 Уровень бизнес-логики

После того, как появилась возможность взаимодействовать с базой данных посредством C# классов, можно приступить к уровню бизнес-логики. На данном этапе будут проводиться все вычисления, которые нужны для функционирования самого сервиса. От создания пользователя и авторизации до добавления пользователя в систему и создания документов.

Всего существуют следующие сервисы:

- сервис авторизации нужен для авторизации, регистрации и выхода пользователя из системы. Реализуется интерфейсом IAuthService. При авторизации проверяет корректность вводимых данных, а также выдает токен на основе claim-based authentication [11] в cookie браузера. При регистрации проверяет корректность данных и добавляет запрос в базу данных запросов. При выходе из системы удаляет токен из cookie.

- сервис выдачи токенов. Реализуется интерфейсом `IClaimService`. Нужен для создания токена на основе `id`, логина и роли пользователя.
- сервис хэширования пароля. Реализуется интерфейсом `IPasswordHasher`. Нужен для создания хэша из пароля, а также проверки соответствия пароля к хэшу.
- сервис обработки запросов на регистрацию. Реализуется интерфейсом `IQueryService`. Нужен для подтверждения запроса, его отмены и получения всех запросов в системе.
- сервис работы с файловым сервером. Реализуется классом `IFileService`. Предназначен для сохранения файлов в системе.

Эти сервисы были выделены в отдельные классы, так как выполняют обособленную часть бизнес-логики. Большая часть бизнес-логики реализована внутри контроллеров.

## **2.5 Уровень взаимодействия**

Следующий уровень в цепочке – уровень отображения данных в бизнес-логике пользователю или другим сервисам, которые хотят взаимодействовать с API. На этом уровне определяется способ взаимодействия сервиса с остальными пользователями сети. Было принято решение выбрать REST (Representation State Transfer) [12] – это набор правил и ограничений для проектирования сервиса, чтобы все системы легко обменивались данными и приложение можно было масштабировать.

На данном этапе важно определить, что именно будет отдаваться для внешнего мира. Было принято решение ограничиться html-страницами, которые обёрнуты в razor-pages. Razor-pages представляют собой html-страницы, с возможной ставкой C# кода. Также в razor-pages есть возможность передавать данные из контроллера, посредством передачи таких классов, как `ModelView`, которые представляют собой данные, необходимые для корректного отображения страницы пользователю.

Чтобы пользователи сети посылали запрос на сервис, нужны конечные точки, по запросу на которые сервис будет отдавать ответ. Для этой цели были

созданы контроллеры, которые управляют запросами, приходящими на сервер. Всего было разработано 7 контроллеров, которые отвечают за запросы для работы с документами, отделами, регистрацией, авторизацией, профилем пользователя, запросами. Также отдельно был создан контроллер домашней страницы.

### 2.5.1 Контроллер домашней страницы

Общий URL для всех запросов будет содержать путь /.

Информацию о контроллере домашней страницы можно увидеть в таблице ниже.

Таблица 2. Конечные точки контроллера домашней страницы

URL запроса	Метод HTTP-запроса	Доступ	Функция конечной точки
/	GET	Авторизованные пользователи	Отображение домашней страницы сайта.
/accessdenied	GET	Авторизованные пользователи	Отображение страницы, когда у пользователя нет прав на просмотр определённой страницы.

### 2.5.2 Контроллер авторизации

Общий URL для всех запросов будет содержать путь /{action}. Где action – название GET метода контроллера. Всего реализуется 2 метода – Login и Logout.

Информацию о контроллере авторизации можно увидеть в таблице ниже.

Таблица 3. Конечные точки контроллера авторизации

URL запроса	Метод HTTP-запроса	Доступ	Функция конечной точки
/login	GET	Любые пользователи	Отображение формы авторизации.
/login	POST	Любые пользователи	Обработка запроса на авторизацию.
/logout	GET	Авторизованные пользователи	Удаление токена из cookie браузера.

### 2.5.3 Контроллер регистрации

Общий URL для всех запросов будет содержать путь /register.

Информацию о контроллере регистрации можно увидеть в таблице ниже.

Таблица 4. Конечные точки контроллера регистрации

URL запроса	Метод HTTP-запроса	Доступ	Функция конечной точки
/register	GET	Любые пользователи	Отображение формы регистрации.
/register	POST	Любые пользователи	Обработка запроса на регистрацию.

### 2.5.4 Контроллер документов

Общий URL для всех запросов будет содержать путь /docs.

Информацию о контроллере документов можно увидеть в таблице ниже.

Таблица 5. Конечные точки контроллера документов

URL запроса	Метод HTTP-запроса	Доступ	Функция конечной точки
/docs	GET	Авторизованные пользователи	Отображение всех документов отдела, в котором находится пользователь. Если роль пользователя администратор, то отображаются все документ в системе.
/docs	POST	Авторизованные пользователи	Поиск по документам.
/docs/create	GET	Авторизованные пользователи	Отображается форма на создание документа.
/docs/create	POST	Авторизованные пользователи	Обработка запроса на создание документа.
/docs/{id}	GET	Пользователи отдела, к	Отображение документа с id, указанного в URL.

		которому принадлежит документ, администраторы	
/docs/{id}/{fileId}	GET	Пользователи отдела, к которому принадлежит документ, администраторы	Отображение файла fileId в документе id
/docs/{id}/add	GET	Создатель документа	Отображение формы для добавления новой версии документа
/docs/{id}/add	POST	Создатель документа	Добавление новой версии документа в систему.

### 2.5.5 Контроллер отделов

Общий URL для всех запросов будет содержать путь /groups.

Информацию о контроллере отделов можно увидеть в таблице ниже.

Таблица 6. Конечные точки контроллера отделов

URL запроса	Метод HTTP-запроса	Доступ	Функция конечной точки
/groups	GET	Авторизованные пользователи	Отображение всех отделов в системе.
/groups/id	GET	Авторизованные пользователи	Отображается сотрудников отдела id.
/groups/create	GET	Администраторы	Обработка формы создания нового отдела.
/groups/create	POST	Администраторы	Обработка запроса на создание нового отдела.

### 2.5.6 Контроллер профиля

Общий URL для всех запросов будет содержать путь /profile.

Информацию о контроллере профиля можно увидеть в таблице ниже.

Таблица 7. Конечные точки контроллера профиля

URL запроса	Метод HTTP-запроса	Доступ	Функция конечной точки
/profile	GET	Авторизованные пользователи	Отображение всех пользователей в системе.
/profile/id	GET	Авторизованные пользователи	Отображается пользователя с id.
/profile/id/password	GET	Пользователь id	Обработка формы изменения пароля пользователя id.
/profile/id/password	POST	Пользователь id	Обработка запроса на изменение пароля пользователя id.
/profile/id/email	GET	Пользователь id	Отображение формы изменения почты пользователя id.
/profile/id/email	POST	Пользователь id	Обработка запроса на изменение почты пользователя id.

### 2.5.7 Контроллер запросов

Общий URL для всех запросов будет содержать путь /query.

Информацию о контроллере запросов можно увидеть в таблице ниже.

Таблица 8. Конечные точки контроллера профиля

URL запроса	Метод HTTP-запроса	Доступ	Функция конечной точки
/query	GET	Администраторы и начальники отделов	Отображение всех запросов в системе.
/query	POST	Администраторы и начальники отделов	Обновление запросов в системе.

## 2.6 Аутентификация пользователей

В сервисе предусмотрена аутентификация пользователей и последующее взаимодействие используя Claim-Based authentication. Это встроенный механизм аутентификации в фреймворке ASP .NET Core MVC. Представляет из себя набор Identities, которые в свою очередь состоят из нескольких Claims. В Claims хранится информация о пользователе, полезная для работы в системе, например, как id и роль.

Когда пользователь проходит процесс авторизации в контроллере LoginController, ему выдается Identity со следующей информацией: id пользователя, роль и логин. Сгенерированный токен хранится в cookie веб-браузера. Реализация создания ClaimsIdentity в классе, реализующего интерфейс IClaimService представлен на рисунке 4.

```
4 namespace Electronic_document_management.Services.Claims
5 {
6     1 reference
7     public class ClaimService : IClaimService
8     {
9         2 references
10        public ClaimsPrincipal BuildClaims(User user)
11        {
12            var claims = new List<Claim> {
13                new Claim(ClaimTypes.NameIdentifier, user.Id.ToString()),
14                new Claim(ClaimTypes.Name, user.UserName),
15                new Claim(ClaimTypes.Role, user.Role.ToString())
16            };
17            var claimIdentity = new ClaimsIdentity(claims, "Cookies");
18            return new ClaimsPrincipal(claimIdentity);
19        }
20    }
21 }
```

Рисунок 4. Создание Claims Identity.

Если пользователь желает выйти из системы, то он переходит по URL /Logout, тем самым токен удаляется. Сервис доступен только авторизованным пользователям.

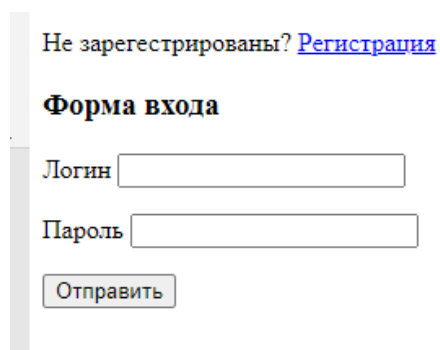


## 2.7 Веб-приложение

### 2.7.1 Интерфейс веб-приложения

Веб-приложение состоит из множества страниц, с помощью которых пользователь может выполнять нужные ему действия. Для работы пользователя в системе, пользователю необходимо пройти процедуру авторизации или регистрации. При наличии валидированного токена в cookie, система выдаёт доступ на пользование сервисом.

Форма авторизации представлена на рисунке 5, а регистрации – на рисунке 6.



Не зарегистрированы? [Регистрация](#)

**Форма входа**

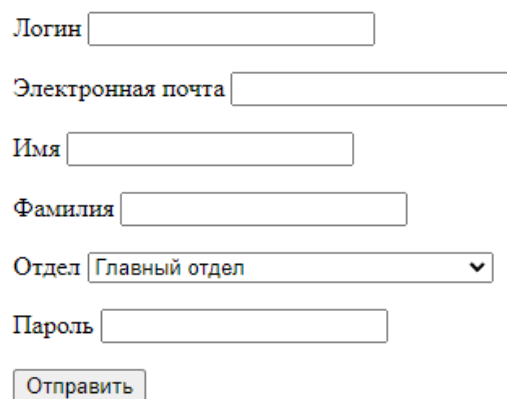
Логин

Пароль

Рисунок 5. Форма авторизации

Уже есть аккаунт? [Войти](#)

#### Форма регистрации



Логин

Электронная почта

Имя

Фамилия

Отдел

Пароль

Рисунок 6. Форма регистрации

В форме авторизации пользователю необходимо ввести свой логин и пароль. Если пользователь не зарегистрирован в системе, ему необходимо пройти процедуру регистрации, в которой следует ввести: логин, электронную почту, имя, фамилию и пароль. Также пользователю следует выбрать из списка отдел, к которому он желает присоединиться.

Если пользователь неверно ввёл логин или пароль от аккаунта, то система попросит ввести данные заново. Пример представлен на рисунке 7.

Если пользователь во время регистрации введёт уже существующий логин или почту, то система попросит выбрать другие данные. Пример представлен на рисунке 8. При успешной регистрации, данные пользователя

сохранятся в базе данных, но подтверждение должен дать либо администратор, либо начальник отдела, к которому хочет присоединиться пользователь.

### Неверный логин или пароль

Не зарегистрированы? [Регистрация](#)

### Форма входа

Логин

Пароль

Рисунок 7. Неверный логин или пароль

Логин уже используется  
Уже есть аккаунт? [Войти](#)

### Форма регистрации

Логин

Электронная почта

Имя

Фамилия

Отдел

Пароль

Рисунок 8. Неверные данные при регистрации

После успешной авторизации пользователь попадает на домашнюю страницу. В зависимости от роли пользователя, у него отображаются различные действия. Домашняя страница пользователя с ролью работник, представлена на рисунке 9, главы отдела – на рисунке 10, а администратора – на рисунке 11.

[Главная страница](#) | [Документы](#) | [Пользователи](#) | [Отделы](#) | [Запросы](#) | [2](#) | [Выйти](#)

### Главная страница

Рисунок 9. Главная страница работника.

[Главная страница](#) | [Документы](#) | [Пользователи](#) | [Отделы](#) | [1](#) | [Выйти](#)

### Главная страница

Рисунок 10. Главная страница начальника отдела.

## Главная страница

Рисунок 11. Главная страница администратора.

Пользователям доступен следующий функционал в веб-приложении:

- 1) Просмотр документов отдела. Пример представлен на рисунке 12. Пользователь может выбрать интересующий его документ и изучить его. Присутствует информацию об авторе, об отделе, дата создания и его статус: в разработке, на рассмотрении или готов. Также присутствует поиск по документам, который основывается на названии и описании документа.

[Главная страница](#) | [Документы](#) | [Пользователи](#) | [Отделы](#) | [1](#) | [Выйти](#)  
[Добавить документ](#)

Текст для поиска

Поиск

Документы					
№	Название	Автор	Отдел	Дата создания	Статус
1	<a href="#">Тестовый документ</a>	<a href="#">Heater</a>	<a href="#">Отдел информационных технологий</a>	20.11.2023 21:59:11	Ready
2	<a href="#">Форматы команд</a>	<a href="#">Heater</a>	<a href="#">Отдел информационных технологий</a>	21.11.2023 16:59:39	Ready
3	<a href="#">1</a>	<a href="#">1</a>	<a href="#">Отдел информационных технологий</a>	24.12.2023 19:17:32	InDeveloping
4	<a href="#">Документ пользователя 3 3</a>	<a href="#">3 3</a>	<a href="#">Отдел информационных технологий</a>	21.11.2023 7:45:49	InDeveloping

Рисунок 12. Страница документов

- 2) Добавление нового документа. Форма добавления документа представлена на рисунке 13. После отправки будет автоматически загружаться страница с документами.

### Выберите файл для загрузки

Название

Описание

Файл документа  Файл не выбран

Рисунок 13. Страница добавления документа

3) Просмотр документа. Страница для автора с ролью работник представлена на рисунке 14, а администратора и начальника отдела – на рисунке 15. Администратор может поменять статус любого документа, а начальник отдела – только документы своего отдела. При этом добавлять новую версию документа может только автор этого документа. Присутствует информация об авторе, отделе, дате создания документа, статусе, последней версии и описание. Присутствует история изменения файлов документа с датами изменения. При выборе определенной версии открывается .pdf файл.

### Документ ПВО

Автор: [1](#)

Отдел: [Отдел информационных технологий](#)

Дата создания документа: 24.12.2023 19:53:20

Статус: InDeveloping

Последняя версия: 2

Описание: Документ, содержащий информации о разработке С-400

История файлов документа

№	Версия документа	Дата добавления
---	------------------	-----------------

1	<a href="#">1</a>	24.12.2023 19:53:20
---	-------------------	---------------------

2	<a href="#">2</a>	24.12.2023 19:53:27
---	-------------------	---------------------

Рисунок 14. Страница просмотра документа автором

Выберите статус для документа В разработке

Отправить

В разработке

На рассмотрении

Готов

### Документ пользователя 3

Автор: [3](#)  
 Отдел: [Отдел информационных технологий](#)  
 Дата создания документа: 21.11.2023 7:45:49  
 Статус: InDeveloping  
 Последняя версия: 2  
 Описание: Описание к 3

#### История файлов документа

№	Версия документа	Дата добавления
1	<a href="#">1</a>	21.11.2023 7:45:49
2	<a href="#">2</a>	21.11.2023 7:46:03

Рисунок 15. Страница просмотра документа администратором.

4) Просмотр файлов определенного документа. Пример представлен на рисунке 16.

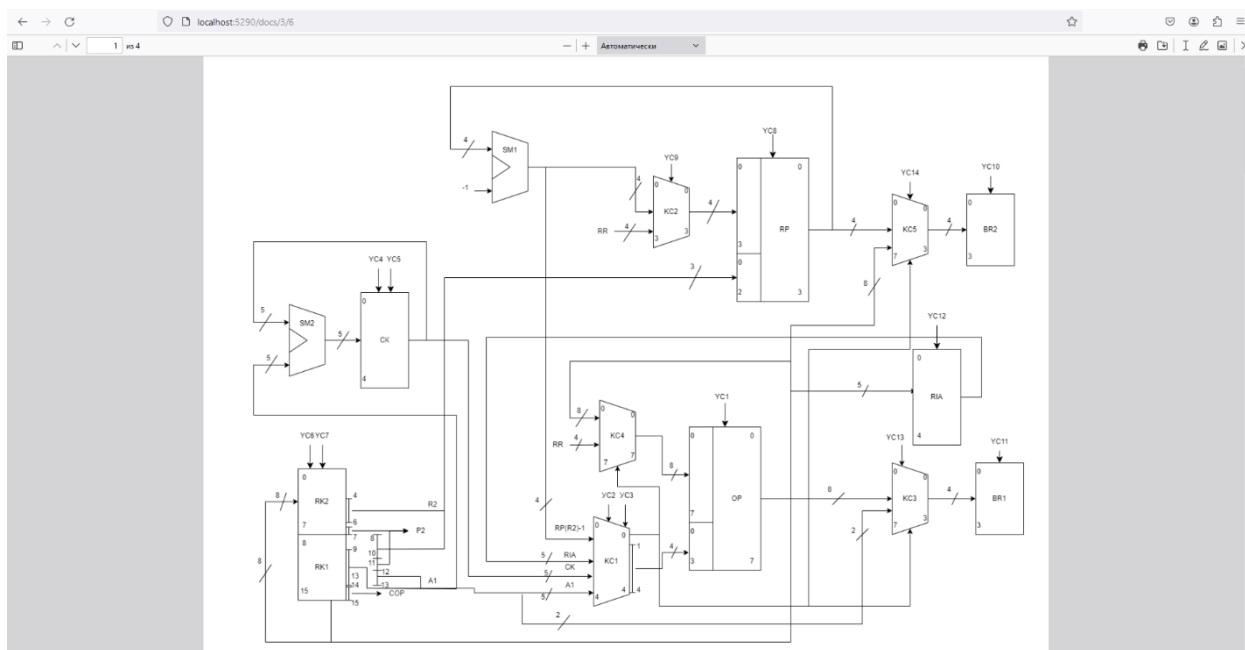


Рисунок 16. Страница просмотра файла документа.

5) Добавление новой версии документа автором документа. Форма представлена на рисунке 17. После добавления версия документа увеличиться на 1. Файлу будет присвоена новая версия документа.

## Выберите файл для новой версии документа

Файл документа  Файл не выбран.

Рисунок 17. Страница формы добавления новой версии документа.

6) Просмотр отделов. У администратора есть возможность создать новый отдел. Страница с аккаунта работника представлена на рисунке 18, а с аккаунта администратора – на рисунке 19.

<a href="#">Главная страница</a>   <a href="#">Документы</a>   <a href="#">Пользователи</a>   <a href="#">Отделы</a>   <a href="#">1</a>   <a href="#">Выйти</a>	
Отделы	
№	Название
1	<a href="#">Главный отдел</a>
2	<a href="#">Отдел информационных технологий</a>
3	<a href="#">Департамент развития</a>
4	<a href="#">12321</a>

Рисунок 18. Страница списка отделов с аккаунта работника.

<a href="#">Главная страница</a>   <a href="#">Документы</a>   <a href="#">Пользователи</a>   <a href="#">Отделы</a>   <a href="#">Запросы</a>   <a href="#">Admin</a>   <a href="#">Выйти</a>	
<a href="#">Создать</a>	
Отделы	
№	Название
1	<a href="#">Главный отдел</a>
2	<a href="#">Отдел информационных технологий</a>
3	<a href="#">Департамент развития</a>
4	<a href="#">12321</a>

Рисунок 19. Страница списка отделов с аккаунта администратора.

7) Создание нового отдела. Новый отдел может создать только администратор. Форма создания представлена на рисунке 20.

### Форма создания отдела

Название

Рисунок 20. Форма создания нового отдела.

8) Просмотр профиля. При этом, через профиль пользователь может поменять почту и пароль. Пример просмотра профиля владельца аккаунта представлен на рисунке 21, а другого пользователя – на рисунке 22. Присутствует информация о логине, имени, фамилии, почте, отделе и роли пользователя.

<a href="#">Главная страница</a>   <a href="#">Документы</a>   <a href="#">Пользователи</a>   <a href="#">Отделы</a>   <a href="#">1</a>   <a href="#">Выйти</a>	
Профиль	
Логин	1
Имя	1
Фамилия	1
Почта	1@1.com
Отдел	Отдел информационных технологий
Роль	Worker
<a href="#">Изменить пароль</a> <a href="#">Изменить почту</a>	

Рисунок 21. Страница профиля владельца аккаунта.

<a href="#">Главная страница</a>   <a href="#">Документы</a>   <a href="#">Пользователи</a>   <a href="#">Отделы</a>   <a href="#">1</a>   <a href="#">Выйти</a>	
Профиль	
Логин	3
Имя	3
Фамилия	3
Почта	3@3.ru
Отдел	Отдел информационных технологий
Роль	HeadOfDepartment

Рисунок 22. Страница профиля пользователя.

9) Изменение пароля и почты. Такая возможность присутствует только у владельца аккаунта. При смене пароля следует указать старый пароль и дважды ввести новый. Если неверно указан старый пароль или новые пароли не совпадают, то система попросит ввести данные заново. При смене почты также проверяется уникальность новой почты. Если указана уже существующая почта, то система попросит ввести другую почту. Пример формы смены пароля представлен на рисунке 23, а почты – на рисунке 24.

[Главная страница](#) | [Документы](#) | [Пользователи](#) | [Отделы](#) | [1](#) | [Выйти](#)

### Изменение пароля

Пользователь 1

Старый пароль

Новый пароль

Новый пароль (повторно)

[Главная страница](#) | [Документы](#) | [Пользователи](#) | [Отделы](#) | [1](#) | [Выйти](#)

### Изменение почты

Пользователь 1

Текущая почта: 1@1.com

Новая почта

Рисунок 23. Изменение пароля.

Рисунок 24. Изменение почты.

10) Просмотр всех пользователей в системе. Пример страницы представлен на рисунке 25.

[Главная страница](#) | [Документы](#) | [Пользователи](#) | [Отделы](#) | [1](#) | [Выйти](#)

### Список пользователей

№	Логин	Фамилия	Имя	Почта	Отдел	Роль
1	<a href="#">Admin</a>	Admin	Admin	bleckbird9@yandex.ru	Главный отдел	Admin
2	<a href="#">Heater</a>	Овсянников	Александр	123@123.com	Отдел информационных технологий	HeadOfDepartment
3	<a href="#">1</a>	1	1	1@1.com	Отдел информационных технологий	Worker
4	<a href="#">2</a>	2	21321	2@2.com	Департамент развития	HeadOfDepartment
5	<a href="#">3</a>	3	3	3@3.ru	Отдел информационных технологий	HeadOfDepartment
6	<a href="#">4</a>	4	4	4@4.com	Отдел информационных технологий	Worker
7	<a href="#">12</a>	12	12	12@12.com	Главный отдел	Worker

Рисунок 25. Страница просмотра списка пользователей.

На странице представлен логин пользователя, по нажатию на который можно будет перейти в его профиль, фамилия, имя, почта, отдел и роль.



11) У администраторов и начальников отделов есть возможность добавлять новых пользователей в систему или отменить их запрос. Пример станицы представлен на рисунке 26. При этом администратор может выставить роль новому пользователю как работник, так и начальник отдела. Сам начальник отдела может выдать только роль работник.

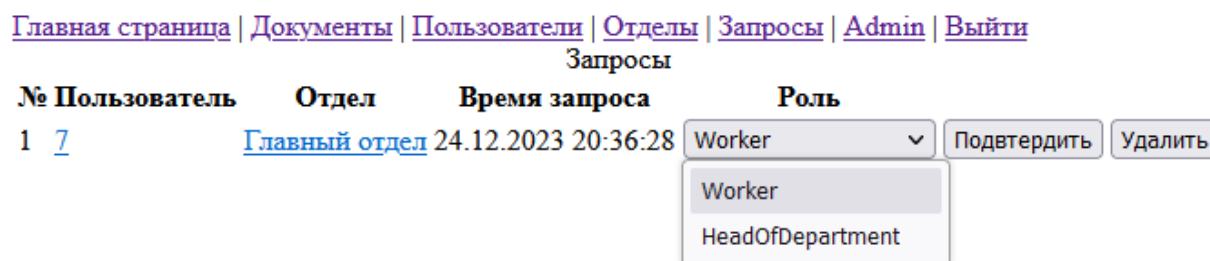


Рисунок 26. Страница просмотра списка запросов.

На данной странице предоставлена следующая информация: логин пользователя с возможностью просмотреть его данные, отдел, к которому он хочет присоединиться, время создания запроса, роль и 2 кнопки – подтвердить и удалить.

### 2.7.2 Взаимодействие с сервером

На данный момент сервер запускается на локальном компьютере в локальной сети. Существует возможность запуска на стороннем сервере.

### **3 Тестирование**

При разработке серверной и клиентской части приложения было проведено только ручное тестирование. Были протестированы все возможности системы, включая некорректный ввод данных, например, таких как не валидный формат почты и попытка получить доступ к ресурсу, прав к которому у пользователя нет.

Тестирование проводилось одновременно как для клиентской части, так и для серверной.

## Заключение

В соответствии с техническим заданием было разработано приложение, которое состоит из серверной части, клиентской части и базы данных.

В приложении представлены такие функции как:

- регистрация и вход в аккаунт пользователя;
- подтверждения добавления в систему вновь зарегистрированных пользователей администраторами и начальниками отделов;
- добавление технических документов и их новых версий;
- просмотр технических документов, а также поиск среди этих документов;
- просмотр пользователей в системе;
- просмотр отделов, создание новых администраторами;
- изменение учётных данных пользователями;

Проведенный обзор существующих решений показал, что на данный момент существует необходимость в создании легковесных сервисов для документооборота, где в качестве клиента будет выступать веб-браузер, которые установлен на любом компьютере в современном мире. Также сервер не нагружен большим количеством функций, поэтому в качестве сервера может выступать даже ноутбук.

Приложение не рассчитано на большую нагрузку, поэтому один из вариантов использования приложения – это использование внутри локальной сети компании.

Была спроектирована база данных, которая позволяет обращаться с информацией, которая нужна для функционирования приложения. Серверное приложение, которое позволяет производить действия по обработке данных и веб-клиент. С помощью него пользователь может получать информацию и вносить изменение.

Произведено ручное тестирования серверной и клиентской части.

Возможное дальнейшее развитие сервиса связано с созданием более удобного клиентского приложения, создания дополнительных функций в зависимости от потребностей конкретной компании, а также использования системы контейнеризации, для удобного запуска на серверах предприятий без дополнительной установки ПО.

## Список используемых источников

1. Панок Д. Г. Сравнительный анализ классических организационных структур управления промышленных предприятий //Проблемы предпринимательства в экономике России. Межвузовский сборник научных трудов. – 2005. – №. 8. (дата обращения: 21.12.2023)
2. Програмсоюз [Электронный ресурс]. 2023 URL: <https://www.programsoyuz.ru/> (дата обращения: 21.12.2023)
3. ПАО НПО «Алмаз» URL: <https://raspletin.com/> (дата обращения: 21.12.2023)
4. ASP .NET Core MVC [Электронный ресурс]. 2023 URL: <https://dotnet.microsoft.com/en-us/apps/aspnet> (дата обращения: 21.12.2023)
5. MVC Pattern [Электронный ресурс]. 2023 URL: <https://dotnet.microsoft.com/en-us/apps/aspnet/mvc> (дата обращения: 21.12.2023)
6. The PostgreSQL Global Development Group, PostgreSQL [Электронный ресурс]. 1996-2022 URL: <https://www.postgresql.org/> (дата обращения: 21.12.2023)
7. Razor Pages with ASP .NET [Электронный ресурс]. 2023 URL: <https://learn.microsoft.com/ru-ru/aspnet/core/tutorials/razor-pages/?view=aspnetcore-8.0> (дата обращения: 21.12.2023)
8. Baeldung. The DAO Pattern in Java [Электронный ресурс]. 2023 URL: <https://www.baeldung.com/java-dao-pattern> (дата обращения: 21.12.2023)
9. Entity Framework Core [Электронный ресурс]. 2023 URL: <https://learn.microsoft.com/ru-ru/ef/core/> (дата обращения: 21.12.2023)
10. Hibernate. What is Object/Relational Mapping? [Электронный ресурс]. 2023 URL: <https://hibernate.org/orm/what-is-an-orm/> (дата обращения: 21.12.2023)
11. Active Directory and claims-based authentication [Электронный ресурс]. 2023 URL: <https://learn.microsoft.com/en-us/dynamics365/customerengagement/on-premises/developer/active-directory-claims-based-authentication?view=op-9-1> (дата обращения: 21.12.2023)

12.Lokesh Gupta. What is REST [Электронный ресурс]. 2023 URL:  
<https://restfulapi.net/> (дата обращения: 21.12.2023)