

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

КУРСОВОЕ ПРОЕКТИРОВАНИЕ

Проектирование веб-сервиса по фотопечати

Б-ИС-21-01-166537.8849-с.18.000.КП.00

Студент
гр. БИС-21-02



В.С. Суховей

Руководитель,
ст. преподаватель



О.Б. Богданова

Руководитель,
Кандидат химических наук,
доцент



Б.К. Васильев

Владивосток 2025

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ
на курсовое проектирование

Студенту: гр. БИС-21-1 Суховой Владиславу Сергеевичу

Срок сдачи работы: 16.01.2025

Задание 1. Описать используемые инструменты для разработки

Задание 2. Описать требования к веб-сервису

Задание 3. Спроектировать веб-сервис

Структура отчета по курсовому проектированию:

- Введение;
- Описание процесса;
- Описание используемых инструментов;
- Описание требований к веб-сервису;
- Проектирование;
- Заключение;
- Список использованных источников;

Отчет по курсовой работе оформляется в соответствии с СК-СТО-ТР-04-1.005-2015 «Требования к оформлению текстовой части выпускных квалификационных работ, курсовых работ (проектов), рефератов, контрольных работ, отчётов по практикам, лабораторным работам».

Руководитель

Задание получил:



Богданова О.Б.

Суховой В.С.

Аннотация

Курсовая работа на тему: «Проектирование системы автоматизации школьной библиотеки».

Автор: студент гр. БИС-21-1, Суховой В.С.

Руководитель: Старший преподаватель Богданова О.Б.

Работа посвящена проектированию веб-сервиса для автоматизации процессов фотопечати. Основная цель — упрощение взаимодействия пользователей с сервисом, включая загрузку, редактирование и оформление заказов на печать фотографий. Важными функциями являются выбор форматов, проверка качества изображений, интеграция с внешними системами (Яндекс.Диск, ЮKassa, Voberry), автоматизация формирования заказов и уведомлений. В работе предложена структура базы данных, описаны алгоритмы работы системы, пользовательские сценарии (User Flow) и проектирование интерфейса. Ожидается, что внедрение сервиса повысит удобство и скорость обработки заказов как для клиентов, так и для администрации.

Работа состоит из 43 страниц, содержит 15 источников, 34 изображений.

Содержание

Введение.....	4
1 Описание процесса.....	5
2 Описание используемых инструментов для разработки.....	5
2.1 Frontend часть.....	6
2.2 Backend часть	6
2.3 Сборка и автоматизация.....	7
2.4 Хостинг и серверная инфраструктура	7
2.5 Интеграции и внешние сервисы.....	8
3 Описание требований к веб-сервису	9
3.1 Основные требования.....	9
3.2 Функциональные требования	9
3.3 Ограничения и допустимые предположения	11
4 Проектирование.....	12
4.1 ER-диаграмма.....	12
4.2 Диаграмма классов	19
4.3 Алгоритм работы веб-сервиса	22
4.4 User Flow	32
4.5 Макеты страниц веб-сервиса	34
Заключение	42
Список использованных источников	43

Введение

Целью данной работы является проектирование веб-сервиса для фотопечати, который позволит пользователям загружать, редактировать и оформлять заказы на печать фотографий через интуитивно понятный интерфейс. Важным аспектом разработки является интеграция с внешними системами (Яндекс.Диск, ЮKassa, Vohberry) для автоматизации загрузки файлов, оплаты и доставки, а также обеспечение качественного контроля фотографий.

В рамках курсовой работы будут рассмотрены основные этапы проектирования веб-сервиса, включая постановку основных и функциональных требований, разработку структуры базы данных, описание алгоритмов работы системы и проектирование пользовательских интерфейсов. Также будут представлены пользовательские сценарии (User Flow), макеты интерфейсов и предложена клиент-серверная архитектура системы.

Ожидается, что разработка такого веб-сервиса повысит удобство пользователей, улучшит их взаимодействие с процессом фотопечати и упростит управление заказами для администрации сервиса.

1 Описание процесса

На текущий момент процесс фотопечати на предприятии ИП Кочеткова Романа осуществляется вручную. Клиенты направляют фотографии на печать через электронную почту, указывая в сообщении необходимые параметры, такие как формат, количество копий и комментарии. Данный процесс представлен на диаграмме на рисунке 1.

Алгоритм текущего процесса фотопечати (до автоматизации)

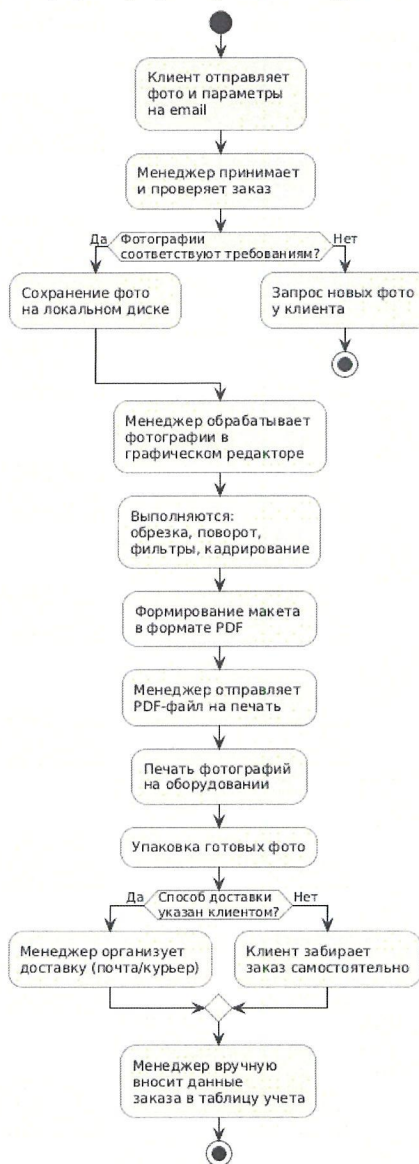


Рисунок 1— Алгоритм работы бизнеса

Автоматизация этого процесса с помощью веб-сервиса позволит исключить ручную обработку на ключевых этапах, повысить точность и скорость выполнения заказов, а также улучшить взаимодействие с клиентами.

2 Описание используемых инструментов для разработки

При создании веб-сервиса foto4ki.ru применяются различные технологии и инструменты, утверждённые совместно с заказчиком, которые помогают решать задачи разработки, автоматизации, обеспечения стабильности и интеграции с внешними системами.

2.1 Frontend часть

Чтобы обеспечить удобное и динамическое взаимодействие пользователей с веб-сервисом, при разработке фронтенда были выбраны следующие инструменты и технологии:

a) WordPress [1]:

1) Используется в качестве основы для сайта и системы управления контентом (CMS);

2) Предоставляет готовую структуру и функционал (плагины, темы, административную панель), упрощая процесс создания и редактирования контента.

b) JavaScript [2]:

1) Основной язык для реализации интерактивных элементов пользовательского интерфейса;

2) Используется как в связке с jQuery, так и автономно для таких функций, как динамический расчёт стоимости заказа, валидация форм, отображение попап-окон и т.д.

c) jQuery [3]:

1) Библиотека JavaScript, которая значительно упрощает работу с элементами DOM (Document Object Model);

2) Обеспечивает удобный API для управления HTML-элементами, обработки событий, создания анимаций и взаимодействия с сервером (AJAX-запросы);

3) Уменьшает объём кода и делает разработку клиентской части более удобной.

В совокупности использование вышеописанных технологий на фронтенде позволяет создать гибкий и настраиваемый пользовательский интерфейс, обеспечивая быструю загрузку страниц, понятную структуру сайта и удобное взаимодействие с контентом.

2.2 Backend часть

Серверная часть отвечает за бизнес-логику, хранение данных и взаимодействие с внешними сервисами. При её реализации задействованы следующие технологии:

a) PHP [4]:

1) Является основным языком серверной части;

2) Обеспечивает логику бизнес-процессов (обработка заказов, взаимодействие с внешними API, генерация PDF, сохранение и чтение из базы данных).

b) MySQL [5]:

1) Система управления базами данных, где хранится основная информация о заказах, пользователях, сессиях и других сущностях сервиса;

2) Хорошо интегрируется с PHP, легко масштабируется и имеет богатый набор инструментов для резервного копирования, репликации и мониторинга.

Использование связки PHP и MySQL обеспечивает высокую производительность и удобство разработки. Новейшие возможности языка позволяют оптимизировать архитектуру приложения, а надёжность MySQL гарантирует сохранность ключевой информации.

2.3 Сборка и автоматизация

Для упрощения и ускорения разработки фронтенда, а также для повышения качества конечного продукта, будет использован специальный инструмент автоматизации Gulp [6]:

a) Позволяет настроить «конвейер» сборки, включающий минификацию [7] HTML, CSS, JavaScript и оптимизацию изображений;

b) Имеет механизм «watch», автоматически реагирующий на изменения в файлах и пересобирающий проект.

Благодаря Gulp значительно сокращается время на выполнение рутинных операций, а разработчики могут сосредоточиться на написании функционального кода. В результате повышается стабильность и скорость загрузки веб-сервиса для конечных пользователей.

2.4 Хостинг и серверная инфраструктура

Для развертывания сервиса foto4ki.ru выбрана конфигурация, позволяющая поддерживать стабильную работу приложения и обрабатывать необходимые объёмы данных:

a) Хостинг на reg.ru — даёт базовые инструменты администрирования, включая регистрацию доменного имени, настройку SSL и управление серверной инфраструктурой;

b) Серверные характеристики:

1) 2 ядра CPU;

2) 4 ГБ оперативной памяти;

3) 40 ГБ SSD;

4) Операционная система Ubuntu 22.04;

5) Веб-сервер Nginx.

Благодаря подобранным параметрам сервер будет надёжно обрабатывать одновременные пользовательские запросы, поддерживать быструю загрузку страниц и обеспечивать возможность масштабирования при росте нагрузки.

2.5 Интеграции и внешние сервисы

Одной из ключевых особенностей foto4ki.ru является тесная интеграция с внешними сервисами, упрощающая и автоматизирующая многие процессы. Ниже представлены основные модули:

- a) Google Sheets: Взаимодействие через Google API, для добавления новых строк в таблицу при каждом заказе.
- a) Vohberry [8]:
 - 1) Предоставляет модуль для выбора пункта выдачи на карте и API для расчёта доставки;
 - 2) Информация о заказе (точка доставки и цена) добавляется в личный кабинет Vohberry, упрощая логистику.
- b) ЮKassa [9]:
 - 1) Предоставляет модуль для безопасной онлайн-оплаты;
 - 2) Информация о платеже добавляется в личный кабинет Vohberry.
- c) Яндекс.Диск [10]:
 - 1) Позволяет загружать фотографии напрямую из облачного хранилища пользователя;
 - 2) Сгенерированные PDF-файлы с фотографиями отправляются на диск администрации сервиса для последующей печати.
- d) Telegram [11]:
 - 1) Уведомляет администратора о новых заказах в реальном времени;
 - 2) Позволяет оперативно реагировать на заявки, повышая качество обслуживания.
- e) Email:
 - 1) Служит для оповещений клиентов о статусе заказа;
 - 2) Дублирует информацию для пользователей, обеспечивая прозрачность и удобство.

Благодаря комплексному взаимодействию со сторонними API существенно сокращается ручная работа менеджеров и повышается удобство пользователей, которые получают быстрые и точные сервисы доставки, оплаты и уведомлений.

3 Описание требований к веб-сервису

3.1 Основные требования

Помимо основного функционала, важную роль играет ряд нефункциональных требований, направленных на надёжность, производительность и удобство использования веб-сервиса:

- a) Производительность:
 - 1) Быстрое открытие страниц и обработка действий пользователя;
 - 2) Низкое время отклика при одновременных запросах.
- b) Надёжность и устойчивость:
 - 1) Сохранение данных сессий до двух недель;
 - 2) Резервное копирование баз данных и основных файлов.
- c) Масштабируемость:
 - 1) Возможность увеличения ресурсов (CPU, RAM, диск) без радикальной перестройки архитектуры;
 - 2) Гибкое разделение сервисов (фоторедактор, платежи, генерация PDF) при росте нагрузки.
- d) Безопасность:
 - 1) Использование SSL/TLS для передачи персональных данных и операций оплаты;
 - 2) Обновление серверного и прикладного ПО для устранения уязвимостей.
- e) Удобство использования:
 - 1) Интуитивно понятная навигация и дизайн;
 - 2) Адаптивная верстка для корректной работы на мобильных устройствах.

Указанные нефункциональные требования обеспечивают стабильную работу сервиса и комфорт пользователя. Они задают ориентиры для качества и доступности ресурса, а также для его будущего развития в условиях возросшей нагрузки и потенциальных изменений во внешних интеграциях.

3.2 Функциональные требования

Для обеспечения корректной работы основного функционала веб-сервиса и удобного взаимодействия пользователей с системой выделены следующие функциональные требования:

- a) Идентификация пользователей:
 - 1) Анонимный доступ с сохранением сессий на основе IP-адреса;

- 2) Возможность продолжить ранее начатый заказ за счёт хранения данных сессии.
- b) Загрузка фотографий:
 - 1) Загрузка с устройства через стандартный диалог выбора файлов;
 - 2) Загрузка с Яндекс.Диска с авторизацией и отображением содержимого облака;
 - 3) Наличие буфера, где можно управлять всеми загруженными снимками.
- c) Рабочее пространство и коллекции:
 - 1) Вкладки-коллекции для разных форматов (7,5 x 7,5 см, 10 x 15 см, 7,5 x 10 см, 13 x 18 см, 10 x 10 см, 20 x 30 см, 10 x 10 см);
 - 2) Masonry Layout для удобного отображения фотографий;
 - 3) Возможность базового редактирования (поворот, обрезка, фильтры) и удаления.
- d) Контроль качества фотографий:
 - 1) Рейтинг в виде звёзд, отображающий уровень качества фотографии;
 - 2) Маркировка фотографий с низким разрешением.
- e) Формирование заказа и корзина:
 - 1) Динамический пересчёт стоимости при добавлении/удалении снимков и выборе дополнительных опций (самоклеящиеся, премиум-качество);
 - 2) Заполнение контактных и адресных данных, интеграция с Vohberry для выбора пункта выдачи;
 - 3) Возможность оплаты онлайн через ЮKassa или после подтверждения заказа.
- f) Оплата и подтверждение:
 - 1) Поддержка онлайн-платежей различными методами (банковские карты, электронные кошельки);
 - 2) Отображение статуса заказа и отправка уведомлений на e-mail.
- g) Автоматическая обработка заказов:
 - 1) Генерация PDF-файлов по шаблонам и сохранение на Яндекс.Диск;
 - 2) Добавление записей в Google Sheets и оповещение менеджера в Telegram.

Данные функциональные требования обеспечивают пользователям удобные инструменты для загрузки и редактирования фотографий, а также для быстрого оформления заказа с нужными параметрами. Система же, в свою очередь, реализует прозрачный процесс обработки, информируя и пользователя, и администратора о всех важных событиях.

3.3 Ограничения и допустимые предположения

Важно учесть, что при разработке сервиса могут возникать ситуации, накладывающие ограничения на объём хранения или продолжительность сохранения данных. Ниже описаны основные из них:

- а) Объём хранимых фотографий: Ограничение размеров загружаемых файлов до 20 МБ на один файл;
- б) Срок хранения фотографий:
 - 1) Хранение загруженных файлов и данных незавершённого заказа не более двух недель;
 - 2) Автоматическое удаление для освобождения ресурсов.
- с) Регион доставки:
 - 1) Взаимодействие с Vohberry ориентировано на доставку по России (включая самовывоз в г. Красногорске);
 - 2) Возможность подключения других логистических сервисов при выходе на новые рынки.

Указанные ограничения и предположения помогают заранее определить границы системы и избежать неоправданной траты ресурсов. При появлении новых требований или расширении географии продаж эти пункты могут быть пересмотрены и доработаны.

4 Проектирование веб-сервиса

4.1 ER-диаграмма

Для эффективного проектирования и управления данными веб-сервиса foto4ki.ru была разработана ER-диаграмма (Entity-Relationship Diagram) [12]. Она является основой для построения структуры базы данных и описывает основные сущности предметной области, их атрибуты и взаимосвязи.

ER-диаграмма помогает визуализировать и упорядочить данные, обеспечивая:

- Чёткое понимание, как информация организована и связана внутри системы;
- Обоснование логической структуры базы данных, что особенно важно для будущего масштабирования и интеграции;
- Возможность минимизировать избыточность данных за счёт правильного проектирования связей между сущностями.

В процессе проектирования веб-сервиса была создана следующая диаграмма, изображённая на рисунке 2.

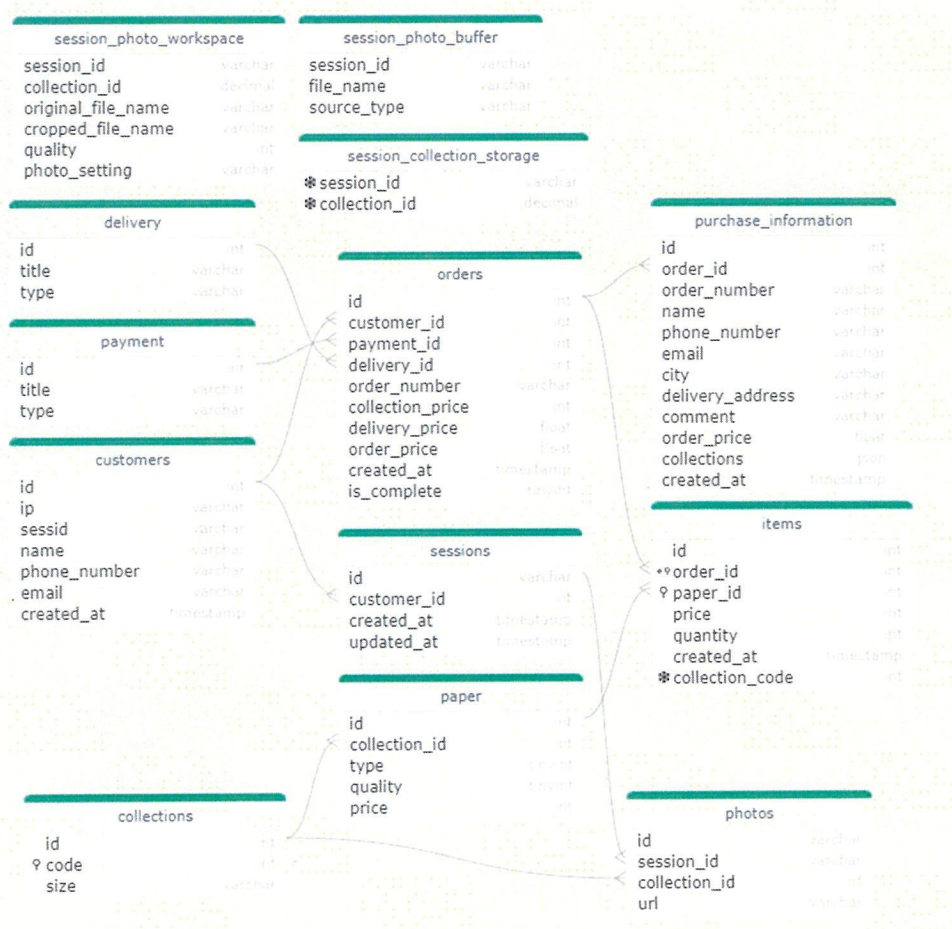


Рисунок 2 — ER-диаграмма

Рассмотрим каждую таблицу подробнее:

а) **customers** (Покупатели) (рисунок 3): Данная таблица содержит информацию о клиентах сервиса. В ней присутствуют следующие поля:

- 1) **id** — уникальный идентификатор покупателя;
- 2) **ip, sessid** — идентификация пользователя по IP-адресу или сессионному идентификатору;
- 3) **name, phone_number, email** — контактные данные;
- 4) **created_at** — дата создания записи о пользователе.

customers	
id	int
ip	varchar
sessid	varchar
name	varchar
phone_number	varchar
email	varchar
created_at	timestamp

Рисунок 3 — Таблица customers

б) **sessions** (Сессии) (рисунок 4): Данная таблица управляет текущей активностью пользователя. В ней присутствуют следующие поля:

- 1) **id** — уникальный идентификатор сессии;
- 2) **customer_id** — ссылка на покупателя;
- 3) **created_at, updated_at** — временные метки создания и обновления сессии.

sessions	
id	varchar
customer_id	int
created_at	timestamp
updated_at	timestamp

Рисунок 4 — Таблица sessions

с) `session_photo_buffer` (Буфер загрузки фото) (рисунок 5): Данная таблица хранит фотографии, загруженные пользователем, до их добавления в коллекции. В ней присутствуют следующие поля:

- 1) `session_id` — идентификатор сессии пользователя
- 2) `file_name` — имя загруженного файла;
- 3) `source_type` — источник файла (например, локальное устройство или Яндекс.Диск).

session_photo_buffer	
session_id	varchar
file_name	varchar
source_type	varchar

Рисунок 5 — Таблица `session_photo_buffer`

д) `session_photo_workspace` (Рабочее пространство с фото) (рисунок 6): Данная таблица содержит фотографии, которые пользователь добавил в коллекции, а также их настройки. В ней присутствуют следующие поля:

- 1) `session_id` — идентификатор сессии пользователя;
- 2) `collection_id` — ссылка на коллекцию;
- 3) `original_file_name`, `cropped_file_name` — исходная и обработанная версии файла;
- 4) `quality` — оценка качества изображения;
- 5) `photo_setting` — дополнительные настройки фото (например, поворот, фильтры).

session_photo_workspace	
session_id	varchar
collection_id	decimal
original_file_name	varchar
cropped_file_name	varchar
quality	int
photo_setting	varchar

Рисунок 6 — Таблица `session_photo_workspace`

е) `session_collection_storage` (Выбранные коллекции в сессии) (рисунок 7): Данная таблица хранит связь между сессией пользователя и выбранными коллекциями. В ней присутствуют следующие поля:

- 1) `session_id` — идентификатор сессии пользователя;
- 2) `collection_id` — идентификатор коллекции.

session_collection_storage	
* session_id	varchar
* collection_id	decimal

Рисунок 7 — Таблица `session_collection_storage`

ф) `collections` (Коллекции) (рисунок 8): Данная таблица описывает форматы фотографий (например, 10×15, 20×30). В ней присутствуют следующие поля:

- 1) `id` — уникальный идентификатор коллекции;
- 2) `code` — числовой код коллекции;
- 3) `size` — размер фотографий в коллекции (например, "10×15").

collections	
id	int
? code	int
size	varchar

Рисунок 8 — Таблица `collections`

г) `photos` (Фотографии) (рисунок 9): Данная таблица хранит фотографии, загруженные пользователем в коллекции. В ней присутствуют следующие поля:

- 1) `id` — уникальный идентификатор коллекции;
- 2) `session_id` — ссылка на текущую сессию;
- 3) `collection_id` — коллекция, к которой относится фотография;
- 4) `url` — ссылка на файл изображения.

photos	
id	varchar
session_id	varchar
collection_id	int
url	varchar

Рисунок 9 — Таблица photos

h) paper (Тип бумаги) (рисунок 10): Данная таблица содержит информацию о типах и качестве бумаги для печати фотографий. В ней присутствуют следующие поля:

- 1) id — уникальный идентификатор;
- 2) collection_id — ссылка на формат коллекции;
- 3) type, quality — тип бумаги (обычная/самоклеящаяся) и её качество (обычное/премиум);
- 4) price — стоимость.

paper	
id	int
collection_id	int
type	tinyint
quality	tinyint
price	int

Рисунок 10 — Таблица paper

i) orders (Заказы) (рисунок 11): Данная таблица хранит информацию о заказах. В ней присутствуют следующие поля:

- 1) id — уникальный идентификатор заказа;
- 2) customer_id — ссылка на покупателя;
- 3) payment_id, delivery_id — способы оплаты и доставки;
- 4) order_number — номер заказа;
- 5) collection_price, delivery_price, order_price — стоимость коллекций, доставки и общая сумма заказа;
- 6) created_at — дата создания заказа;
- 7) is_complete — флаг завершённости заказа.

orders	
id	int
customer_id	int
payment_id	int
delivery_id	int
order_number	varchar
collection_price	int
delivery_price	float
order_price	float
created_at	timestamp
is_complete	tinyint

Рисунок 11 — Таблица orders

j) items (Позиции заказа) (рисунок 12): Данная таблица хранит информацию о конкретных коллекциях в заказе. В ней присутствуют следующие поля:

- 1) id — уникальный идентификатор заказа;
- 2) order_id — ссылка на заказ;
- 3) paper_id — ссылка на тип бумаги;
- 4) price, quantity — цена и количество фотографий в данной позиции;
- 5) collection_code — код коллекции;
- 6) created_at — дата создания позиции заказа.

items	
id	int
order_id	int
paper_id	int
price	int
quantity	int
created_at	timestamp
collection_code	int

Рисунок 12 — Таблица items

к) `purchase_information` (Информация о покупке) (рисунок 13): Данная таблица хранит контактные данные покупателя и подробности покупки. В ней присутствуют следующие поля:

- 1) `order_id` — ссылка на заказ;
- 2) `order_number` — номер заказа;
- 3) `name`, `phone_number`, `email`, `city`, `delivery_address` — контактные данные и адрес доставки;
- 4) `comment` — комментарий покупателя;
- 5) `order_price` — итоговая сумма заказа;
- 6) `collections` — JSON-структура с деталями коллекций;
- 7) `created_at` — дата создания информации о заказе;

purchase_information	
id	int
order_id	int
order_number	varchar
name	varchar
phone_number	varchar
email	varchar
city	varchar
delivery_address	varchar
comment	varchar
order_price	float
collections	json
created_at	timestamp

Рисунок 13 — Таблица `purchase_information`

л) `payment` (Способы оплаты) (рисунок 14): Данная таблица описывает доступные способы оплаты. В ней присутствуют следующие поля:

- 1) `id` — уникальный идентификатор;
- 2) `title` — название способа (например, "ЮKassa");
- 3) `type` — технический тип оплаты.

payment	
id	int
title	varchar
type	varchar

Рисунок 14 — Таблица payment

m) delivery (Способы доставки) (рисунок 15): Данная таблица хранит данные о вариантах доставки. В ней присутствуют следующие поля:

- 1) id — уникальный идентификатор;
- 2) title — название (например, "Boxberry");
- 3) type — описание доставки.

delivery	
id	int
title	varchar
type	varchar

Рисунок 15 — Таблица delivery

Представленная ER-диаграмма демонстрирует логику взаимодействия данных в веб-сервисе foto4ki.ru.

4.2 Диаграмма классов

Для реализации бизнес-логики веб-сервиса foto4ki.ru необходима структура программных компонентов, которая бы эффективно отражала предметную область системы.

Диаграмма классов [12] служит нескольким целям:

- a) Отражение предметной области: классы описывают сущности, связанные с пользователями, фотографиями, заказами и их атрибутами;
- b) Определение архитектуры приложения: помогает задать структуру кода и определить роли различных компонентов;
- c) Упрощение разработки: чёткая визуализация взаимодействия классов снижает вероятность ошибок при проектировании и реализации системы.

В процессе проектирования веб-сервиса была создана следующая диаграмма, изображённая на рисунке 16.

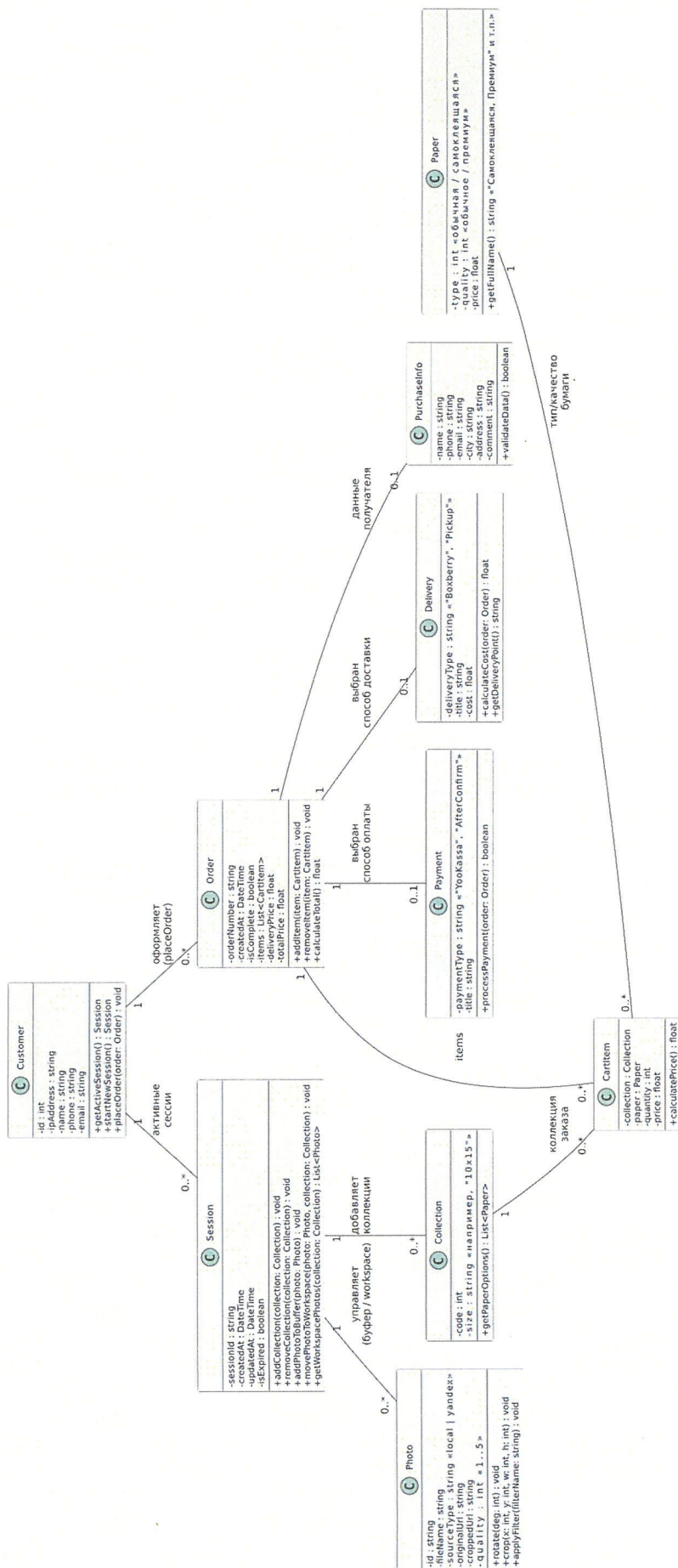


Рисунок 16 — Диаграмма классов

Основные классы:

- a) Customer (Покупатель)
 - 1) Хранит данные о пользователе (IP-адрес, контактные данные).
 - 2) Может запускать сессии и оформлять заказы.
- b) Session (Сессия)
 - 1) Управляет текущей активностью пользователя: выбранные коллекции, фотографии в буфере, фотографии в рабочем пространстве.
 - 2) Позволяет перемещать фотографии между буфером и рабочим пространством.
- c) Photo (Фотография)
 - 1) Представляет загруженные изображения: их URL, источник (локальный диск или Яндекс.Диск), качество.
 - 2) Имеет методы для редактирования (поворот, кадрирование, фильтры).
- d) Collection (Коллекция)
 - 1) Описывает форматы фотографий (например, 10×15 см).
 - 2) Может возвращать доступные параметры бумаги для текущей коллекции.
- e) Paper (Тип бумаги)
 - 1) Характеризует тип и качество бумаги (обычная, самоклеящаяся, премиум).
 - 2) Влияет на стоимость печати.
- f) CartItem (Позиция заказа)
 - 1) Представляет отдельный элемент заказа, связанный с коллекцией и типом бумаги.
 - 2) Рассчитывает цену в зависимости от параметров.
- g) Order (Заказ)
 - 1) Хранит информацию о заказе: список позиций, общая стоимость, статус завершенности.
 - 2) Связан с покупателем, а также со способами доставки и оплаты.
- h) Payment (Оплата)
 - 1) Описывает способ оплаты (например, через ЮKassa).
 - 2) Содержит метод обработки платежа.
- i) Delivery (Доставка)

- 1) Содержит информацию о способе доставки (Boxberry, самовывоз) и расчёте стоимости доставки.
- j) PurchaseInfo (Информация о заказе)
 - 1) Хранит контактные данные получателя, адрес доставки и комментарии.
 - 2) Обеспечивает проверку корректности введённых данных.

Диаграмма классов UML позволяет визуализировать и описать архитектуру приложения, включая основные классы, их атрибуты, методы и связи.

4.3 Алгоритм работы веб-сервиса

Алгоритм работы веб-сервиса foto4ki.ru описывает последовательность действий пользователя и внутренние процессы системы, которые обеспечивают выполнение основных функций сервиса: загрузку и редактирование фотографий, формирование заказов, оплату и доставку.

Рассмотрим каждый модуль алгоритма отдельно.

Модуль «Start» (рисунок 17):

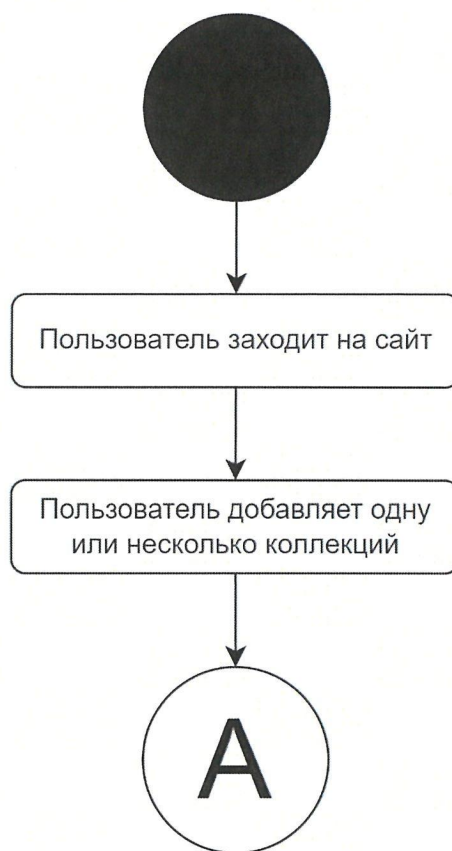


Рисунок 17 — Стартовый модуль алгоритма

На первом этапе пользователь заходит на сайт, где система инициализирует сессию и предоставляет доступ к основным функциям. Затем пользователь выбирает одну или несколько коллекций, соответствующих форматам фотографий, которые он планирует заказать. Эти коллекции появляются в рабочем пространстве и готовы для добавления фотографий.

Модуль формирования коллекций (рисунок 18):

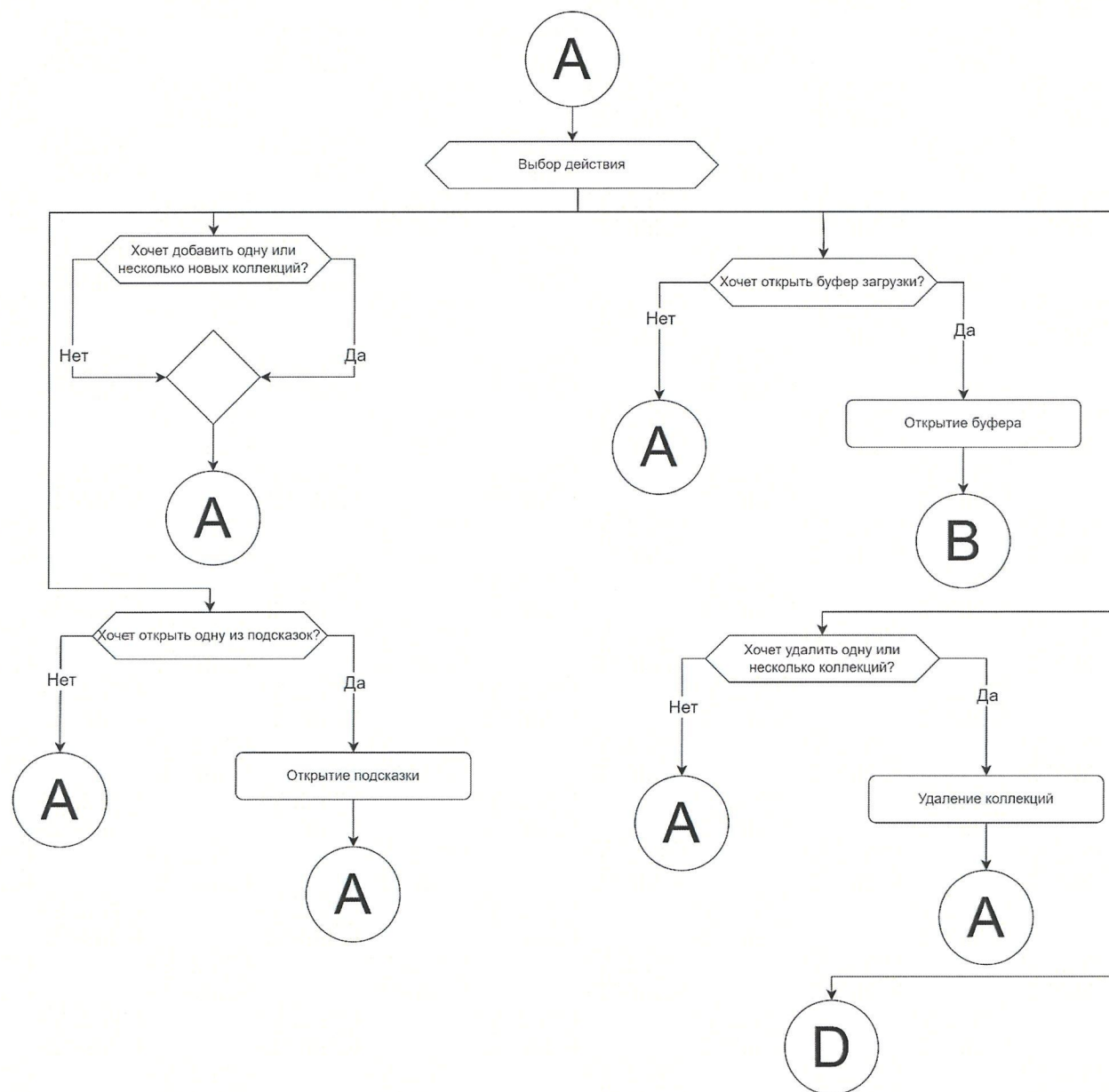


Рисунок 18 — Модуль формирования коллекций

На следующем этапе пользователь выбирает одно из доступных действий: добавление новых коллекций, открытие буфера загрузки, просмотр подсказок или удаление

существующих коллекций. Если пользователь решает добавить коллекции, система переходит к выбору форматов. Если открывается буфер загрузки, пользователь может начать загрузку фотографий. При выборе подсказок отображается справочная информация, а при удалении коллекций они удаляются из текущего рабочего пространства.

Модуль загрузки фотографий в буфер (рисунок 19):

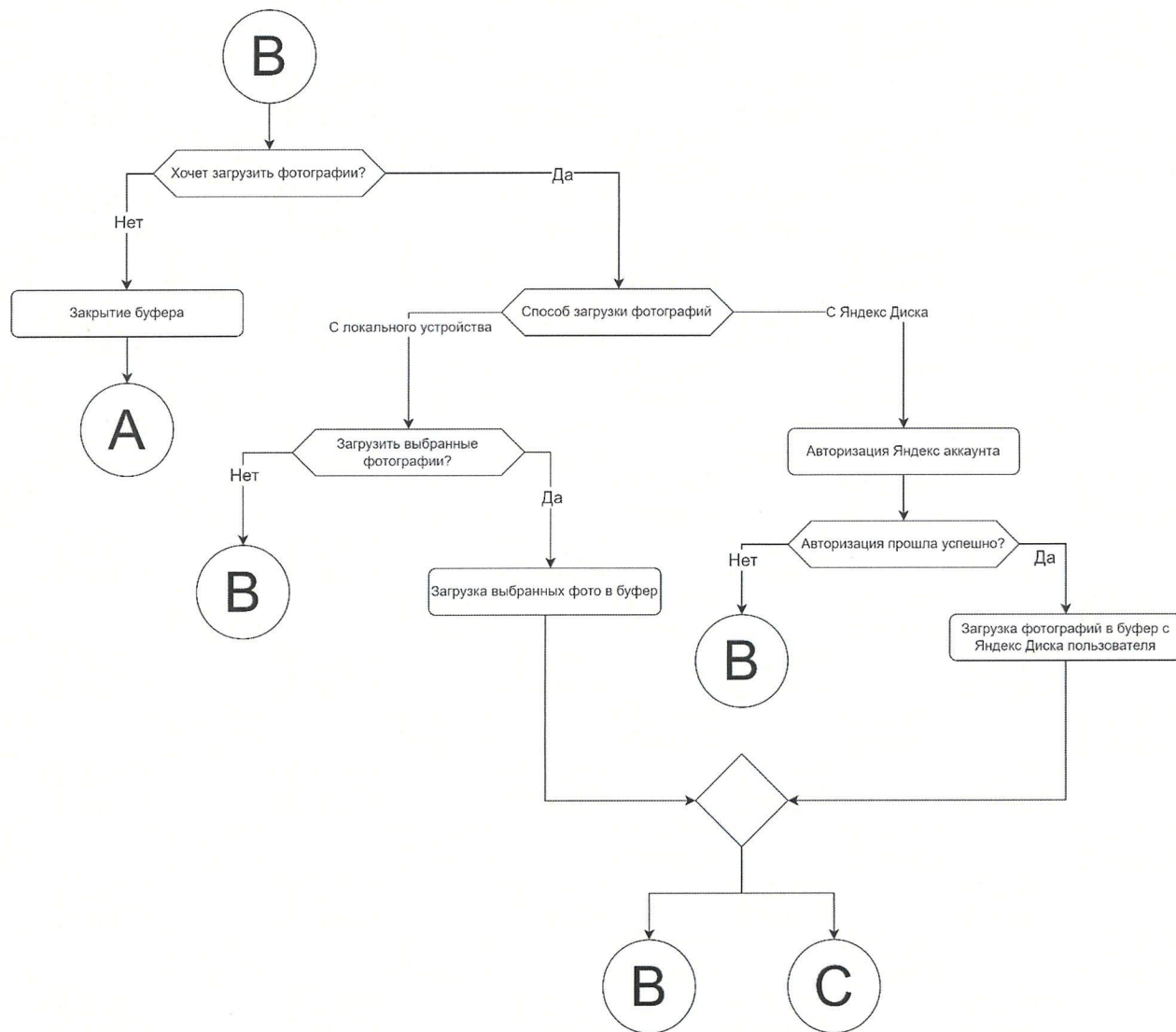


Рисунок 19 — Модуль загрузки фотографий в буфер

На данном этапе пользователь принимает решение о загрузке фотографий. Если пользователь выбирает загрузку, ему предлагается выбрать способ: с локального устройства или из Яндекс.Диска. При выборе загрузки с локального устройства пользователь выбирает файлы на своём устройстве, которые затем загружаются в буфер. Если выбирается загрузка из Яндекс.Диска, необходимо авторизоваться в аккаунте Яндекс. При успешной авторизации пользователю предоставляется доступ к файлам на его Яндекс.Диске, которые автоматически

подгружаются в буфер. Если авторизация не удалась или пользователь отменяет процесс, буфер закрывается, и пользователь возвращается к рабочему пространству.

Модуль работы с фотографиями в буфере (рисунок 20):

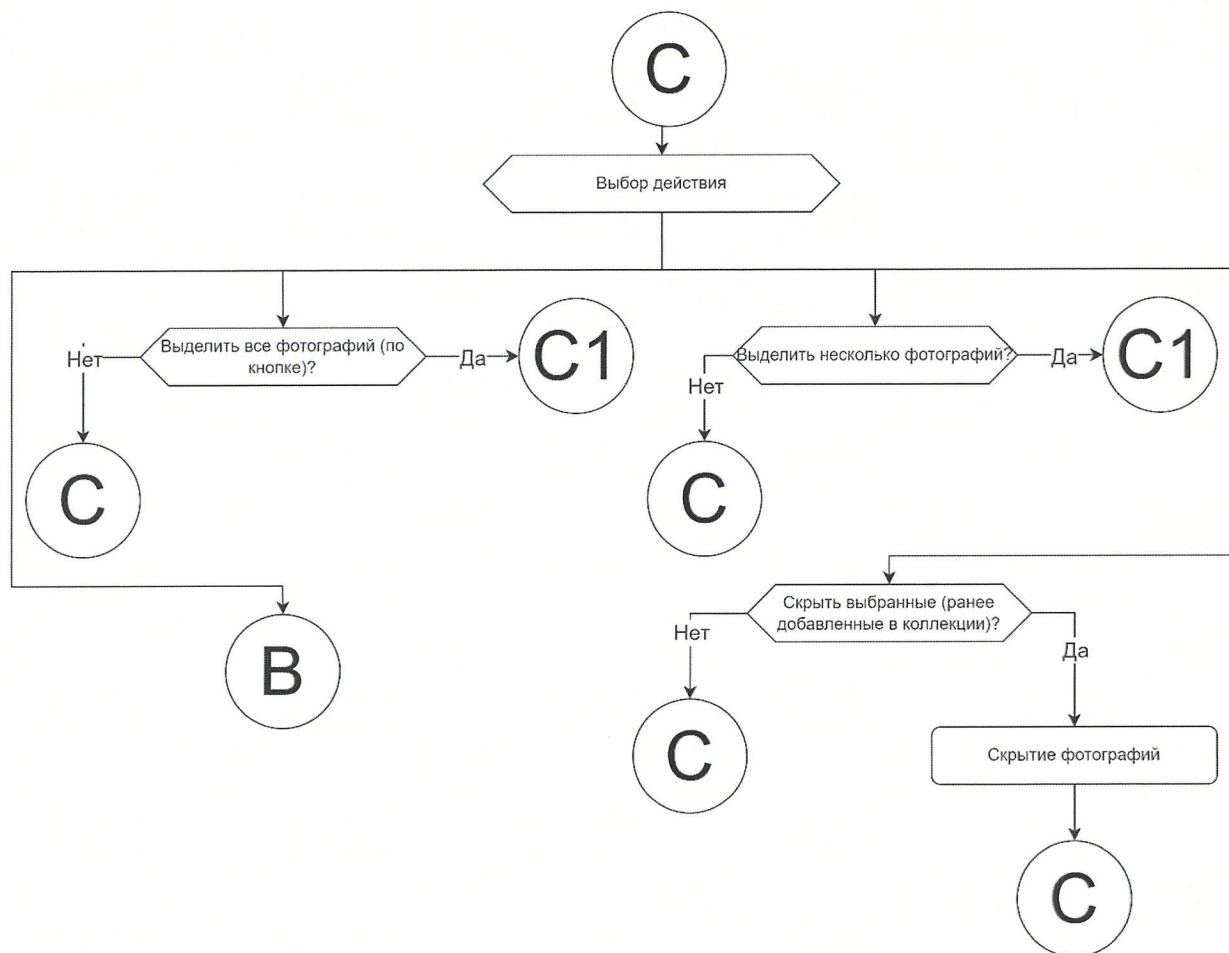


Рисунок 20 — Модуль работы с фотографиями в буфере

На этом этапе пользователь выполняет действия с фотографиями, загруженными в буфер. Пользователь может выбрать все фотографии сразу, нажав соответствующую кнопку, или выделить несколько снимков по своему усмотрению. После этого можно выполнить действия с выбранными фотографиями: добавить их в коллекцию, удалить или скрыть ранее добавленные фотографии, которые помечены как уже использованные в коллекции. Если никаких действий не предпринимается, пользователь остаётся в текущем состоянии и может продолжить работу с буфером или вернуться в рабочее пространство.

Подмодуль работы с фотографиями в буфере (рисунок 21):

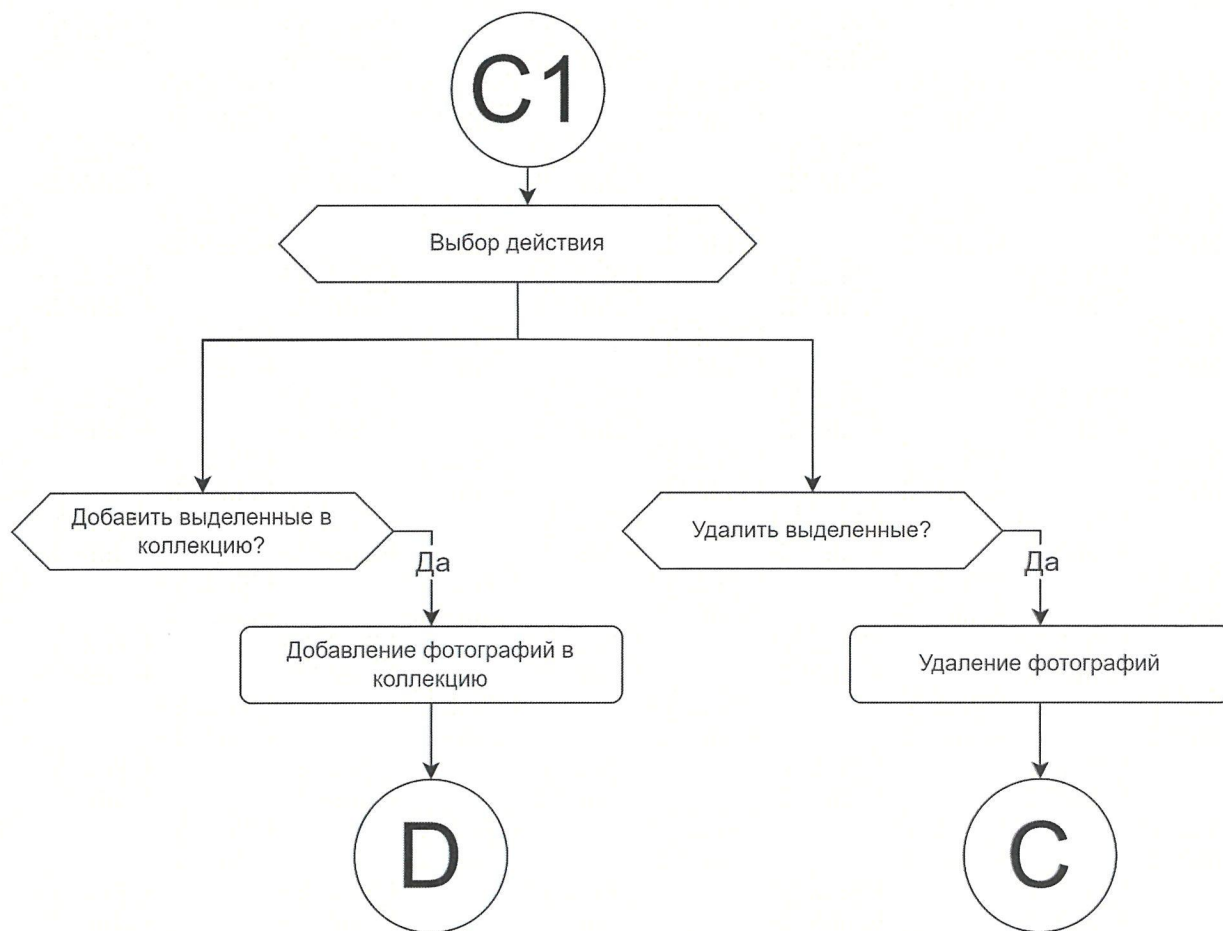


Рисунок 21 — Подмодуль работы с фотографиями в буфере

На этом этапе пользователь принимает решение, что делать с выбранными фотографиями. Он может либо добавить их в коллекцию, либо удалить из буфера. Если пользователь выбирает добавление, фотографии переносятся в выбранную коллекцию, и система продолжает обработку. Если выбрано удаление, фотографии удаляются из буфера, и пользователь может продолжить работу с оставшимися изображениями. После выполнения действий система возвращается к следующему этапу работы.

Модуль работы с фотографиями в коллекции (рисунок 22):

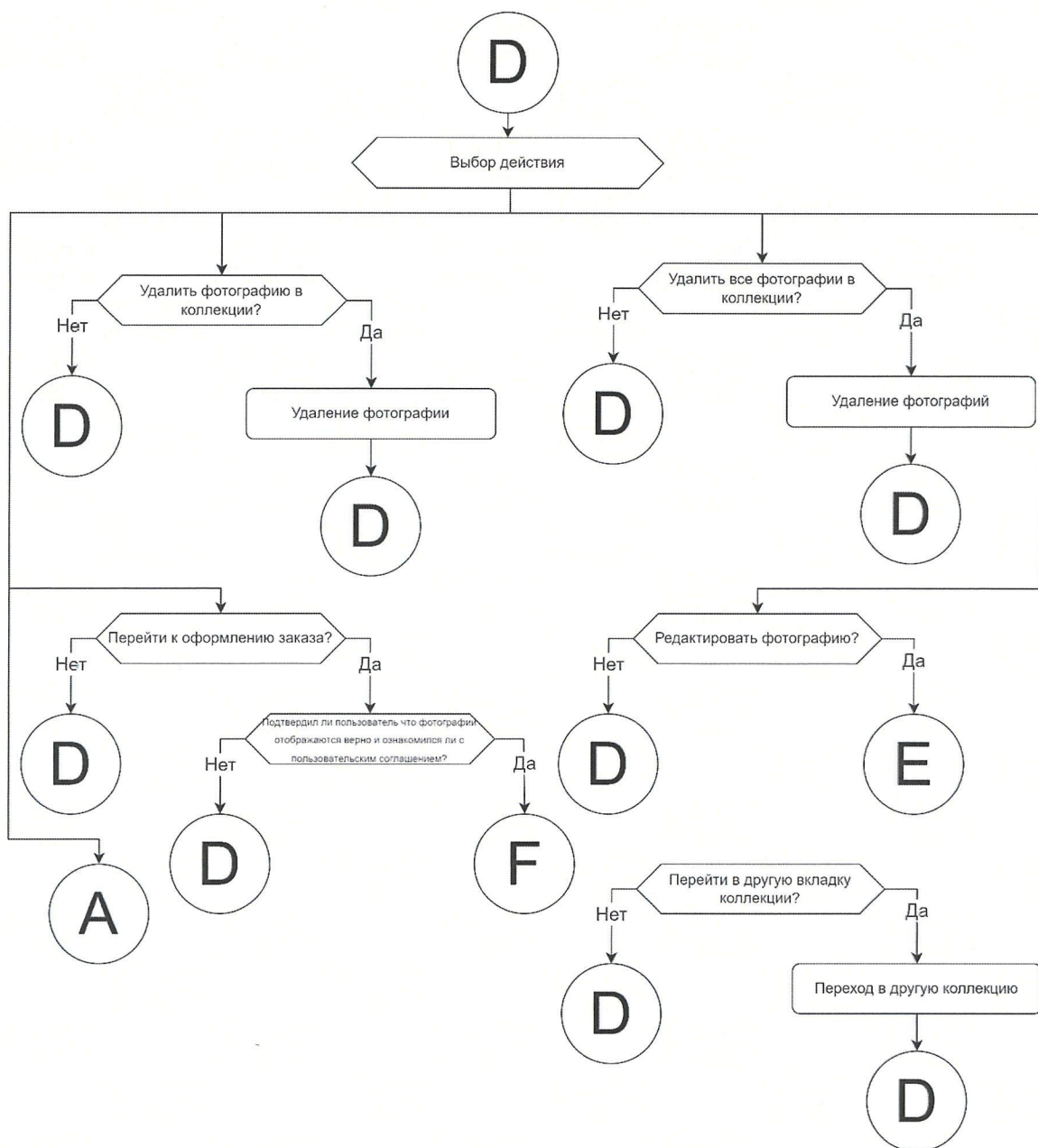


Рисунок 22 — Модуль работы с фотографиями в коллекции

На этом этапе пользователь взаимодействует с фотографиями в коллекции. Он может удалить одну или все фотографии из коллекции, если они больше не нужны. Также доступна возможность перейти в редактирование фотографий. Пользователь может переключаться между вкладками разных коллекций для управления другими фотографиями.

Если пользователь готов к оформлению заказа, он переходит к соответствующему этапу. Перед этим система запрашивает подтверждение, что все фотографии отображаются корректно и пользователь ознакомлен с пользовательским соглашением. При положительном ответе начинается процесс оформления заказа, при отрицательном — пользователь возвращается к редактированию коллекции.

Модуль редактора с фотографиями (рисунок 23):

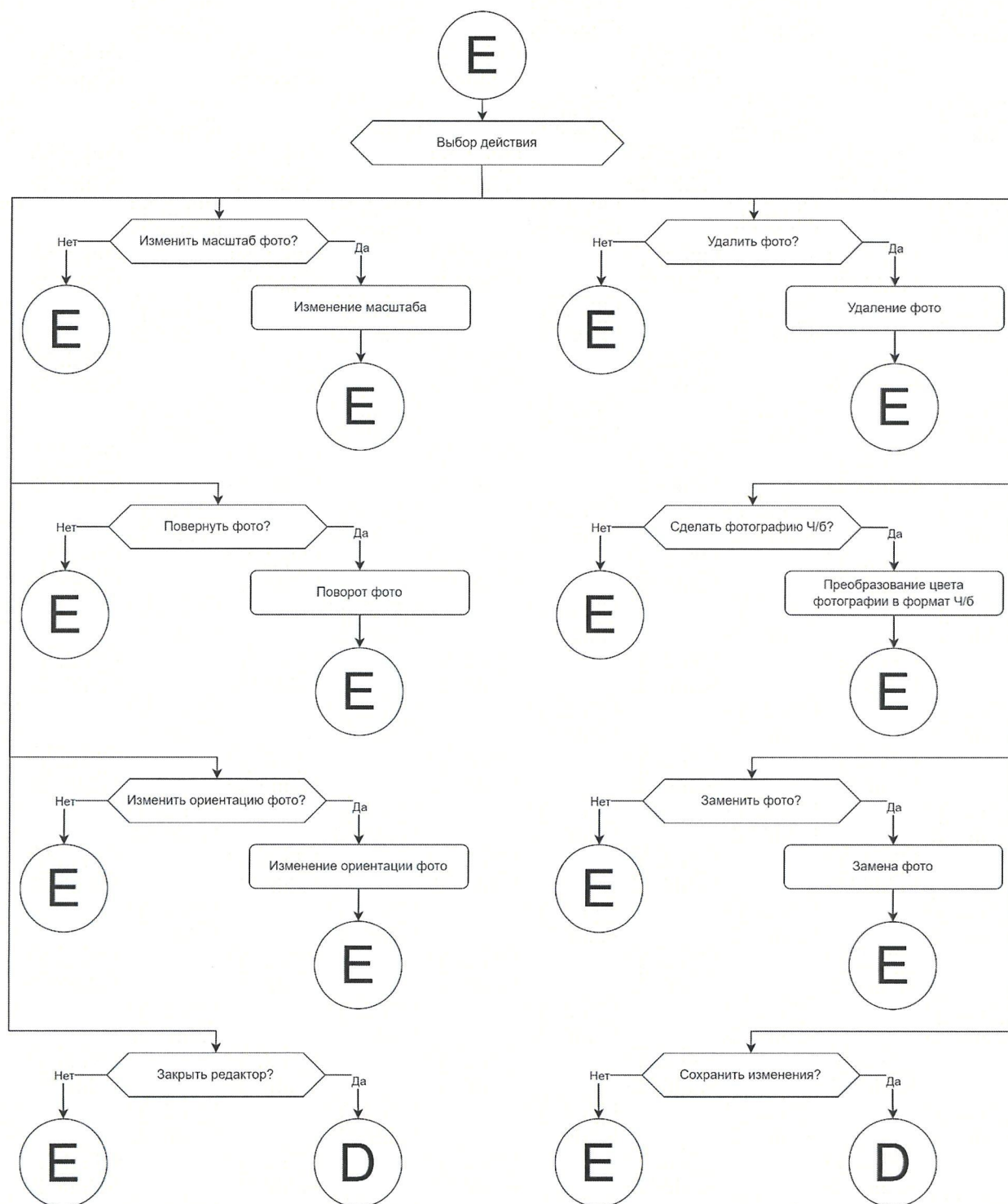


Рисунок 23 — Модуль редактора с фотографиями

На данном этапе пользователь работает с редактором фотографий. В редакторе доступны различные действия, такие как изменение масштаба изображения, его ориентации, поворот, преобразование в чёрно-белый формат, замена фото или удаление. После

выполнения нужных изменений пользователь может сохранить результаты или выйти из редактора без сохранения. Если пользователь выбирает выход, изменения либо фиксируются, либо отменяются в зависимости от выбранного варианта, после чего происходит возврат к предыдущему этапу работы с коллекцией.

Модуль оформления заказа (рисунок 24):

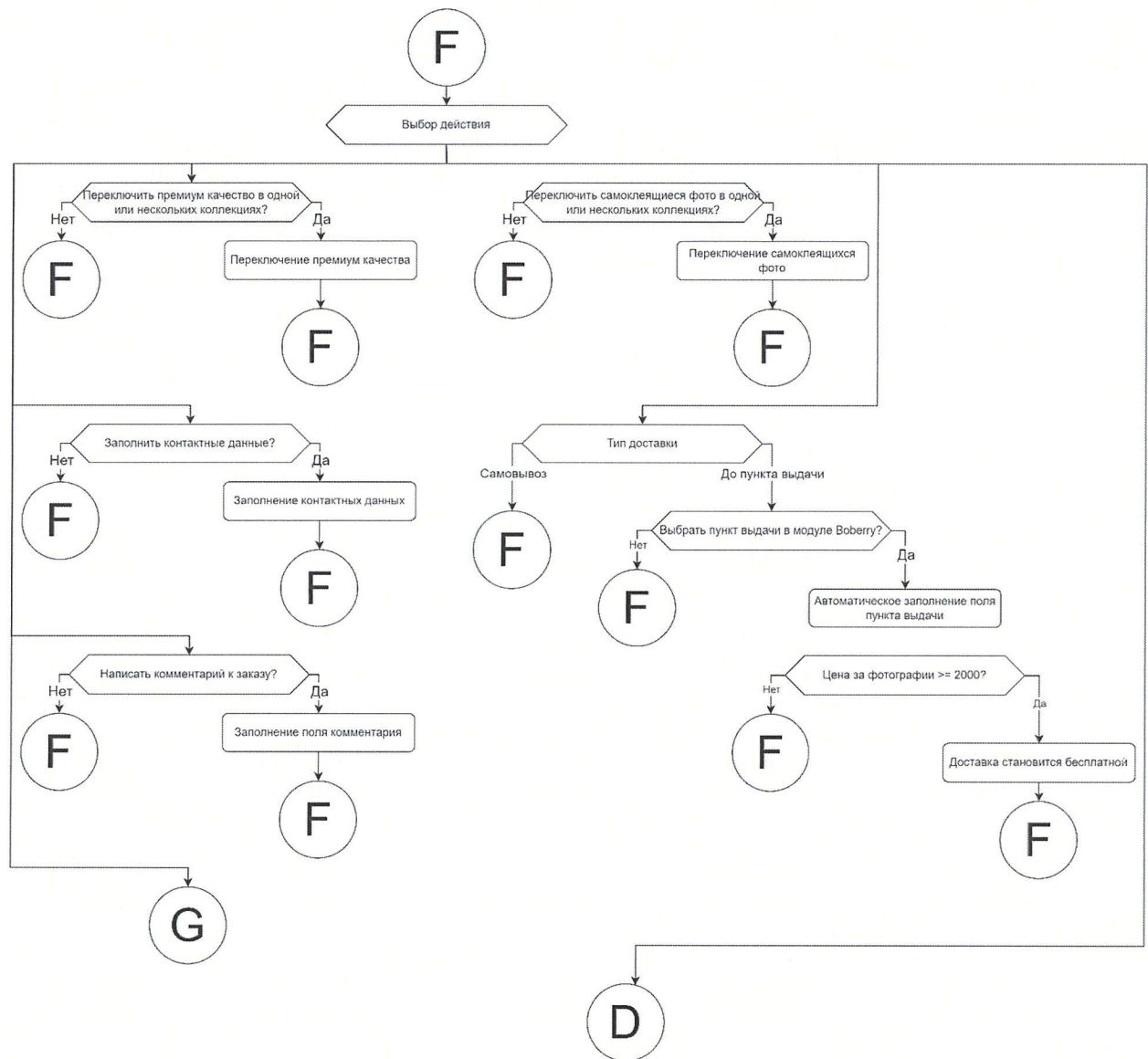


Рисунок 24 — Модуль оформления заказа

На данном этапе пользователь переходит к настройкам параметров заказа. Он может выбрать дополнительные параметры, такие как переключение на премиум-качество или использование самоклеющихся фотографий для одной или нескольких коллекций. После этого пользователь вводит контактные данные, выбирает тип доставки (самовывоз или до пункта выдачи). В случае доставки до пункта выдачи система предлагает выбрать ПВЗ через модуль Voberry, автоматически заполняя поле адреса.

Если сумма заказа превышает 2000 рублей, доставка становится бесплатной. Также пользователь может добавить комментарий к заказу. После выполнения всех действий пользователь готов к следующему этапу — оформлению и оплате заказа.

Модуль проверки данных и оплаты (рисунок 25):

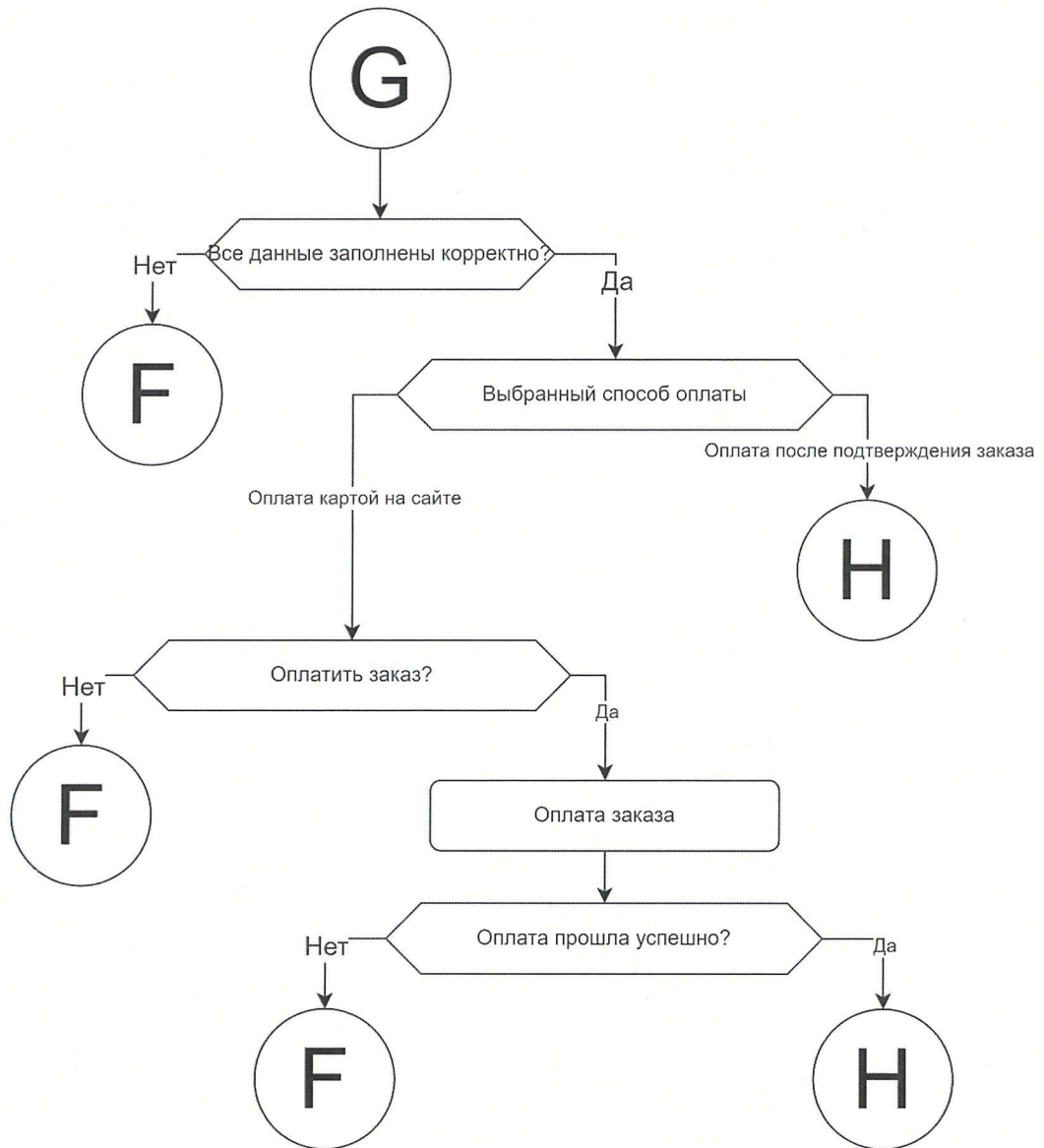


Рисунок 25 — Модуль проверки данных и оплаты

На данном этапе пользователь проверяет корректность введенных данных. Если данные заполнены неверно, пользователь возвращается к их редактированию. Если всё правильно, он выбирает способ оплаты: оплата картой на сайте или оплата после подтверждения заказа.

При выборе оплаты картой открывается модуль платёжной системы Юkassa и пользователь переходит к процессу оплаты. Если он решает не оплачивать заказ, возвращается

на предыдущий этап. Если оплата начинается, система проверяет её успешность. В случае успешного проведения платежа пользователь переходит к следующему этапу обработки заказа, а в случае ошибки ему предоставляется возможность повторить попытку.

Модуль обработки заказа на сервере (рисунок 26):

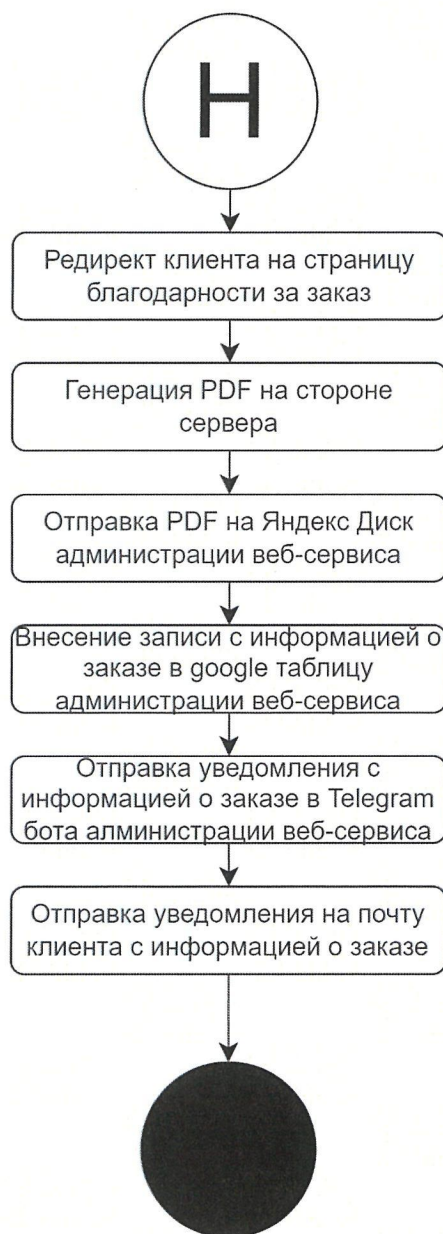


Рисунок 26 — Модуль обработки заказа на сервере

На финальном этапе система завершает обработку заказа. Пользователь перенаправляется на страницу благодарности, подтверждающую успешное оформление. Одновременно на сервере выполняются автоматические действия: генерируется PDF-файл с фотографиями заказа, который сохраняется на Яндекс.Диск администрации сервиса. Информация о заказе добавляется в Google Таблицу для учёта. Администратор получает

уведомление в Telegram с деталями заказа, а клиенту отправляется подтверждение на email, содержащее всю необходимую информацию о заказе. Этот этап завершает процесс взаимодействия пользователя с сервисом.

Описание алгоритма работы веб-сервиса позволяет увидеть взаимосвязь между действиями пользователя и внутренними процессами системы, что способствует точному проектированию и дальнейшему развитию функционала.

4.4 User Flow

User Flow — это карта пользовательских взаимодействий, которая демонстрирует возможные пути навигации по веб-сервису foto4ki.ru. Данный инструмент используется для визуализации структуры сайта и позволяет определить, как пользователь переходит между различными страницами и функциональными модулями сервиса.

Рассмотрим user flow веб-сервиса по фотопечати на рисунке 27.

Данный User Flow отображает структуру взаимодействия пользователя с веб-сервисом foto4ki.ru. Основные страницы (вертикальные прямоугольники) включают начальную страницу (рабочее пространство), вкладки коллекций, редактор фотографий, корзину и страницу благодарности за заказ.

Пользователь начинает с начальной страницы, где ему доступны информационные модальные окна (горизонтальные прямоугольники в виде диалогового окна), такие как «Цены и доставка», «Подсказки» и «Контакты». Далее он может добавить коллекции, выбрав их формат, а затем перейти к загрузке фотографий из устройства или Яндекс.Диска через буфер загрузки.

После загрузки пользователь работает с вкладками коллекций, где может редактировать фотографии, используя инструменты редактора. После завершения работы с фотографиями пользователь переходит в корзину, где оформляет заказ, выбирая способ доставки и оплаты (например, модуль Voksberry или ЮKassa).

Финальный этап — перенаправление на страницу благодарности за заказ, что завершает пользовательский путь.

Такой подход обеспечивает логичную и удобную навигацию по всем ключевым функциям сервиса.

4.5 Макеты страниц веб-сервиса

Макеты (mockup) представляют собой визуализацию интерфейса веб-сервиса foto4ki.ru, демонстрируя внешний вид страниц, расположение элементов и основные взаимодействия пользователя с системой. Эти графические прототипы создаются на этапе проектирования и служат ключевым инструментом для понимания структуры и функциональности сервиса.

Цель макетов — обеспечить ясное представление о пользовательском интерфейсе, что упрощает согласование дизайна, выявление возможных проблем на этапе проектирования и дальнейшую реализацию интерфейса. Макеты позволяют увидеть расположение таких элементов, как кнопки, модальные окна, формы, и определить их взаимосвязь на разных этапах пользовательского пути.

Ниже представлена часть фреймов макетов:

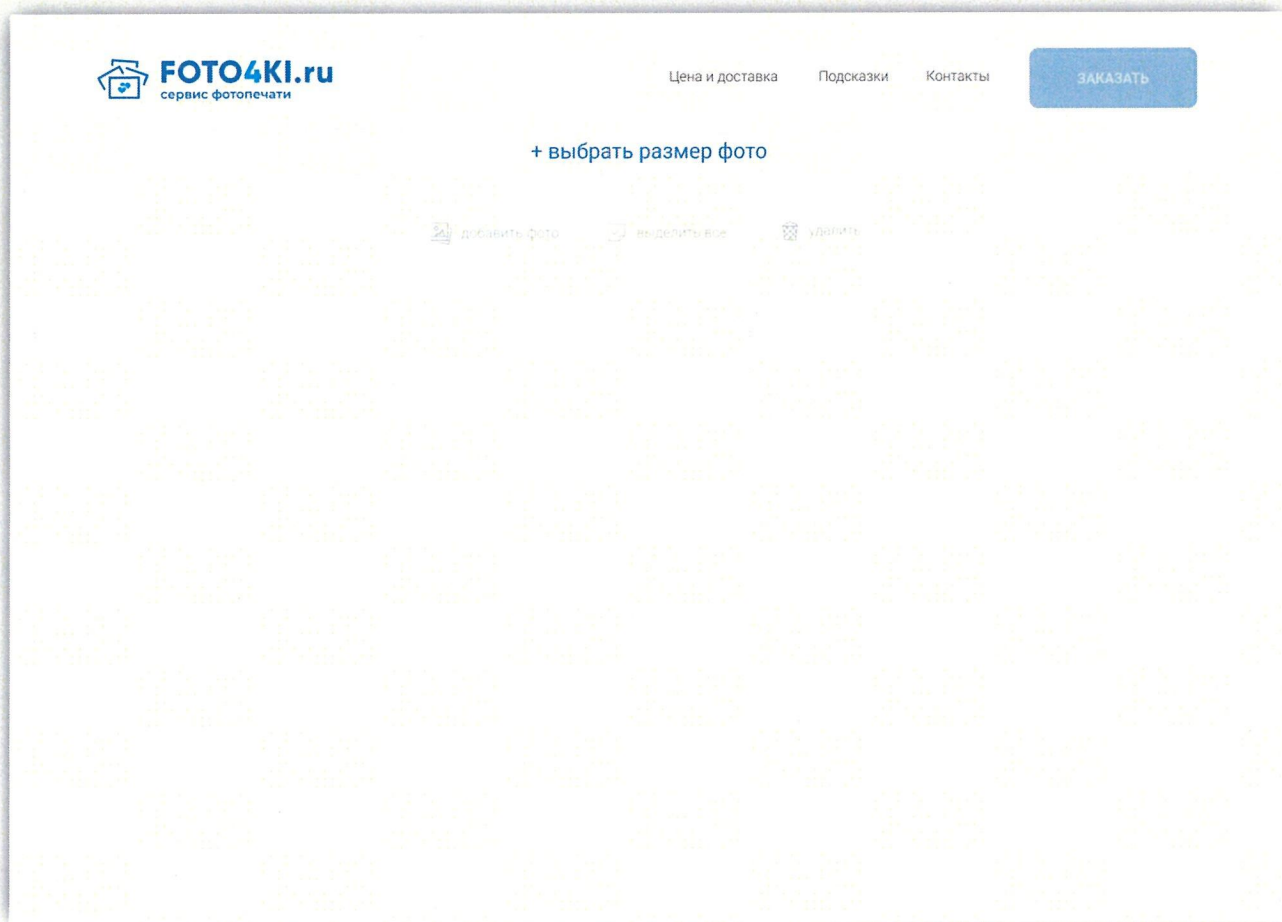


Рисунок 28 — Главная страница веб-сервиса

Макет главной страницы веб-сервиса foto4ki.ru представляет рабочее пространство пользователя. В верхней части размещены логотип, навигационная панель (разделы «Цена и доставка», «Подсказки», «Контакты») и кнопка «Заказать». Центральный элемент — кнопка «+ выбрать размер фото» для добавления коллекций. Под ней расположены инструменты управления фотографиями: «Добавить фото», «Выделить всё», «Удалить». Интерфейс минималистичен и интуитивен, с акцентом на основные действия.

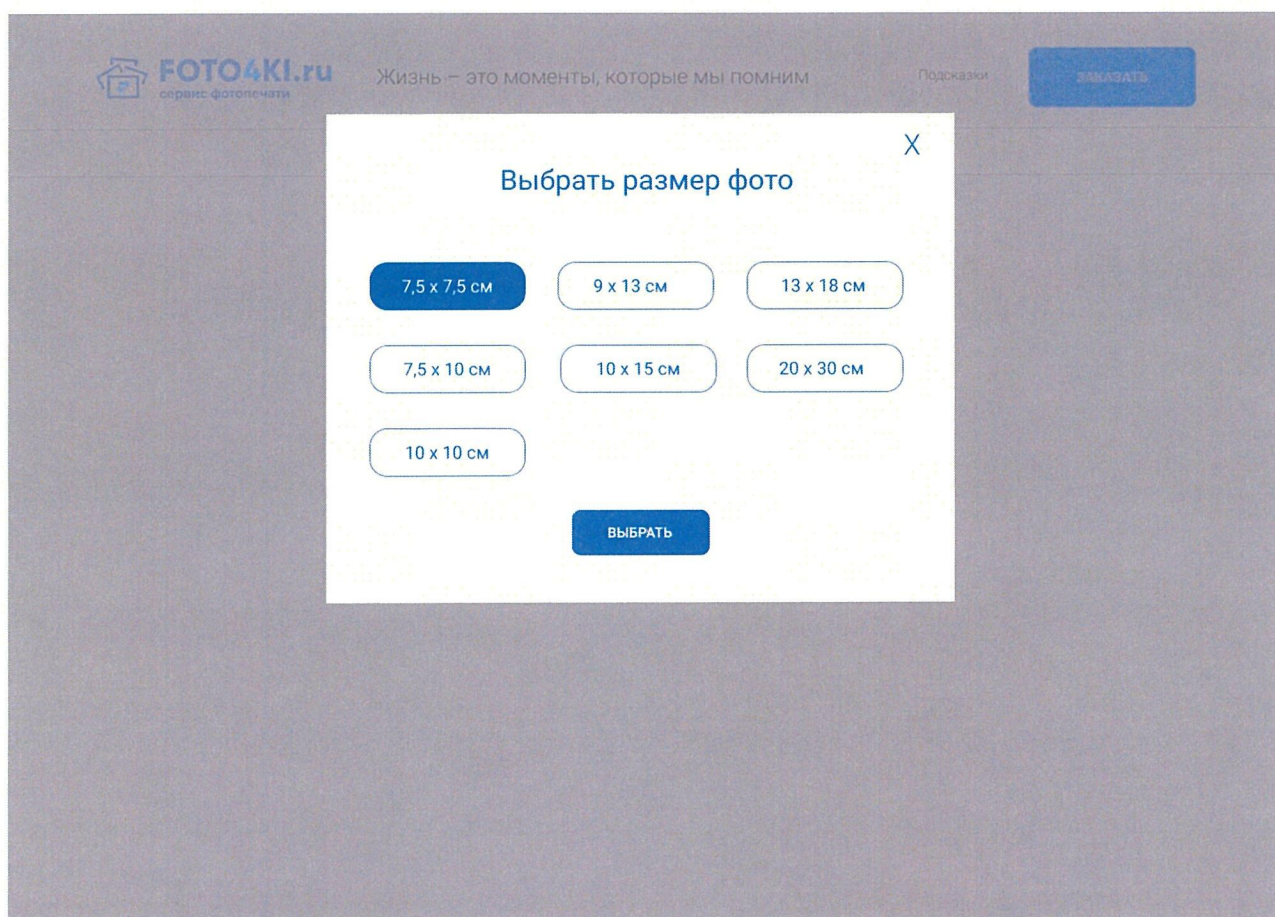


Рисунок 29 — Попап выбора размера фотографий

Этот макет представляет окно выбора размера фотографий. Пользователь может выбрать один из доступных форматов (например, 7,5×7,5 см, 10×15 см, 20×30 см) и подтвердить выбор, нажав кнопку «Выбрать».

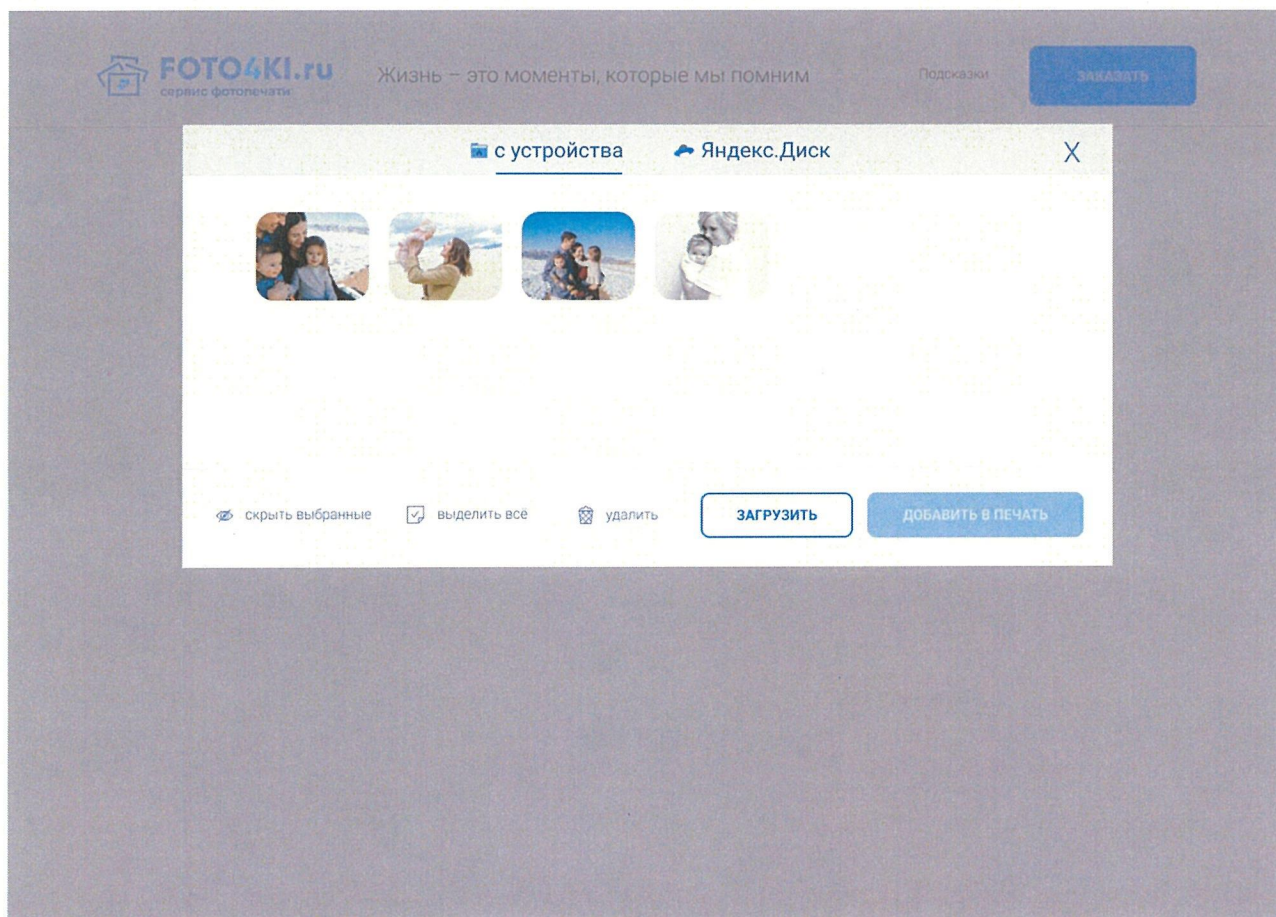


Рисунок 30 — Буфер загрузки фотографий

Этот макет отображает буфер загрузки фотографий с возможностью выбора источника (с устройства или Яндекс.Диска). Загруженные фотографии представлены в виде миниатюр. В нижней части доступны инструменты управления: скрывание, выделение всех, удаление и кнопки действий. Можно заметить, что кнопка «Добавить в печать» неактивна, так как ни одно фото не выбрано для добавления.

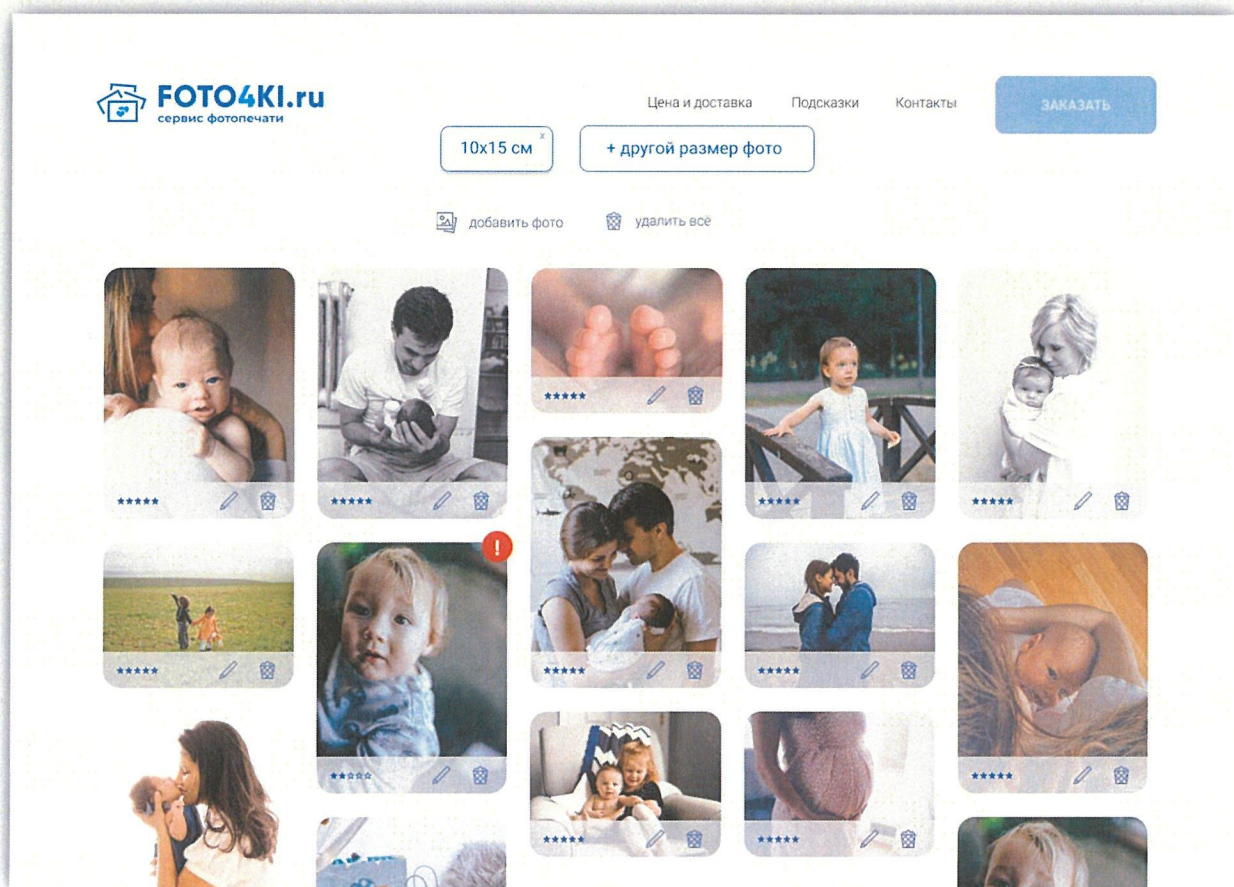


Рисунок 31 — Рабочая область во вкладке коллекции

Этот макет отображает рабочее пространство с добавленными фотографиями в коллекции формата 10×15 см. Каждая фотография сопровождается рейтингом качества (звёзды) и инструментами для редактирования или удаления. Красная метка с восклицательным знаком у одной из фотографий указывает на её низкое качество. В верхней части доступны кнопки для добавления нового размера фото, загрузки дополнительных изображений и удаления всех фотографий. Интерфейс предназначен для удобного управления коллекцией.



Рисунок 32 — Редактор фотографий

Редактор фотографий предоставляет пользователю инструменты для работы с изображением: поворот, изменение ориентации, применение чёрно-белого фильтра, замена фото и удаление. В верхней части отображается оценка качества фото («отличное качество»). Кнопка «Готово» позволяет сохранить изменения и выйти из редактора.


FOTO4KI.ru
сервис фотопечати

[Цена и доставка](#)
[Подсказки](#)
[Контакты](#)

ЗАКАЗАТЬ

[Вернуться к фотографиям](#)

Оформление заказа

Вы можете изменить тип фотобумаги с обычной на самоклеющуюся. И качество со стандартного на премиум.

	Самоклеящиеся фото	Премиум качество	Сумма, Р
1. Фото 10x15, 10 шт.	☒	☐	150 Р
2. Фото 7,5x10, 7 шт.	☐	☐	70 Р
3. Фото 7,5x7,5, 5 шт.	☐	☐	50 Р

Имя и фамилия получателя

Телефон

e-mail

Город

☒ До пункта выдачи от 1 раб. дня
☐ Самовывоз – г. Красногорск, ул. Спасская, 1а

Пункт выдачи

Комментарий к заказу

Оплата
☒ Оплата картой на сайте
☐ Оплата после подтверждения заказа

Сумма заказа: 890 Р
Доставка: 210 Р
Итого к оплате: 2 100 Р

ОФОРМИТЬ ЗАКАЗ

нажимая на кнопку, вы даете согласие на обработку персональных данных и соглашаетесь с политикой безопасности

Рисунок 33 — Страница оформления заказа (корзина)

Этот макет представляет форму оформления заказа, где пользователь указывает параметры печати, контактные данные и способ доставки. В левой части отображён список выбранных фотографий с настройками типа бумаги (самоклеящиеся или премиум качество) и расчётом стоимости. В правой части пользователь вводит имя, телефон, email, город, выбирает способ доставки (до пункта выдачи или самовывоз) и способ оплаты. Итоговая сумма заказа отображается внизу, а кнопка «Оформить заказ» становится активной после заполнения всех обязательных полей.

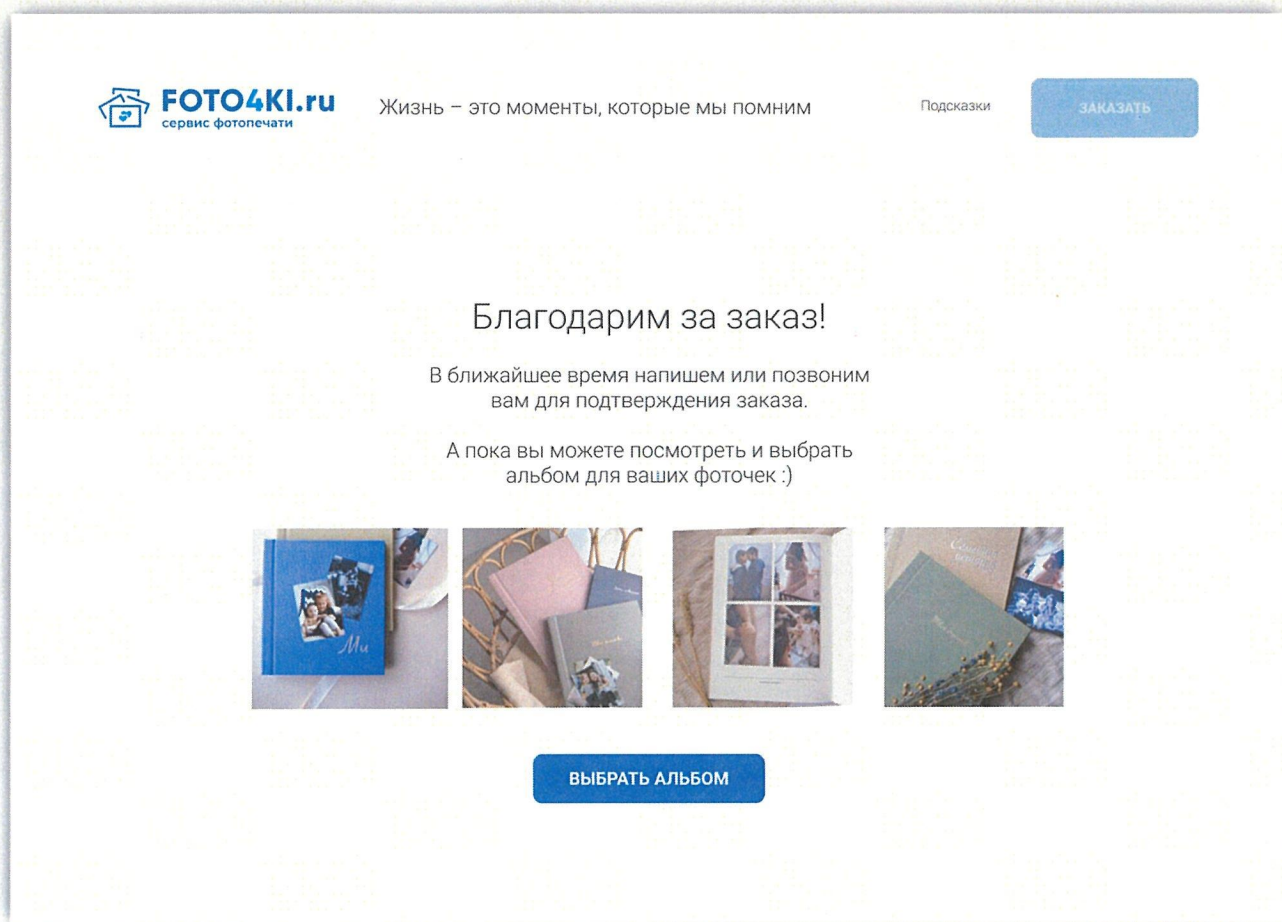


Рисунок 34 — Страница благодарности

Этот макет представляет страницу благодарности за заказ, завершающую пользовательский путь. На странице размещено сообщение с подтверждением оформления заказа и информацией о том, что заказ будет подтверждён через связь с клиентом. Также пользователю предлагается посмотреть и выбрать фотоальбомы для своих фотографий, с представленными примерами продукции.

Заключение

В ходе выполнения курсовой работы был спроектирован веб-сервис foto4ki.ru для удобной и быстрой печати фотографий. Были рассмотрены все ключевые аспекты проектирования и реализации: от анализа требований и разработки структуры базы данных до построения интерфейсов и алгоритмов работы системы.

Сервис охватывает все этапы взаимодействия пользователя, включая выбор форматов, загрузку и редактирование фотографий, оформление заказа и его оплату. Особое внимание уделено интуитивности интерфейса, автоматизации обработки заказов и интеграции с внешними системами, такими как Яндекс.Диск, ЮKassa и Voberry. Это позволяет достичь высокого уровня удобства как для клиентов, так и для администрации сервиса.

Разработанные макеты и пользовательские сценарии (User Flow) демонстрируют, что сервис ориентирован на простоту и эффективность работы. Алгоритмы, архитектура базы данных и диаграммы классов обеспечивают надёжность и масштабируемость системы.

Таким образом, выполненная работа охватывает полный цикл проектирования веб-сервиса, отражая его функциональность и пользовательский опыт. Это создаёт основу для дальнейшей реализации сервиса и его адаптации под реальные условия использования. foto4ki.ru имеет потенциал для успешного внедрения и удовлетворения потребностей целевой аудитории в печати фотографий.

Список использованных источников

- 1 WordPress. Официальный сайт. URL: <https://wordpress.com/ru> (дата обращения: 18.11.2024);
- 2 JavaScript. Интерактивный учебник по JavaScript. URL: <https://learn.javascript.ru/> (дата обращения: 5.12.2024);
- 3 jQuery. Официальный сайт. URL: <https://jquery.com/> (дата обращения: 5.12.2024);
- 4 PHP. Официальный сайт. URL: <https://www.php.net/> (дата обращения: 6.12.2024);
- 5 Gulp.js. Официальный сайт. URL: <https://gulpjs.com/> (дата обращения: 6.12.2024);
- 6 Скиллфэктори. Глоссарий: Минификация. URL: <https://blog.skillfactory.ru/glossary/minifikacziya/> (дата обращения: 6.12.2024);
- 7 Boxberry. Официальный сайт [Электронный ресурс]. URL: <https://boxberry.ru/> (дата обращения: 8.12.2024);
- 8 ЮKassa. Официальный сайт [Электронный ресурс]. URL: <https://yookassa.ru/> (дата обращения: 8.12.2024);
- 9 Яндекс.Диск. Документация для разработчиков [Электронный ресурс]. URL: <https://yandex.ru/dev/disk/> (дата обращения: 8.12.2024);
- 10 Telegram. Приложения Telegram [Электронный ресурс]. URL: <https://telegram.org/apps> (дата обращения: 8.12.2024);
- 11 Что такое ER-диаграмма [Электронный ресурс]. URL: <https://practicum.yandex.ru/blog/chto-takoe-er-diagramma/> (дата обращения: 16.12.2024);
- 12 Руководство по UML-диаграммам классов. Visual Paradigm [Электронный ресурс]. URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/> (дата обращения: 25.12.2024);
- 13 Что такое user flow и зачем его разрабатывать [Электронный ресурс]. URL: <https://practicum.yandex.ru/blog/chto-takoe-user-flow-i-zachem-ego-razrabatyvat/> (дата обращения: 3.01.2025);
- 14 Что такое мокап [Электронный ресурс]. URL: <https://practicum.yandex.ru/blog/chto-takoe-mokap/> (дата обращения: 8.01.2025);
- 15 Figma. Файлы и команды [Электронный ресурс]. URL: <https://www.figma.com/files/team/1287644311037700628/recents-and-sharing?fuid=1169120362715942086> (дата обращения: 8.01.2025);