

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

БИРСКИЙ ФИЛИАЛ УУН_{ИТ}
ФАКУЛЬТЕТ ФИЗИКИ И МАТЕМАТИКИ
КАФЕДРА ИНФОРМАТИКИ И ЭКОНОМИКИ

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ¹

Преддипломная

ОБУЧАЮЩЕГОСЯ
4 курса 4 группы очного отделения

Хусаинова Рената Маратовича

Уровень высшего образования – программа:	бакалавриат
Специальность (направление подготовки)	<u>09.03.03 Прикладная информатика</u>
Направленность (профиль) программы	<u>Прикладная информатика в информационной сфере</u>
Срок проведения практики:	<u>8</u> семестр <u>2022/2023</u> учебного года объемом <u>6</u> зачетных единиц.

Бирск– 2023 г.

¹ указывается вид практики: «учебной», «производственной».

1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ

1. База практики – место прохождения практики обучающимся (профильная организация или УУНиТ).
2. Обучающийся – физическое лицо, осваивающее образовательную программу по направлению подготовки бакалавриата, магистратуры и специальности.
3. Вид практики – учебная, производственная или преддипломная.
4. Каждый обучающийся, находящийся на практике, обязан вести отчет по практике.
5. Отчет по практике служит основным и необходимым материалом для составления обучающимся отчета о своей работе на базе практики.
6. Заполнение отчета по практике производится регулярно, аккуратно и является средством самоконтроля. Отчет можно заполнять рукописным и (или) машинописным способами.
7. Иллюстративный материал (чертежи, схемы, тексты и т.п.), а также выписки из инструкций, правил и других материалов могут быть выполнены на отдельных листах и приложены к отчету.
8. Записи в отчете о практике должны производиться в соответствии с программой по конкретному виду практики.
9. После окончания практики обучающийся должен подписать отчет у руководителя практики, руководителя от базы практики и сдать свой отчет по практике вместе с приложениями (при наличии) на кафедру.
10. При отсутствии сведений в соответствующих строках ставится прочерк.

2. ОБЩИЕ ПОЛОЖЕНИЯ

Фамилия, инициалы, должность руководителя практики от факультета (института)	Хузина Ф.Р., заместитель декана по учебной работе
Фамилия, инициалы, должность руководителя практики от кафедры	Тазетдинов Б.И. доцент кафедры информатики и экономики
Полное наименование базы практики	ООО "ЭОС Платформа"
Наименование структурного подразделения базы практики	-
Адрес базы практики (индекс, субъект РФ, район, населенный пункт, улица, дом, офис)	450071, Башкортостан, г. Уфа, ул. 50 лет СССР, 27
Фамилия, инициалы, должность руководителя практики от профильной организации	Миннихметова Н. В.
Телефон руководителя практики от базы практики	+79174021873

3. РАБОЧИЙ ГРАФИК (ПЛАН) ПРОВЕДЕНИЯ ПРАКТИКИ

Срок проведения практики: с « 13 » марта 2023 по « 9 » апреля 2023

№ п/п	Разделы (этапы) практики	Виды и содержание учебной работы на практике, включая самостоятельную работу обучающихся
1	Подготовительный этап	<ul style="list-style-type: none">- Участие в установочном собрании по практике;- Подготовка документов, подтверждающих факт направления на практику;- Выбор темы исследования, получение задания от руководителя практики;- Производственный инструктаж;- Инструктаж по технике безопасности.
2	Основной этап	<p>Выполнение заданий:</p> <ul style="list-style-type: none">- формирование плана исследования;- изучение научной литературы по теме ВКР;- сбор и анализ необходимой документации для проведения анализа предметной области управленческой деятельности предприятия;- изучение деятельности предприятия, описание информационного и технического обеспечения базы практики, изучение механизмов обеспечения информационной безопасности и построение структурно-функциональной диаграммы организации работы и ее описание по типу «Как есть»;- анализ существующих разработок и применяемого программного обеспечения;- разработка структурно-функциональной диаграммы по типу «Как должно быть»;- описание задач автоматизации информационных процессов (развернутая постановка цели, подзадач автоматизации информационных процессов);- проведение технико-экономического обоснования автоматизации обработки информации;- разработка программного и информационного обеспечения по автоматизируемой задаче;- систематизация и оформление библиографии в соответствии с требованиями;- оформление ВКР в соответствии с требованиями;- подготовка доклада и презентации для защиты выпускной квалификационной работы.
3	Заключительный этап	<p>Составление отчетной документации о прохождении практики. Оформление отчета по практике, сдача зачета по практике</p>

Руководитель практики от кафедры

_____/ Б.И. Тазетдинов
подпись И.О. Фамилия

Руководитель практики от профильной
организации²

_____/ _____
подпись И.О. Фамилия

² При проведении практики в профильной организации руководителем практики от кафедры и руководителем практики от профильной организации составляется совместный рабочий график (план) проведения практики.

4. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Содержание и планируемые результаты практики:

1. Знакомство с организацией, рабочим местом (условия деятельности организации, специфика и структура организации предприятия; особенности его материально-технического обеспечения; нормативная база; система управления; должностные обязанности профильного специалиста).

2. Вводный инструктаж.

3. Оформление и ведение дневника.

4. Выполнение индивидуального задания:

- Формулирование целей и задач, описание предметной области.
- Полное наименование предприятия (организации), где студент проходит практику и являющееся объектом дальнейшей автоматизации. Экономический анализ деятельности организации (миссия организации, система целей и ключевых показателей, стратегия развития, бизнес-архитектура предприятия).
- Характеристики предприятия, включая описание организационной структуры подразделения, где студент проходит практику.
- Состояние и стратегия развития информационных технологий в организации (степень автоматизации процессов, покрытие функциональных областей, ИТ-архитектура, определение уровня зрелости управления ИТ).
- Формирование предложений по автоматизации (информатизации) существующих бизнес-процессов (решения задач, комплекса задач, подсистем) с учетом анализа успешных ИТ-проектов в рассматриваемой области, рынка программного обеспечения и ИТ-технологий.
- Выполнение заданий №1-№6

Задание № 1

Анализ предметной области исследования управленческой деятельности предприятия.

- Сбор и анализ необходимой документации для проведения анализа предметной области управленческой деятельности предприятия.
- Изучение деятельности предприятия, описание информационного и технического обеспечения базы практики, изучение механизмов обеспечения информационной безопасности и построение структурно-функциональная диаграммы организации работы и ее описание по типу «Как есть».
- Анализ существующих разработок и применяемого программного обеспечения.
- Сравнение существующих разработок и применяемого программного обеспечения с известными аналогами.

Задание № 2

Систематизация и оформление библиографии в соответствии с требованиями

- Составление библиографического списка источников информации по теме исследования.
- Оформление библиографического списка источников информации по теме исследования в соответствии с требованиями.

Задание № 3

Описание задач автоматизации

- Разработка структурно-функциональная диаграммы по типу «Как должно быть?»
- Описание задач автоматизации информационных процессов (развернутая постановка цели, подзадач автоматизации информационных процессов).
- Проведение технико-экономического обоснования автоматизации обработки информации.
- Описание задач автоматизации информационных процессов поведенческими диаграммами в нотации UML (диаграмма вариантов использования, диаграмма последовательности, диаграмма состояний).

Задание № 4

Проектирование и разработка программного и информационного обеспечения реализации задач автоматизации

- Инфологическое и даталогическое (при необходимости использовать диаграмму классов) проектирование базы данных.
- Разработка программного и информационного обеспечения по автоматизируемой задаче.

Задание № 5

Оформление ВКР в соответствии с требованиями;

- Оформление материалов квалификационной работы в соответствии с требованиями.
- Создание проекта выпускной квалификационной работы.

Задание № 6

Подготовка доклада и презентации для предзащиты выпускной квалификационной работы

- Составление доклада по теме выпускной квалификационной работы.
- Составление презентации к докладу по теме выпускной квалификационной работы.

5. Оформление отчёта практики.

6. Создание презентации по результатам практики

Руководитель практики от кафедры

_____/ Б.И. Тазетдинов
подпись И.О. Фамилия

Руководитель практики от профильной
организации

_____/ _____
подпись И.О. Фамилия

ОЗНАКОМЛЕН:
Обучающийся

_____/ _____
подпись И.О. Фамилия

5. ИНСТРУКТАЖ ПО ОХРАНЕ ТРУДА

Наименование и реквизиты локального нормативного акта, регламентирующего систему управления охраной труда, техники безопасности, пожарной безопасности профильной организации:

Наименование и реквизиты локального нормативного акта, устанавливающего правила внутреннего трудового распорядка профильной организации:

Инструкция о мерах пожарной безопасности в Башкирском государственном университете, утвержденная приказом УУНиТ.

Правила внутреннего трудового распорядка обучающихся в Башкирском государственном университете, утвержденные приказом УУНиТ.

Перед началом практики инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка прошел:

обучающийся _____ / _____
подпись И.О. Фамилия

Перед началом практики инструктаж обучающегося по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка провел:

_____ / _____
должность подпись И.О. Фамилия

6. ДНЕВНИК РАБОТЫ ОБУЧАЮЩЕГОСЯ

Дата	Информация о проделанной работе, использованные источники и литература (при наличии)
13.03.2023 – 14.03.2023	Производственный инструктаж. Инструктаж по технике безопасности. Анализ предметной области
15.03.2023 – 17.03.2023	Систематизация и оформление библиографии в соответствии с требованиями
20.03.2023 – 21.03.2023	Описание задач автоматизации
22.03.2023 – 31.03.2023	Проектирование и разработка программного и информационного обеспечения реализации задач автоматизации
03.04.2023 – 05.04.2023	Оформление ВКР в соответствии с требованиями;
06.04.2023 – 09.04.2023	Подготовка доклада и презентации для предзащиты выпускной квалификационной работы

Руководитель практики от кафедры³

_____/ Б.И. Тазетдинов
подпись И.О. Фамилия

Руководитель практики от профильной организации

_____/ _____
подпись И.О. Фамилия

³ При прохождении практики в БашГУ.

7. ОТЧЕТ ОБУЧАЮЩЕГОСЯ О ПРАКТИКЕ

с «13» марта 2023 по « 9 » апреля 2023

7.1. Характеристика предметной области и организации.

7.1.1. Краткая характеристика организации.

7.1.2. Описание информационного и технического обеспечения предприятия, используемых функциональных возможностей. Обеспечение безопасности.

7.1.3. Структурно-функциональная диаграмма организации (финансовой структуры) «КАК ЕСТЬ» и ее описание.

7.2. Описание задач автоматизации.

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

7.2.2. Описание задач автоматизации информационных процессов и технико-экономического обоснования автоматизации обработки информации.

7.3. Анализ существующих разработок и применяемого программного обеспечения для автоматизации информационных процессов

7.3.1. Анализ существующих разработок и применяемого программного обеспечения.

7.3.2. Обоснование выбора программных средств для разработки программного обеспечения решаемой задачи.

7.3.3. Описание технического задания на разрабатываемый программный продукт.

7.4. Разработка программного и информационного обеспечения реализации задач автоматизации

7.4.1. Описание информационной модели комплекса задач (схема потоков данных).

7.4.2. Описание программного и информационного обеспечения по автоматизируемой задаче.

7.1. Характеристика предметной области и организации.

7.1.1. Краткая характеристика организации.

Компания «Электронные Офисные Системы» (ЭОС) – ведущий российский производитель и поставщик программных продуктов для автоматизации управления корпоративным контентом, документами и бизнес-процессами.

7.1.2. Описание информационного и технического обеспечения предприятия, используемых функциональных возможностей. Обеспечение безопасности.

Предприятие использует следующий набор технического оборудования:

- 30 персональных компьютеров.
- 4 принтера.
- 1 маршрутизатор для доступа в Интернет.
- 2 сервера.

Предприятие использует следующее программное обеспечение:

- 30 операционных систем Windows 10;
- 30 пакетов MS office 2017.

7.1.3. 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".

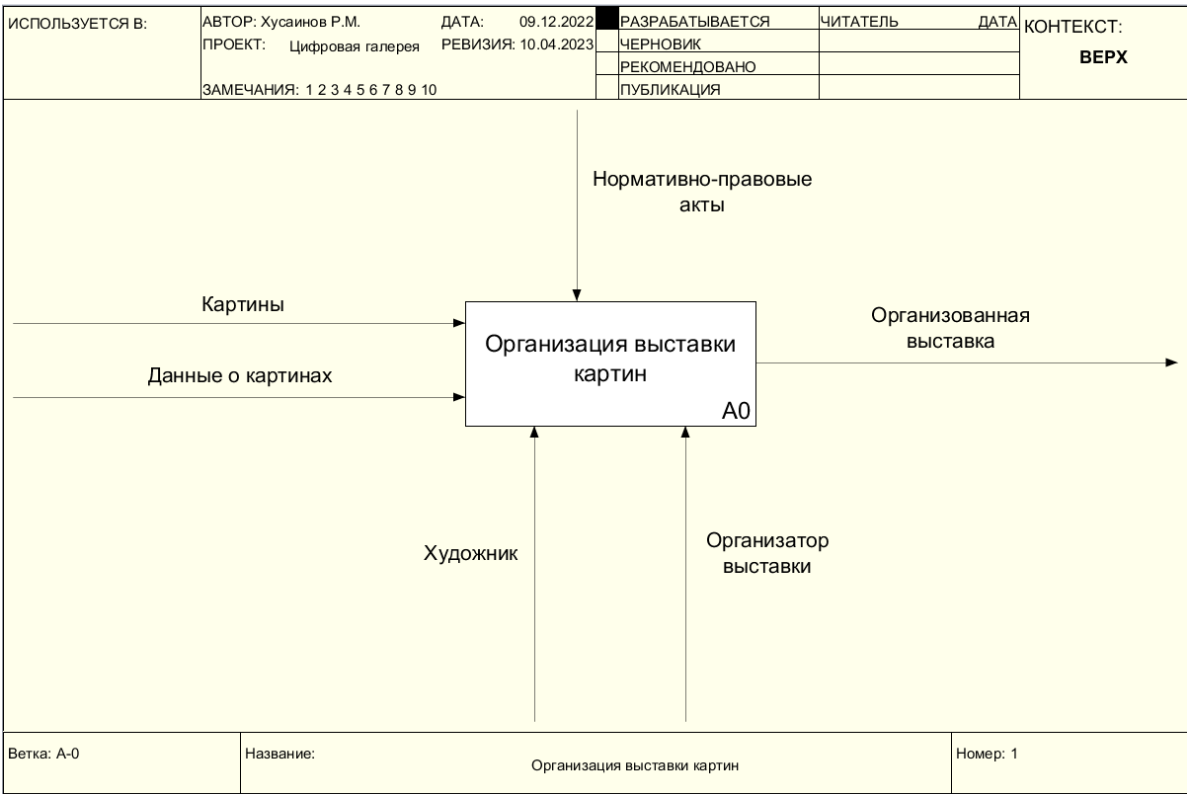


Рисунок 1. Контекстная диаграмма в нотации IDEF0, модель «Как есть», информационного процесса «Организация выставки картин».

Нотация IDEF0 предназначена для формализации и описания бизнес – процессов. С помощью средств данной нотации была создана контекстная диаграмма (Рисунок 1), 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 – выставленные картины, 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, матриц 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 2):

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

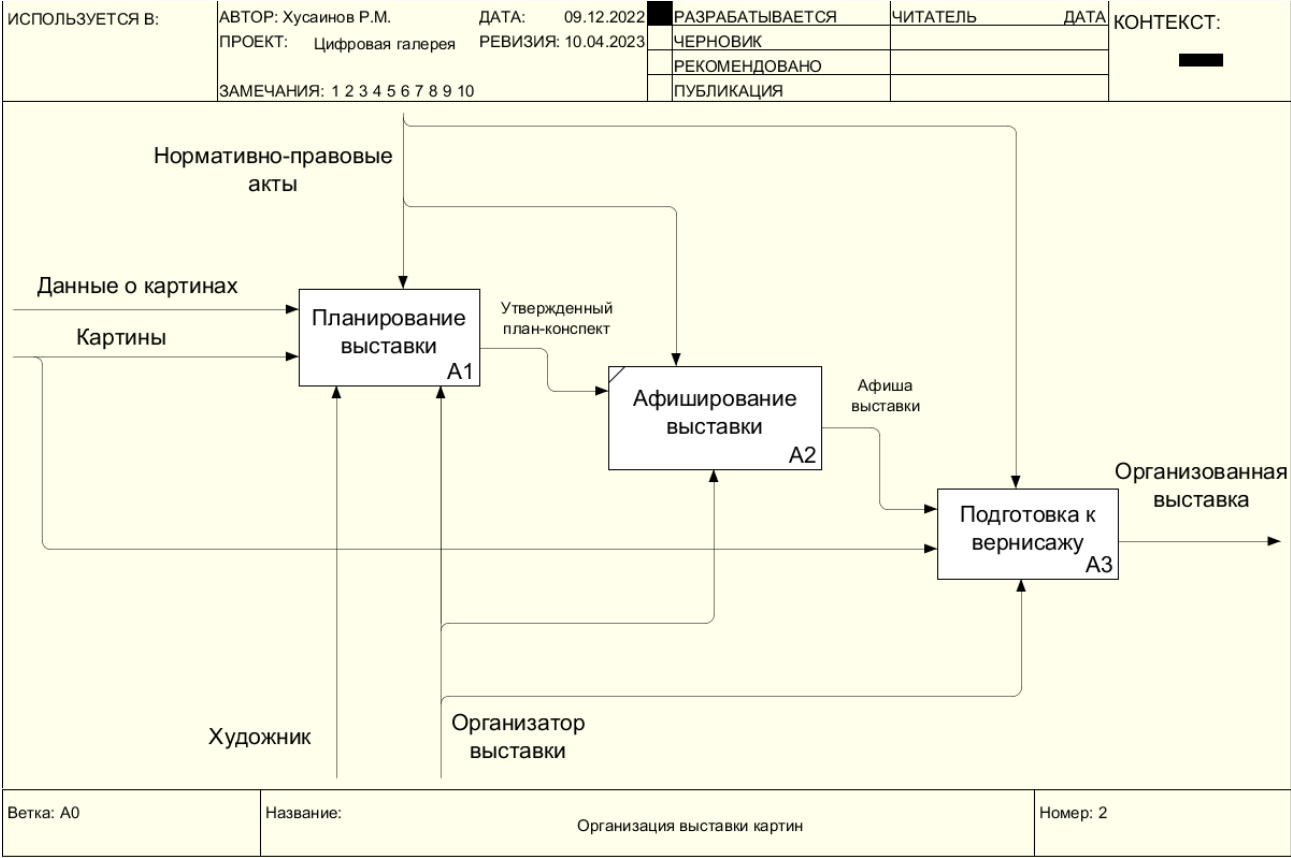


Рисунок 2. Результат декомпозиции контекстной диаграммы.

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

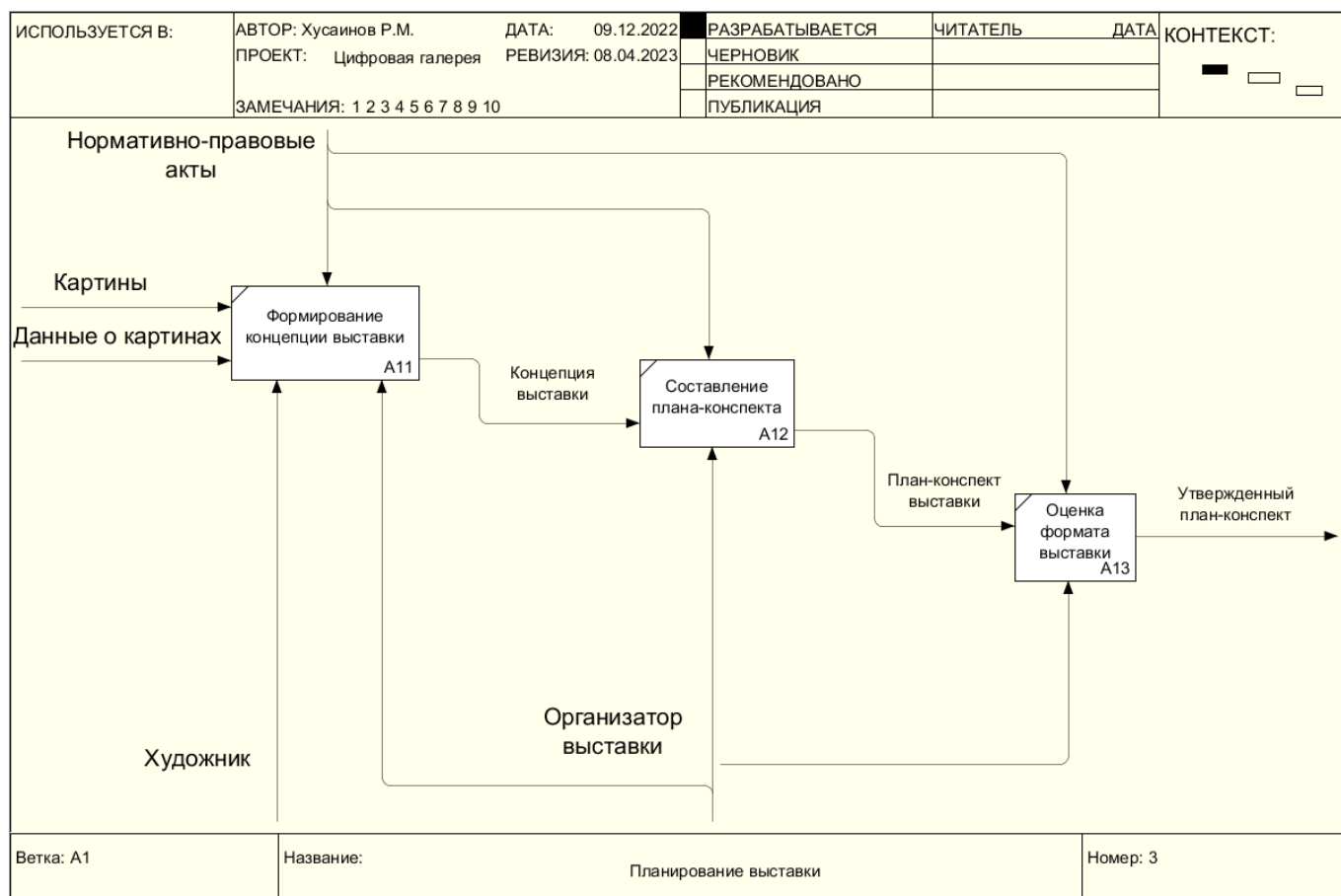


Рисунок 3. Диаграмма декомпозиции процесса "Планирование выставки".

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.

7.2. Описание задач автоматизации.

7.2.1. 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 4).

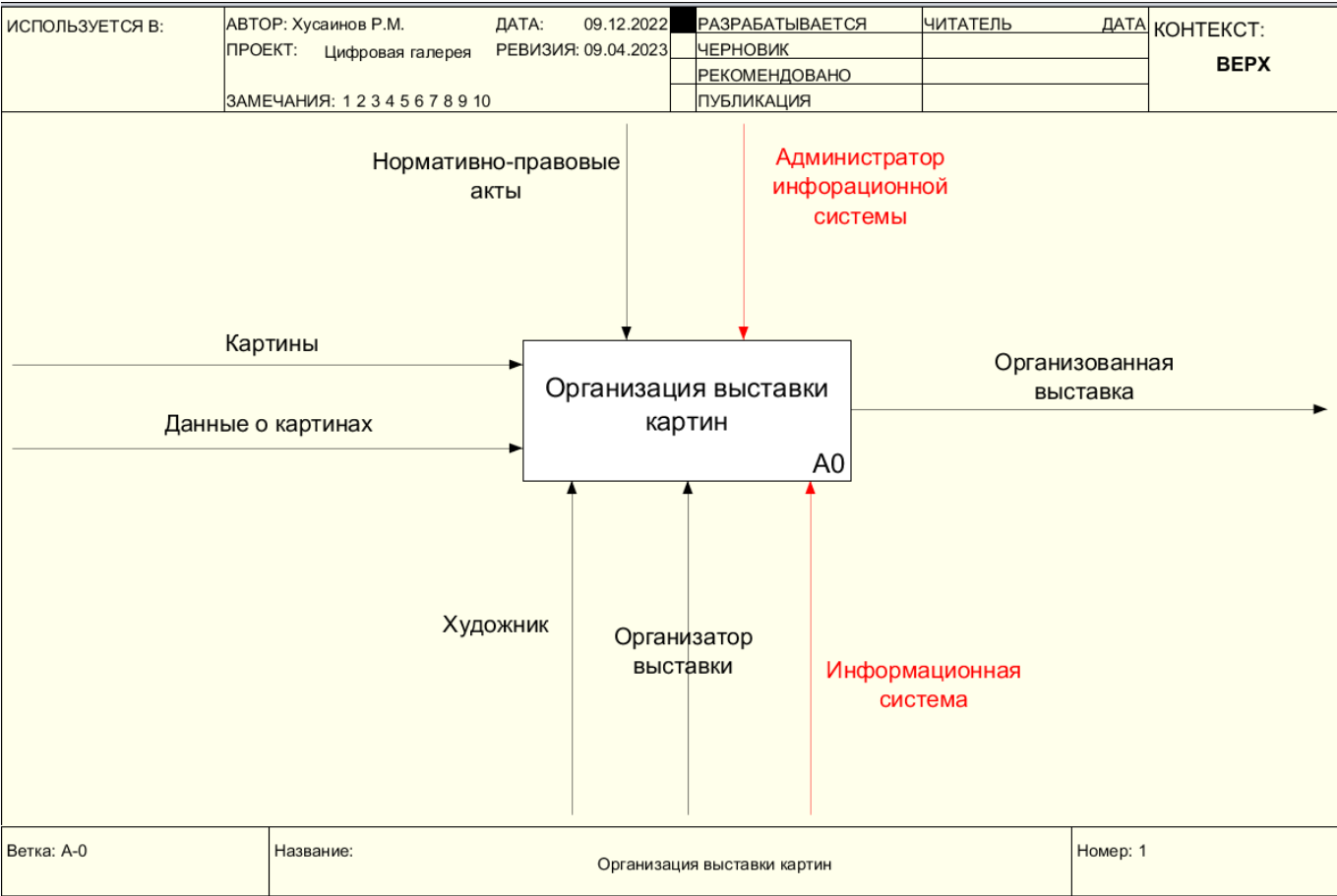


Рисунок 4. Контекстная диаграмма информационного процесса “Организация выставки картин”, модель "Как должно быть?".

Декомпозиция контекстной диаграммы включает в себя следующие функциональные блоки (Рисунок 5):

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

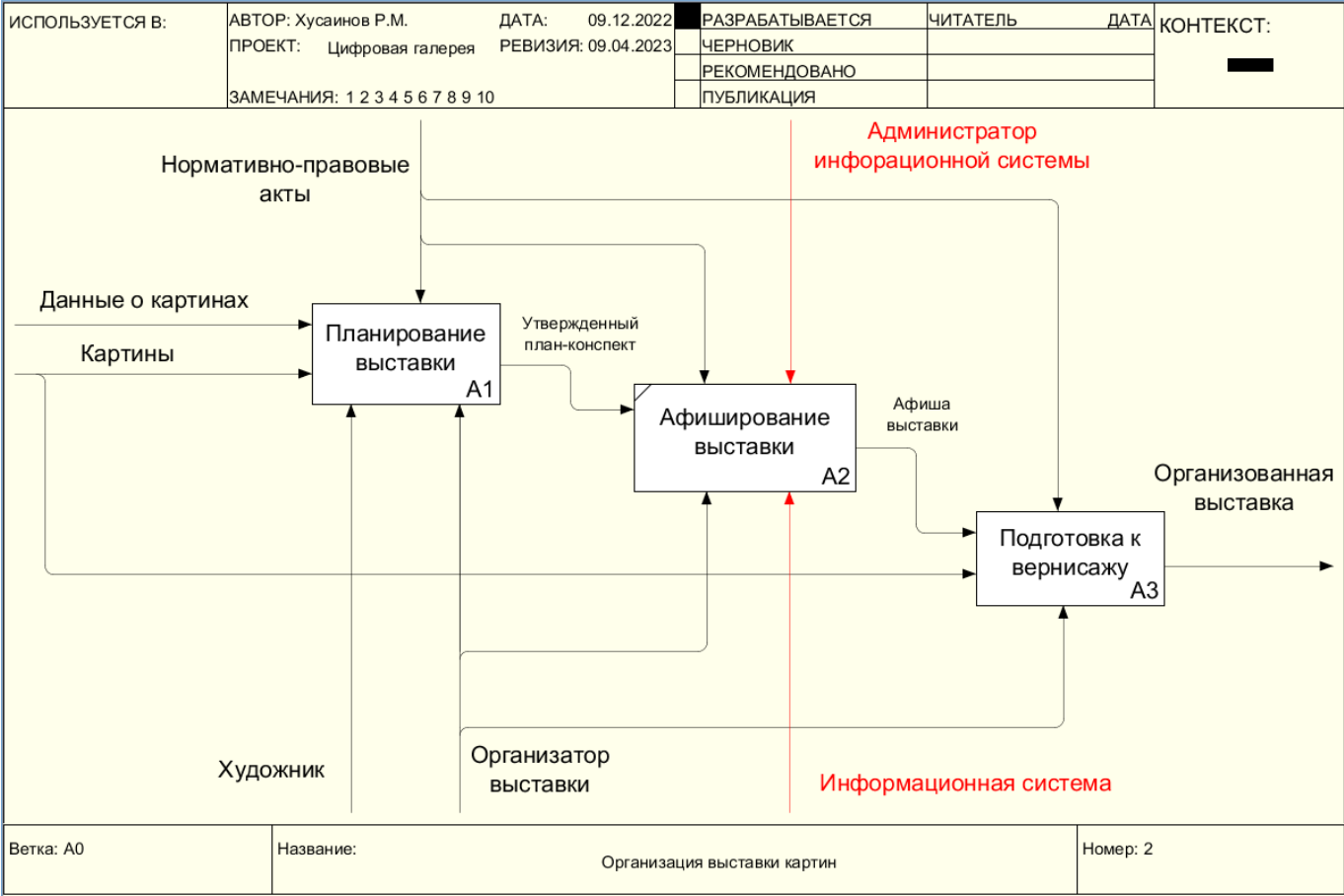


Рисунок 5. Контекстная диаграмма информационного процесса “Организация выставки картин”, модель "Как должно быть?".

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 проведенных выставках.

Для этого необходимо рассмотреть процесс "Проведение выставки" (Рисунок 6).

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 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 6. Контекстная диаграмма информационного процесса “Проведение выставки”, модель “Как должно быть?”.

7.2.2. Описание задач автоматизации информационных процессов и технико-экономического обоснования автоматизации обработки информации.

Задачи автоматизации информационных процессов:

- 1) Предусмотреть систему афиширования выставки в виде показа текущих выставок посредством информационной системы;
- 2) Реализовать возможность автоматизированного архивирования проведенных выставок с возможностью их просмотра пользователями в ИС.

7.3. Анализ существующих разработок и применяемого программного обеспечения для автоматизации информационных процессов

7.3.1. Анализ существующих разработок и применяемого программного обеспечения.

Галереи России преимущественно используют информационные системы, использующие web-технологии (например, web-сайт галереи <https://www.miras.ru/>). Это обусловлено тем, что web-приложения кроссплатформенны и доступны широкому кругу пользователей. Технологии HTML+CSS+JavaScript позволяют реализовать удобный и функциональный пользовательский интерфейс. Стоит отметить, что NodeJS как программная платформа, основанная на движке V8, превращает JavaScript из узкоспециализированного языка в язык общего назначения, позволяя реализовать сервер составляющую web-приложения. В свою очередь, 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. изменения backend составляющей сайта,

4. размещение сайта исключительно на хостинге Tilda,
5. Относительно мало места на хостинге.

7.3.2. Обоснование выбора программных средств для разработки программного обеспечения решаемой задачи.

Для решения поставленных задач используется следующий стек технологий:

1. Библиотека React. Предназначена для разработки пользовательских интерфейсов. Предоставляет возможность разработки интерфейса на основе web-компонентов, которые, в свою очередь, можно повторно применять в приложении без дублирования кода. Также, имеет огромное сообщество разработчиков, следовательно, упрощает процесс разработки при возникновении проблем.
2. CMS Strapi. Является системой управления контентом. Предоставляет удобную, достаточно гибкую административную панель. Она написана на языке JavaScript, соответственно, использует в основе платформу NodeJS.
3. СУБД SQLite. SQLite — компактная встраиваемая реляционная база данных. Исходный код библиотеки передан в общественное достояние. Является чисто реляционной базой данных. Слово «встраиваемый» означает, что SQLite не использует парадигму клиент-сервер. Т.е. движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а предоставляет библиотеку, с которой программа компонуется и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором исполняется программа. Простота реализации достигается за счёт того, что перед началом исполнения транзакции записи весь файл, хранящий базу данных, блокируется.

7.3.3. Описание технического задания на разрабатываемый программный продукт. **Общие сведения**

Полное наименование системы: Информационная система «Цифровая галерея».

Краткое наименование: «Цифровая галерея».

Назначение и цели создания системы

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

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.

«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.

Характеристика объектов автоматизации

Объектом автоматизации являются совокупность бизнес-процессов, выполняемых в галерее, выставляющей произведения изобразительного искусства.

Для удобства изложения и восприятия информации далее участником процессов автоматизации будет выступать не произведение изобразительного искусства в целом, а более конкретно – картина.

Главным объектом автоматизации является процесс афиширования выставок галереи в процессе Организации выставки. Художник создает картины, обращается в галерею с предложением о выставке, галерея назначает организатора выставки и начинается процесс организации выставки, включающий в себя информирование круга лиц о предстоящей выставке. Побочным объектом автоматизации является сохранение материала прошедших выставок в виде архивированных выставок, которые будут представлены в ИС.

Требования к системе в целом

Требования к структуре и функционированию системы

Определяется перечень функциональных подсистем, их назначение и основные характеристики, требования к числу уровней иерархии и степени централизации системы.

Информационная система «Цифровая галерея» должна быть централизованной, т.е. все данные должны располагаться в центральном хранилище. Система должна иметь двухуровневую архитектуру:

1. Источник;
2. Хранилище

В Системе предлагается выделить следующие функциональные подсистемы:

- подсистема хранения данных;
- подсистема формирования и визуализации изображений;
- подсистема авторизации администратора;
- подсистема управления контентом информационной системы.

В качестве протокола взаимодействия между компонентами Системы на транспортно-сетевом уровне необходимо использовать протокол TCP/IP.

Для организации доступа пользователей к отчетности должен использоваться протокол презентационного уровня HTTP и его расширение HTTPS.

Источниками данных для Системы являются хранилища, содержащие изображения картин, полученных с помощью фотографических средств или графического редактора.

Система должна поддерживать следующие режимы функционирования:

- Режим пользователя, в котором все подсистемы, кроме подсистемы управления контентом, выполняют все свои основные функции;
- Режим администратора, в котором присутствует возможность использования подсистемы управления контентом.

В основном режиме функционирования Информационная система «Цифровая галерея» должна обеспечивать:

- работу пользователей в режиме – 24 часов в день, 7 дней в неделю (24x7);
- выполнение своих функций – просмотр страниц веб-приложения, содержащих информацию, поступающую из БД;

- хранение данных.

В режиме администратора она должна обеспечивать возможность проведения следующих работ:

- создание объекта "Выставка" посредством заполнения формы с полями: наименование выставки, краткие сведения о выставке, развернутое описание выставки, обложка выставки, имена художников, чьи картины представлены на выставке, даты начала и окончания выставки и медиа поле для добавления фотоотчета о проведенном мероприятии.

- удаление объекта "Выставка"
- ручная архивация объекта "Выставка"
- снятие с публикации объекта "Выставка" без удаления, с возможностью повторной публикации в ИС.

- загрузка изображений, удаление, обновление и просмотр загруженных изображений. При создании объекта "Выставка" и добавлении изображения запись в БД должна иметь поле, хранящее место хранения данного загруженного изображения в файловой системе сервера;

Для обеспечения высокой надежности функционирования Системы как системы в целом, так и её отдельных компонентов должно обеспечиваться выполнение требований по диагностированию ее состояния.

Диагностирование Системы должно осуществляться следующими штатными средствами:

- СУБД - SQLite
- Браузер с инструментами отладки DevTools – Google Chrome
- Отдельный HTTP-клиент - Postman

Для всех технических компонентов необходимо обеспечить регулярный и постоянный контроль состояния и техническое обслуживание, что должно обеспечиваться организацией, предоставляющей сервер для развертывания Информационной Системы.

Требования к численности и квалификации персонала системы и режиму его работы

В состав персонала, необходимого для обеспечения эксплуатации «Цифровой галереи» в рамках соответствующих подразделений Заказчика, необходимо выделить следующих ответственных лиц:

- Администратор подсистем хранения данных, формирования и визуализации изображений, авторизации администратора. На всем протяжении функционирования ИС обеспечивает распределение дискового пространства, модификацию структур БД, оптимизацию производительности

- Администратор подсистемы управления контентом. Занимается обеспечением подготовки и загрузки данных из внешних источников в хранилище данных.

К квалификации персонала, эксплуатирующего Систему «Цифровая галерея», предъявляются следующие требования:

- Администратор подсистемы управления контентом - знание соответствующей предметной области, непосредственное отношение к учреждению, использующему данную ИС, а также навыки работы с CMS системами.

- Администратор подсистем хранения данных, формирования и визуализации изображений, авторизации администратора. Требуется знание и опыт администрирования СУБД, знание языка запросов SQL, владение навыками работы с реляционной базой данных SQLite.

Требования к режимам работы персонала

Персонал, работающий с Системой «Цифровая галерея» и выполняющий функции её сопровождения и обслуживания, должен работать в следующих режимах:

- Администратор подсистемы управления контентом - знание соответствующей предметной области, непосредственное отношение к учреждению, использующему данную ИС, а также навыки работы с CMS системами.

- Администратор подсистем хранения данных, формирования и визуализации изображений, авторизации администратора – двухсменный график, поочередно.

Требования к показателям назначения

Система должна обеспечивать следующие количественные показатели, которые характеризуют степень соответствия ее назначению:

- Количество пользователей, одновременно использующих ИС = 1000;

- Максимальное количество загружаемых изображений = 1000000;

Обеспечение приспособляемости системы должно выполняться за счет:

- своевременности администрирования;

- модернизации процессов сбора, обработки и загрузки данных в соответствии с новыми требованиями;

- модификации процедур доступа и представления данных конечным пользователям;

- наличия настроечных и конфигурационных файлов у ПО подсистем;

Требования к надежности

Уровень надежности должен достигаться согласованным применением организационных, организационно-технических мероприятий и программно-аппаратных средств.

Надежность должна обеспечиваться за счет:

- применения технических средств, системного и базового программного обеспечения, соответствующих классу решаемых задач;
- своевременного выполнения процессов администрирования Системы «Цифровая галерея»;
- соблюдения правил эксплуатации и технического обслуживания программно-аппаратных средств;
- предварительного обучения обслуживающего персонала.

Время устранения отказа должно быть следующим:

- при перерыве и выходе за установленные пределы параметров электропитания - не более 1 сутки.
- при перерыве и выходе за установленные пределы параметров программного обеспечения - не более 6 часов.

Система должна соответствовать следующим параметрам:

- среднее время восстановления 3 часа - определяется как сумма всех времен восстановления за заданный календарный период, поделенные на продолжительность этого периода;

При работе системы возможны следующие аварийные ситуации, которые влияют на надежность работы системы:

- сбой в электроснабжении сервера;
- сбой в электроснабжении у пользователей системы;
- сбой в электроснабжении обеспечения локальной сети (поломка сети);
- ошибки Системы «Цифровая галерея», не выявленные при отладке и испытании системы;
- сбои программного обеспечения сервера.

К надежности оборудования предъявляются следующие требования:

- в качестве аппаратных платформ должны использоваться средства с повышенной надежностью;
- применение технических средств соответствующих классу решаемых задач;
- аппаратно-программный комплекс Системы должен иметь возможность восстановления в случаях сбоев.

К надежности электроснабжения предъявляются следующие требования:

- с целью повышения отказоустойчивости системы в целом необходима обязательная комплектация серверов источником бесперебойного питания с возможностью автономной работы системы не менее 2 часов;
- система должны быть укомплектована подсистемой оповещения Администраторов о переходе на автономный режим работы;
- должно быть обеспечено бесперебойное питание активного сетевого оборудования.

Надежность аппаратных и программных средств должна обеспечиваться за счет следующих организационных мероприятий:

- предварительного обучения обслуживающего персонала;
- своевременного выполнения процессов администрирования;

- соблюдения правил эксплуатации и технического обслуживания программно-аппаратных средств;

- своевременное выполнение процедур резервного копирования данных.

Надежность программного обеспечения подсистем должна обеспечиваться за счет:

- надежности общесистемного ПО и ПО, разрабатываемого Разработчиком;

- проведением комплекса мероприятий отладки, поиска и исключения ошибок.

- ведением журналов системных сообщений и ошибок по подсистемам для последующего анализа и изменения конфигурации.

Требования к методам оценки и контроля показателей надежности на разных стадиях создания системы в соответствии с действующими нормативно-техническими документами

Проверка выполнения требований по надежности должна производиться на этапе проектирования расчетным путем, а на этапах испытаний и эксплуатации - по методике Разработчика, согласованной с Заказчиком.

Требования к эргономике и технической эстетике

Подсистема авторизации должна быть представлена в виде веб-страницы с интуитивным и понятным интерфейсом.

Внешнее оформление подсистемы авторизации должно отвечать следующим требованиям:

- интерфейс подсистемы должен быть выполнен в стиле остальных подсистем;

- должно быть обеспечено наличие локализованного (русскоязычного) интерфейса пользователя;

- должен использоваться шрифт: Roboto

- размер шрифта должен быть: 14

- цветовая палитра должна состоять из следующих цветов: #FFFFFF, #009933,

К подсистеме управления контентом предъявляются следующие требования к эргономике и технической эстетике.

В части внешнего оформления:

- интерфейс подсистемы должен быть выполнен в стиле остальных подсистем;

- должно быть обеспечено наличие локализованного (русскоязычного) интерфейса пользователя;

- должен использоваться шрифт: Roboto

- размер шрифта должен быть: 14

- интерфейс должен быть понятным, функциональным и простым.

В части диалога с пользователем:

- при возникновении ошибок в работе подсистемы на экран монитора должно выводиться сообщение с наименованием ошибки и с рекомендациями по её устранению на русском языке.

К подсистеме формирования и визуализации изображений предъявляются следующие требования к эргономике и технической эстетике.

В части внешнего оформления:

- изображения должны быть представлены в масштабе, комфортном для просмотра. Для этого требуется предусмотреть возможность увеличения изображения на весь экран.

- для уменьшения нагрузки на сеть пользователя требуется уменьшать размер изображения для его отображения в качестве миниатюры.

Требования к функциям, выполняемым системой

Подсистема хранения данных

Таблица 2

Функции подсистемы хранения данных

Функция	Задача
Управляет процессами сбора, обработки и загрузки данных о выставках, картинах, художниках	Создание, редактирование и удаление процессов сбора, обработки и загрузки данных
	Формирование последовательности выполнения процессов сбора, обработки и загрузки данных
Протоколирует результаты сбора, обработки и загрузки данных	Ведение журналов результатов сбора, обработки и загрузки данных

Таблица 3

Временной регламент реализации каждой функции, задачи

Задача	Требования к временному регламенту
Создание, редактирование и удаление процессов сбора, обработки и загрузки данных	Весь период функционирования системы, при возникновении необходимости изменения процессов сбора, обработки и загрузки данных
Формирование последовательности выполнения процессов сбора, обработки и загрузки данных	Весь период функционирования системы, при возникновении необходимости модификации регламента загрузки данных
Ведение журналов результатов сбора, обработки и загрузки данных	Регулярно, при работе подсистемы

Таблица 4

Требования к качеству реализации функций, задач

Задача	Форма представления выходной информации	Характеристики точности и времени выполнения
Создание, редактирование и удаление процессов сбора, обработки и загрузки данных	Данные в структурах БД, которые представляются в компонентах веб-сайта	Определяется регламентом эксплуатации
Формирование последовательности выполнения процессов сбора, обработки и загрузки данных	Данные в структурах БД, которые представляются в компонентах веб-сайта	Определяется регламентом эксплуатации
Ведение журналов результатов сбора, обработки и загрузки данных	Текстовые файлы	В момент выполнения сбора, обработки и загрузки данных

Таблица 5

Перечень критериев отказа для каждой функции

Функция	Критерии отказа	Время восстановления	Коэффициент готовности
Управляет процессами сбора, обработки и загрузки данных	Не выполняется одна из задач функции.	8 часов	0.85
Протоколирует результаты сбора, обработки и загрузки данных	Не выполняется одна из задач функции.	12 часов	0.75

Подсистема формирования и визуализации изображений

Таблица 6

Перечень функций, задач подлежащих автоматизации

Функция	Задача
Формирует изображения картин, оптимизированных для использования в веб-сайтах	Обеспечение веб-сайта изображениями картин оптимального размера
Для полноэкранного просмотра подгружает изображение высокого разрешения	Обеспечение пользователя возможностью просмотра изображений картин в режиме полноэкранного просмотра

Таблица 7

Временной регламент реализации каждой функции, задач

Задача	Требования к временному регламенту
Обеспечение веб-сайта изображениями картин оптимального размера	Реализация до начала функционирования системы
Обеспечение пользователя возможностью просмотра изображений картин в режиме полноэкранного просмотра	Реализация до начала функционирования системы

Таблица 8

Требования к качеству реализации функций, задач

Задача	Форма представления выходной информации	Характеристики точности и времени выполнения
Обеспечение веб-сайта изображениями картин оптимального размера данных	Изображения с расширениями .jpeg, .png со сжатием	Определяется регламентом эксплуатации
Обеспечение пользователя возможностью просмотра изображений картин в режиме полноэкранного просмотра	Изображения с расширениями .jpeg, .png без сжатия	Определяется регламентом эксплуатации

Таблица 9

Перечень критериев отказа для каждой функции

Функция	Критерии отказа	Время восстановления	Коэффициент готовности
Формирует изображения картин, оптимизированных для использования в веб-сайтах	Не выполняется задача функции.	2 часа	0.85
Для полноэкранного просмотра подгружает изображение высокого разрешения	Не выполняется задача функции.	2 часа	0.75

Подсистема авторизации администратора

Таблица 10

Перечень функций, задач подлежащих автоматизации

Функция	Задача
Обеспечивает доступ Администратора к системе	Вход в систему через специальную страницу авторизации
Обеспечивает защиту от злоумышленников	Защита системы от неправомерного доступа

Таблица 11

Временной регламент реализации каждой функции, задачи

Задача	Требования к временному регламенту
Вход в систему через специальную страницу авторизации	Реализация до начала функционирования системы
Защита системы от неправомерного доступа	Реализация до начала функционирования системы

Таблица 12

Требования к качеству реализации функций, задач

Задача	Форма представления выходной информации	Характеристики точности и времени выполнения
Вход в систему через специальную страницу авторизации	При успешной авторизации происходит загрузка веб-страницы системы управления контентом.	Определяется регламентом эксплуатации
Защита системы от неправомерного доступа	Текстовый документ, содержащий информацию о попытках входа в систему	Определяется регламентом эксплуатации

Таблица 13

Перечень критериев отказа для каждой функции

Функция	Критерии отказа	Время восстановления	Коэффициент готовности
Обеспечивает доступ Администратора к системе	Не выполняется задача функции.	2 часа	0.85
Обеспечивает защиту от злоумышленников	Не выполняется задача функции.	2 часа	0.75

Подсистема управления контентом информационной системы

Таблица 14

Перечень функций, задач подлежащих автоматизации

Функция	Задача
Формирует интерфейс для работы с данными (изображениями, объектами автоматизации, создание, чтение, изменение и удаление объектов)	Обеспечение администратора подсистемой для работы с данными в информационной системе «Цифровая галерея»
Внесение изменений в базу данных	Добавление/чтение/обновление/удаление данных в БД без необходимости написания запросов к БД

Таблица 15

Временной регламент реализации каждой функции, задачи

Задача	Требования к временному регламенту
Обеспечение администратора подсистемой для работы с данными в информационной системе «Цифровая галерея»	Реализация до начала функционирования системы
Добавление/чтение/обновление/удаление данных в БД без необходимости написания запросов к БД	Реализация до начала функционирования системы

Таблица 16

Требования к качеству реализации функций, задач

Задача	Форма представления выходной информации	Характеристики точности и времени выполнения
Обеспечение администратора подсистемой для работы с данными в информационной системе «Цифровая галерея»	Веб-приложение, позволяющее производить CRUD операции над сущностями БД	Реализация до начала функционирования системы
Добавление, чтение, обновление, удаление данных в БД без необходимости написания запросов к БД	Запросы к БД, сформированные с помощью подсистемы	Реализация до начала функционирования системы

Таблица 17

Перечень критериев отказа для каждой функции

Функция	Критерии отказа	Время восстановления	Коэффициент готовности
Формирует интерфейс для работы с данными (изображений, художников)	Не выполняется задача функции.	2 часа	0.85
Внесение изменений в базу данных	Не выполняется задача функции.	2 часа	0.75

7.4. Разработка программного и информационного обеспечения реализации задач автоматизации

7.4.1. Описание информационной модели.

Модели базы данных

Сформированы инфологическая и даталогическая модели БД.

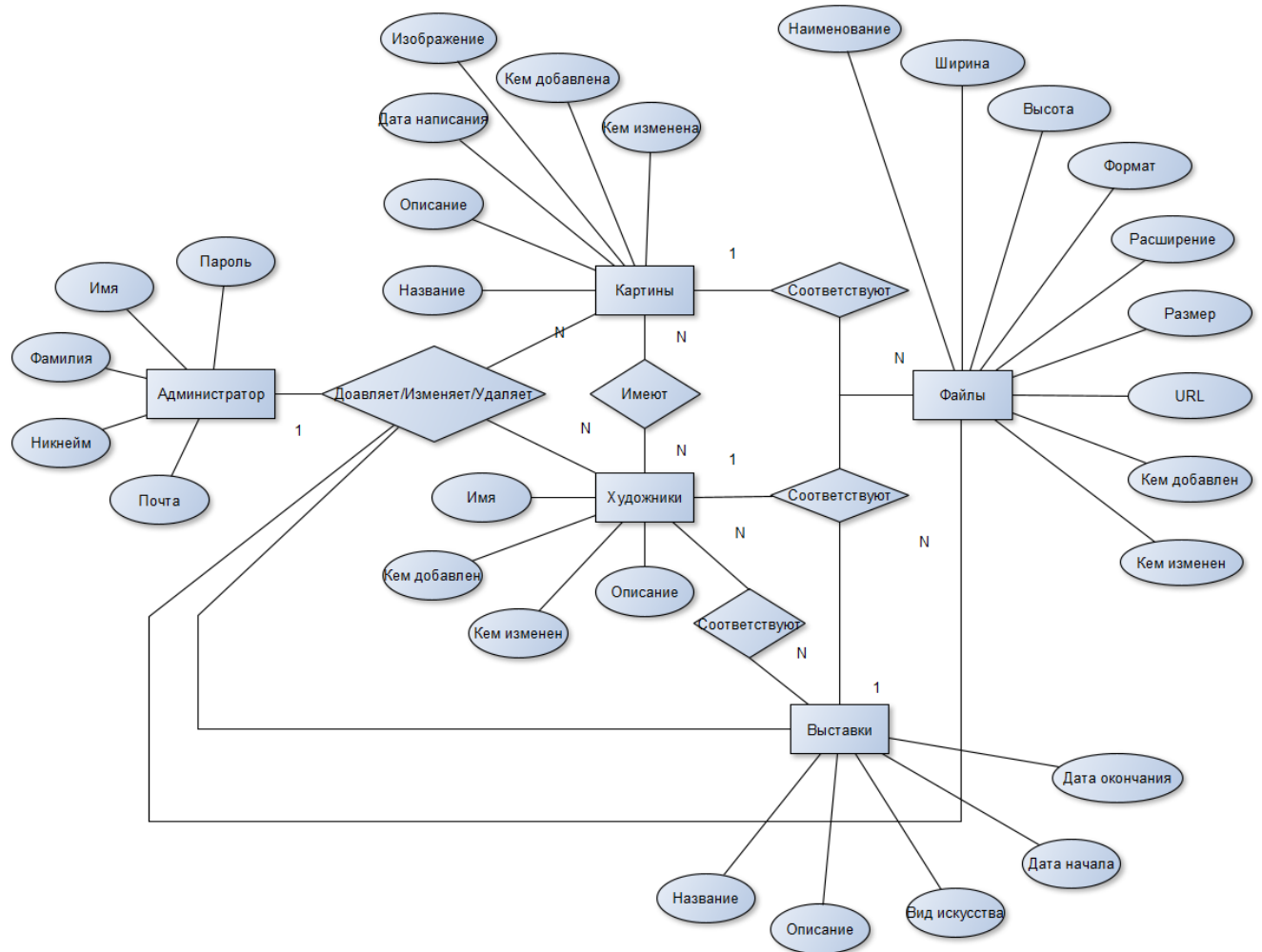


Рисунок 7. Инфологическая модель БД.

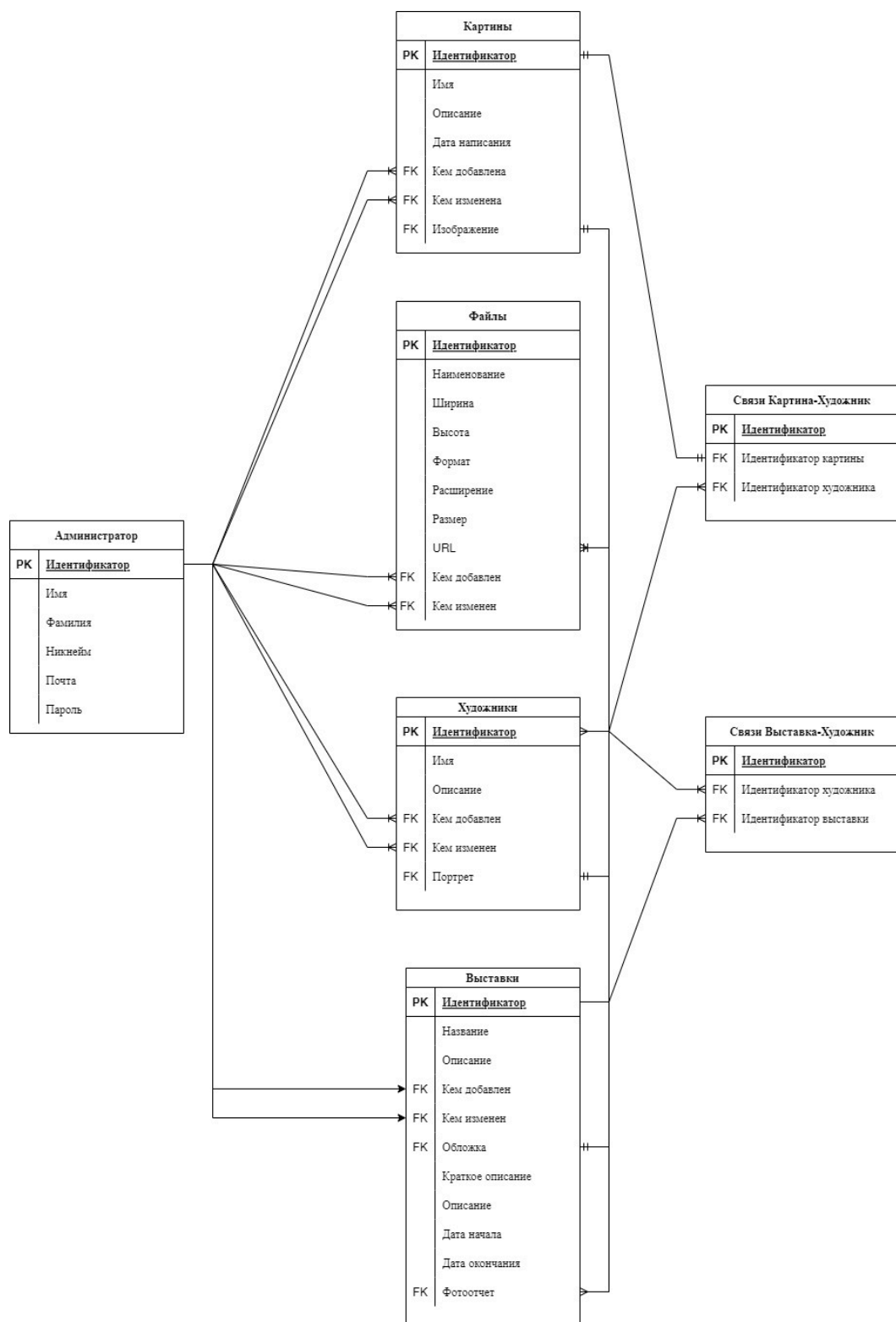


Рисунок 8. Дatalogическая модель БД.

7.4.2. Описание программного и информационного обеспечения по автоматизируемой задаче.

Реализация front-end части информационной системы

Для реализации пользовательского интерфейса была использована JavaScript библиотека React. Разработаны компоненты для их повторного использования и страницы, где эти компоненты используются.

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

1. Header;
2. Footer;
3. Exhibition;
4. Main;
5. App.

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

1. Main;
2. Exhibitions;
3. ExhibitionDetail.

Точкой входа для выполнения веб-приложения является файл index.js. Листинг кода представлен в разделе «Приложение 2».

Реализация back-end части информационной системы

Для реализации 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 актуальности 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:

```

# ┌─ минуты (0-59)
# │ ┌── часы (0-23)
# │ │ ┌── день месяца (1-31)
# │ │ │ ┌── месяц (1-12)
# │ │ │ │ ┌── день недели (0-6)
# │ │ │ │ │
# │ │ │ │ │
# │ │ │ │ │
# * * * * * выполняемая команда

```

Рисунок 9. 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.

Реализация базы данных

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

Листинг 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.

Заключение

Я, _____⁴ прошел _____⁵ практику с «___»
_____ 20__ по «___» _____ 20__.

В соответствии с программой практики и индивидуальным заданием я выполнял следующую работу:

В результате прохождения практики поставленные задачи были решены в полном объеме, профессиональные компетенции (профессиональные умения, навыки и опыт профессиональной деятельности) приобретены.

Обучающийся

_____/_____
подпись И.О. Фамилия

⁴ Фамилия имя отчество обучающегося

⁵ Указывается вид практики (учебная, производственная или преддипломная).

8. ОТЗЫВ О ПРАКТИКЕ ОБУЧАЮЩЕГОСЯ

Обучающийся _____⁶ прошел _____⁷ практику с «13» марта 2023 по « 9 »апреля 2023.

Перед обучающимся во время прохождения практики были поставлены следующие профессиональные задачи: _____

Краткая характеристика проделанной работы и полученных результатов: _____

Во время прохождения практики обучающийся проявил себя как (достоинства, уровень теоретической подготовки, дисциплина, недостатки, замечания) _____

Рекомендации (пожелания) по организации практики: _____

Руководитель практики от профильной организации

_____/_____
М.П. подпись И.О. Фамилия
«__» _____ 20__

⁶ Фамилия имя отчество обучающегося

⁷ Указывается вид практики (учебная, производственная или преддипломная).

9. РЕЗУЛЬТАТ ЗАЩИТЫ ОТЧЕТА

В результате прохождения практики поставленные задачи были решены в полном объеме, профессиональные компетенции (профессиональные умения, навыки и опыт профессиональной деятельности) приобретены.

Результат прохождения практики обучающимся оценивается на: _____

Руководитель практики от кафедры

_____ / Б.И. Тазетдинов

подпись

И.О. Фамилия

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-chto-takoe-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-proektirovaniia-intierfieisov-kotoryie-nielzia-narushat-21m6>, свободный. – (Дата обращения: 17.01.2023).

Приложение 1

1. ...Components/Header.js:

```
import './header.css'
import SearchBar from '../components/SearchBar'
import {useNavigate} from 'react-router-dom'

function Header({searchValue, setSearchValue}) {

  const navigate = useNavigate()

  return (
    <header>
      <div className='wrapper'>
        <div className='logo' onClick={() => navigate('/')}>
          <h2>Цифровая галерея</h2>
        </div>
        <SearchBar searchValue={searchValue} setSearchValue={setSearchValue}/>
      </div>
    </header>
  )
}

export default Header
```

2. ...Components/Exhibition.js:

```
import './exhibition.css';
import { useNavigate } from 'react-router-dom';
import { Divider, Card, CardMedia, CardContent, Typography, CardActions, Button } from
"@mui/material";
import { useEffect } from 'react';

function Exhibition({ cardTitle, about, image, id, startDate, finishDate, onClick = () => { } }) {

  const navigate = useNavigate();
  useEffect(() => {
    console.log(image.data);
  });
  return (
    <div className='exhibition' onClick={() => {
      navigate(`/exhibition/${id}`, {
        state: { id }
      }); onClick()
    }}>
      <Card className="exhibitionCard" sx={{ height: 430 }}>
        <CardMedia
          sx={(image === '/noImage.svg') ? {height: 300,
            transform: 'scale(0.5)',
            backgroundSize: 'contain',
```

```

        WebkitBackgroundSize: 'none'
      } : {height: 300}}
      image={image}
      title={cardTitle}
    />
    <CardContent >
      <Typography noWrap gutterBottom variant="h5" component="div">
        {cardTitle}
      </Typography>
      <Typography variant="body1" color="text.secondary" noWrap>
        {about}
      </Typography>
      <Typography sx={{ fontWeight: 'bold' }} variant="body1" color="text.secondary">
        {startDate} {finishDate ? ` — ${finishDate}` : ""}
      </Typography>
    </CardContent>
    {/* <CardActions>
      <Button size="small">Узнать подробнее</Button>
    </CardActions> */}
  </Card>

</div>
)
}

```

export default Exhibition;

3. ...Components/Footer.js:

```

import './footerStyles.js'
import React from "react";
import {
  Box,
  Container,
  Row,
  Column,
  FooterLink,
  Info,
  Heading,
  Img
} from './footerStyles';
import LocalPhoneOutlinedIcon from '@mui/icons-material/LocalPhoneOutlined';
import PlaceOutlinedIcon from '@mui/icons-material/PlaceOutlined';

const Footer = () => {
  return (
    // <img src='/logo.svg' width='200px'></img>
    <Box>
      {/* <h1 style={{ color: "green",
        textAlign: "center",
        marginTop: "-50px" }}>

```

```

</h1> */}
<Container>
  <Row>
    {/* <Column>
      <img src='/logo.svg' width='200px'></img>
    </Column> */}
    <Column>
      <Heading>Посетителям</Heading>
      <FooterLink to="/visit">Экскурсии</FooterLink>
      <FooterLink to="/about">О галерее</FooterLink>
    </Column>
    <Column>
      <Heading>Выставки</Heading>
      <FooterLink to="/exhibitions">Ближайшие выставки</FooterLink>
      <FooterLink to="/exhibitionsArchive">Архив выставок</FooterLink>
    </Column>
    <Column>
      <Heading>Коллекция</Heading>
      <FooterLink to="/paintings">Вся коллекция</FooterLink>
      <FooterLink to="#">Живопись</FooterLink>
      <FooterLink to="#">Графика</FooterLink>
      <FooterLink to="#">Декоративно-прикладное искусство</FooterLink>
    </Column>
    <Column>
      <Heading>Образование</Heading>
      <FooterLink to="/education">Образовательные услуги</FooterLink>
      <FooterLink to="#">Запись</FooterLink>
      <FooterLink to="#">Расписание</FooterLink>
    </Column>
  </Row>
  <Row>
    <Column>
      <img src='/logo.jpg'></img>
    </Column>
    <Column>
      <Heading>Контакты</Heading>
      <Info>
        Приемная галереи
      </Info>
      <FooterLink to="tel:+7(347)2999999">
        +7-(347)-2-99-99-99
      </FooterLink>

```

```

      <FooterLink to="https://yandex.ru/maps/org/birskiy_filial_uunit/1210611238/?ll=55.526310%2C55.411783&z=17.41" target="_blank">

```

```

        Бирск, Республика Башкортостан<br></br>Интернациональная улица, 10
      </FooterLink>
    </Column>
    <Column>
      <Heading>Социальные сети</Heading>
      <FooterLink to="https://vk.com/birskbgu" target="_blank">

```

```

        <i className="fab fa-facebook-f">
          <span style={{ marginLeft: "10px" }}>
            ВКонтакте
          </span>
        </i>
      </FooterLink>
      <FooterLink to="#">
        <i className="fab fa-youtube">
          <span style={{ marginLeft: "10px" }}>
            Youtube
          </span>
        </i>
      </FooterLink>
    </Column>
  </Column></Column>

</Row>
</Container>
</Box>
);
};
export default Footer;

```

4. App.js

```

import './App.css';
import { Routes, Route } from "react-router-dom"
import Home from './pages/Home';
import PictureDetail from './pages/PictureDetail';
import Header from './components/Header';
import Search from './pages/Search';
import { useState } from 'react';

function App() {

  const [searchValue, setSearchValue] = useState("")

  return (
    <div className="App">
      <Header searchValue={searchValue} setSearchValue={setSearchValue}/>
      <Routes>
        <Route path="/" exact element={<Home searchValue = {searchValue}
setSearchValue={setSearchValue}/>}/>
        <Route path="/detail/:id" element={<PictureDetail />}/>
        <Route path="/search/:query" element={<Search searchValue={searchValue}/>}/>
      </Routes>
    </div>
  );
}

export default App;

```


Приложение 2

1. Pages/Main.js

```
import { useEffect, useState, useRef } from "react";
import "../main.css";
import "../ckeditor.css"

import axios from "axios";
import { Link } from 'react-router-dom';
import { Swiper, SwiperSlide } from "swiper/react";
import { Navigation, Pagination, Autoplay, A11y } from 'swiper';
import { useSwiper } from 'swiper/react';
import ArrowForwardOutlinedIcon from '@mui/icons-material/ArrowForwardOutlined';
import ArrowBackOutlinedIcon from '@mui/icons-material/ArrowBackOutlined';
import Artist from "../components/Artist";
import Exhibition from "../components/Exhibition";
import { textAlign } from "@mui/system";

function Main({ searchValue, setSearchValue }) {
  useEffect(() => {
    setSearchValue("");
    window.scrollTo(0, 0);
  }, []);

  const [slides, setSlides] = useState([]);
  const [articles, setArticles] = useState();
  const rootURL = "http://localhost:1337";
  const URL = "http://localhost:1337/api/slides/?populate[picture][populate]=*";
  const articlesURL = "http://localhost:1337/api/articles/3";

  const fetchData = async (setHook, URL) => {
    let data;
    const result = await axios.get(URL);
    data = result.data.data;
    setHook(data);
  };

  useEffect(() => {
    fetchData(setSlides, URL);
    fetchData(setArticles, articlesURL);
    // parseHtml(articles?.attributes?.Content);
    // console.log(typeof articles?.attributes?.Content);
  }, []);

  // console.log(JSON.stringify(slides[0]?.attributes?.picture?.data?.id));
  const ApiAddress = axios.create({
    baseURL: "http://localhost:1337/api/exhibitions"
  });

  const [exhibitionApiData, setExhibitionApiData] = useState([]);
```

```

const imageURL = "http://localhost:1337";
useEffect(() => {

ApiAddress.get(`?pagination[page]=1&pagination[pageSize]=6&filters[${$or}][0][isArchived][${$eq}]=false&
filters[${$or}][1][isArchived][${$null}]=true&populate=*&sort=createdAt:asc`)
  .then((response) => setExhibitionApiData(response.data.data))
  .catch((err) => console.log(err));
}, []);

const swiperRef = useRef();
const swiperHeroRef = useRef();

const [artistApiData, setArtistApiData] = useState([]);
const ArtistApiAddress = axios.create({
  baseURL: "http://localhost:1337/api/artists"
});
useEffect(() => {
  //ArtistApiAddress.get(`?pagination[page]=1&pagination[pageSize]=8&populate=*`)

  ArtistApiAddress.get(`?filters[onMainPage][${$eq}]=true&populate=*`)
    .then((response) => setArtistApiData(response.data.data))
    .catch((err) => console.log(err));
}, []);

return (
  <div className="main">
    <section className="hero">
      <div className="heroContentContainer wrapper">
        <h1>Изобразительное искусство <br></br>Республики Башкортостан</h1>
        <div className="sliderWrapper sliderWrapperHero wrapper">
          <button className="sliderBtn sliderBtnPrev" onClick={() =>
swiperHeroRef.current.slidePrev()}><ArrowBackOutlinedIcon /></button>
          <Swiper className="slider sliderHero"
            modules={[Navigation, Pagination, A11y, Autoplay]}
            spaceBetween={30}
            centeredSlides
            loop
            slidesPerView={3}
            onSlideChange={() => console.log("slide change")}
            onSwiper={(swiper) => {
              swiperHeroRef.current = swiper;
            }}
            // pagination={{ clickable: true }}
            autoplay={{ delay: 5000 }}
          >
            </* <SwiperSlide className="SlideContainer" style={{ backgroundImage:
'url(${slide})' }}>
              </SwiperSlide> */
            </* достаем слайдеры из api strapi, созданные администратором */>
            {slides?.map((slide) => {

```

```

                                let temp =
slide.attributes?.picture?.data?.attributes?.image?.data[0]?.attributes?.formats?.large?.url ??

slide.attributes?.picture?.data?.attributes?.image?.data[0]?.attributes?.formats?.medium?.url ??

slide.attributes?.picture?.data?.attributes?.image?.data[0]?.attributes?.formats?.small?.url ??

slide.attributes?.picture?.data?.attributes?.image?.data[0]?.attributes?.formats?.thumbnail?.url ??
    "/noImage.svg";
    return <SwiperSlide className="slideContainer">
    <Link to={`/painting/${slide.attributes?.picture?.data?.id}`}
    state={slide.attributes?.picture?.data} className="slideLink">
    <img src={temp === "/noImage.svg" ? "/noImage.svg" : (imageURL + temp)} alt=""
/>

    <h1>{slide.attributes?.picture?.data?.attributes?.name}</h1>
    <p>{slide.attributes?.picture?.data?.attributes?.artist?.data?.attributes?.name}</p>
    {/* <p>{slide.attributes?.picture?.data?.attributes?.description}</p> */}
    </Link>
    </SwiperSlide>
    )})
    </Swiper>

                                <button className="sliderBtn sliderBtnNext" onClick={() =>
swiperHeroRef.current.slideNext()}><ArrowForwardOutlinedIcon /></button>

    </div>
    </div>
    </section>

    <Divider></Divider>
    <section className="wrapper">
    <h1 className="sectionTitle">Выставки</h1>
    <div className="sliderWrapper">
                                <button className="sliderBtn sliderBtnPrev" onClick={() =>
swiperRef.current.slidePrev()}><ArrowBackOutlinedIcon /></button>
    <Swiper className="sliderExhibition wrapper"
    slidesPerView={3}
    modules={[Navigation, Pagination, A11y, Autoplay]}
    spaceBetween={40}
    onSlideChange={() => console.log("slide change")}
    // pagination={{ clickable: true }}
    // autoplay={{ delay: 5000 }}
    onSwiper={(swiper) => {
    swiperRef.current = swiper;
    }}
    >

    <div className="exCardWrapper">
    {exhibitionApiData?.map((exhib) => {
    return (
    <SwiperSlide className="slideContainer">
    <Exhibition

```



```

        key={exhib.id}
        cardTitle={exhib.attributes.title}
        about={exhib.attributes.about}
        image={exhib.attributes.cover?.data ?
          (imageUrl + exhib.attributes.cover.data[0].attributes.formats.small.url)
          :
          "/noImage.svg"}
        id={exhib.id}
        startDate={exhib.attributes.startDate}
        finishDate={exhib.attributes.finishDate}
      />
    </SwiperSlide>
  )
  )}
  <SwiperSlide className="slideContainer">
    <Card className="exhibitionCard lastCard" sx={{ height: 430 }}>
      <Link to={"/exhibitions"} className="linkToMore">
        <CardContent>
          <Typography gutterBottom variant="h5" component="div" sx={{ fontWeight: 400
}}>
            Смотреть еще...
          </Typography>
        </CardContent>
      </Link>
    </Card>
  </SwiperSlide>
</div>
</Swiper>
    <button className="sliderBtn sliderBtnNext" onClick={() =>
swiperRef.current.slideNext()}><ArrowForwardOutlinedIcon /></button>
  </div>
</section>
<Divider></Divider>
<section className="wrapper">
  <h1 className="sectionTitle">Художники</h1>
  <div className="pictures">
    {artistApiData?.map((art) => (
      <Artist
        key={art.id}
        name={art.attributes.name}
        image={imageUrl + art.attributes.avatar?.data.attributes.formats.thumbnail.url}
        id={art.id}
      />
    ))}
  </div>
  <div className="linkContainer">
    <Link to={"/artists"} className="linkToMore">
      Смотреть еще...
    </Link>
  </div>
</section>
<Divider></Divider>

```

```

    <section className="wrapper">
      <h1 className="sectionTitle">Образовательные услуги</h1>
      <div className="wrapperArticle">
        <div className="ck-content text-indent" dangerouslySetInnerHTML={{ __html:
articles?.attributes?.Content }}>
          </div>
        </div>
      </section>

    </div>
  );
}

```

export default Main;

2. Pages/ExhibitionDetail.js

```

import './exhibitionDetail.css'
import '../ckeditor.css'
import { useLocation, useParams } from 'react-router-dom'
import { useEffect, useState, useRef } from 'react'
import axios from 'axios'
import Zoom from 'react-medium-image-zoom'
import 'react-medium-image-zoom/dist/styles.css'

import { Swiper, SwiperSlide } from "swiper/react";
import { Navigation, Pagination, Autoplay, A11y } from 'swiper';
import ArrowForwardOutlinedIcon from '@mui/icons-material/ArrowForwardOutlined';
import ArrowBackOutlinedIcon from '@mui/icons-material/ArrowBackOutlined';

import ConvertDate from '../helpers/ConvertDate'

function ExhibitionDetail() {

  const [exhibition, setExhibition] = useState({});
  const [image, setImage] = useState("");
  const { id } = useParams();
  console.log('id = ' + id);

  const URL = `http://localhost:1337/api/exhibitions/${id}?populate=*`;
  console.log(URL);
  const imageURL = "http://localhost:1337";

  const swiperRef = useRef();

  const fetchData = async () => {
    let data;
    const result = await axios.get(URL).catch(
      (error) => {
        if (error.response) {
          // The request was made and the server responded with a status code

```

```

        // that falls out of the range of 2xx
        console.log(error.response.data);
        console.log(error.response.status);
        console.log(error.response.headers);
    } else if (error.request) {
        // The request was made but no response was received
        // `error.request` is an instance of XMLHttpRequest in the browser and an instance of
        // http.ClientRequest in node.js
        console.log(error.request);
    } else {
        // Something happened in setting up the request that triggered an Error
        console.log('Error', error.message);
    }
    console.log(error.config);
  }
);
data = result?.data?.data;
console.log('checking data ' + data);
setExhibition(data);
}

useEffect(() => {
  fetchData();
  window.scrollTo(0, 0);
}, []);

useEffect(() => {
  let temp = (exhibition?.attributes?.cover?.data[0]?.attributes.formats.large?.url ??
    exhibition?.attributes?.cover?.data[0]?.attributes.formats.medium?.url ??
    exhibition?.attributes?.cover?.data[0]?.attributes.formats.small?.url ??
    exhibition?.attributes?.cover?.data[0]?.attributes.formats.thumbnail.url ?? "");
  setImage((temp === "") ? "/noImage.svg" : imageURL + temp);
  }, [Array.isArray(exhibition?.attributes?.cover?.data)
exhibition?.attributes?.cover?.data?.length])

return (
  <div className='exhibitionDetail wrapper'>
    <div className='exhibitionDetail__topDec'>
      <div className='imgContainer'>
        <Zoom>
          <img src={image} alt='Выставка' />
        </Zoom>
      </div>

      <div className='exhibitionDetail__textBlock1'>
        <h2 className='exhibitionDates'>
          {`${(exhibition.attributes?.startDate) === null ?
            "
            :
            exhibition.attributes?.finishDate ?
            ConvertDate(exhibition.attributes?.startDate, false)
            :

```

```

        ConvertDate(exhibition.attributes?.startDate, true)}

        ${exhibition.attributes?.finishDate} === null ?
        "
        :
        `— ${ConvertDate(exhibition.attributes?.finishDate)}`${
        }
    </h2>
    <h1>{exhibition.attributes?.title}</h1>
    <h2>{exhibition.attributes?.typeOfFineArt}</h2>
    <div className='exhibitionDetail__artists'>
        {exhibition.attributes?.artists?.data.map((name) => {
            return <h2>{name.attributes.name}</h2>
        })}
    </div>
    <h3 className='shortDescription'>{exhibition.attributes?.about}</h3>
</div>

<div className='exhibitionDetail__bottomDec'>
    <div className='ck-content text-indent' dangerouslySetInnerHTML={{ __html:
exhibition.attributes?.descriptionNew }}></div>
</div>

{exhibition.attributes?.pictureStory.data ?
    <div className='titleContainer'>
        <h1 className="sectionTitle">Фотоотчет с мероприятия</h1>
    </div>
    <div className='sliderWrapper sliderWrapperExhib'>
        <button className="sliderBtnExhib sliderBtnPrev" onClick={() =>
swiperRef.current.slidePrev()}><ArrowBackOutlinedIcon /></button>
        <Swiper className="sliderExhibition wrapper"
            slidesPerView={1}
            modules={[Navigation, Pagination, A11y, Autoplay]}
            spaceBetween={40}
            loop
            onSlideChange={() => console.log("slide change")}
            pagination={{ clickable: true }}
            // autoplay={{ delay: 5000 }}
            onSwiper={(swiper) => {
                swiperRef.current = swiper;
            }}
        >

        <div >
            {exhibition.attributes?.pictureStory.data.map((pic) => {
                let temp = pic.attributes.formats?.large?.url ??
                pic.attributes.formats?.medium?.url ??
                pic.attributes.formats?.small?.url ??
                pic.attributes.formats?.thumbnail?.url ??

```

```

        "/noImage.svg";
    return (
        <SwiperSlide className="slideContainer exhibitionSlide">
            <Zoom>
                <img className='pictureStoryImg' src={
                    temp === "/noImage.svg" ? "/noImage.svg" : (imageUrl + temp)
                }></img>
            </Zoom>
        </SwiperSlide>
    )
    }}}
</div>
</Swiper>
        <button className="sliderBtnExhib sliderBtnNext" onClick={() =>
swiperRef.current.slideNext()}><ArrowForwardOutlinedIcon /></button>
    </div>
</>
:
<></>
}
</div>
)
}

```

export default ExhibitionDetail;

3. Pages/Exhibitions.js

```

import { useEffect, useState } from "react";
import "../exhibitions.css";
import "../ckeditor.css"

```

```

import axios from "axios";
import { Link } from 'react-router-dom';
import Card from '@mui/material/Card';
import CardActions from '@mui/material/CardActions';
import CardContent from '@mui/material/CardContent';
import CardMedia from '@mui/material/CardMedia';
import Button from '@mui/material/Button';
import Typography from '@mui/material/Typography';
import { Pagination } from '@mui/material';
import Exhibition from "../components/Exhibition";

```

function Exhibitions({ searchValue, setSearchValue, type }) { //type - тип архива, используется для страницы с архивом выставок

```

    useEffect(() => {
        setSearchValue("");
        window.scrollTo(0, 0);
    }, []);

```

```

    const imageUrl = "http://localhost:1337"

```

```

const [pageApi, setPageApi] = useState(1);
const [apiData, setApiData] = useState([]);
const [count, setCount] = useState(0);
const PER_PAGE = 12; //отвечает за кол-во элементов на странице
const [isArchived, setIsArchived] = useState((type === 'arch') ? true : false);

```

```

const ApiAddress = axios.create({
  baseURL: "http://localhost:1337/api/exhibitions"
});

```

```

useEffect(() => {
  if (isArchived) {

```

```

    ApiAddress.get(`?pagination[page]=${pageApi}&pagination[pageSize]=${PER_PAGE}&filters[$and][0][isArchived][$eq]=true&populate=*`)
      .then((response) => {setApiData(response.data.data);
        console.log(response.data.data)})
      .catch((err) => console.log(err));
  }
  else {

```

```

    ApiAddress.get(`?pagination[page]=${pageApi}&pagination[pageSize]=${PER_PAGE}&filters[$or][0][isArchived][$eq]=false&filters[$or][1][isArchived][$null]=true&populate=*`)
      .then((response) => {setApiData(response.data.data);
        console.log(response.data.data)})
      .catch((err) => console.log(err));
  }

```

```

  }, [pageApi, type]);

```

```

//забираем из api кол-во элементов для вычисления количества страниц
useEffect(() => {
  ApiAddress.get()
    .then((response) => {
      setCount(Math.ceil(response.data.meta.pagination.total / PER_PAGE));
    })
    .catch((err) => console.log(err));
}, []);

```

```

return (
  <div className="main">
    <h1 className="title">{isArchived ? 'Архив выставок' : 'Выставки'}</h1>
    <div className="wrapper cardWrapper">

      {apiData?.map((exhib) => {
        return <Exhibition
          key={exhib.id}
          cardTitle={exhib.attributes.title}
          about={exhib.attributes.about}

```

```

        image={exhib.attributes.cover?.data ?
            (imageUrl + exhib.attributes.cover.data[0].attributes.formats.small.url)
            :
            "/noImage.svg"}
        id={exhib.id}
        startDate={exhib.attributes.startDate}
        finishDate={exhib.attributes.finishDate}
    />

    )))
</div>
<Pagination
  className='wrapper pagination'
  count={count} // передаем количество страниц
  size="large"
  variant="outlined"
  shape="rounded"
  onChange={(e, value) => {
    window.scrollTo(0, 0);
    setPageApi(value)
  }}
  sx={{
    display: 'flex',
    justifyContent: 'flex-end',
    marginBottom: '80px',
  }}
/>
</div>
);
}

```

export default Exhibitions;

3./Index.js

```

import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import { BrowserRouter } from "react-router-dom";

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <BrowserRouter>
    <App />
  </BrowserRouter>
);

```

Приложение 3

Листинг файла cron-tasks.js:

```
module.exports = {
  exhibitionsUpdate: {
    task: async ({ strapi }) => {
      try {
        let currentDate = new Date();
        currentDate.setHours(0,0,0,0);
        console.log(`${currentDate.getDate()}.${currentDate.getMonth()}.${currentDate.getFullYear()}`);
        const exhibitions = await strapi.entityService.findMany('api::exhibition.exhibition', {
          fields: ['id', 'isArchived', 'finishDate'],
        });
        exhibitions.map(async (exhib) => {
          try {
            if (exhib.finishDate === null) return;
            if (exhib.isArchived === true) return;
            const isValidDate = (Y, M, D) => {
              var d = new Date(Y, --M, D);
              return Y === d.getFullYear() && M === d.getMonth() && D === d.getDate();
            }
            let dateArr = exhib.finishDate.split('.');
            let day = parseInt(dateArr[0]);
            let month = parseInt(dateArr[1]);
            let year = parseInt(dateArr[2]);
            if (!isValidDate(year, month, day)) { //проверка даты на валидность
              console.log('!----- Выставка с id='+exhib.id+' имеет некорректную дату');
              return;
            }
            const finishDate = new Date(dateArr[2], dateArr[1] - 1, dateArr[0]);
            console.log('дата окончания ' + finishDate.toString());
            if (finishDate < currentDate) {
              let result = await strapi.entityService.update('api::exhibition.exhibition', exhib.id, {
                data: {
                  isArchived: true,
                }
              });
              console.log('Архивировано при сравнении месяца, id выставки:' + exhib.id);
            }
            else console.log('Архивации не произошло, id выставки:' + exhib.id);
          } catch (error) {
            console.log(error);
          }
        });
      } catch (error) {
        console.log(error)
      }
    },
    options: {
      rule: "00 00 00 * * *",
    },
  },
};
```