

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки / специальности)

«Бизнес-информатика»

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка облачного хранилища для корпоративной среды»

Обучающийся

И.С. Абрамов

(Инициалы Фамилия)

(личная подпись)

Руководитель

канд.пед.наук, доцент, Т.А. Агошкова

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2022

Аннотация

С. 67, рисунок 39, табл. 17, лит. 29 источников

ИНФОРМАЦИОННАЯ СИСТЕМА, ОБЛАЧНОЕ ХРАНИЛИЩЕ, БАЗА ДАННЫХ, IDEF0, IDEF1X, ER-МОДЕЛЬ, SQL, PYTHON, JAVASCRIPT, DJANGO, HTTP.

Разработана модель корпоративного облачного хранилища для ОАО «Концерн «АБС».

Дано описание ОАО «Концерн «АБС», выполнен технико-экономический анализ, определены участки рабочего процесса, требующие автоматизации. Построена модель «как есть». Сформулированы цель и задачи проектирования и требования к функциональности облачного хранилища. Произведен выбор и обоснование методов проектирования облачного хранилища, а также проектных решений СУБД и клиента облачного хранилища.

Построена функциональная модель «как должно быть», описано программное и технологическое обеспечение. Разработана физическая модель данных и спроектирован интерфейс клиента облачного хранилища. Дано описание функциональности программного продукта.

Оценена экономическая эффективность реализации проекта.

Работа находится на стадии приёмо-сдаточных испытаний.

Оглавление

Введение.....	4
Глава 1 Функциональное моделирование предметной области.....	7
1.2 Техничко-экономическая характеристика предметной области .	7
1.3 Концептуальное моделирование предметной области	11
1.4 Анализ существующих разработок на предмет соответствия сформулированным требованиям	16
1.5 Разработка модели бизнес-процесса «Как должно быть»	21
Глава 2 Логическое проектирование облачного хранилища	24
2.1 Выбор технологии логического моделирования облачного хранилища.....	24
2.2 Информационное обеспечение облачного хранилища	26
2.3 Проектирование базы данных облачного хранилища.....	29
2.4 Требования к аппаратно-программному обеспечению облачного хранилища.....	32
Глава 3 Физическое проектирование облачного хранилища	34
3.1 Выбор архитектуры облачного хранилища.....	34
3.2 Выбор технологии разработки программного обеспечения облачного хранилища	35
3.3 Выбор СУБД облачного хранилища	39
3.4 Разработка физической модели данных облачного хранилища ..	39
3.5 Разработка программного обеспечения облачного хранилища...	42
3.6 Оценка и обоснование экономической эффективности разработки облачного хранилища	57
Заключение	64
Список используемой литературы	65

Введение

В современном мире существует масса способов хранения информации: начиная от аналоговых носителей и заканчивая современными цифровыми решениями. Одним из стремительно набирающих популярность вариантов является облачное хранилище [29].

Несомненными преимуществами облачных решений являются возможность доступа пользователей из любой точки мира, где есть доступ к Интернет; независимость от специфики деятельности пользователей и компаний; возможность простого увеличения используемых мощностей для хранения данных при первой потребности.

В условиях глобальной цифровизации общества практически невозможно представить себе бизнес, который может оставаться успешным и прибыльным и при этом абсолютно изолированным от каких-либо ИТ-решений.

Информация приобретает невероятную ценность, является несомненным конкурентным преимуществом. Количество производимой и потребляемой информации растет огромными темпами и в какой-то момент для каждой организации становится актуальным вопрос о надежном решении по её хранению и использованию.

При этом важным условием является минимизация финансовых издержек и операционных рисков.

Эффективным способом по снижению стоимости обработки данных является переход к использованию облачных вычислений: бизнес становится динамичнее, повышается качество работы пользователей, появляется экономия за счет оптимизации поддержки решения, гарантируется соответствующий уровень безопасности и производительности.

Использование облачных технологий позволяет ощутимо упростить ежедневную работу потребителей информации: появляется возможность

беспрепятственного обмена информацией в любых форматах с разных устройств из любой локации.

При этом всегда остается возможным сохранить файл на локальное устройство для индивидуальной работы.

Объектом исследования является ОАО «Концерн «АБС».

Предмет исследования – процесс проектирования и разработки облачного хранилища.

Целью данной работы является разработка облачного хранилища для корпоративной среды.

Ключевыми задачами по достижению поставленной цели являются:

- анализ предметной области, а именно: характеристика предприятия, его организационной структуры, существующих бизнес-процессов;

- анализ существующих систем для решения поставленной задачи;
- обоснование необходимости разработки;
- выбор стратегии и способа автоматизации;
- выбор информационного, программного и технического обеспечения;

- логическое и физическое проектирование облачного хранилища;

- разработка программного обеспечения;

- описание функциональности облачного хранилища;

- расчет экономической эффективности проекта.

При выполнении работы были использованы такие методы исследования, как аналитический, сравнительный, методы расчета экономической эффективности.

В первой главе дана характеристика компании, а также проведен анализ деятельности предприятия при подходе «Как есть». Выбрана технология концептуального моделирования. Проведен анализ существующих разработок, выделены критерии для сравнения. Построена модель «Как будет».

Во второй главе проведено логическое проектирование облачного хранилища. Выбрана технология проектирования. Разработана концептуальная и логическая модель. Разработано информационное обеспечение: классификаторы, системы кодирования, входная и выходная информация. Определены требования к аппаратному и программному обеспечению.

В третьей главе осуществлено физическое проектирование облачного хранилища. Выбрана архитектура и технология разработки. Выбрана конкретная СУБД и построена физическая модель. Разработано программное обеспечение: программные модули и их взаимосвязь, описана функциональность системы. Также в третьей главе была проведена экономическая оценка эффективности проекта, получены экономические показатели проекта.

Глава 1 Функциональное моделирование предметной области

1.1 Техничко-экономическая характеристика предметной области

ОАО «Концерн «АБС» специализируется на разработке и внедрении автоматизированных систем управления технологическими процессами, изготовлении широкого спектра средств автоматизации для систем промышленной автоматизики.

Основными направлениями деятельности предприятия являются:

- создание полностью готовых к эксплуатации систем информатизации и телекоммуникаций;
- проектирование, разработка и поставка информационных распределенных автоматизированных систем управления (АСУ) общего и специального назначения с необходимым классом аппаратно-программной защиты информации;
- разработка общего, общесистемного и прикладного программного обеспечения средств АСУ;
- разработка программных и программно-аппаратных средств защиты информации, в том числе криптографической защиты;
- проектирование, разработка и поставка телекоммуникационного оборудования для обмена открытой и закрытой информацией, универсальных модулей для сопряжения с широким спектром оборудования;
- серийное изготовление, техническая поддержка, абонентское, сервисное и гарантийное обслуживание АСУ;
- проектирование, разработка и поставка, строительство и монтаж локально-вычислительных сетей (ЛВС) и связи на основе волоконно-оптических линий, в том числе с мониторингом оптической сети и защитой от несанкционированного доступа (НСД);
- аттестация объектов информатизации по требованиям безопасности информации, включая специсследования и спецпроверки.

Оргструктура ОАО «Концерн «АБС» соответствует направлениям деятельности предприятия: проведение научных исследований и изысканий, включая создание различных прототипов и опытных образцов; проектирование и разработка информационно-управляющих систем, включая создание проектной и эксплуатационной документации; гарантийное и сервисное обслуживание; авторский надзор и аудит. На рисунке 1 показана структура ОАО «Концерн «АБС»

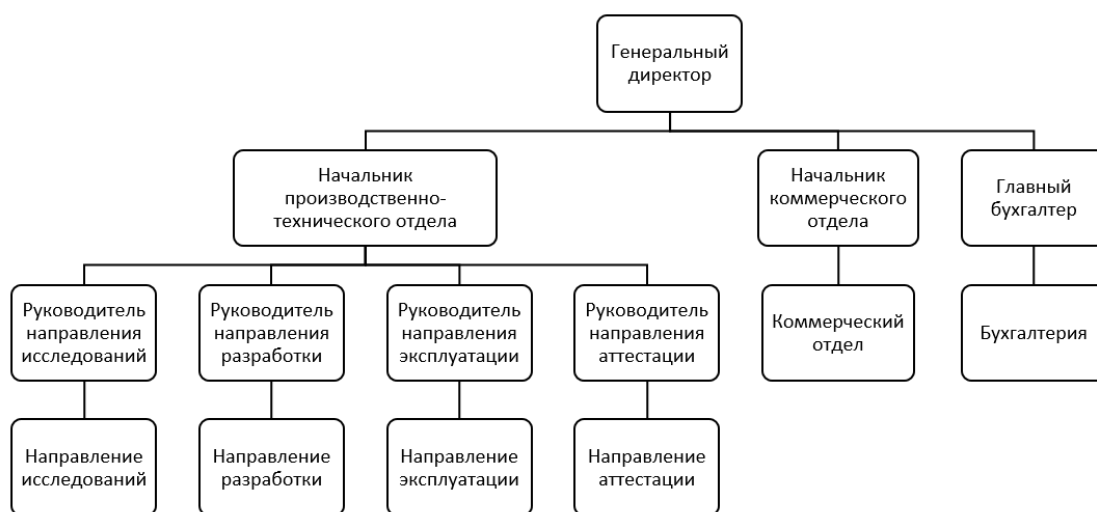


Рисунок 1 – Структура ОАО «Концерн «АБС»

В числе сотрудников ОАО «Концерн «АБС» два академика, один член-корреспондент РАН, 19 докторов наук, 48 кандидатов физико-математических, технических и военных наук. Из них 16 профессоров, 9 доцентов, 23 старших научных сотрудника. В настоящее время на предприятии ведутся свыше 30 научно-исследовательских и опытно-конструкторских работ (НИОКР) в области информационных технологий.

На предприятии действуют отдел технического контроля, нормоконтроль, подразделения материально-технического обеспечения, сопровождения и сборочно-монтажные участки, которые оснащены современным стендовым оборудованием, инструментом и оснасткой, налажено серийное производство. Имеется архив конструкторской

документации и программных изделий, соответствующий нормативным требованиям.

Предприятие имеет библиотеку научно-технической документации, насчитывающую более 1500 наименований, действующих российских и международных стандартов и другой нормативной документации.

На рисунке 2 приведена техническая архитектура информационной системы предприятия.

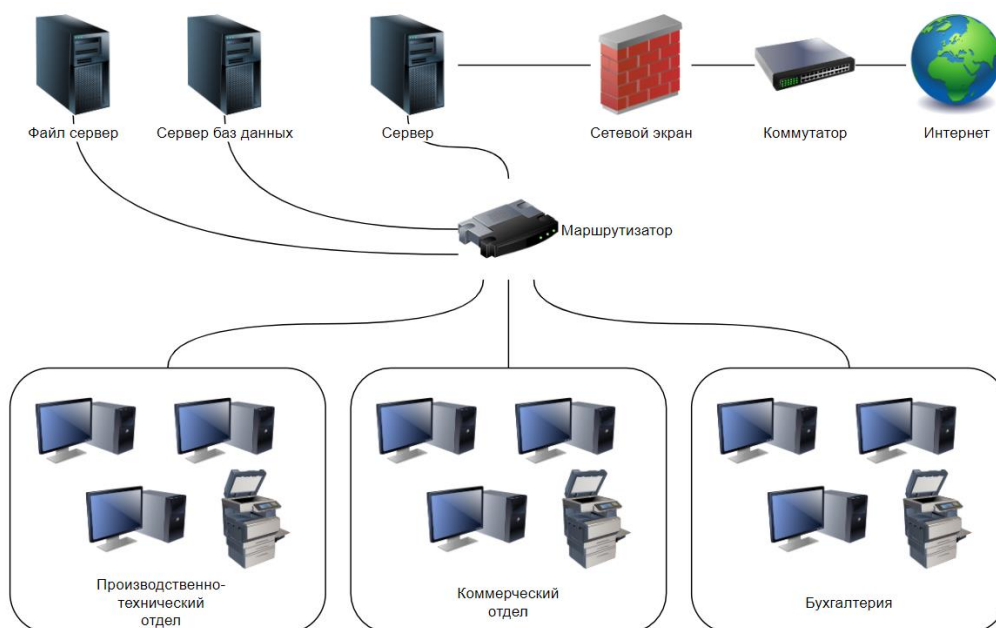


Рисунок 2 – Техническая архитектура

У каждого сотрудника есть оборудованное рабочее место, включающее в себя персональный компьютер, телефон. В рабочих компьютерах используются операционные системы серии Microsoft Windows или Linux. Программное обеспечение включает в себя стандартный офисный набор, стандартные средства Windows, а также специализированное ПО, необходимое для работы сотрудника.

Для сотрудников заводятся индивидуальные учетные записи Windows, которые активируются в первый рабочий день и блокируются в день увольнения сотрудника. Установка программного обеспечения возможна из

стандартного набора ПО, не требующая согласования с руководством. Также есть возможность оставления заявки на необходимые инструменты для работы. Для каждого сотрудника заводится внутренний номер телефона и внутренняя электронная почта.

На текущий день потребность в модернизации программно-аппаратного обеспечения рабочих мест сотрудников не выявлена.

Корпоративная информация хранится на локальном сервере. Сервера: главный сервер домена, сервер баз данных, файл сервер - сконструированы по одной конфигурации с применением многоцелевых серверов Dell PowerEdge T410. В таблице 1 приводятся все основные технические характеристики сервера.

Таблица 1 – Технические характеристики аппаратного обеспечения

Параметры	Значения
Название	Dell PowerEdge T410
Конструктивное исполнение	Tower
Тип установленного процессора	Intel® Xeon™ E5502
Тактовая частота процессора, ГГц	1.86
Установленных процессоров	2
Поддерживаемых процессоров	2
Набор микросхем	Intel® E5500
Тип установленной оперативной памяти	DDR3 1066Мгц
Объем установленной оперативной памяти, Мб	4096
Поддержка ЕСС оперативной памятью	Есть
Макс. объем оперативной памяти, Гб	64.0
Графическая плата	Matrox G200eW
Объем видеопамяти графической платы, Мб	8
RAID-контроллер	- интегрированный двухканальный SATA (RAID 0, 1) - поддержка SAS (опционально)
Жесткие диски	- SATA от 160 Гб - SAS от 146 Гб
Тип оптического привода	DVD-ROM (опции CD-RW, DVD-RW)
Сетевой контроллер	встроенный двухпортовый Gigabit Ethernet

Dell Power Edge T410 – это высокоэффективный двухпроцессорный сервер в корпусе Tower, предназначенный для решения задач различной направленности. Для минимизации времени простоя вычислительной сети даже стандартные модели этой серии поставляются с функциями «горячего резервирования» жестких дисков.

1.2 Концептуальное моделирование предметной области

1.2.1 Моделирование бизнес-процессов предметной области для постановки задачи автоматизированного варианта решения

«Концептуальная модель предметной области должна быть создана без сосредоточения внимания на конкретных системах управления БД и моделях данных» [3]. Основное предназначение – построение семантической модели бизнес-процессов на высшем уровне абстракции.

Моделирование подразумевает визуальное проектирование с помощью компьютерных программ, которые заменяют разработчику карандаш и бумагу. Специализированные программы способны не только создавать, но и анализировать построенные схемы (модели).

Существует множество методик и программных продуктов, позволяющих моделировать бизнес-процессы согласно требованиям конкретного предприятия [6], [7]. В основном, «интегрированные системы содержат средства анализа и моделирования деятельности предприятия, позволяющие не только строить бизнес-процессы, но и осуществлять стратегическое и финансовое управление» [7].

Основные моменты, которые должны быть учтены при построении модели:

- сохранение всей необходимой информации;
- получение данных по всем запросам;
- исключение дублирования и избыточности данных;
- целостность базы данных;

– многопользовательский доступ с различных устройств.

На этапе «построения логической модели программного продукта использовались методологии объектно-ориентированного анализа и проектирования, использующие нотацию языка UML» [15]. На рисунке 3 показана диаграмма вариантов использования (use case diagram) [14] облачного хранилища.

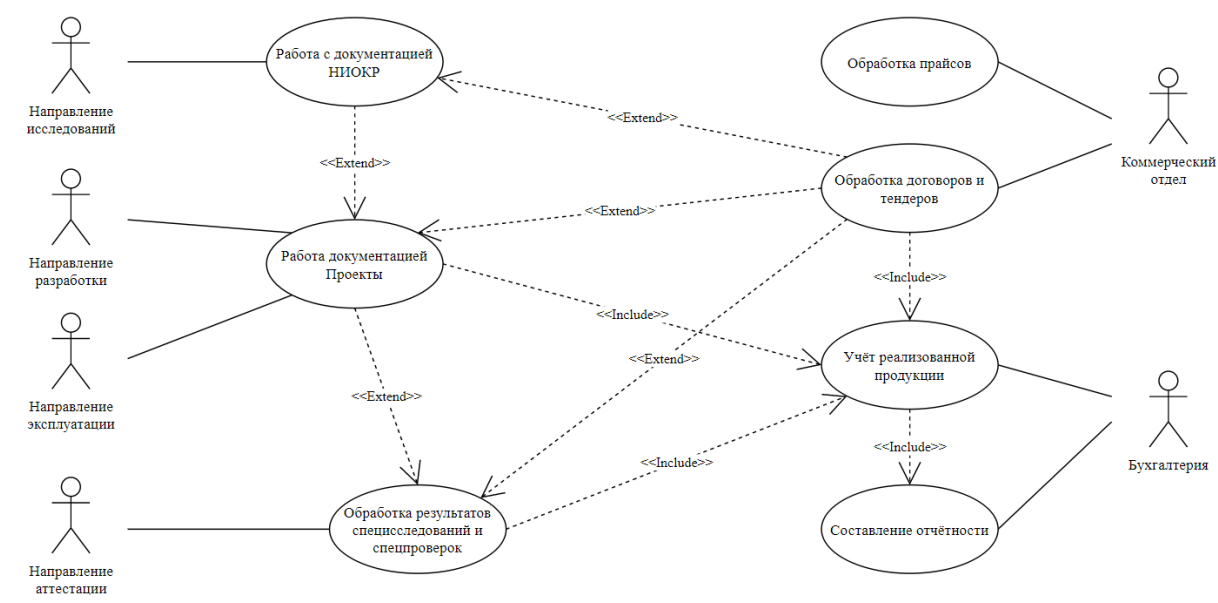


Рисунок 3 – Диаграмма вариантов использования облачного хранилища

Процесс бизнес-моделирования заключается в определении и описании существующих бизнес-процессов предприятия, а также проектировании новых. Моделирование бизнес-процессов – полезный инструмент, позволяющий анализировать всю деятельность компании внутри, то есть определять, как организована работа в каждом процессе, а также наблюдать за взаимодействием фирмы со внешней средой (клиентами, конкурентами, партнерами, государством).

Ввиду специфики предметной области как таковых информационных потоков не выделяется, все действия пользователей представляют собой

однотипную работу с файлами различных типов: создание, удаление, редактирование, настройка уровня доступа и т.д.

1.2.2 Разработка и анализ модели бизнес-процесса «Как есть»

«Для выполнения структурно-функционального анализа процесса работы предприятия было выполнено проектирование на основе методологии IDEF0» [19]. На рисунке 4 показана контекстная диаграмма деятельности ОАО «Концерн «АБС», а именно работы с корпоративной информацией.

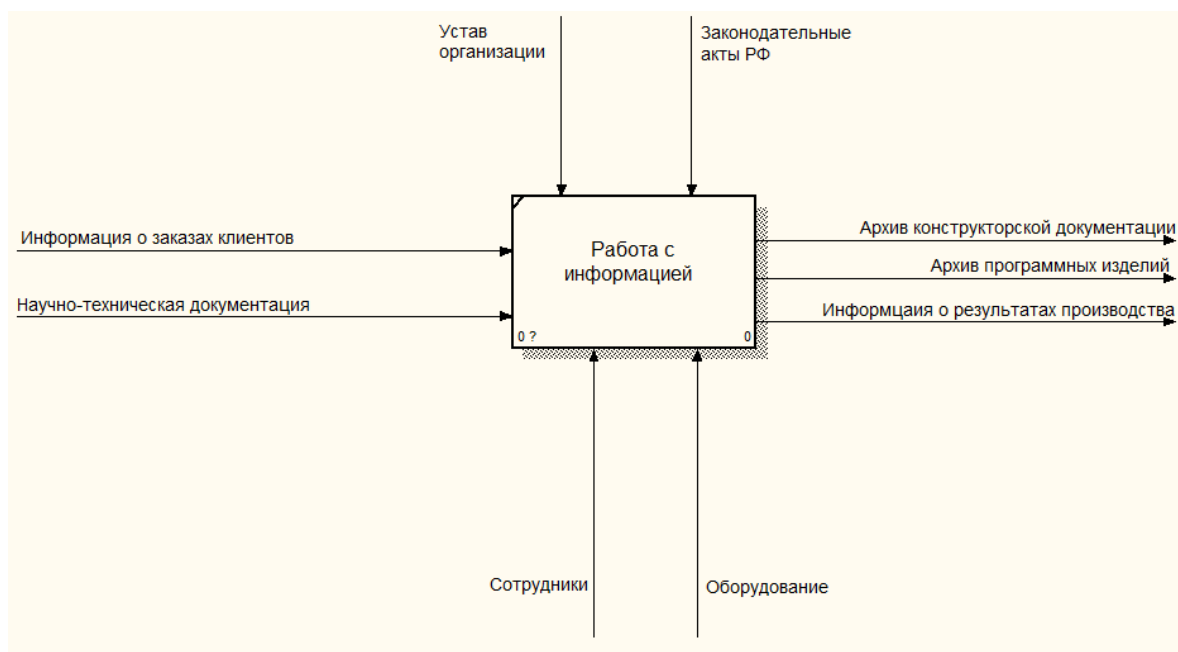


Рисунок 4 – Модель работы с организацией «Как есть»

Корпоративной информацией является информация о заказах клиентов, научно-техническая документация, информация о результатах производства и т.д. На вход поступает информация о заказах клиентов и научно-техническая документация. Бизнес-процессы регулируются Уставом организации и законодательными актами. Для обработки информации используются трудовые ресурсы - сотрудники, а также оборудование. К оборудованию относятся сервера, где хранится информация, также ПК, печатающие устройства и все, что используется для сбора, печати, передачи информации.

Основные подпроцессы работы с информацией:

- сбор информации;
- сортировка информации;
- хранение информации;
- передача информации.

На рисунке 5 приведена декомпозиция модели.

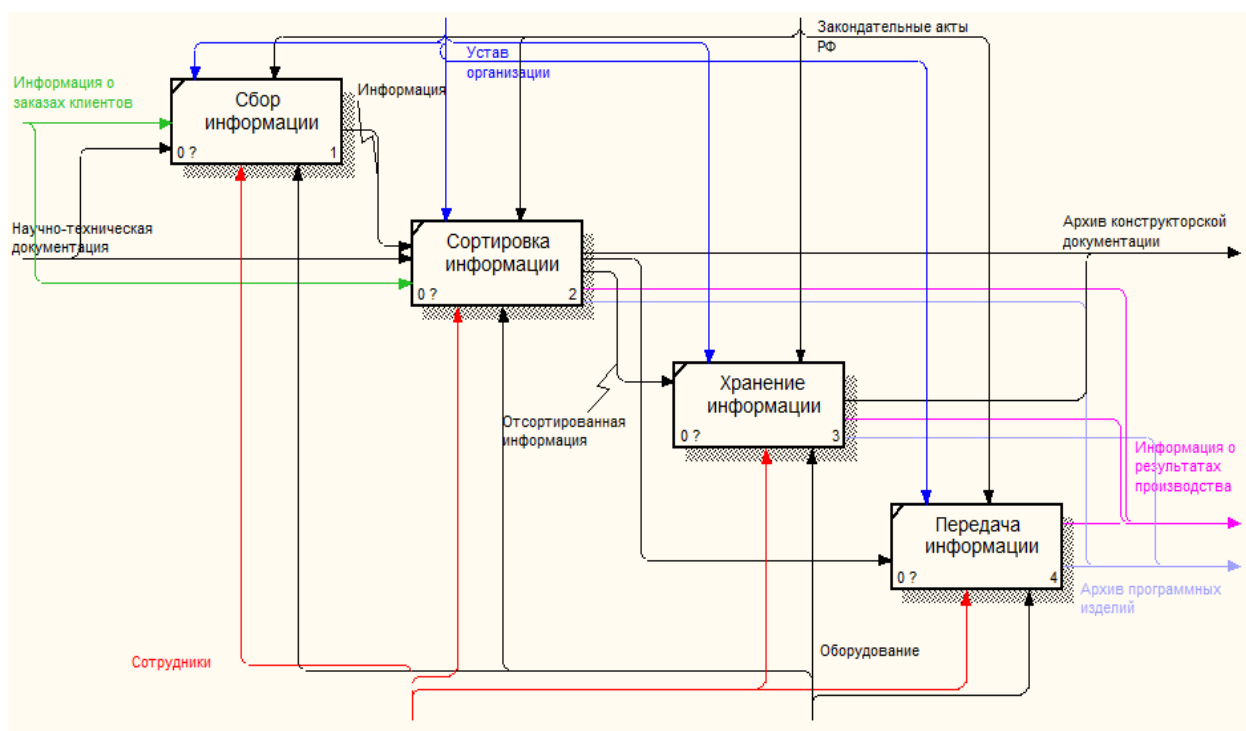


Рисунок 5 – Декомпозиция концептуальной модели предметной области

Все эти процессы на данный момент осуществляются разрозненно. Сбор и передача информации осуществляются сотрудниками. Хранение информации осуществляется с помощью серверов.

1.2.3 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии

В таблице 2 приведены затраты времени на выполнение процессов до и после внедрения облачного хранилища.

Таблица 2 – Показатели затрат времени на задачу до и после автоматизации

Аспект деятельности предприятия	Временные затраты	
	До автоматизации	После автоматизации
Сбор информации	20 мин	5 мин.
Сортировка информации	20 мин.	5 мин.
Хранение информации	10 мин	0
Передача информации	20 мин.	0
Итого затраты времени	70 мин.	10 мин.

Для работы с информацией можно внедрить другие информационные системы, например системы электронного документооборота. Но проблема заключается не только в затратах времени на обработку информации, но в и том, что информация занимает терабайты памяти.

Как было сказано выше, информация на данный момент хранится на локальном сервере. Данное «решение обладает рядом несомненных преимуществ:

- быстрый доступ к данным;
- конфиденциальность» [9] (в том числе ограничение физического доступа к оборудованию);
- отсутствие ограничений по выбору конфигурации оборудования и программного обеспечения.

Главный недостаток локального сервера - высокие затраты:

- приобретение оборудования;
- установка и настройка;
- лицензионное ПО;
- расходы на обслуживание и сопровождение.

Также необходимо обеспечить правильное определение потребностей бизнеса в хранении данных, основываясь на которых подбирается техника.

Достоинства облачного хранилища:

- снижение затрат за счет отказа от приобретения и поддержки собственного оборудования;

- готовое решение, которое можно легко масштабировать при возрастании потребностей бизнеса;
- многопользовательский удаленный доступ с различных устройств при наличии подключения к сети Интернет.

Таким образом, облачные решения являются обоснованным выбором для организаций любого масштаба и рода деятельности.

1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям

Для анализа были поставлены критерии выборки:

- Свободно распространяемое ПО или коммерческая лицензия;
- поддержка кириллицы;
- отслеживание версионности документов;
- возможность расширенного поиска по тэгам;
- многопользовательский доступ из любой точки;
- поддержка метаданных для документов.

Сейчас на рынке представлены как зарубежные облачные хранилища, так и отечественные, дающие разный набор функций подходящие под текущие требования.

«Google Drive» – это «сервис хранения, редактирования и синхронизации файлов разработанный компанией Google» [26]. Это условно бесплатное облачное хранилище с множеством функций, таким как многопользовательский доступ, многопользовательская работа с файлами. Данный сервис позволяет работать как через веб-интерфейс, так и через специальный клиент, устанавливаемый на различные устройства будь то рабочая станция или планшет, или телефон, не зависит от операционной системы. Данный сервис очень сильно завязан на другие сервисы от компании Google. Бесплатная версия имеет ряд ограничений таких как: размер пространства 15 гигабайт, ограничения на предоставление доступа целевой

группе. Общие пространства для групп. На рисунке 6 изображен интерфейс ПО.

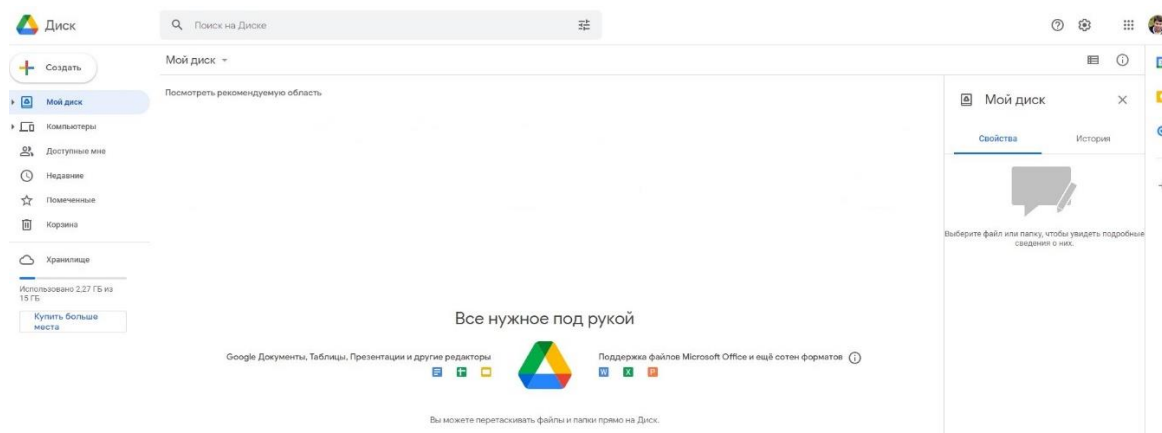


Рисунок 6 – Пользовательский интерфейс сервиса «Google Drive»

Плюсы:

- доступ к файлам не зависимо от устройства;
- легкая загрузка файлов как путем перетаскивания, так и синхронизация через, клиент программы;
- история файла хранится, на протяжении всей жизни файла;
- предоставления доступа к файлам для целевой группы;
- общие диски для группы, для работы с общими файлами.

Минусы:

- малый объем диска при бесплатном использовании;
- большинство функций не доступно при бесплатном использовании;
- поиск файлов по метаданным, не поддерживает все форматы файлов.

«GitFlic»– отечественный «веб-сервис для разработки проектов основан на системе контроля GIT» [17]. Бесплатная версия имеет ограничения на приватный репозиторий, с ним могут работать только 5 человек, в платной

версии данное ограничение снимается. На рисунке 7 изображен интерфейс ПО.

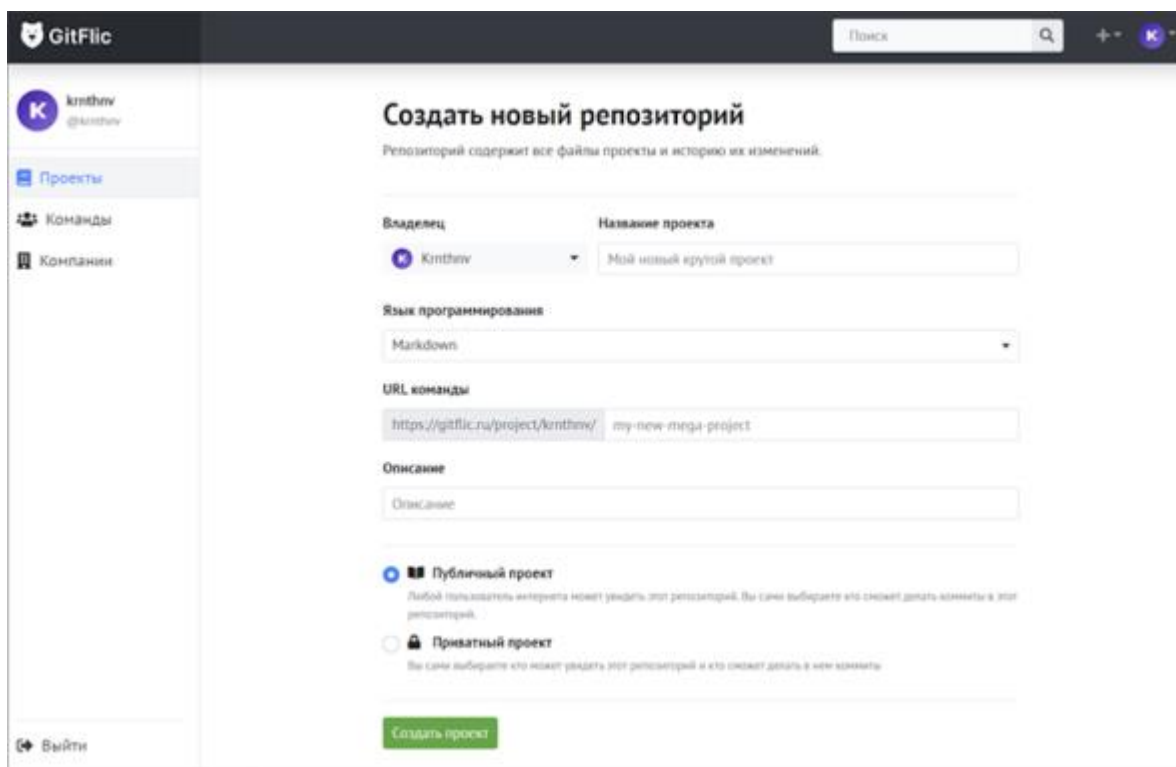


Рисунок 7 – Пользовательский интерфейс веб-сервиса «GitFlic»

Плюсы:

- владельцу доступна вся история изменений;
- возможность вернуться к любой версии документа;
- многопользовательская работа с файлами проекта.

Минусы:

- мало инструментов для совместной работы;
- сервис имеет узкую направленность и имеет ряд ограничений.

«LanDocs» – «платформа для управления корпоративным контентом.

Данная система включает в себя модуль электронный архив, данный модуль позволяет оперативное хранение, долговременное хранение, индексирование» [20]. На рисунке 8 отображено окно поиска документов в «LanDocs».

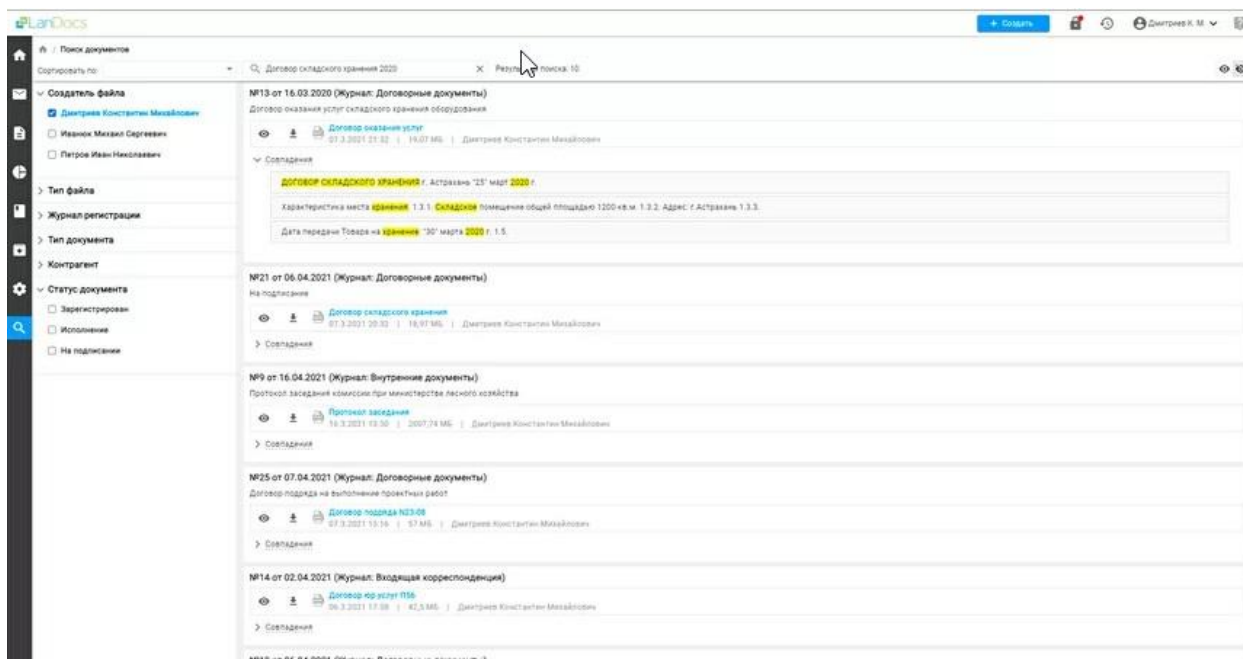


Рисунок 8 – Пример окна поиска в системе «LanDocs»

Для работы с системой можно использовать любой удобный способ, клиент на рабочем месте или веб клиент с любого устройства. Разницы работы в различных клиентах нет. В системе есть множество модулей для работы с файлами: модули сравнения, модули распознавания текста, модуль полнотекстового поиска, модуль конструктора документов.

Плюсы платформы:

- полноценная история работы с документами;
- восстановление прошлых версий документов;
- разграничение прав доступа к документам;
- поиск по фразам в документе.

Минусы платформы:

- высокие требования к внедрению и сопровождению платформы;
- большая стоимость лицензий и поддержки платформы.

Текущий анализ существующих решений в данной области показывает, что существующие решения имеют ряд недостатков для решения поставленных задач, из-за данных недостатков есть потребность разработать

облачное хранилище, которое будет отвечать заданным требованиям. Анализ существующих сервисов приведен в таблице 3.

Таблица 3 – Анализ существующих сервисов

Программное обеспечение	Сервис «Google Drive»	Веб-сервис «GitFlic»	Платформа «LanDocs»
Поддержка кириллицы	+	-	+
Возможность расширенного поиска по тэгам	-	-	+
Возможность восстановить прошлую версию файла	+	+	+
Коммерческая лицензия	-/+	-/+	+
Многопользовательский доступ к файлам	+	+	+
Простота использования	+	+	-

Облачное хранилище «Google Drive» имеет различные способы для подключения к хранилищу для работы с файлами. Но большинство функций таких как, многопользовательский доступ, общее пространство, не доступно в бесплатной версии.

Сервис «GitFlic» несмотря на удовлетворение некоторых требований задачи, имеет узкую направленность для разработчиков, и не предназначен для работы с различными файлами. На данном этапе большинство заявленных функций еще не включены.

Платформа «LanDocs» имеет большинство инструментов для работы с файлами, включая хранение, распознавание, поиск по внутреннему содержанию файлов. Имеет большое количество дополнительных модулей. Данная платформа имеет высокую стоимость, сложно во внедрение и сопровождении. Данный анализ показывает, что разработка и внедрение облачного хранилища будет лучшим результатом для решения поставленных задач.

Анализ существующих сервисов с разрабатываемым облачным хранилищем показан в таблице 4.

Таблица 4 – Анализ существующих сервисов с разрабатываемым облачным хранилищем

Программное обеспечение	Сервис «Google Drive»	Веб-сервис «GitFlic»	Платформа «LanDocs»	Облачное хранилище собственной разработки
Поддержка кириллицы	+	-	+	+
Возможность расширенного поиска по тэгам	-	-	+	+
Возможность восстановить прошлую версию файла	+	+	+	+
Коммерческая лицензия	-/+	-/+	+	-
Многопользовательский доступ к файлам	+	+	+	+
Простота развертывания	+	+	-	+

Разрабатываемое облачное хранилище является экономически выгодным, поддерживает различные платформы, обеспечивает работу с метаданными документов и имеет гибкий поиск по ним, многопользовательский доступ, обладает повышенной производительностью.

1.4 Разработка модели бизнес-процесса «Как должно быть»

Цель – разработать облачное хранилище для корпоративной среды. Пользователями системы будут сотрудники компании. Основные функции, которые должна выполнять данная система:

- хранение информации;
- возможность загрузки, сортировки, удаления информации;
- получение аналитической и статистической информации;
- получение справочной информации в печатном виде;
- резервное копирование данных.

Задачи, решаемые автоматизацией:

- повышение оперативности обмена информацией внутри организации;

- уменьшение времени процессов обработки и преобразования информации за счет сокращения количества выполняемых рутинных операций по созданию документации;
- расширение возможностей эффективного использования имеющихся информационных ресурсов;
- повышение уровня информационной безопасности.

На рисунке 9 представлена целевая модель бизнес-процессов, согласно которой корпоративная информация хранится не на локальном сервере, а в облачном хранилище.

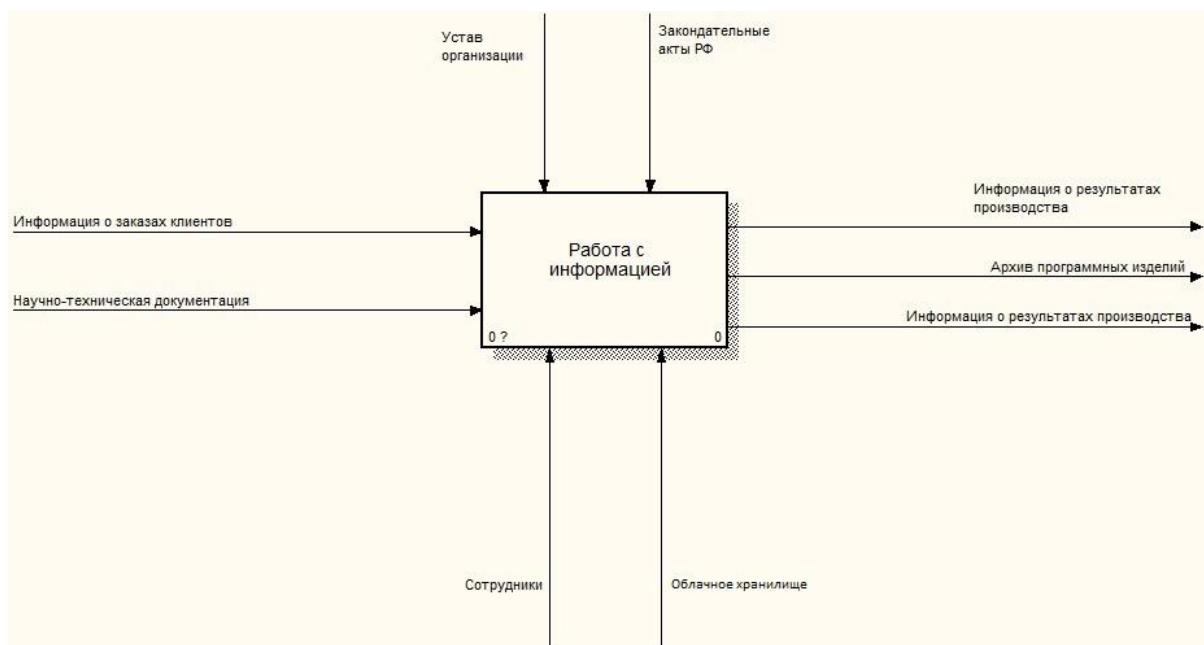


Рисунок 9 – Модель «Как будет»

Модель включает следующие блоки:

- сбор информации;
- сортировка информации;
- хранение информации.

На рисунке 10 показана декомпозиция модели.

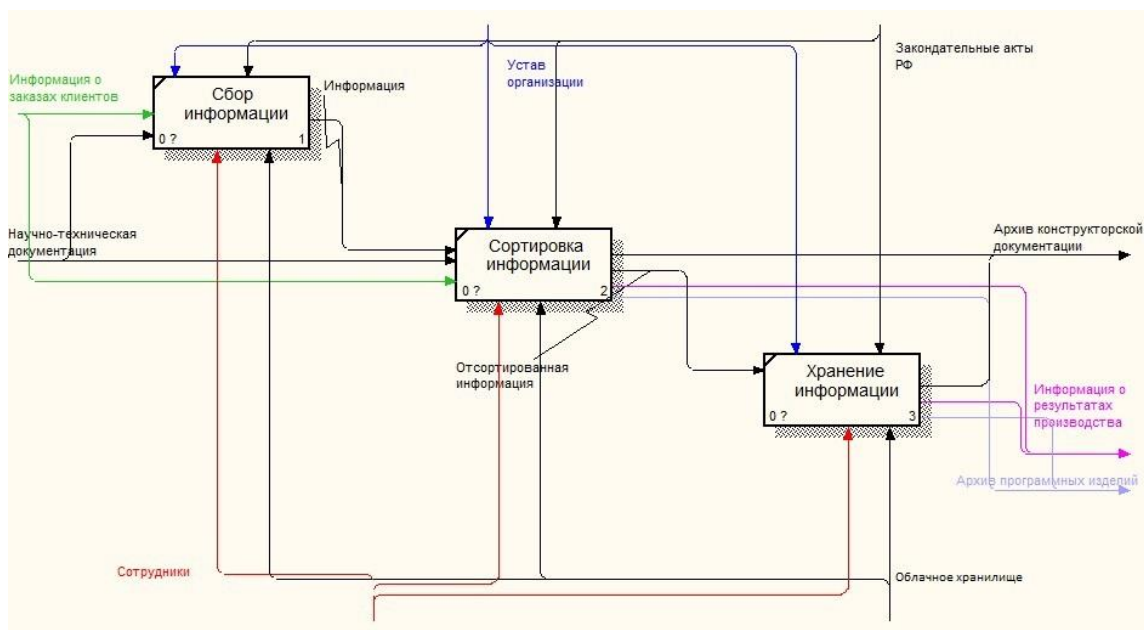


Рисунок 10 - Декомпозиция модели «Как будет»

Сбор и сортировка информации в целевой модели будут занимать гораздо меньше времени. Блока «Передача информации» теперь нет, так как благодаря использованию облачного хранилища у сотрудников есть совместный доступ ко всей корпоративной информации.

Выводы по главе 1

В данной главе был проведен анализ предметной области. На данный момент в организации для работы с корпоративной информацией используется локальный сервер. Покупка, обслуживание локального сервера обходится дорого, кроме того, некоторые необходимые функции у него отсутствуют. Поэтому было решено внедрить корпоративное облачное хранилище.

Глава 2 Логическое проектирование облачного хранилища

2.1 Выбор технологии логического моделирования облачного хранилища

Формирование логической модели заключается в построении диаграмм, отражающих информационную структуру, относящуюся к предметной области. Информация об информационных структурах, полученная на этапе анализа предметной области, используется для построения этих диаграмм. При «моделировании бизнес-процессов разработчики выделяют некоторые структурные элементы, с которыми должны работать пользователи будущей информационной системы. Обычно такие информационные структуры представлены документами или виртуальными представлениями данных, структурированными в соответствии с правилами сборки документов или структурами таблиц» [11].

Учитывая множество вариантов построения моделей базы данных, можно выделить два основных подхода:

- документарный - представляя документы в виде информационных структур, модель базы данных формируют в виде сущностей, идентичных документам предметной области, с выделением отдельных атрибутивных элементов в виде сущностей и проведением нормализации внутри сущностей и по связям между выделенными самостоятельными атрибутивными сущностями;
- объектный - подход, представляющий выделение в предметной области по бизнес-процессам и вспомогательным функциям информационных объектов, которые могут быть документами или информационными бизнес-элементами, с последующей функционализацией модели и нормализацией полученных из объектов сущностей.

Оба подхода учитывают правила и особенности проектирования реляционной базы данных и нормализации отношений. Основная разница

заключается в том, что при документарном подходе основу рассмотрения составляют документы и документооборот, из которых выделяется множество атрибутов, представляемых сущностями модели базы данных

«Объектный же подход рассматривает не каждый атрибут в отдельности, а их совокупность, формируемую в рамках бизнес-элементов или единых документов, создавая в модели базы данных множество отношений, к которым могут быть применены все известные правила нормализации» [17].

Для построения логической модели облачного хранилища был выбран объектный подход.

Логическая модель облачного хранилища представлена на рисунке 11.

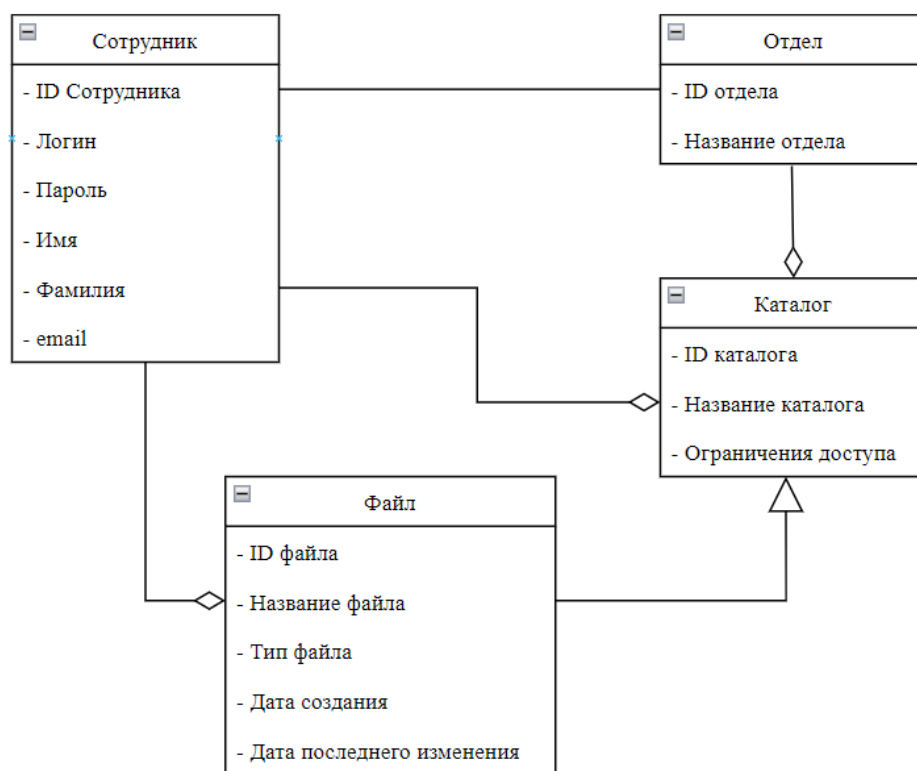


Рисунок 11 – Диаграмма классов облачного хранилища

Логическая модель содержит четыре уникальных класса данных, взаимосвязанных между собой.

2.2 Информационное обеспечение облачного хранилища

В системе мы используем четыре классификатора, которые были разработаны ранее:

- сотрудники;
- отделы;
- каталоги;
- файлы.

Классификатор «Сотрудники» был введён для идентификации сотрудников организации:

- Система кодирования – порядковая;
- Количество признаков классификации – 1;
- Объекты кодирования: сотрудники;
- Структурная формула классификатора.

Классификатор «Отделы» был введён для идентификации отделов организации:

- Система кодирования – порядковая;
- Количество признаков классификации – 1;
- Объекты кодирования: отделы;
- Структурная формула классификатора.

Классификатор «Каталоги» был введён для идентификации каталогов, используемых для хранения файлов и обеспечивающих ограничение доступа к информации:

- Система кодирования – порядковая;
- Количество признаков классификации – 1;
- Объекты кодирования: каталоги;
- Структурная формула классификатора.

Классификатор «Файлы» был введён для идентификации файлов, хранящихся в облачном хранилище:

- Система кодирования – порядковая;
- Количество признаков классификации – 1;
- Объекты кодирования: файлы;
- Структурная формула классификатора.

В таблице 5 приведены кодируемые множества.

Таблица 5 – Кодируемые множества

Наименование	Значность кода	Система кодирования
Сотрудники	4	порядковая
Отделы	2	порядковая
Каталоги	5	порядковая
Файлы	7	порядковая

На основании данных классификаторов производится взаимодействие между таблицами базы данных.

Объектом автоматизации является работа с документами, а именно:

- создание новых документов;
- просмотр, добавление, удаление или изменение документов;
- сохранение версий документов и возможность управления версиями;
- поиск документов по метаданным (названию, описанию, тэгам, атрибутам);
- разграничение доступа к документам разных пользователей, выдача прав на совершение определенных действий с документами.

На различных страницах облачного хранилища будут разные входные данные. В таблице 6 показаны названия страниц и что ожидается для входа:

На различных страницах облачного хранилища будут разные входные данные. Это сделано специально для реализации SOLID (Single responsibility, Open-Closed Principe, Liskov substitution, Interface segregation, Dependency inversion), а точнее Interface Segregation Principe.

Таблица 6 – Входные данные

Название страницы	Входные данные
Авторизация	Логин – строка от 3 до 25 символов; Пароль – строка от 8 до 100 символов.
Регистрация	Логин – строка от 3 до 25 символов (уникальный); Пароль – строка от 8 до 100 символов; Электронная почта – строка вида *@*. (уникальная), где * - любое количество любых символов (соблюдая правила написания электронной почты)
Главная страница, или страница с файлами	Быть авторизованным, то есть получив свой токен (скрыт от пользователя)
Загрузка файлов	Быть авторизованным, то есть получив свой токен (скрыт от пользователя), предоставить файлы с разрешенными форматами (pdf, doc, docx и др.)

Как видно в таблице 6, некоторые страницы имеют похожие входные данные, что значительно упрощает дальнейшую поддержку и расширение функциональности облачного хранилища.

В таблице 7 показаны названия страниц и что ожидается для входа.

Таблица 7 – Выходные данные

Название страницы	Выходные данные
Авторизация	Получен токен пользователя (занесен в local storage или в session storage, в зависимости от выбора пользователя)
Регистрация	Пользователь занесен в базу данных и переброшен на страницу с авторизацией
Главная страница, или страница с файлами	Список ваших файлов
Загрузка файлов	Файлы сохранены в хранилищах, занесены в БД со связью с пользователем. Релокация на страницу с файлами

Для каждой из страниц описаны выходные данные и принцип Interface Segregation Principle не нарушен.

2.3 Проектирование базы данных облачного хранилища

«ER-моделирование, или, другими словами, метод «сущность-связь», представляет собой один из способов создания инфологической модели. При использовании данного метода рассматриваются объекты, имеющие экземпляры, обладающие набором свойств – атрибутов» [23]. На рисунке 12 показана концептуальная ER-модель облачного хранилища.

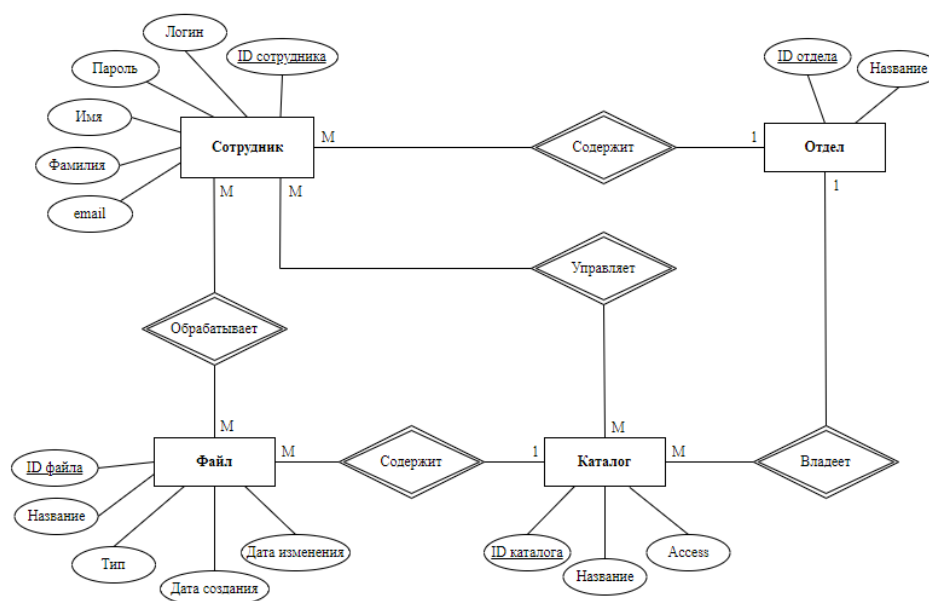


Рисунок 12 – Концептуальная ER-модель






В таблице 8 дано описание соответствия сущностей модели реальным объектам предметной области.

Таблица 8 – Сущности модели и реальные объекты

Название сущности	Реальный объект предметной области
Сотрудник	Сотрудники организации, имеющие доступ к облачному хранилищу согласно должностным обязанностям
Отдел	Структурное подразделение организации
Каталог	объект в файловой системе, упрощающий организацию файлов
Файл	именованная область данных на носителе информации, используемая как базовый объект взаимодействия с данными

В таблице 9 дано описание отношений между сущностями.

Таблица 9 – Отношения между сущностями модели

Отношение	Взаимодействие реальных объектов
	<p>Каждый сотрудник организации может работать только в одном отделе. В каждом отделе может числиться любое количество сотрудников</p>
	<p>Любой каталог облачного хранилища может иметь только один отдел в качестве подразделения-владельца</p>
	<p>Любой сотрудник организации может управлять любым количеством каталогов, которые в свою очередь могут управляться любым количеством сотрудников</p>
	<p>Любой сотрудник организации может обрабатывать любое количество файлов (создавать, удалять, изменять и т.д.), которые в свою очередь могут обрабатываться любым количеством сотрудников</p>
	<p>Каждый файл может храниться только в одном каталоге. Каждый каталог может содержать в себе любое количество файлов</p>

Для реализации облачного хранилища была выбрана реляционная база данных.

«Реляционная база данных – представляет собой базу данных, которая используется для хранения и предоставление доступа к взаимосвязанным элементам информации» [13]. «Реляционная база данных основана на реляционной модели, интуитивно понятном наглядном табличном предоставлении информации. Каждая строка в таблице такой базе данных, представляет собой запись с уникальным идентификатором, который является ключом» [18].

«Работа с реляционной базой данных происходит с помощью декларативного языка программирования SQL (Structured Query Language — «язык структурированных запросов»), при помощи данного языка запросов можно получать данные из таблиц и столбцов» [21].

Реляционная модель данных имеет такие достоинства как:

- строгие правила проектирования, базирующиеся на математическом аппарате;
- для организации запросов и написания программного обеспечения нет необходимости знать конкретную организацию базы данных во внешней памяти.

Также он обладает и недостатками, а именно:

- в результате логического проектирования появляется множество «таблиц». Это приводит к трудности понимания структуры данных;
- далеко не всегда предметная область может быть представлена в виде «таблиц».

Диаграмма сущностей реляционной базы данных облачного хранилища приведена на рисунке 13.

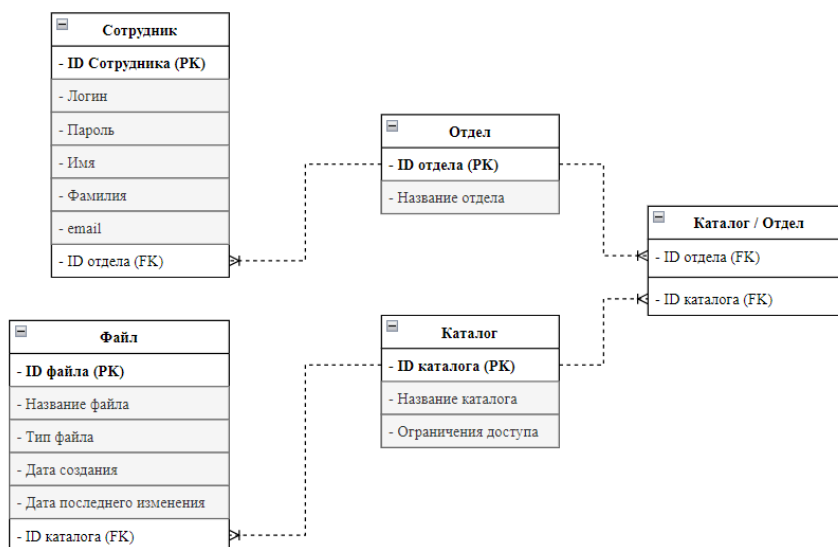


Рисунок 13 – Диаграмма сущностей реляционной базы данных

При построении модели данных, необходимо помнить, что «таблица» есть понятие нестрогое и неформальное и часто означает не «отношение» как абстрактное понятие, а визуальное представление отношения на бумаге или экране.

Для трансформации диаграммы классов в логическую модель данных и построения диаграммы сущностей реляционной БД использовалась методология IDEF1X.

2.4 Требования к аппаратно-программному обеспечению облачного хранилища

Техническое обеспечение — это комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.

Для обеспечения работы с облачным хранилищем требуются следующие технические средства:

- автоматизированные рабочие места сотрудников;
- сервер для работы облачного хранилища и его базы данных;
- локальная вычислительная сеть.

Текущая конфигурация автоматизированных рабочих мест, обеспечивает достаточную производительность для комфортной работы с облачным хранилищем (таблица 10). Все автоматизированные рабочие места сотрудников подключены к локальной вычислительной сети и имеют выход в глобальную сеть интернет.

Таблица 10 – Минимальные требования к ПК на рабочем месте сотрудника

Характеристика	Значение, не менее
Частота процессора	1 ГГц
ОЗУ	2 Гб
Объем жесткого диска	500 Гб
Монитор	1240x1024

Требуется купить сервер для работы базы данных для облачного хранилища. Данный сервер, будет функционировать стандартный рабочий день. Должен обеспечивать работу 100 пользователей.

Технические требования к серверу:

- Процессор не хуже Intel Xeon Gold 6250L;
- Объем оперативной памяти минимум 8Гб;
- Жесткие диски 1Тб не менее двух с возможностью собрать из них RAID1;

Сервер Lenovo SR530 – имеет форм-фактор 1U, подходит для монтажа в стандартную стойку 19 дюймов. Имеет возможность установки 1 или двух процессоров в зависимости от требований. Поддержка до 12 модулей оперативной памяти. Возможность установить до 12 жестких дисков различного формата. Имеет модуль удаленного управления.

Как видно из таблицы 11, при одинаковой конфигурации серверы имеют ощутимое различие в стоимости – около 800 у.е.

Таблица 11 – Сравнение серверов

Параметр	Lenovo SR530	Lenovo ST50
Корпус	1U Rackmount	MidiTower
Центральный процессор	3.9GHz Intel Xeon Gold 6250L, 8 core, TDP 185W	3.9GHz Intel Xeon Gold 6250L, 8 core, TDP 185W
Оперативная память	2xDIMM 2048 DDR4 2666MHz	2xDIMM 2048 DDR4 2666MHz
Сетевой адаптер	Broadcom 5720 1Gbe RJ45 2-port Broadcom 57416 10GBASE-T-2Port	Broadcom 5720 1Gbe RJ45 2-port Broadcom 57416 10GBASE-T-2Port
Жесткие диски	2 SATA 3.5 1Tb 7.2k 6GB	2 SATA 3.5 1Tb 7.2k 6GB
Стоимость	4095 у.е.	3272 у.е.

Выводы по главе 2

В данной главе была выбрана технология логического моделирования облачного хранилища:

- построена логическая модель облачного хранилища;
- построена модель базы данных;
- спроектировано информационное обеспечение: входная и выходная информация, классификаторы и системы кодирования.

Глава 3 Физическое проектирование облачного хранилища

3.1 Выбор архитектуры облачного хранилища

Для реализации облачного хранилища сознательно выбрана клиент-серверная архитектура или сетевая архитектура [1], [16], [24]. Пример, отображающий клиент-серверную архитектуру изображен на рисунке 14.

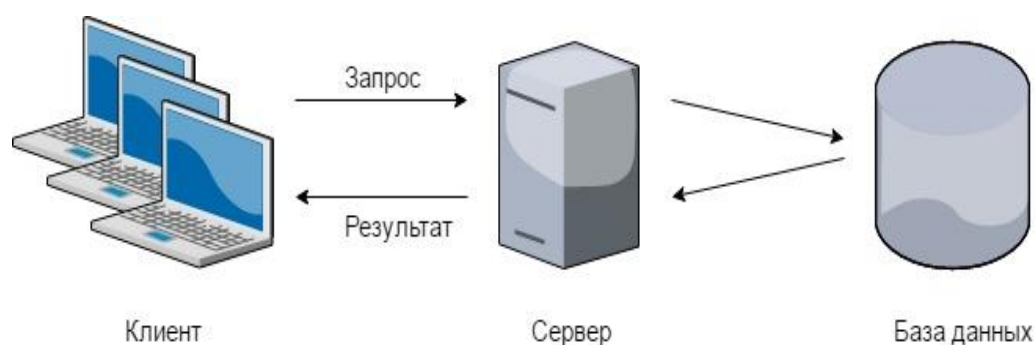


Рисунок 14 – Общая схема клиент-серверной архитектуры

«Клиент – может выступать как рабочая станция, так и планшет или смартфон, данный объект инициирует запрос к серверу на предоставление информации» [21].

«Сервер – в качестве сервера может выступать как рабочая станция, так и сервер, данный элемент в архитектуре принимает запросы от клиентов и предоставляет информацию по требуемым запросам клиентов» [22].

«База данных – совокупность данных, хранимых в соответствие со схемой данных, работу с данными можно проводить с помощью языка структурированных запросов» [22].

«Клиент и сервер производят взаимодействия в любой сети передачи данных будь то глобальная сеть или локальная» [22]. Взаимодействие клиента и сервера в разработанной системе осуществляется при помощи протокола HTTPS.

HTTPS (HyperText Transfer Protocol Secure – «протокол передачи гипертекста») – расширенный протокол передачи данных. Ранее использовался для передачи гипертекстовых документов, в современных реалиях он получил распространение для передачи любых данных. Данный протокол передаёт данные в режиме «запрос-ответ». Ключевым объектом в данном протоколе является ресурс с уникальным идентификатором клиентского запроса. Типовыми ресурсами могут являться файлы на сервере или другие логические, или абстрактные данные.

Таким образом клиент-серверная архитектура обладает рядом преимуществ: более высокой производительностью, улучшенными средствами обеспечения безопасности и целостности данных, возможностью задания бизнес-правил.

3.2 Выбор технологии разработки программного обеспечения облачного хранилища

У этапа верстки основных страниц есть свои особенности. «Показателем хорошей верстки является отлаженная и быстрая работа приложения на всех браузерах, которыми только может воспользоваться сотрудник организации» [10]. В основном большинство пользователей не станут устанавливать дополнительное ПО для просмотра содержимого. Для создания html-кода страниц будет использоваться IDE IntelliJ Webstorm и фреймворк Vue.js. Для реализации «привлекательности» и сохранения одного стиля страниц будет использоваться фронтенд фреймворк Vuetify. Это идейный последователь Material Design для React.js

Показателем функциональности является скорость обработки данных, переданных пользователем. Для решения этой задачи будет использоваться асинхронное программирование и, в частности, веб-фреймворк Django, так как он включает в себя множество компонентов для удобной разработки.

Для объединения фронтенда и бэкенда должны быть разработаны единые интерфейсы для «общения» пользователя и сервера. Например, одним из таких интерфейсов является REST API. У веб-фреймворка Django [8] есть модуль Django Rest Framework, который позволяет с легкостью разработать API.

Следующим этапом является разработка прототипа клиента облачного хранилища. «Прототипом называется схематичное представление страницы будущего сайта, отражающее расположение всех необходимых элементов – навигационное меню, формы захвата и обратной связи, зона просмотра основной информации, кнопки и т.д.» [10]. Прототипирование позволяет продумать размещение основных блоков и взаимодействие людей с сайтом заранее. На данной стадии любые изменения вносить гораздо удобнее и дешевле по сравнению с готовым продуктом. Таким образом, составление прототипов страниц позволяет сократить сроки и объем работ по разработке клиента.

Следующим этапом является программирование клиента облачного хранилища. Для выполнения работ в данном этапе были выбраны следующие инструменты:

- язык программирования Python 3.8 и веб-фреймворк Django;
- язык программирования JavaScript и фреймворк Vue.js;
- ПО IntelliJ Pycharm;
- ПО IntelliJ Webstorm;
- браузер Google Chrome;
- ПО Postman.

Язык программирования Python 3.8 [5] – «высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика и читаемости кода» [28]. Синтаксис ядра Python минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации.

Python является мультипарадигмальным языком программирования, поддерживающим императивное, процедурное, структурное, объектно-ориентированное программирование, метапрограммирование и функциональное программирование. Аспектно-ориентированное программирование частично поддерживается через декораторов, более полноценная поддержка обеспечивается дополнительными фреймворками. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений с глобальной блокировкой интерпретатора, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

Эталонной реализацией Python является интерпретатор CPython, поддерживающий большинство активно используемых платформ. Он распространяется под свободной лицензией Python Software Foundation License, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные.

Django [8], [25] - свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation. Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий этого фреймворка от некоторых других (например, Ruby on Rails). Один из основных принципов фреймворка - DRY (англ. Don't repeat yourself). Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.

JavaScript [2] - мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. «JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Основные архитектурные

черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса» [11].

Vue.js - это прогрессивный фреймворк для создания пользовательских интерфейсов. В отличие от фреймворков-монолитов, Vue создан пригодным для постепенного внедрения. Его ядро в первую очередь решает задачи уровня представления (view), что упрощает интеграцию с другими библиотеками и существующими проектами. С другой стороны, Vue полностью подходит и для создания сложных одностраничных приложений (SPA, Single-Page Applications), если использовать его совместно с современными инструментами и дополнительными библиотеками.

IntelliJ Pycharm является одним из самых популярных текстовых редакторов для работы с языком программирования Python. В нем включена поддержка синтаксиса HTML, CSS, JavaScript (без авто дополнения), есть возможности работы с базами данных, так же в этой версии намного удобнее заниматься «научным» программированием. Поддерживает чтение и редактирование широкого диапазона файлов: текстовые документы, файлы исходных кодов разных языков программирования, баз данных.

IntelliJ Webstorm - является одним из самых удобных текстовых редакторов для работы с языком программирования Python. Умное авто дополнение кода может определить с каким синтаксисом JavaScript вы работаете: TypeScript, Native, JQuery и т.д. В Webstorm встроен собственный файловый менеджер, с помощью которого можно просматривать содержимое компьютера и запускать выбранные файлы в браузере.

Google Chrome выбран для основной работы и отладки функционала из всего многообразия браузеров неслучайно. С помощью Google DevTools можно узнать размер фото, размер и тип шрифта, отладить CSS, найти ошибку в JS, найти причину медленной работы скрипта и т.п.

Postman – один из инструментов тестирования API. Позволяет проводить тестирование API не только вручную, но и автоматизировано.

3.3 Выбор СУБД облачного хранилища

В качестве СУБД [12] для работы была выбрана MariaDB.

MariaDB [27] – ответвление от системы управления базами данных MySQL не требует лицензирования.

Критерии выбора:

- свободна от аппаратной и софтверной платформы – не имеет зависимости в отличие от MS SQL Server;
- полностью бесплатна в отличие от MySQL, Oracle, MS SQL server;
- поскольку является наследником MySQL достаточно проста в разворачивание и обслуживание;
- не требовательна к аппаратным ресурсам в отличии от СУБД MS SQL Server и Oracle;
- имеет высокую производительность;
- стоимость сопровождения намного ниже, чем у СУБД MS SQL Server и Oracle;
- есть поддержка многопользовательских подключений.

СУБД MariaDB подходит для организаций, которым требуется надежный инструмент управления базами данных, но бесплатный и не требует высококвалифицированных специалистов.

3.4 Разработка физической модели данных облачного хранилища

Физическая модель данных (рисунок 15), в отличие от логической, зависит от конкретной СУБД, фактически являясь отображением системного каталога. В физической модели содержится информация обо всех объектах БД. Поскольку стандартов на объекты БД не существует (например, нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей. Если в логической

модели не имеет значения, какой конкретно тип данных имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах – таблицах, колонках, индексах, процедурах и т. д.

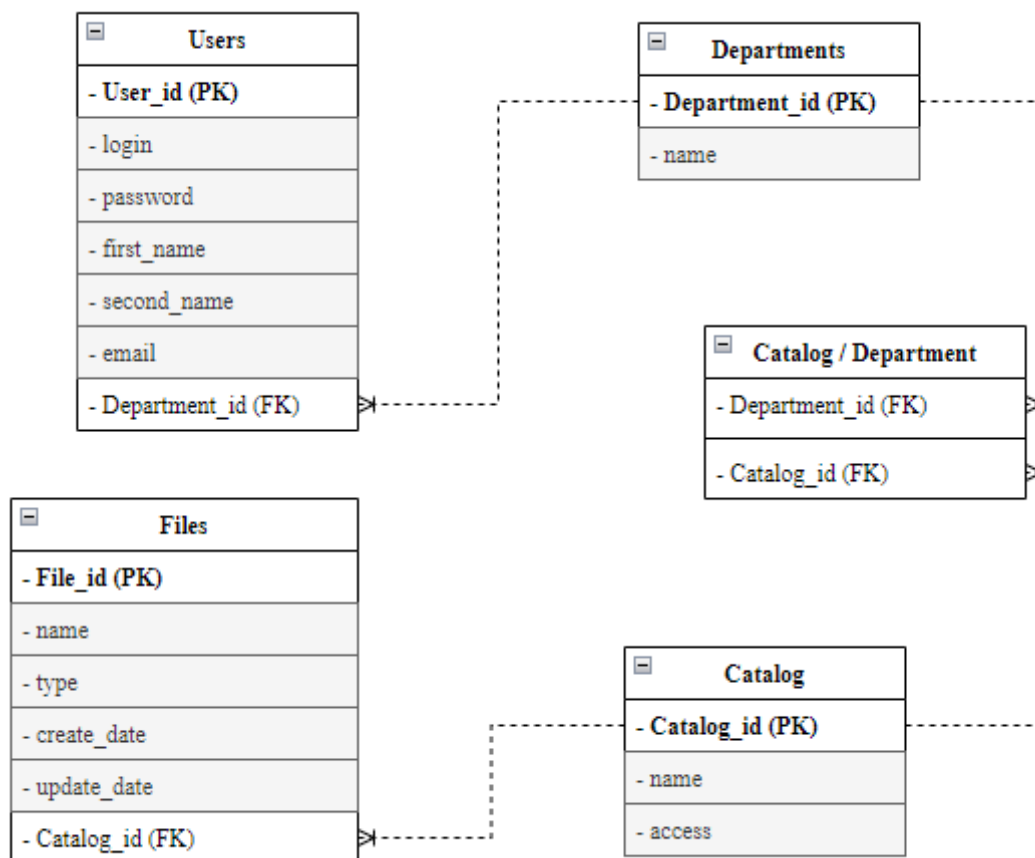


Рисунок 15 – Структура базы данных облачного хранилища

Описание структуры и назначения каждой из созданных таблиц:

Таблица «Departments» (Таблица 12) – служит для хранения информации об оргструктуре организации.

Таблица 12 – Описание таблицы «Departments»

Атрибут	Тип данных	Описание атрибута
Department_id	Целочисленный	Идентификатор отдела (первичный ключ)
name	Текстовый	Название отдела

Таблица «Users» (Таблица 13) – содержит данные о сотрудниках организации.

Таблица 13 – Описание таблицы «Users»

Атрибут	Тип данных	Описание атрибута
User_id	Целочисленный	Идентификатор (первичный ключ)
login	Текстовый	Пользовательский логин
password	Текстовый	Пользовательский пароль
first_name	Текстовый	Имя пользователя
second_name	Текстовый	Фамилия пользователя
Email	Текстовый	Адрес электронной почты пользователя
Department_id	Целочисленный	Идентификатор отдела (внешний ключ)

Таблица «Files» (Таблица 14) – содержит информацию о файлах и месте их хранения.

Таблица 14 – Описание таблицы «Files»

Атрибут	Тип данных	Описание атрибута
File_id	Целочисленный	Идентификатор файла (первичный ключ)
name	Текстовый	Название документа
type	Текстовый	Тип документа
create_date	Дата	Дата создания
update_date	Дата	Дата обновления
Catalog_id	Целочисленный	Идентификатор каталога (внешний ключ)

Таблица «Catalog» (Таблица 15) – содержит полную информацию об используемых в облачном хранилище каталогах, с учетом ограничений прав доступа пользователей.

Таблица 15 – Описание таблицы «Catalog»

Атрибут	Тип данных	Описание атрибута
Catalog_id	Целочисленный	Идентификатор каталога (первичный ключ)
name	Текстовый	Название каталога
access	Текстовый	Уровень доступа к каталогу

Разработанная физическая модель способна удовлетворить текущие потребности организации, а также легко может быть масштабирована при первой же потребности.

3.5 Разработка программного обеспечения облачного хранилища

На рисунке 16 обозначена вложенность составных элементов клиентской части облачного хранилища. Все эти элементы состоят из элементов фреймворка Vuetify.js, который в свою очередь является набором обычных HTML-тегов и CSS.

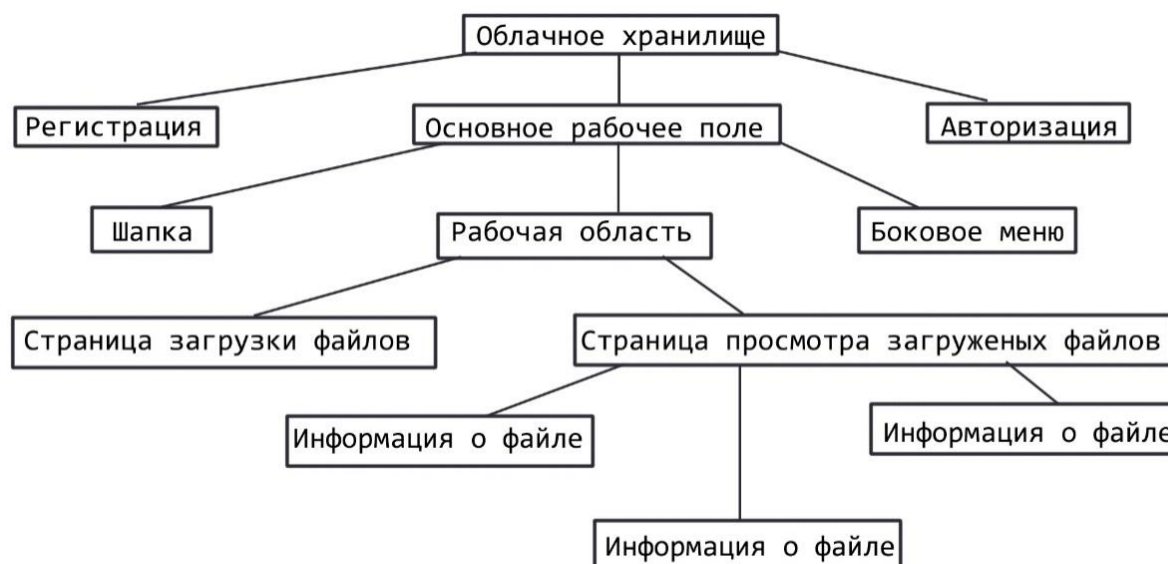


Рисунок 16 – Структура модулей фронтенд-части облачного хранилища

Как видно на рисунке 17, сервис хранилище и сервис управления пользователями содержит одинаковые по названию файлы.

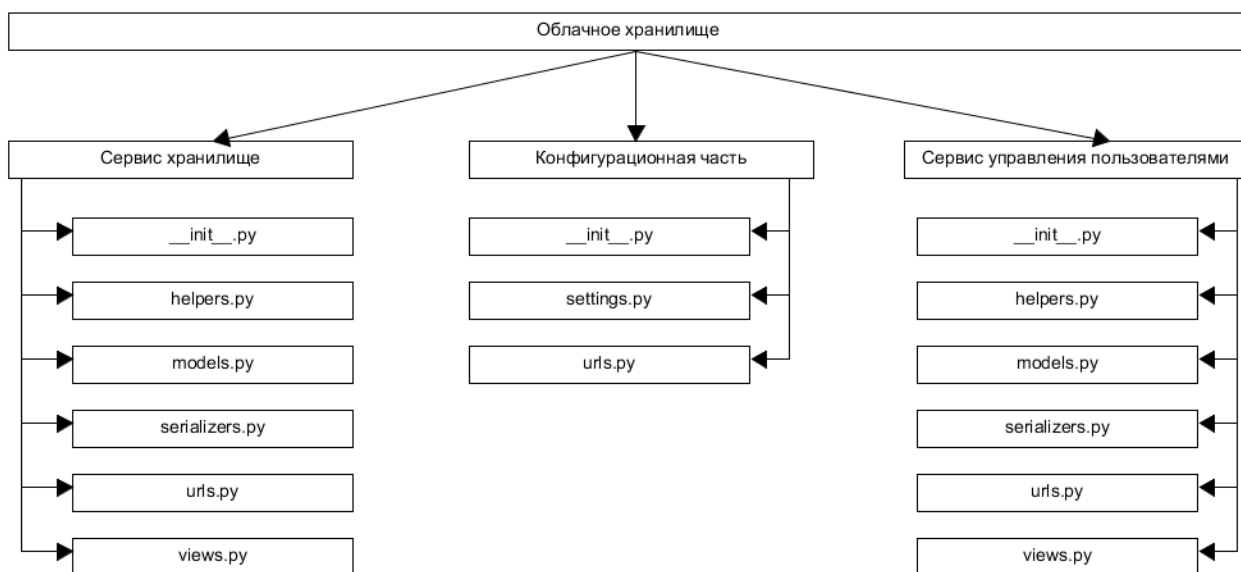


Рисунок 17 – Структура папок бэкенд-части облачного хранилища

Рассмотрим каждый файл по отдельности, в чем их схожесть и в чем отличие:

- `__init__.py` – файл инициализации пакета в python.
- `helpers.py` – файл с дополнительными методами. Например, в сервисе хранилища в этом файле лежит функция удаления файла. В то же время в сервисе управления пользователями файл пустой.
- `models.py` – файл с моделями для Django ORM. Например, в сервисе хранилища в этом файле лежит модель `File`. В тоже время в сервисе управления пользователями `Profile` - наследник `User`, а также генерация `token`.
- `serializers.py` – файл сериализации моделей Django ORM, для корректного вывода в формате JSON.
- `urls.py` – файл, содержащий пути к различным точкам вывода ответа от API

– views.py – файл, содержащий основную логику вывода информации.

На рисунке 18 приведена «внешняя» схема работы облачного хранилища.

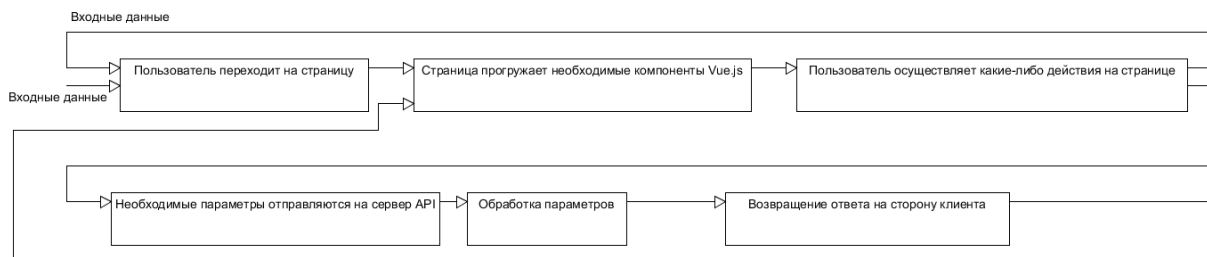


Рисунок 18 – Схема работы облачного хранилища

Входные данные на этой схеме могут отличаться для каждой страницы облачного хранилища. Подробнее про схему:

– пользователь переходит на страницу. Пользователь, переходит по одному из URL-адресов облачного хранилища, разрешенных ему для посещения, например, неавторизованный пользователь всегда перебрасывается на страницу авторизации;

– страница прогружает необходимые компоненты Vue.js. При обращении к определенному URL-адресу Vue.js понимает какие компоненты используются на странице;

– пользователь осуществляет какие-либо действия на странице. Например, для загрузки файлов на сервер, пользователь должен сначала загрузить их на клиенте и нажать на кнопку «Загрузить»;

– необходимые параметры отправляются на сервер API. Обязательными параметрами для всех обращений, кроме авторизации и регистрации является добавление в headers-заголовки запроса добавление: Authorization Token token, где token-ваш токен;

– обработка параметров. Сервер API принимая параметры, понимает, какой пользователь к нему обратился и зачем. Используя эти данные, он отдает команду хранилищу, что необходимо сделать;

– возвращение ответа на сторону клиента. После всех манипуляций API формирует ответ, в зависимости от успешности операции.

Схема использования файлов такая, как показана на рисунке 19.

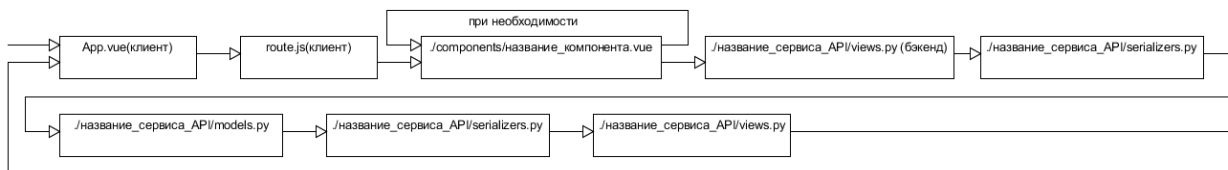


Рисунок 19 – Схема использования файлов

Где:

– App.vue – базовая страница для фреймворка Vue.js. Внутри нее преобразовываются компоненты и отрисовывается на стороне клиента;

– Route.js – страница с URL-адресами и компонентами для отрисовки;

– Название_компонента.vue – сам компонент Vue.js. В нем содержится верстка компонента, JavaScript-методы и другие составляющие для формирования страницы;

– Views.py – в этом файле содержится вся основная логика API. Здесь парсятся данные для сериализации и необходимая команда для хранилища;

– Serializers.py - файл сериализации моделей Django ORM, для корректного вывода в формате JSON;

– Models.py - файл с моделями для Django ORM.

Так как для доступа к основным функциональностям облачного хранилища необходима авторизация, то пользователю надо запустить клиент и появится окно авторизации. Окно авторизации представлено на рисунке 20.

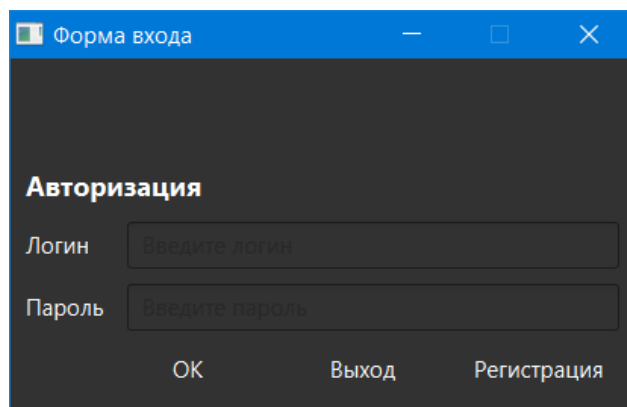


Рисунок 20 – Форма авторизации пользователей

Если учетная запись отсутствует, то необходимо зарегистрироваться с помощью формы регистрации. Для этого надо нажать на кнопку «Регистрация», которая загрузит форму регистрации (рисунок 21). В поле «Логин» ввести уникальное имя своей будущей учетной записи. В поле «Пароль» ввести пароль своей будущей учетной записи. Если какие-то данные будут подставлены не верно, то будет показана ошибка, и регистрация не будет произведена.

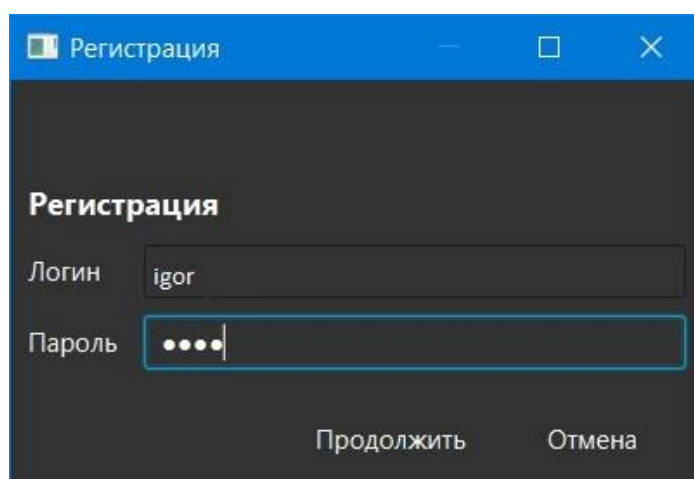


Рисунок 21 – Форма регистрации пользователей

После авторизации вы попадаете на страницу с файлами или главную страницу облачного хранилища. Здесь будет показан список ваших файлов, а

также меню с помощью которого можно перемещаться по облачному хранилищу (рисунок 22). На главной странице имеется список каталогов, а также название краткое, полное название и описание файлов.

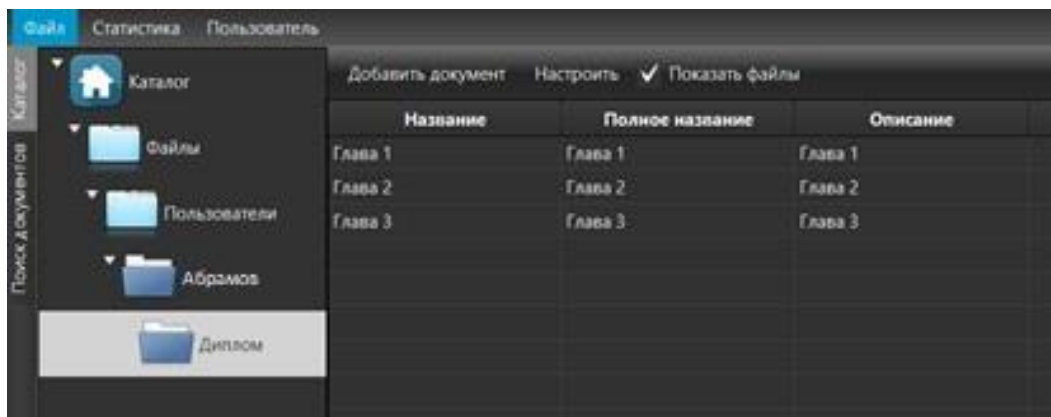


Рисунок 22 – Главная страница облачного хранилища

Структуру каталога (рисунок 23) можно увидеть, нажав на левую кнопку мыши по соответствующей иконке. Откроется таблица документов с кратким названием, полным названием и описанием.

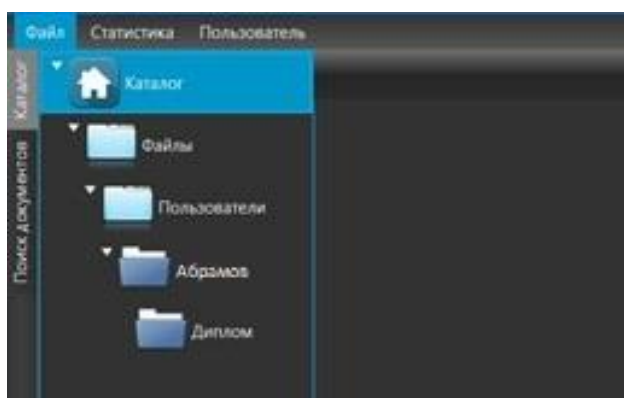


Рисунок 23 – Просмотр структуры каталога

Для отображения информации о файле, нужно вызвать контекстное меню. В открывшемся окне помимо информации о файле, будут отображены следующие опции: добавление тэгов, удаление файла, изменение информации

обновление файла, сохранение копии текущего файла, и настройки приватности доступа и безопасности файла.

Операция «Сохранить копию» даёт возможность сохранить копию текущего файла в указанное место на диске пользователя. При нажатии на кнопку появляется всплывающее окно, в котором пользователь может выбрать путь, по которому будет сохранена копия. Работа операции «Сохранить копию» показана на рисунке 24.

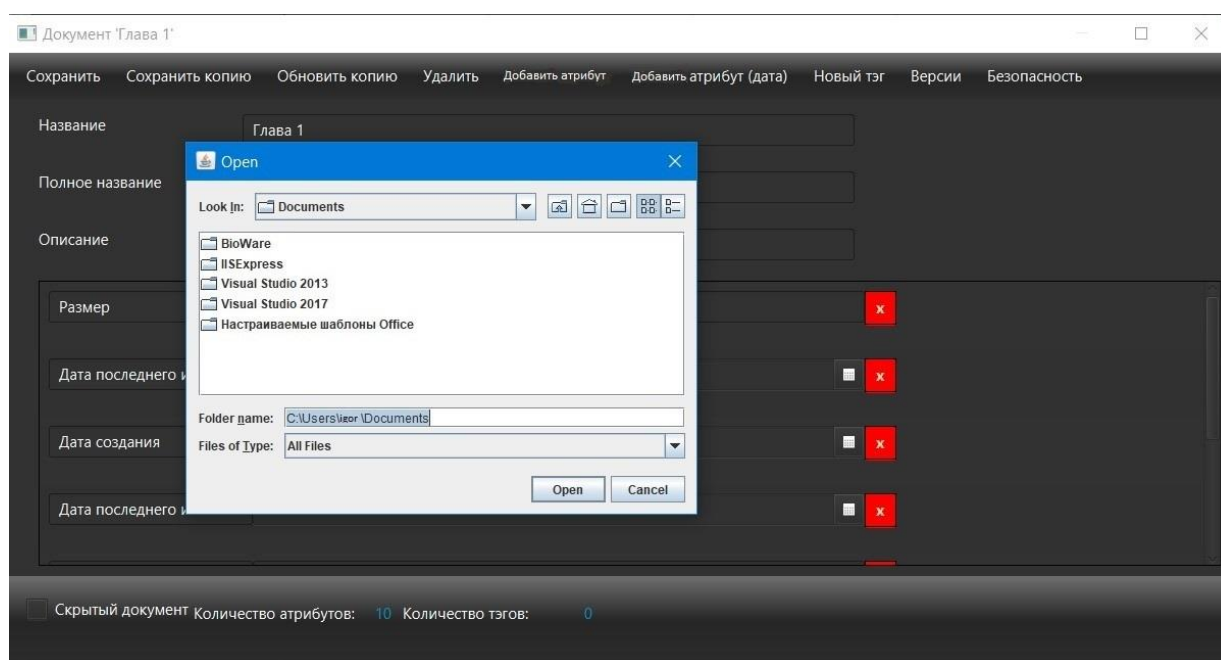


Рисунок 24 – Работа операции «Сохранить копию»

Операция «Обновить копию» загружает измененную версию текущего файла. Если нажать на кнопку, то откроется всплывающее окно, где пользователь может указать путь к каталогу, в котором находится измененная версия файла. Работа операции «Обновить копию» изображена на рисунке 25.

Операция «Удалить» удаляет выбранный файл из облачного хранилища. Операция «Добавить атрибут» добавляет новые атрибуты к файлу. Это позволяет пользователю добавлять заметки для описания файла, которые сочтёт необходимыми. В будущем, данные атрибуты позволят провести по ним поиск.

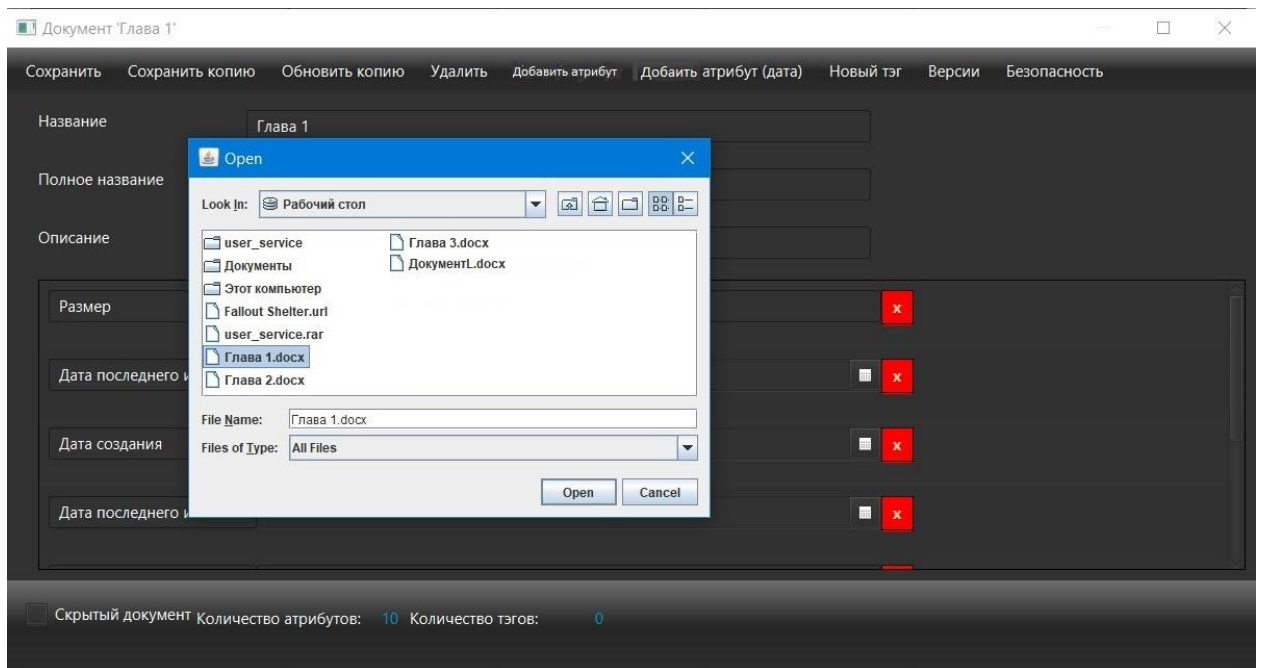


Рисунок 25 – Работа операции «Обновить копию»

При нажатии на кнопку «Добавить атрибут», в описании документа появляются новые поля, в которых нужно указать название и описание нового атрибута. Работа операции «Добавить атрибут» показана на рисунке 26.

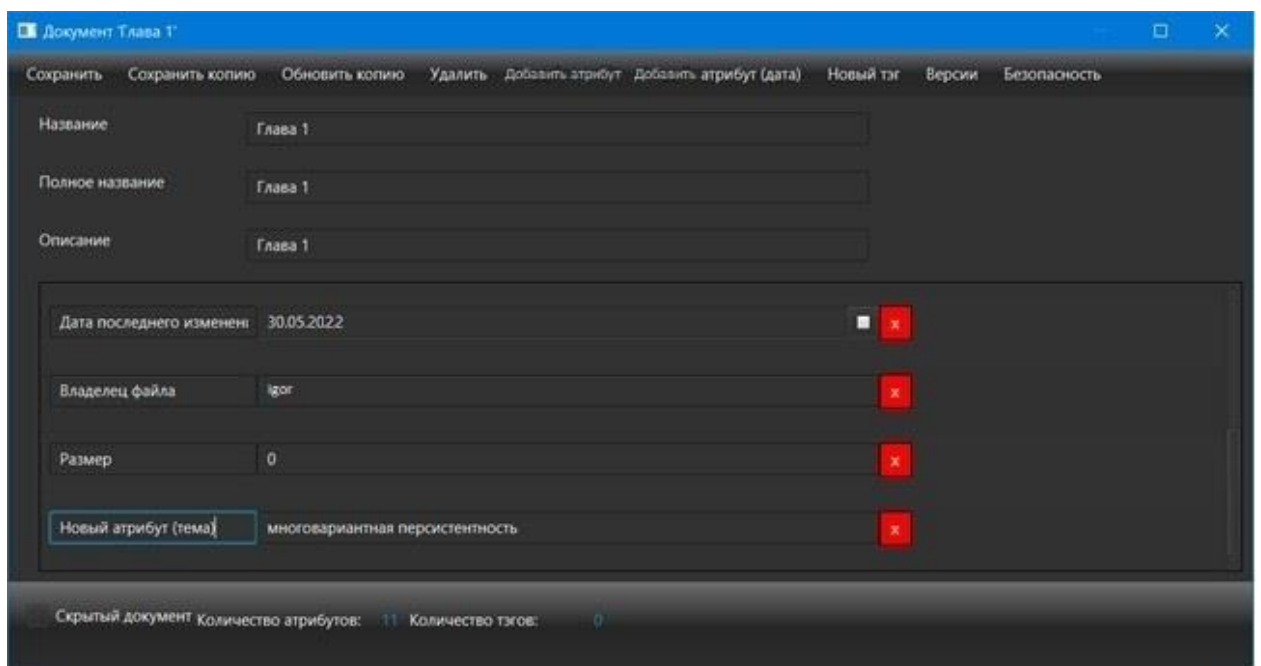


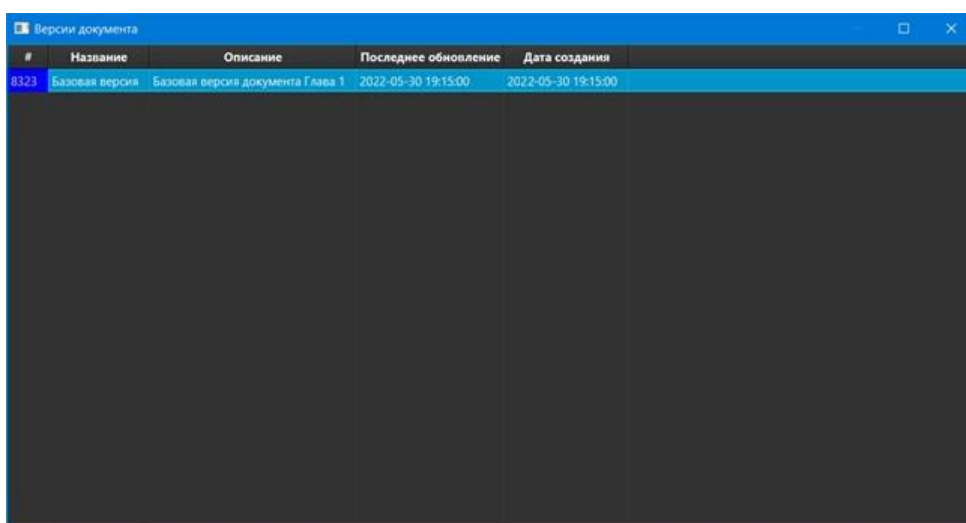
Рисунок 26 – Работа опции «Добавить атрибут»

Операция «Добавить атрибут (дата)» даёт возможность добавить к файлу пользовательский атрибут в виде даты. При нажатии на кнопку «Добавить атрибут (дата)», в описании к файлу появятся дополнительные поля, в которых нужно указать название атрибута и дату. Проставить в поле дату можно как в ручном режиме, так и путем выбора иконки, где в раскрываем окне можно выбрать нужную дату.

Операция «Новый тэг» даёт возможность задать тэги для файла, а также ключевые слова, например, для более лёгкого поиска данного файла. При нажатии на кнопку «Новый тэг» появляется всплывающее окно, в данном окне пользователь может указать значение тэга. После сохранения значения тэги будут доступны к просмотру в информации о файле.

Операция «Скрытый документ» данная операция задаёт видимость файла только для автора, другие пользователи не смогут увидеть данный файл. Задать данный атрибут можно поставив галочку путём выбора этого атрибута.

Вкладка «Версии документа» (рисунок 27) отображает информацию о файле, содержит полное название, описание с ключевыми тэгами, дату последнего изменения, а также дату создания. На данной вкладке можно вызвать контекстное меню для создания копии данного файла.



#	Название	Описание	Последнее обновление	Дата создания
8323	Базовая версия	Базовая версия документа Глава 1	2022-05-30 19:15:00	2022-05-30 19:15:00

Рисунок 27 – Общий вид вкладки «Версии документа»

При вызове контекстного меню, отобразится меню, в данном меню есть возможность задать опции сохранения копии файла на основе выбранного файла, также задать атрибуты для новой копии файла. При активации операции «Сохранить копию» появляется окно (рисунок 28), в данном окне пользователь указывает путь где будет сохранена копия на локальном диске.

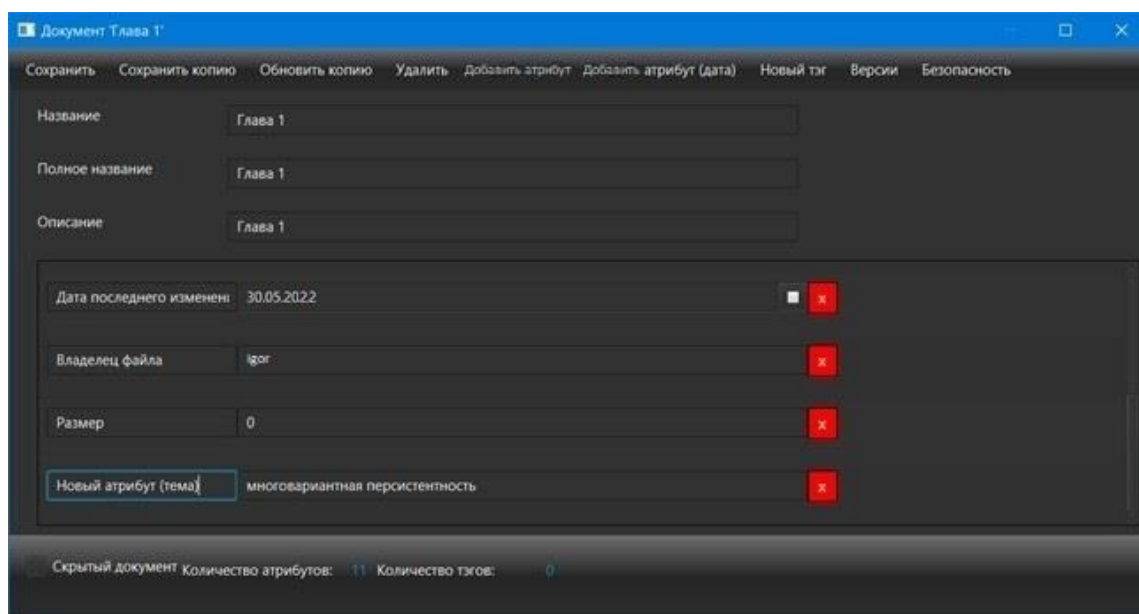


Рисунок 28 – Работа операция «Сохранить копию»

При выборе операции «Создать документ» (рисунок 29), на основе выбранного файла, будет создана копия, данная копия будет отображаться как самостоятельный файл.

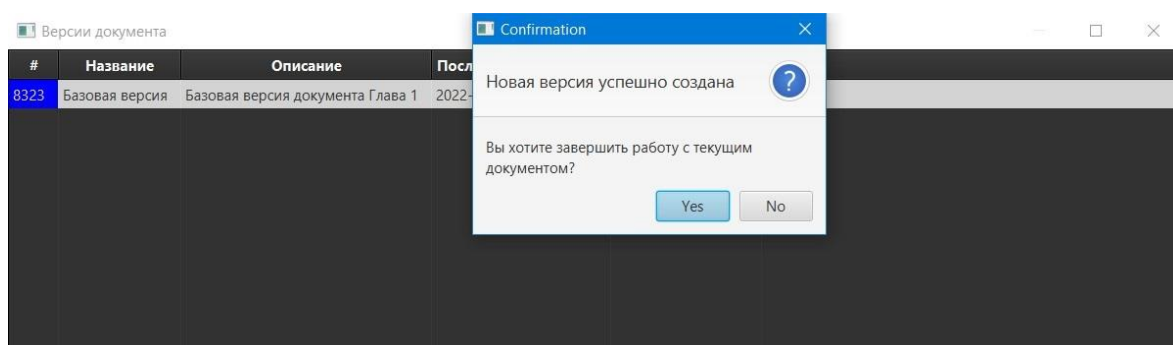


Рисунок 29 – Работа операции «Создать документ»

Итога работы операции «Создать документ» показан на рисунке 30. Как показано на данном рисунке, отображается новый файл и поскольку он создан на основе существующего файла у него есть метка [CLONE].



Рисунок 30 – Создание нового файла с помощью операции «Создать документ» из уже существующего файла.

На вкладке «Настройка безопасности» пользователь может задать базовые ограничения по работе с файлом для остальных пользователей.

Для пользователя доступны такие настройки безопасности как:

- ограничение загрузки файлов;
- ограничение удаления файлов;
- ограничение обновления файлов;
- ограничение правки тэгов и метаданных для файла;
- ограничение копирования версии файла;
- ограничение копирования любого файла;
- ограничение просмотра файла.

Для пользователя доступны варианты просмотра и изменений настройки безопасности других пользователей.

Общий вид вкладки «Настройки безопасности» показан на рисунке 31

Для задания настроек безопасности требуется выбрать пункт «Добавить пользователя». В открывшемся контексте меню списка пользователей, выбираем нужного пользователя и добавляем ему нужные настройки

безопасности, которые нам требуются, так же имеется возможность задать все настройки сразу выбрав пункт «Выбрать все».

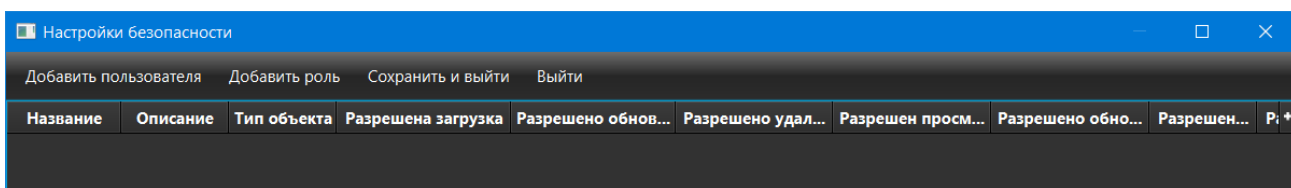


Рисунок 31 – Общий вид вкладки «Настройки безопасности»

Базовый вид настроек безопасности для пользователя показан на рисунке 32.

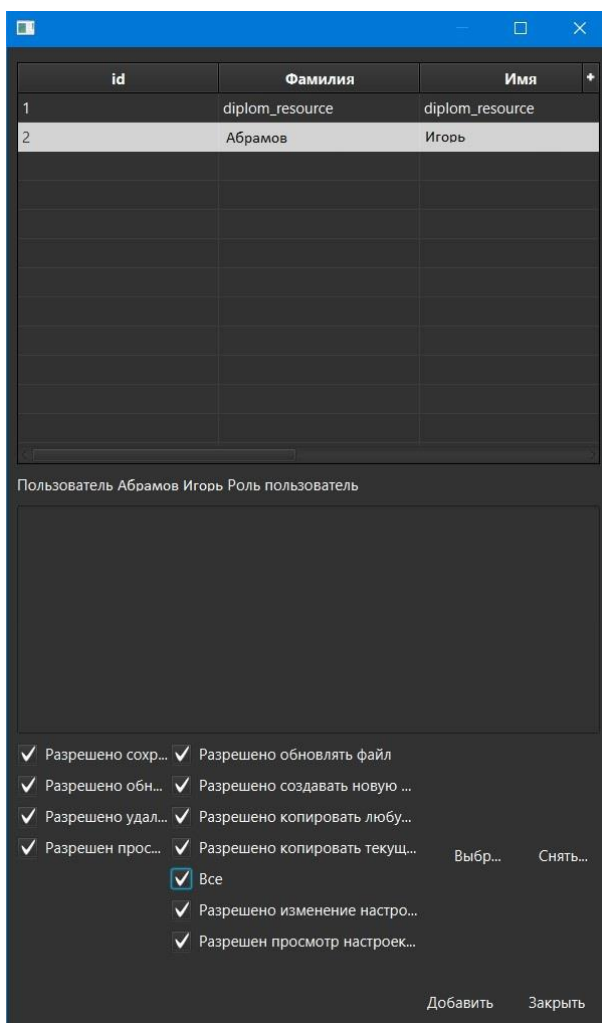


Рисунок 32 – Базовый вид настроек безопасности

Так же настройки безопасности можно применять к группам пользователей по ролям. Для использования данной операции, нужно на вкладке «Настройка безопасности» выбрать «Добавить роль», в открывшемся контекстом меню выбрать требуемую роль из списка и задать требуемые настройки безопасности. Базовый вид настроек безопасности для ролей отображен на рисунке 33.

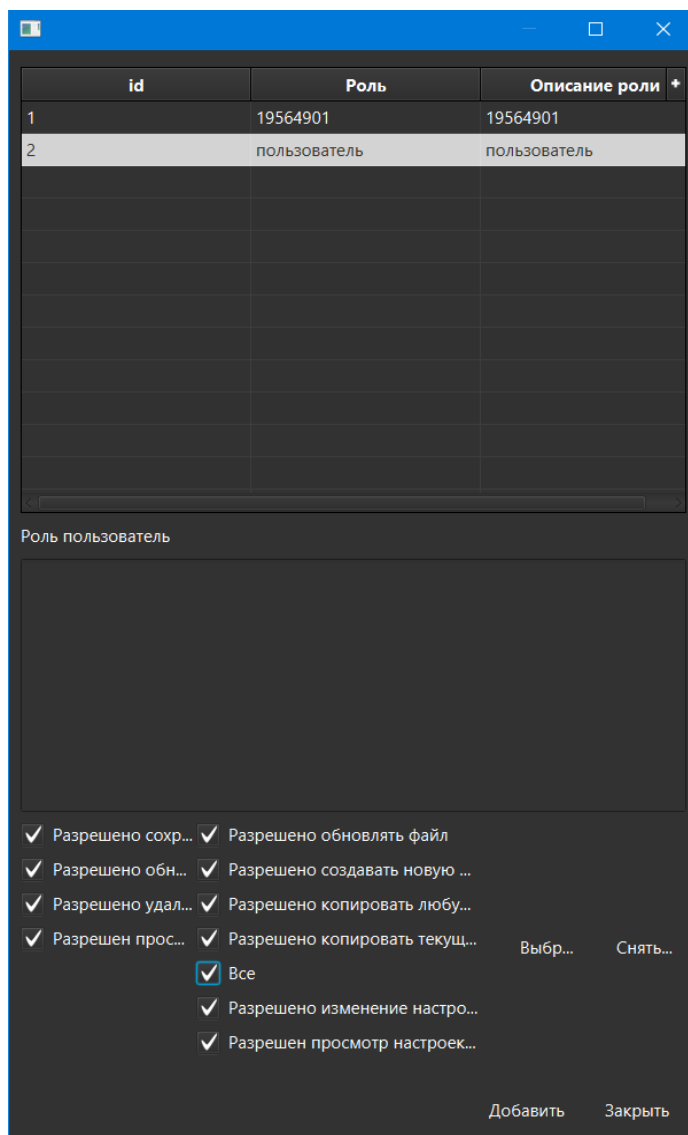


Рисунок 33 – Базовый вид настроек безопасности по ролям.

У пользователя имеется возможность удалять и править каталог, для этого требуется вызвать контекстное меню, в котором есть возможность,

добавлять директорию, добавить группу, добавить автопоиск. Возможности работы с каталогом из контекстного меню показаны на рисунке 34.

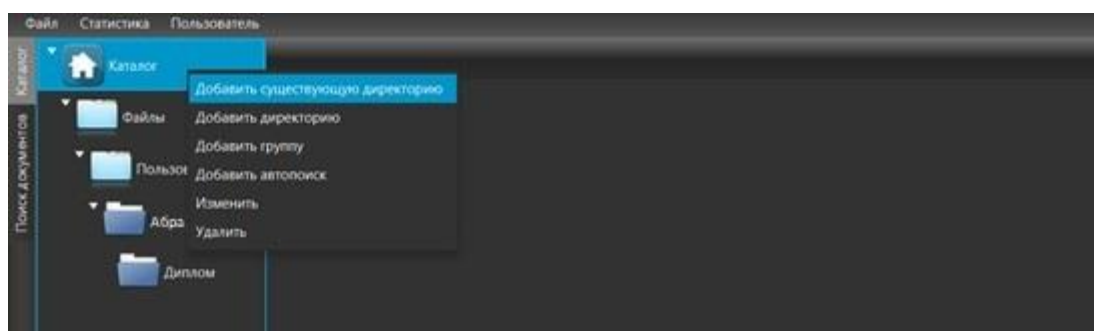


Рисунок 34 – Возможности работы с каталогом из контекстного меню

При активации операции «Добавить существующую директорию», появляется контекстное меню (рисунок 35), в данном окне пользователь указывает локальный путь на диске, где будет находиться директория. Когда данная директория будет добавлена, она отобразится вместе со всеми файлами, которые в ней есть.

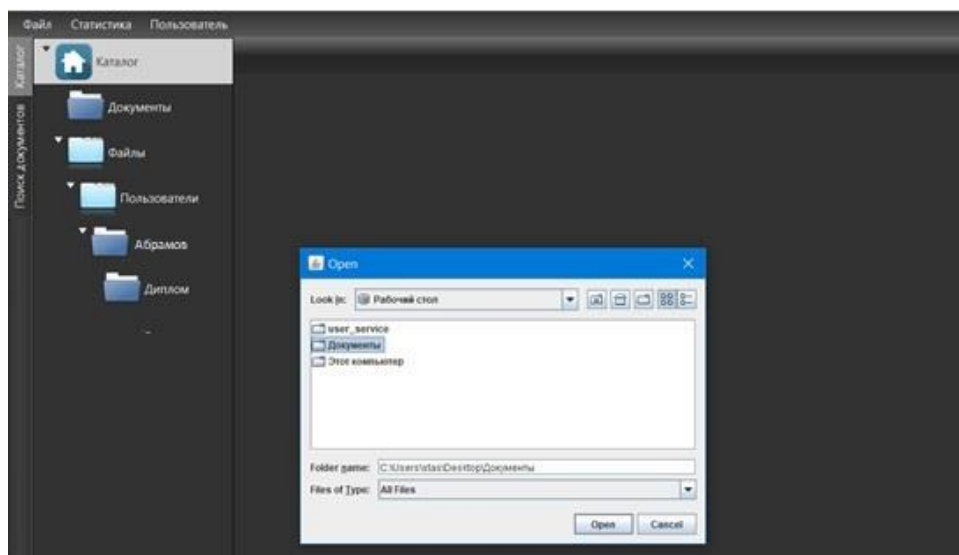


Рисунок 35 – Отображение работы операции «Добавление существующей директории»

Автопоиск – операция, которая переносит все найденный файлы в папку пользователя, данная операция использует ранее заданные пользователем критерии поиска. При активации операции «Добавить автопоиск» откроется окно «Настройка поиска», в данном окне задаются критерии поиска. Окно «Настройка поиска» показана на рисунке 36.

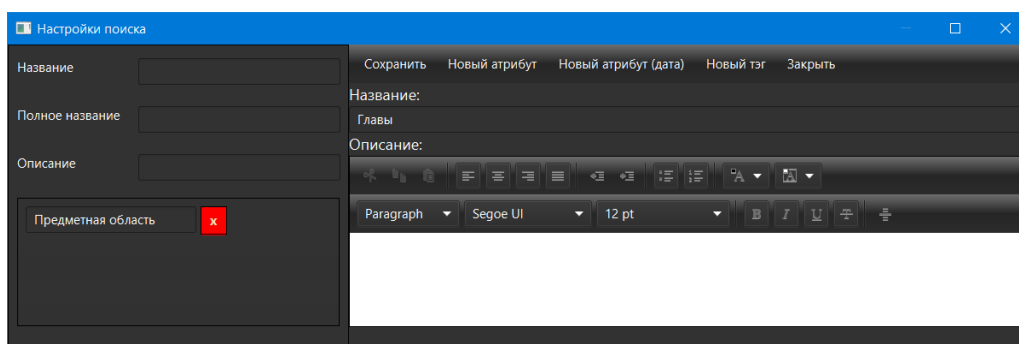


Рисунок 36 – Окно «Настройка поиска» для операции «Добавить автопоиск»

Операция «Добавить группу» даёт возможность объединять директории для автопоиска. Вызвать данную операцию можно из меню. Общий вид операции «Добавить группу» показан на рисунке 37.

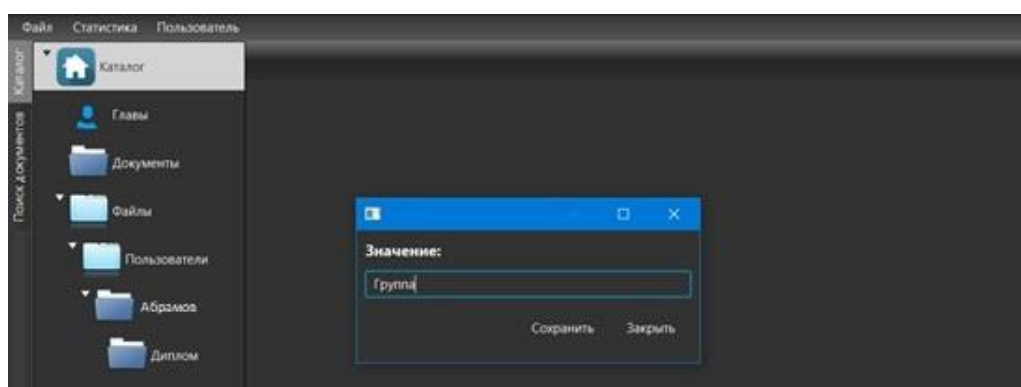


Рисунок 37 – Общий вид работы операции «Добавить группу»

Открытие окна «Поиск документов» позволяет пользователю производить поиск по следующим параметрам: тэгам, краткому названию,

полному названию, описанию. Общий вид окна «Поиск документов» представлен на рисунке 38.

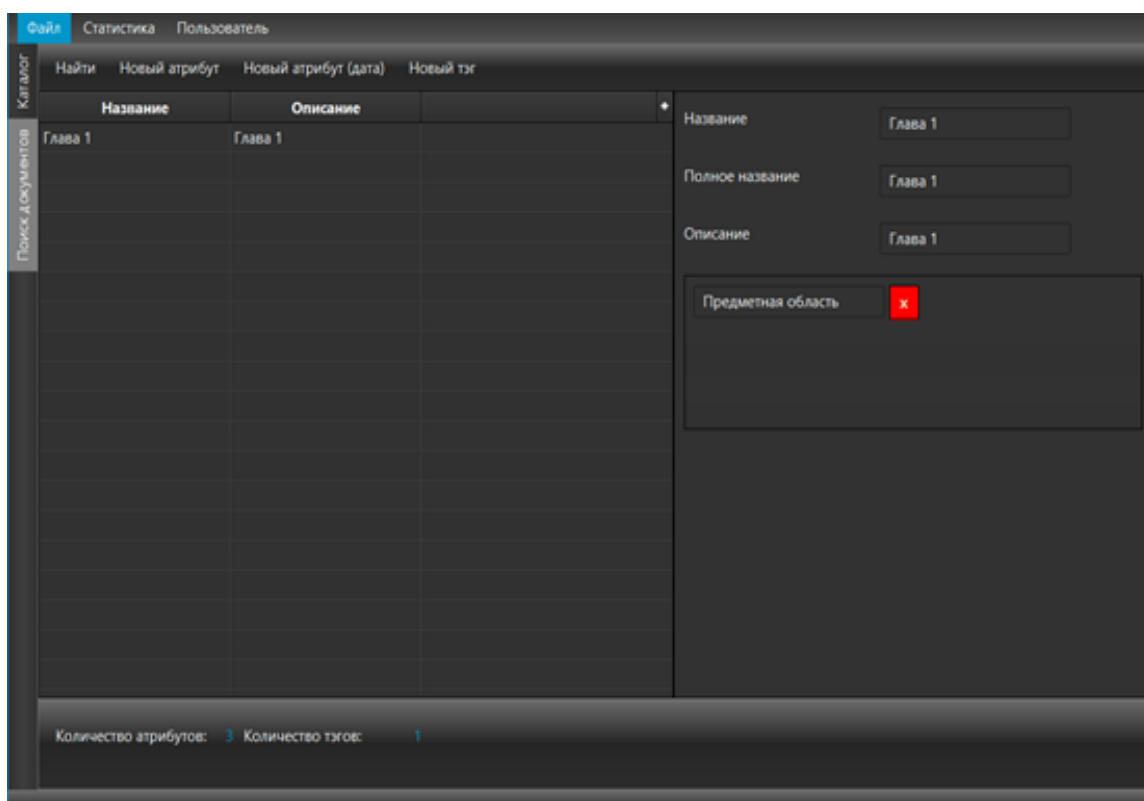


Рисунок 38 – Общий вид окна «Поиск документов»

В данном окне пользователь может задать столько атрибутов поиска в том числе тэгов и атрибутов сколько ему нужно для достижения результата поиска файлов.

3.6 Оценка и обоснование экономической эффективности разработки облачного хранилища

Для оценки рентабельности и целесообразности внедрения облачного хранилища могут использоваться типовые методики [4]. Для принятия решения относительно экономической эффективности предлагается воспользоваться методом сравнения двух вариантов реализации: разрабатываемое облачное хранилище и уже используемое в организации решение.

Экономическая эффективность складывается из двух составляющих:

- косвенного эффекта, который в общем случае характеризуется увеличением прибыли, привлечением большего числа клиентов, снижением уровня брака в производстве, снижением затрат на сырье и материалы, уменьшением сумм штрафов, неустоек и т.д.;
- прямого эффекта, который характеризуется снижением трудовых и стоимостных показателей.

В данном случае будет оцениваться только прямой эффект от разработки, внедрения и последующей эксплуатации облачного хранилища.

К трудовым показателям прямого эффекта относятся следующие:

- абсолютное снижение трудовых затрат (ΔT) в часах за год (1):

$$\Delta T = T_{\text{баз}} - T_{\text{цел}}, \quad (1)$$

где $T_{\text{баз}}$ - трудовые затраты за год на обработку информации по базовому варианту, ч.;

$T_{\text{цел}}$ - трудовые затраты за год на обработку информации по целевому варианту, ч.;

- коэффициент относительного снижения трудовых затрат ($K_{ТЗ}$) (2):

$$K_{ТЗ} = \Delta T / T_{\text{баз}} * 100\% \quad (2)$$

- индекс снижения трудовых затрат или повышение производительности труда ($Y_{ТЗ}$) (3):

$$Y_{ТЗ} = T_{\text{баз}} / T_{\text{цел}} \quad (3)$$

К стоимостным показателям прямого эффекта относятся: абсолютное снижение стоимостных затрат (ΔC) в рублях за год, коэффициент

относительного снижения стоимостных затрат (K_{C3}) индекс снижения стоимостных затрат (Y_{C3}), рассчитываемые аналогично.

Помимо рассмотренных показателей рассчитаем также срок окупаемости затрат на внедрение облачного хранилища ($T_{окуп}$), рассчитываемые в годах (4):

$$T_{окуп} = K_{Ox} / \Delta C, \quad (4)$$

где K_{Ox} - затраты на создание, внедрение и обслуживание облачного хранилища.

Затраты на создание, проектирование и внедрение рассчитаем по формуле (5):

$$K_{ox} = Z_{разр} + Z_{внедр} + Z_{обсл}, \quad (5)$$

где: $Z_{разр}$ – затраты на разработку;

$Z_{внедр}$ – затраты на внедрение;

$Z_{обсл}$ – затраты на обслуживание.

Затраты на разработку будем считать равными затратам на заработную плату руководителя проекта и программиста.

Затраты на обслуживание будем считать по формуле (6):

$$Z_{обсл} = O_{аморт} + Z_{ээ}, \quad (6)$$

где: $Z_{ээ}$ – затраты на электроэнергию для ПЭВМ, руб.;

$O_{аморт}$ – амортизационные отчисления, руб.

Расчет затрат на обработку информации по базовому и проектному варианту представлен в таблице 16.

Таблица 16 – Расчет стоимостных затрат по базовому / проектному варианту

Наименование операций технологического процесса	Оборудование	Ед. Изм.	Объем работ в год	Норма выработки / производительность ЭВМ (опер/в час.)	Трудоемкость	Средне-часовая зарплата специалиста (руб.)	Часовая норма амортизации (руб. за час) / ст. 1 маш. часа (руб.)	Часовая стоимость накладных расходов (руб.)	Стоимостные затраты для ручных операций	Стоимостные затраты операций, вып. на ЭВМ
Базовый вариант										
Сбор информации	Компьютер	шт	96	0,125	768	300	22	30	-	270 336
Сортировка информации	Компьютер	шт	48	0,25	192	300	22	30	-	62 400
Хранение информации	Компьютер	шт	9720	1	9720	300	22	30	-	3 421 440
Передача информации	Компьютер	шт	136	0,016	2,27	300	22	30	-	799,04
ИТОГО					10693,62					3758970,24
Проектный вариант										
Сбор информации	Компьютер	шт	96	1	96	300	22	30	-	33792
Сортировка информации	Компьютер	шт	48	2	24	300	22	30	-	8448
Хранение информации	Компьютер	шт	9720	4	2430	300	22	30	-	855360
Передача информации	Компьютер	шт	136	0,016	2,27	300	22	30	-	799,04
ИТОГО					2563,62					902394,24

По формуле (1) рассчитаем абсолютное снижение трудовых затрат в часах за год: $\Delta T = T_{\text{баз}} - T_{\text{цел}} = 10693,62 - 2563,62 = 8130$

По формуле (2) рассчитаем коэффициент относительного снижения трудовых затрат ($K_{TЗ}$): $K_{TЗ} = \Delta T / T_{\text{баз}} * 100 = 8130 / 10693,62 * 100 = 76,02$

Индекс снижения трудовых затрат или повышение производительности труда ($Y_{TЗ}$) рассчитаем по формуле (3): $Y_{TЗ} = T_{\text{баз}} / T_{\text{цел}} = 10693,62 / 2563,62 = 4,17$

Таким образом, производительность труда повысится более чем в 4 раза.

Аналогично рассчитаем стоимостные показатели. Абсолютное снижение стоимостных затрат в рублях за год: $\Delta C = C_{\text{баз}} - C_{\text{цел}} = 3758970,24 - 902394,24 = 2856576$

Коэффициент относительного снижения стоимостных затрат: $K_{CЗ} = \Delta C / C_{\text{баз}} * 100 = 2856576 / 3758970,24 * 100 = 75,99$

Индекс снижения стоимостных затрат: $Y_{CЗ} = C_{\text{баз}} / C_{\text{цел}} = 3758970,24 / 902394,24 = 4,17$

Таким образом, мы видим снижение стоимостных затрат также имеет место, что, в совокупности с повышением производительности труда, подтверждает экономическую целесообразность разработки и внедрения облачного хранилища.

Дополнительно рассчитаем срок окупаемости затрат на внедрение облачного хранилища ($T_{\text{окуп}}$). Для этого сначала по формуле (6) рассчитаем затраты на обслуживание.

Локальный сервер организации работает 8 часов в день. Количество рабочих дней в году 264. Стоимость электроэнергии 5.1 руб. за киловатт. Мощность, потребляемая локальным сервером 700 Вт. Годовые затраты на электроэнергию, потребляемую локальным сервером, руб.;

$$Z_{\text{эз}} = (8 * 264) * 5,1 * 0,700 = 7539,84 \text{ (руб.)}$$

Срок службы локального сервера составляет 7 лет. В этом случае амортизационные отчисления за год составляют:

$O_{\text{аморт}} = (35000 + 37000) / 7 = 10876$ (руб.), где $Z_{\text{ээ}}$ – затраты на электроэнергию для локального сервера, руб.; $O_{\text{аморт}}$ – амортизационные отчисления, руб.;

Тогда: $Z_{\text{обсл}} = O_{\text{аморт}} + Z_{\text{ээ}} = 7539,84 + 10876 = 17825,55$ (руб.).

Затраты на разработку облачного хранилища будем считать равными затратам на заработную плату программиста. Также здесь учитываются отчисления во внебюджетные фонды, величина которых составляет 30%.

Расчет затрат фонда оплаты труда представлен в таблице 17.

Таблица 17 – Фонд оплаты труда

Исполнители	Оклад, руб./мес	Трудоём– кость, дн.	Основная зарплата, руб
Разработчик (программист)	60000	22	60000
Отчисления в бюджетные и внебюджетные фонды 30%:		-	18300
Итого с отчислениями во внебюджетные фонды:		-	78300

Так же в затраты на внедрение облачного хранилища будут входить затраты на написание документации, фонд оплаты труда службы внедрения и сопровождения данного программного продукта, затраты на проведения обучения пользователей облачным хранилищем. Поскольку размер этих затрат зависит от многих факторов, то можно оценить только примерные затраты, и предположить, что данные затраты будут равны 10% от стоимости разработки: $Z_{\text{внедр}} = 78300 * 0,1 = 7830$ (руб.)

Таким образом, подставляя вычисленные значения затрат в формулу (5), получим стоимость создания, проектирования и внедрения облачного хранилища: $K_{\text{ох}} = 78300 + 7830 + 17825,55 = 103955,55$ (руб.).

На рисунке 39 представлена круговая диаграмма затрат на разработку, внедрение и обслуживание облачного хранилища.

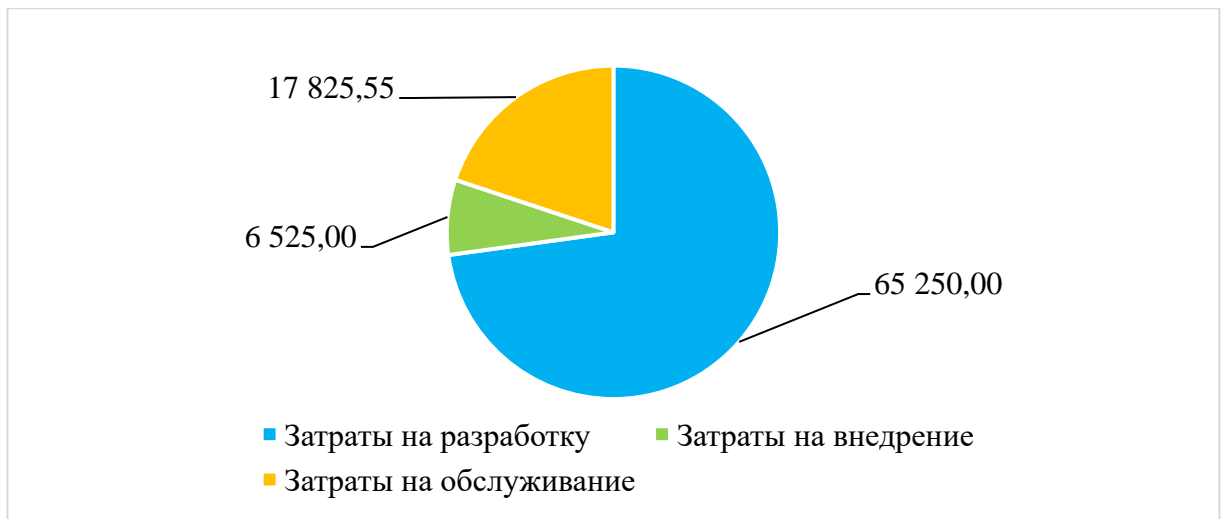


Рисунок 39 - Затраты на разработку, внедрение и обслуживание

Последний расчёт показывает срок окупаемости разработки и внедрение облачного хранилища ($T_{\text{окуп}}$), рассчитываемые в годах по формуле (4):

$$T_{\text{окуп}} = 103955,55 / 2856576 = 0,36 \text{ (лет)}$$

Расчёты выше показывают разработка и внедрение облачного хранилища окупятся в течение пяти месяцев.

Из этого следует, что разработка и внедрение облачного хранилища показывает экономическую эффективность.

Заключение

Результатом выполнения выпускной квалификационной работы стало разработанное, протестированное и запущенное облачное хранилище данных.

В ходе работы была проанализирована деятельность производственно-технического отдела в ОАО «Концерн «АБС».

Также был проанализированы текущие решения внутри ОАО «Концерн «АБС». Было выявлен ряд проблем, с которыми сталкиваются сотрудники.

В ходе работы над проектом были выполнены следующие задачи:

- изучены и выбраны технологии и инструменты для реализации проекта;
- создан веб-клиент на основе Web фреймворка Django;
- реализован пользовательский веб-интерфейс с помощью фреймворка Vue.js и CSS-фреймворка Vuetify.js.

Созданное облачное хранилище имеет следующие преимущества:

- расширяемость системы. Сервисы хранилища можно в любой момент запустить на другом сервере, и он сразу же включится в работу;
- синхронизация между серверами-хранилищами. При любой ситуации, когда сервер-хранилище выходит из строя, файлы пользователя будут ему доступны, так как хранятся на всех серверах одновременно;
- подключение к любой базе данных. Возможность использования любой SQL базы данных, так как используется Django ORM.

В ближайшее время для развития разработанного приложения предполагается:

- разбиение логики сервисов-хранилищ и сервиса управления пользователями на микросервисы для более гибкой настройки;
- расширение функциональности облачного хранилища за счет сбора обратной связи;
- доработка API для более гибкого обращения к нему.

Список используемой литературы

1. Архитектура клиент-сервер [Электронный ресурс] – режим доступа: [<http://portal.tpu.ru/SHARED/f/FAS/study/avis/lectures/cli-se.pdf>]
2. Бейтс, Марк CoffeeScript. Второе дыхание JavaScript / Марк Бейтс. – М.: ДМК Пресс, 2016. – 310 с.
3. Бритов Г., Осипова Т. Моделирование бизнес-процессов – М.: LAP, 2014 –124 с.
4. Варфоломеева Е.В. Информационные системы в экономике: Учебное пособие / Е.В. Варфоломеева, Т.В. Воропаева и др.; Под ред. Д.В. Чистова - М.: НИЦ ИНФРА-М, 2015. - 234 с.
5. Васильев, А. Н. Python на примерах. Практический курс по программированию / А.Н. Васильев. - М.: Наука и техника, 2016. - 432 с.
6. Гвоздева В.А. Базовые и прикладные информационные технологии: Учебник / Гвоздева В. А. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2015. - 384 с.
7. Гвоздева В.А. Информатика, автоматизированные информационные технологии и системы: Учебник / В.А. Гвоздева. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2015. - 544 с.
8. Головатый, А. Django. Подробное руководство / А. Головатый. - М.: Символ-плюс, 2014. - 320 с.
9. ГОСТ Р 50922-2006. Защита информации. Основные термины и определения
10. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств
11. Дронов В. JavaScript в Web-дизайне / В. Дронов. - М.: БХВ-Петербург, 2016. - 880 с.
12. Иванов А.В., Келасьев А.С. Исследование индексных структур в реляционных базах данных // Математические методы в технике и

технологиях – ММТТ-29: сб. трудов XXIX Междунар. науч. конф.: в 12 т. Т.4. / под общ. ред. А.А. Большакова. – Саратов: Саратов. гос. техн. ун-т; Санкт-Петербург: СПбГТИ(ТУ), СПбПУ, СПИИРАН; Самара: Самарск. гос. техн. ун-т, 2016,123-125 с.

13. Исаев Г. Проектирование информационных систем. Учебное пособие. - М.: Омега-Л, 2015. - 432с.

14. Маклаков С.В. BPwin ERwin CASE – средства разработки информационных систем. М.: Диалог МИФИ, 2011. 304 с.

15. Моделирование на UML – Онлайн-проект [Электронный ресурс] – режим доступа: [<http://book.uml3.ru>]

16. О модели взаимодействия клиент-сервер простыми словами. Архитектура «клиент-сервер» с примерами | Блог о создании сайтов, продвижение сайтов, заработке в интернете, веб-технологиях, базах данных и СУБД. [Электронный ресурс] – режим доступа: [<http://zametkinapolyah.ru/servera-i-protokoly/o-modeli-vzaimodejstviya-klient-server-prostymi-slovami-arxitektura-klient-server-s-primerami.html>] от 09.05.2022.

17. Официальный сайт [Электронный ресурс] – режим доступа: GitFlic <https://gitflic.ru/>

18. Пирогов, В. Ю. Информационные системы и базы данных. Организация и проектирование / В.Ю. Пирогов – СПб.: БХВ-Петербург, 2009. – 590 с.

19. Применения нотаций IDEF0 и IDEF3 для моделирования бизнес-процессов [Электронный ресурс] – режим доступа: [<https://bpmssoft.org/idef0-and-idef3/>] от 09.05.2022.

20. Система электронного документооборота LanDocs [Электронный ресурс] – режим доступа: <https://landocs.ru/docflow>

21. Сравнительный анализ реляционных систем управления базами данных – статья [Электронный ресурс] – режим доступа:

[<http://docplayer.ru/42646091-Sravnitelnyy-analiz-relyacionnyh-sistem-upravleniya-bazami-dannyh.html>]

22. Томсан Бил. Storm warning for cloud computing». [Электронный ресурс] – режим доступа: [<http://technorati.com/posts/lv3vwRkNoZ4b0%3D?reactions> «news.bbc.co.uk/2/hi/technology/7421099.stm/]

23. Шаймарданов Р.Б. Моделирование и автоматизация проектирования структур баз данных - М.: Юнити, 2016. - 469 с.

24. Cloud Storage Architecture: Issues, Challenges and Opportunities / International Journal of Innovative Research in Computer Science & Technology (IJIRCST) ISSN: 2347-5552, Volume-9, Issue-3, May 2021 <https://doi.org/10.21276/ijircst.2021.9.3.12> Article ID IRP1164, Pages 70-73

25. Django documentation [Электронный ресурс] – режим доступа: [<https://docs.djangoproject.com/>]

26. Google Drive [Электронный ресурс] – режим доступа: [<https://drive.google.com/drive/>]

27. MariaDB Server: The open source relational database [Электронный ресурс] – режим доступа: [<https://mariadb.org/>]

28. Python. Programming language [Электронный ресурс] – режим доступа: [<https://www.python.org/>]

29. R. Arokia Paul Rajan, S. Shanmugapriyaa / Evolution of Cloud Storage as Cloud Computing Infrastructure Service ISSN : 2278-0661 Volume 1, Issue 1 (May-June 2012), PP 38-45 [Электронный ресурс] – режим доступа: <https://www.iosrjournals.org/>