

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»

БИРСКИЙ ФИЛИАЛ УУН<sub>ИТ</sub>  
ФАКУЛЬТЕТ ФИЗИКИ И МАТЕМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ И ЭКОНОМИКИ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ПО ПРОГРАММЕ БАКАЛАВРИАТА

ХУСАИНОВ РЕНАТ МАРАТОВИЧ

ЦИФРОВАЯ ГАЛЕРЕЯ

Выполнил:  
Студент 4 курса очной формы  
обучения  
Направление подготовки  
09.03.03 Прикладная информатика  
Направленность (профиль)  
Прикладная информатика в  
информационной сфере

Руководитель  
к.физ-мат.н., доцент  
Ю.А. Тазетдинова

БИРСК – 2023

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	3
ГЛАВА 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ .....	4
1.1. Обзор деятельности организации.....	4
1.2. Структурно-функциональная диаграмма бизнес-процесса организации по типу «КАК ЕСТЬ» и ее описание .....	4
1.3. Анализ существующих разработок и применяемого программного обеспечения.....	9
1.4. Анализ затрат, выгод и рисков .....	12
Выводы по главе 1 .....	15
ГЛАВА 2. ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	16
2.1. Разработка структурно-функциональной диаграммы по типу «Как должно быть?» .....	16
2.2. Описание задач автоматизации информационных процессов поведенческими диаграммами в нотации UML.....	20
2.3. Формулирование технического задания .....	24
2.3.1. Общие сведения .....	24
2.3.2. Назначение и цели создания системы .....	24
2.5. Проектирование Базы Данных.....	27
2.6. Проектирование интерфейса .....	29
Выводы по главе 2 .....	30
ГЛАВА 3. РЕАЛИЗАЦИЯ КОМПОНЕНТОВ ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	31
3.1. Реализация front-end части информационной системы .....	31
3.2. Реализация back-end части информационной системы.....	33
3.3. Реализация базы данных .....	37
3.4. Руководство пользователя .....	38
Выводы по главе 3 .....	39
ЗАКЛЮЧЕНИЕ .....	40
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ.....	41
ПРИЛОЖЕНИЕ .....	<b>Ошибка! Закладка не определена.</b>

## ВВЕДЕНИЕ

Ещё 15–20 лет назад нельзя было предположить, что веб-приложения станут неотъемлемой частью жизни. Сегодня эта разновидность онлайн-инструментов используется для различных задач, включая оптимизацию бизнес-процессов, продажу товаров и услуг, распространение информации, общение пользователей друг с другом.

Веб-приложение представляет собой веб-сайт, на котором размещены страницы с частично либо полностью несформированным содержанием. Окончательное содержание формируется только после того, как посетитель сайта запросит страницу с веб-сервера. В связи с тем, что окончательное содержание страницы зависит от запроса, созданного на основе действий посетителя, такая страница называется динамической.

Любой информационный процесс может быть автоматизирован с помощью веб-приложения. Оно имеет неоспоримый ряд преимуществ: удобство развертывания, надежность, доступность, кроссплатформенность и функциональность. По этим причинам в данной работе оно было выбрано в качестве средства автоматизации.

Объект исследования – выставочная деятельность художественной галереи.

Предмет исследования – автоматизация информационных процессов сбора, хранения и обработки информации об объектах галереи.

Цель исследования – разработка прототипа информационной системы, осуществляющей автоматизацию процессов сбора, хранения и обработки информации об объектах галереи.

Задачи:

- Провести анализ предметной области;
- Спроектировать компоненты ИС;
- Разработать компоненты ИС;

# **ГЛАВА 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

## **1.1. Обзор деятельности организации**

В качестве исследуемого учреждения для реализации проектного решения была выбрана галерея Бирского филиала Уфимского университета науки и технологий, которая ведет свою деятельность 52 года, начиная с 1971 года.

Основной вид деятельности – выставка произведений искусств студентов и художников из других регионов и стран.

Галерея располагается по адресу Республика Башкортостан, г. Уфа, ул. Давлетшиной, д. 18.

Основные цели художественной галереи – поддержание преемственности художественного творчества, культурно-просветительская работа среди населения, сохранение культурной среды города Бирска.

Непосредственно для данного данной галереи будет предназначена разрабатываемая информационная система.

## **1.2. Структурно-функциональная диаграмма бизнес-процесса организации по типу «КАК ЕСТЬ» и ее описание**

Исследуем информационные процессы предметной области в нотации IDEF0. Рассмотрим процесс "Организация выставки картин".

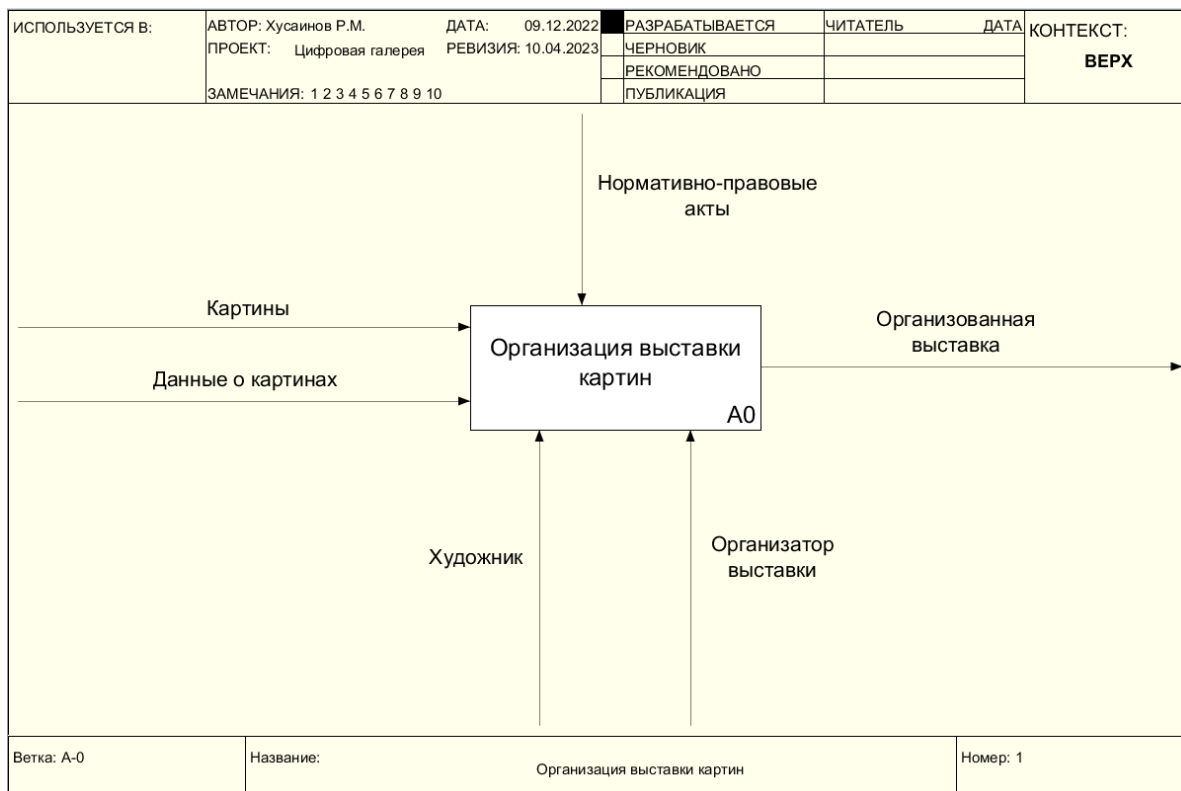


Рисунок 1.1 Контекстная диаграмма в нотации IDEF0, модель «Как есть» информационного процесса «Организация выставки картин»

Нотация IDEF0 предназначена для формализации и описания бизнес – процессов. С помощью средств данной нотации была создана контекстная диаграмма (Рисунок 1.1), направленная на отражение процесса создания картины и организации ее выставки.

На представленной контекстной диаграмме можно выявить следующие типы стрелок:

**Вход** – картины и данные о картинах. Они представляют собой объекты, которые будут использованы или преобразованы для получения результата (выхода);

**Выход** – организованная выставка, это объект, в который преобразуются входы;

Управление – нормативно-правовые акты, регулирующие выставочную деятельность. Они представляет собой правила и требования, регламентирующие правомерность организации выставки.

Механизм – художник, автор выставляемых произведений, и организатор выставки, непосредственно участвующий в организации данной конкретной выставки.

В настоящее время создание и выставка картины происходит следующим образом:

1. Художник обращается в галерею для выставления своих картин. Назначается организатор выставки
2. Происходит планирование выставки, формируется план-конспект. План-конспект имеет свою структуру, вмещающую краткое описание всех составляющих данного процесса:
  - цели выставки;
  - впечатления, которые ожидается получить от посетителей;
  - пошаговое описание выставки с указанием коммуникативных задач и средств для их решения, направлений движения потока посетителей, эскизов экспозиции.
3. Выставка афишируется кругу лиц для привлечения посетителей.
4. Совершается подготовка к вернисажу, то есть торжественному открытию выставки. На данном этапе происходит монтаж выставки и оценка результатов проведенной работы, при необходимости вносятся коррективы.
5. Проведение выставки.

Данный процесс можно оформить следующей декомпозицией контекстной диаграммы, в которой были выявлены следующие функциональные блоки (Рисунок 1.2):

- 1. Планирование выставки
- 2. Афиширование выставки
- 3. Подготовка к вернисажу

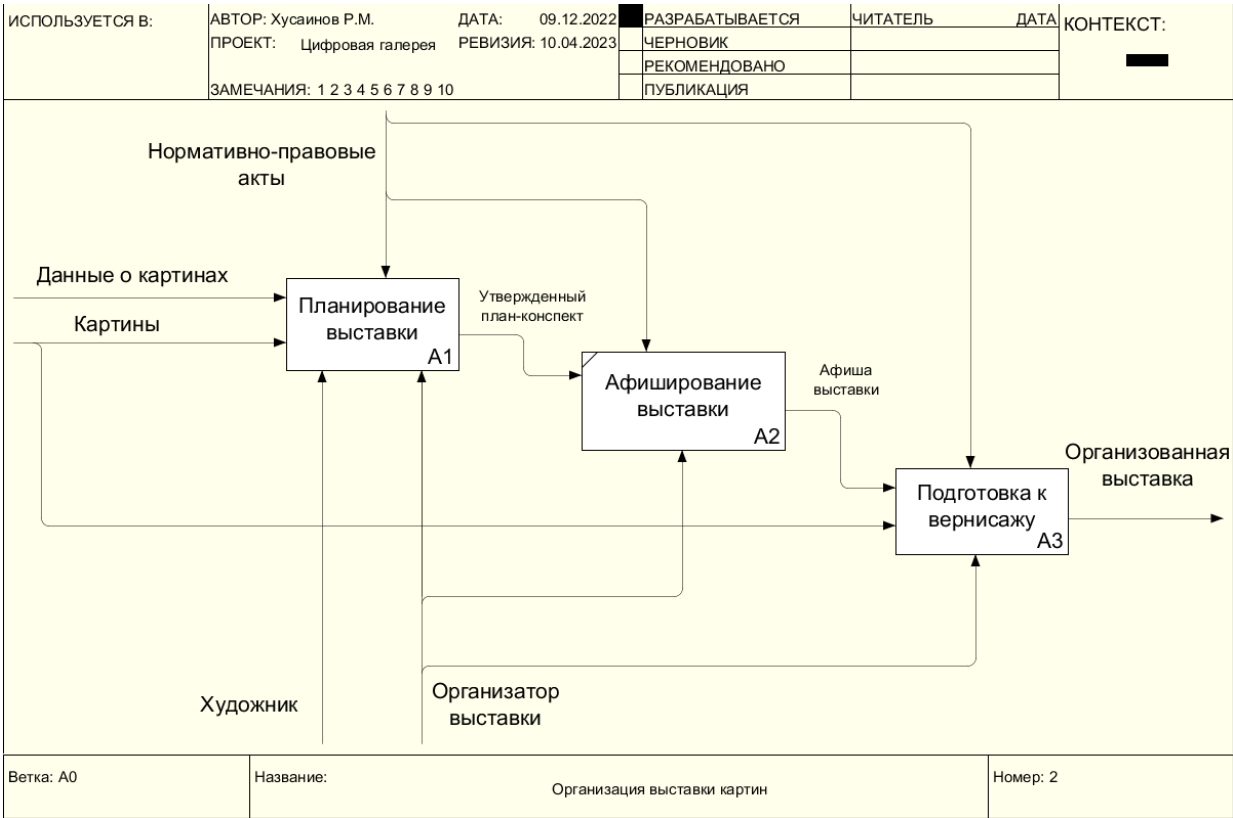


Рисунок 1.2 Результат декомпозиции контекстной диаграммы

Ниже на Рисунке 1.3 представлена иллюстрация процесса планирования выставки. Он происходит в 3 этапа: формирование концепции выставки, составление плана-конспекта и оценки формата выставки.

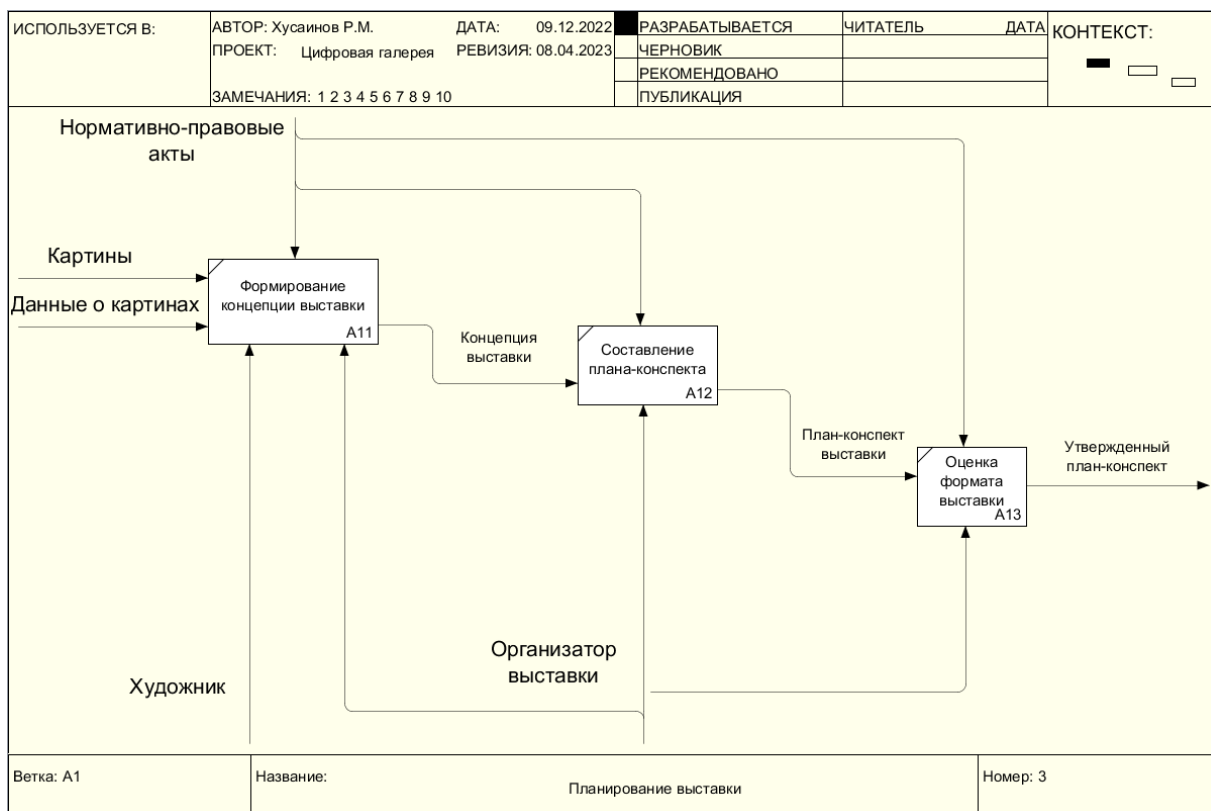


Рисунок 1.3 Диаграмма декомпозиции процесса "Планирование выставки"

В ходе разработки структурно-функциональной диаграммы по типу «Как есть» были выявлены следующие недостатки:

- выставляемые картины художника недоступны широкому кругу лиц;
- публикация картин только в офлайн галерее лишает художника шанса получить большее количество поклонников его творчества;
- отсутствует удобная централизованная площадка, где могли бы афишироваться выставки.



### **1.3. Анализ существующих разработок и применяемого программного обеспечения**

Галереи России преимущественно используют информационные системы, использующие web-технологии (например, web-сайт галереи <https://www.miras.ru/>). Web-сайт галереи "Мирас" был разработан с помощью конструктора сайтов Tilda, это влечет ряд весомых недостатков, свойственных конструкторам сайтов, таких как:

1. дороговизна тарифов,
2. отсутствие масштабируемости,
3. отсутствие изменения backend составляющей сайта,
4. размещение сайта исключительно на хостинге Tilda,
5. Относительно мало места на хостинге.

Вопрос о выборе инструментов разработки для создания информационной системы возникает всегда, т.к. от них зависит скорость и качество разработки. Вышеперечисленных недостатков конструкторов сайтов лишен самостоятельно разработанный продукт.

Для разработки frontend части приложения решено использовать технологии HTML, CSS и JavaScript, которые позволяют реализовать удобный и функциональный пользовательский интерфейс.

Одним из популярных библиотек для разработки интерфейса является React, который имеет неоспоримый ряд преимуществ:

1. построение интерфейса из отдельных компонентов, которые легко поддерживать;

2. добавление удобного слоя абстракции, который избавляет от необходимости работать с DOM (от англ. Document Object Model — «объектная модель документа») напрямую;
3. Благодаря сообществу, у React хорошо проработанная документация и большой опыт, накопленный в статьях, курсах и конференциях. Это значительно облегчает не только изучение библиотеки новичками, но и поиск ответов на всевозможные вопросы в процессе разработки;
4. React — это проект с открытым исходным кодом. Благодаря этому, его можно безопасно использовать даже в коммерческих приложениях.

Неотъемлемой частью информационной системы является ее серверная, backend составляющая, позволяющая реализовать логику работы приложения при взаимодействии пользователя с системой. Для ее реализации была выбрана программная платформа NodeJS, основанная на JavaScript-движке V8. V8 – это движок с открытым исходным кодом, который написан на C++. NodeJS превращает JavaScript из узкоспециализированного языка в язык общего назначения, позволяя реализовать сервер составляющую web-приложения.

Взаимодействие пользовательского интерфейса с серверной частью основано на протоколе HTTP. Используются конечные точки (endpoint), по которым клиентская часть приложения обращает свои запросы к программному интерфейсу приложения (API).

Для управления содержимым информационной системы используется система Strapi. Она представляет собой фреймворк для управления контентом, работающий на Node.js. Это полностью бесплатный проект с открытым исходным кодом. Система разворачивается локально на собственном сервере, что обеспечивает безопасность данных.

Главные особенности и преимущества Strapi:

- Открытый исходный код. Система разработана энтузиастами и поддерживается сотнями участников GitHub, которые развивают ее в соответствии с новыми требованиями и технологиями. Она всегда будет доступна и бесплатна.
- Широкие и гибкие настройки. Панель администратора, как и API, настраиваются достаточно гибко. Возможно расширение функционала за счет плагинов.
- RESTful или GraphQL. CMS поддерживает передачу данных посредством таких архитектурных решений как REST, так и GraphQL. Это расширяет возможности взаимодействия с разными клиентами, мобильными приложения, IoT-устройствами.
- Локальное размещение. Размещение на собственном сервере владельца системы гарантирует конфиденциальность и обеспечивает повышенный уровень защиты данных.
- Один язык. Система использует JavaScript, что позволяет работать с одним языком как в сервер составляющей, так и во фронтенде.

## **Выбор СУБД**

В качестве СУБД была выбрана SQLite.

База данных — это набор структурированной информации. Для ее изменения требуются системы управления — СУБД. Как и любая СУБД, SQLite позволяет записывать новую и запрашивать существующую информацию, изменять ее, настраивать доступ.

Благодаря своим свойствам, SQLite применяется:

- на сайтах с низким и средним трафиком;

- в локальных однопользовательских, мобильных приложениях или играх, не предназначенных для масштабирования;
- в программах, которые часто выполняют прямые операции чтения/записи на диск;
- в приложениях для тестирования бизнес-логики.

#### 1.4. Анализ затрат, выгод и рисков

Каждый этап разработки требует времени. Для оценки трудоемкости разработки в Таблице 1.1 указаны ориентировочные затраты времени для каждого этапа. Общая трудоемкость складывается из суммарного количества часов каждого этапа.

Таблица 1.1  
Трудоемкость разработки по этапам

№	Название этапа	Количество часов, ч
1.	Анализ предметной области	32
2.	Проектирование веб-приложения	96
3.	Разработка веб-приложения	85
4.	Тестирование и отладка	32
	Общая трудоемкость:	245

В соответствии с данными таблицы, общая трудоемкость  $T = 245$  ч.

#### Расчет заработной платы

Заработная плата рассчитывается исходя из размера часовой заработной платы специалиста, которая в нашем случае составляет 200 руб/ч., тогда:

$$\text{Заработная плата} = 200 \cdot 245 = 49\,000 \text{ руб.}$$

На заработную плату делаются начисления в размере 30% от начисленной заработной платы.

$$\text{НЗП} = 49\,000 \cdot 0.3 = 14\,700 \text{ руб.}$$

Итоговая заработная плата разработчика: 49 000 руб. + 14 700 руб. = 63 700 руб.

### **Расчет затрат на электроэнергию**

Потребление электричества компьютером составляет примерно 220 Ватт в час. Если компьютер работает сутки, то расход 5,28 кВт\*ч за сутки.

За год расход на электроэнергию на компьютер менеджера с округлением 1927 кВт\*ч в год.

Цена электроэнергии для юридических лиц 2,81 руб. за 1кВт\*ч.

Итого 5 415 руб. за год.

### **Хостинг**

Для размещения проекта АИС в сети интернет необходимо приобретения хостинга. При низко нагруженном проекте подходящим вариантом будет виртуальный хостинг, стоимостью примерно 200 руб. в месяц.

Покупка имени домена 500 руб. / год.

### **Цена оргтехники**

Цена компьютера, необходимого для работы администратора ИС, примерно равна 30 000 руб.

### **Расчет итоговых затрат**

В Таблице 1.2 приводятся все затраты на разработку и эксплуатацию информационной системы «Цифровая галерея» за 1 год работы.

Расчеты являются примерными и не учитывают многих нюансов, вроде амортизационных отчислений, ремонта и обслуживание компьютера и др.

Таблица 1.2

Затраты на разработку и эксплуатацию ИС	Стоимость, руб.
Заработная плата разработчика	63 700
Электроэнергия	5 415
Домен	500
Хостинг	2 400
Оргтехника	30 000
<b>Общие затраты:</b>	102 015

В результате расчетов за траты на создание составили 102 015 рублей, что ниже.

Практически одну треть стоимости составляет стоимость компьютерной техники. Но она как правило уже есть в организациях, даже если они не используют ее для автоматизации бизнес-процессов.

### **Риски**

Внедрение АИС несет в себе некоторые риски. В частности, риски потери данных при сбое в программном обеспечении (сбой хостинга, атака на сайт, вирус). Для минимизации данных рисков необходимо регулярное копирование данных для их резервного хранения.

## **Выводы по главе 1**

Проведен анализ предметной области – художественная галерея Бирского филиала УУНиТ. С помощью диаграмм IDEF0 описаны существующие бизнес-процессы. Приведены примерные расчеты затрат и рисков внедрения ИС «Цифровая галерея».

## ГЛАВА 2. ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

### 2.1. Разработка структурно-функциональной диаграммы по типу «Как должно быть?»

По ходу разработки бизнес-процесса была построена структурно-функциональная диаграмма «Как должно быть?». Был добавлен новый ресурс – информационная система (Рисунок 2.1). Добавлены новые ресурсы: механизм «Информационная система» и управление «Администратор информационной системы» (выделены красным), благодаря которым повысится эффективность организации.



Рисунок 2.1 Контекстная диаграмма информационного процесса «Организация выставки картин», модель «Как должно быть?»



Декомпозиция контекстной диаграммы включает в себя следующие функциональные блоки (Рисунок 2.2):

1. Планирование выставки
2. Афиширование выставки
3. Подготовка к вернисажу

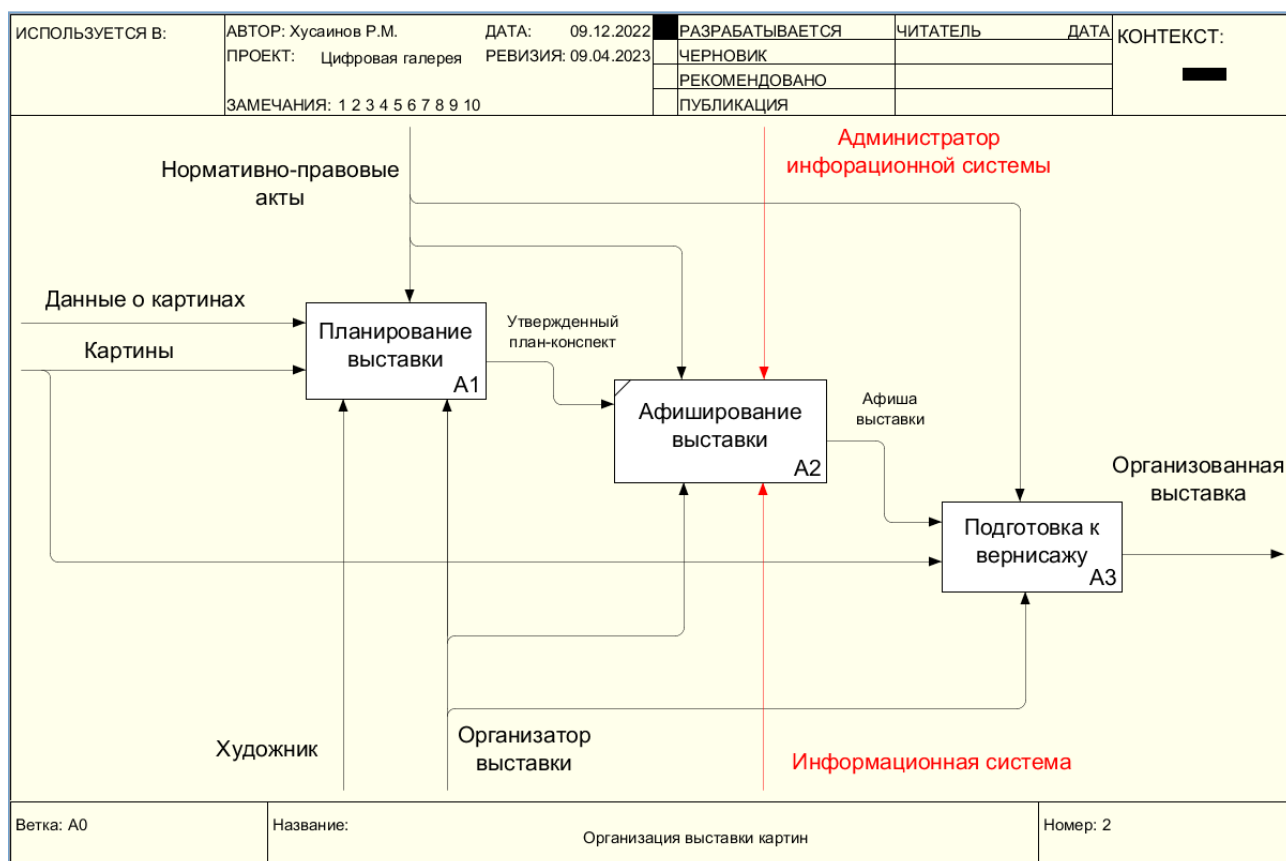


Рисунок 2.2 Диаграмма декомпозиции информационного процесса “Организация выставки картин”, модель "Как должно быть?"

На основании анализа существующей процедуры организации выставки, можно сделать вывод о необходимости ее улучшения путем автоматизации этого процесса. Применение специализированной информационной системы нацелено на решение данной задачи.

При анализе существующего процесса организации выставки была выявлена потребность заказчика в том, чтобы проведенные выставки были

доступны пользователям системы в качестве архивированных выставок, содержащих отчет о проведенном мероприятии.

Для этого необходимо рассмотреть процесс "Проведение выставки" (Рисунок 2.3).

Вход – "Организованная выставка". Это объект, который был получен в процессе "Организации выставки картин".

Выход – "Проведенная выставка", означает окончание проведения выставки. Информационная система в данном случае добавляет еще один объект – "Архивированная выставка". Администратор ИС по окончании выставки посредством панели управления контентом добавляет фотоотчет к существующей выставке, которая является уже архивированной в ИС по истечению времени проведения выставки.

Управление – нормативно-правовые акты, регулирующие выставочную деятельность. Они представляет собой правила и требования, регламентирующие правомерность организации выставки.

Механизм – художник, автор выставляемых произведений, организатор выставки, непосредственно участвующий в организации данной конкретной выставки, и информационная система.



Рисунок 2.3 Контекстная диаграмма информационного процесса “Проведение выставки”, модель "Как должно быть?"

## 2.2. Описание задач автоматизации информационных процессов поведенческими диаграммами в нотации UML

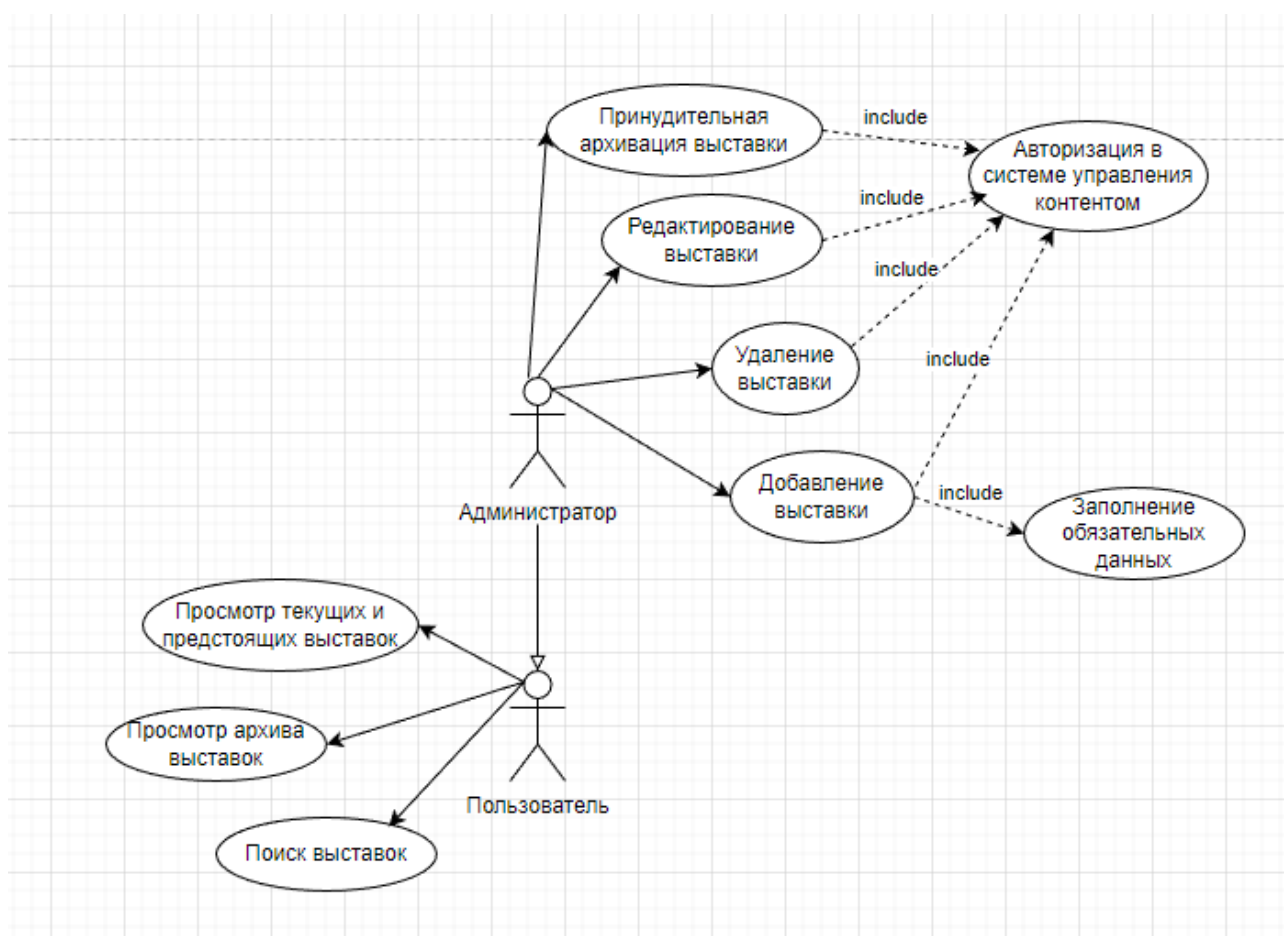


Рисунок 2.4 Диаграмма вариантов использования ИС «Цифровая галерея»

Диаграмма прецедентов или диаграмма вариантов использования – это диаграмма, на которой отражаются отношения между актерами и прецедентами системы, позволяющая описывать систему на концептуальном уровне.

Прецедент – это возможность системы, часть ее функциональности, с помощью которой актер может получить нужный ему результат. Прецедент является соответствием для отдельного сервиса системы и определяет варианты ее использования.

На рисунке 2.4 представлена диаграмма вариантов использования, где описаны действия, возможные для акторов «Администратор» и «Пользователь».

Наследование «Администратором» действий «Пользователя» показано через обобщение.

Рисунки 2.5 и 2.6 содержат диаграммы последовательностей для существующих прецедентов в информационной системе. Они уточняют диаграммы прецедентов, более детально описывают логику сценариев использования.

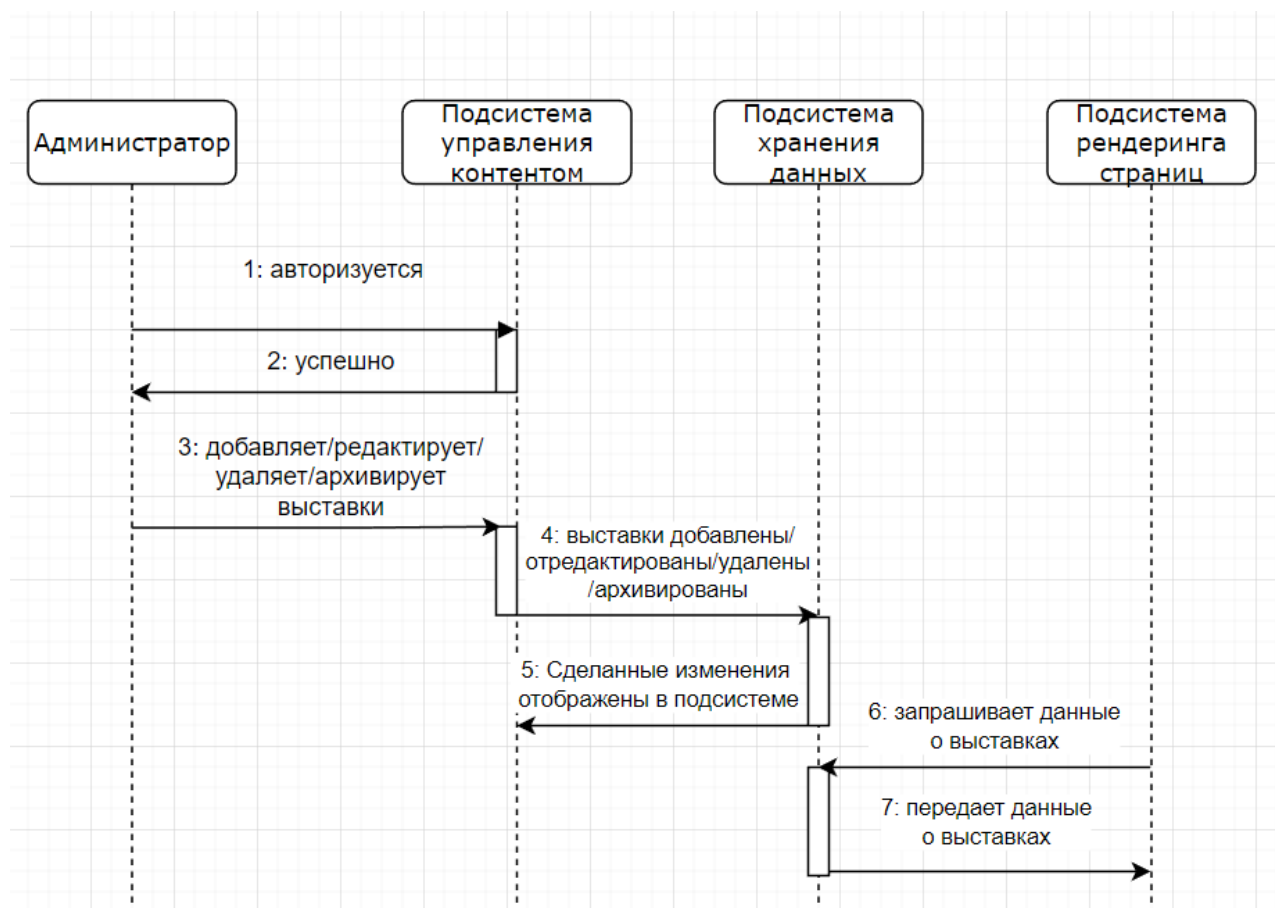


Рисунок 2.4 Диаграмма последовательностей для прецедентов актора «Администратор»



Рисунок 2.5 Диаграмма последовательностей для прецедента поиска актора «Пользователь»

Для моделирования динамических аспектов системы были использованы диаграммы состояний, представленные на рисунках 2.6 и 2.7. Данные диаграммы полезны при моделировании жизненного цикла объекта. От других диаграмм диаграмма состояний отличается тем, что описывает процесс изменения состояний только одного экземпляра определенного класса - одного объекта, причем объекта реактивного, то есть объекта, поведение которого характеризуется его реакцией на внешние события.

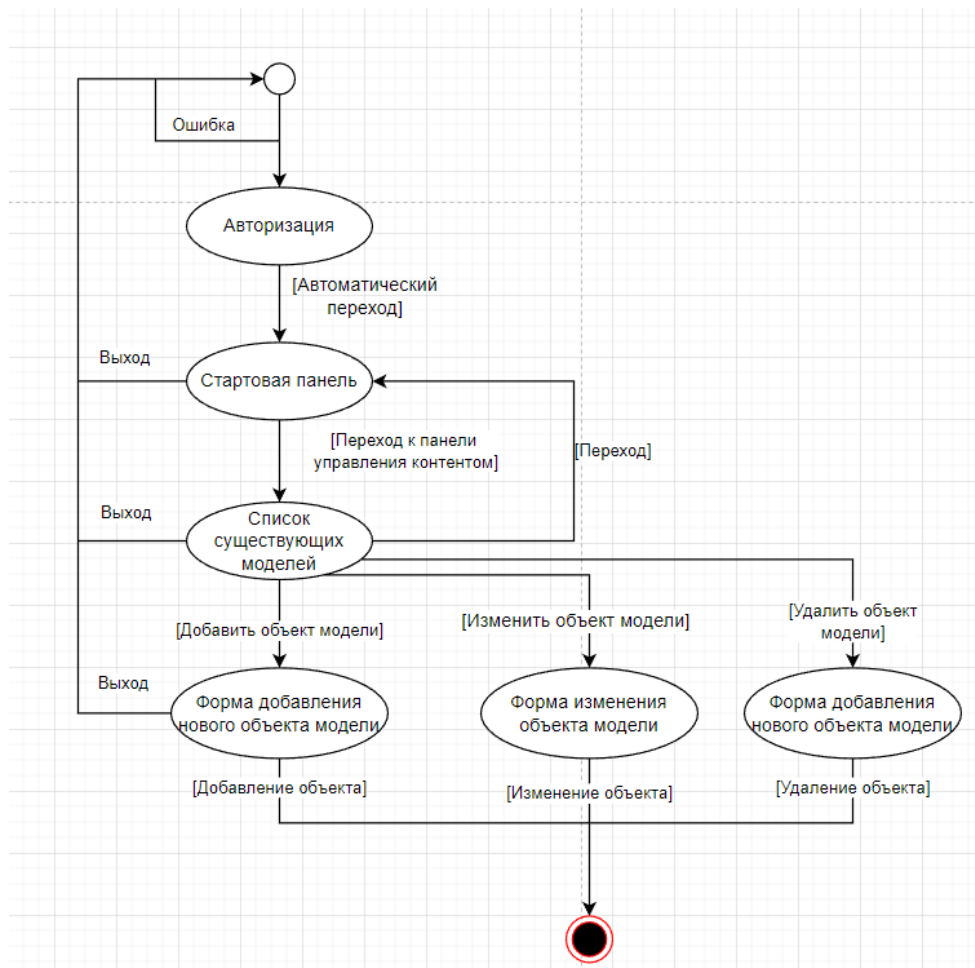


Рисунок 2.6 Диаграмма состояний подсистемы управления контентом для прецедентов Администратора.

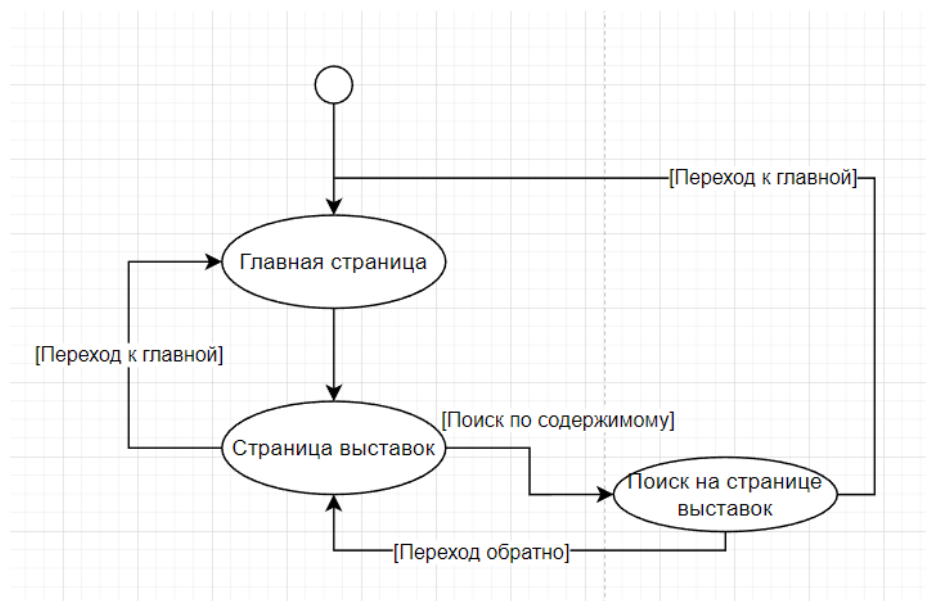


Рисунок 2.7 Диаграмма состояний системы для прецедента «Поиск» актора «Пользователь».

## **2.3. Формулирование технического задания**

### **2.3.1. Общие сведения**

Полное наименование системы: Информационная система «Цифровая галерея».

Краткое наименование: «Цифровая галерея».

### **2.3.2. Назначение и цели создания системы**

Система предназначена для предоставления пользователю доступа к информации о выставках галереи посредством веб-сайта.

В рамках проекта автоматизируется информационно-аналитическая деятельность в следующих бизнес-процессах:

1. Хранение данных о текущих, прошедших и предстоящих выставках галереи;
2. Предоставление доступа к произведениям изобразительного искусства и выставкам, путем их публикации в информационной системе;
3. Афиширование выставок.

Предусматривается установка системы в пространствах, предназначенных для демонстрации изобразительного искусства.

ИС «Цифровая галерея» создается с целью:

- обеспечения удобного доступа к произведениям и выставкам галереи;



- создания единой системы хранения цифровых копий произведений изобразительного искусства, выставляющихся в галереях;
- предоставить возможность художникам осуществлять показ выполненных работ широкому кругу любителей искусства.

В результате создания информационной системы должны быть улучшены значения следующих показателей:

- посещаемость галереи;
- количество просмотров произведений изобразительного искусства галереи;
- доступность искусства и информации о выставках.

#### **2.4. Описание информационной модели комплекса задач (схема потоков данных)**

Диаграмма DFD наглядно отображает течение информации в пределах процесса или системы. Для изображения входных и выходных данных, точек хранения информации и путей ее передвижения между источниками и пунктами доставки в таких диаграммах применяются стандартные фигуры, такие как прямоугольники и круги, а также стрелки и краткие текстовые метки. На рисунке 12 представлена контекстная диаграмма потоков данных в нотации Гейна-Сарсона. В ней «Пользователь» является внешней сущностью. При обращении к системе выступает как источник данных, а при получении данных как адресат.

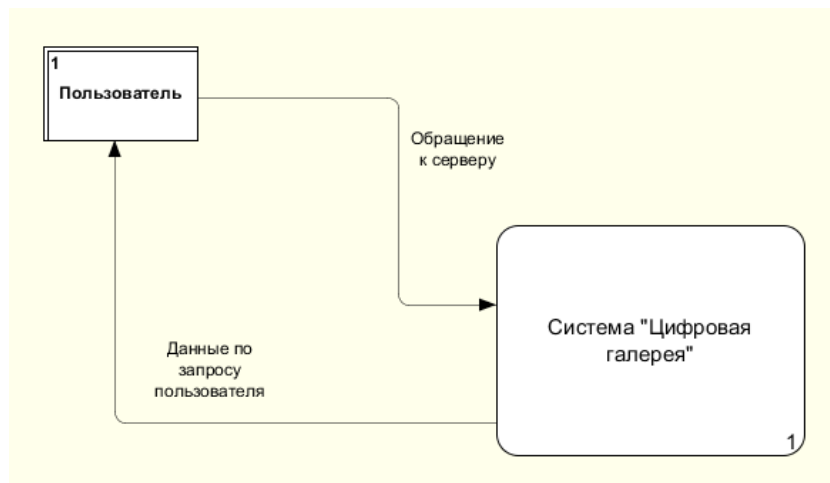


Рисунок 2.8 Контекстная диаграмма потоков данных в нотации Гейна-Сарсона

Декомпозиция контекстной диаграммы, что представлена на Рисунке 2.9, детализирует процессы, происходящие при взаимодействии «Пользователя» и «Системы». Наглядно демонстрируя поток данных в системе.

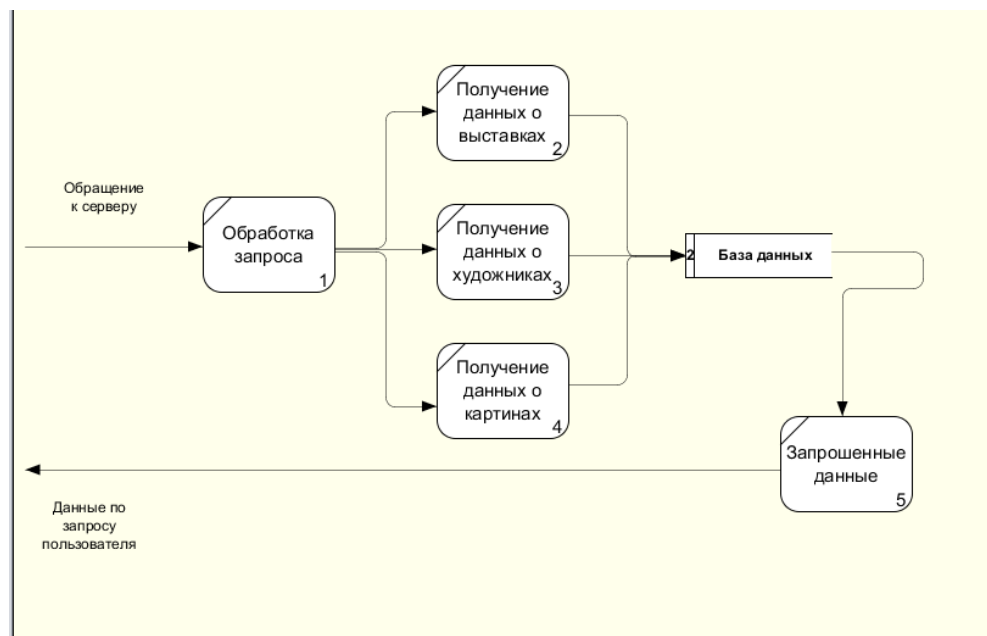


Рисунок 2.9 Декомпозиция контекстной диаграммы

## 2.5. Проектирование Базы Данных

Для создания UML-диаграмм использовалась программа в виде веб-приложения [app.diagrams.net](http://app.diagrams.net). На рисунке 2.10 изображена инфологическая ER-модель в нотации Питера Чена, включающая основные сущности и связи между ними.

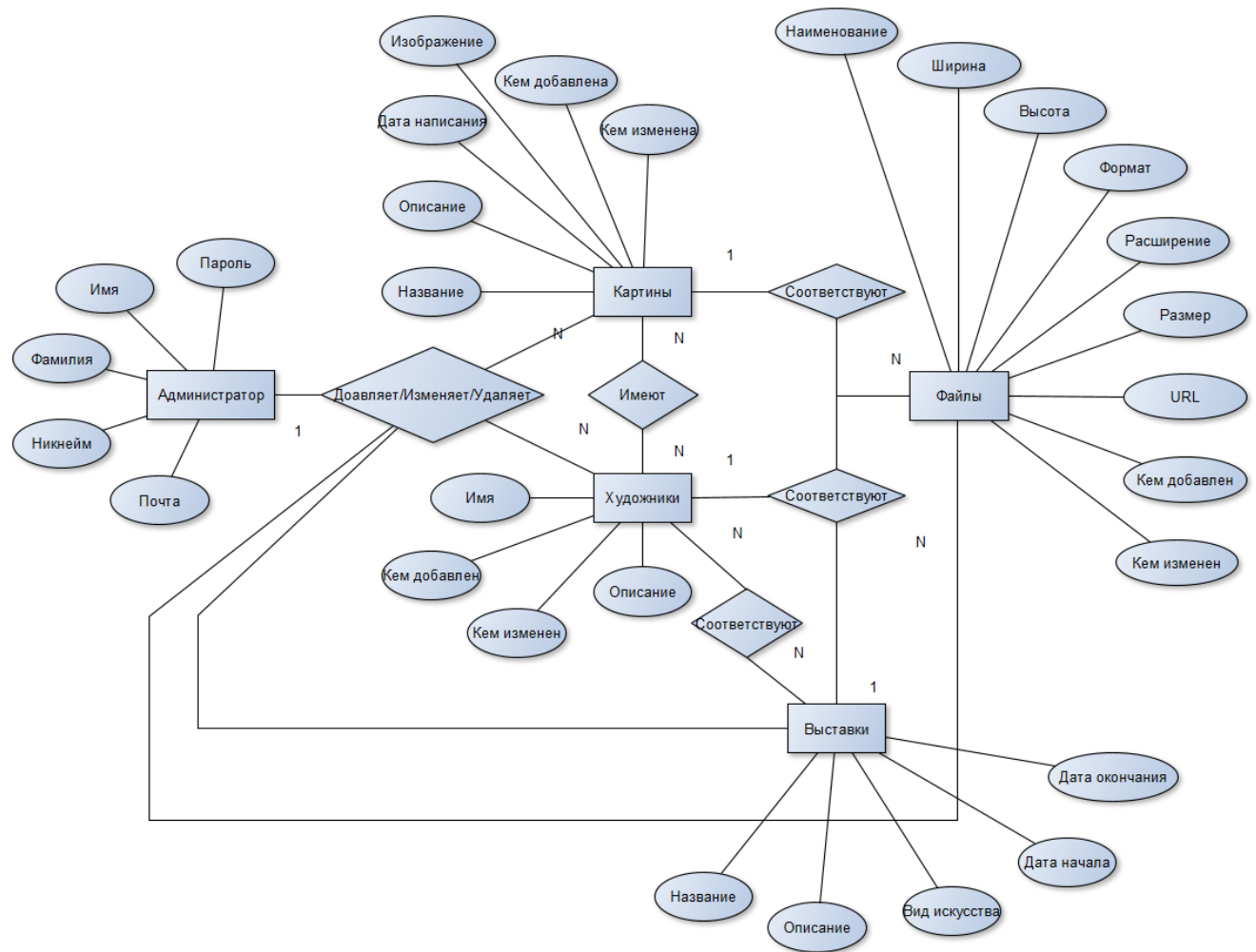


Рисунок 2.10 Инфологическая модель БД в нотации Питера Чена

На Рисунке 2.11 представлена даталогическая модель базы данных разрабатываемой системы.

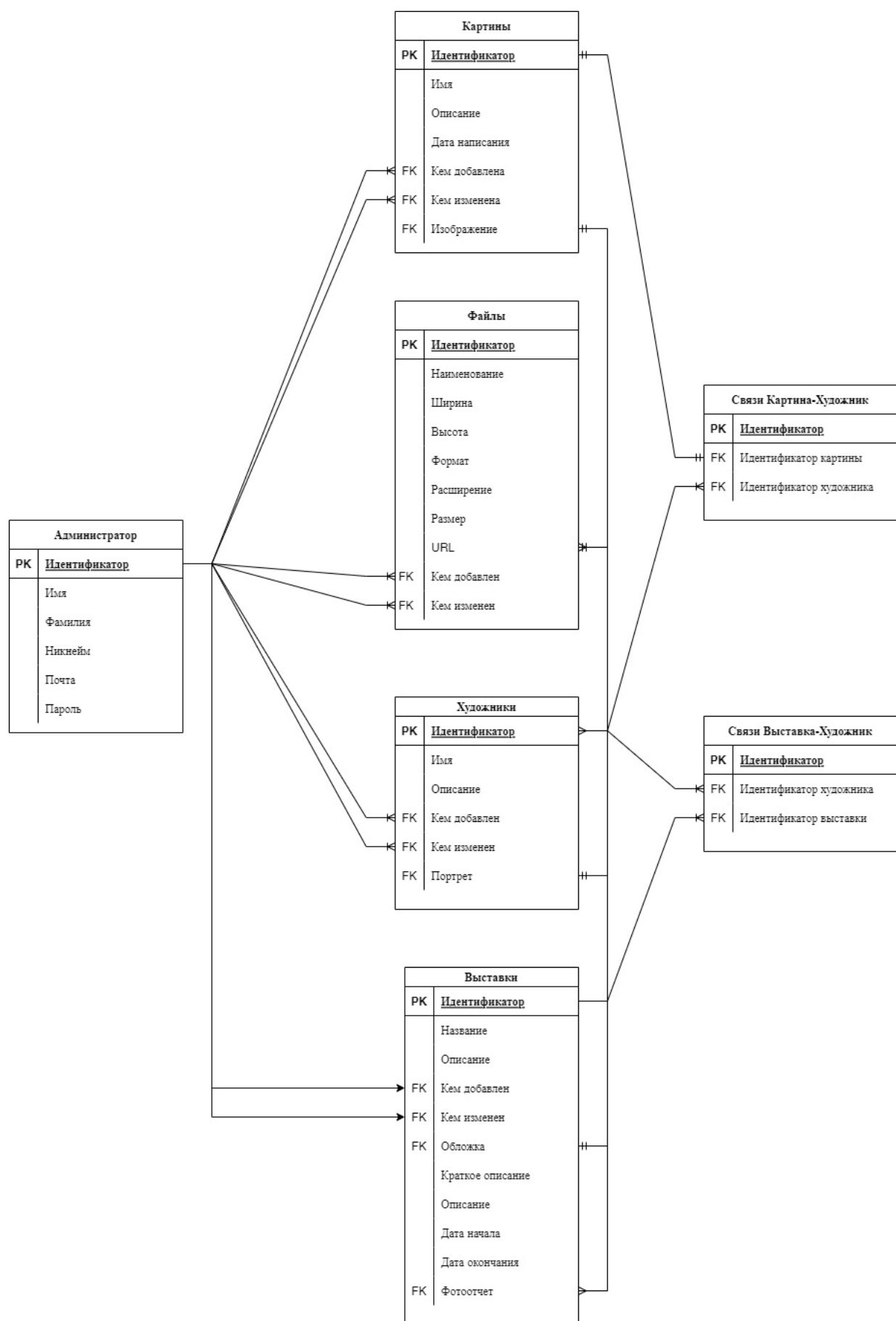


Рисунок 2.11 Даталогическая модель БД

## 2.6. Проектирование интерфейса

Интерфейс приложения является его ключевым звеном, так как он определяет, что увидит конечный пользователь приложения.

Веб-интерфейс – это веб-страница или совокупность веб-страниц, предоставляющая пользовательский интерфейс для взаимодействия с сервисом или устройством посредством протокола HTTP и веб-браузера. Веб-интерфейсы получили широкое распространение в связи с ростом популярности всемирной паутины (интернета), и соответственно повсеместного распространения веб-браузеров.

Одним из основных требований к веб-интерфейсам является их одинаковый внешний вид и одинаковая функциональность при работе в различных браузерах.

Согласованный с заказчиком веб-интерфейс пользователя должен содержать следующие элементы:

- Главный раздел, содержащую выдающиеся произведения галереи, карточки художников и последние выставки галереи;
- Раздел «О галерее» с информацией об истории галереи;
- Раздел «Ближайшие выставки»;
- Раздел «Архив выставок» с уже прошедшими выставками, которые попадают в него по истечении их срока проведения;
- Раздел «Коллекция» с подразделами категорий коллекций, включающих виды изобразительного искусства. Содержит карточки произведений искусства. Каждая карточка содержит соответствующую информацию о работе художника, внесенную при ее добавлении в систему или изменении;
- Раздел «Художники», в нем содержатся все художники, добавленные в систему. Аналогично «Коллекциям» карточка художника позволяет

перейти на его индивидуальную страницу. При этом возможно осуществление поиска по работам данного художника.

## **Выводы по главе 2**

В ходе работы над данной главой построены: диаграмма IDEF0 «Как должно быть», даталогическая и инфологическая модели базы данных, диаграмма прецедентов, диаграммы последовательностей, диаграммы состояний, а также, схема потоков данных. Сформулированы требования заказчика к информационной системе.

## ГЛАВА 3. РЕАЛИЗАЦИЯ КОМПОНЕНТОВ ИНФОРМАЦИОННОЙ СИСТЕМЫ

### 3.1. Реализация front-end части информационной системы

Для реализации пользовательского интерфейса была использована JavaScript библиотека React, что позволило разработать компоненты для их повторного использования и страницы, где эти компоненты используются. Такой подход к разработке называется SPA (Single Page Application). SPA – это одностраничное веб-приложение, которое загружается на одну HTML-страницу. Благодаря динамическому обновлению с помощью JavaScript, во время использования не нужно перезагружать или подгружать дополнительные страницы. На практике это означает, что пользователь видит в браузере весь основной контент, а при прокрутке или переходах на другие страницы, вместо полной перезагрузки нужные компоненты просто подгружаются.

Далее перечислены ключевые компоненты пользовательского интерфейса, листинг кода которых доступен в приложении 1:

1. Header. Отвечает за отрисовку «шапки» страницы приложения;
2. Footer. Ответственен за «подвал» страницы;
3. Artist, Picture, Exhibition. Используются для отображения миниатюр художников, коллекций и выставок;
4. SearchBarOnPage. Поисковая строка;
5. NavBar. Навигационное меню. Используется как в «шапке», так и в отдельных страницах приложения. В мобильной версии принимает вид меню, выпадающего сбоку;

Далее перечисляются страницы, использующие вышеперечисленные компоненты пользовательского интерфейса, листинг кода которых так же доступен в приложении 2:

1. Main;
2. Exhibitions, Paintings, Artists;
3. ExhibitionDetail, PictureDetail, ArtistDetail.

Точкой входа для выполнения веб-приложения является файл `index.js`. Данный файл отображает компонент `App.js`, который хранит в себе маршруты страниц приложения. Листинг кода представлен в разделе «Приложение 2».



### 3.2. Реализация back-end части информационной системы

Для реализации серверной составляющей приложения использовалась система управления контентом (headless-CMS) под названием Strapi, в которой создается API приложения. В качестве архитектурного стиля API был выбран REST.

Для url-адреса веб-приложения определены конечные точки (endpoint), к которым можно обращаться с помощью различных методов сетевых запросов. Данные запросы обрабатываются посредством методов контроллера конкретной модели объекта.

Для каждого контроллера модели определены следующие методы:

1. Find – получение всех объектов данной конечной точки:

```
async find(ctx) {  
    const { query } = ctx;  
  
    const { results, pagination } = await strapi.service(uid).find(query);  
  
    const sanitizedResults = await this.sanitizeOutput(results, ctx);  
  
    return this.transformResponse(sanitizedResults, { pagination });  
}
```

2. FindOne – получение одного объекта данной конечной точки:

```
async findOne(ctx) {  
    const { id } = ctx.params;  
  
    const { query } = ctx;  
  
    const entity = await strapi.service(uid).findOne(id, query);  
  
    const sanitizedEntity = await this.sanitizeOutput(entity, ctx);  
}
```

```

    return this.transformResponse(sanitizedEntity);
  }

```

### 3. Create – создание нового объекта:

```

async create(ctx) {
  const { query } = ctx.request;

  const { data, files } = parseBody(ctx);

  if (!isObject(data)) {
    throw new ValidationError('Missing "data" payload in the request body');
  }

  const sanitizedInputData = await this.sanitizeInput(data, ctx);

  const entity = await strapi
    .service(uid)
    .create({ ...query, data: sanitizedInputData, files });

  const sanitizedEntity = await this.sanitizeOutput(entity, ctx);

  return this.transformResponse(sanitizedEntity);
}

```

### 4. Update – обновление объекта.

```

async update(ctx) {
  const { id } = ctx.params;

  const { query } = ctx.request;

  const { data, files } = parseBody(ctx);

  if (!isObject(data)) {

```

```

        throw new ValidationError('Missing "data" payload in the request body');
    }

    const sanitizedInputData = await this.sanitizeInput(data, ctx);

    const entity = await strapi

        .service(uid)

        .update(id, { ...query, data: sanitizedInputData, files });

    const sanitizedEntity = await this.sanitizeOutput(entity, ctx);

    return this.transformResponse(sanitizedEntity);
}

```

#### 5. Delete –удаление объекта:

```

async delete(ctx) {

    const { id } = ctx.params;

    const { query } = ctx;

    const entity = await strapi.service(uid).delete(id, query);

    const sanitizedEntity = await this.sanitizeOutput(entity, ctx);

    return this.transformResponse(sanitizedEntity);

}

```

Для реализации архивации выставок по истечении их сроков проведения были использованы `cron jobs`. Это один из часто используемых инструментов для Unix-систем. Его используют для планирования выполнения команд на определённое время. Шаблон задания для `cron job` выглядит следующим образом:

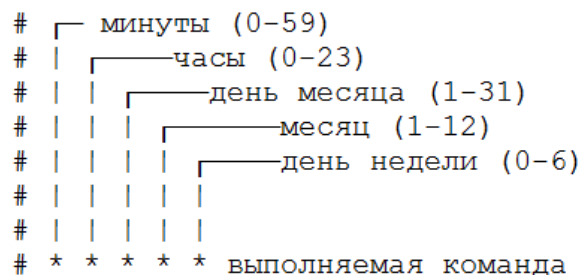


Рисунок 2.12 Шаблон cron задания

Для архивации выставок задано выражение: «00 00 00 \* \* \*». Это соответствует запуску задания ежедневно в полночь. Код самого cron job приведен в «Приложении 3». При выполнении задания проверяется, архивирована ли выставка, имеется ли у нее дата окончания, если она имеется, то проверяется, является ли она валидной. Таким образом, обрабатываются все возможные исключительные ситуации, которые могут возникнуть при эксплуатации информационной системы.

### 3.3. Реализация базы данных

Для хранения данных в системе используется встраиваемая СУБД SQLite. Подключение к БД происходит через конфигурационный файл database.js.

Фрагмент кода database.js:

```
module.exports = ({ env }) => ({  
  
  connection: {  
  
    client: 'sqlite',  
  
    connection: {  
  
      filename: path.join(__dirname, '..', env('DATABASE_FILENAME',  
'./tmp/data.db')),  
  
    },  
  
    useNullAsDefault: true,  
  
  },  
  
});
```

Запросы к БД формируются с помощью генератора SQL запросов Knex.js.

### **3.4. Руководство пользователя**

В ходе выполнения выпускной квалификационной работы поставлена задача разработать и реализовать автоматизированную информационную систему для галереи.

Руководство для пользователя как правильно использовать веб-приложение АИС «Гостиница». Как только пользователь откроет веб-приложение гостиницы, перед ним откроется веб-интерфейс (Рисунок 3.1).

### **Выводы по главе 3**

В данной главе выпускной квалификационной работы описаны реализации клиентской и серверной составляющей информационной системы «Цифровая галерея» и приведена реализация подключения к базе данных.

Для административной панели описано руководство пользователя, упрощающее освоение веб-приложения администратором информационной системы.

## **ЗАКЛЮЧЕНИЕ**

Выпускная квалификационная работа посвящена разработке информационной системы для художественной галереи, а также внедрению готовой системы в бизнес-процессы организации, поскольку работа имеет проектный тип. Данные бизнес-процессы были отражены в диаграммах нотации IDEF0 вида «как есть» и «как должно быть».

В ходе работы исследовалась предметная область. Производился анализ затрат, выгод и рисков результата разработки и внедрения информационной системы. Для разработки были выбраны наиболее актуальные инструменты разработки, подходящие под специфику данного проекта. Таким образом, поставленная цель и задачи выпускной квалификационной работы выполнены.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Правовое регулирование отношений в сфере выставочно-ярмарочной деятельности [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/pravovoe-regulirovanie-otnosheniy-v-sfere-vystavочно-yarmarочноy-deyatelnosti>, свободный. – (Дата обращения: 11.01.2023).
2. Strapi v4 developer documentation [Электронный ресурс]. – Режим доступа: <https://docs.strapi.io/developer-docs/latest/getting-started/introduction.html>, свободный. – (Дата обращения: 11.02.2023).
3. React. JavaScript-библиотека для создания пользовательских интерфейсов [Электронный ресурс]. – Режим доступа: <https://ru.reactjs.org/docs/getting-started.html>, свободный. – (Дата обращения: 16.01.2023).
4. Бибо Бер , Кац Иегуда jQuery. Подробное руководство по продвинутому JavaScript; Символ-плюс - М., 2017. - 624 с.
5. IBM. Documentation. UNIX cron format [Электронный ресурс]. – Режим доступа <https://www.ibm.com/docs/en/db2/11.5?topic=task-unix-cron-format>, свободный. – (Дата обращения: 16.02.2023).
6. Дронов Владимир JavaScript и AJAX в Web-дизайне; БХВ-Петербург - М., 2015. - 736 с.
7. Дронов Владимир JavaScript. Народные советы; БХВ-Петербург - М., 2016. - 458 с.
8. Дунаев Вадим JavaScript. Самоучитель; Питер - М., 2015. - 400 с.
9. Дунаев Вадим HTML, скрипты и стили; БХВ-Петербург - М., 2015. - 816 с.
10. Изучаем Node.js; Питер - М., 2015. - 400 с.
11. Клименко Роман Веб-мастеринг на 100%; Питер - М., 2015. - 920 с.

12. Климов Александр JavaScript на примерах; БХВ-Петербург - М., 2017. - 812 с.
13. Крокфорд Д. JavaScript. Сильные стороны; Питер - М., 2016. - 262 с.
14. Лазаро Исси Коэн, Джозеф Исси Коэн Полный справочник по HTML, CSS и JavaScript; ЭКОМ Паблишерз - М., 2016. - 311 с.
15. Макфарланд Дэвид JavaScript. Подробное руководство; Эксмо - М., 2015. - 608 с.
16. Никсон Робин Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5; Питер - М., 2016. - 768 с.
17. IBM. What is a REST API? [Электронный ресурс]. – Режим доступа <https://www.ibm.com/docs/en/db2/11.5?topic=task-unix-cron-format>, свободный. – (Дата обращения: 16.02.2023).
18. Роберт Мартин. Чистая архитектура. Искусство разработки программного обеспечения; Библиотека программиста (Питер) - М., 2018. - 410 с.
19. Прохоренок Н. А. Python. Самое необходимое; БХВ-Петербург - М., 2015. - 416 с.
20. Резиг Джон , Бибо Беэр Секреты JavaScript ниндзя; Вильямс - М., 2015. - 416 с.
21. Роббинс Дженнифер HTML5, CSS3 и JavaScript. Исчерпывающее руководство (+ DVD-ROM); Эксмо - М., 2017. - 528 с.
22. Фримен Адам jQuery для профессионалов; Вильямс - М., 2015. - 960 с.
23. Херман Дэвид Сила JavaScript. 68 способов эффективного использования JS; Питер - М., 2015. - 952 с.
24. Чаффер Д. Изучаем jQuery 1.3. Эффективная веб-разработка на JavaScript; Символ-плюс - М., 2015. - 391 с.

25. Knex Query Builder [Электронный ресурс]. – Режим доступа: <https://knexjs.org/guide/query-builder.html>, свободный. – (Дата обращения: 24.02.2023).
26. Andy, Harris HTML, XHTML and CSS All-In-One For Dummies® / Andy Harris. - Москва: Наука, 2014. - 173 с.
27. Ben, Henick HTML & CSS – The Good Parts / Ben Henick. - Москва: СИНТЕГ, 2013. - 350 с.
28. Ed, Tittel HTML, XHTML & CSS For Dummies® / Ed Tittel. - Москва: Гостехиздат, 2012. - 416 с.
29. Гаевский, А.Ю. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. - М.: Триумф, 2014. - 464 с.
30. Общие сведения о веб-приложениях [Электронный ресурс]. – Режим доступа: <https://helpx.adobe.com/ru/dreamweaver/using/web-applications.html>, свободный. – (Дата обращения: 25.01.2023).
31. Что такое веб-приложение простыми словами: виды и алгоритм разработки [Электронный ресурс]. – Режим доступа: <https://vc.ru/dev/397220-что-такое-veb-prilozhenie-prostymi-slovami-vidy-i-algoritm-razrabotki>, свободный. – (Дата обращения: 26.01.2023).
32. A Complete Guide to CSS Grid [Электронный ресурс]. – Режим доступа: <https://css-tricks.com/snippets/css/complete-guide-grid/>, свободный. – (Дата обращения: 26.03.2023).
33. Mdn web docs. JavaScript [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>, свободный. – (Дата обращения: 26.03.2023).

34. Mdn web docs. CSS: Cascading Style Sheets [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/en-US/docs/Web/CSS>, свободный. – (Дата обращения: 22.02.2023).
35. Mdn web docs. HTML: HyperText Markup Language [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/en-US/docs/Web/HTML>, свободный. – (Дата обращения: 24.03.2023).
36. Geeks for geeks. React Suite Dropdown Component [Электронный ресурс]. – Режим доступа: <https://www.geeksforgeeks.org/react-suite-dropdown-component/>, свободный. – (Дата обращения: 24.03.2023).
37. Strapi. Forum [Электронный ресурс]. – Режим доступа: <https://forum.strapi.io/>, свободный. – (Дата обращения: 27.03.2023).
38. Habr. React: лучшие практики [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/541320/>, свободный. – (Дата обращения: 17.03.2023).
39. Марк Майерс. A Smarter Way to Learn JavaScript: The New Tech-Assisted Approach that Requires Half the Effort JS; CreateSpace Independent Publishing Platform; F First Edition Used - М., 2014. - 254 с.
40. Хоган, Уоррен, Уэбер: Книга веб-программиста: секреты профессиональной разработки веб-сайтов; Питер – М., 2013. – 288с.
41. 10 правил проектирования интерфейсов, которые нельзя нарушать [Электронный ресурс]. – Режим доступа: <https://ux.pub/editorial/10-pravil-proiektirovaniia-intierfieisov-kotoryie-nielzia-narushat-21m6>, свободный. – (Дата обращения: 17.01.2023).