Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра «Информатика»

УТВЕРЖДАЮ
Заведующий кафедрой

_____ А. С. Кузнецов
подпись инициалы, фамилия

«_____» _____ 2021 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.04 Программная инженерия

Разработка клиентской части веб-сервиса «Электронный фотоальбом»

Руководитель ВКР	_____	доцент, канд. техн. наук	А. В. Хныкин
	подпись, дата	должность, ученая степень	инициалы, фамилия
Выпускник	_____		А. Н. Конюхова
	подпись, дата		инициалы, фамилия

Красноярск 2021

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра «Информатика»

УТВЕРЖДАЮ
Заведующий кафедрой

_____ А. С. Кузнецов
подпись инициалы, фамилия

«_____» _____ 2021 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Студенту Конюховой Анастасии Николаевне
фамилия, имя, отчество

Группа КИ17-16Б Направление (специальность) 09.03.04
номер код

Программная инженерия
наименование

Тема выпускной квалификационной работы

Разработка клиентской части веб-сервиса «Электронный фотоальбом»

Утверждена приказом по университету № 5308/с от 19.04.2021

Руководитель ВКР А. В. Хныкин, доцент, канд. техн. наук, ИКИТ СФУ
инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР описание предметной области

Перечень разделов ВКР введение, анализ предметной области, планирование, управление и проектирование, разработка клиентской части, описание работы программного продукта, заключение, список использованных источников

Перечень графического материала Презентация

Руководитель ВКР _____ А. В. Хныкин
подпись инициалы и фамилия

Задание принял к исполнению _____ А. Н. Конюхова
подпись инициалы и фамилия студента

« ___ » _____ 2021 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка клиентской части веб-сервиса «Электронный фотоальбом»» содержит 46 страниц текстового документа, 26 использованных источников, 35 иллюстраций, 7 таблиц.

ВЕБ-СЕРВИС, КЛИЕНТСКАЯ СТОРОНА, ЦИФРОВИЗАЦИЯ, ФОТОГРАФИЯ, ФОТОАЛЬБОМЫ, УПРАВЛЕНИЕ ПРОЕКТОМ, МЕТОДОЛОГИЯ РАЗРАБОТКИ.

Целью данной выпускной квалификационной работы является разработка клиентской части веб-сервиса «Электронный фотоальбом». Разработанный программный продукт нацелен на широкий круг потребителей за счет понятной структуры и лаконичного дизайна. Он позволяет загружать, хранить, просматривать и добавлять в архив свои цифровые фотографии.

Для достижения поставленной цели были решены следующие задачи:

– проанализированы существующие решения;
– сформированы и задокументированы требования к системе;
– определен технологический стек проекта;
– спроектирована структура веб-сервиса;
– разработана спроектированная клиентская часть веб-сервиса.

В результате исследования предметной области были изучены существующие решения. Ни один из аналогов полностью не повторяет функционал разработанного продукта, это значит, что удалось создать продукт, отличающийся от конкурентов.

Разработанный веб-сервис позволяет загружать, хранить и просматривать цифровые фотографии для личного пользования, дает возможности добавлять к фотоальбомам и фотографиям описательные характеристики, просматривать фотографии в режиме «Демонстрация», имеет «Архив» с возможностью восстановления удаленных объектов.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

На сегодняшний день переход от реального мира к цифровому ускоряется с каждым днем. Этот процесс охватывает практически все сферы нашей жизни. Различные отрасли с каждым годом модернизируются за счет внедрения новых технологий, в том числе динамически растет компьютеризация различных сфер производства. Например, чтобы подать заявление в банк не нужно приходить в офис, данная процедура стала возможна онлайн благодаря электронной подписи. Люди отказываются от бумажных заявлений, книг, а также фотографий. Миллиарды фотоснимков хранятся в интернете. Различные социальные сети позволяют мгновенно добавлять фотографии для публичного просмотра. Но хранение фотографий для общего обозрения не является инструментом всеобщего пользования.

Целью данной выпускной квалификационной работы является разработка клиентской части веб-сервиса «Электронный фотоальбом», который представляет собой аналог бумажного фотоальбома. Просмотр фотоальбомов в кругу семьи и друзей должен стать доступнее благодаря онлайн версии. Сортировка альбомов по датам, событиям и друзьям, личное пользование альбомов и демонстрация фотографий благодаря автоматическому воспроизведению – основная функциональность веб-сервиса «Электронный фотоальбом» [1]. Структура данного сервиса является интуитивной, чтобы обеспечить максимальный спрос у людей разных возрастов. Главные задачи со стороны UI-UX разработки – сделать интерфейс доступным, легким и удобным в использовании. Такой фотоальбом отличает и простота в обслуживании. Отличительные черты разрабатываемого электронного фотоальбома – информативность, лаконичность, индивидуальность, интуитивный и привлекательный дизайн [2].

Настоящий проект направлен на разработку клиентской части веб-сервиса, где будут представлены следующие возможности для пользователей:

– регистрация и авторизация;

– изменение данных, введенных при регистрации, в личном кабинете;

– создание фотоальбомов и добавление им описательных характеристик;

– добавление фотографий в созданные фотоальбомы и дополнение фотографий описательными характеристиками;

– осуществление поиска фотоальбомов и фотографий;

– просмотр фотоальбомов и фотографий;

– просмотр фотографий в режиме «Демонстрация»;

– помещение фотоальбомов и фотографий в «Архив» с возможностью их восстановления.

1 Анализ предметной области

Перед началом проектирования и разработки любого проекта необходимо выполнить важную работу – проанализировать предметную область. Данный этап занимает достаточное количество времени, так как предметная область влияет на содержание любого проекта.

Кроме того, исследования предметной области должны быть задокументированы и постоянно обновляться, чтобы точно задать границы проекта. Благодаря этим исследованиям можно четко определить цели внедрения системы, ее функционал и конкурентоспособность. Также полученные знания помогут IT-менеджеру правильней организовать разработку проекта, определить временные и материальные затраты на создание программного продукта и так далее [3].

Рассмотрим основную задачу нашей области. Фотоальбомы пришли в нашу жизнь еще около 1830-х годов [4]. С тех пор их постоянно улучшают и модернизируют, с каждым годом улучшается и качество фотографий. Сегодня напечатанные фотографии до сих пор пользуются спросом, но с появлением интернета и его внедрением в нашу жизнь перспектива использования фотоальбомов значительно снизилась. На смену таким фотоальбомам приходят цифровые аналоги. Они удобнее и надежнее обычных фотоальбомов.

1.1 Введение в анализ

Данный проект относится к электронному ресурсу и является веб-сервисом. Взаимодействие с таким типом происходит онлайн.

Это электронный фотоальбом, который в свою очередь имеет следующие характеристики:

– регистрация и наличие личного кабинета;

– возможность создания личного фотоальбома, его редактирование и удаление;

– наличие корзины с возможностью восстановления удалённых объектов;

– автоматическая демонстрация фотографий;

– возможность добавления описательных характеристик для фотографий.

Дополнительные сервисы, такие как полезная информация, рекламный блок в сочетании с грамотной раскруткой, могут сделать веб-сервис привлекательным для сторонних рекламодателей. Однако, данный программный продукт сосредоточен на контенте пользователя, а не на потреблении стороннего контента. И подобные решения могут только отталкивать заинтересованные стороны данного проекта [5].

1.2 Анализ существующих решений

При исследовании предметной области на просторах интернета был обнаружен сайт с подобным функционалом – Photo-Pick [6]. Его основная задача заключается в работе с командой. Альбомами можно делиться по ссылке, просматривать друзьям и ставить лайки. Очевидный минус данного сервиса – устаревший дизайн интерфейсов (рисунок 1).

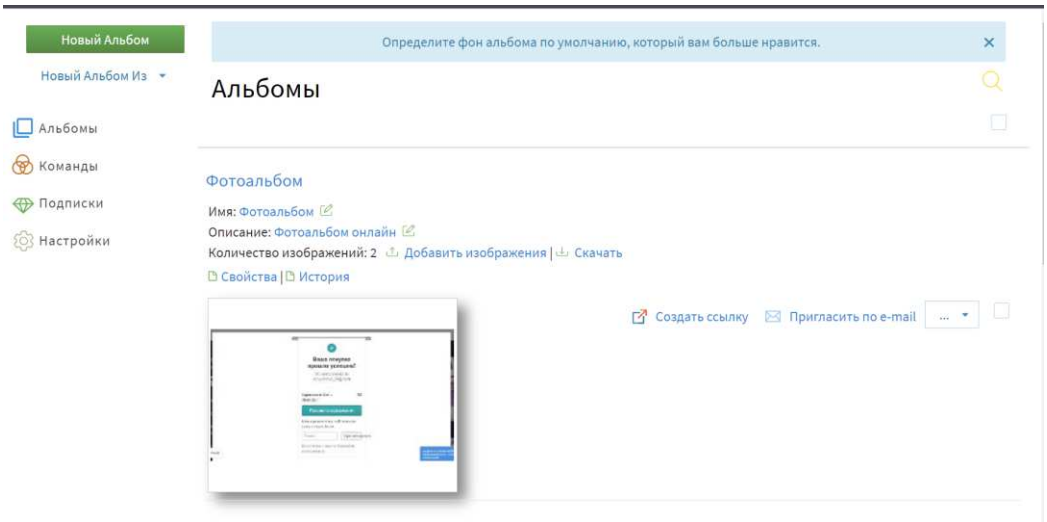

Рисунок 1 – Интерфейс аналога Photo-Pick

Без внимание не остались социальные сети, такие как vkontakte, instagram, facebook, twitter, pinterest. Все они являются публичными носителями фотографий, а также непосредственного общения между знакомыми и не знакомыми людьми. Цель данных социальных сетей демонстрация своих фотографий и получение обратной связи от других людей.

Для приватного доступа существуют Гугл-диск и Яндекс-диск, они предназначены для хранения фотографий и различных файлов. Минус этих хранилищ заключается в том, что фотографии хранятся по папкам с единственным заголовком, и альбомы не имеют описательных характеристик. Сложно отслеживать различные события. Несмотря на минималистичный дизайн, Яндекс-диск является сложным в использовании фотоальбомом, так как добавление фотографий в существующий альбом не является простой задачей. При нажатии на кнопку «Добавить фото» открывается папка «Фото» хранилища, а не папки компьютера (рисунок 2). Помимо этого существенным недостатком является ограничение памяти в добавлении файлов.

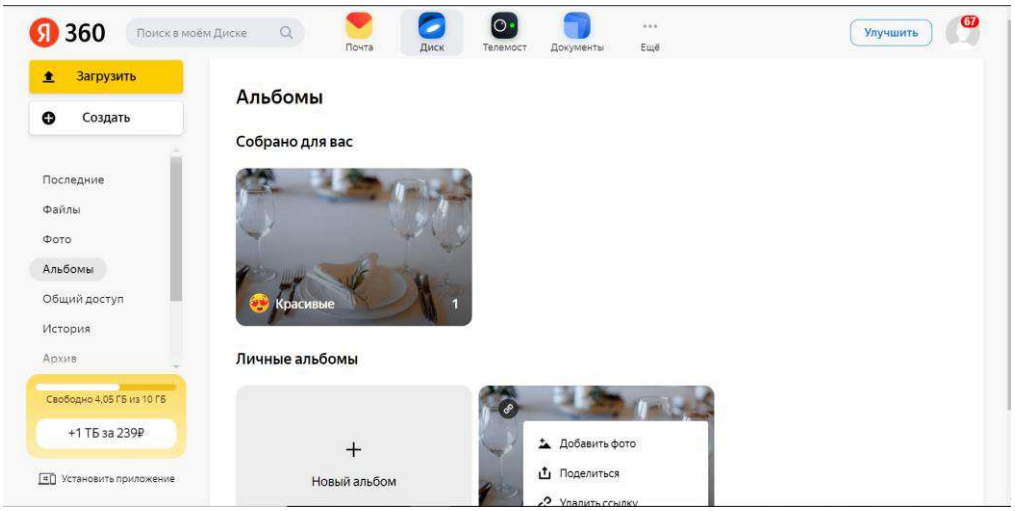

Рисунок 2 – Яндекс-диск

Также были изучены цифровые фоторамки, это совсем другой продукт, но для анализа необходимо было изучить данный аспект. Основной момент, на

который мы обратили внимание, это автоматическая демонстрация фотографий для комфортного просмотра на удобном расстоянии.

1.3 Определение требований к программному продукту

При разработке веб-сервиса «Электронный фотоальбом» необходимо было удовлетворить следующие требования:

– простая навигация по разделам сайта;

– продуманный дизайн, соответствующий тематике сайта;

– возможность загрузки фотографий;

– наличие регистрации и личного кабинета;

– возможность создания личного фотоальбома, его редактирования и удаления;

– наличие корзины с возможностью восстановления удалённых объектов;

– своевременный отчёт о проделанной работе и показ промежуточных разработок;

– разработка макета сайта и согласование его с заказчиком.

Проанализировав конкурентов, были выявлены и добавлены к разработке следующие требования:

– автоматическая демонстрация фотографий;

– возможность добавления описательных характеристик для фотографий и альбомов.

Данные функции не предусмотрены среди таких сервисов, как Гугл-диск и Яндекс-диск. Наша цель приблизить создаваемый программный продукт к реальным фотоальбомам. Добавление и редактирование фотографий, а именно добавление описательных характеристик помогут сохранить воспоминания о важном событии. Автоматическая демонстрация фотографий поможет просматривать снимки на расстоянии в кругу друзей или семьи, данный

функционал был заимствован у цифровых фоторамок, которые были приведены при анализе конкурентов.

1.4 Выбор инструментов разработки

Стек технологий – это набор технологий и инструментов, на основе которых разрабатывается программный продукт.

По версии сайта proglib React превосходит Angular и Vue [7]. React это все возможности языка программирования JavaScript, а Vue в свою очередь по большей части основывается на классических веб-технологиях, например, любой валидный HTML также будет валидным шаблоном Vue. Большим преимуществом React является простое и функциональное создание компонентов, которые помимо удобств в работе поддерживают элегантность кода API. Библиотека React очень популярна, наибольшим спросом пользуется в стартапах. Данная библиотека на рынке уже давно и благодаря этому можно найти ответ на любой вопрос. Помимо этого, для нее найдется множество опенсорсных плагинов и расширений, что позволяет разработать практически любой тип веб-сайта. Исходя из данных характеристик, библиотекой для написания фронтенда был выбран React [8].

Для разработки клиентской части был выбран язык JavaScript [9]. Это популярный и не сложный в понимании язык для разработки веб-сервиса. Среда программирования была выбрана Visual Studio Code. Данная среда не имеет аналогов по качеству поддержки всех необходимых инструментов для разработки данного проекта. Взаимодействие бэкенда с фронтендом будет реализовано с помощью API.

Для постановки задач, отслеживания работы каждого из участников команды была выбрана онлайн платформа управления проектами – Trello. Чаще всего данную систему используют небольшие компании и стартапы. Основное ее преимущество – это организация работы с помощью методологии канбан-досок [10]. Trello была выбрана по ряду следующих причин:

– интуитивный интерфейс;

– бесплатный доступ в рамках разработки проекта;

– возможность взаимодействия с другими инструментами для работы онлайн.

Для совместной работы над проектом выбрана распределенная система контроля версий – Git. Ее разработкой занимался создатель Linux – Линус Торвальдс. Цель данной разработки заключалась в совместном создании наработок разработчиками для ядра Linux. Git распространен за счет своей скорости, незамысловатого дизайна. Он поддерживает нелинейную разработку и полную децентрализацию[11]. Достоинства Git:

– является бесплатным ресурсом;

– имеет программное обеспечение с открытым исходным кодом;

– осуществляет все операции локально, что в свою очередь влияет на скорость работы;

– имеет возможность локального сохранения всего репозитория без потери качества;

– эффективен в хранении бэкапов;

– простое и удобное ветвление.

Для размещения репозитория был выбран сервис онлайн-хостинга репозиториев – GitHub [12]. GitHub выбирают за возможность контролировать доступ, багтрекинг, управление задачами и вики для каждого проекта. Основное преимущество это использование git-репозитория, загруженного на GitHub, через командную строку Git и Git-команд. Также доступны и все остальные функции, например, документация, запросы на принятие изменений (pull requests), история коммитов и другие [13].

1.5 Выводы

В данном разделе были проанализированы аналоги разрабатываемого веб-сервиса и выявлены задачи для реализации продукта, которые включают в

себя конкурентные преимущества. Готового аналогичного решения найти не удалось, поэтому с большой вероятностью продукт будет востребован.

Далее был определен технологический стек для разработки клиентской части веб-сервиса:

– библиотека для разработки пользовательского интерфейса React.js;

– язык программирования для клиентской части JavaScript;

Взаимодействие команды для синхронизации задач по разработке клиентской и серверной частей происходило посредством использования Trello. Для хранения версий проекта использовалась система контроля версий Git, а размещение репозитория было выполнено на платформе GitHub.

2 Планирование, управление и проектирование

2.1 Планирование

Планирование и управление являются важной частью создания продукта. Распределение работ должно производиться на первом этапе, чтобы разработка программного продукта успешно завершилась в установленные сроки [14].

В начале работы над проектом был определен подробный перечень задач, которые необходимо выполнить для достижения поставленной цели:

- изучение предметной области;
- выбор инструментов разработки;
- проектирование архитектуры информационной системы;
- проектирование интерфейса;
- разработка клиентской части веб-приложения.

2.1.1 Выбор методологии разработки

Разработка программного продукта имеет много достойных методологий, каждая из которых учитывает различные аспекты. Чтобы определиться с выбором методологии, сравним самые популярные из них.

Waterfall Model. Эта модель является достаточно старой и предполагает прохождение всех стадий последовательно, то есть следующая стадия начнется только после того, как закончится предыдущая. При работе с данной моделью управление проекта будет простым, а его разработка будет проходить быстро. Но для успешного завершения проекта необходимо заранее определить требования и способы его реализации [15]. Основной минус данной модели состоит в том, что нельзя вернуться к предыдущей стадии (рисунок 3).

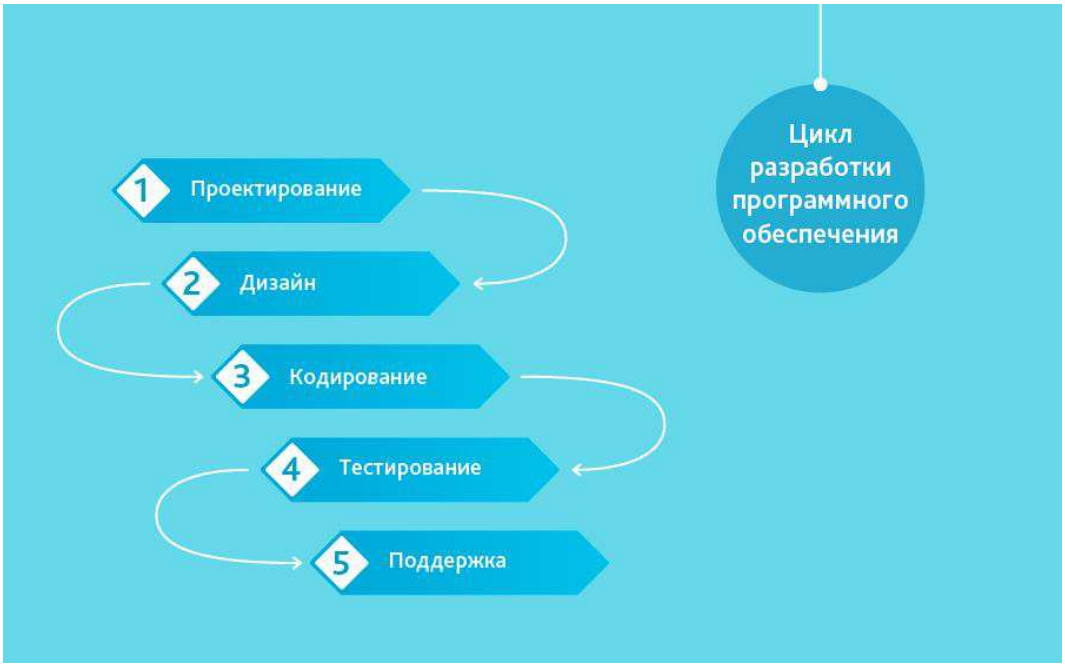

Рисунок 3 – Waterfall Model

Каскадную методологию рекомендует использовать:

– если все требования понятны, определены заранее и зафиксированы;

– уверенность в квалификации сотрудников;

– в малых или средних проектах.

Agile Model. При работе с данной моделью есть возможность контролировать результат после каждого шага. «Гибкая» методология строится за счет коротких ежедневных встреч, так называемых «Scrum», а также повторяющихся собраний [16]. Данный подход рекомендован для больших проектов с длительным жизненным циклом, где происходит постоянная адаптация к условиям рынка (рисунок 4).

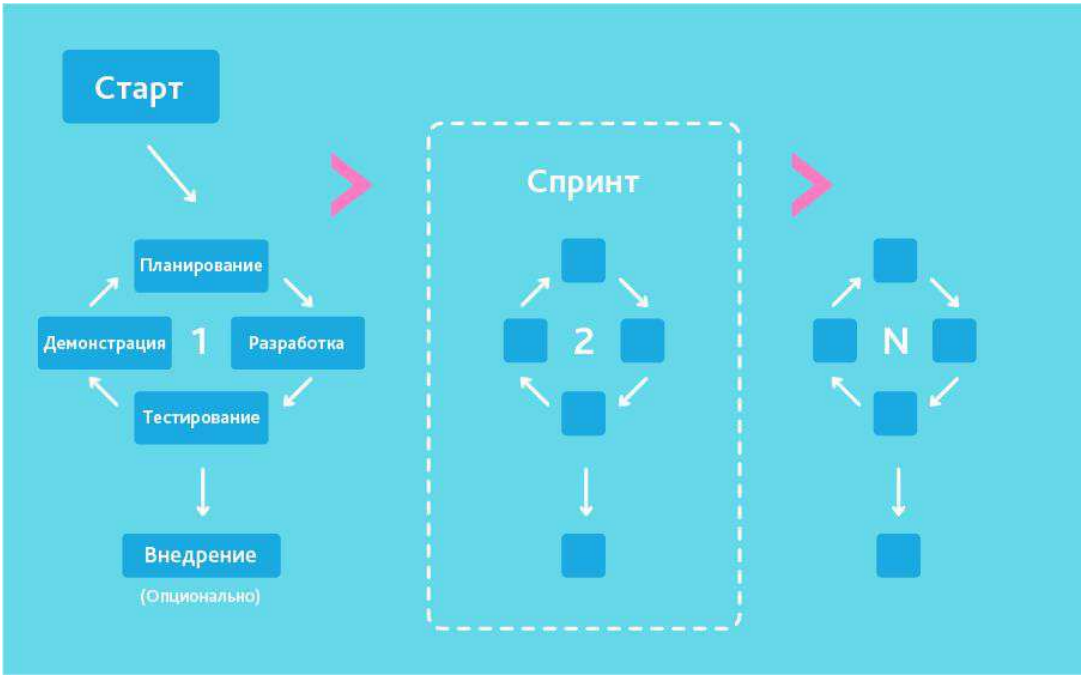

Рисунок 4 – Agile Model «Scrum»

Гибкую методологию рекомендуют использовать:

– когда продукт сильно зависим от внешних факторов, например, потребностей пользователей, и его нужно постоянно корректировать;

– когда нет четкого плана действий, а есть только небольшое планирование на ближайшее время.

Одна из разновидностей Agile методологии является Kanban – это наглядная система разработки, и для достижения наглядности используются канбан-доски [17]. Такая доска имеет минимум три колонки: «сделать» (to-do), «в процессе» (in progress) и «сделано» (done).

При грамотном управлении Kanban предоставляет предельную скорость работы и скорость реагирования на изменения. Пользуется большим спросом среди стартапов, которые активно ведут разработку без четко разработанного плана (рисунок 5).

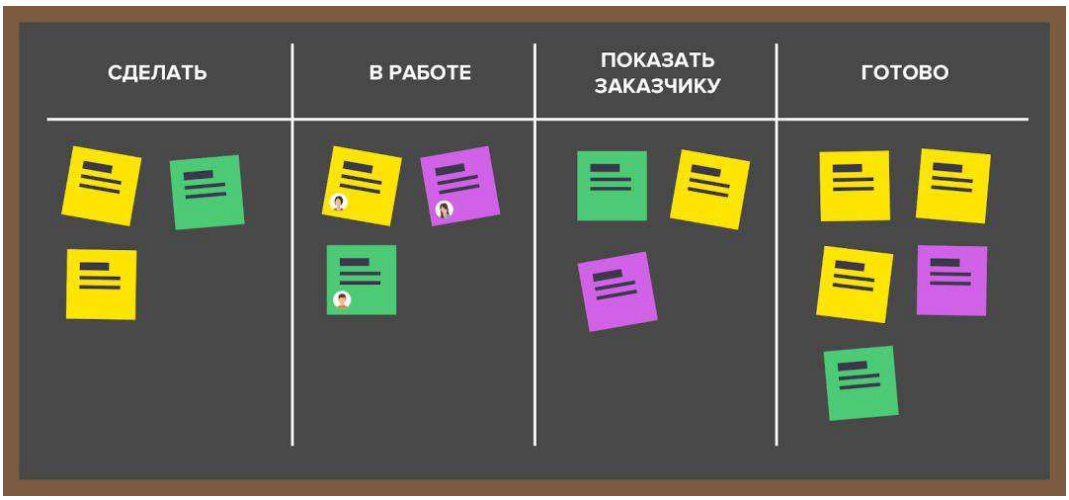

Рисунок 5 – Agile Model «Kanban»

Ограничениями данной модели можно отметить работу в больших командах (более 5 человек) и сложность применения для долгосрочного планирования. Для данного проекта ограничения не являются критичными, поэтому методология Kanban выбрана для управления разработкой программного продукта.

2.2 Управление

Для распределения задач была выбрана система управления проектами Trello. Данная система используется для небольших проектов, но при этом имеет развернутый функционал [18]. Более того, она позволяет организовать работу с помощью методологии Kanban, выбранной для данного проекта.

2.2.1 Заинтересованные стороны

Таблица 2.2.1 – Заинтересованные стороны проекта

Код	**Фамилия, инициалы**	**Должность**	**Полномочия**	**Интерес**
А	Хныкин А. В.	Научный руководитель	Значительные	Существенный

D	Потенциальные пользователи		Незначительные	Несущественный

Таблица 2.2.2 – Стратегия управления заинтересованными сторонами проекта

Фамилия, инициалы	**Должность**	**Код**	**Стратегии управления**
Хныкин А. В.	Научный руководитель	A	Тесно сотрудничать
Потенциальные пользователи		D	Наблюдать

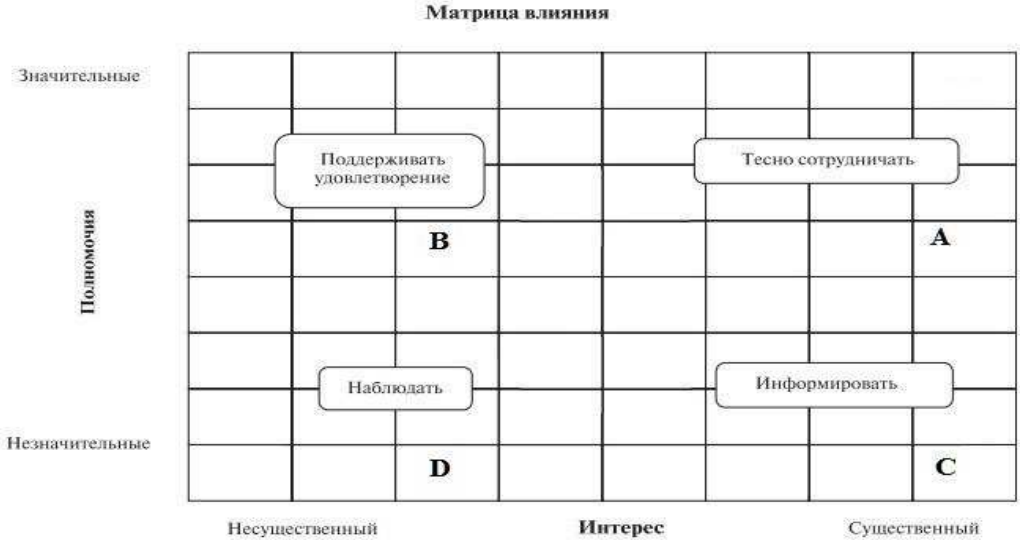

Рисунок 6 – Матрица анализа влияния заинтересованных сторон проекта

Таблица 2.2.3 – Реестр заинтересованных сторон проекта

Фамилия, инициалы	**Роль в проекте**	**Требования**	**Влияние**
Хныкин А. В.	Научный руководитель	Разработать веб-сервис	Среднее влияние
Хныкин А. В.	Проектный менеджер со стороны заказчика	Разработать веб-сервис	Сильное влияние
ИКИТ СФУ	Заказчик	Разработать веб-сервис	Среднее влияние
Пользователи	Конечные пользователи	Получить веб-сервис	Слабое влияние

Таблица 2.2.4 – План управления персоналом проекта

№	Описание роли	Описание полномочий	Описание дефицитов
1	**Проектный менеджер**		
	Специалист, занимающийся управлением проектом, поддерживающий постоянную связь с заказчиком	Управление проектом, проектирование и расстановка приоритетов, планирование выполнения задач	
2	**Тим-лидер**		
	Специалист, занимающийся управлением проектом	Проведение совещаний внутри команды, планирование выполнения задач, обсуждение требований к продукту с заказчиком	Опыт управления программными проектами
3	**Бэкенд-разработчик**		
	Специалист, занимающийся разработкой серверной части проекта	Разработка серверной части приложения, обсуждение задач с тим-лидером	Опыт создания express приложений
4	**Фронтенд-разработчик**		
	Специалист, занимающийся разработкой клиентской части проекта	Разработка клиентской части проекта, обсуждение задач с тим-лидером	Опыт создания react приложений
5	**Тестировщик**		
	Специалист, занимающийся тестированием продукта	Тестирование продукта, обсуждение задач с тим-лидером	

Рисунок 7 – План управления персоналом проекта

2.2.2 Коммуникации

Для достижения эффективного взаимодействия и обмена информацией между участниками проекта были обсуждены способы и частота коммуникаций (таблица 2.2.5 – 2.2.6) [19].

Таблица 2.2.5 – Участники коммуникации и их потребности

Участник коммуникации	**Информационные потребности**
Заказчик	Отчет о ходе разработки проекта
Проектный менеджер со стороны заказчика	Отчет о ходе разработки проекта; данные о технической составляющей проекта (архитектура, используемы технологии и др.).
Тим-лидер	Отчет о ходе разработки проекта, данные о технической составляющей проекта (архитектура, используемы технологии и др.); данные о результатах тестирования проекта.
	Отчет о ходе разработки проекта; данные о технической составляющей проекта

Разработчики	(архитектура, используемы технологии и др.); данные о результатах тестирования проекта.
Тестировщик	Отчет о ходе разработки проекта; данные о технической составляющей проекта (архитектура, используемы технологии и др.).

Таблица 2.2.6 – Средства и способы коммуникации

Отправитель	**Получатель**	**Средство коммуникации**	**Частота**	**Способ коммуникации**	**Ожидаемый результат**
Тим лидер	Менеджер проекта со стороны заказчика	Email	1-2 раза в неделю	Письменная коммуникация	Изменение в знаниях получателя
Тим лидер	Заказчик	Отчет	1-2 раз в месяц	Письменная коммуникация	Изменение в знаниях получателя
Разработчик, тестировщик, тим лидер	Куратор проекта	Scrum-митинг	1 раз в неделю	Устная коммуникация, письменная коммуникация	Изменение в знаниях получателя
Разработчик, тестировщик	Тим лидер	Scrum-митинг	1 раз в неделю	Устная коммуникация	Изменение установок получателя
Разработчики	Тестировщик	Совещание	1 раз в неделю	Устная коммуникация	Изменения установок получателя

2.2.3 Качество

Необходимо было обратить внимание на качество проекта и выделить для этого определенные требования. Качество включает все модули проекта, которые создают полноценный продукт. Качество является важной составляющей, так как помогает зафиксировать цели, делает их задокументированными [20].

Для управления качеством была составлена таблица, в которой внесены критерии оценки требований качества (таблица 2.2.7) [21].

Таблица 2.2.7 – План управления качеством

	Стандарт качества/показатели	**Размерность**	**Плановые значения**	**Фактические значения**	**Отклонение**
1. Время выполнения этапа					
1	Время выполнения этапа «Проектирование»	день	21	21	-
2	Время выполнения этапа «Создание серверной части»	день	90	90	-
3	Время выполнения этапа «Создание клиентской части»»	день	90	90	-
4	Время выполнения этапа «Тестирование»	день	72	-	-
5	Время выполнения этапа «Сдача проекта»	день	5	14	9
2. Тестирование					
1	Тестирование интерфейса	день	38	-	-
2	Ручное тестирование	день	42	-	-
3. Публикация приложения					
1	Разработка и размещение документации	день	82	60	22

2.3 Проектирование

2.3.1 Проектирование архитектуры веб-сервиса

Архитектура веб-сервиса – это концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь ее компонентов [22].

Разработанный программный продукт имеет клиент-серверную модель взаимодействия (рисунок 8).

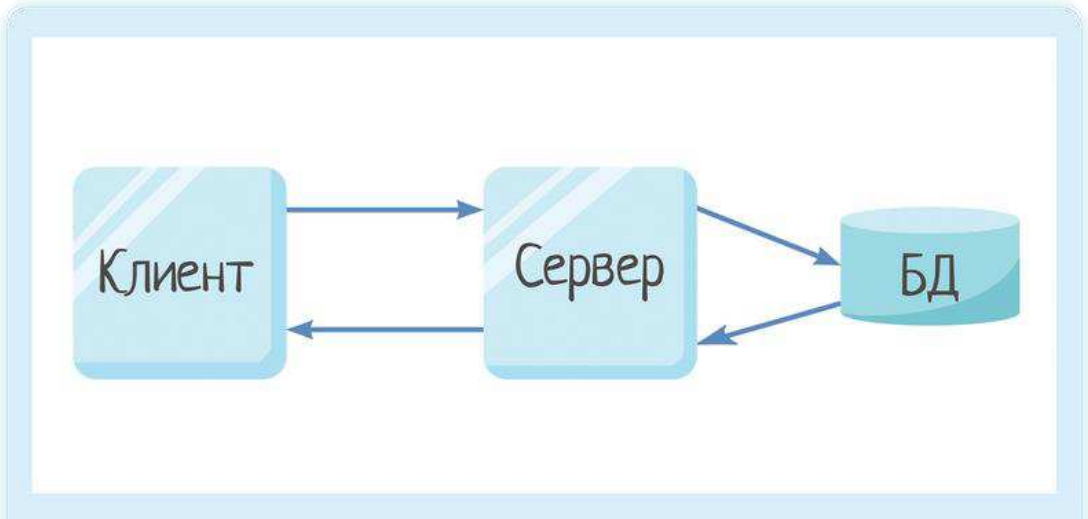

Рисунок 8 – Клиент-серверная архитектура

Данная архитектура выбрана из-за разделения программного кода на клиентскую и серверную стороны. Благодаря этому требования к машинам клиентов могут быть пониженными, ведь большая часть вычислительных операций будет производиться на сервере. Помимо этого, архитектура клиент-серверного приложения является достаточно гибкой, что позволяет администратору сделать локальную сеть более защищенной. Данный продукт имеет три независимые части: клиентская часть, разрабатываемая с использованием HTML, CSS, JavaScript и React.js; сервер веб-приложения; сервер БД.

2.3.2 Диаграмма вариантов использования

Диаграмма вариантов использования (англ. use case diagram) – это диаграмма, на которой изображаются актёры и варианты использования, выполненная с использованием языка моделирования UML [23]. Диаграммы помогают описать взаимодействие пользователя с сайтом, показывая конкретные действия, которые может осуществлять актер.

Диаграмма вариантов использования веб-сервиса «Электронный фотоальбом» для пользователя продемонстрирована на рисунке 9.

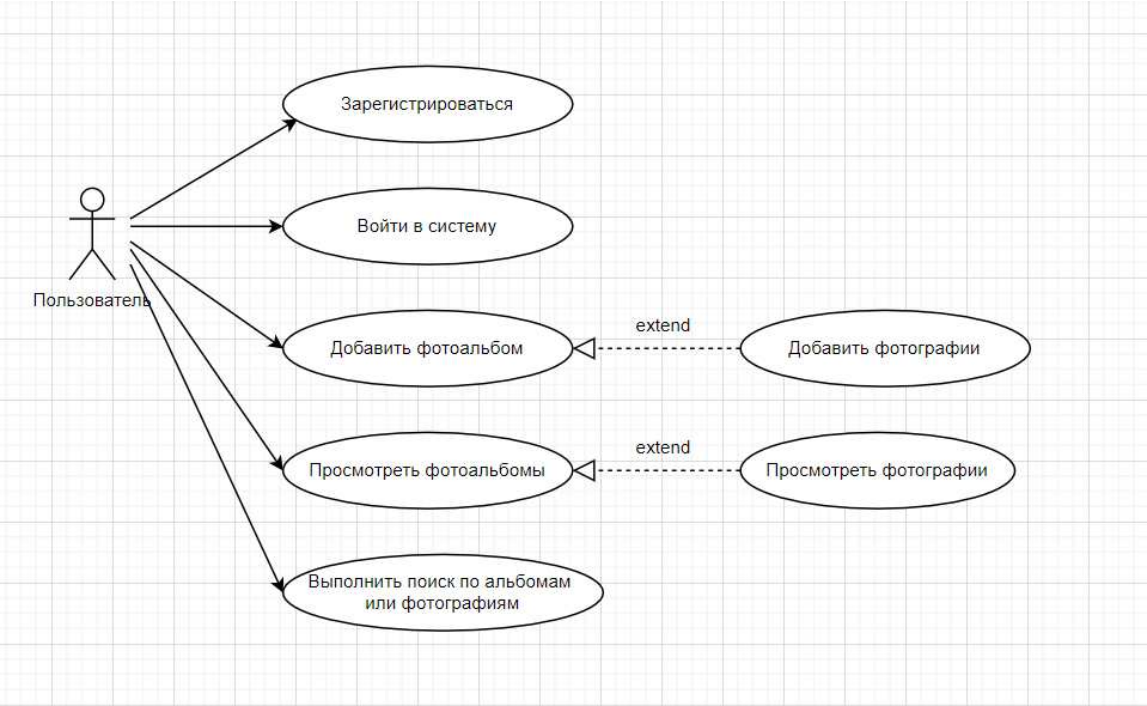

Рисунок 9 – Диаграмма вариантов использования пользователя веб-сервиса

2.3.3 Проектирование интерфейса

На сегодняшний день важным показателем востребованности продукта является его визуальная составляющая. Интерфейс веб-приложения должен быть не только понятным и удобным, но и отличаться собственным стилем и подачей. Пользователь решает в течение 5 секунд продолжать взаимодействие с сайтом или нет, и если интерфейс является не дружелюбным или сложным, то с вероятностью 96% пользователь не станет тратить на него свое время [24].

Паттерны поведения на сайтах складываются уже более 25 лет, к ним можно отнести: верхнее или боковое меню; контент, расположенный слева направо; логотип, который находится в верхнем левом углу. Все эти и другие паттерны необходимо учитывать при разработке интерфейса веб-сервиса [25].

Данный продукт не имеет ограничений по возрасту или полу, но его целевая аудитория – женщины и мужчины 25-50 лет. Поэтому интерфейс

должен быть интуитивно понятным и удобным. Выделим основные характеристики:

– приятная цветовая палитра, без кричащих цветов;
– расположение важных элементов в доступных местах;
– подробное описание при возникновении ошибок;
– демонстрация контента на всю возможную ширину.

Ниже представлены некоторые прототипы, разработанные для понимания структуры проекта (рисунок 10-13).

Рисунок 10 – Прототип формы регистрации

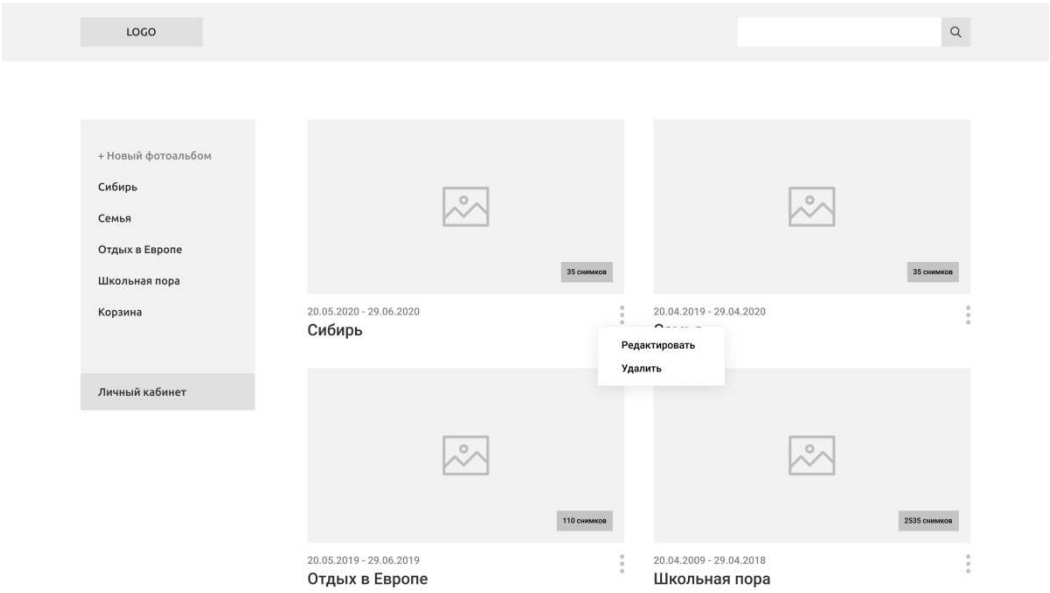

Рисунок 11 – Прототип страницы альбомов

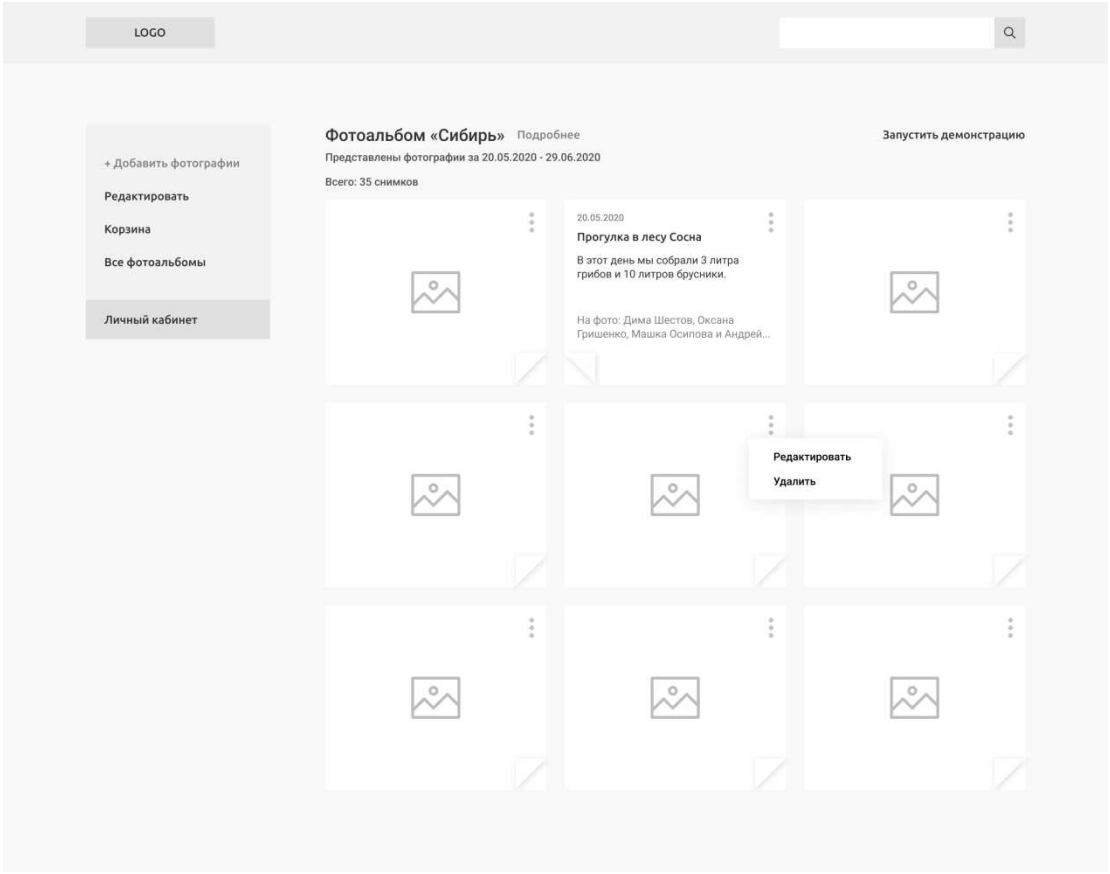

Рисунок 12 – Прототип страницы альбома с фотографиями

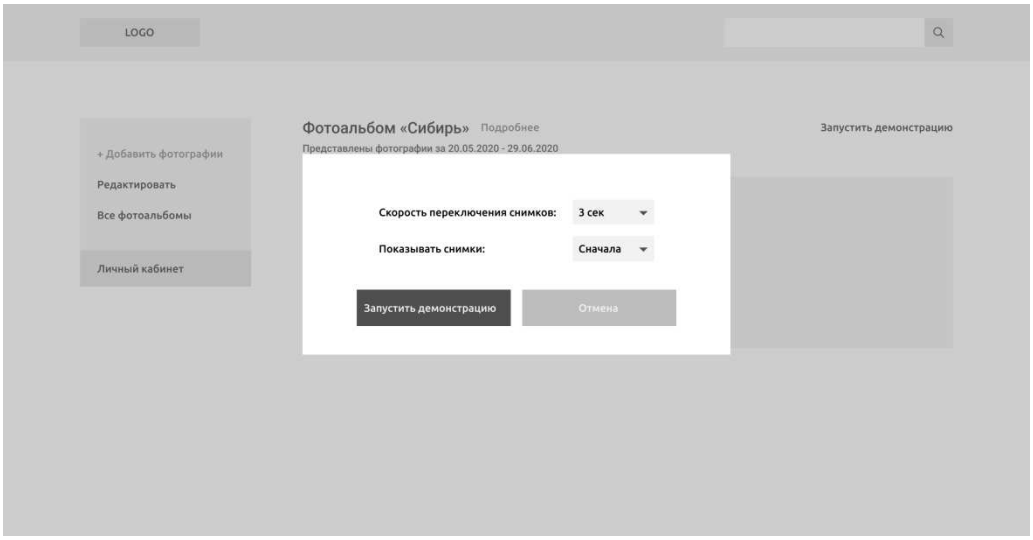

Рисунок 13 – Прототип модального окна

2.4 Выводы

В подразделе «Планирование» были проанализированы методологии разработки программного продукта и выявлена наиболее подходящая для данного проекта. Среди Waterfall Model, Agile Model «Scrum», Agile Model «Kanban» была выделена Agile Model «Kanban», так как проект является небольшим и не имеет строгой структуры. Для распределения задач была выбрана система управления проектами Trello.

В подразделе «Управление» были описаны заинтересованные стороны, коммуникации между участниками проекта и определены требования для управления качеством проекта.

Далее были спроектированы архитектура информационной системы, UML диаграмма пользователя веб-сервиса, а также представлены требования к интерфейсу программного продукта.

3 Разработка клиентской части

Разработка клиентской части проводилась в редакторе исходного кода Visual Studio Code. Данный редактор был выбран из-за следующих характеристик:

– бесплатное пользование;

– не требует много памяти;

– поддерживает множество языков программирования;

– имеет автодополнение, подсветку синтаксиса, отладку программ, плагины и прочее.

Далее представлена структура проекта со стороны клиента (рисунок 14).

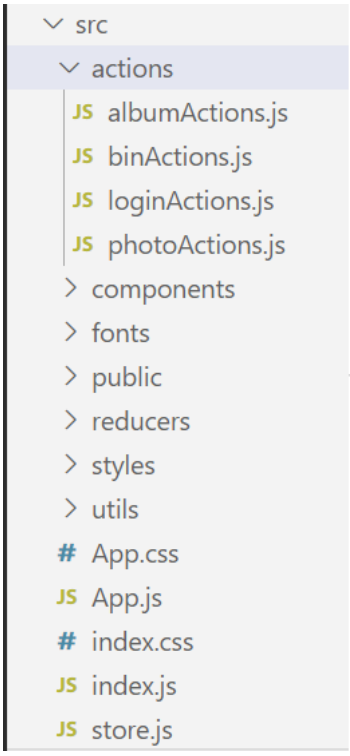

Рисунок 14 – Структура со стороны клиента

3.1 Разработка клиентской стороны веб-сервиса

Разработка данного проекта происходила с помощью библиотеки React.js. Что позволило создавать различные компоненты и повторно их использовать. Так в директории components имеются папки Navbar и Dashboard. Navbar

используется практически на каждой странице проекта кроме страниц Входа и Регистрации. Папка Dashboard включает в себя несколько компонентов. Например, там находятся карточка альбома, модальное окно создания/редактирования альбома, модальное окно удаления альбома и другие. Далее представлен компонент CreateAlbumModal, который принимает такие параметры, как название фотоальбома и возможность установления обложки для альбома (рисунок 15-18).

src > components > Dashboard > JS CreateAlbumModal.js > CreateAlbumModal

```
1   import React, { useEffect } from 'react';
2   import withModal from '../common/withModal';
3   import { Modal } from 'antd';
4   import { useFormik } from 'formik';
5   import * as Yup from 'yup';
6   import { message, Button } from 'antd';
7
8   import TextFieldGroup from '../common/TextFieldGroup';
9   import '../../styles/modals/createAlbumModal.scss';
10
11  function CreateAlbumModal({
12    children,
13    showModal,
14    closeModal,
15    isVisible,
16    createAlbum,
17    editAlbum,
18    type,
19    album,
20  }) {
21    const formik = useFormik({
22      initialValues: {
23        name: '',
24        file: '',
25        previewUrl: '',
26      },
27
28      validationSchema: Yup.object({
29        name: Yup.string().required(),
30      }),
```

Рисунок 15 – Компонент CreateAlbumModal

Рисунок 16 – Компонент CreateAlbumModal

Рисунок 17 – Компонент CreateAlbumModal

src > components > Dashboard > JS CreateAlbumModal.js > CreateAlbumModal

```
106                 <div className="createModal__emptyPreview">
107                   <input
108                     id="fileInput"
109                     type="file"
110                     name="file"
111                     onChange={(e) => {
112                       const file = e.target.files[0];
113                       formik.setFieldValue('file', file);
114                       const reader = new FileReader();
115 
116                       reader.onloadend = function (e) {
117                         formik.setFieldValue('previewUrl', reader.result);
118                       };
119                     }}
120                   />
121                   <label
122                     onClick={() => {
123                       const fileInput = document.getElementById('fileInput');
124                       fileInput.click();
125                     }}>
126                     + Добавить обложку
127                   </label>
128                 </div>
129               )}
130               {formik.values.previewUrl ? (
131                 <Button className="createModal__submit" size="large" htmlType="submit">
132                   Сохранить изменения
133                 </Button>
134               ) : (
135                 <Button className="createModal  submit" size="large" htmlType="submit">
```

Рисунок 18 – Компонент CreateAlbumModal

Для упрощения передачи данных хранилища через контекст использовалась библиотека для JavaScript – Redux. Основным ее преимуществом являются чистые функции, это означает изолированные друг от друга функции. С его помощью были созданы небольшие функции, которые делают что-то одно и являются независимыми [26]. Были написаны следующие события:

– получение всех альбомов;
– получение конкретного альбома;
– создание альбома;
– редактирование альбома;
– удаление альбома;
– добавление фотографий;
– редактирование фотографии;
– удаление фотографии.

Далее представлен один из запросов, написанных с помощью Redux – получение всех альбомов (рисунок 19).

src > actions > JS albumActions.js > ...

```
import axios from "axios";
import {
  ALBUMS_FAILURE,
  ALBUMS_SUCCESS,
  ALBUM_FAILURE,
  ALBUM_REQUEST,
  ALBUM_SUCCESS,
  ApiUrl,
} from "../utils/constants";

export const getAlbums = () => async (dispatch) => {
  try {
    const msg = await axios({
      method: "GET",
      url: ApiUrl + "albums/",
    });

    return dispatch({ type: ALBUMS_SUCCESS, data: msg.data });
  } catch (error) {
    return dispatch({ type: ALBUMS_FAILURE });
  }
};
```

Рисунок 19 – Запрос на получение всех альбомов

В Redux каждое действие имеет свойство type для описания действий. А для вызова действия в любом месте приложения, необходимо использовать метод dispatch(). Таким образом, при получении всех альбомов можно получить два варианта событий: первое – успешное отображение альбомов и второе – получение ошибки.

Одним из преимуществ Redux является использование редюсеров, далее представлены редюсеры для предыдущего примера (рисунок 20).

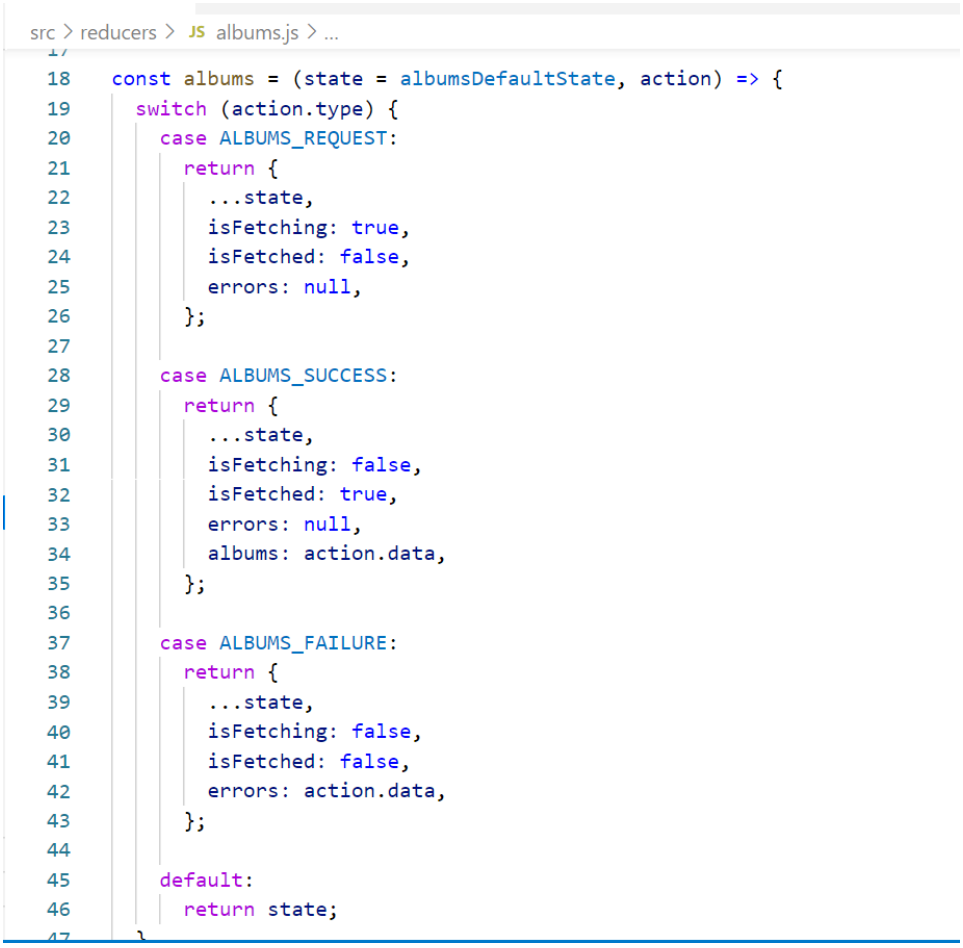

Рисунок 20 – Редюсеры альбомов

Их использование помогает считывать из хранилища текущее состояние через отправленное действие, после чего они возвращают новое состояние приложения. В данном случае состояние зависит от таких характеристик, как isFetching, isFetched, errors.

Чтобы иметь возможность пользоваться веб-сервисом в браузере, необходимо использовать react-router. Так как происходит обработка динамических запросов на сервере, следует использовать BrowserRouter. Отображение страницы входа зависит от статуса пользователя. Если пользователь не вошел в систему, его будет приветствовать страница авторизации. Если пользователь вошел в систему, отобразится страница с альбомами (рисунок 21-22).

src > JS App.js > ...

```
import React from "react";
import {
  BrowserRouter as Router,
  Route,
  Switch,
  Redirect,
} from "react-router-dom";
import { connect } from "react-redux";
import Cookie from "js-cookie";

import "./App.css";
import AlbumPage from "./components/AlbumPage";
import Bin from "./components/Bin";
import Dashboard from "./components/Dashboard";
import Navbar from "./components/Navbar";
import WelcomePage from "./components/WelcomePage";
import Register from "./components/Register";
import Login from "./components/Login";

const isAuth = () => {
  const token = Cookie.get("photoGallery");

  if (!token) return false;

  return true;
};

const IsAuthRoute = ({ component: Component, ...rest }) => (
  <Route
    {...rest}
```

Рисунок 21 – Пути веб-сервиса при первоначальном взаимодействии

src > JS App.js > ...

```
const IsAuthRoute = ({ component: Component, ...rest }) => (
  <Route
    {...rest}
    render={(props) =>
      isAuth() ? (
        <Redirect
          to={{
            pathname: "/",
          }}
        />
      ) : (
        <Component {...props} />
      )
    }
  />
);

const AuthRoute = ({ component: Component, ...rest }) => (
  <Route
    {...rest}
    render={(props) =>
      isAuth() ? (
        <div>
          <Navbar />
          <Component {...props} />
        </div>
      ) : (
        <Redirect
          to={{
            pathname: "/login",
```

Рисунок 22 – Пути веб-сервиса при первоначальном взаимодействии

3.2 Выводы

В данном разделе были рассмотрены ключевые моменты разработки клиентской части веб-сервиса. В основу данной разработки легла библиотека React.js, которая позволила использовать компоненты. Также было рассмотрено взаимодействие с библиотекой для JavaScript с открытым исходным кодом – Redux. И приведены некоторые запросы на сервер.

4 Описание работы программного продукта

4.1 Результаты разработки программного продукта

В результате разработки клиентской части веб-сервиса были получены следующие решения:

– страница регистрации и авторизации;

– страница личного кабинета;

– страница альбомов;

– модальные окна для добавления, редактирования и удаления альбома;

– страница с фотографиями;

– модальные окна для добавления, редактирования и удаления фотографий;

– реализация просмотра фотографий в режиме «Демонстрация»;

– разработка «Архива» с возможностью восстановления удаленных объектов.

Данные страницы будут представлены ниже на рисунках 23-35.

Страницы авторизации и регистрации имеют только нужные поля для заполнения. Регистрация аккаунта проходит в один этап и занимает не более 5 минут, так как не требует подтверждения электронной почты (рисунок 23-24).

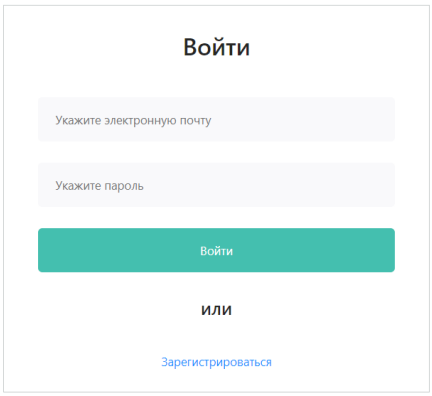

Рисунок 23 – Страница авторизации

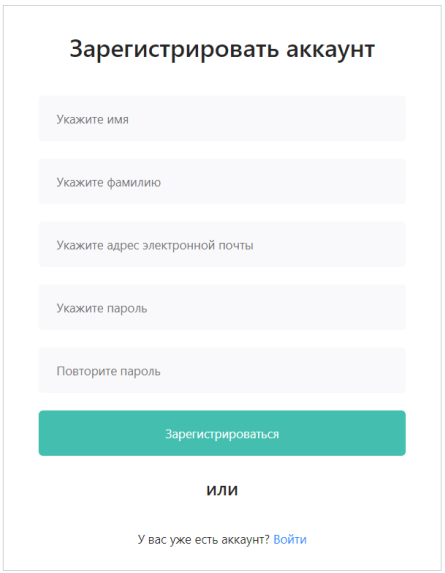

Рисунок 24 – Страница регистрации

Страница альбомов имеет левое боковое меню с важными элементами. Первым пунктом стоит создание нового альбома, далее представлены созданные альбомы, архив фотографий и личный кабинет с возможностью выхода. Шапка веб-сервиса содержит логотип, который ведет на страницу альбомов и поиск по датам или описанию альбомов, фотографий (рисунок 25).

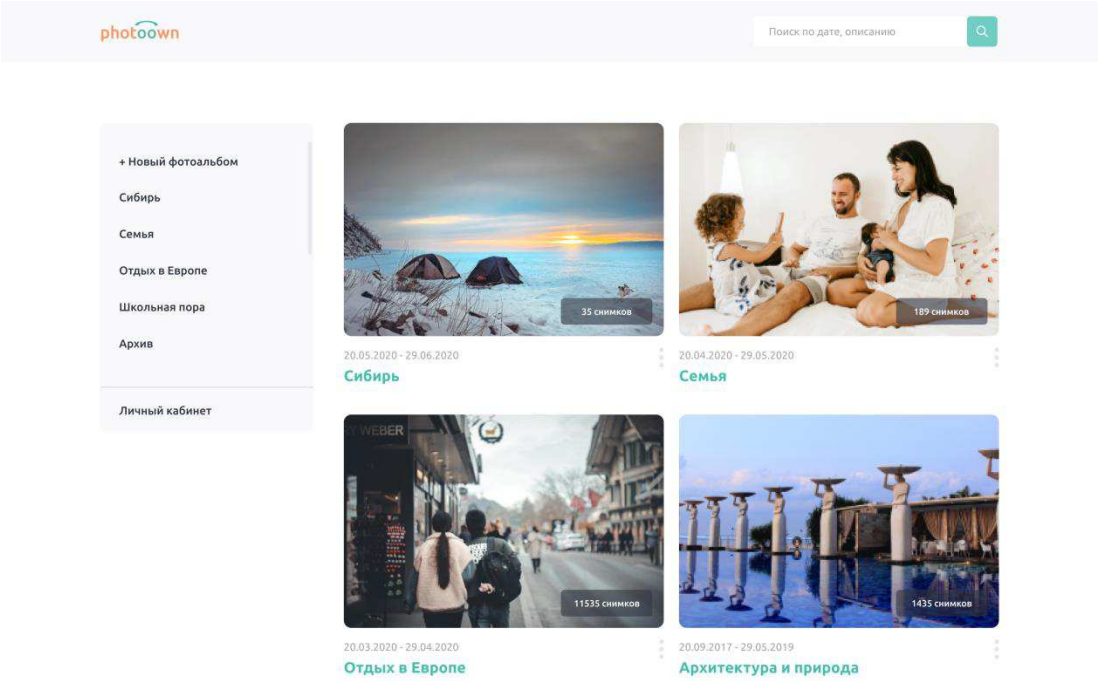

Рисунок 25 – Главная страница альбомов

Далее представлены модальные окна для создания и редактирования фотоальбома. Если обложка не выбирается пользователем, то ставится заглушка, а после добавления фотографий на обложку автоматически устанавливается первая фотография (рисунок 26-27).

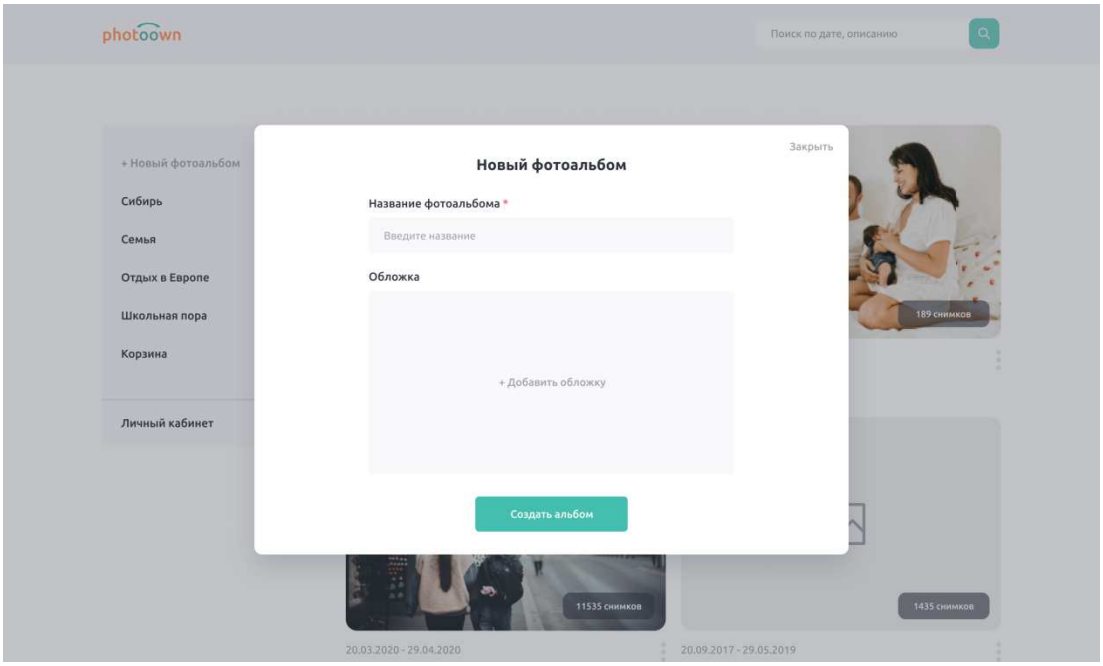

Рисунок 26 – Создание нового фотоальбома

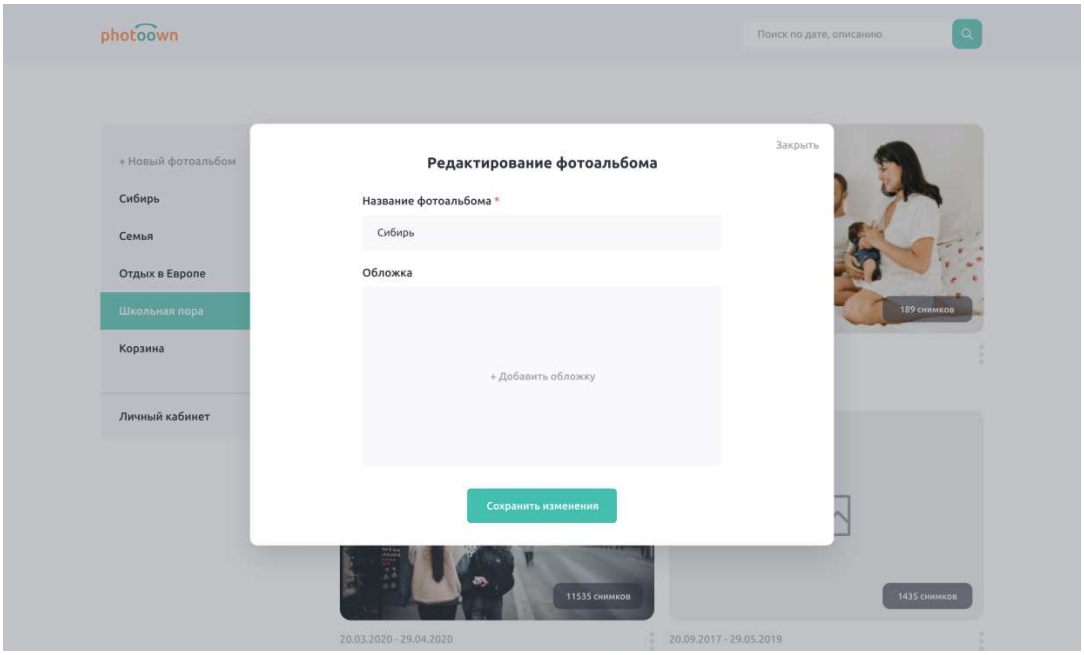

Рисунок 27 – Редактирование фотоальбома

Модальное окно для удаления имеет на выбор две кнопки. Можно переместить альбом в корзину или удалить безвозвратно (рисунок 28).

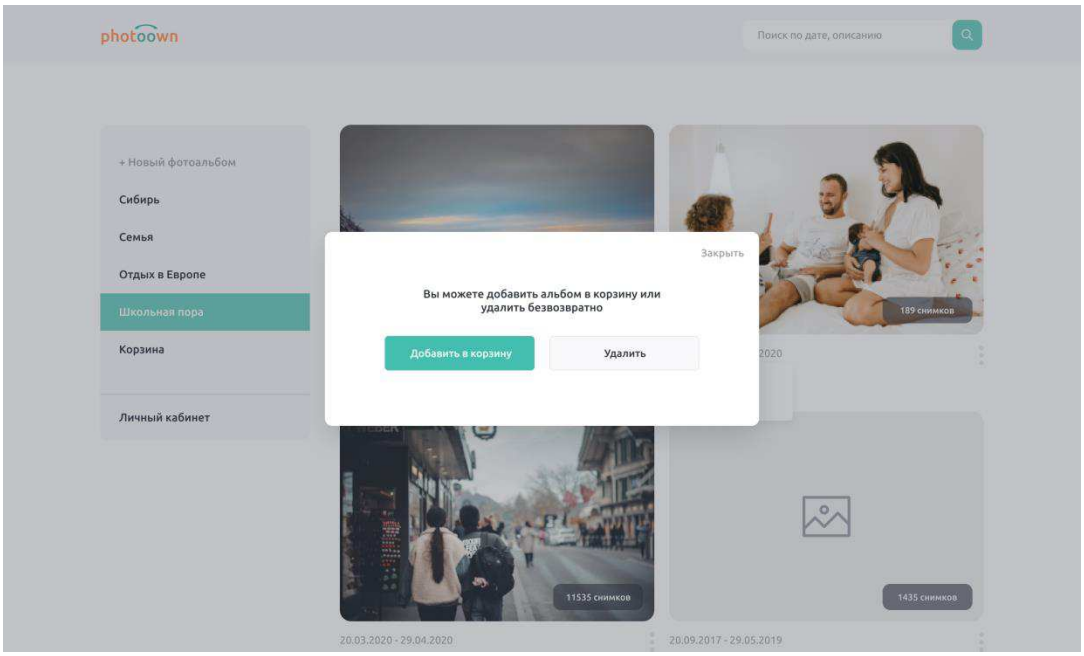

Рисунок 28 – Удаление фотоальбома

Альбом без фотографий представлен на рисунке 29.

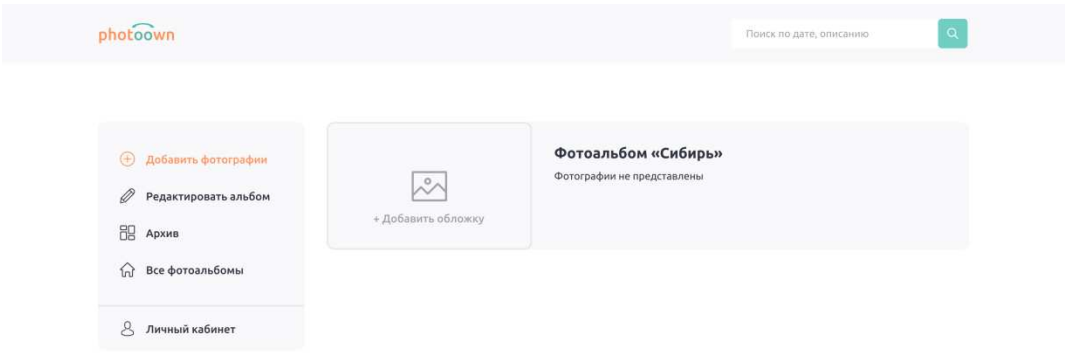

Рисунок 29 – Пустой фотоальбом

На данный момент фотографии загружаются с папок компьютера в любом количестве, пока фотографии не загружены кнопка «Выгрузить в альбом» не доступна (рисунок 30). После выбора фотографий пользователю предлагается прикрепить еще фотографии, выгрузить выбранные фотографии в альбом или удалить любые из добавленных фотографий (рисунок 31).

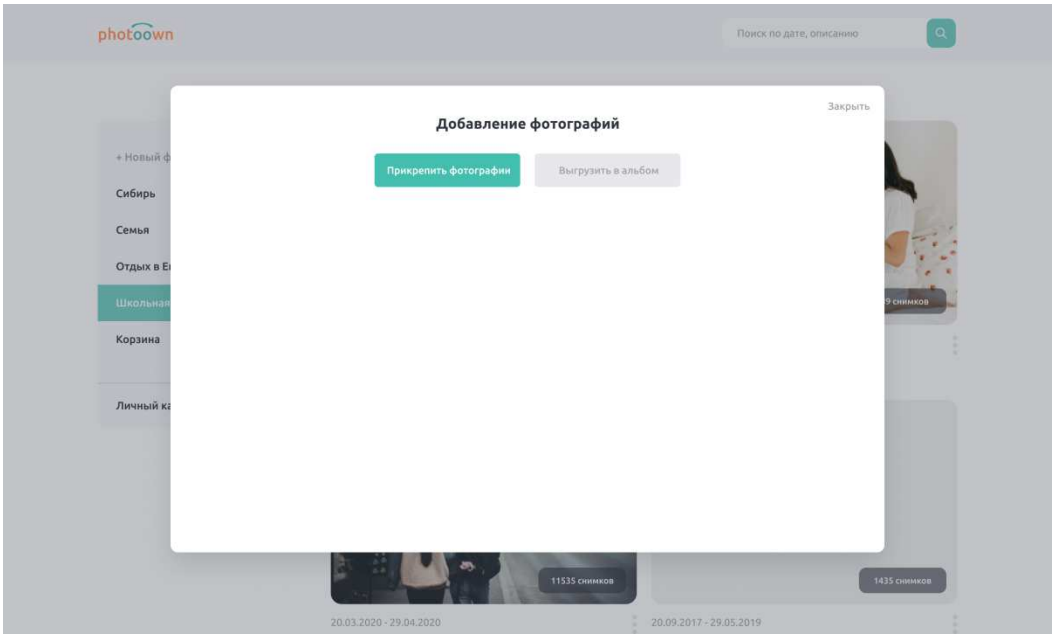

Рисунок 30 – Первоначальный экран добавления фотографий

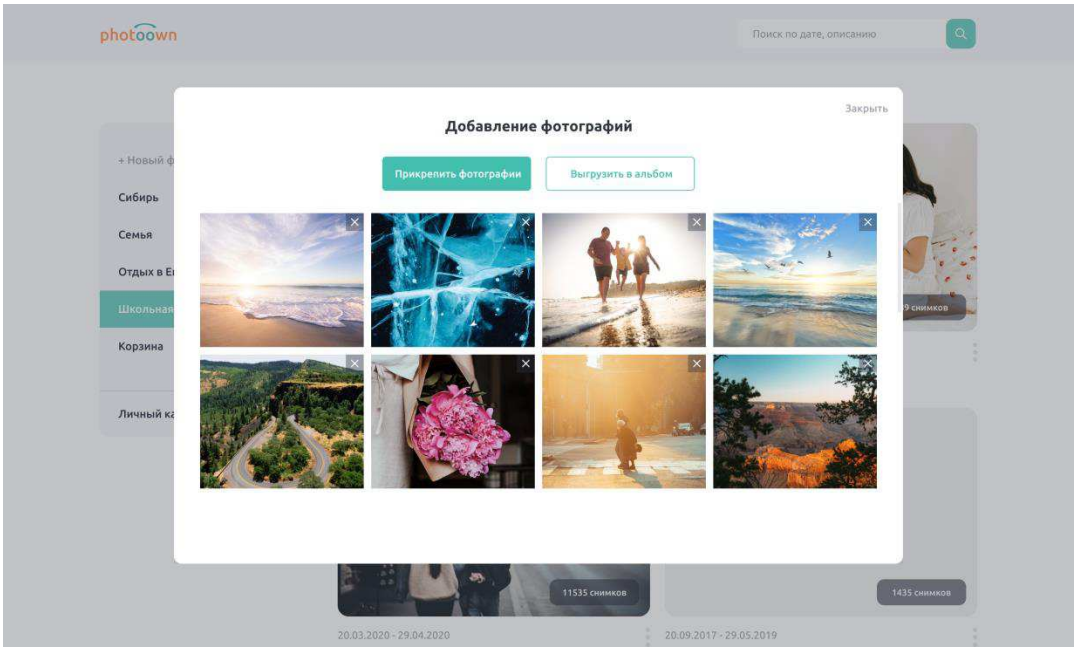

Рисунок 31 – Добавление фотографий

Страница альбома с фотографиями имеет фиксированное боковое меню с необходимыми действиями, такими как добавить фотографии, редактировать альбом, переход в архив, возврат к странице альбомов или переход в личный кабинет. Также на данной странице представлена информации об альбоме, его обложка и кнопка для запуска демонстрации. На третью фотографию наведен

курсор, при котором фотография оборачивается другой стороной, с веденными пользователем характеристиками. Если у фотографии отсутствует описание, то фотография оборачивается серым пустым квадратом (рисунок 32).

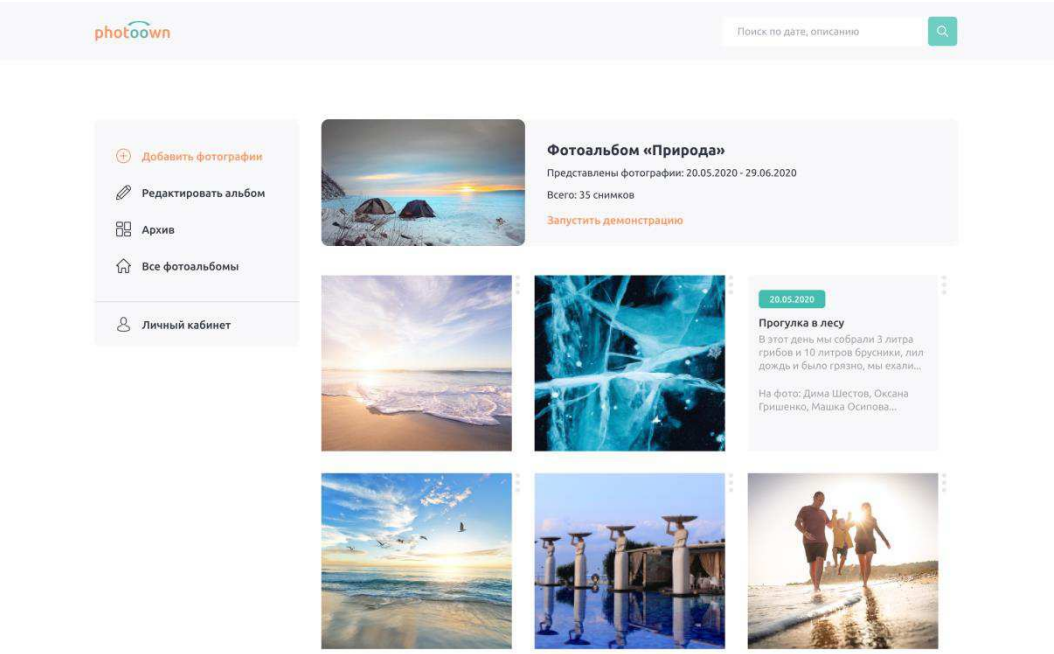

Рисунок 32 – Альбом с фотографиями

Модальное окно для добавления описательных характеристик содержит промежуток дат, чтобы отметить, когда была сделана фотография. Есть возможность добавить заголовок, отметить людей и добавить описательных характеристики в любом объеме (рисунок 33).

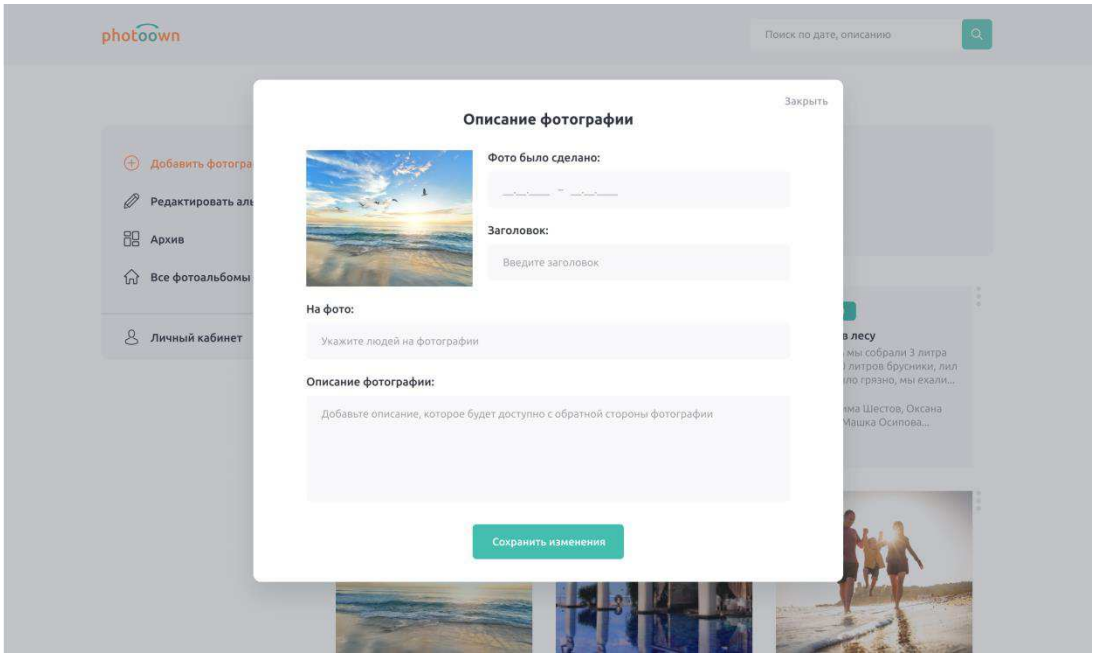

Рисунок 33 – Описание фотографии

Далее представлен запуск демонстрации. Демонстрация задается двумя характеристиками, такими как скорость перелистывания снимков от 3 до 7 секунд, а также показ снимков с начала или с конца (рисунок 34). Запущенная демонстрация представлена ниже (рисунок 35).

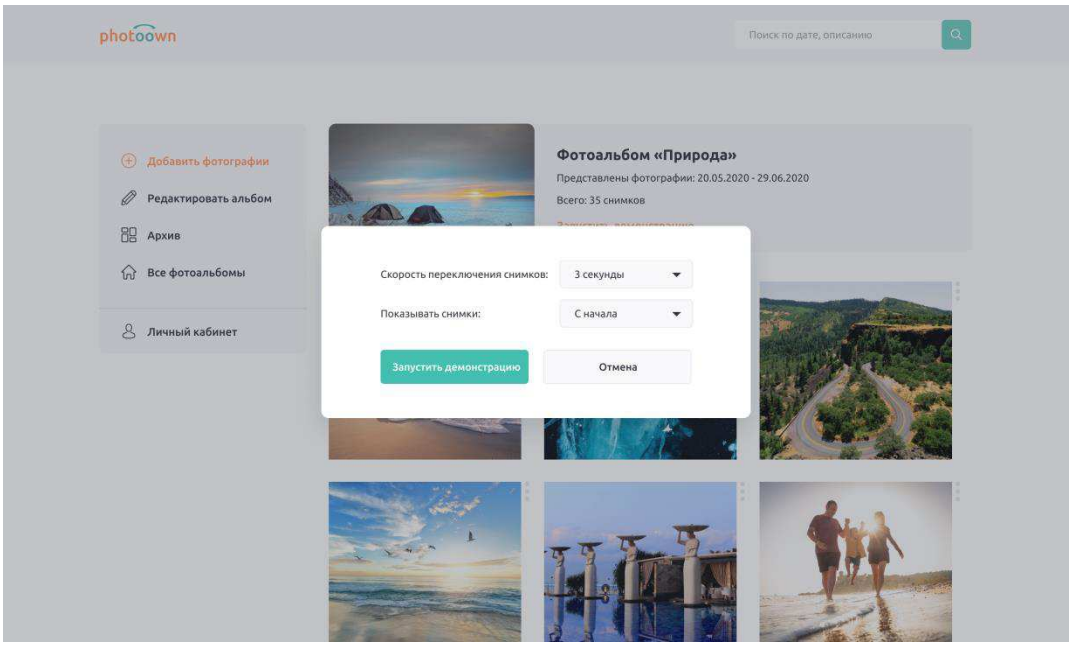

Рисунок 34 – Запуск демонстрации фотографий

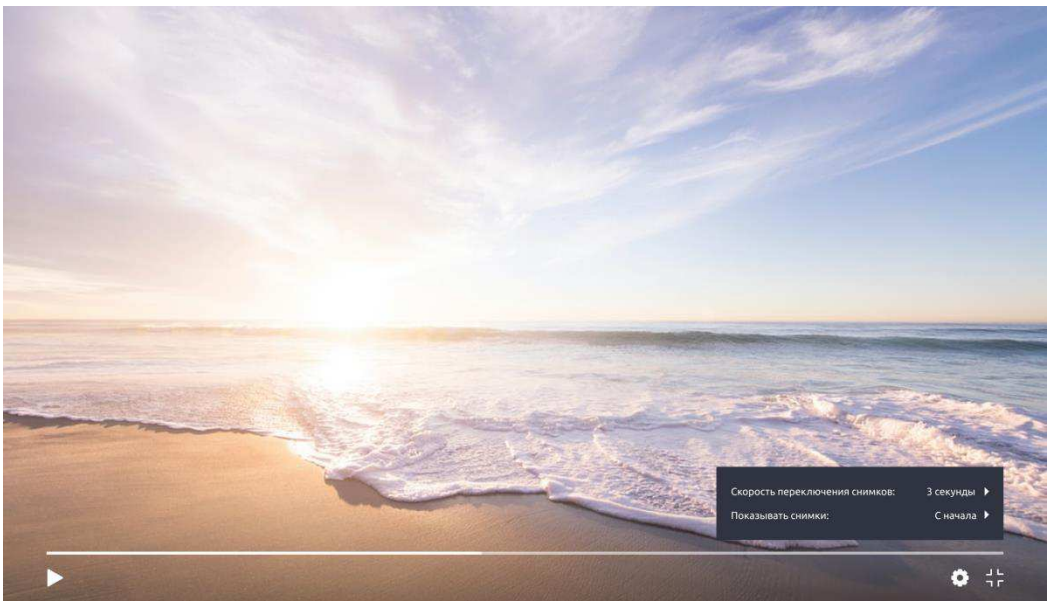

Рисунок 35 – Демонстрация фотографий

4.2 Выводы

Целью данного раздела являлось описание ключевых моментов реализованного программного продукта. Были продемонстрированы скриншоты интерфейсов клиентской части веб-сервиса «Электронный фотоальбом». А именно, показаны последовательные шаги взаимодействия пользователя и веб-сервиса. Описаны все элементы данных страниц, их функционал и возможности.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы было проведено изучение предметной области и анализ существующих решений разработанного веб-сервиса, была определена актуальность проекта. Создание продукта заключалось в формировании и документировании требований к системе, в выборе инструментов разработки, в проектировании интерфейсов и разработке клиентской части программного продукта.

При создании программного продукта были получены навыки в работе с командой посредством одной из разновидностей Agile методологии – Kanban. Со стороны технической части была изучена и применена библиотека React, которая позволила использовать все возможности языка программирования JavaScript. Также для данного проекта была изучена библиотека для JavaScript – Redux, что помогло упростить передачу данных хранилища через контекст.

Результатом ВКР послужил веб-сервис, который представляет замены бумажных фотоальбомов с дополнительными возможностями, которые невозможно предусмотреть на бумажных носителях фотографий. Разработанный программный продукт позволяет:

– регистрироваться и авторизоваться;

– изменять данные, введенные при регистрации, в личном кабинете;

– создавать фотоальбомы и добавлять им описательные характеристики;

– добавлять фотографии в созданные фотоальбомы и дополнять фотографии описательными характеристиками;

– осуществлять поиск фотоальбомов и фотографий;

– просматривать фотоальбомы и фотографии;

– просматривать фотографии в режиме «Демонстрация»;

– помещать фотоальбомы и фотографии в «Архив» с возможностью их восстановления.

Помимо того, что все поставленные задачи были выполнены, проект имеет дальнейшие перспективы развития, которые включают в себя следующие улучшения и дополнительный функционал:

– регистрация пользователей с помощью различных социальных сетей или сервисов, к примеру, регистрация с помощью уже созданного аккаунта в google, facebook, mail.ru или vkontakte;

– динамическое отображение фотографий по их размерам;

– синхронизация с уже действующими социальными сетями или сервисами;

– адаптивность к мобильным устройствам.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Определение целей и задач WEB- сайта [Электронный ресурс]. URL: https://studopedia.su/10_50432_opredelenie-tseley-i-zadach-WEB--sayta.html (Дата обращения 20.04.2021).

2. Тенденции развития информационных технологий [Электронный ресурс]. URL: https://scienceforum.ru/2018/article/2018005029 (Дата обращения 20.04.2021).

3. Определение и применение [Электронный ресурс]. URL: http://www.webtec.com.ua/articles/index/view/2011-03-31/web-site (Дата обращения 20.04.2021).

4. История альбомной культуры [Электронный ресурс]. URL: https://medium.com/russian/история-альбомной-культуры-76cb9f7a6a68 (Дата обращения 20.04.2021).

5. Вахрушева М.Ю., Евдокимов И.В. Разработка программного обеспечения аналитических информационных систем // Труды Братского государственного университета. Серия: Экономика и управление. 2014. Т. 1. № 1. С. 196-199.

6. Создать онлайн фотоальбом Различные виды ссылок Интеграция альбома в сайт [Электронный ресурс]. URL: https://www.photo-pick.com/ru/ (Дата обращения 20.04.2021).

7. Vue vs React vs Angular: какой фронтенд-фреймворк выбрать? [Электронный ресурс]. URL: https://proglib.io/p/vue-vs-react-vs-angular-kakoy-frontend-freymvork-vybrat-2020-08-03 (Дата обращения 30.05.2021).

8. A JavaScript library for building user interfaces [Электронный ресурс]. URL: https://reactjs.org/ (Дата обращения 20.04.2021).

9. HTML5 JavaScript Component Suite for Responsive Web Development: [Электронный ресурс]. URL: https://js.devexpress.com. (Дата обращение 20.04.2021).

10. Что такое Trello и как им пользоваться [Электронный ресурс]. URL: https://netology.ru/blog/trello (Дата обращения 30.05.2021).

11. Гики и Git-ы. Почему Git так популярен [Электронный ресурс]. URL: https://gb.ru/posts/why_git (Дата обращения 30.05.2021).

12. Git и GitHub: что это такое и в чём разница [Электронный ресурс]. URL: https://tproger.ru/translations/difference-between-git-and-github/ (Дата обращения 30.05.2021).

13. Что такое Git? [Электронный ресурс]. URL: https://www.atlassian.com/ru/git/tutorials/what-is-git (Дата обращения 30.05.2021).

14. Гурьянов Л. В., Дзюба Е. А., Кульков И. В. GitLab - современный подход к совместной работе студентов, преподавателей и профессиональных программистов // Новые информационные технологии и системы. – 2016. – С. 164-165.

15. Agile или Waterfall — какойвариант соответствуетвашему бизнесу? [Электронный ресурс]. URL: https://worksection.com/blog/waterfall-vs-agile.html (Дата обращения 30.05.2021).

16. Agile, scrum, kanban: в чем разница и для чего использовать? [Электронный ресурс]. URL: https://rb.ru/story/agile-scrum-kanban/ (Дата обращения 30.05.2021).

17. Scrum vs Kanban: в чем разница и что выбрать? [Электронный ресурс]. URL: https://habr.com/ru/company/hygger/blog/351048/ (Дата обращения 30.05.2021).

18. Chen J., Soundararajan G., Amza C. Autonomic provisioning of backend databases in dynamic content web servers //Autonomic Computing, 2006. ICAC'06. IEEE International Conference on. – IEEE, 2006. – С. 231-242.

19. Шилдт, Герберт. Java 8: руководство для начинающих, 6-е изд.: Пер. с англ. - М. ООО "И.Д. Вильямс", 2015. - 720 с.: ил. - Парал. тит. англ.

20. Вахрушева М.Ю., Евдокимов И.В. Разработка программного обеспечения аналитических информационных систем // Труды Братского

государственного университета. Серия: Экономика и управление. 2014. Т. 1. № 1. С. 196-199.

21. Бегг К. и др. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. (Database Systems. A Practical Approach to Design, Implementation, and Management). – 2000.

22. Архитектура информационной системы [Электронный ресурс]. URL:https://spravochnick.ru/bazy_dannyh/bazy_dannyh_vvedenie/arhitektura_informacionnoy_sistemy/ (Дата обращения 30.05.2021).

23. UML для бизнес-моделирования: зачем нужны диаграммы процессов [Электронный ресурс]. URL: https://evergreens.com.ua/ru/articles/uml-diagrams.html (Дата обращения 30.05.2021).

24. Blanchard B. S., Fabrycky Wolter J. Systems engineering and analysis. — 4- е изд. — Prentice Hall, 2006

25. Паттерны поведения пользователей [Электронный ресурс]. URL: https://habr.com/ru/post/337382/ (Дата обращения 30.05.2021).

26. Краткое руководство по Redux для начинающих [Электронный ресурс]. URL: https://tproger.ru/translations/redux-for-beginners/ (Дата обращения 30.05.2021).

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра «Информатика»

УТВЕРЖДАЮ
Заведующий кафедрой

_____ А. С. Кузнецов
подпись инициалы, фамилия

«18» 06 2021 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.04 Программная инженерия

Разработка клиентской части веб-сервиса «Электронный фотоальбом»

Руководитель ВКР 17.06.21 доцент, канд. техн. наук А. В. Хныкин
подпись, дата должность, ученая степень инициалы, фамилия

Выпускник , 17.06.21. А. Н. Конюхова
подпись, дата инициалы, фамилия

Красноярск 2021