Geekbrains

**Разработка веб-приложения по учету и контролю входящей и исходящей переписки, а также учету извещений, договоров и выпущенной текстовой и графической документации для Конструкторского комплекса №3 АО "Обуховский завод".**

Программа: Разработчик

Специализация: Разработчик – Fullstack разработчик.

Улога Д.В.

Санкт-Петербург

2025

Содержание

Введение 3

1 Цель работы 3

2 Актуальность 3

3 Объект и предмет исследования 4

4 Задачи работы 4

5 Методы и инструменты 5

6 Научная новизна и практическая значимость 6

Глава 1 Анализ предметной области 7

1.1 Общее описание предприятия и его деятельности 7

1.2 Описание текущего состояния документооборота: 7

1.3 Проблемы текущей системы 8

1.4 Обзор существующих решений 11

1.5 Выводы из анализа предметной области 12

Глава 2 Концептуальный анализ разработки веб-приложения для автоматизации документооборота 13

2.1 Цели и задачи разработки системы 13

2.2 Архитектура и выбор технологий 13

2.3 Описание функциональных требований 14

2.4 Проектирование базы данных 16

2.5 Процесс разработки и этапы реализации 18

2.6 Оценка эффективности 23

Глава 3 Реализация веб-приложения 28

3.1 Выбор технологий 28

3.2 Проектирование базы данных 31

3.3 Разработка клиентской части 41

3.4 Разработка серверной части 60

Глава 4 Тестирование системы 62

4.1 Методика тестирования 62

4.2 Функциональное тестирование 63

4.3 Нагрузочное тестирование 64

4.4 Выводы по тестированию 66

Заключение 67

Список используемой литературы 69

Приложения 70

Введение

1 Цель работы

Целью данного дипломного проекта является разработка веб-приложения по учету и контролю входящей и исходящей переписки, а также учету извещений, договоров и выпущенной текстовой и графической документации. Веб-приложение ориентировано на использование в конструкторском комплексе № 3 АО "Обуховский завод".

Разработка произведена по заказу Главного конструктора конструкторского комплекса № 3 АО "Обуховский завод".

2 Актуальность

Конструкторский комплекс № 3 (далее по тексту - КК-3) столкнулся с необходимостью обработки большого объема документации: входящих писем, исходящих ответов, договоров, извещений и прочих документов. В большинстве случаев процесс документооборота осуществляется вручную с использованием обычных бумажных журналов или с использованием простейших систем (например, Excel), что приводит к следующим проблемам:

- Потеря данных;

- Дублирование данных;

- Большое время на обработку документов;

- Высокий риск ошибок в записях, связанный с человеческим фактором;

- Отсутствие централизованного доступа к информации

- Нарушение последовательности ведения журнала, в случае ошибочного заполнения или при получении письма с опозданием (например, если письма изначально отправили в другое подразделение).

Для решения этих проблем требуется специализированное программное обеспечение, которое позволит:

- Автоматизировать процессы документооборота.

- Обеспечить безопасное хранение данных.

- Повысить прозрачность и эффективность работы сотрудников.

Актуальность данной работы определяется стремлением КК-3 обеспечить своевременное выполнение работ в рамках Государственного Оборонного заказа, улучшить свои бизнес-процессы и соответствовать современным стандартам управления документацией.

****3 Объект и предмет исследования****

Объект исследования – процесс документооборота в КК-3 АО "Обуховский завод". Предмет исследования – автоматизация учета и контроля документов, включая входящую/исходящую переписку, извещения, договоры и текстово-графические материалы.

4 Задачи работы

Для достижения цели необходимо решить следующие задачи:

1) Провести анализ текущего состояния документооборота в КК-3;

2) Разработать архитектуру веб-приложения, включающую фронтенд, бэкенд и базу данных;

3) Реализовать основные функциональные модули системы, включая:

- Добавление записей и загрузка файлов с отсканированным оригиналом документа;

- Редактирование записей, удаление/добавление файлов;

- Поиск документов по датам или ключевым фразам.

4) Провести тестирование приложения для проверки его функциональности и производительности.

5) Подготовить экономическое обоснование эффективности внедрения системы.

5 Методы и инструменты

В работе используются следующие методы и инструменты:

1) Методы проектирования:

- использование диаграмм (ER-диаграммы, диаграммы компонентов) для описания структуры системы;

- построение архитектуры на основе клиент-серверной модели.

2) Технологии разработки:

- фронтенд: HTML, CSS, JavaScript, React с использованием Material UI для создания удобного интерфейса;

- бэкенд: PHP для обработки запросов и взаимодействия с базой данных;

- база данных: MySQL для хранения информации и данных пользователей.

3) Методы тестирования:

- функциональное тестирование (проверка работы основных модулей).

- нагрузочное тестирование (проверка системы при одновременном доступе нескольких пользователей).

****6 Научная новизна и практическая значимость****

Научная новизна работы заключается в разработке специализированного решения для учета и контроля документации, учитывающего особенности документооборота предприятия. Практическая значимость заключается в создании системы, которая улучшит прозрачность и точность работы с документами, сократит временные затраты на обработку информации и снизит количество ошибок.

Анализ предметной области

 ****Общее описание предприятия и его деятельности****

Конструкторский комплекс № 3 АО "Обуховский завод" – это подразделение завода, занимающегося разработкой военной и гражданской техники, включая разработку чертежей, расчетов и эксплуатационной документации. КК-3 ведет активную переписку с предприятиями соисполнителями, а также с организациями заказчиками. Также ведется авторский надзор за изделиями, собирающимися в производственных подразделениях завода, что сопровождается выпуском извещений об изменении и перепиской с поставщиками, заказчиками, соисполнителями, а также перепиской внутри предприятия.

В настоящее время вся переписка КК-3 обрабатывается внутренней почтовой системой завода, что делает невозможным создание общедоступного архива переписки.

 Описание текущего состояния документооборота:

1) Обработка переписки.

На текущий момент вся переписка (входящая и исходящая переписка, а также внутренняя переписка) обрабатываются почтовой системой завода. Процесс начинается с получения физического или электронного письма (включая приложения). После получения письмо сканируется (в случае получения физического письма), затем регистрируется в общем архиве почтовой системы с указанием по какому изделию это письмо. Далее сотрудник канцелярии перенаправляет письмо начальнику подразделения, отвечающему за указанное изделие. Начальник подразделения перенаправляет письмо начальнику отдела, который в свою очередь пересылает его конкретному сотруднику для подготовки ответа.

После подготовки ответа, сотрудник согласовывает письмо с непосредственным начальником и отправляет его через канцелярию. Канцелярия делает сканированную копию письма и отправляет его во внешнюю организацию и сотруднику непосредственно его подготовившего.

2) Учет извещений об изменении.

Предприятие не ведет электронного учета извещений об изменении. Сотрудник, выпустивший извещение, регистрирует его в бумажном журнале отдела, затем сдает в отдел документации и учета. Отдел учета также регистрирует его в бумажном журнале, и проводит его в документации, в соответствии с указанным в извещении.

3) Учет выпускаемой документации.

Отделы подразделения не ведут учета выпускаемой документации. Сотрудники, выпуская документ сдают его в отдел документации и учета, где его и учитывают в бумажном журнале.

4) Учет договоров.

Учет ведется сотрудниками экономического отдела в стандартных офисных приложениях (Microsoft Excel и Word).

 Проблемы текущей системы

1) Проблемы учета переписки

Сотрудники подразделений имеют ограниченный доступ к архиву входящей и исходящей переписки. Например, сотрудник видит только те письма, которые ему были адресованы начальником, а исходящие письма доступны только тому, кто их отправил. Даже непосредственный начальник сотрудника не имеет доступа к архиву отправленных писем. Полный доступ к архиву имеется только у сотрудников канцелярии.

Архив переписки неструктурирован: письма не классифицируются по темам, подразделениям или изделиям, из-за чего поиск информации занимает много времени.

Сотрудники сталкиваются с трудностями при получении информации о замечаниях от внешних организаций или о действиях коллег по подготовке ответов. Это приводит к ошибкам и снижению эффективности работы.

В условиях предприятия с более чем 200 подразделениями, на которые ежедневно поступает огромное количество переписки, система учета не справляется с текущими объемами. Завод получает и отправляет более 90,000 писем в год, из которых 10% приходится на КК-3 (около 9,000 писем в год). При этом, доступ к письмам ограничен, и поиск нужной информации становится трудоемким процессом.

2) Проблемы учета извещений об изменении

Отсутствие электронного учета извещений об изменении приводит к неэффективному документообороту. Все данные хранятся в бумажных журналах, что затрудняет поиск и контроль.

Ручной процесс регистрации извещений увеличивает вероятность ошибок (например, ошибки в записях или их отсутствие).

Отсутствие интеграции с другими процессами (например, с учетом документации) приводит к дополнительным затратам времени и снижению прозрачности.

3) Проблемы учета выпускаемой документации

Учет документации ведется вручную в бумажных журналах, что усложняет контроль за выпущенными документами.

Отсутствие единой системы учета приводит к невозможности быстрого доступа к нужной информации.

4) Проблемы учета договоров

Учет договоров ведется сотрудниками экономического отдела в стандартных офисных приложениях (Microsoft Excel и Word). Однако проблемы возникают из-за того, что сотрудники подразделений не знают, какие договоры в настоящее время действуют, какие сроки и этапы выполнения работ по каждому договору, и по какому договору выпускаются изделия.

Вся информация по договорам хранится в бумажных архивах, и для того, чтобы получить текст или уточнить условия, нужно обращаться к экономистам. Однако у них самих большие архивы, и поиск нужной информации в них занимает много времени. Это значительно затрудняет оперативный доступ к данным и снижает эффективность работы.

Отсутствие централизованного хранилища и системы быстрого поиска по договорным данным приводит к потерям времени и путанице, что влияет на своевременность выполнения работ и контроль за сроками.

5) Общие проблемы текущей системы:

- ручной процесс обработки данных увеличивает трудозатраты и риск ошибок;

- отсутствие централизованного хранилища данных и системы поиска снижает скорость выполнения задач;

- ограниченный доступ к информации затрудняет коммуникацию между сотрудниками и подразделениями;

- система не справляется с большими объемами данных, что приводит к задержкам в обработке и потерям информации.

 Обзор существующих решений

На рынке существуют системы, которые частично решают задачи автоматизации документооборота, однако большинство из них не адаптированы под специфику предприятия.

****1) 1С: Документооборот****

Это комплексное решение для автоматизации документооборота, которое позволяет вести учет документов, регистрировать входящие и исходящие письма, а также контролировать выполнение задач. Однако внедрение этой системы на заводе потребует значительных финансовых и временных затрат. Также система ориентирована на общие задачи документооборота и не учитывает специфику производства, связанного с учетами по изделиям и договорами.

****2) Microsoft**** SharePoint

Платформа для совместной работы с документами, которая обеспечивает централизованное хранение данных и поддержку рабочих процессов. Однако она требует высокой квалификации пользователей и значительных ресурсов для настройки. Кроме того, SharePoint не подходит для интеграции с существующими локальными процессами предприятия без серьезной переработки.

****3) Локальные решения (Excel, бумажные журналы)****

Большинство предприятий, аналогичных КК-3, используют простые инструменты, такие как **Excel** или бумажные журналы, для учета переписки и договоров. Эти решения не требуют значительных вложений, но не обеспечивают ни безопасности, ни удобства работы, особенно в условиях увеличивающегося объема данных.

****Вывод из обзора существующих решений:****

Существующие системы либо слишком дорогие и сложные для интеграции, либо не решают проблемы конкретного предприятия. Поэтому необходимо разработать специализированное веб-приложение, адаптированное к специфике документооборота КК-3.

****Выводы из анализа предметной области****

На основе анализа текущего состояния документооборота и обзора существующих решений можно сделать следующие выводы:

1) Существующая система учета переписки, извещений об изменении, выпускаемой документации и договоров не соответствует требованиям современного предприятия.

2) Основные проблемы связаны с отсутствием автоматизации, структурированного хранения данных и поиска информации.

3) Использование стандартных офисных приложений и бумажных журналов не позволяет эффективно работать с большим объемом информации, что снижает производительность и увеличивает риск ошибок.

4) Существующие рыночные решения не адаптированы под специфику предприятия и требуют значительных затрат на внедрение.

Таким образом, для решения выявленных проблем и повышения эффективности документооборота необходимо разработать веб-приложение, которое обеспечит:

- централизованное хранение данных;

- автоматизированную регистрацию и обработку документов;

- удобную систему поиска и фильтрации;

- интеграцию с существующими процессами учета.

Концептуальный анализ разработки веб-приложения для автоматизации документооборота

Цели и задачи разработки системы

Веб-приложение для автоматизации документооборота в КК-3 направлено на решение следующих задач:

**Автоматизация учета переписки** – создание системы, которая позволит регистрировать и классифицировать входящие и исходящие письма, а также предоставит удобный интерфейс для поиска и отслеживания корреспонденции.

**Автоматизация учета договоров** – разработка функционала для отслеживания активных договоров, их сроков и статусов, а также возможность быстрого поиска текста договора и соответствующих документов.

**Автоматизация учета документации** – создание системы для ведения учета выпускаемой документации (как текстовой, так и графической), с возможностью сортировки и поиска по типам документов, датам и другим параметрам.

**Упрощение доступа к данным** – улучшение доступа к необходимой информации для сотрудников подразделений и экономистов, минимизация необходимости в ручном поиске и обращении к архивам.

**Разграничение прав доступа** – обеспечение безопасного доступа к документам для разных категорий пользователей в зависимости от их полномочий.

****Архитектура и выбор технологий****

Для реализации веб-приложения будет использована клиент-серверная архитектура, которая позволит разделить обработку данных на сервере и отображение информации на клиенте.

****Основные компоненты архитектуры:****

**- клиентская часть (Frontend) - и**нтерфейс веб-приложения, реализованный с использованием HTML, CSS, JavaScript, **React**. Для создания удобных и адаптивных компонентов интерфейса будет использоваться библиотека **Material UI**. Это позволит создать современный и интуитивно понятный пользовательский интерфейс.

**- серверная часть (Backend) – серверное** приложение, которое обрабатывает запросы от клиента, выполняет логику работы с данными и взаимодействует с базой данных. Для серверной части будет выбран PHP.

**- база данных (Database) – д**ля хранения данных будет использоваться My**SQL** – реляционная база данных, которая обеспечит надежность и удобство работы с большими объемами информации.

**- безопасность данных - для** обеспечения безопасности данных и защиты от несанкционированного доступа будут использоваться современные методы шифрования, включая **JWT (JSON Web Tokens)** для аутентификации и авторизации пользователей.

****Описание функциональных требований****

****Технические требования:****

**1) поддержка браузера – в**еб-приложение должно быть оптимизировано для работы в Mozilla Firefox, так как этот браузер является основным в организации. Поддержка других браузеров (например, Yandex, Chrome или Edge) может быть ограничена, т.к. они не являются приоритетными.

**2) технические характеристики серверной части:**

- серверная часть приложения будет работать на **PHP 7.3**, что связано с текущими стандартами разработки в организации;

- база данных будет использовать MySQL **5.5**, что необходимо для совместимости с существующими инфраструктурными решениями.

**3) инфраструктурные требования:**

- приложение должно быть интегрировано в существующую инфраструктуру предприятия, без необходимости значительных изменений в серверной части;

- серверное окружение должно поддерживать работу с PHP 7.3 и   
MySQL 5.5 для корректной работы приложения.

****Учет переписки:****

- регистрация входящих и исходящих писем с возможностью привязки к конкретному изделию или подразделению;

- автоматическая сортировка переписки по категориям (например, по подразделению, теме письма, дате и т.д.);

- возможность быстрого поиска по архиву переписки с фильтрацией, по ключевым словам, типу письма, подразделению и срокам;

- возможность для сотрудников подразделений и руководителей видеть только те письма, которые им необходимы.

****Учет договоров:****

- хранение данных по действующим договорам (сроки, статусы, даты начала и окончания, этапы разработки);

- возможность привязки договоров к изделиям, которые выпускаются по этим договорам;

- быстрый доступ к текстам договоров и возможность их просмотра в интерфейсе;

- возможность загрузки и хранения сканов или PDF-версий договоров.

****Учет документации:****

- реализация централизованного хранилища для текстовой и графической документации;

- классификация документов по типам, подразделениям и статусам;

- функции для скачивания, загрузки и просмотра документации через веб-интерфейс.

****Управление пользователями и правами доступа:****

- модуль аутентификации;

- разграничение прав доступа для различных пользователей (например, для сотрудников подразделений, экономистов, руководителей).

****Проектирование базы данных****

Для реализации функционала системы необходимо спроектировать структуру базы данных, которая будет поддерживать хранение и поиск информации по переписке, извещениям, договорам и документации.

Основные требования к возможностям базы данных:

1) журнал переписки:

а) учет входящих писем, с сортировкой по теме, кратким описанием и отсканированным оригиналом в приложении;

б) учет исходящих писем, с сопоставлением ответа к входящему письму с кратким описанием, указанием ответственного за ответ, приложением отсканированный оригинал ответа;

в) поиск писем по дате, номеру, описанию, адресу, исполнителю;

2) журнал извещений об изменении:

а) учет извещений по теме и номеру с кратким содержанием, основанием и кодом изменения по ГОСТ 2.503-2013, отсканированный оригинал извещения, дата сдачи в учет;

б) сортировка извещений по номеру или дате выпуска;

в) поиск извещений по номеру, описанию, исполнителю, дате выпуска;

3) журнал учета выпущенных документов:

а) учет разработанных документов по теме, наименованию выпускаемого документа, отдел, разработчик и дата сдачи в архив;

б) поиск по дате, наименованию, фамилии, отделу;

4) учет договоров комплекса № 3:

а) договорные документы;

б) этапы договора;

в) планируемая трудоемкость;

г) техническое задание;

5) учет срочной переписки:

а) выделение в спец категорию писем с контролируемой датой ответа;

б) отслеживание срока до окончания времени ответа с подсветкой цветом писем, в зависимости от времени до окончания срока ответа;

6) раздел поиска документов по разным разделам;

7) построение каталога изделий комплекса:

а) простой список и сортировкой по алфавиту и поиску по индексу;

б) дерево каталогов, изделия отсортированы по темам, также имеется поиск по индексам;

в) избранное, составляется в виде простого списка из тем, которые выбрал пользователь для быстрого доступа к разделим изделий;

8) авторизация в системе.

****Процесс разработки и этапы реализации****

****Этап 1 – анализ требований и проектирование архитектуры****

**Цель этапа**: Сбор и уточнение всех функциональных и нефункциональных требований к системе, а также проектирование архитектуры системы и базы данных.

**Сбор требований**:

- анализ текущих бизнес-процессов на предприятии, связанных с документооборотом, перепиской и учётом договоров;

- проведение интервью с ключевыми пользователями (начальник комплекса, начальники отделов, сотрудниками экономического отдела и сотрудники отделов) для понимания их нужд;

- определение ключевых задач, которые система должна решать (например, упрощение поиска и обработки писем, улучшение учета договоров, оптимизация хранения документов);

- составление списка функциональных требований (например, возможность поиска по письмам и документам, создание отчетов по договорам и срокам их выполнения, возможность выгрузки данных в различные форматы);

- определение нефункциональных требований (например, производительность системы, безопасность данных, доступность, масштабируемость).

**Проектирование архитектуры системы**:

- разработка **архитектуры клиент-сервер**: выбор технологий для клиентской части (React, HTML, CSS, JavaScript) и серверной части (PHP);

- определение структуры базы данных: проектирование **ER-диаграммы**, создание моделей данных для таблиц (например, изделия, переписка, договора, документы, извещения и т.д.);

- разработка схемы **взаимодействия компонентов системы**: какие данные будут передаваться между клиентом и сервером, какой API будет использоваться, какие будут задействованы сторонние сервисы или базы данных;

- разработка **моделей безопасности**: определение, какие данные будут храниться в защищенном виде, как будет реализована аутентификация и авторизация пользователей, управление доступом к различным данным.

**Подготовка к реализации**:

- определение технологии разработки и инструментов для тестирования.

- планирование этапов работы с учетом сложности задач и сроков.

- разработка плана по защите данных и обеспечения их безопасности.

**Ожидаемый результат**: подробный документ с требованиями и техническими характеристиками системы, готовая архитектура системы, проект базы данных, список задач для дальнейшей разработки.

****Этап 2 – разработка интерфейса и серверной части****

**Цель этапа**: Реализация пользовательского интерфейса и серверной логики системы. Включает разработку всех основных компонентов системы, начиная от клиентской части и заканчивая серверной.

**Разработка клиентской части (интерфейса)**:

**- проектирование интерфейса**: создание макетов пользовательского интерфейса (UI/UX), учитывая удобство пользователей и их роль в системе (например, экономисты, сотрудники подразделений);

**- разработка фронтенда**: реализация интерфейса с использованием **HTML, CSS и JavaScript** (с использованием фреймворка **React**);

**- реализация взаимодействия с сервером;**

**- проверка удобства использования интерфейса** (например, провести тестирование с конечными пользователями, чтобы улучшить взаимодействие).

**Разработка серверной части**:

**- создание серверной логики** с использованием **PHP 7.3**;

- реализация работы с базой данных MySQL: создание **SQL-запросов**, организация связей между таблицами, создание процедур для работы с данными;

- реализация функционала для **обработки переписки, документов и договоров** (например, обработка входящих и исходящих писем, создание отчетов по договорам);

- реализация системы **авторизации и аутентификации пользователей**: создание системы прав доступа, разграничение пользователей по ролям;

**- Обработка ошибок**: создание системы логирования и обработки ошибок;

**Интеграция с внешними системами** (если необходимо):

- интеграция с корпоративными базами данных (например, если предприятие использует другие системы учета);

- интеграция с почтовыми системами для автоматической обработки входящих писем.

**Ожидаемый результат**: Рабочая версия системы с полностью реализованным интерфейсом и серверной логикой, готовая к тестированию. Система должна включать все ключевые функциональности: обработку переписки, учет договоров, извещений, работу с документами.

****Этап 3 – интеграция и тестирование****

**Цель этапа**: завершение разработки, интеграция всех компонентов системы и тестирование на наличие ошибок.

**Интеграция компонентов**:

- интеграция клиентской части с серверной: подключение интерфейса к серверным API, проверка правильности работы запросов и ответов;

- интеграция с внешними системами предприятия (например, с учетными системами или почтовыми серверами);

- проверка корректности работы всех связей между базами данных (например, правильная обработка записей в таблицах).

**Тестирование системы**:

**- функциональное тестирование**: проверка всех ключевых функций системы (например, создание новых записей, загрузка файлов в систему, работа с договорами и документацией);

**- тестирование безопасности**: проверка системы на уязвимости (например, попытки несанкционированного доступа, утечка данных);

**- тестирование производительности**: проверка, как система работает при большом объеме данных;

**- тестирование удобства использования**: тестирование интерфейса с реальными пользователями для выявления проблем в пользовательском опыте;

**- тестирование на совместимость**: проверка работы системы в разных браузерах и на разных устройствах, включая **Firefox**, так как это является требованием заказчика.

**Исправление ошибок и доработка**:

- анализ результатов тестирования, исправление выявленных ошибок;

- оптимизация системы для улучшения производительности и устранения слабых мест;

- проведение повторного тестирования после внесения изменений.

**Ожидаемый результат**: система, которая прошла полное тестирование, интеграцию и готова к внедрению. Все ошибки исправлены, система работает стабильно и соответствует всем требованиям.

****Этап 4 – внедрение и обучение пользователей****

**Цель этапа**: внедрение системы в эксплуатацию, обучение сотрудников работе с системой и поствнедренческая поддержка.

**Внедрение системы**:

- развертывание системы на производственном сервере предприятия;

- обеспечение стабильной работы системы в рабочей среде предприятия;

- перенос данных (если необходимо) с существующих систем в новую систему.

**Обучение пользователей**:

- проведение обучающих сессий для сотрудников, которые будут работать с системой;

- разработка и передача **инструкций и руководств пользователя**, включая обучающие материалы по использованию интерфейса и выполнению основных операций (например, создание новых записей, загрузка файлов в систему, удаление записей и файлов, общие правила заполнения таблиц и разделов, принцип работы поиска и сортировки);

- ответы на вопросы пользователей и помощь в освоении системы.

**Поддержка после внедрения**:

- обеспечение **поствнедренческой поддержки** для решения возникающих проблем;

- мониторинг работы системы и устранение возможных ошибок;

- регулярные обновления системы, исправления багов и добавление новых функций по мере необходимости.

**Ожидаемый результат**: Система внедрена и функционирует на предприятии, сотрудники обучены, и система готова к повседневному использованию. Обеспечена поддержка пользователей.

****Оценка эффективности****

Целью внедрения разработанного веб-приложения является решение актуальных проблем документооборота, учета договоров и выпускаемой документации на предприятии, а также повышение скорости и качества обработки информации. В данном разделе оценивается, как новая система помогает достигнуть поставленных целей.

****Сравнение текущего состояния и внедренной системы****

****До внедрения системы:****

**Обработка переписки**:

- все письма обрабатывались вручную, архив не структурирован;

- доступ к архиву переписки был ограничен для сотрудников подразделений, что замедляло процесс работы;

- в среднем обработка одного письма занимала до 1-2 рабочих дней, учитывая пересылку документов и ожидание ответов от сотрудников.

**Учет договоров**:

- сотрудники подразделений не имели доступа к актуальной информации по договорам.

- сложности в поиске текста договора или уточнении его условий из-за бумажных архивов.

- поиск одного договора занимал до 1-2 часов рабочего времени.

**Учет документации и извещений**:

- отсутствие электронной системы учета выпусков документации.

- учет велся вручную, что увеличивало риск ошибок и потерь данных.

**Проблемы взаимодействия между подразделениями**:

- процесс согласования писем и документов был затруднен из-за отсутствия прозрачной системы контроля и отслеживания.

****После внедрения системы:****

**Обработка переписки**:

- вся переписка структурирована по изделиям, подразделениям и темам.

- сотрудники подразделений имеют доступ к архиву писем, что исключает необходимость обращения к канцелярии.

- обработка одного письма сокращается до нескольких часов, благодаря автоматизации пересылки, регистрации и поиска.

**Учет договоров**:

- у сотрудников подразделений появился доступ к списку активных договоров, срокам выполнения работ и текстам договоров.

- вся информация доступна через интерфейс веб-приложения.

- время поиска договора сокращено до 1-2 минут.

**Учет документации и извещений**:

- вся документация хранится в электронном виде с привязкой к изделиям и договорам.

- реализована автоматическая регистрация и контроль выпуска документации.

**Повышение взаимодействия между подразделениями**:

- система позволяет отслеживать статус писем, договоров и документов.

- реализована прозрачная система согласований и контроля.

****Оценка экономической эффективности****

Экономия времени **сотрудников:**

- **переписка**: сокращение времени обработки одного письма на 8 нормочасов. Учитывая объём в более чем 90,000 писем в год, из которых 10% приходится на КК-3 (примерно 9,000 писем), общее время экономии для подразделения составляет 72,000 нормочасов.

- **поиск договоров**: уменьшение времени поиска одного договора с   
1-2 часов до 2 минут. Для подразделения с активным участием в договорах это экономит сотни рабочих часов в год.

- **документация**: упрощение учета снижает нагрузку на сотрудников документационного отдела.

**Снижение риска ошибок:**

- учет документации и договоров в электронном виде позволяет избежать потерь данных, которые могли возникать при бумажном документообороте.

- автоматизация процессов снижает вероятность человеческого фактора.

**Экономия на бумажном документообороте:**

- уменьшение необходимости хранения бумажных архивов благодаря переходу на электронный учет.

**Расчет экономии:**

- допустим, один день нормо-час работы сотрудника стоит в среднем 500 рублей.

- Экономия времени только на обработке переписки составляет:  
72000 н/ч × 500 руб. = 36,000,000 руб. в год.

Эта сумма может быть увеличена, если учесть экономию на других процессах (поиск договоров, учет документации).

****Оценка социальной эффективности****

**Повышение качества работы сотрудников:**

- доступ к архивам переписки и информации по договорам позволяет сотрудникам быстрее принимать решения и выполнять свою работу;

- снижение уровня стресса, связанного с длительным поиском информации.

**Улучшение взаимодействия между подразделениями:**

- прозрачность процессов снижает число конфликтных ситуаций и недопонимания;

- возможность отслеживать статус задач в реальном времени повышает уровень доверия между отделами.

**Улучшение клиентского сервиса:**

- быстрое реагирование на запросы внешних организаций благодаря сокращению времени обработки входящих писем и подготовки ответов.

****Оценка технической эффективности****

**Совместимость с существующей инфраструктурой:**

- веб-приложение успешно работает на существующем сервере предприятия и поддерживает браузер **Firefox**.

**Масштабируемость:**

- система легко масштабируется для использования другими подразделениями предприятия.

**Снижение нагрузки на ИТ-отдел:**

- упрощение процессов администрирования и поддержки системы.

****Оценка внедрения и возврата инвестиций****

**Стоимость разработки и внедрения:**

- разработка веб-приложения велась в рамках дипломного проекта с минимальными затратами на оборудование и программное обеспечение (использовались существующие ресурсы предприятия).

**Срок окупаемости:**

- при экономии 36,000,000 рублей в год только на обработке переписки система окупается в течение первого года эксплуатации.

**Вывод: р**азработанная система позволяет значительно сократить время обработки информации, повысить прозрачность процессов и улучшить взаимодействие между подразделениями. Экономический эффект от внедрения системы превышает затраты на её разработку, что делает проект перспективным для дальнейшего масштабирования на другие подразделения завода.

Реализация веб-приложения

****Выбор технологий****

 Для разработки веб-приложения, предназначенного для учета переписки, договоров и документации на предприятии, было принято решение использовать набор современных и проверенных технологий. Этот выбор был сделан с учетом специфики работы на предприятии, существующей инфраструктуры и требований к функциональности, производительности и безопасности системы.

**Фронтенд (Клиентская часть)**

Для разработки клиентской части веб-приложения были выбраны **HTML5, CSS3,** **JavaScript,** React.

Причины выбора HTML5 и CSS3:

- HTML5 был выбран для разметки веб-страниц, поскольку это стандартная современная технология для создания структуры веб-страниц, которая поддерживается всеми современными браузерами и позволяет использовать новые возможности, такие как мультимедиа и геолокация;

- CSS3 – используется для стилизации элементов интерфейса, создания адаптивных страниц с использованием медиазапросов и современных технологий для улучшения внешнего вида.

Причины выбора JavaScript:

- JavaScript является основным языком для разработки клиентской логики, взаимодействия с пользователем и работы с API. Он необходим для реализации динамических элементов интерфейса, таких как формы, таблицы с данными, а также для обработки событий.

Причины выбора React:

- реактивность и компоненты: React использует подход на основе компонентов, что позволяет создавать гибкие и модульные интерфейсы. Это особенно важно для веб-приложений, где множество компонентов могут изменяться независимо друг от друга;

- производительность: React использует виртуальный DOM (Document Object Model), что позволяет эффективно обновлять только те части страницы, которые были изменены, минимизируя количество перерисовок и улучшая производительность, особенно при работе с большими объемами данных;

- широкая поддержка и сообщество: React имеет обширное сообщество разработчиков и большую библиотеку готовых решений и компонентов (например, Material-UI), что значительно ускоряет процесс разработки.

Причины выбора Material-UI:

- Для упрощения и ускорения разработки интерфейса был выбран Material-UI – библиотека компонентов для React. Это решение основано на принципах дизайна Material Design от Google, что позволяет создать современный, интуитивно понятный и визуально привлекательный интерфейс.

- В Material-UI уже есть готовые компоненты для работы с кнопками, формами и т.д., что позволяет существенно ускорить процесс разработки приложения.

**Бэкенд (Серверная часть)**

Для серверной части приложения был выбран **PHP 7.3**, который является одной из наиболее стабильных и распространенных версий этого языка программирования, поддерживающих все необходимые функциональные возможности для работы с серверной логикой и базой данных.

Причины выбора PHP 7.3:

- совместимость с существующей инфраструктурой: веб-приложение должно работать на существующем сервере предприятия, который уже настроен для работы с PHP. Это снижает затраты на внедрение и настройку;

- простота интеграции с MySQL: PHP имеет встроенные функции и расширения для работы с базами данных MySQL, что упрощает взаимодействие с системой хранения данных;

- быстрая обработка запросов: PHP позволяет быстро обрабатывать HTTP-запросы и отправлять ответы клиенту, что критически важно для работы с большими объемами данных и множества пользователей одновременно.

Для хранения данных о переписке, договорах, документации и других сущностях выбрана реляционная база данных MySQL.

Причины выбора MySQL 5.5:

**-**стабильность и поддержка: MySQL 5.5 является достаточно старой, но стабильной версией базы данных, которая используется на сервере предприятия. Она предоставляет необходимую функциональность для работы с данными.

- поддержка реляционных данных: база данных MySQL идеально подходит для реляционных структур данных, таких как таблицы для писем, документов и договоров, которые могут быть связаны друг с другом через уникальные идентификаторы;

- производительность: MySQL поддерживает индексирование, которое ускоряет поиск по таблицам и работу с большими объемами данных.

Другие варианты баз данных:

- в качестве альтернативы можно было рассмотреть PostgreSQL или SQLite, но MySQL была выбрана, поскольку это уже используется на предприятии, и нет необходимости в изменении инфраструктуры.

**Инфраструктура и сервер**

Веб-приложение должно работать на существующем сервере предприятия, что позволяет минимизировать затраты на инфраструктуру и использовать уже имеющиеся ресурсы. Сервер настроен на работу с PHP и MySQL, что обеспечивает совместимость с выбранными технологиями.

**Использование существующего сервера**: разработка приложения для работы на сервере, который уже используется предприятием, помогает избежать дополнительных затрат на приобретение нового оборудования и настройку новых серверов.

**Производительность сервера**: важно, чтобы сервер мог обрабатывать значительные объемы данных. Система должна быть готова к высоким нагрузкам, что было учтено при выборе технологии.

****Проектирование базы данных****

На основе проведенного анализа было принято решение создать следующие таблицы.

****Основные таблицы базы данных:****

**таблица пользователей (users) – х**ранит информацию о пользователях системы, их ролях и правами доступа;

Таблица 3. - Структура таблицы users

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | int(11) auto\_increment | Уникальный идентификатор пользователя |
| login | text | Логин пользователя |
| pass | text | Хэш пароля |
| other | longtext | Вся информация о пользователе в формате JSON |
| type | int(11) | Статус пользователя |

 таблица изделие (izdelie) – хранит информацию о наименованиях выпускаемых изделий, а также информацию для построения дерева каталогов изделий статусе проекта

Таблица 3. - Структура таблицы izdelie

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | int(11) auto\_increment | Уникальный идентификатор изделия |
| hide | int(11) | Безопасное удаление изделия |
| name | text | Короткое имя изделия |
| fullname | text | Полное наименование проекта изделия |
| notactive | int(11) | Статус договора по изделию |
| sort | int(11) | Порядок сортировки в списке изделий |
| catalog | int(11) | id каталога, для дерева каталогов |

 таблица каталогов (catalog) – содержит информацию о дереве каталогов изделий и вложенности каталогов друг в друга

Таблица 3. - Структура таблицы catalog

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | int(11) auto\_increment | Уникальный идентификатор каталога |
| name | text | Наименование каталога |
| catalog | int(11) | id каталога, в которое вложена данная запись |

 таблица договоров (dogovor) – содержит информацию о дереве каталогов изделий и вложенности каталогов друг в друга

Таблица 3. - Структура таблицы dogovor

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | int(11) auto\_increment | Уникальный идентификатор договора |
| hide | int(11) | Безопасное удаление |
| detid | int(11) | Идентификатор связи с таблицей izdelie |
| name | text | Наименование темы |
| fullname | text | Полное наименование темы |
| indeks | text | Индекс изделия по договору |
| client | text | Заказчик изделия |
| nomer | text | Номер договора |
| date | text | Дата заключения договора |
| capacity | text |  |
| list | text |  |
| otd | text | Отдел ведущий договор |
| prim | text | Примечание |
| controlvpmo | text | Договор на контроле ВП МО |
| dogopen | tinyint(4) | Статус подписи договора |
| dogclose | tinyint(4) | Триггер закрытия договора |

 таблица этапов договоров (stage) – содержит информацию о этапе договоров, краткое содержание необходимых работ по этапу, дату начала и окончания работ, а также трудоёмкость работы

Таблица 3. - Структура таблицы stage

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | int(11) auto\_increment | Уникальный идентификатор этапа |
| hide | int(11) | Безопасное удаление |
| detid | int(11) | Идентификатор связи с таблицей dogovor |
| nomer | text | Номер этапа по договору |
| content | text | Содержание этапа |
| datestart | text | Дата начала работ по этапу договора |
| datestop | text | Дата окончания работ по этапу договора |
| dateprobablestop | text | Дата вероятного окончания работ по этапу |
| period | text | Вспомогательное поле для расчета трудоемкости |
| report | text | Ссылка на отчетные документы |
| status | text | Статус работы |
| prim | text | Примечание |
| 31trd | text | Трудоемкость отдела 31 |
| 31slg | text | Коэффициент новизны для отдела 31 |
| 32trd | text | Трудоемкость отдела 32 |
| 32slg | text | Коэффициент новизны для отдела 32 |
| 33trd | text | Трудоемкость отдела 33 |
| 33slg | text | Коэффициент новизны для отдела 33 |
| 34trd | text | Трудоемкость отдела 34 |
| 34slg | text | Коэффициент новизны для отдела 34 |
| 35trd | text | Трудоемкость отдела 35 |
| 35slg | text | Коэффициент новизны для отдела 35 |
| upd | tinyint(4) | Статус перерасчета трудоемкости |

 таблица календарь (calendar) – производственный календарь, используется для расчета количества дней по этапам договора и расчета фактической трудоёмкости работ. По умолчанию календарь строится стандартно, т.е. с понедельника по пятницу рабочие дни, суббота и воскресенье выходные. Для изменения используются поля date и holydays.

Таблица 3. - Структура таблицы calendar

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | int(11) auto\_increment | Уникальный идентификатор дня |
| date | text | Дата метки |
| holydays | int(11) | Отметка о выходном или рабочем дне |

 таблица контрагентов по договорам (contragent)– содержит информацию о контрагентах по этапам договоров, их наименование, номер этапа договора, даты начала и окончания работ, отчетные документы.

Таблица 3. - Структура таблицы contragent

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | int(11) auto\_increment | Уникальный идентификатор записи |
| hide | int(11) | Безопасное удаление |
| detid | int(11) | Идентификатор связи с таблицей izdelie |
| nomeretap | text | Номер этапа по договору |
| namecontr | text | Наименование организации контрагента |
| contentwork | text | Краткое содержание выполняемых работ |
| nomdog | text | Номер договора |
| datestart | text | Дата начала работ по договору с контрагентом |
| datestop | text | Дата окончания работ по договору с контрагентом |
| otchet | text | Отчетные документы |
| prim | text | Примечание |

 таблица техническое задание (techzadanie) – содержит информацию о техническом задании, ссылку на полный текст технического задания, дату оформления и статус отработки по документу.

Таблица 3. - Структура таблицы techzadanie

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | int(11) auto\_increment | Уникальный идентификатор записи |
| hide | int(11) | Безопасное удаление |
| detid | int(11) | Идентификатор связи с таблицей izdelie |
| doc | text | Наименование технического задания |
| reason | text | Причина выпуска документа |
| link | text | Ссылка на pdf файл с полным документом |
| date | text | Дата оформления технического задания |
| status | text | Статус работы |
| prim | text | Примечание |

 таблица развитие проекта (history) – содержит информацию о текущем состоянии работы по проекту, заполняется ведущим специалистом для сведения другим сотрудникам.

Таблица 3. - Структура таблицы history

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | int(11) auto\_increment | Уникальный идентификатор записи |
| hide | int(11) | Безопасное удаление |
| detid | int(11) | Идентификатор связи с таблицей izdelie |
| history | mediumtext | Информация о проекте |

 таблица документов, работы и извещений (docwork) – содержит информацию о выпущенных документах и извещениях.

Таблица 3. - Структура таблицы docwork

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | int(11) auto\_increment | Уникальный идентификатор записи |
| hide | int(11) | Безопасное удаление |
| detid | int(11) | Идентификатор связи с таблицей izdelie |
| naimenovenie | text | Наименование выпущенного документа |
| date | text | Дата выпуска |
| numstage | text | Этап договора, по которому выпущено изделие |
| otd | text | Отдел выпускавший документ |
| fio | text | ФИО исполнителя документа |
| codnorm | text | Код норматива |
| normativtruda | text | Норматив трудоемкости |
| kolvoformatov | text | Количество форматов документа |
| planchas | text | Плановая трудоемкость |
| ispolnchas | text | Трудоемкость исполнителя |
| factchas | text | Фактическая трудоёмкость |
| sumnormachas | text | Общая трудоемкость |
| chernovik | text | Ссылка на файл документа (dwg, doc, cdw и т.п.) |
| scan | text | Полный документ в формате pdf (или jpg) |
| gotovnost | int(11) | Готовность работы (по оценке исполнителя) |
| prim | text | Примечание |
| upd | tinyint(4) | Маркер перерасчета для вывода статистики |
| numii | text | Номер извещения об изменении |
| editdoc | text | Индекс корректируемого документа |
| reason | text | Причина изменения |
| codii | text | Код изменения по ГОСТ 2.503-2013 |
| zadel | text | Указание о заделе |

Продолжение таблицы 3.10

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| vnedrenie | text | Указание о внедрении |
| numish | text | Номер исходящего на отправку дубликата |
| trudoemc | text | Трудоемкость |
| doctype | tinyint(4) | Тип таблицы документ/извещение |

 таблица учета переписки (mailbox) – содержит информацию о полученных и отправленных письмах.

Таблица 3. - Структура таблицы mailbox

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | int(11) auto\_increment | Уникальный идентификатор записи |
| hide | int(11) | Безопасное удаление |
| detid | int(11) | Идентификатор связи с таблицей izdelie |
| datevh | text | Дата входящего письма |
| nomervh | text | Номер входящего письма |
| adresvh | text | Адрес отправителя |
| contentvh | text | Краткое описание письма |
| scanvh | text | Копия письма в формате pdf или jpg |
| countlistvh | text | Количество листов в письме |
| sumnormchasvh | text | Трудоемкость обработки письма в н/ч |
| dateish | text | Дата исходящего письма |
| nomerish | text | Номер исходящего письма |
| adresish | text | Адрес получателя |
| contentish | text | Краткое содержание письма |
| scanish | text | Копия письма в формате pdf или jpg |
| countlistish | text | Количество листов в письме |
| fioispish | text | ФИО исполнителя |
| sumnormchasish | text | Трудоемкость написания письма в н/ч |

Продолжение таблицы 3.11

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| upd | tinyint(4) | Маркер перерасчета для вывода статистики |
| datereg | text | Дата регистрации в канцелярии завода |
| nomerreg | text | Регистрационный номер письма |
| datecontrol | text | Дата обязательного ответа |
| prim | text | Примечание |

 таблица файлов (uplfiles) – содержит ссылки на загруженные в дисковое хранилище файлы. Для хранения файлов используется файловая система.

Таблица 3. - Структура таблицы uplfiles

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | int(11) auto\_increment | Уникальный идентификатор файла |
| hide | int(11) | Безопасное удаление |
| tabname | text | Имя таблицы из которой сохранили файл |
| type | int(11) | Столбец, из которого сохранили файл |
| detid | int(11) | Идентификатор связи со строкой из таблицы |
| prefix | int(11) | Префикс для уникальности имени файла |
| filename | text | Имя файла |
| maskname | text | Всплывающая подсказка к ссылке на файл |
| prim | text | Примечание |
| local\_path | text | Путь к файлу в файловой системе |

 таблица настроек (settings) – административная таблица, для настройки отображения колонок и имен полей в таблицах.

Таблица 3. - Структура таблицы settings

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | int(11) auto\_increment | Уникальный идентификатор записи |
| hide | int(11) | Безопасное удаление |
| tablename | text | Имя таблицы |

Продолжение таблицы 3.13

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| field | text | Имя колонки в базе данных |
| showname | text | Отображаемое имя в интерфейсе пользователя |
| size | text | Ширина колонки |
| hidefield | int(11) | Отображение поля |

 таблица логов (log) – административная таблица, для сохранения изменений в таблицах, вносимых пользователями.

Таблица 3. - Структура таблицы settings

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | int(11) auto\_increment | Уникальный идентификатор записи |
| date | date | Дата изменения |
| time | time | Время изменения |
| tab | text | Таблица, в которой произведено изменение |
| field\_id | text | id строки |
| deistvie | text | Данные изменившего |
| content | text | Содержание изменения |

Связи между таблицами:

**Таблица izdelie** является **центральной таблицей** и связывается с другими таблицами через поле id в таблице изделие и поле detid в подчиненных таблицах.

****Нормализация базы данных:****

Для упрощения структуры базы данных и предотвращения дублирования данных, все таблицы будут нормализованы до третьей нормальной формы (3NF), что обеспечит минимизацию избыточности и максимальную эффективность запросов.

****Разработка клиентской части****

Клиентская часть веб-приложения была разработана с использованием HTML5, CSS3 и JavaScript для оформления дизайна страниц, а также React с Material-UI для стилизации компонентов. Интерфейс приложения был спроектирован так, чтобы быть максимально простым и интуитивно понятным для пользователей.

Каждый из экранов приложения разработан с учетом специфики предприятия, а также требований удобства работы с большими объемами данных. В этом разделе представлены основные интерфейсы системы и их описание.

**Окно авторизации**

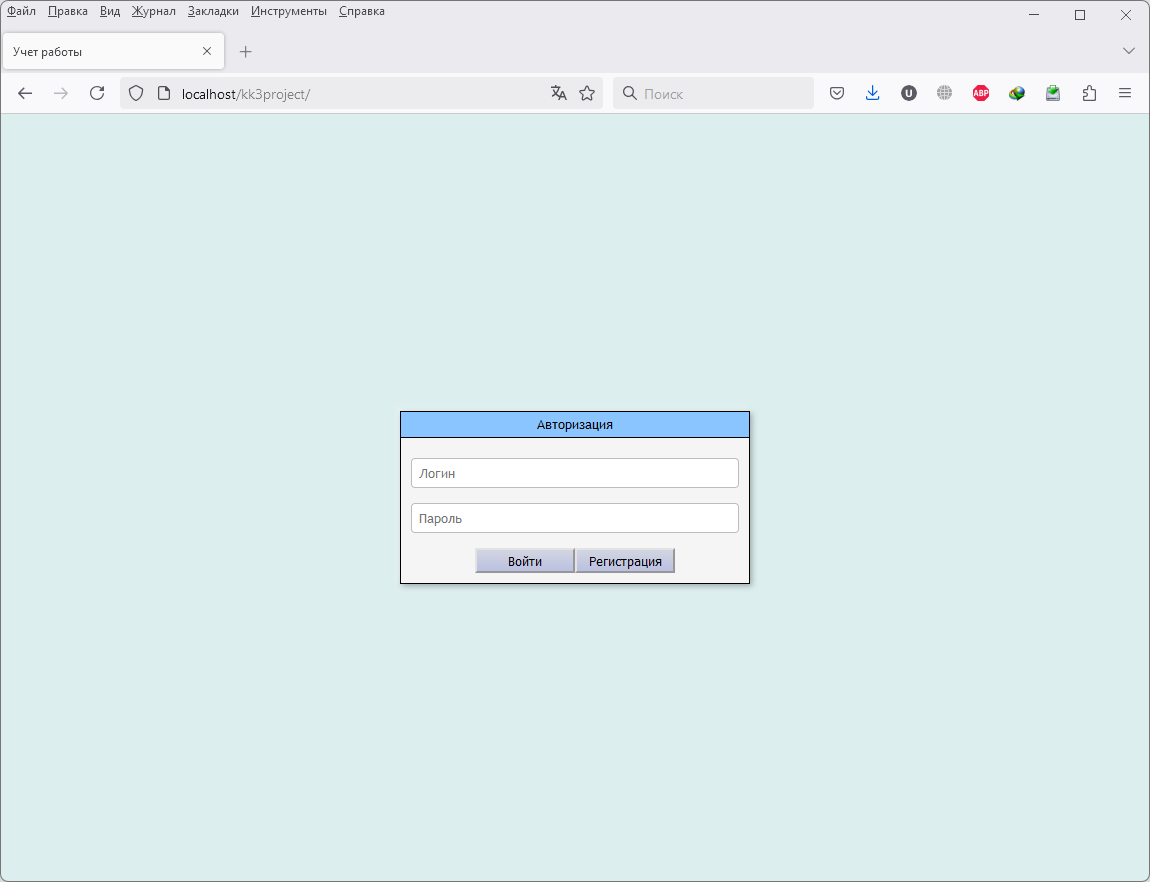


Рисунок - Окно авторизации

Описание:

- окно авторизации представляет собой форму, написанную на React с двумя полями ввода - Логин и Пароль, а также кнопками Войти и Регистрация.

**Окно регистрации**

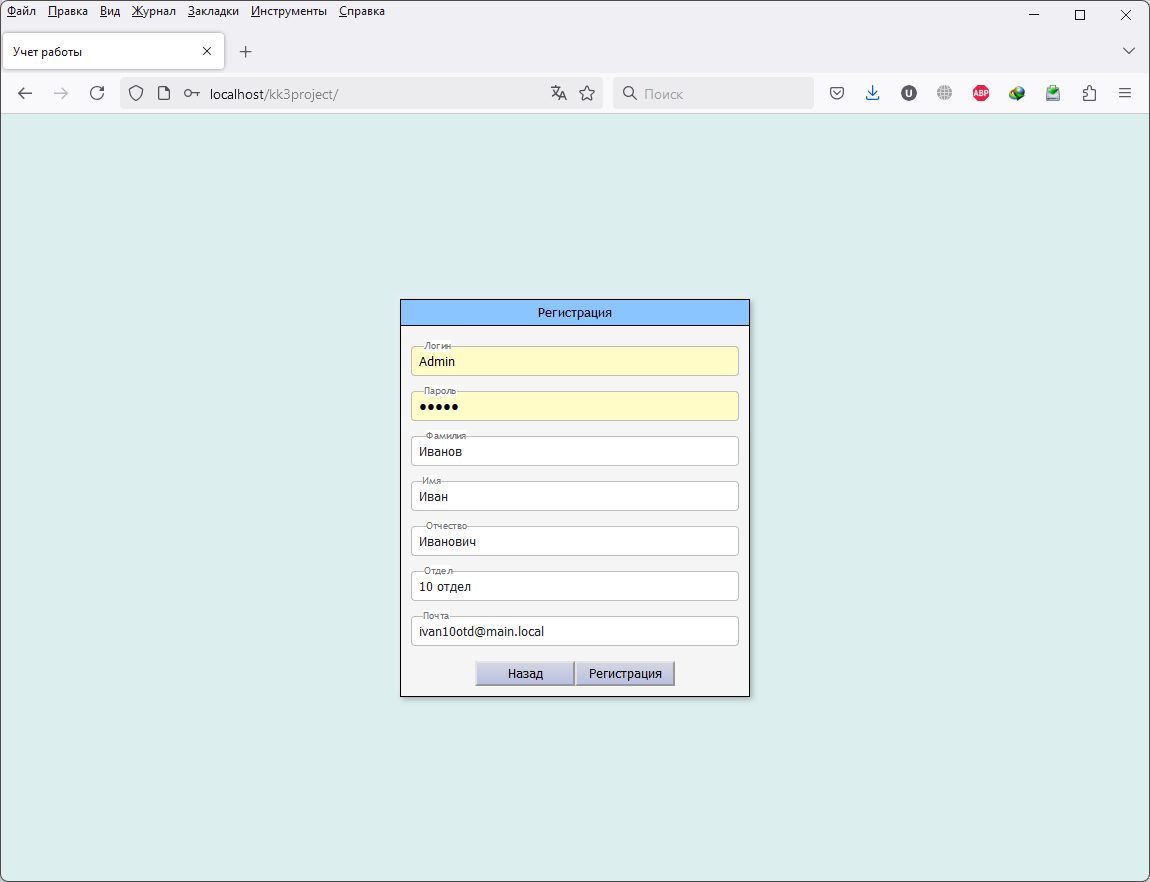


Рисунок - Окно авторизации

Описание:

- окно регистрации с плавающими подсказками;

- все поля обязательны для заполнения;

- если оставить пустое поле, отправка формы не производится, появится подсказка и необходимости заполнить поле.

**Окно ошибка авторизации**

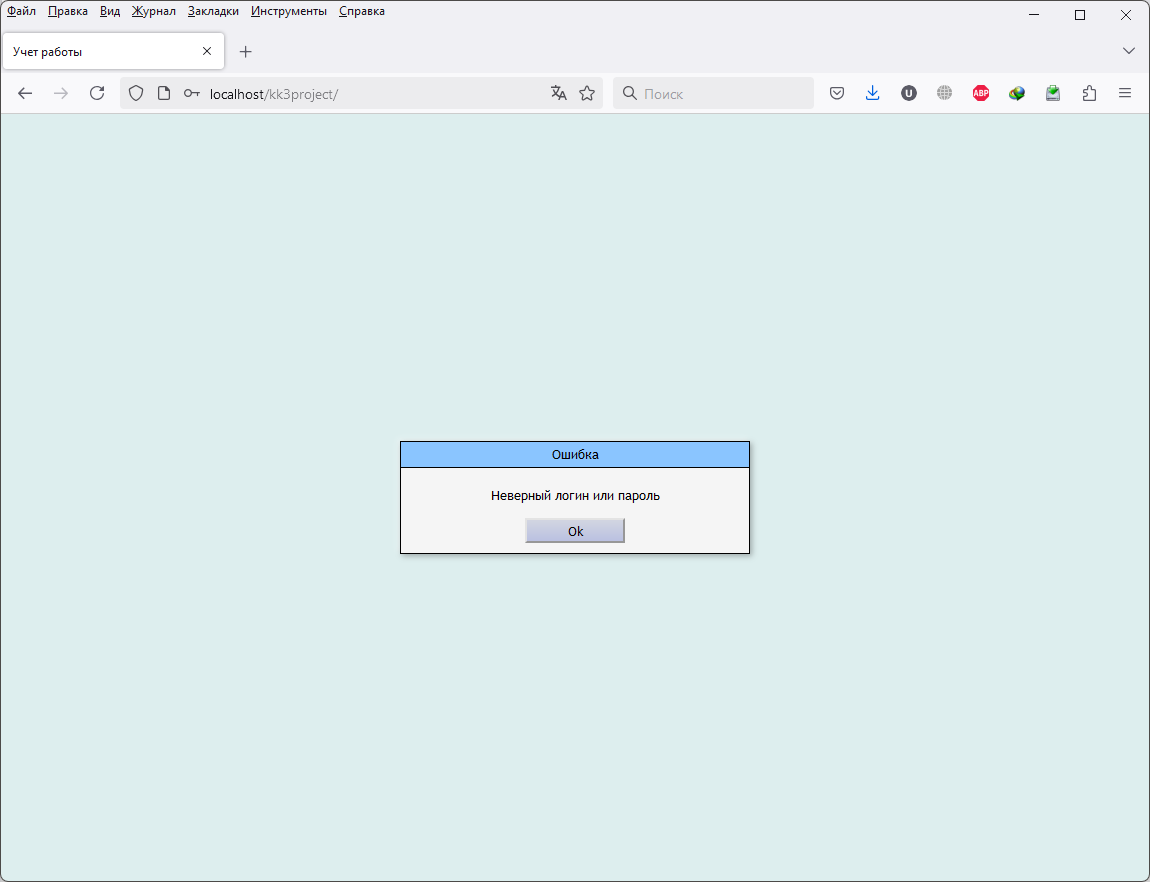


Рисунок - Окно ошибка авторизации

Описание:

- в случае неверного ввода логина или пароля будет выдано окно об ошибке.

Главная страница

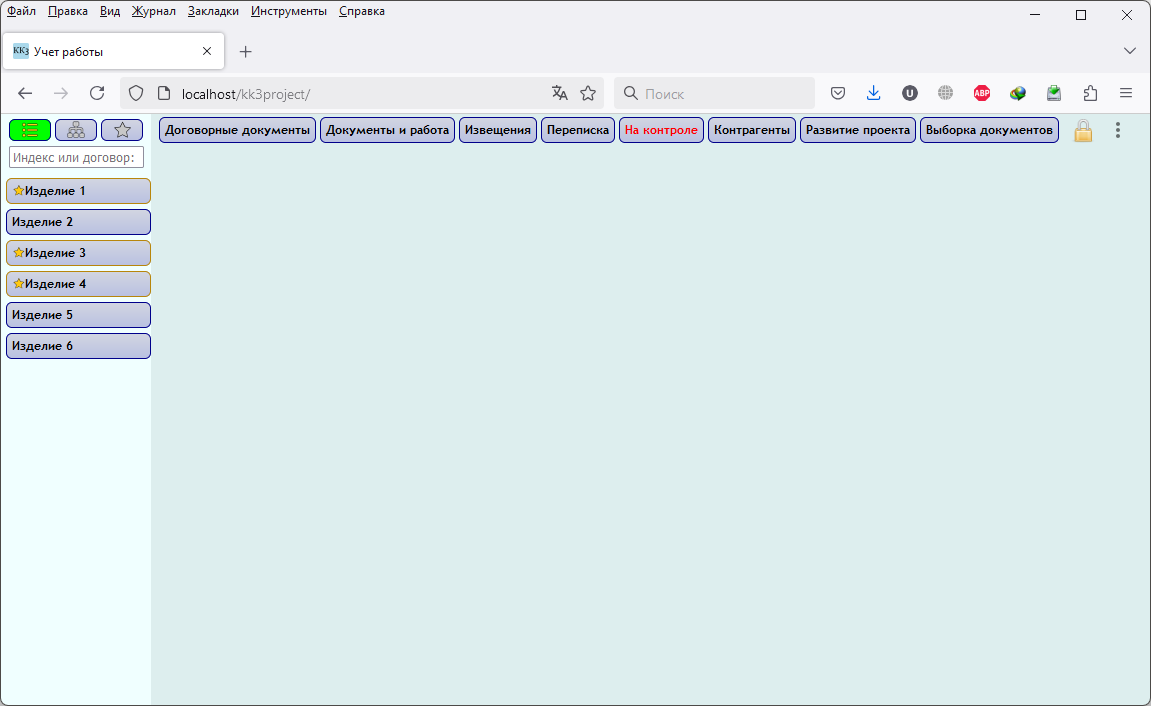


Рисунок  - Главная страница веб-приложения

**Описание:**

**а) главная страница состоит из двух основных элементов - боковое меню (слева) и верхнее меню;**

**б) боковое меню состоит из трех частей:**

**- переключатель отображения списка;**

**- интерактивный поиск/фильтр по индексу изделий или номеру договора;**

**- вертикальное меню.**

**в) верхний блок состоит из:**

**- горизонтального меню;**

**- модуля "замок", представляет собой триггер для разблокировки редактирования;**

**- меню авторизации.**

**г) для загрузки контента необходимо выбрать индекс изделия из боковой части и модуль из верхней.**

Вертикальное меню

|  |  |  |
| --- | --- | --- |
| Рисунок | Рисунок | Рисунок |

Описание:

**- представляет собой список (рисунок 5), дерево каталогов (рисунок 6) или список избранное (рисунок 7), в зависимости от выбранного режима;**

- кнопка "Добавить/Удалить" в режиме "Избранное" открывает окно редактирования избранного.

Редактирование списков изделий

|  |  |  |
| --- | --- | --- |
| Рисунок | Рисунок | Рисунок |

Описание:

- для редактирования изделий и списка изделия необходимо открыть замок в правом верхнем углу страницы (рисунок 4), в нижнем левом углу появится иконка двойных окон (рисунок 8). По нажатию на иконку откроется окно добавления/удаления изделий;

- при удержании кнопки "Ctrl" появится иконка настройки "" (рисунок 9);

- при нажатии на иконку настройки ее цвет изменится "", панель настроек зафиксируется и появится иконка ""

|  |  |
| --- | --- |
| Рисунок | Рисунок |

Описание:

- при удержании "Ctrl" у появится знак крестика, означающий, что каталог можно расформировать (рисунок 11);

- для добавления нового каталога нужно нажать кнопку "", для изменения названия каталога нужно нажать на него правой кнопкой мыши (рисунок 12).

Окно редактирование избранного

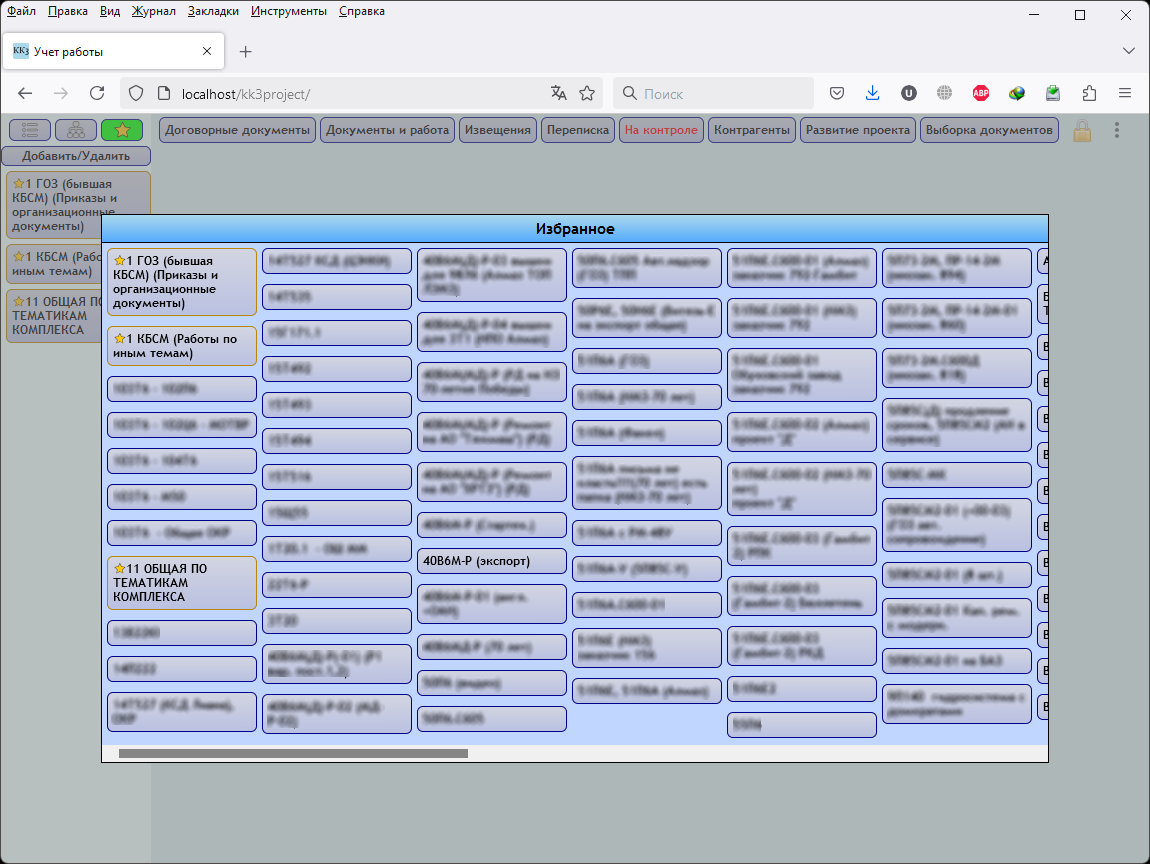


Рисунок - Редактирование избранного (рисунок из рабочей базы данных)

Описание:

- для добавления в избранное выбрать нужные элементы;

- для отмены выбора, выбрать элемент повторно;

- окно закрывается через Esc или при клике мышкой за пределы окна "Избранное".

Окно договорные документы

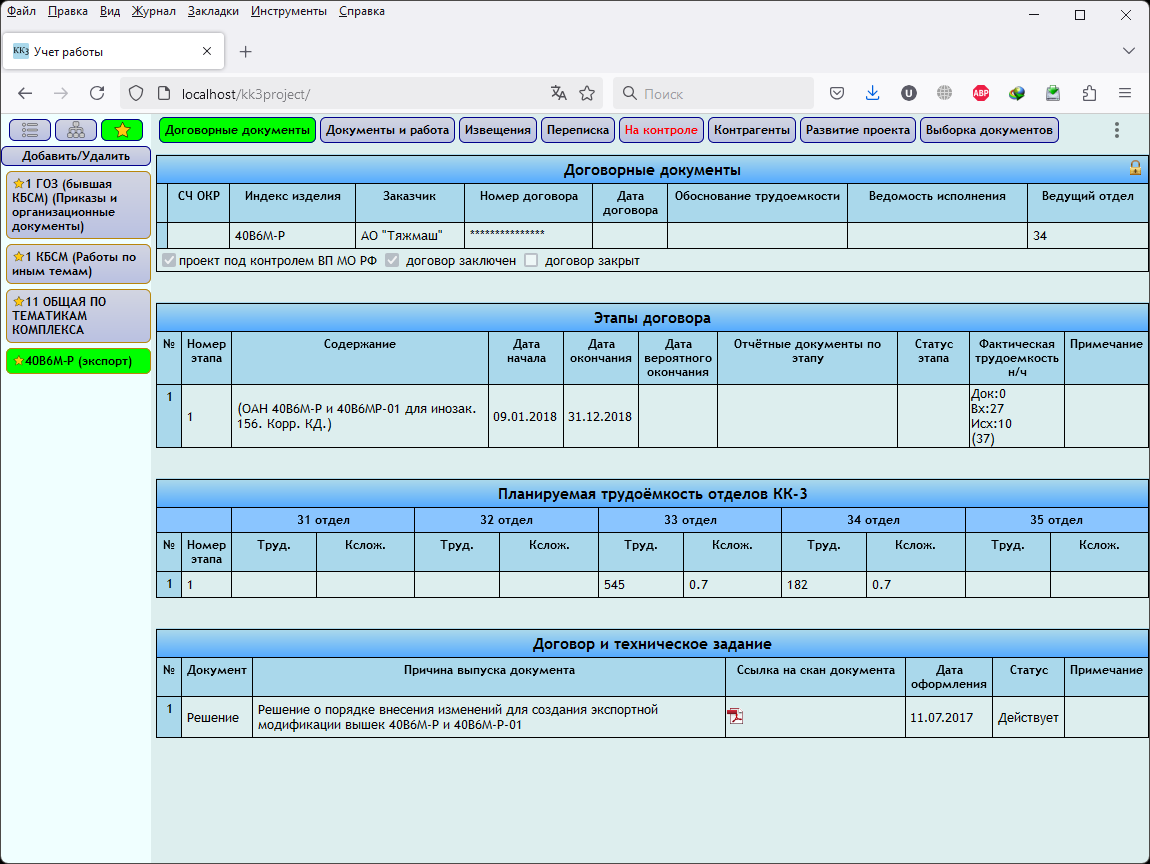


Рисунок - Договорные документы (рисунок из рабочей базы данных)

Описание:

- страница предназначена для редактирования только экономистами

Окно документы и работа

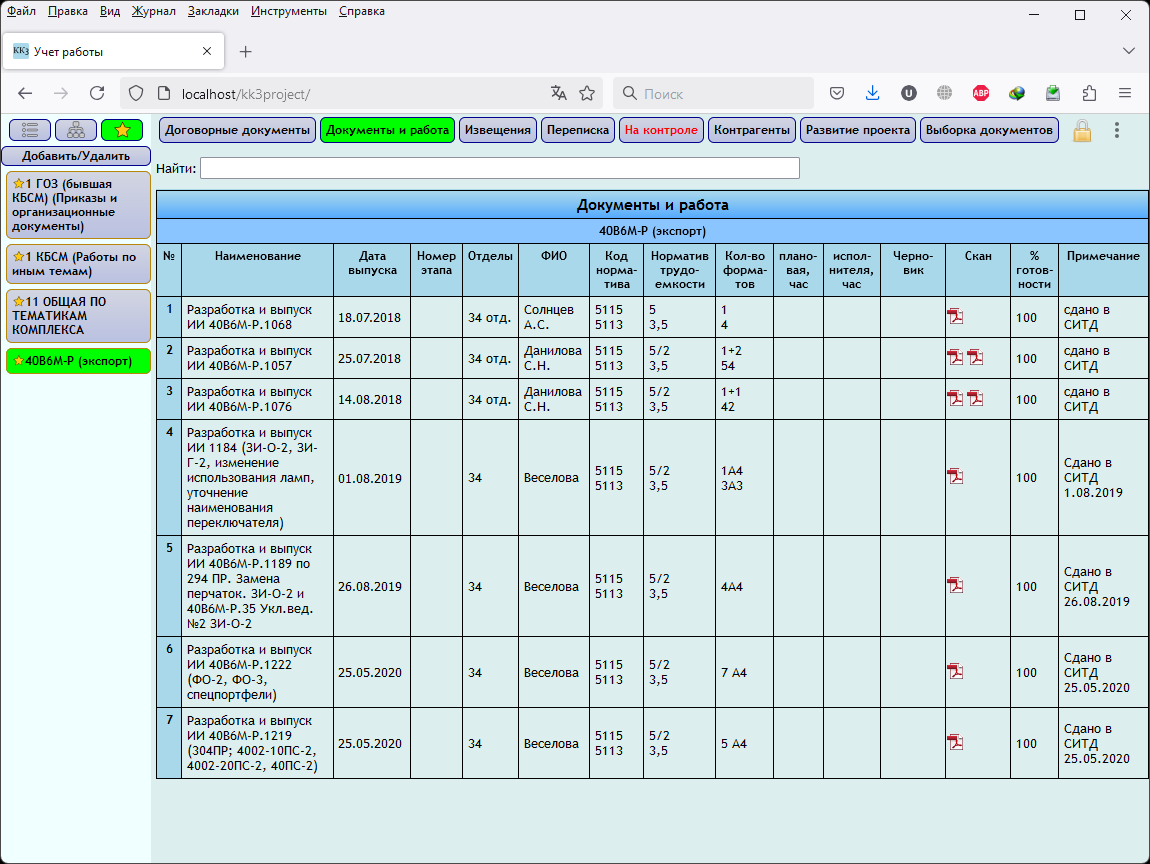


Рисунок - Документы и работа (рисунок из рабочей базы данных)

Описание:

- для редактирования страницы необходимо нажать на замок "закрыто" и открыть его "открыто";

- для добавления новых строк нажать на кнопку "http://localhost/kk3project/include/addline.png" под таблицей;

- для поиска/фильтрации документов использовать поисковую строку над таблицей.

Окно извещения

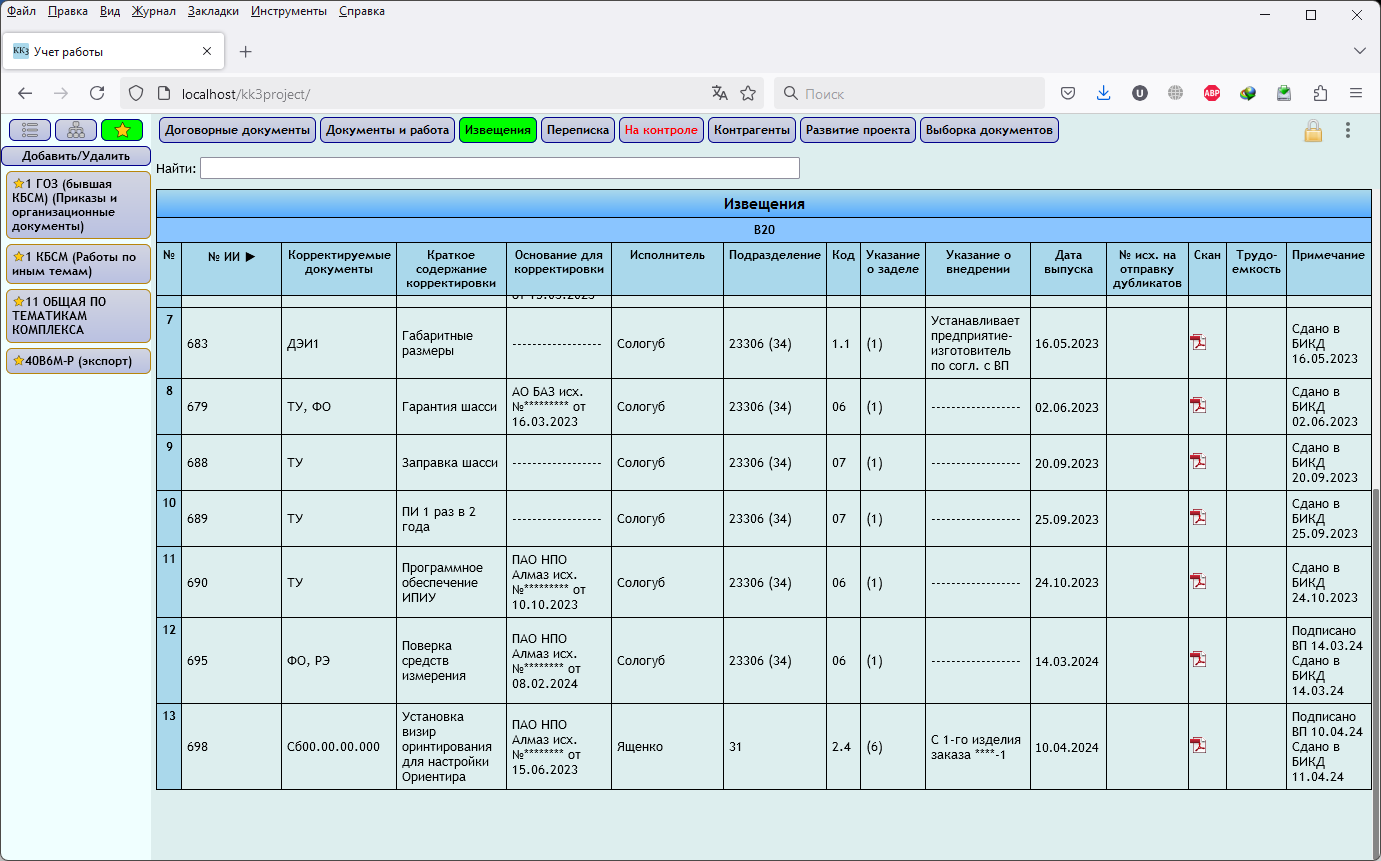


Рисунок - Извещения (рисунок из рабочей базы данных)

Описание:

- для редактирования страницы необходимо нажать на замок "закрыто" и открыть его "открыто";

- для добавления новых строк нажать на кнопку "http://localhost/kk3project/include/addline.png" под таблицей;

- для поиска/фильтрации документов использовать поисковую строку над таблицей.

Окно переписка

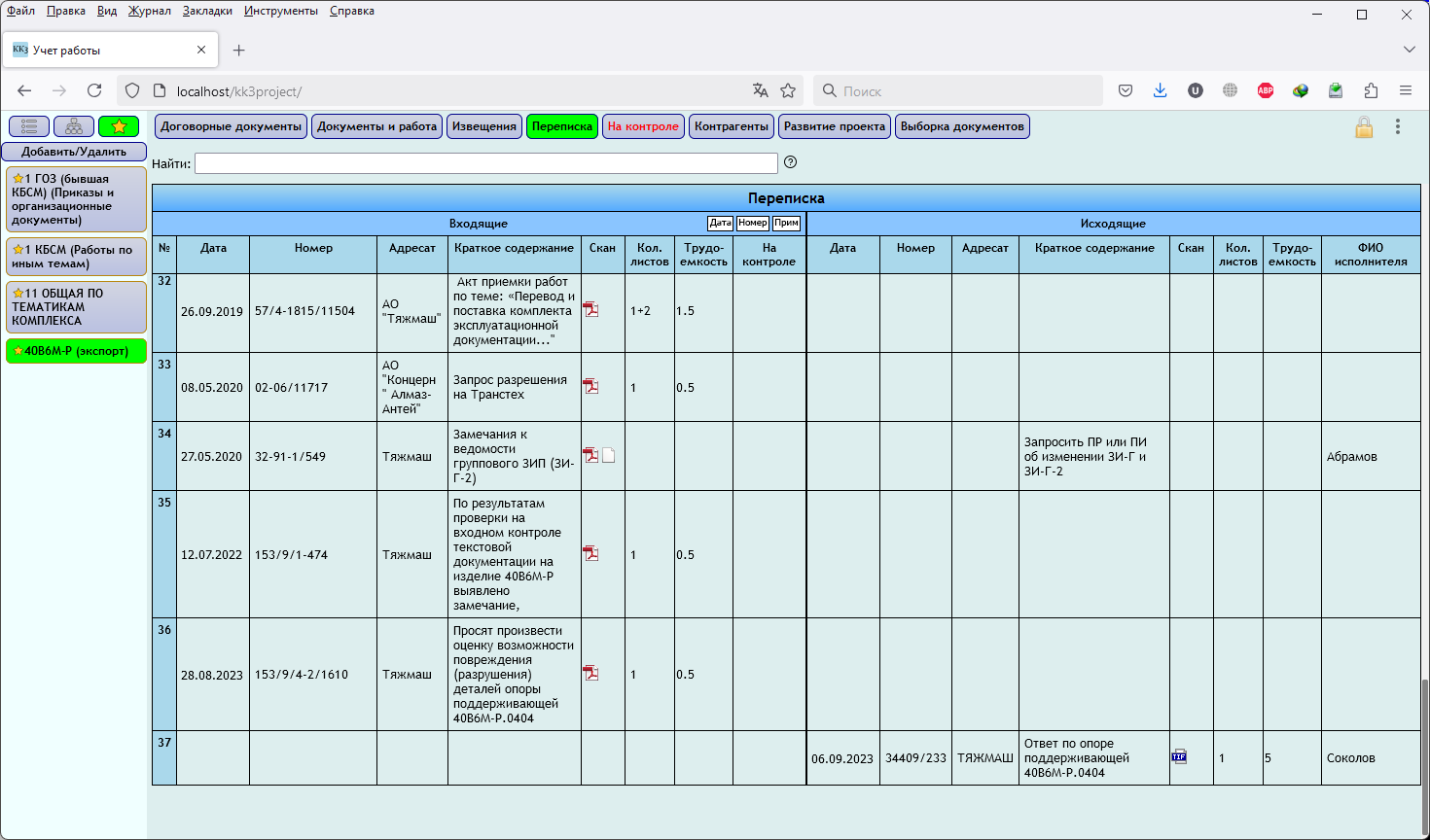


Рисунок - Переписка (рисунок из рабочей базы данных)

Описание:

- таблица состоит из двух частей - входящие и исходящие;

- для ответа на входящий, записывать исходящий напротив письма запроса;

- для редактирования страницы необходимо нажать на замок "закрыто" и открыть его "открыто";

- для добавления новых строк нажать на кнопку "http://localhost/kk3project/include/addline.png" под таблицей;

- для поиска/фильтрации документов использовать поисковую строку над таблицей.

Окно "на контроле"

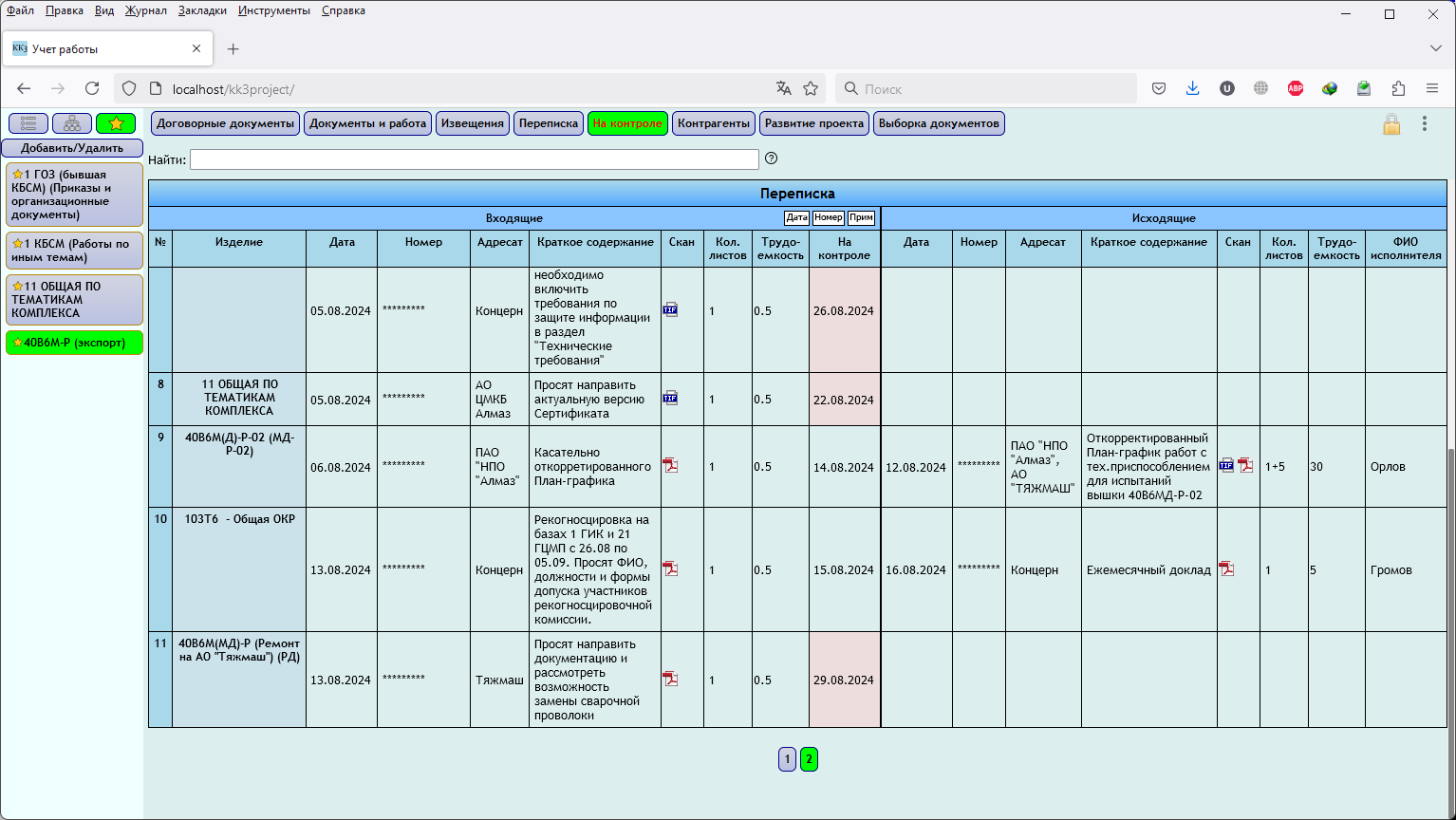


Рисунок - На контроле (рисунок из рабочей базы данных)

Описание:

- таблица состоит из двух частей - входящие и исходящие;

- для ответа на входящий, записывать исходящий напротив письма запроса;

- дополнительное поле "на контроле" указывает на крайний срок ответа на письмо, о просроченном сроке свидетельствует красная маркировка даты;

- для редактирования страницы необходимо нажать на замок "закрыто" и открыть его "открыто";

- для добавления новых строк нажать на кнопку "http://localhost/kk3project/include/addline.png" под таблицей;

- для поиска/фильтрации документов использовать поисковую строку над таблицей.

Окно контрагенты

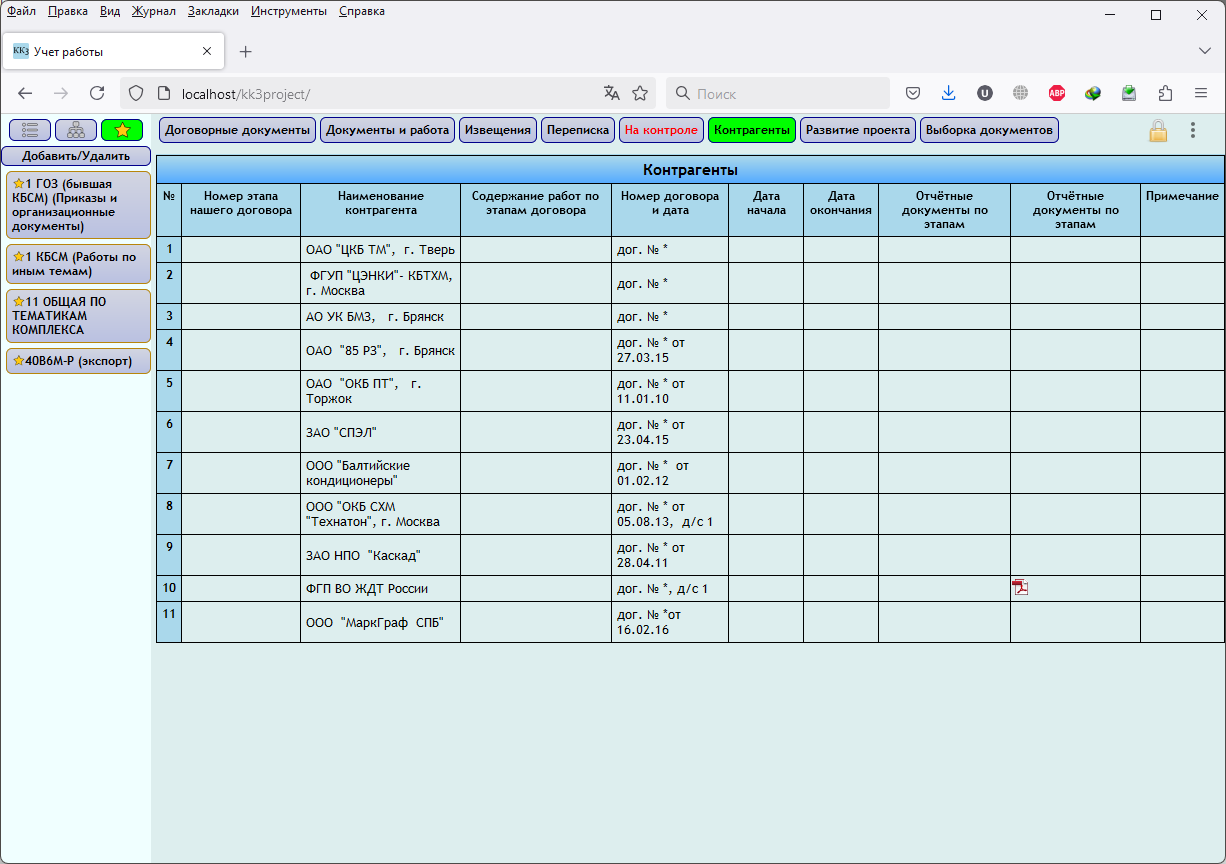


Рисунок - Контрагенты (рисунок из рабочей базы данных)

Описание:

- для редактирования страницы необходимо нажать на замок "закрыто" и открыть его "открыто";

- для добавления новых строк нажать на кнопку "http://localhost/kk3project/include/addline.png" под таблицей.

Окно развитие проекта

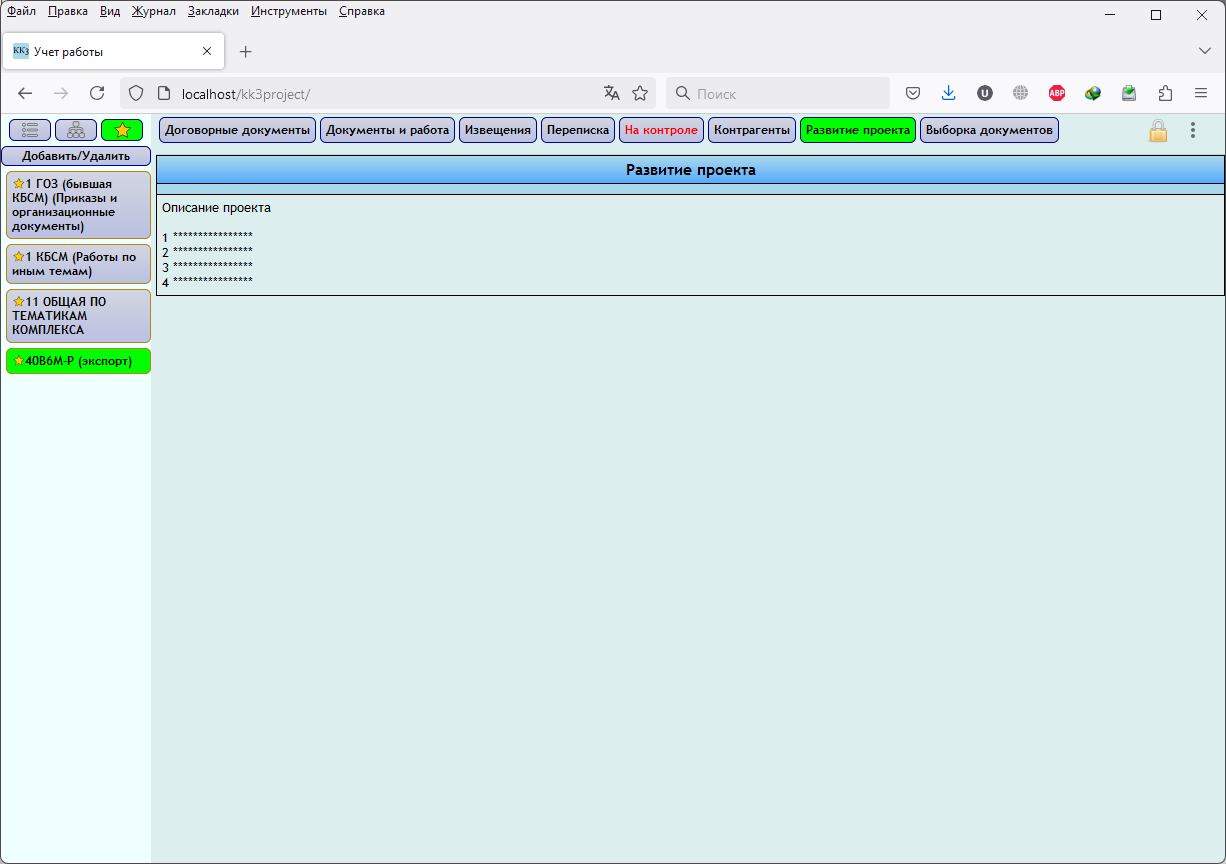


Рисунок - Развитие проекта

Описание:

- заполнятся ведущим специалистом для информирования сотрудников о проекте.

Окно выборка документов (документы и работа)

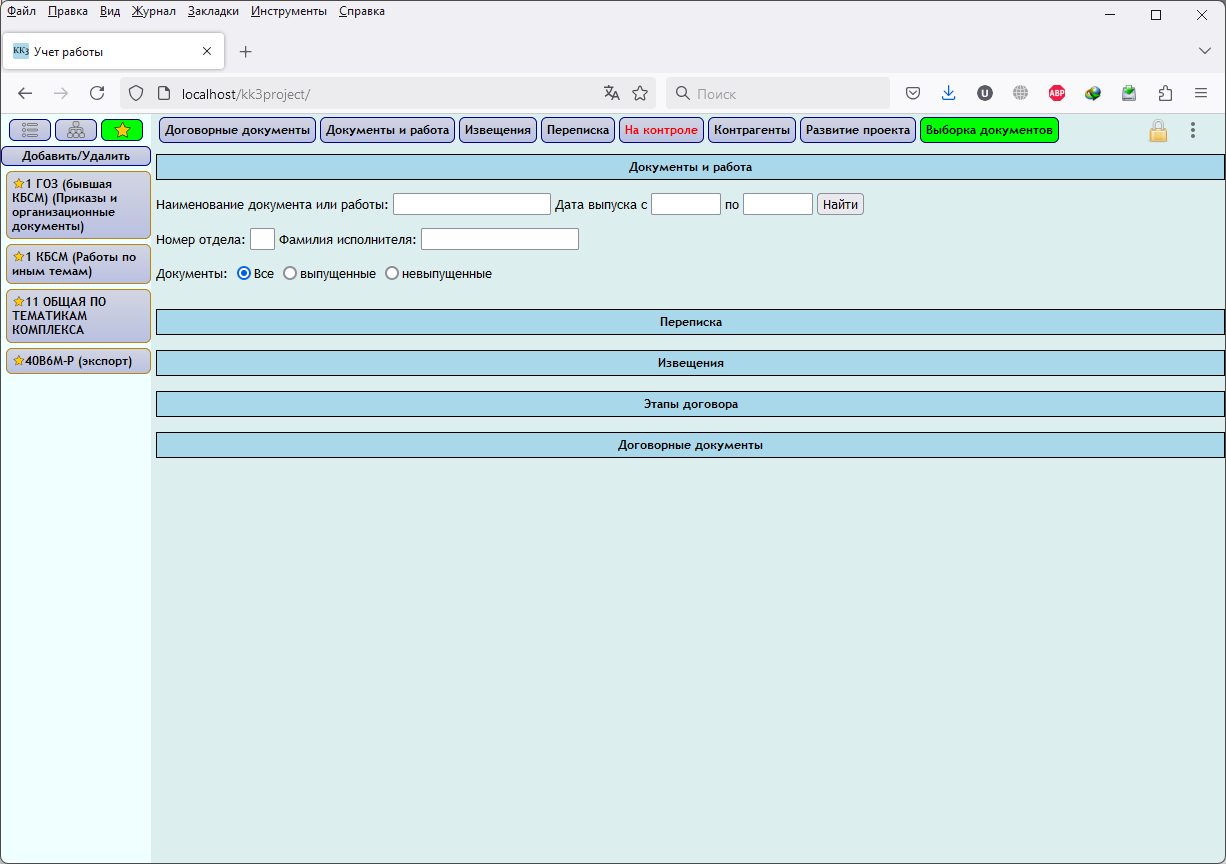


Рисунок - Поиск по таблице документы и работа

Описание:

- поиск осуществляется по всем полям таблицы;

- отдельно выделен фильтр по дате, номеру отдела и фамилии исполнителя.

Окно выборка документов (переписка)

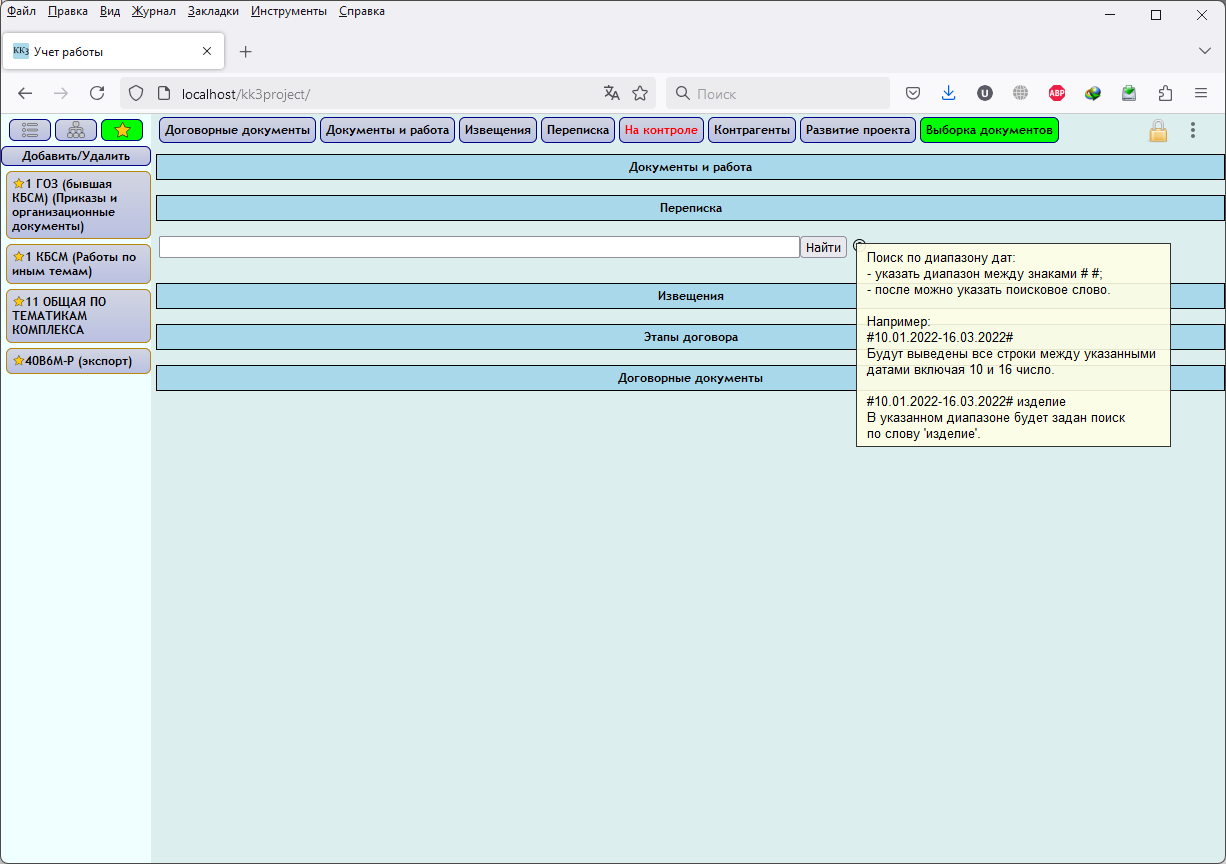


Рисунок - Поиск по таблице переписка

Описание:

- поиск осуществляется по всем полям таблицы;

- для фильтрации по дате использовать указать в поисковом запросе диапазон дат между знаками "#".

Окно выборка документов (извещения)

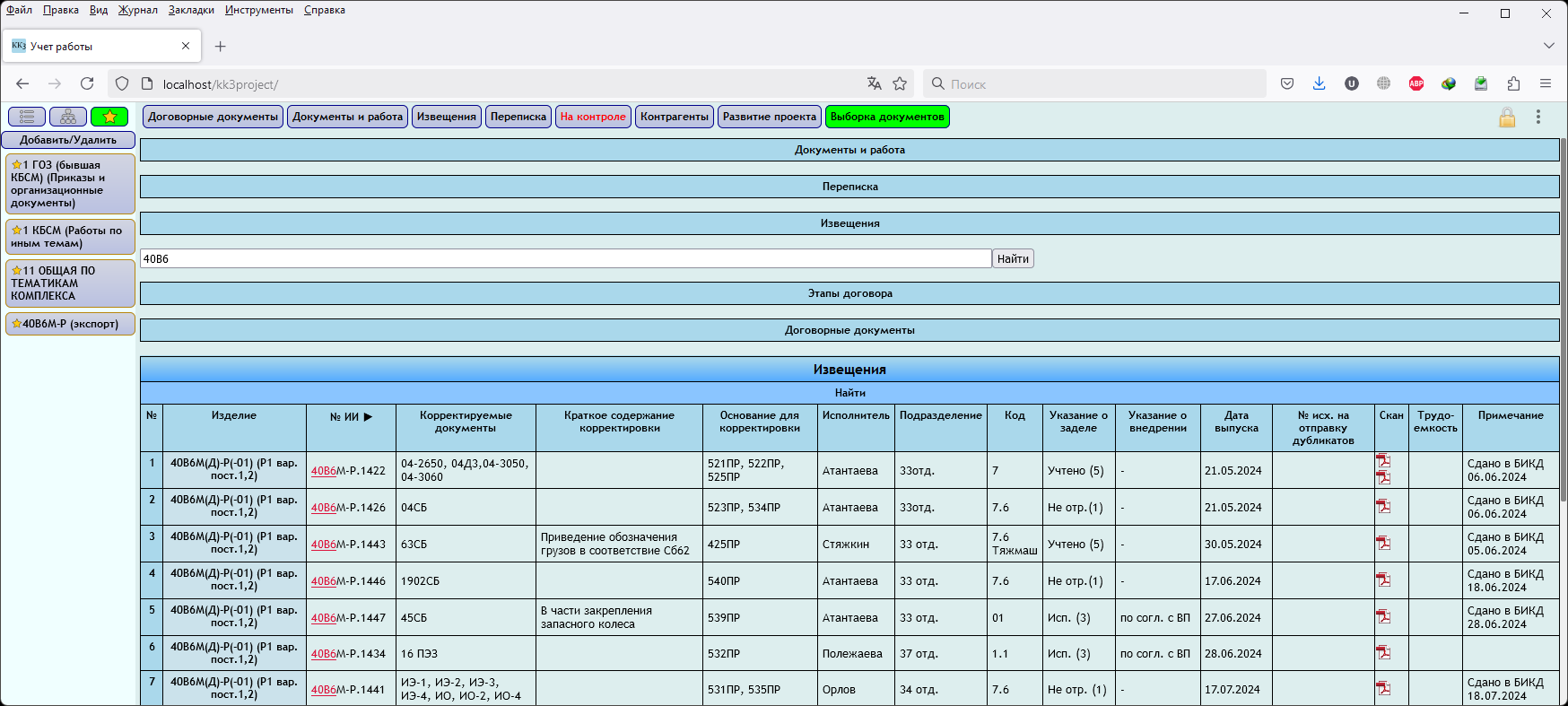


Рисунок - Поиск по таблице извещения (рисунок из рабочей базы данных)

Описание:

- поиск осуществляется по всем полям таблицы;

- найденное значение будет промаркировано красным цветом.

Окно выборка документов (этапы договора)

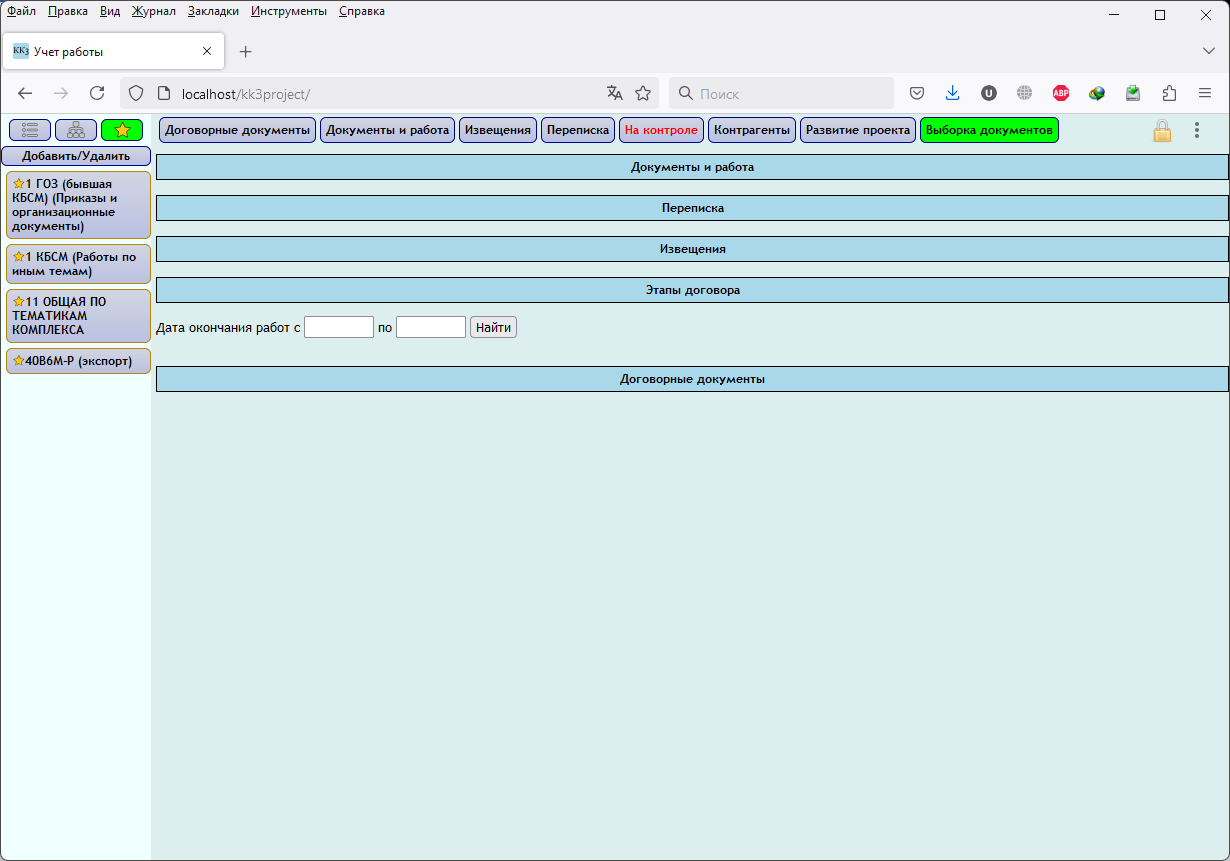


Рисунок - Поиск по этапам договора

Описание:

- поиск осуществляется только датам;

- выводятся этапы удовлетворяющие критерию поиска по всем договорам.

****Подход к разработке интерфейсов****

При разработке интерфейсов основное внимание уделялось следующим аспектам:

**- простота и интуитивность**:

- все модули приложения построены по принципу минимализма: только необходимые элементы, чтобы не перегружать интерфейс.

**- удобство навигации**;

- использованы простые меню, чтобы пользователь мог легко переключаться между модулями.

**- быстрая работа с данными**:

- реализованы функции поиска, фильтрации и сортировки, чтобы пользователи могли быстро находить нужную информацию.

****Разработка серверной части****

Серверная часть веб-приложения реализована на языке PHP версии 7.3 и взаимодействует с базой данных MySQL версии 5.5. Её основная задача – принимать запросы от клиентской части, обрабатывать их, взаимодействовать с базой данных и возвращать данные в формате JSON.

Принцип работы серверной части

1)  **Получение запроса:** клиентская часть отправляет HTTP-запросы методом POST к серверным скриптам на PHP.

2) **Обработка запроса:** серверный скрипт на PHP обрабатывает данные, переданные через POST-запрос, выполняет запросы к базе данных и обрабатывает результаты.

3) **Возврат данных:** после обработки сервер возвращает результат в формате JSON обратно в клиентскую часть для отображения или дальнейшей обработки.

****Работа с базой данных****

Взаимодействие с базой данных построено с использованием стандартного расширения MySQLi. Для обеспечения безопасности применяются подготовленные запросы (prepared statements), что предотвращает SQL-инъекции.

Все данные передаются клиенту в формате JSON. Этот формат удобен для обработки на фронтенде, так как он легко интегрируется в JavaScript-приложения.

****Преимущества архитектуры:****

**- простота реализации и расширения;**

- новые функциональные возможности можно легко добавить через создание дополнительных PHP-скриптов;

**- универсальность формата JSON;**

- удобная интеграция с клиентской частью и возможность обработки данных в любых приложениях;

- высокая надежность и безопасность;

- использование подготовленных запросов для предотвращения SQL-инъекций;

**- минимальная** нагрузка на сервер;

- простая архитектура обеспечивает высокую производительность даже при большом объёме запросов.

Выводы: серверная часть была разработана с учётом всех требований и обеспечивает стабильную работу системы. Она связана с клиентской частью через POST-запросы, взаимодействует с базой данных для обработки данных и возвращает результат в формате JSON. Разработанная архитектура позволяет легко дорабатывать функционал системы в будущем.

****Тестирование системы****

****Методика тестирования****

Для проверки качества разработанного веб-приложения была проведена серия тестов. Тестирование проводилось с целью выявления ошибок, подтверждения соответствия функционала требованиям и оценки устойчивости системы под нагрузкой. Основное внимание уделялось двум типам тестирования:

**1) Функциональное тестирование:**

-  проверка корректной работы всех функций веб-приложения:

-  проверка работы модуля "Договорные документы";

-  проверка работы модуля "Документы и работа";

-  проверка работы модуля "Извещения";

-  проверка работы модуля "Переписка";

-  проверка работы модуля "На контроле";

-  проверка работы модуля "Контрагенты";

-  проверка работы модуля "Развитие проекта";

-  проверка правильности работы поисковой системы;

-  правильность отображения интерфейсов на клиентской части.

**2) Нагрузочное тестирование**

-  проверка системы под нагрузкой, имитирующей одновременную работу нескольких пользователей;

-  моделировалась ситуация с одновременным доступом 10 пользователей;

-  изучалось время отклика сервера при большом числе запросов.

****Функциональное тестирование****

В рамках функционального тестирования проверялись основные модули системы. Результаты приведены в таблице:

|  |  |  |  |
| --- | --- | --- | --- |
| **Модуль** | **Тестируемая функция** | **Результат** | **Замечания** |
| Договорные документы | Заполнение таблиц | Пройдено | – |
|  | Загрузка файлов на сервер | Пройдено | – |
|  | Отображение загруженных файлов | Пройдено | – |
| Документы и работа | Заполнение таблиц | Пройдено | – |
|  | Загрузка файлов на сервер | Пройдено | – |
|  | Отображение загруженных файлов | Пройдено | – |
| Извещения | Заполнение таблиц | Пройдено | – |
|  | Загрузка файлов на сервер | Пройдено | – |
|  | Отображение загруженных файлов | Пройдено | – |
| Переписка | Заполнение таблиц | Пройдено | – |
|  | Загрузка файлов на сервер | Пройдено | – |
|  | Отображение загруженных файлов | Пройдено | – |
| На контроле | Заполнение таблиц | Пройдено | – |
|  | Загрузка файлов на сервер | Пройдено | – |
|  | Отображение загруженных файлов | Пройдено | – |
| Контрагенты | Заполнение таблиц | Пройдено | – |
|  | Загрузка файлов на сервер | Пройдено | – |
|  | Отображение загруженных файлов | Пройдено | – |
| Развитие проекта | Заполнение таблиц | Пройдено | – |
| Поисковая система | Верное отображение поисковых запросов | Пройдено | – |
| Работа с интерфейсами | Отображение всех элементов в Firefox | Пройдено | – |

****Нагрузочное тестирование****

Для проведения нагрузочного тестирования использовался инструмент Apache JMeter. Было выполнено моделирование сценария, в котором:

- 10 пользователей одновременно отправляют запросы к серверу.

- каждый пользователь выполняет по 100 запросов, общее количество запросов составило 1000.

- тип запросов: HTTP POST для выполнения основных операций (запись и чтение данных из базы).

- продолжительность теста: 1 минута 39 секунд.

**Для проведения тестирования был выбран следующий сценарий**:

- имитация нагрузки от одновременных пользователей.

- постепенное увеличение количества активных потоков для анализа стабильности системы.

**Цели теста**:

- определить среднее время отклика системы.

- проверить устойчивость системы при нагрузке.

- выявить потенциальные узкие места (например, длинные задержки в обработке запросов).

В результате тестирования были получены следующие показатели:

**- среднее время отклика**: **24 мс**, что свидетельствует о высокой скорости обработки запросов.

**- медианное время отклика**: **17 мс**, что подтверждает, что большинство запросов обрабатываются быстрее среднего времени.

**- отклонение**: **69 мс**, что указывает на наличие запросов с длительным временем отклика. Этот параметр требует анализа для выявления потенциальных проблем.

**- пропускная способность (Throughput)**: **604.625 запросов/минуту** (~10 запросов в секунду). Данный показатель соответствует нагрузке в 10 пользователей.

Результаты тестирования визуализированы на графике, представленном ниже:

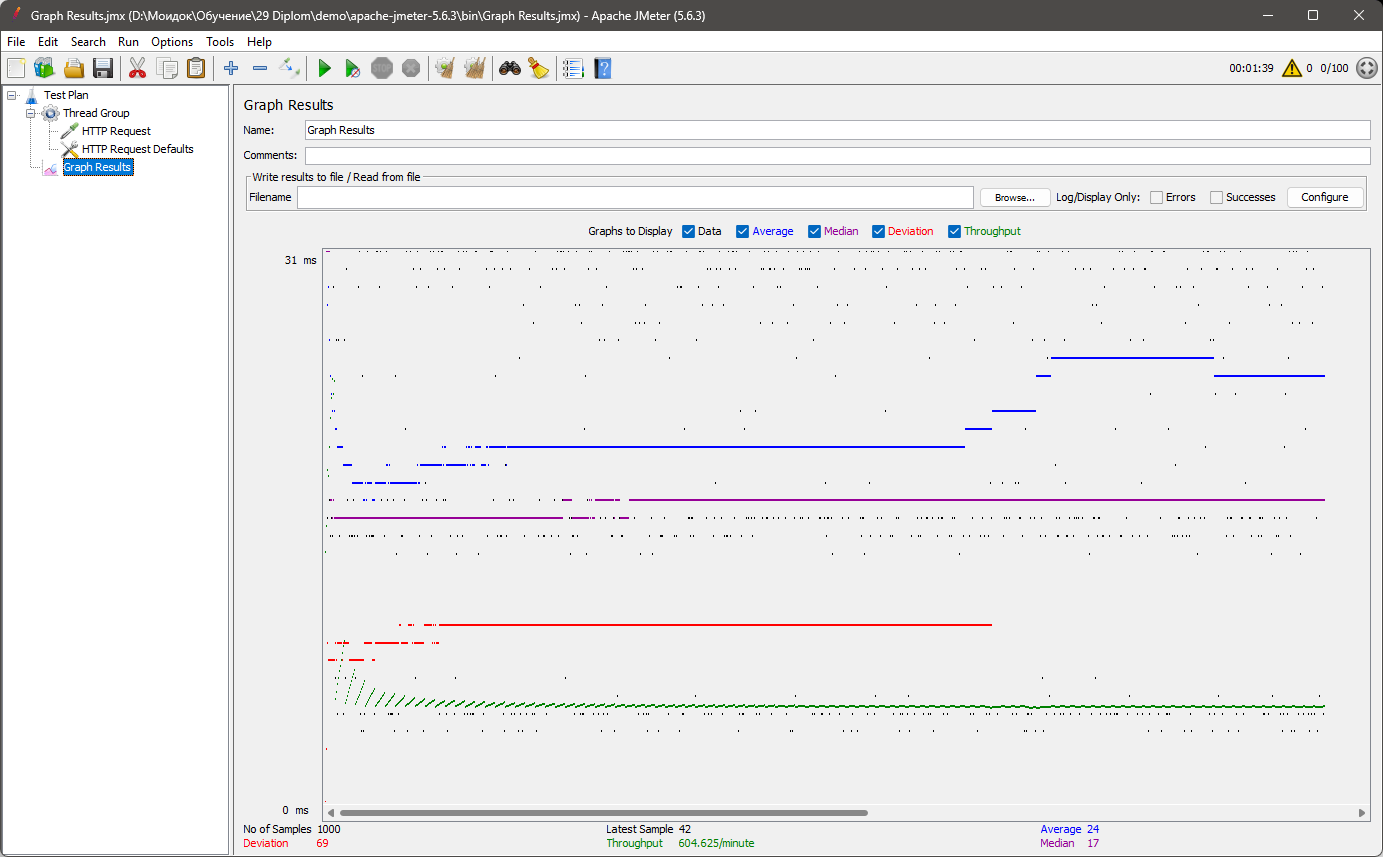


Рисунок - Результаты нагрузочного тестирования

На графике можно видеть:

- стабильность среднего времени отклика (синяя линия), что говорит об отсутствии деградации производительности при текущей нагрузке;

- пики отклонений (красная линия), которые могут быть связаны с более сложными запросами или нагрузкой на базу данных;

- устойчивый показатель пропускной способности (зелёная линия), что подтверждает способность системы обрабатывать запросы с заданной нагрузкой.

****Выводы по тестированию****

**Функциональное тестирование: в**се основные функции веб-приложения успешно прошли тестирование. Ошибок в работе модулей выявлено не было.

**Нагрузочное тестирование:**

- Система показала хорошую устойчивость при одновременной нагрузке от 10 пользователей;

- среднее время отклика значительно ниже критического значения (100 мс), что удовлетворяет требованиям к производительности.

- отклонения требуют анализа, но их влияние на общую производительность системы минимально.

- тест подтвердил готовность системы к работе в реальных условиях предприятия.

**Заключение:** Проведенное тестирование подтвердило, что разработанное веб-приложение соответствует функциональным требованиям, имеет стабильную работу и готово к внедрению на предприятии.

Заключение

В ходе выполнения дипломного проекта была разработана система для учета и контроля входящей и исходящей переписки, извещений, учета договоров и выпущенной документации, предназначенная для использования во внутренней сети предприятия. Данная система направлена на решение основных проблем документооборота, с которыми сталкивается предприятие: отсутствие структурированного и доступного архива переписки, сложность получения информации о текущих договорах и сроках выполнения работ, а также проблемы учета выпущенной документации.

В процессе работы была создана централизованная база данных, что позволило структурировать все документы, договоры, письма и извещения по конкретным изделиям. Клиентская часть системы разработана с использованием технологий, HTML, CSS, JavaScript и React, а серверная часть реализована на PHP с взаимодействием через JSON. Это обеспечивает удобный интерфейс для пользователей и надежное взаимодействие с базой данных MySQL.

Проведенное тестирование системы подтвердило ее работоспособность и готовность к использованию на предприятии. Среднее время отклика системы составляет 24 мс, что соответствует требованиям к производительности.

Перспективы развития системы:

**1) Внедрение оповещений о сроках** – для повышения эффективности работы пользователей системы целесообразно добавить функцию автоматических уведомлений о приближающихся сроках выполнения работ или окончании действия договоров. Это позволит избежать задержек и повысить общую дисциплину работы.

**2) Добавление системы чатов и сообщений – в** рамках дальнейшего развития системы можно реализовать функционал чатов для внутренней коммуникации сотрудников. Это улучшит координацию между подразделениями и ускорит обмен информацией по документам или договорам.

**3) Интеграция с почтовой системой предприятия – д**ля упрощения работы с перепиской возможно дополнение системы модулем интеграции с существующей почтовой системой предприятия. Это позволит автоматизировать процесс получения и отправки писем, а также упростит их регистрацию в базе данных.

Реализация указанных улучшений в будущем позволит сделать систему более функциональной, удобной и полезной для предприятия, обеспечивая полную цифровизацию документооборота и повышение эффективности рабочих процессов.

Список используемой литературы

1 Официальная документация PHP: <https://www.php.net/>

2 Официальная документация React: <https://react.dev/>

3 Metanit.com. Руководство по PHP и MySQL: <https://metanit.com/>

4 Официальная документация MySQL: <https://dev.mysql.com/doc/>

Приложения

Расположение практической части дипломной работы:

https://github.com/iDmDa/diplom