**Содержание**

[Введение 2](#_Toc143616149)

[Цель и задачи 6](#_Toc143616150)

[Средства реализации 7](#_Toc143616151)

[Постановка задачи на проектирование 9](#_Toc143616152)

[Реализация 14](#_Toc143616153)

[Тестирование 20](#_Toc143616154)

[Выводы 22](#_Toc143616155)

[Список литературы 23](#_Toc143616156)

# **Введение**

Тема практической работы – веб-приложение (портал) для хранения и распространения файлов, сборок приложений, архивов, документаций. Актуальность: Технологический прогресс задал скорость и вектор на сближение с глобальными изменениями во всех сферах жизнедеятельности общества, процесс цифровизации и внедрения информационных технологий происходит повсеместно. Это говорит о том, что теперь практически каждый человек должен иметь базовое понимание об информационных технологиях, методах и инструментах хранения и распространения данных [1].

У каждого из нас имеются данные, которые находятся в электронном виде, поэтому очевидна проблема хранения и передачи информации. Одним из вариантов решения данной задачи является разработка специальных средств для защищённого расположения конфиденциальной информации в хранилищах, которые включают в себя возможность безопасно обмениваться данными между локальными устройствами пользователей и порталом распространения [2].

В наши дни в большинстве случаев используются системы удаленного хранения данных (облачные хранилища), что обусловлено определенными преимуществами, которые получают пользователи, работающие с данным сервисом. Облачное хранилище данных – виртуальный носитель информации, в котором данные располагаются и обрабатываются на серверах [3]. Основным преимуществом данной технологии является не только конфиденциальность и надёжность, но и возможность совместной работы с файлами и наличие удалённого доступа из любой точки мира с устройства, имеющего доступ к сети Интернет [4-6].

В наши дни применение корпоративных сетевых хранилищ для обмена файлами, используемыми во внутренних бизнес-процессах, фактически является стандартом защиты информации предприятия. Связано это с тем, что практически любой бизнес-процесс предусматривает использование различного типа файлов и не только [7]. Действительно, подобный сервис помогает организациям управлять документацией и автоматизировать хранение и обмен данных, значительно упрощая работу сотрудникам.

В качестве хранимых данных (в зависимости от вида деятельности предприятия) могут быть: файлы проектной и рабочей документации, сборки приложений, архивы, офисная документация и т.д. Таким образом, специализированные системы обмена данными привносят ряд особенно важных методов сохранения уникальности и доступности файлов среди доверенного круга лиц: резервное копирование и восстановление данных, возможность гибкой настройки прав доступа к хранимым данным, а также различные средства интеграции в информационную сеть предприятия [8].

Несмотря на вышеуказанные преимущества, данный способ распространения информации имеет ряд определённых проблем, например, риски снижения уровня компьютерной безопасности, которые связаны с утратой, хищением и изменением персональных и/или служебных данных [9].

По статистике, наиболее распространённой угрозой для информационных систем, в которых хранятся данные, являются несанкционированные попытки доступа (НСД) к ним. Обеспечение безопасности информации означает внедрение комплексных алгоритмов защиты доступа к хранимой информации с целью устранения (снижения) риска её потери в результате НСД [10]. Ключевыми процессами защиты доступа к данным являются идентификация и аутентификация [11].   
На официальной странице всем известной антивирусной программы «Антивирус Касперского» предлагается следующая формулировка данных процессов:

Идентификация – это процедура, в результате выполнения которой для субъекта идентификации выявляется его идентификатор, однозначно определяющий этого субъекта в информационной системе.

Аутентификация – процедура проверки подлинности, например, проверка подлинности пользователя путём срабатывания введённого им пароля с паролем, сохранённым в базе данных (БД).

Авторизация – предоставление определённому лицу или группе лиц прав на выполнение определённых действий в рамках информационной системы.

В соответствии с этим в нашем веб-сервисе предусмотрены процедуры аутентификации и разграничения доступа, что является обеспечением информационной безопасности.

Как мы определили выше, хранение данных на внешнем контуре может не соответствовать внутренней корпоративной политике ряда организаций, а, если организация хранит личные данные пользователей, то данное решение будет противоречить законодательству Российской Федерации [11]. Было рассмотрено альтернативное решение - разработка вспомогательной информационной системы (сервиса), позволяющей централизованно взаимодействовать с различными уровнями пользователей и их данными.

Убедившись в необходимости создания корпоративного сетевого хранилища и вспомогательной информационной системы (сервиса), позволяющей взаимодействовать с хранилищем удалённо, рассмотрим далее основные компоненты предлагаемого решения и их взаимодействие. Основополагающим компонентом информационной системы является классический REST API, который предоставляет необходимые HTTP-методы для управления данными в хранилищах. Также данный компонент выполняет функции маршрутизации запросов к соответствующим хранилищам, используя их основные протоколы (FTP, SMB и т.д.). Запрос к API может иметь тип application/json или multipart/form-data (для больших файлов), при этом непосредственно в хранилища передаются только данные файла (payload), а метаданные (metadata), находящиеся в других полях передаваемого json сохраняются в БД. Метаданные могут содержать точный адрес сохраняемого файла (с указанием типа и адреса хранилища), формат, теги, дополнительное описание [12].

Наше веб-приложение создается с целью организации безопасного и автоматизированного процесса распространения файлов. С помощью данного приложения можно будет не только хранить данные, но и делиться ими с пользователями, чей уровень доступа соответствует, при помощи ссылок, переходя по которым, будет возможность загрузить, к примеру, приложение, и установить его у себя на устройстве для дальнейшего использования.

# **Цель и задачи**

Цель практической работы – создание веб-приложения для хранения файлов распространяемого контента. В нашем приложении доступ к данным организован через пространства, поделённые на разделы. Также приложение включает множество страниц, например, страница гостя, администратора пространств либо администратора. В связи с этим, разработка единого приложения была разделена между отдельными членами команды по страницам. В частности, моей задачей была реализация пользовательского интерфейса страницы администратора пространств, которая включает вкладки: редактирование пространства, раздела и версии. Доступ к каждой вкладкой осуществляется с помощью перехода по дереву.

Задачи производственной практики:

1. Изучение процессов разработки в крупной компании, в том числе: работа в команде, постановка задач на проектирование и разработку, обсуждение этапов разработки и документирование важных элементов системы, ввод системы в эксплуатацию
2. Проектирование пользовательского интерфейса в онлайн-редакторе Figma
3. Вёрстка итогового пользовательского интерфейса (страницы администратора пространств)
4. Изучение библиотеки React.js для реализации веб-приложения
5. Изучение библиотеки Zustand для управления состояниями
6. Изучение архитектурного стиля REST API для интеграции пользовательского интерфейса со внутренними данными портала
7. Изучение инструмента Swagger для описания возможностей API
8. Работа с Git для контроля версий кода и удобной работы над общими задачами с другими разработчиками
9. Тестирование работы реализованного функционала, исправление выявленных ошибок.

# **Средства реализации**

В качестве средства реализации пользовательского интерфейса веб-приложения была выбрала библиотека React.JS.

React – это библиотека для создания пользовательских интерфейсов. Одной из ее отличительных особенностей является возможность использовать JSX - язык программирования с близким к HTML синтаксисом, который компилируется в JavaScript. Разработчики могут добиваться высокой производительности приложений с помощью Virtual DOM. C React можно создавать изоморфные приложения. Virtual DOM может повысить производительность высоконагруженных приложений, что может снизить вероятность возникновения возможных неудобств и улучшает пользовательский опыт. Использование изоморфного подхода помогает производить рендеринг страниц быстрее, тем самым позволяя пользователям чувствовать себя более комфортно во время работы с приложением. Поисковые системы индексируют такие страницы лучше, поскольку один и тот же код может быть использован как в клиентской, так и в серверной части приложения, нет необходимости в дублировании одного и того же функционала [13-15].

Zustand — это библиотека управления состоянием с открытым исходным кодом, разработанная создателями Jotai и React-spring (лучшая библиотека анимации React). На данный момент это одна из самых легких библиотек управления состоянием, ее размер составляет 1,5 КБ. Несмотря на то, что она легкая, она решает некоторые важные проблемы, такие как:

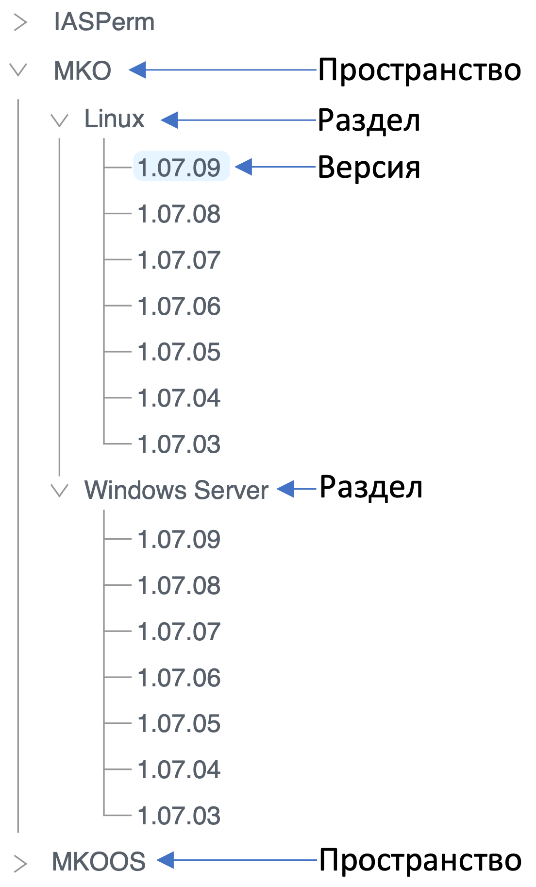
* Zombie Children
* Параллельность в React
* Потеря контекста

Zustand использует внешнее хранилище и предоставляет несколько хуков для его подключения.

# **Постановка задачи на проектирование**

В процессе определения целей, задач и средств реализации, были описаны все ключевые элементы интерфейса и логика работы при нажатии или альтернативном взаимодействии пользователя с ним. Перед разработкой приложения необходимо определить его функционал.

Базовым объектом для приложения является **пространство** для распространения файлов (или просто пространство). Каждое пространство разделено на **разделы**. Каждый раздел, разделен на **версии**. Каждая версия содержит группу распространяемых файлов.

  
Рис. 1. Дерево пространств с разделами и версиями

Каждое пространство группирует разделы и хранит следующую информацию:

* описание пространства
* иллюстрации пространства
* управляет общим доступом к пространству (логин и пароль).

Пользователь, зашедший в определенное пространство, видит все разделы и версии файлов. Каждый раздел группирует версии.

Кроме группы распространяемых файлов каждая версия содержит:

* описание
* иллюстрации
* дату публикации
* идентификатор (используется в iOS)

Приложение работает с четырьмя типами пользователей:

* гости – люди, скачивающие контент
* администраторы пространств - управляют контентом
* администраторы – управляют пользователями и контентом
* root - отдельный пользователь, тот кто управляет вообще всем: настройками приложения, пользователями (в том числе и администраторами) и контентом.

Поскольку в мою область обязанностей входит страница администратора пространств, опишу постановку задач именно на её проектирование.

Предполагается, что портал будет распространять множество различных файлов и приложений, соответственно тематических пространств будет много. Каждое такое пространство должно иметь своего администратора, который будет управлять их содержимым. Администратор может вести несколько пространств, поэтому за каждым администратором закрепляется некоторое множество пространств. Доступ к каждому пространству задает администратор.

Администраторы пространств — это основные администраторы портала, управляющие всем контентом, который видят пользователи.   
Такой администратор может:

* добавлять новые версии файлов, удалять их
* управлять списком файлов на скачивание
* задавать комментарии к файлам и определять их тэг (отражается на сопровождающей файл иконке).

Именно администратор пространства должен, например, взять файл приложения от разработчика и разместить его на портале. То же самое и с документацией. Администратор пространства может добавлять новые пространства и удалять их, если это разрешено администратором.   
Может добавлять и удалять новые разделы и версии.

[**Работа с новым пространством**](https://conf.parma.ru/pages/viewpage.action?pageId=223049255#id-%D0%9E%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F-%D0%A0%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%D1%81%D0%BD%D0%BE%D0%B2%D1%8B%D0%BC%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE%D0%BC.)

Панель администратора пространства разделена на две части: левую и правую. Слева администратор видит список пространств, с которыми он может работать. Справа, функциональная панель, которая определяется в зависимости от того, с чем работает администратор.

Для добавления и удаления нового пространства администратор должен нажать кнопку «Добавить» / «Удалить» в левой панели с пространствами.   
Для удаления, предварительно, требуется выбрать соответствующее пространство в левой панели. Обязательно нужно переспросить пользователя, для подтверждения удаления.

Когда пользователь выбирает пространство слева, справа, появляется предварительно заполненная панель для редактирования пространства.   
Если создается новое пространство, то панель для редактирования пространства заполняется значениями по умолчанию.

При определении пространства пользователь может задать следующие параметры:

* иконка пространства
* имя пространство – должно быть уникальным
* описание пространства
* логин и пароль – можно не задавать
* разделы в пространстве
* иллюстрации
* копирайт-подпись
* файл лицензионного соглашения.

Для файлов с иллюстрациями требуется показать preview для выбранного в списке файла.

[**Работа с новым разделом**](https://conf.parma.ru/pages/viewpage.action?pageId=223049255#id-%D0%9E%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F-%D0%A0%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%D1%81%D0%BD%D0%BE%D0%B2%D1%8B%D0%BC%D1%80%D0%B0%D0%B7%D0%B4%D0%B5%D0%BB%D0%BE%D0%BC.)

При выборе одного из разделов слева, на правой панели следует вывести предварительно заполненную панель редактирования раздела. При создании нового раздела нужно заполнить раздел значениями по умолчанию.

В панели раздела пользователь может задать:

* наименование раздела
* список версий раздела
* определить публиковать ли раздел
* предоставлять ли файлы приложений напрямую или через манифест
* задать шаблон манифеста, при необходимости.

URL для пользователя создается автоматически и администратором пространства не задается. При выборе раздела и нажатии кнопки «Добавить» в левой панели, добавляется новый пустой раздел.

[**Работа с версией**](https://conf.parma.ru/pages/viewpage.action?pageId=223049255#id-%D0%9E%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F-%D0%A0%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%D1%81%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B5%D0%B9)

Основное содержание портала определяется в функциональной панели. На ней администратор задает файлы на скачивание и их описание. При выборе в левой части дерева листочка «версии» в правой функциональной панели появляется заполненная панель для редактирования версии. В этой панели отображается не редактируемая иконка пространства и не редактируемый URL пользовательской страницы. URL должен быть копируемый, чтобы администратор мог взять его со страницы и отправить пользователю.   
Ниже идет выпадающий список с разделами. Выбрав в данном меню другой раздел, описание версии «переедет» в указанный раздел. «Переезд» невозможен если там уже есть описание той же самой версии.

Наименование версии определяет то, как данная версия будет представлена пользователю. Это текстовая строка, поэтому предполагается, что администратор волен задавать версию на свое усмотрение. Никакого контроля делать не нужно, кроме проверки на существование той же самой версии. Также необходимо дать пользователю возможность управлять отображением версии, при необходимости спрятать её от публикации.

Дата по умолчанию выставляется текущая, но в окне выбора даты можно дать администратору выбрать любую дату. Проверок в этом поле никаких делать не нужно.

Описание версии – текст в свободной форме, который скомпонуется с описанием пространства.

Иллюстрации — это список файлов с картинками или скриншотами с информацией о текущей версии контента. Администратор может добавить или удалить картинки. На странице гостя эти иллюстрации компонуются иллюстрациями пространства. Здесь необходимо предоставить администратору возможность preview. Также есть возможность спрятать иллюстрации на версию от пользователей.

В самом низу у нас идет список файлов которую мы предлагаем пользователям для скачивания. При добавлении файла мы можем определить его тэг и описание. Кнопки «Добавить» и «Удалить» позволяют управлять списком файлов. Все теги предварительно задаются администратором (не администратором пространства).

# **Реализация**

Первым этапом разработки пользовательского интерфейса является создание проволочной диаграммы – своеобразного рабочего чертежа, описывающего функциональные возможности и содержимое приложения.   
На основе её затем реализуется макет пользовательского интерфейса со стилевыми особенностями (дизайном).

Был определён следующий вид для страницы администратора:

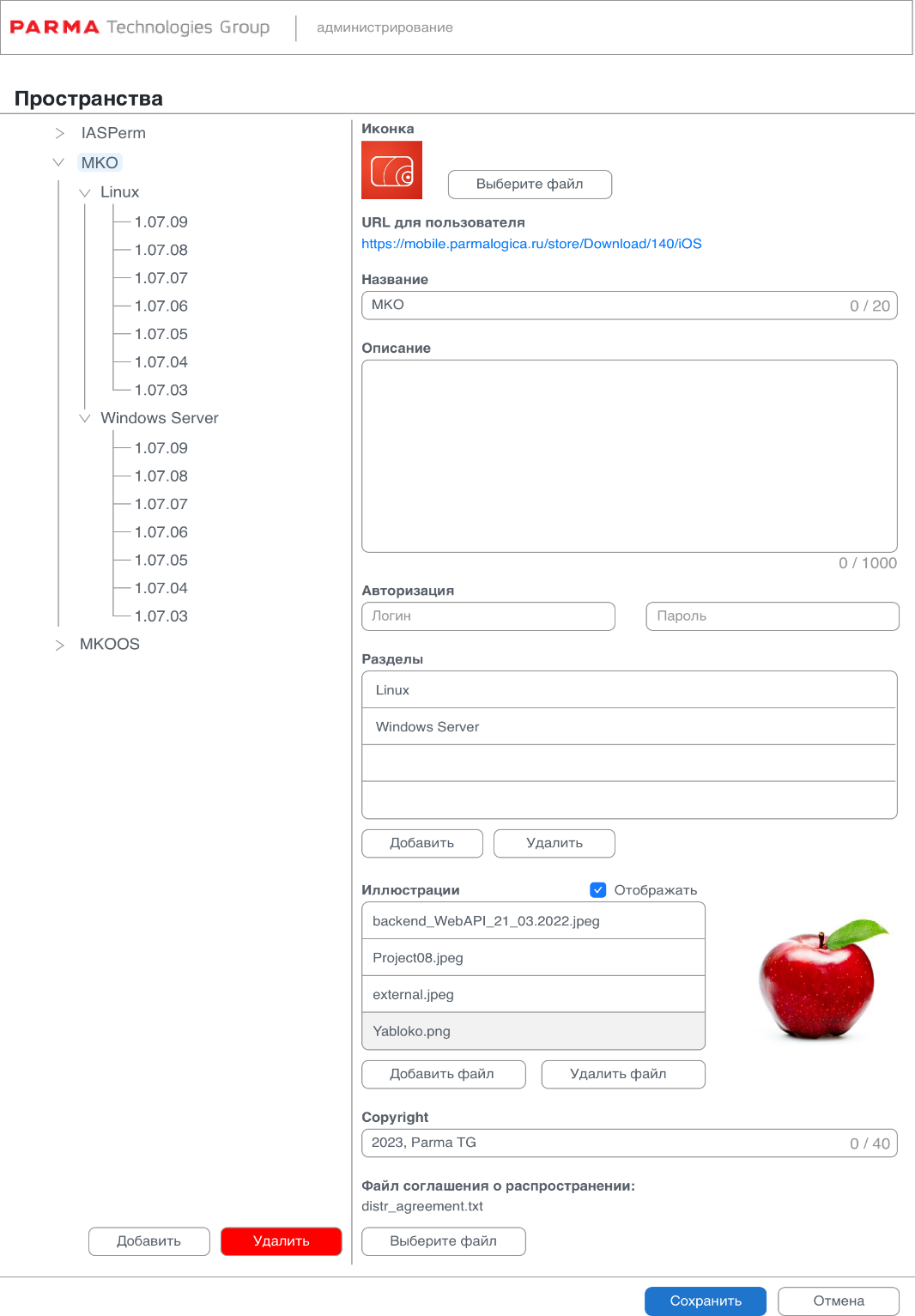


Рис. 2. Вкладка работы с пространством

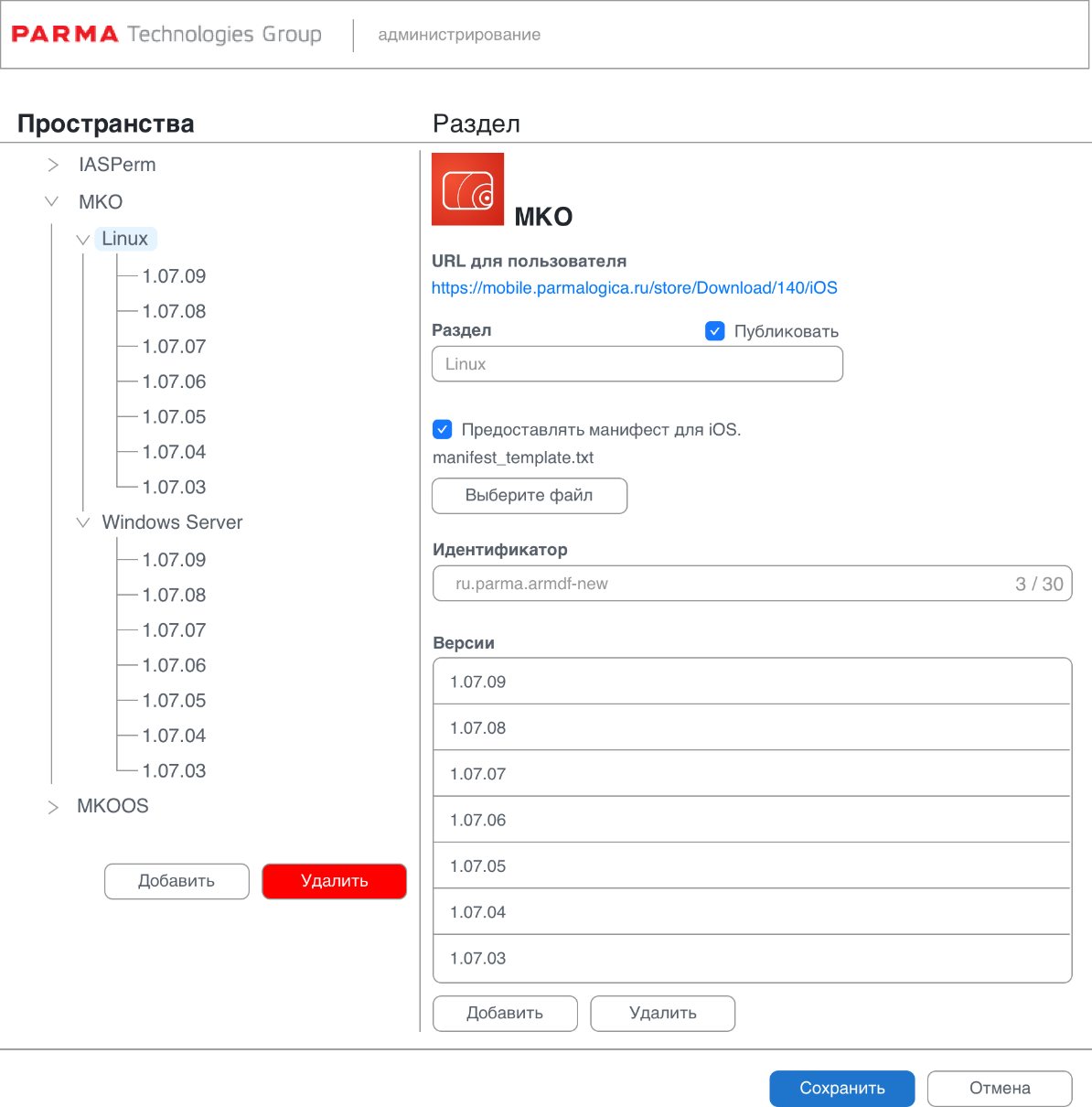


Рис. 3. Вкладка работы с разделом

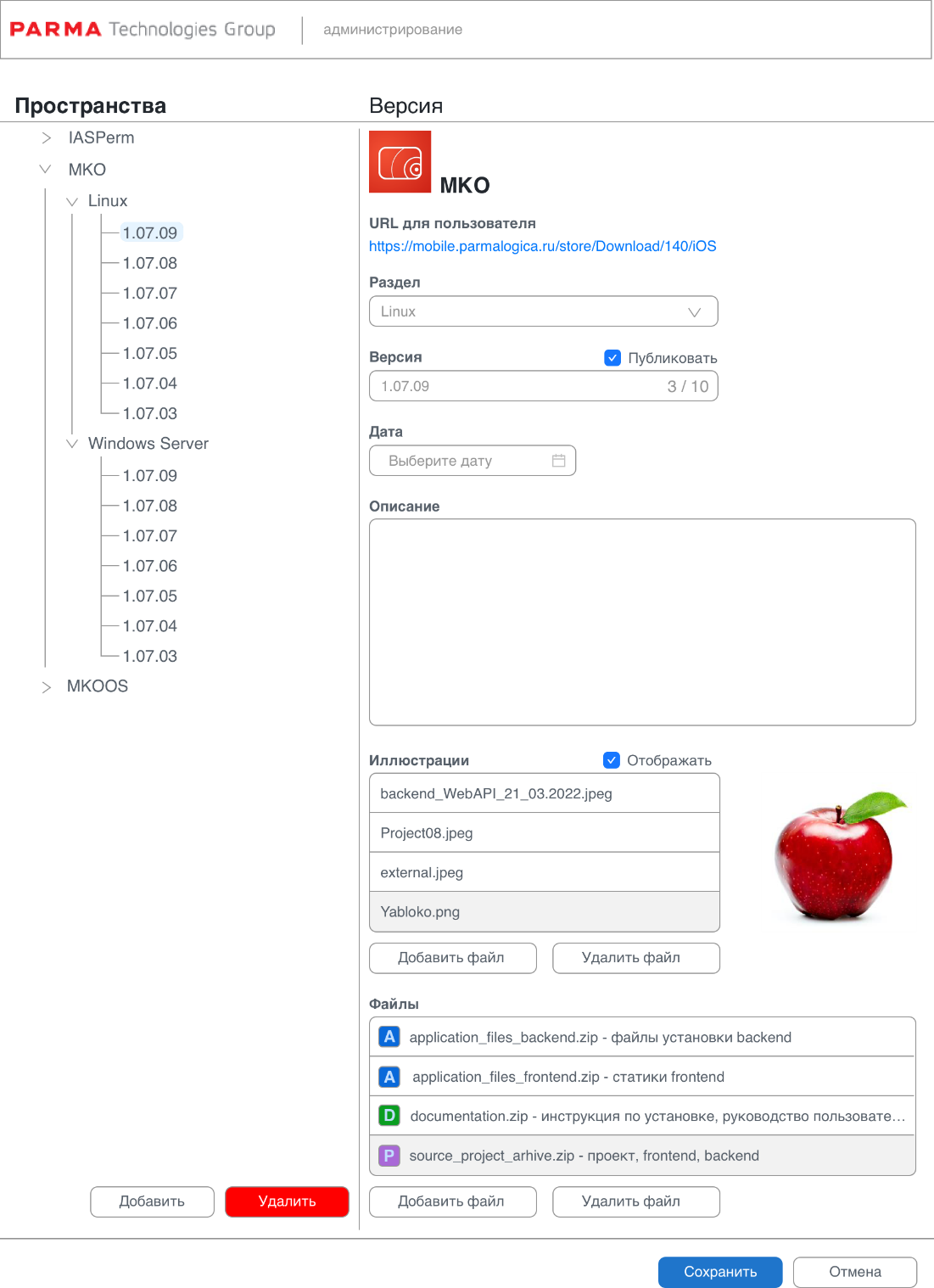


Рис. 4. Вкладка работы с версией

Следующим этапом разработки страницы является вёрстка на HTML и CSS, каждая вкладка выделяется как самостоятельный React-компонент.

Также были созданы отдельные компоненты для дерева, полей ввода, кнопок, списков и т.д. для удобства наполнения страницы контентом без дублирования участков кода.

Следующим этапом является интеграция с Backend посредством REST API-запросов. При взаимодействии пользователя с интерфейсом отправляется запрос с ожидаемыми параметрами ответа и вид HTTP-метода, к которым относятся:

* GET — получение ресурса
* POST — создание ресурса
* PUT — обновление ресурса
* DELETE — удаление ресурса

По виду метода определяется, что клиент хочет сделать с сущностью. Например, для методов создания вызывается метод контроллера с HTTP-методом POST.

# **Тестирование**

Было проведено ручное тестирование с помощью расширения в браузере, которое позволяет проводить отлаживание кода в реальном времени и видеть все изменения интерфейса. Запросы с данными и типом отслеживались в консоли браузера.

По итогу страница администратора пространств работает должным образом.

# **Выводы**

В ходе производственной практики были успешно применены теоретические и практические умения, полученные в ходе учебы, а также получены новые знания о работе в большой компании. Был приобретен полезный опыт по активной работе над интересным приложением, а также изучены процессы, происходящие во время работы над программным продуктом. Пройдены процессы разработки в крупной компании, в том числе: работа в команде, постановка задач на проектирование и разработку, обсуждение этапов разработки и документирование важных элементов системы, ввод системы в эксплуатацию.

Во время практики было разработано веб-приложение для хранения и распространения файлов, сборок приложений и архивов. При этом были выполнены все поставленные задачи, а именно:

1. Спроектирован пользовательский интерфейс (страницы администратора пёёространств) в онлайн-редакторе Figma и реализована его вёрстка.
2. Изучены библиотеки React.js и Zustand для создания веб-приложения и управления его состояниями. Также был реализован архитектурный стиль REST API для интеграции пользовательского интерфейса со внутренними данными портала и инструмент Swagger для описания возможностей API.
3. Проведено тестирование работы реализованного функционала, исправлены выявленные ошибки.

Ценный опыт, полученный во время производственной практики, поможет в дальнейшем не только проектировать и создавать серьёзные веб-приложения, но и работать в компании с осознанием работы различных бизнес-процессов.

# **Список литературы**

1. Такуадина, А. И. Информатика и её применение в современном ВУЗе / А. И. Такуадина, М. С. Абдыкалыкова, М. Р. Бикбавлеев // Технические науки - от теории к практике. – 2016. – № 10 (58). – С. 64-69. – EDN WXRGLF
2. Горев, А. И., Симаков, А. А. Обеспечение информационной безопасности: учеб. пособие. Омск: Изд-во ОмГУ, 2005. 80[1] с.
3. Котяшичев, И. А., Бырылова, Е.А. Защита информации в «Облачных технологиях» как предмет национальной безопасности // Молодой ученый. – 2015. – №64. – С. 30–34.
4. Шнайер Брюс. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си: монография. М.: Триумф, 2012. 816 c.
5. Афанасьев, Н. С. Исследование технологии облачного хранилища // Актуальные научные исследования в современном мире. – 2021г. – №7-2. – С. 91 – 96.
6. Денисенко, А. А. Исследование и разработка системы распознавания текста на изображении / А. А. Денисенко // Международный журнал прикладных и фундаментальных исследований. – 2020. – № 5. – С. 87-91. – EDN BGLRLB.
7. Алексеева, А. В., Юдинцев, Б. С. Информационная система для доступа к сетевым файловым хранилищам различного типа // Инновационные научные исследования. – 2023г. – №4-1 (28). – С. 95 – 107.
8. Круглик, Р. И. Создание веб-приложения с помощью библиотеки React.JS // Постулат. – 2019г. – №1-1 (39). – С. 125.
9. Аникин, И. В., Глова, В.И. Методы и средства защиты компьютерной информации: учеб. пособие. Казань: Изд-во Казанского государственного технического университета, 2008. 260 с.
10. Ходашинский, И. А., Савчук, М. В., Горбунов, И. В., Мещеряков, Р. В. Технология усиленной аутентификации пользователей информационных процессов // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2011г. – №2-3 (24). – С. 236 – 248.
11. КонсультантПлюс. Статья 1 [Электронный ресурс] – URL: <https://www.consultant.ru/document/cons_doc_LAW_165838/3d0cac60971a511280cbba229d9b6329c07731f7/> (дата обращения: 09.07.2023).
12. Мадатов, О. Я. Некоторые вопросы обеспечения информационной безопасности облачного хранения данных // Безопасность информационных технологий. – 2018г. – Т. 25. – №3 (24). – С. 45 – 52.
13. Введение в REST API – RESTful веб-сервисы [Электронный ресурс] – URL: https://habr.com/ru/articles/483202/ (дата обращения 20.06.2023).
14. Потовиченко, М. А., Шатилов, Ю. Ю. Разработка клиентской части одностраничного web-приложения с использованием библиотеки React // Научное обозрение. Технические науки. – 2020г. – №1. – С. 39 – 43.
15. Аргасова, В. В. Применение web-фреймворка React при разработке информационных систем // Аллея науки. – 2018г. – Т. 2. – №7 (23). – С. 526 – 528.