**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 3](#_Toc193975876)

[ГЛАВА 1. ПРЕДМЕТНАЯ ОБЛАСТЬ 4](#_Toc193975877)

[1.1. Исходная постановка задачи 4](#_Toc193975878)

[1.2. Среда разработки 5](#_Toc193975879)

[1.3. Техническое задание 6](#_Toc193975880)

[1.4. Руководство по стилю 12](#_Toc193975881)

[1.5. Диаграммы UML 14](#_Toc193975882)

[1.6. Wireframe 18](#_Toc193975883)

[ГЛАВА 2. РАЗРАБОТКА БАЗЫ ДАННЫХ 20](#_Toc193975884)

[2.1. Разработка логической модели данных 20](#_Toc193975885)

[2.2. Добавление таблиц к базе данных 20](#_Toc193975886)

[2.3. Просмотр связей на диаграмме базы данных 21](#_Toc193975887)

[2.4. Определение ограничений, налагаемы на данные 21](#_Toc193975888)

[2.5. Определение объемы и типов данных 24](#_Toc193975889)

[2.6. Определение способа использования данных 25](#_Toc193975890)

[2.7. Разработка физической структуры данных 26](#_Toc193975891)

[2.7.1. Создание таблиц в БД (скрипт) 26](#_Toc193975892)

[2.7.2. Данные (таблицы в режиме первые 200 строк) 27](#_Toc193975893)

[ГЛАВА 3. РАЗРАБОТКА ПРИЛОЖЕНИЯ 32](#_Toc193975894)

[3.1. Разработка пользовательского интерфейса 32](#_Toc193975895)

[3.2. Листинг программы с комментариями 36](#_Toc193975896)

[3.3. Руководство пользователя 51](#_Toc193975897)

[ГЛАВА 4. РАЗРАБОТКА 1С 56](#_Toc193975898)

[4.1 Разработка подсистем 56](#_Toc193975899)

[4.2. Разработка справочников 57](#_Toc193975900)

[4.3. Разработка документов и регистров 57](#_Toc193975901)

[4.4. Разработка отчетов 57](#_Toc193975902)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 58](#_Toc193975903)

# ВВЕДЕНИЕ

Учебная практика на тему: УП 02 Разработка модулей программного обеспечения для компьютерных систем «АИС Производство баннеров»

Автоматизированная информационная система (АИС) «Производство баннеров» представляет собой ключевой инструмент для оптимизации процессов в сфере рекламных услуг. Внедрение таких систем позволяет значительно улучшить организацию учета и обработки информации о заказах, клиентских предпочтениях и производственных процессах. АИС способствует упрощению и автоматизации этих процессов, что особенно актуально в условиях растущей конкуренции на рынке рекламных услуг и необходимости соответствия современным требованиям к качеству продукции.

**Цель работы:**

Разработать систему АИС «Производство баннеров»

Задачи:

1. Разработать схемы разработки
2. Разработать Wireframe
3. Разработать документацию
4. Разработать БД
5. Разработать предприятие 1С
6. Разработать приложение Windows Forms
7. Провести тестирование разработанных модулей

# ГЛАВА 1. ПРЕДМЕТНАЯ ОБЛАСТЬ

# 1.1. Исходная постановка задачи

В этом приложении хранится информация о заказах на баннеры, текущем статусе выполнения, материалах, используемых в производстве, а также информация о клиентах и рабочем персонале. Автоматизированная информационная система (АИС) «Производство баннеров» будет включать следующие таблицы для эффективного управления данными.

Таблица 1

Набор данных к задаче

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Поле | Тип | Размер | Описание |
| 1 | ID\_Order | Числовой | 10 | Уникальный идентификатор заказа |
| 2 | Client\_Surname | Текстовый | 50 | Фамилия клиента |
| 3 | Client\_Name | Текстовый | 50 | Имя клиента |
| 4 | Client\_Phones | Текстовый | 50 | Номер телефона клиента |
| 5 | Order\_Date | Дата |  | Дата оформления заказа |
| 6 | Banner\_Type | Текстовый | 50 | Тип баннера (PVC, ткань и т.д.) |
| 7 | Size | Текстовый | 10 | Размер баннера |
| 8 | Status | Текстовый | 20 | Текущий статус заказа (выполнен, в работе) |
| 9 | ID\_Product | Числовой | 10 | Идентификатор используемого продукта |
| 10 | ID\_Designer | Числовой | 10 | Идентификатор дизайнера, ответственного за проект |

# 1.2. Среда разработки

Visual Studio – среда разработки, позволяющая создавать консольные/веб/цифровые приложения для windows 10/11. Мой выбор остановился на ней, потому что мне удобен интерфейс, он понятен и прост, современный дизайн приложения, большой выбор функций, возможность разработать приложение в деталях. В ней доступны большое кол-во языков программирования (C++, C#, Visual Basic, CSS, html)

SQL Management Studio – среда разработки базы данных, позволяющая хранить информацию на сервере. На нем можно хранить данные в таблицах, диаграммы, представления, запросы и т.д. В таблицах можно указывать уникальные ключи и так же тип столбца (числовой, текстовый, дата). Можно создавать запросы, позволяющие фильтровать данные в таблицах. На диаграмме можно создать связи между таблицами.

1С предприятие - Среда разработки представляет собой платформу, на которой осуществляется создание и настройка программного обеспечения для автоматизации бизнес-процессов. Язык 1С, встроенный в эту среду, позволяет разрабатывать решения, адаптированные под специфические задачи пользователей, включая бухгалтерский учет, управление товарными запасами, кадровый учет и многое другое

# 1.3. Техническое задание

УТВЕРЖДАЮ УТВЕРЖДАЮ

Директор ГБПОУ «ВВК» Директор филиала

\_. \_ . \_ \_\_\_\_\_\_\_/Арсеньев А.В./ \_. \_. \_ \_\_\_\_\_\_/Михайлов А.С./

СОГЛАСОВАНО

Зам. директора ГБПОУ «ВВК»

\_ . \_ . \_ \_\_\_\_\_\_\_\_\_\_\_\_/ Бордодымова Т.В./

Электронная книга служебных контактов

Техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

02068120.508100.027-02 2А 01.М-ЛУ

Листов 4

Представители разработчика

Начальник отдела ИУ

\_ . \_ . \_ \_\_\_\_\_\_\_/Яковлева Ж.С./

Руководитель разработки

Ведущий программист

\_ . \_ . \_ \_\_\_\_\_\_\_/Бацев М. И./

2025

УТВЕРЖДЕНО

-022А 01.М-ЛУ к Договору № 2 от 02.04.23

АИС «Производство баннеров»

Техническое задание

02068120.508100.027-02 2А 01.М

Листов 4

2025

**1.3.1 Введение**

Наименование: Автоматизированная информационная система Производство баннеров, «АИС» Производство баннеров

Краткая характеристика: Система предоставляет возможность управления процессом производства баннеров, обеспечивая ввод, обработку и сохранение информации на сервере.

**1.3.2 Основание для разработки**

Документы: План разработки.

Организация: Производственное предприятие баннеров. Дата утверждения: 10.02.2025

**1.3.3 Назначение разработки**

Функциональное назначение: Обеспечение учета заказов, истории производственных процессов, назначения материалов, спецификаций, контактной информации клиентов (ФИО, телефон, электронная почта, детали заказа).

Эксплуатационное назначение: Поддержка работы производственного персонала и управление потоками заказов.

**1.4. Требования к программе или программному изделию**

**1.4.1 Требования к функциональным характеристикам**

1. Учёт заказов на баннеры.
2. Запись деталей заказа (размеры, материалы, цвета).
3. Ведение электронной базы данных клиентов и их заказов.
4. Генерация спецификаций на производство.

**1.4.2 Требования к надежности**

Надежное функционирование системы, защита от несанкционированного доступа и контроль данных.

Автоматическое резервирование данных каждые 12 часов.

Быстрое восстановление системы после сбоев (время восстановления не должно превышать 1 часа).

**1.4.3 Условия эксплуатации**

Оборудование должно функционировать в условиях офисного помещения с поддержкой скоростного интернета, а также электропитание (220В)

**1.4.4 Требования к составу и параметрам технических средств**

Сервера для хранения всей информации и поддержки функционала, рабочие станции и сетевая инфраструктура

1. Сервер: не менее 16 ГБ ОЗУ, 4 процессора, SSD объемом от 1 ТБ.
2. Рабочие станции: не менее 8 ГБ ОЗУ, процессор не ниже Intel Core i5.

**1.4.5 Требования к информационной и программной совместимости**

Совместимость с существующими базами данных (1С, Excel, SQL server) и серверами (Windows Server, Linux).

Поддержка стандартных форматов обмена данными (XML, CSV, Word, Visio).

**1.4.6 Требования к маркировке и упаковке**

Программа сохраняется на установочном носителе и в будущем устанавливается на системы.

**1.4.7 Требования к транспортированию и хранению**

USB-носители, облачные хранилища, жесткий диск.

**1.5 Требования к программной документации**

Пользовательская документация: руководство пользователя, инструкции по использованию.

Техническая документация: инструкции по установке и настройке системы.

**1.6 Технико-экономические показатели**

Оценка экономической эффективности: планируемая экономия времени на административные задачи.

Предполагаемая годовая потребность в программном обеспечении: 2000 клиентов в месяц.

**1.7 Стадии и этапы разработки**

Подготовительный этап: 01.02.2025 - 01.03.2025. (Подготовка документов, оборудования);

Этап проектирования: 02.03.2025 - 01.05.2025. (Наброски, wireframe);

Этап разработки: 02.05.2025 - 01.09.2025. (Начало разработки приложения);

Этап тестирования и внедрения: 02.09.2025 - 01.10.2025. (Окончание разработки, тестирования, проверки и упаковка продукта).

**1.8 Порядок контроля и приемки**

Виды испытаний: функциональное, нагрузочное тестирование.

Общие требования к приемке работы: соответствие системы заложенным требованиям и срокам.

**1.9 Приложения**

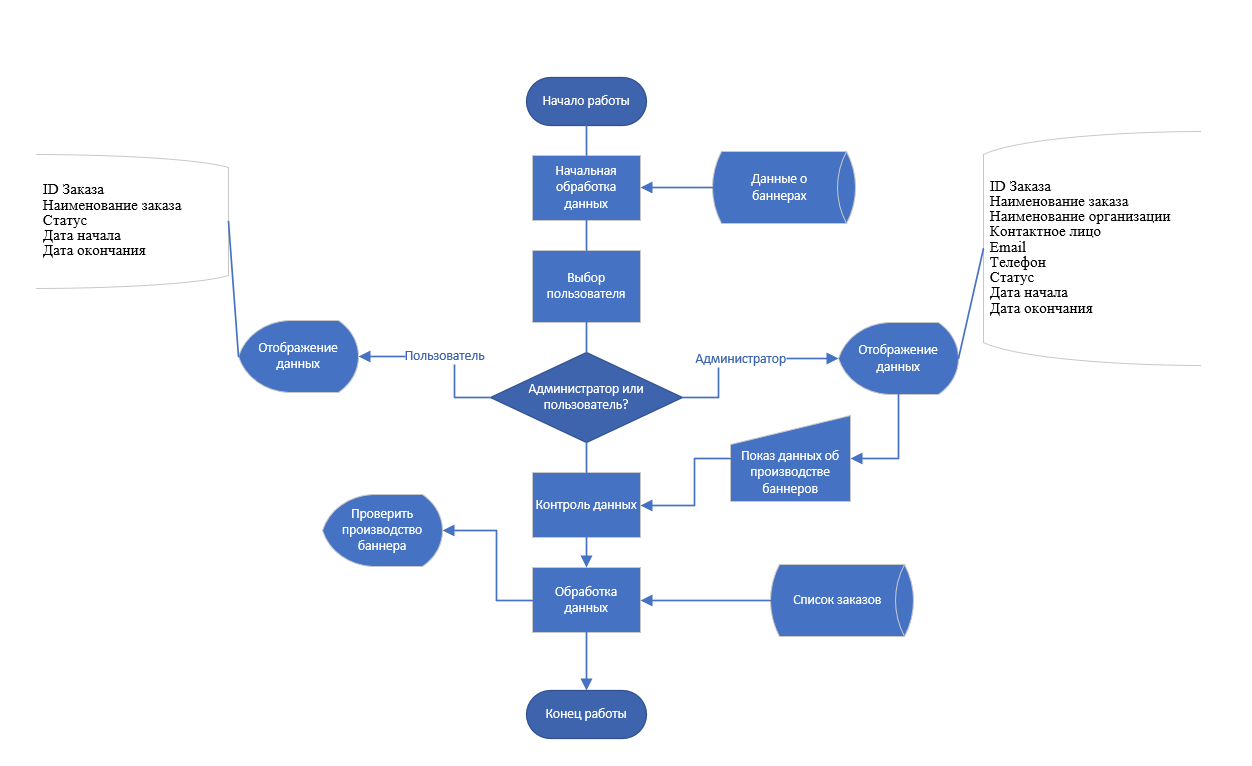


Рис. 1 Алгоритм работы приложения

# 1.4. Руководство по стилю

Все экранные формы пользовательского интерфейса должны иметь строгий и профессиональный стиль, соответствующий тематике производства баннеров. Каждая форма должна включать заголовок с логотипом организации и название приложения, что поможет идентифицировать систему и создает визуальное единство.

Все визуальные компоненты должны быть четко выровнены и удобны для восприятия. Элементы интерфейса должны иметь согласованный масштаб и не содержать излишних пустых пространств. Орфографические и грамматические ошибки недопустимы, так как это влияет на восприятие системы клиентами и пользователями.

Для надписей предпочтительно использовать черный цвет (RGB: 0, 0, 0). Шрифт должен быть простым и хорошо читаемым, с использованием обычного или полужирного начертаний. Размер шрифта должен составлять от 14 до 18 пунктов, чтобы гарантировать легкость чтения.

Все элементы интерфейса должны быть логически сгруппированы. Например, элементы, относящиеся к заказу, (идентификация клиента, дата заказа, статус и детали баннера) должны находиться в одной секции, чтобы облегчить взаимодействие пользователя с системой. Это позволит пользователям эффективно навигировать по интерфейсу и находить нужные функции без затруднений.

Внешний вид охватывает все визуальные элементы, с которыми взаимодействует конечный пользователь, включая экранные формы, отчеты, надписи и все текстовые элементы. Например, экранные формы для создания и редактирования заказов должны включать поля для ввода данных и выпадающие списки для выбора параметров баннера.

Все функции должны быть интуитивно понятны и доступны пользователям. Выполнение действий должно сопровождаться четкими и ясными инструкциями, чтобы обеспечить комфортную работу и минимизировать время на обучение новым пользователям. Пользовательский интерфейс должен включать подсказки и кнопки «Справка» для облегчения освоения системы.

**Применяемые цвета**

Основные

RGB: 46, 117, 182

RGB: 000, 000, 000

Для текста

RGB: 255, 255, 255

RGB: 000, 000, 000

# 1.5. Диаграммы UML

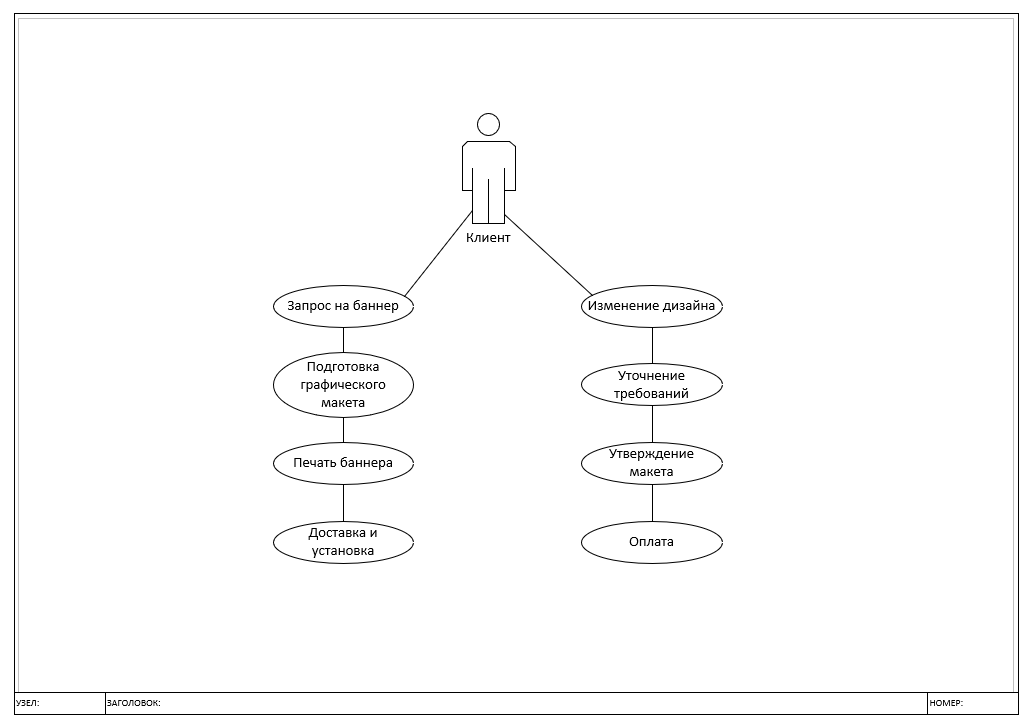
****

Рис. 2. Диаграмма вариантов использования

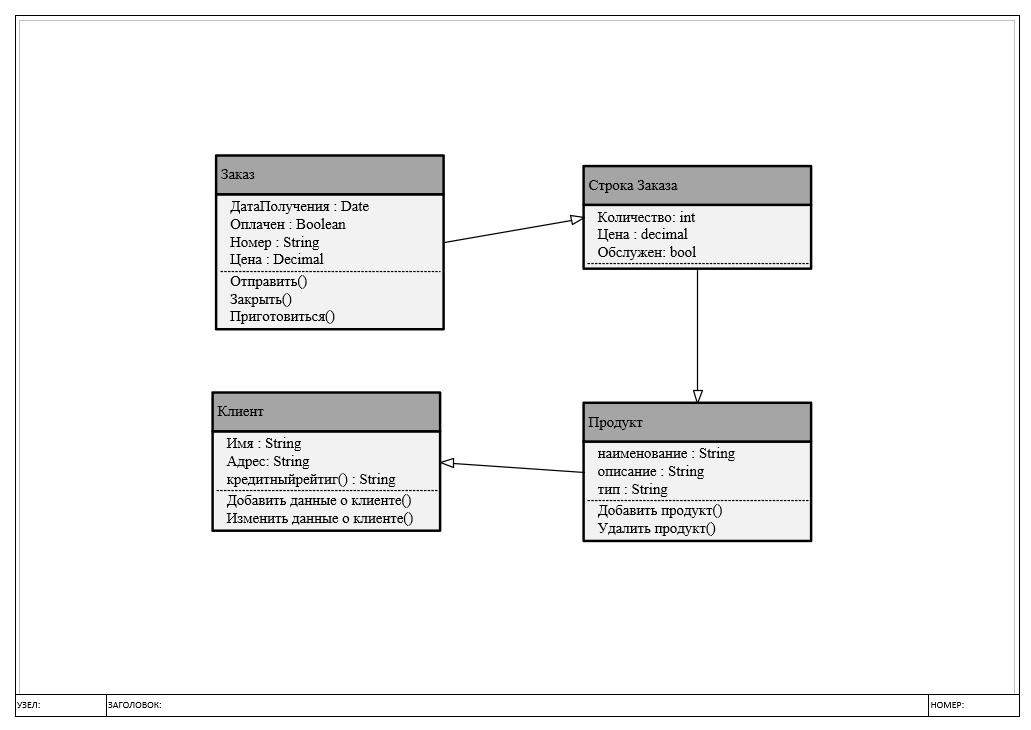


Рис. 3 Диаграмма классов



Рис. 4 Диаграмма состояний

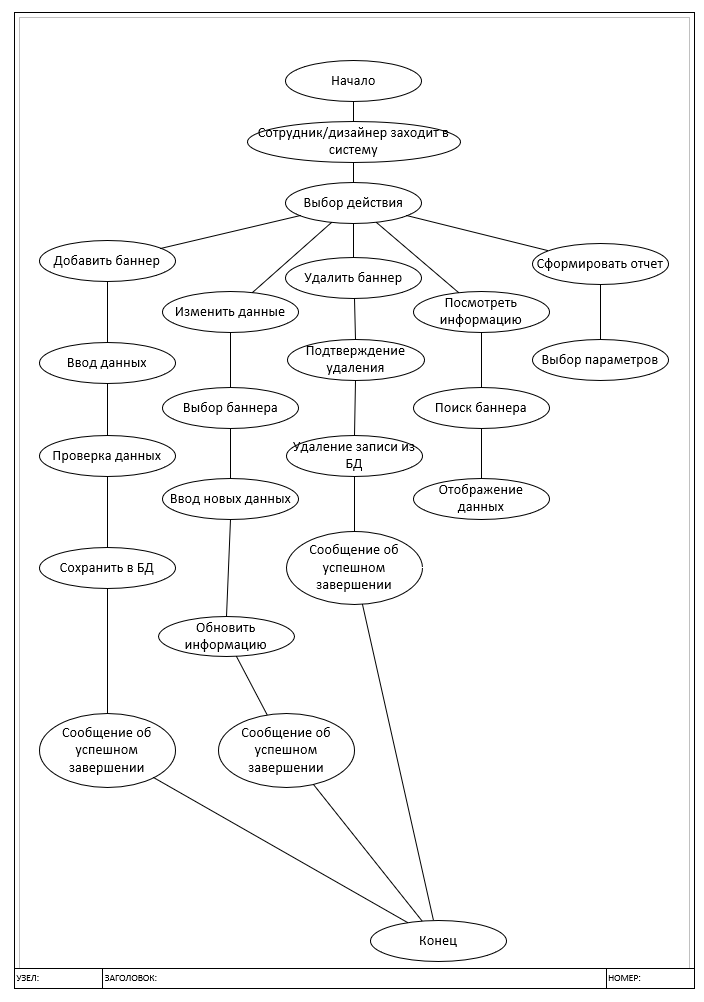


Рис. 5 Диаграмма деятельности

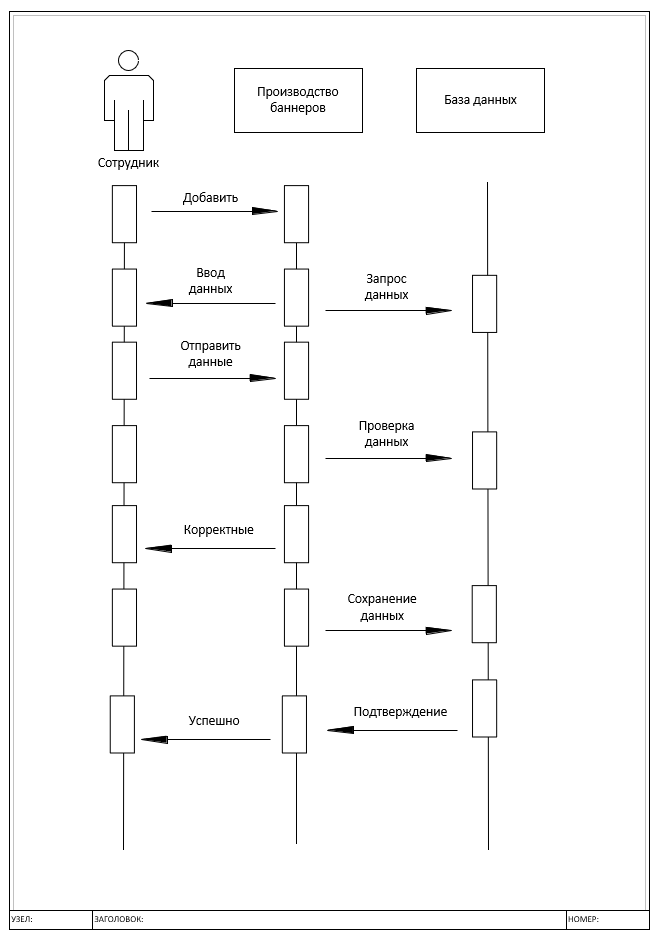


Рис. 6 Диаграмма взаимодействия

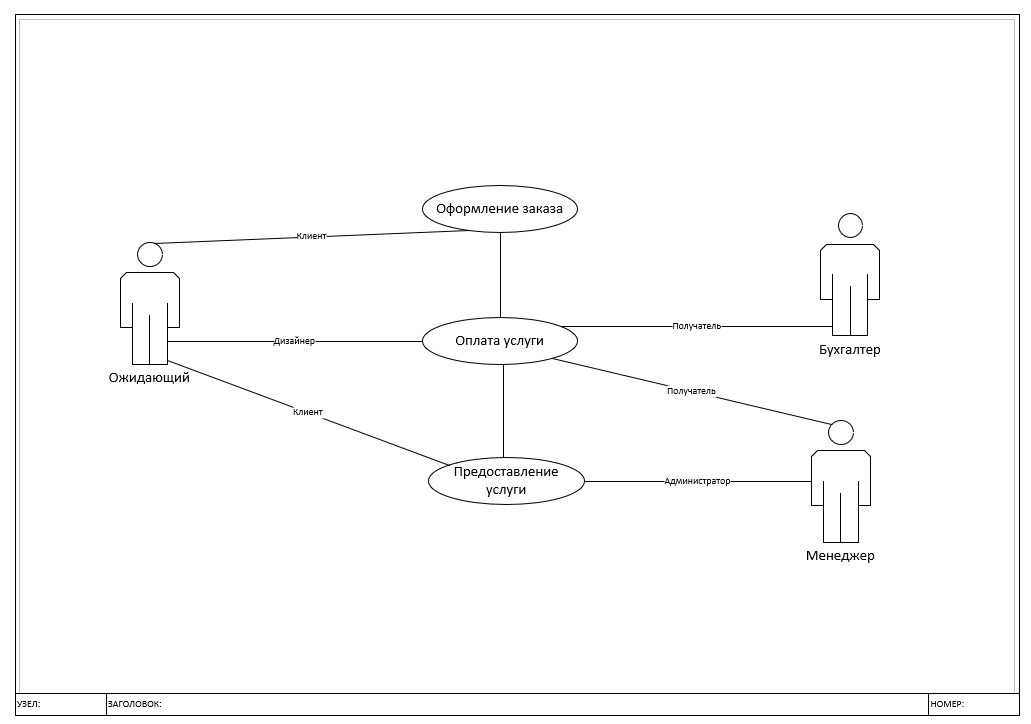


Рис. 7 Диаграмма кооперации

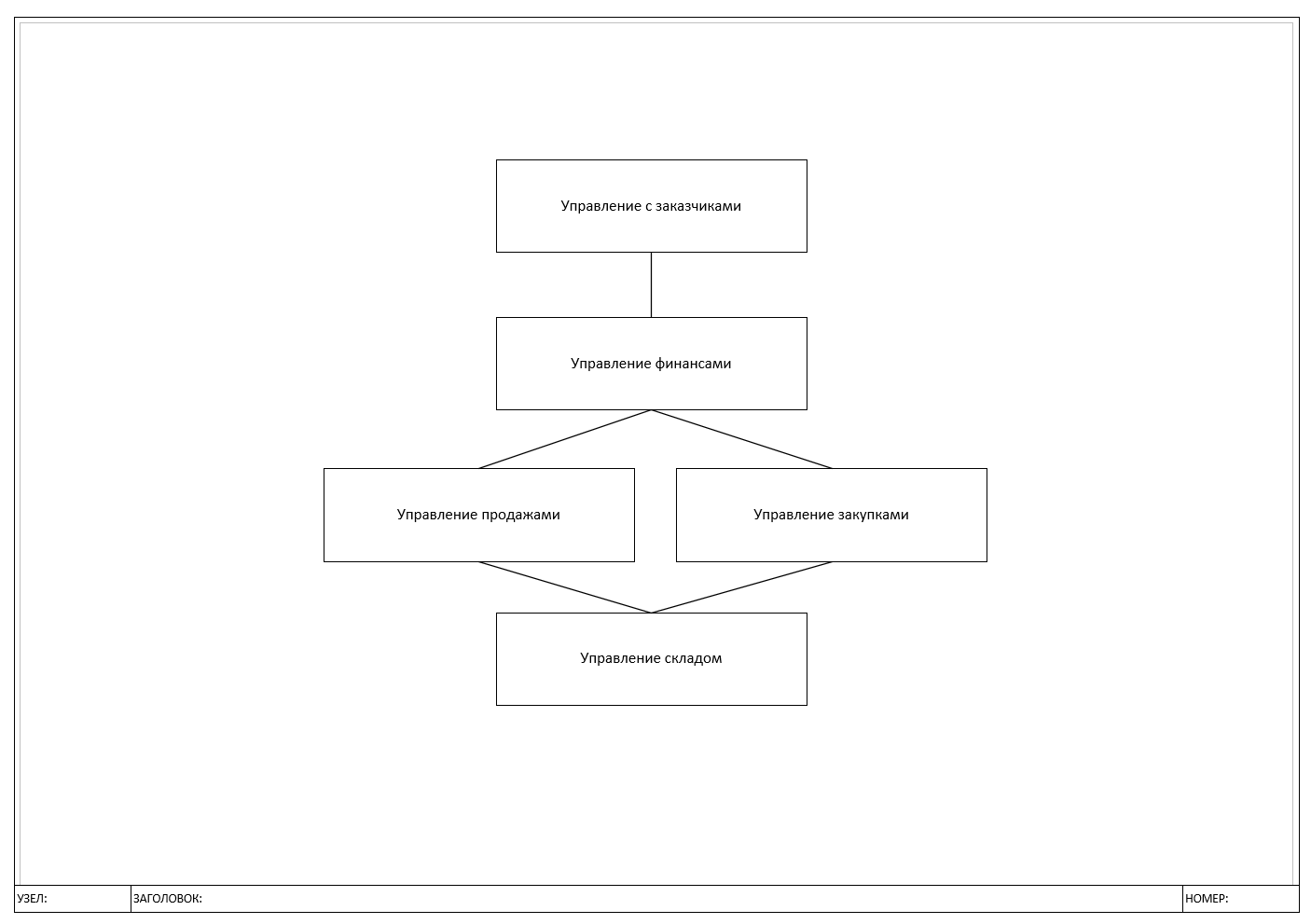


Рис. 8 Диаграмма компонентов

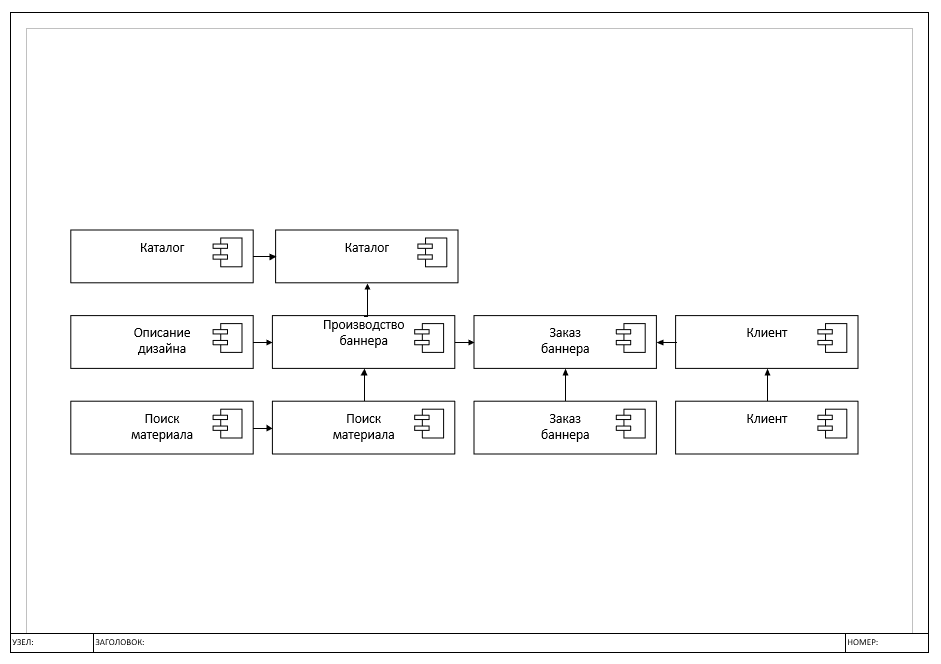


Рис. 9 Диаграмма развёртывания

# 1.6. Wireframe

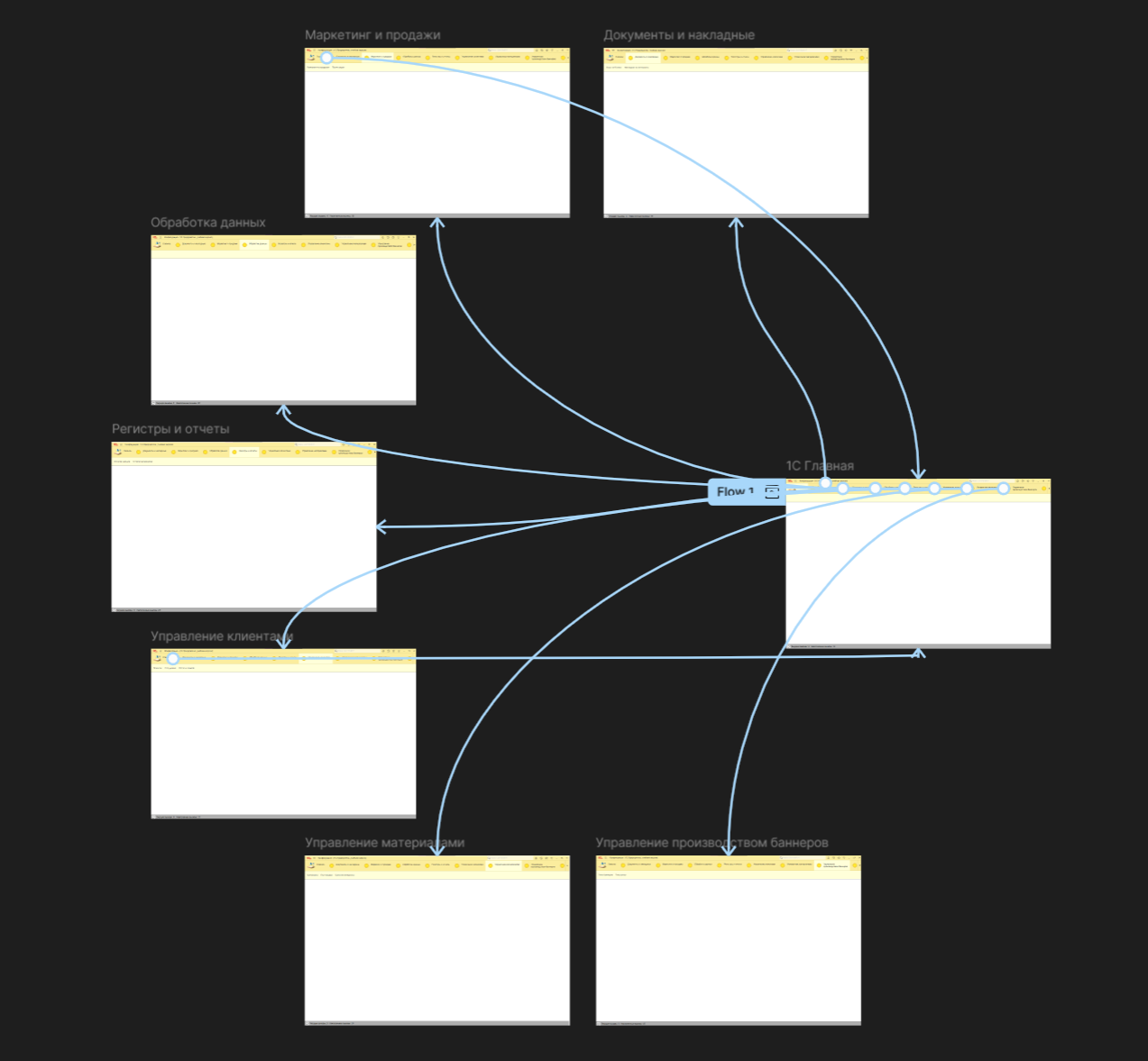


Рис. 10 Прототип wireframe

Ссылка на проект:

<https://www.figma.com/design/sg6xkIzibktvacqP3pfMSP/Производство-баннеров?node-id=0-1&p=f&t=TW4oYahdc0jTEobS-0>

**1.7. Схемы структуры**

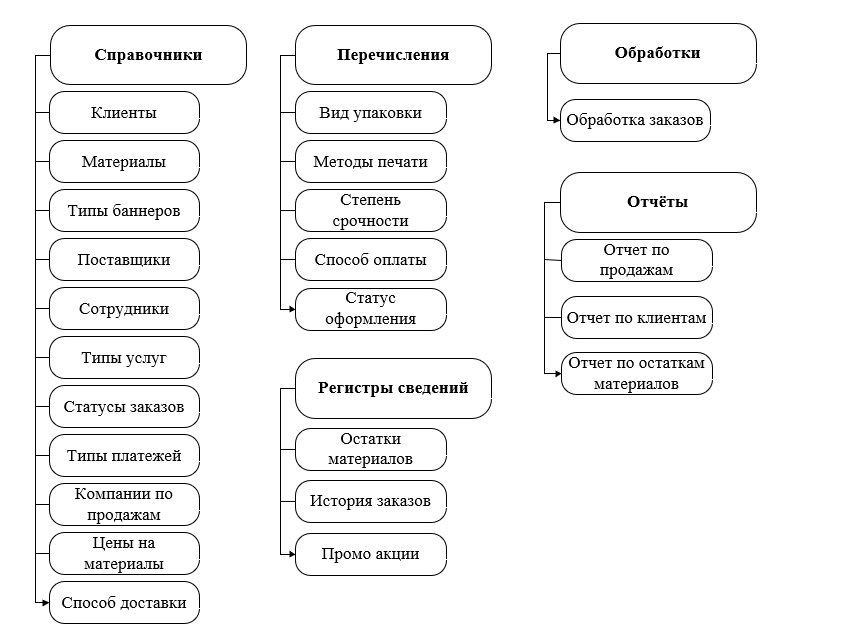
****

Рис. 11 Комплексная схема конфигурации

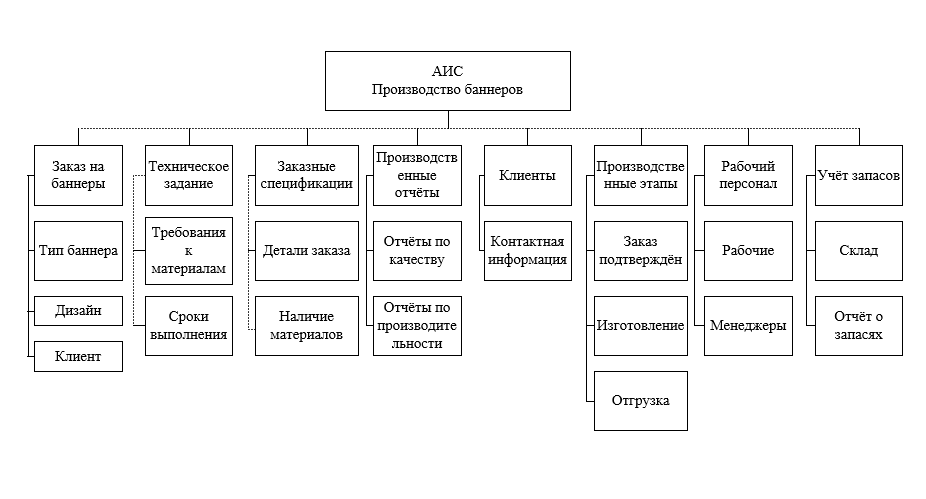


Рис. 12 Структура конфигурации

# ГЛАВА 2. РАЗРАБОТКА БАЗЫ ДАННЫХ

# 2.1. Разработка логической модели данных

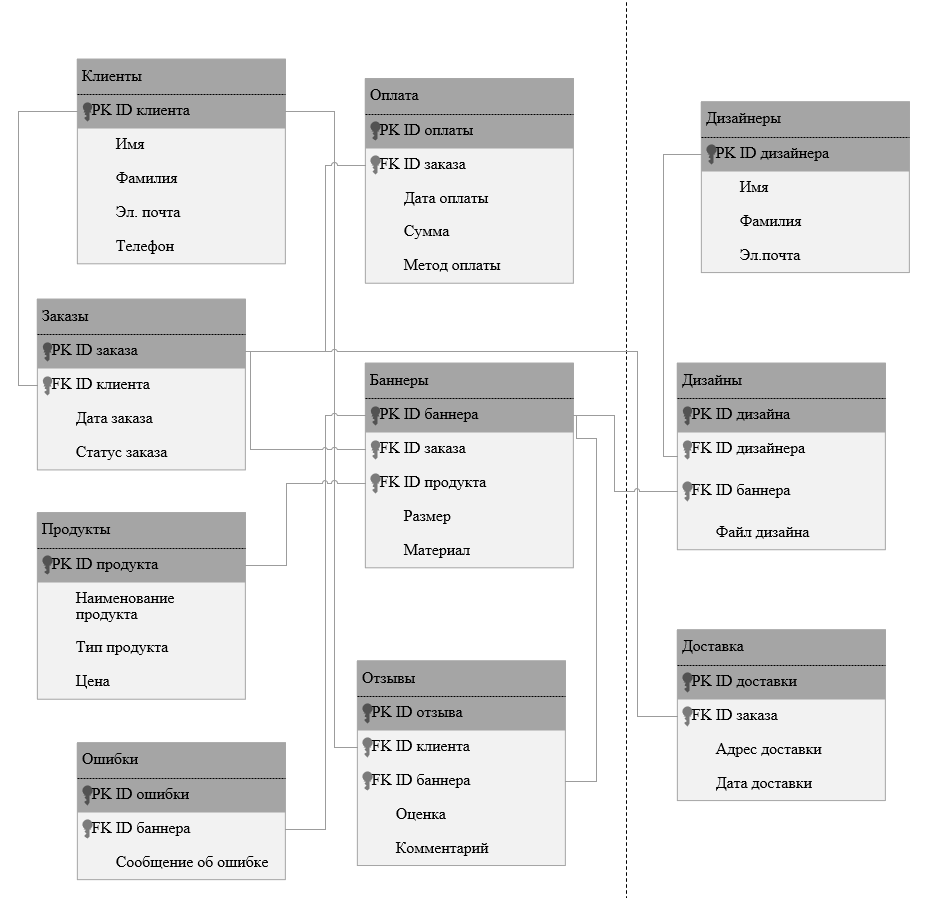


Рис. 13 ER - диаграмма

# 2.2. Добавление таблиц к базе данных

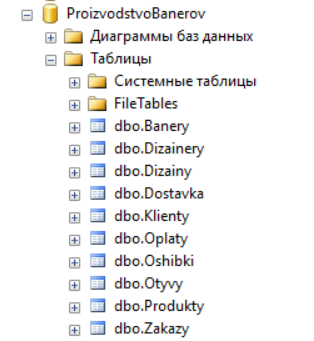


Рис. 14 Список таблиц

# 2.3. Просмотр связей на диаграмме базы данных

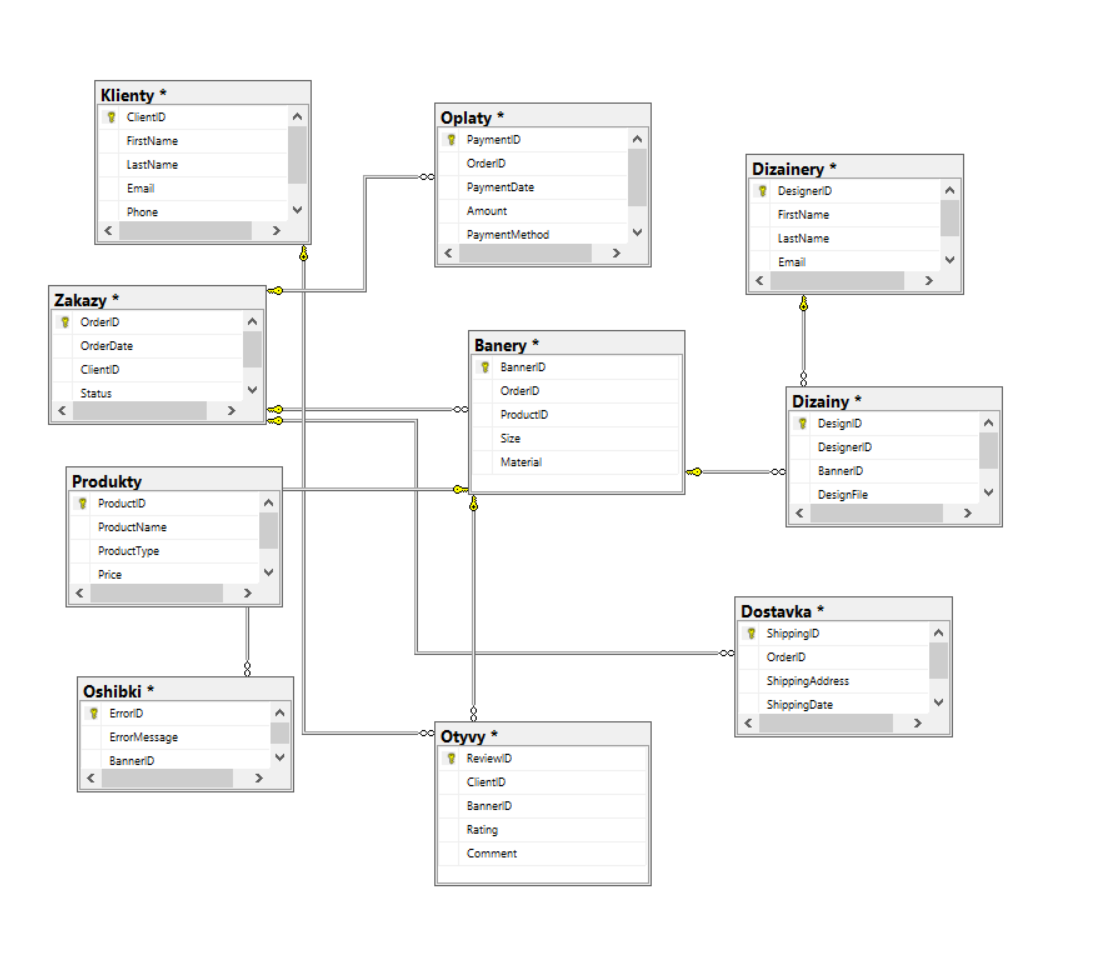


Рис. 15 Диаграмма базы данных

# 2.4. Определение ограничений, налагаемы на данные

Каждый столбец в таблице имеет свои подходящие данные, которые могут быть указаны только в ином столбце:

Таблица «Клиенты»

1. ID\_Client «int» - Должен включать в себе значения int и не иметь нулевого значения.
2. FirstName «varchar» - Должен содержать текстовые значения с ограничением в 50 символов.
3. Surname «varchar» - Должен содержать текстовые значения с ограничением в 50 символов.
4. Email «varchar» - Должен содержать текстовые значения с ограничением в 100 символов.
5. Phone «varchar» - Должен содержать текстовые значения с ограничением в 15 символов.

Таблица «Заказы»

1. ID\_Order «int» - Должен включать в себе значения int и не иметь нулевого значения.
2. ID\_Client «int» - Должен включать в себе значения int и не иметь нулевого значения.
3. OrderDate «date» - Должен содержать дату оформления заказа.
4. OrderStatus «varchar» - Должен содержать текстовые значения с ограничением в 50 символов.

Таблица «Продукты»

1. ID\_Product «int» - Должен включать в себе значения int и не иметь нулевого значения.
2. ProductName «varchar» - Должен содержать текстовые значения с ограничением в 100 символов.
3. ProductType «varchar» - Должен содержать текстовые значения с ограничением в 50 символов.
4. Price «decimal» - Должен иметь десятичное значение, представляющее цену продукта.

Таблица «Баннеры»

1. ID\_Banner «int» - Должен включать в себе значения int и не иметь нулевого значения.
2. ID\_Order «int» - Должен включать в себе значения int и не иметь нулевого значения.
3. ID\_Product «int» - Должен включать в себе значения int и не иметь нулевого значения.
4. Size «varchar» - Должен содержать текстовые значения с ограничением в 50 символов.
5. Material «varchar» - Должен содержать текстовые значения с ограничением в 50 символов.

Таблица «Оплата»

1. ID\_Payment «int» - Должен включать в себе значения int и не иметь нулевого значения.
2. ID\_Order «int» - Должен включать в себе значения int и не иметь нулевого значения.
3. PaymentDate «date» - Должен содержать дату оплаты.
4. Amount «decimal» - Должен иметь десятичное значение, представляющее сумму оплаты.
5. PaymentMethod «varchar» - Должен содержать текстовые значения с ограничением в 50 символов.

Таблица «Дизайнеры»

1. ID\_Designer «int» - Должен включать в себе значения int и не иметь нулевого значения.
2. FirstName «varchar» - Должен содержать текстовые значения с ограничением в 50 символов.
3. Surname «varchar» - Должен содержать текстовые значения с ограничением в 50 символов.
4. Email «varchar» - Должен содержать текстовые значения с ограничением в 100 символов.

Таблица «Дизайны»

1. ID\_Design «int» - Должен включать в себе значения int и не иметь нулевого значения.
2. ID\_Banner «int» - Должен включать в себе значения int и не иметь нулевого значения.
3. ID\_Designer «int» - Должен включать в себе значения int и не иметь нулевого значения.
4. DesignFile «varchar» - Должен содержать текстовые значения с ограничением в 255 символов.

Таблица «Отзывы»

1. ID\_Review «int» - Должен включать в себе значения int и не иметь нулевого значения.
2. ID\_Client «int» - Должен включать в себе значения int и не иметь нулевого значения.
3. ID\_Banner «int» - Должен включать в себе значения int и не иметь нулевого значения.
4. Rating «int» - Должен иметь значение от 1 до 5, представляющее оценку.
5. Comment «text» - Должен содержать текстовые значения, представляющие комментарий клиента.

Таблица «Доставка»

1. ID\_Delivery «int» - Должен включать в себе значения int и не иметь нулевого значения.
2. ID\_Order «int» - Должен включать в себе значения int и не иметь нулевого значения.
3. DeliveryAddress «varchar» - Должен содержать текстовые значения с ограничением в 255 символов.
4. DeliveryDate «date» - Должен содержать дату доставки.

# 2.5. Определение объемы и типов данных

Чтобы разработать приложение, нужно выделить данные, которые будет содержать наше приложение. Вот категории, которые можно вывести:

1. Администраторы
2. Клиенты
3. Заказы
4. Продукты
5. Баннеры
6. Дизайнеры
7. Доставка

# 2.6. Определение способа использования данных

Для разработки приложения важно определить роли и задачи двух категорий пользователей: Администратор и Дизайнер. Ниже приведены описания этих пользователей и их задач.

Таблица 2

Данные пользователей

|  |  |
| --- | --- |
| **Категория пользователей** | **Существующее число** |
| Администратор | 1 |
| Дизайнер | 1 |

Таблица 3

Задачи пользователей

|  |  |
| --- | --- |
| **Категория пользователей** | **Задачи** |
| Администратор | Поддержка информации, хранимой в БД; Обеспечение безопасности и надежности данных; Мониторинг активности пользователей; Управление доступом к системе; Настройка уровней доступа и ролей. |
| Дизайнер | Доступ к заказам на производство баннеров; Создание и редактирование дизайнов баннеров; Обработка отзывов клиентов; Взаимодействие с клиентами по вопросам дизайна и материалов. |

# 2.7. Разработка физической структуры данных

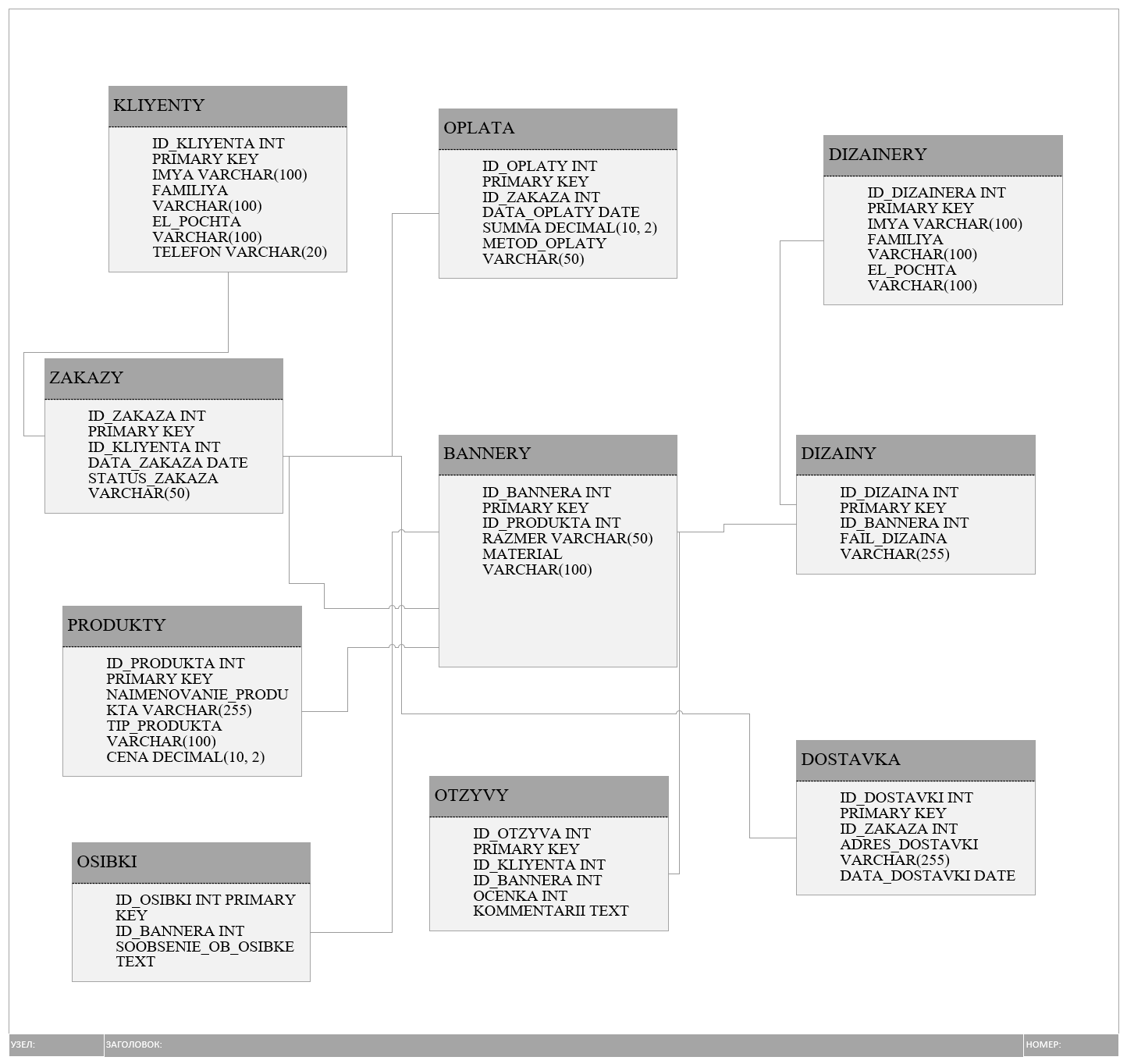


Рис. 16 Структура данных

# 2.7.1. Создание таблиц в БД (скрипт)

CREATE TABLE Klienty (

ClientID INT PRIMARY KEY,

FirstName VARCHAR(255),

LastName VARCHAR(255),

Email VARCHAR(255),

Phone VARCHAR(50)

);

CREATE TABLE Zakazy (

OrderID INT PRIMARY KEY,

OrderDate DATE,

ClientID INT,

Status VARCHAR(50)

);

CREATE TABLE Produkty (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(255),

ProductType VARCHAR(100),

Price VARCHAR(255)

);

CREATE TABLE Banery (

BannerID INT PRIMARY KEY,

OrderID INT,

ProductID INT,

Size VARCHAR(100),

Material VARCHAR(100)

);

CREATE TABLE Dizainery (

DesignerID INT PRIMARY KEY,

FirstName VARCHAR(255),

LastName VARCHAR(255),

Email VARCHAR(255)

);

CREATE TABLE Dizainy (

DesignID INT PRIMARY KEY,

DesignerID INT,

BannerID INT,

DesignFile VARCHAR(255)

);

CREATE TABLE Oshibki (

ErrorID INT PRIMARY KEY,

ErrorMessage TEXT,

BannerID INT

);

CREATE TABLE Oplaty (

PaymentID INT PRIMARY KEY,

OrderID INT,

PaymentDate VARCHAR(255),

Amount VARCHAR(255),

PaymentMethod VARCHAR(50)

);

CREATE TABLE Dostavka (

ShippingID INT PRIMARY KEY,

OrderID INT,

ShippingAddress VARCHAR(255),

ShippingDate VARCHAR(255)

);

CREATE TABLE Otyvy (

ReviewID INT PRIMARY KEY,

ClientID INT,

BannerID INT,

Rating VARCHAR(255),

Comment VARCHAR(255)

);

# 2.7.2. Данные (таблицы в режиме первые 200 строк)

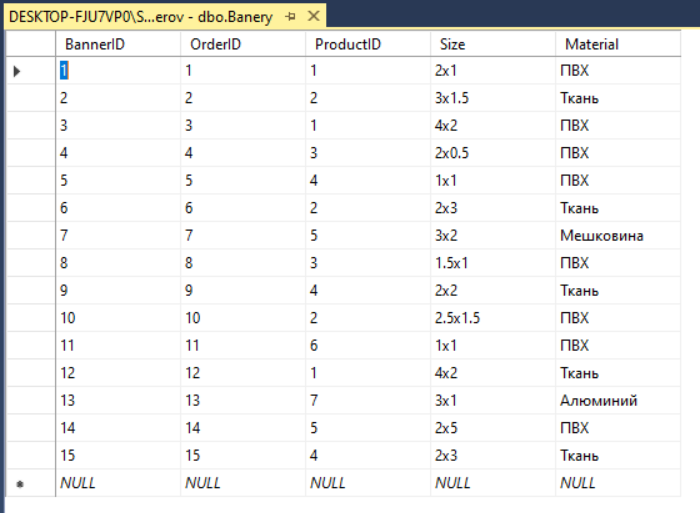


Рис. 17 Таблица «Баннеры»

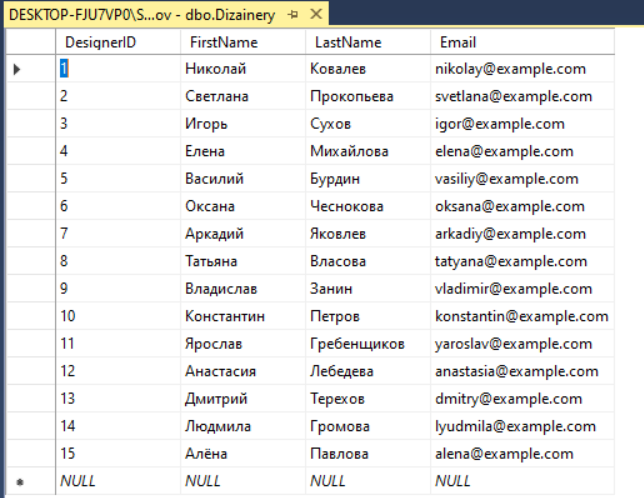


Рис. 18 Таблица «Дизайнеры»

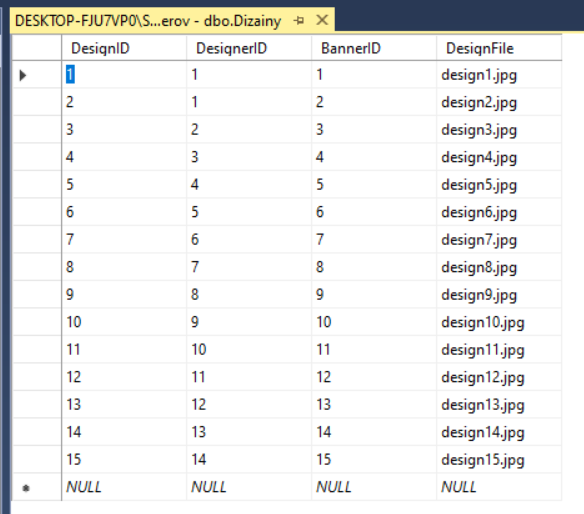


Рис. 19 Таблица «Дизайны»

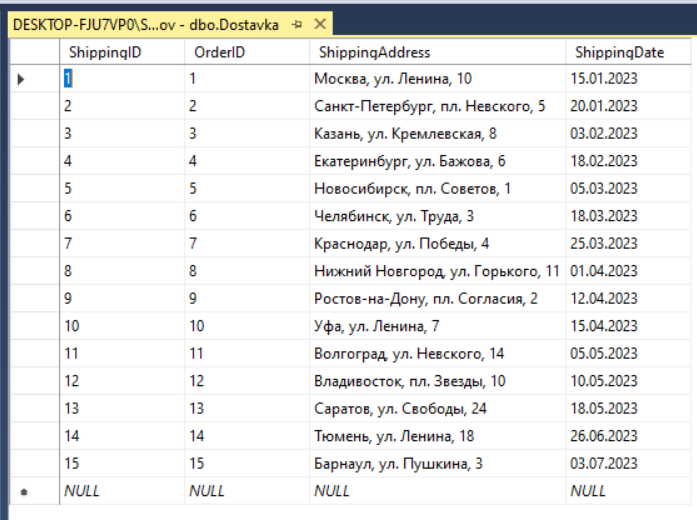


Рис. 20 Таблица «Доставка»

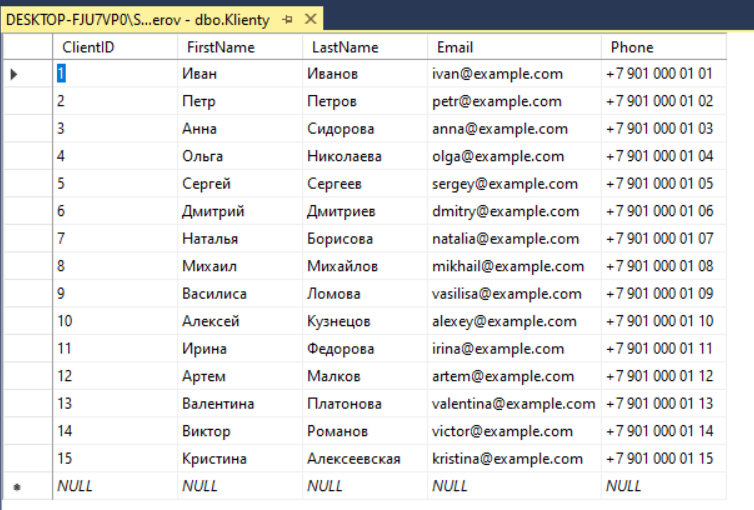


Рис. 21 Таблица «Клиенты»

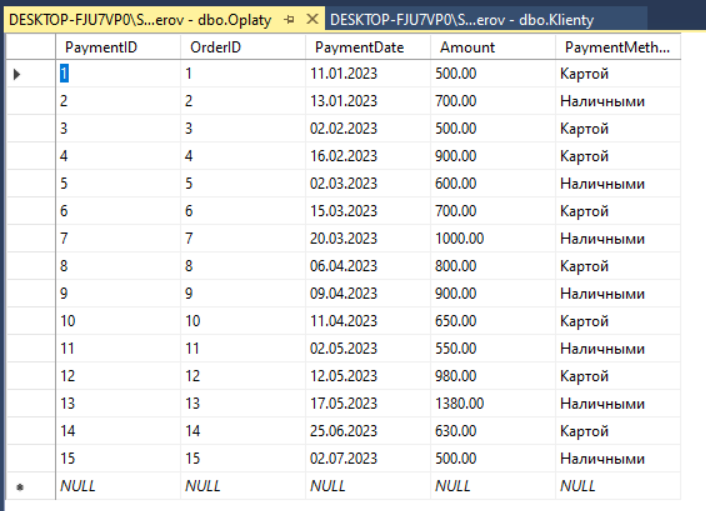


Рис. 22 Таблица «Оплата»

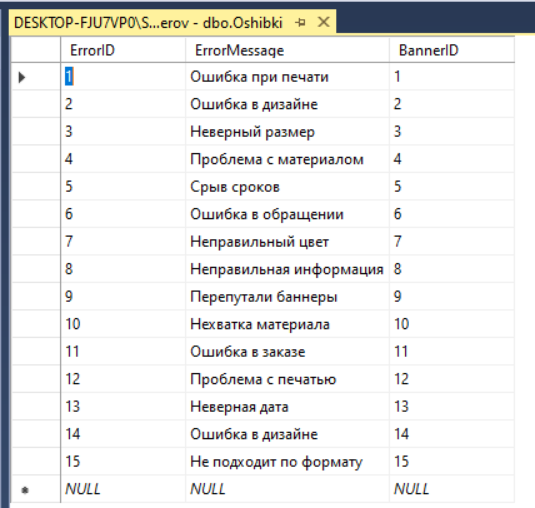


Рис. 23 Таблица «Ошибки»

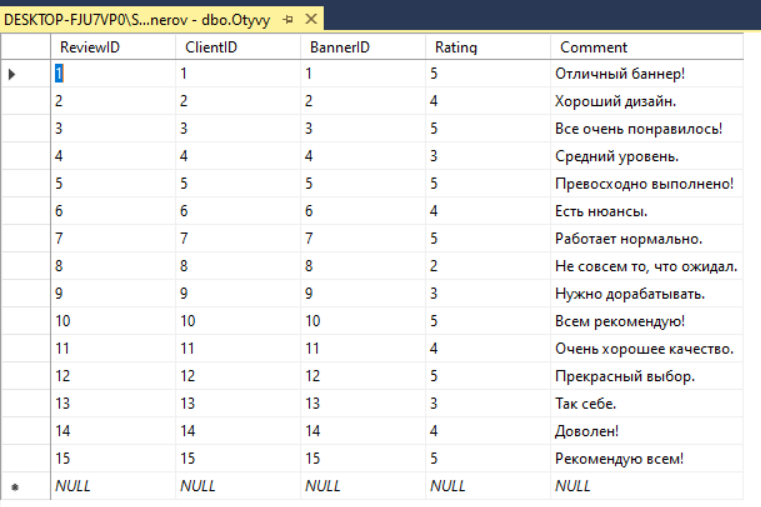


Рис. 24 Таблица «Отзывы»

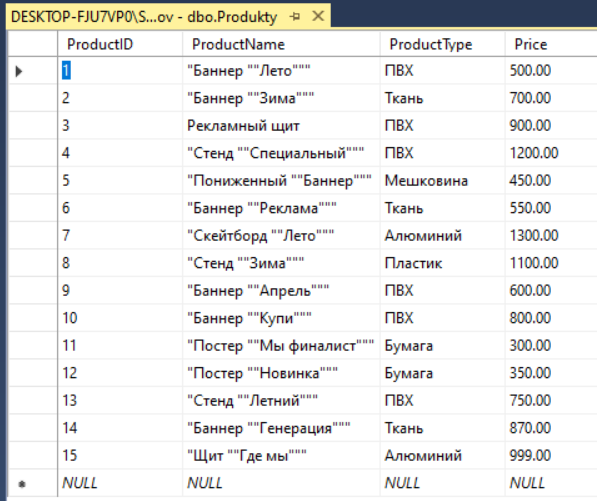


Рис. 25 Таблица «Продукты»

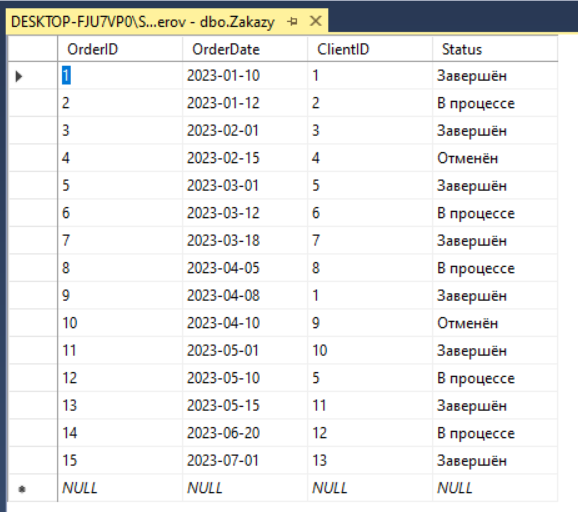


Рис. 26 Таблица «Заказы»

# ГЛАВА 3. РАЗРАБОТКА ПРИЛОЖЕНИЯ

# 3.1. Разработка пользовательского интерфейса

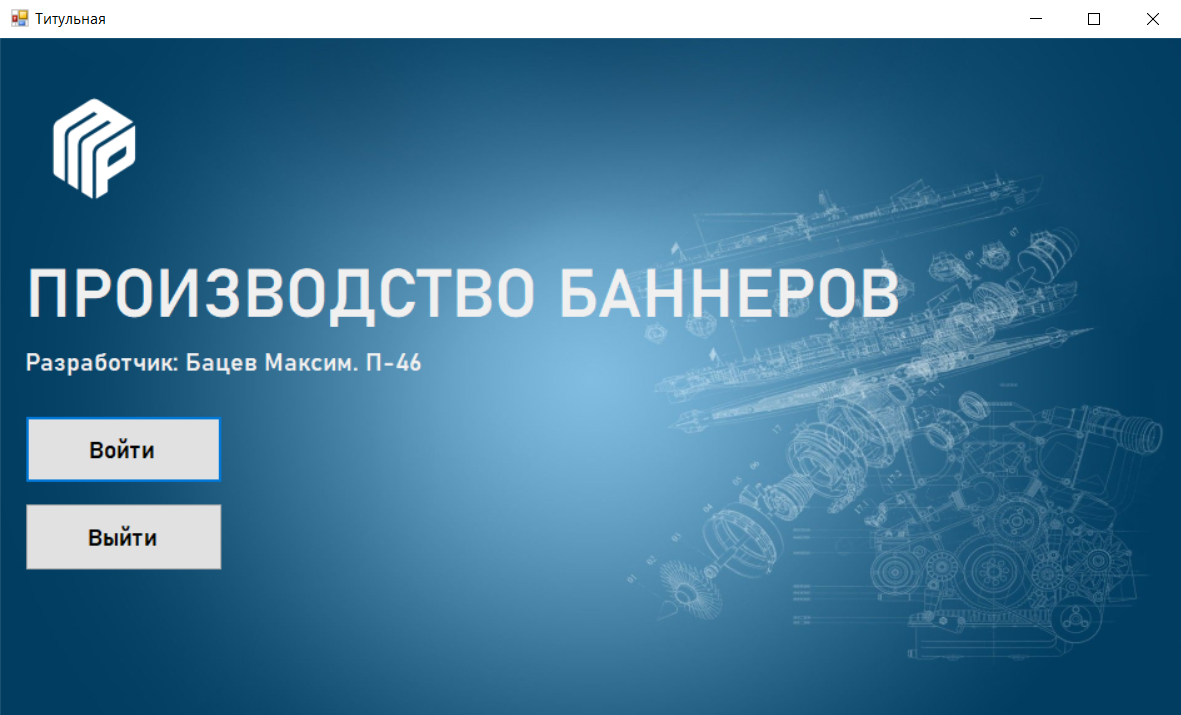
****

Рис. 27 Титульная форма

****

Рис. 28 Форма «Меню»

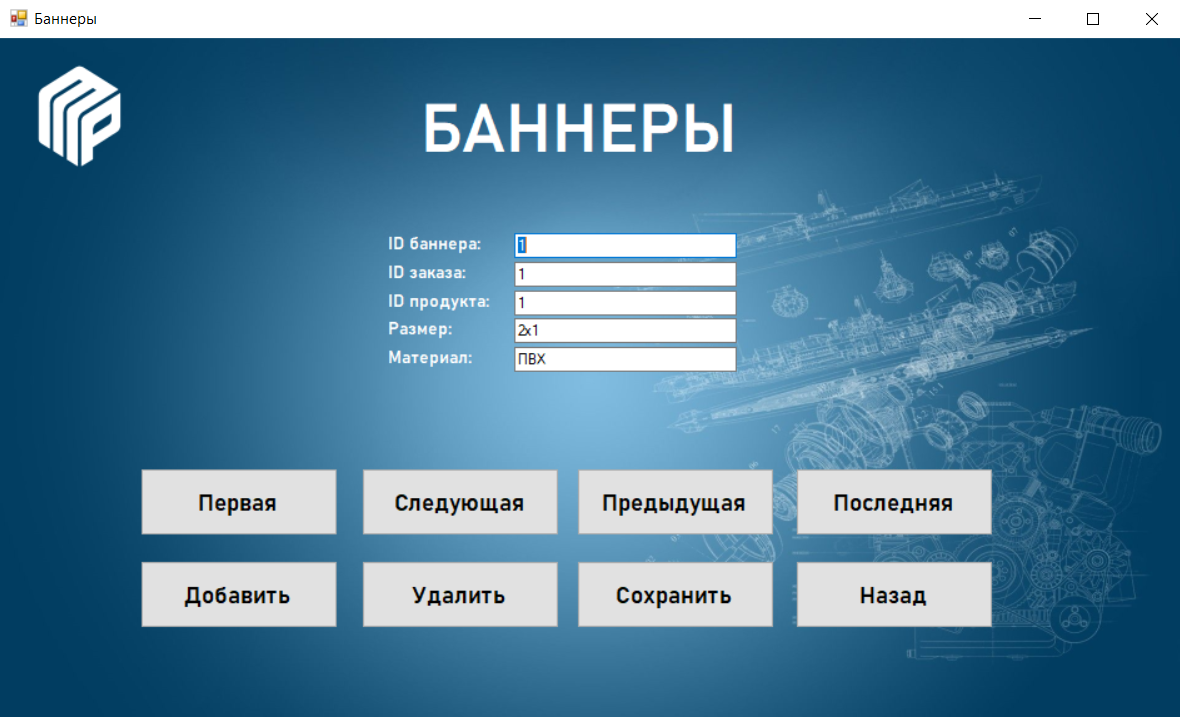


Рис. 29 Форма «Баннеры»

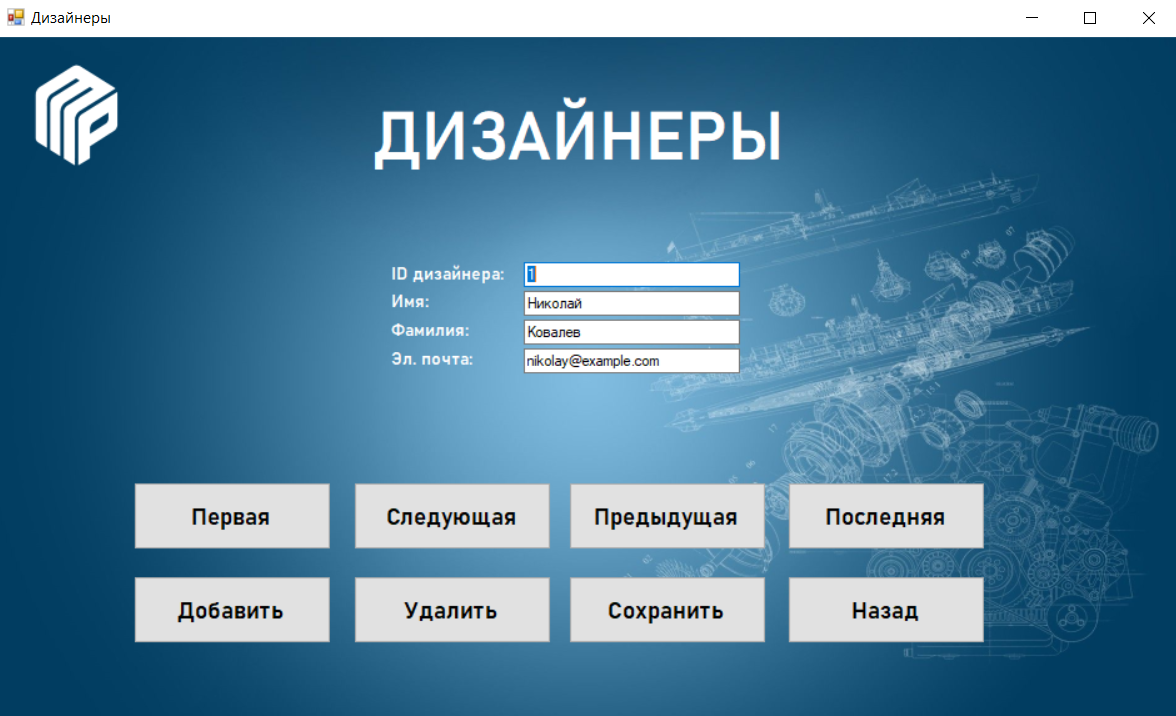


Рис. 30 Форма «Дизайнеры»

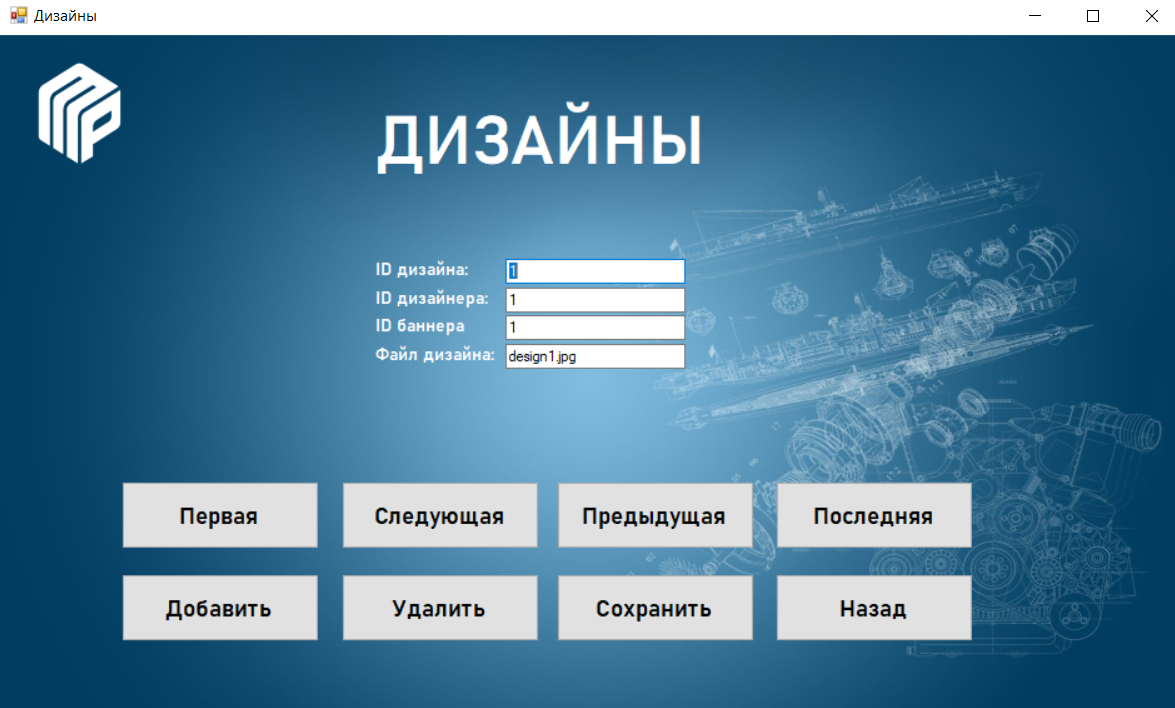


Рис. 31 Форма «Дизайны»

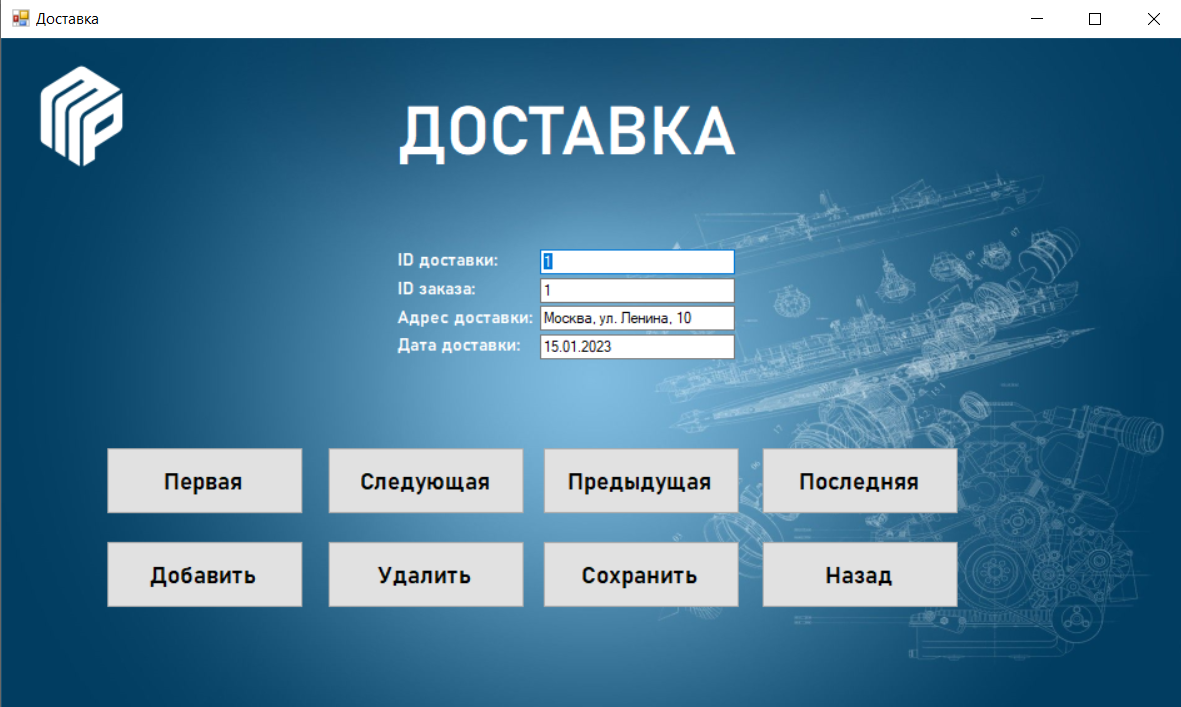


Рис. 32 Форма «Доставка»

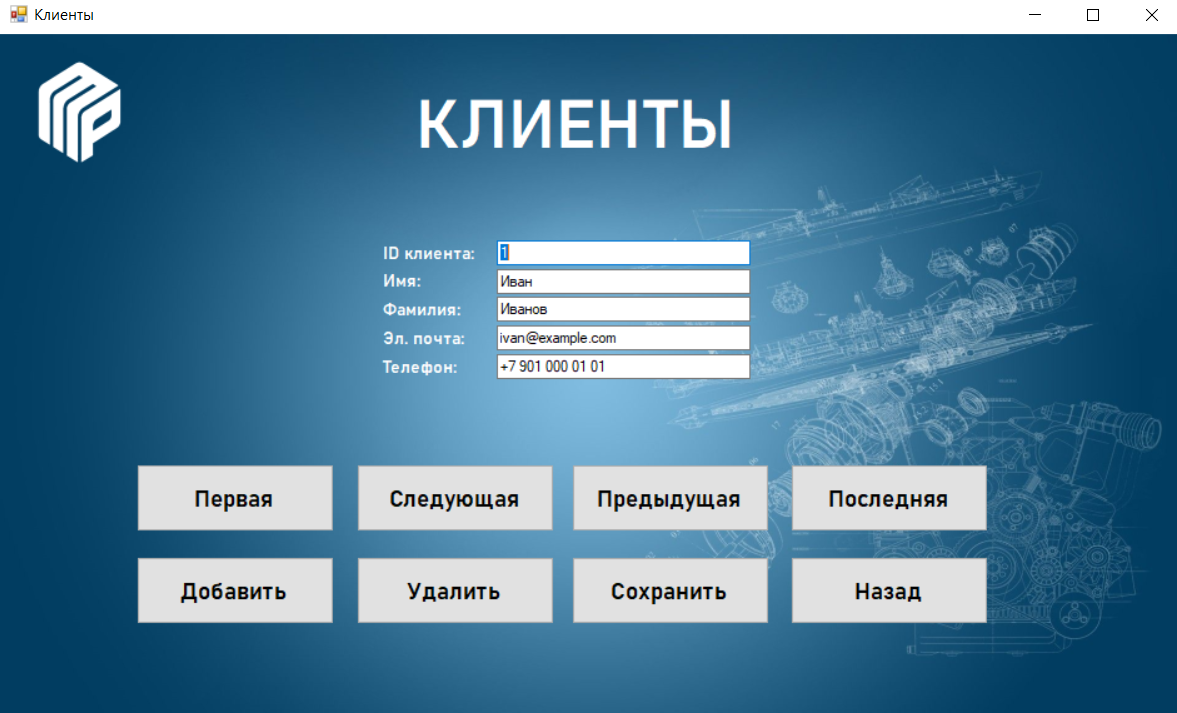


Рис. 33 Форма «Клиенты»



Рис. 34 Форма «Оплата»

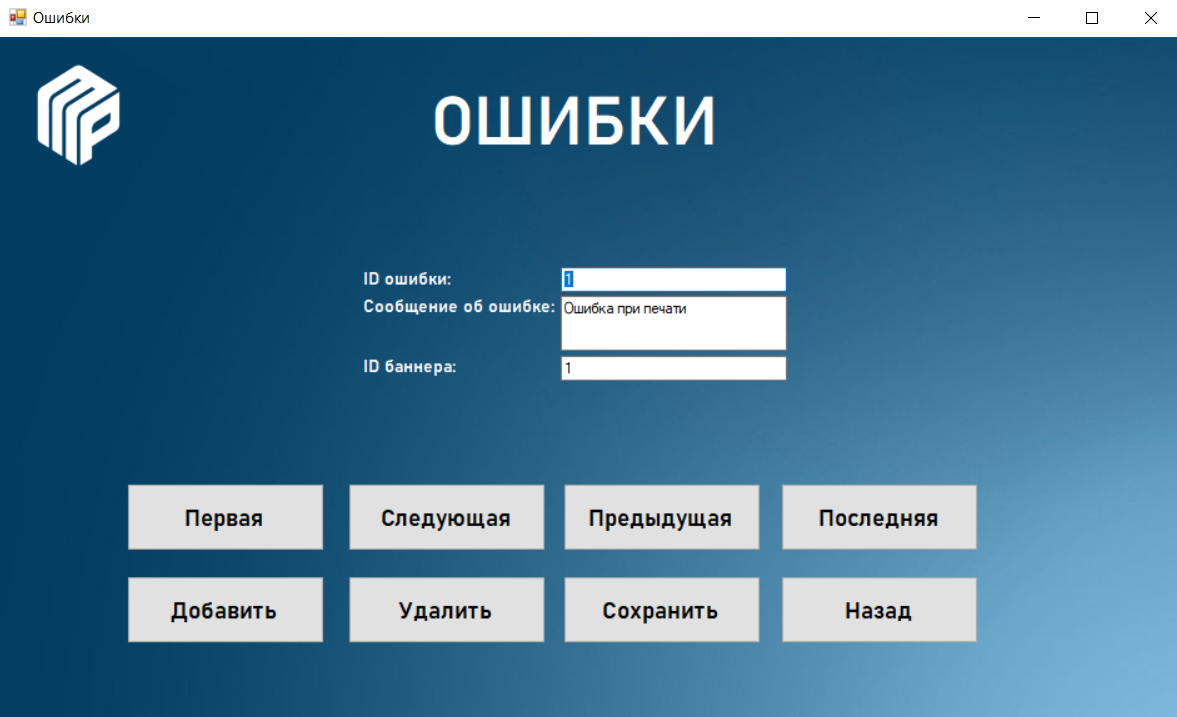


Рис. 35 Форма «Ошибки»

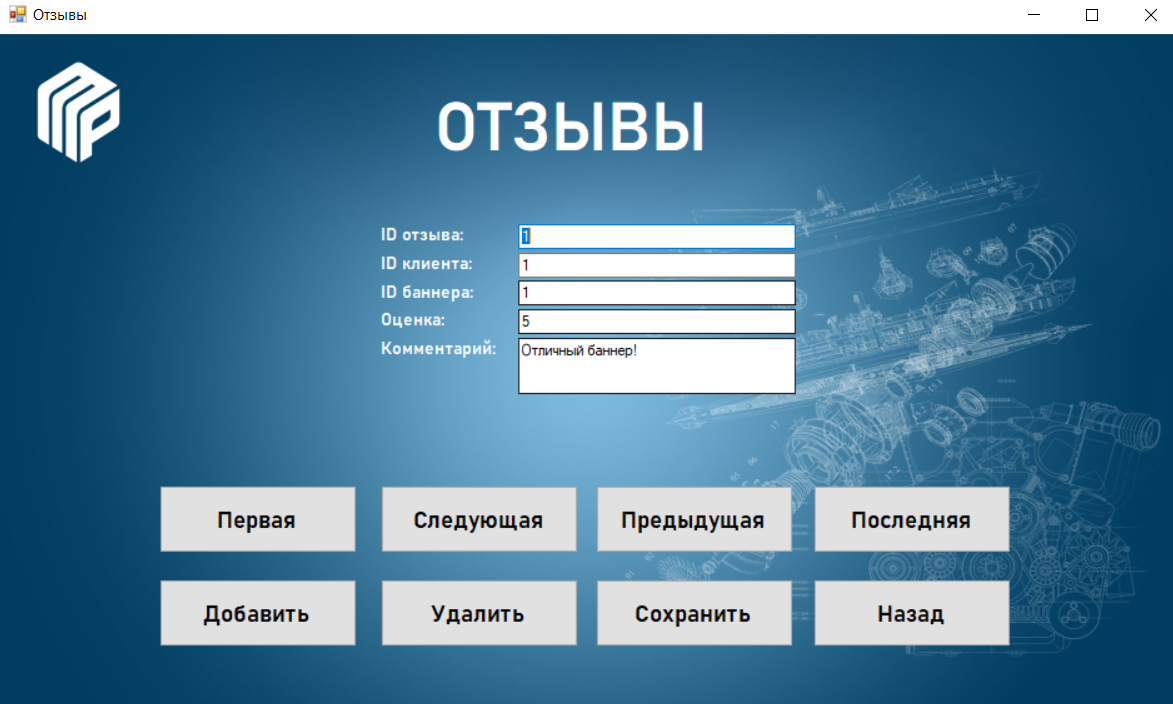


Рис. 36 Форма «Отзывы»

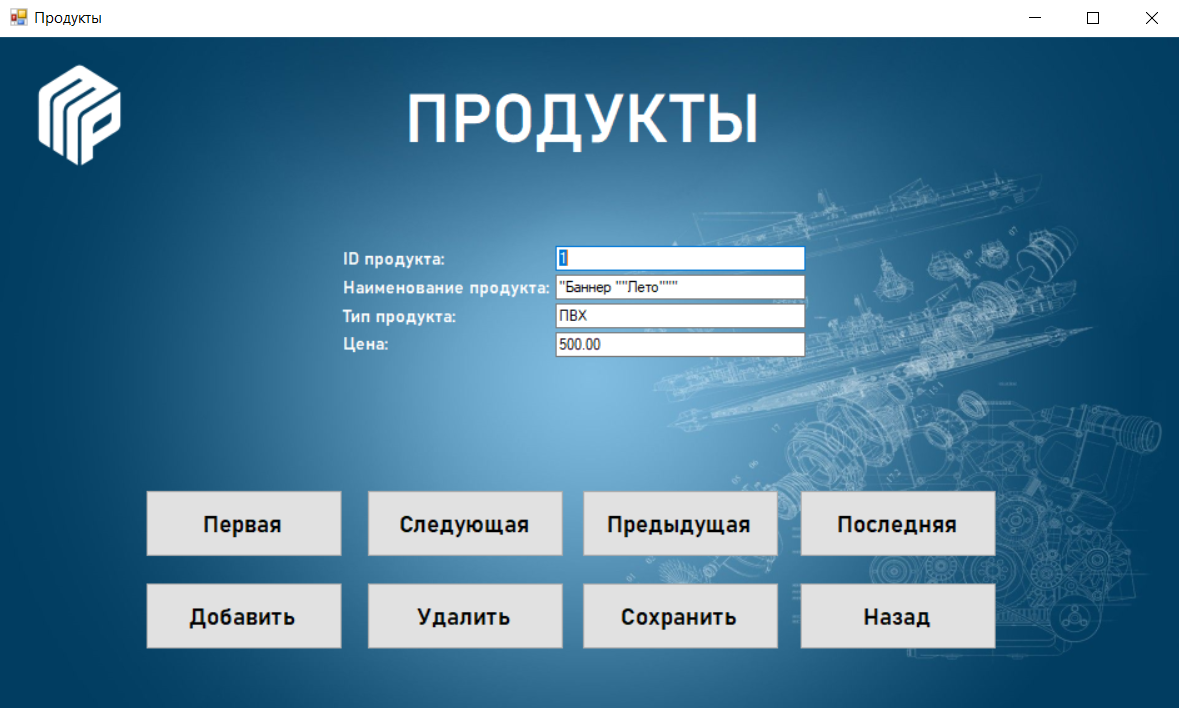


Рис. 37 Форма «Продукты»

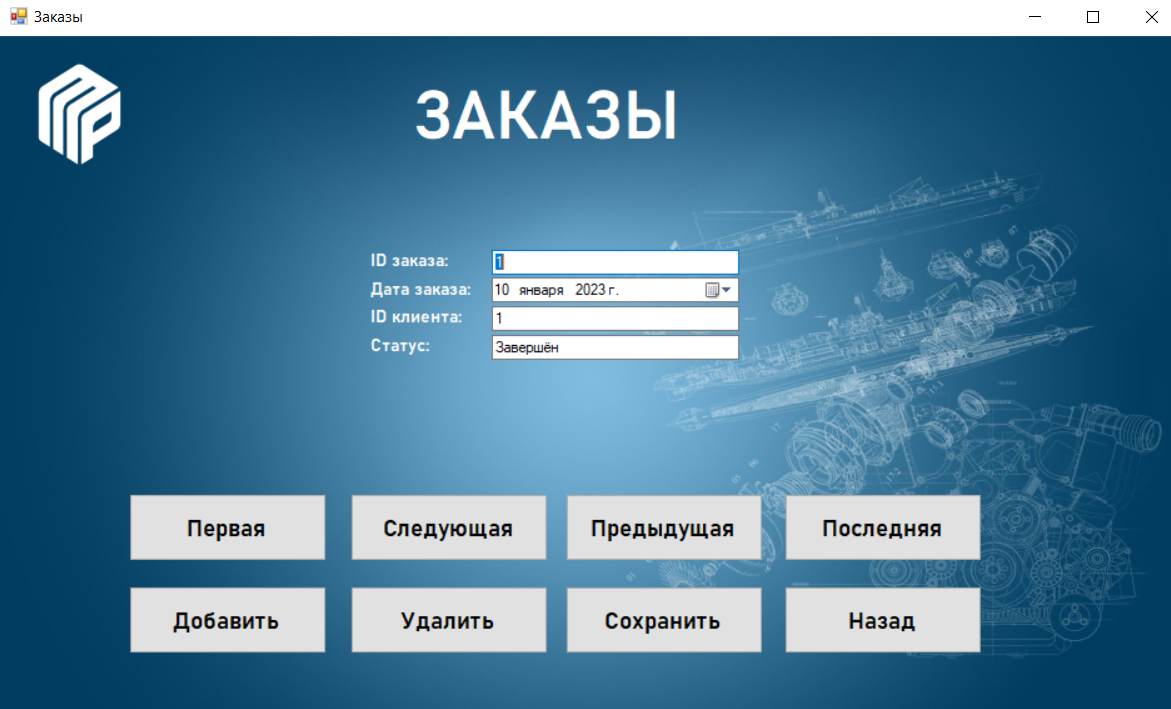


Рис. 38 Форма «Заказы»

# 3.2. Листинг программы с комментариями

Титульная форма

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace UP\_ProizvodstvoBannerov

{

public partial class Titylnaya : Form

{

public Titylnaya()

{

InitializeComponent();

}

private void Enter\_Click(object sender, EventArgs e)

{

Menu frm = new Menu();

frm.Show();

this.Hide();

}

private void Exit\_Click(object sender, EventArgs e)

{

Close();

}

}

}

Форма «Меню»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace UP\_ProizvodstvoBannerov

{

public partial class Menu : Form

{

public Menu()

{

InitializeComponent();

}

private void Bannery\_Click(object sender, EventArgs e)

{

Bannery frm = new Bannery();

frm.Show();

this.Hide();

}

private void Back\_Click(object sender, EventArgs e)

{

Titylnaya frm = new Titylnaya();

frm.Show();

this.Hide();

}

private void Dizainery\_Click(object sender, EventArgs e)

{

Dizaynery frm = new Dizaynery();

frm.Show();

this.Hide();

}

private void Dizainy\_Click(object sender, EventArgs e)

{

Dizainy frm = new Dizainy();

frm.Show();

this.Hide();

}

private void Dostavka\_Click(object sender, EventArgs e)

{

Dostavka frm = new Dostavka();

frm.Show();

this.Hide();

}

private void Klienty\_Click(object sender, EventArgs e)

{

Klienty frm = new Klienty();

frm.Show();

this.Hide();

}

private void Oplata\_Click(object sender, EventArgs e)

{

Oplata frm = new Oplata();

frm.Show();

this.Hide();

}

private void Oshibki\_Click(object sender, EventArgs e)

{

Oshibki frm = new Oshibki();

frm.Show();

this.Hide();

}

private void Otzivy\_Click(object sender, EventArgs e)

{

Otzyvy frm = new Otzyvy();

frm.Show();

this.Hide();

}

private void Zakazy\_Click(object sender, EventArgs e)

{

Zakazy frm = new Zakazy();

frm.Show();

this.Hide();

}

private void Produkty\_Click(object sender, EventArgs e)

{

Produkty frm = new Produkty();

frm.Show();

this.Hide();

}

}

}

Форма «Баннеры»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace UP\_ProizvodstvoBannerov

{

public partial class Bannery : Form

{

public Bannery()

{

InitializeComponent();

}

private void baneryBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.baneryBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Bannery\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "proizvodstvoBanerovDataSet.Banery". При необходимости она может быть перемещена или удалена.

this.baneryTableAdapter.Fill(this.proizvodstvoBanerovDataSet.Banery);

}

private void First\_Click(object sender, EventArgs e)

{

baneryBindingSource.MoveFirst();

}

private void Save\_Click(object sender, EventArgs e)

{

this.Validate();

this.baneryBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Delete\_Click(object sender, EventArgs e)

{

baneryBindingSource.RemoveCurrent();

}

private void Add\_Click(object sender, EventArgs e)

{

baneryBindingSource.AddNew();

}

private void Previous\_Click(object sender, EventArgs e)

{

baneryBindingSource.MovePrevious();

}

private void Last\_Click(object sender, EventArgs e)

{

baneryBindingSource.MoveLast();

}

private void Next\_Click(object sender, EventArgs e)

{

baneryBindingSource.MoveNext();

}

private void Back\_Click(object sender, EventArgs e)

{

Menu frm = new Menu();

frm.Show();

this.Hide();

}

}

}

Форма «Дизайнеры»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace UP\_ProizvodstvoBannerov

{

public partial class Dizaynery : Form

{

public Dizaynery()

{

InitializeComponent();

}

private void dizaineryBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.dizaineryBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Dizaynery\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "proizvodstvoBanerovDataSet.Dizainery". При необходимости она может быть перемещена или удалена.

this.dizaineryTableAdapter.Fill(this.proizvodstvoBanerovDataSet.Dizainery);

}

private void First\_Click(object sender, EventArgs e)

{

dizaineryBindingSource.MoveFirst();

}

private void Save\_Click(object sender, EventArgs e)

{

this.Validate();

this.dizaineryBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Delete\_Click(object sender, EventArgs e)

{

dizaineryBindingSource.RemoveCurrent();

}

private void Add\_Click(object sender, EventArgs e)

{

dizaineryBindingSource.AddNew();

}

private void Previous\_Click(object sender, EventArgs e)

{

dizaineryBindingSource.MovePrevious();

}

private void Last\_Click(object sender, EventArgs e)

{

dizaineryBindingSource.MoveLast();

}

private void Next\_Click(object sender, EventArgs e)

{

dizaineryBindingSource.MoveNext();

}

private void Back\_Click(object sender, EventArgs e)

{

Menu frm = new Menu();

frm.Show();

this.Hide();

}

}

}

Форма «Дизайны»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace UP\_ProizvodstvoBannerov

{

public partial class Dizainy : Form

{

public Dizainy()

{

InitializeComponent();

}

private void dizainyBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.dizainyBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Dizainy\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "proizvodstvoBanerovDataSet.Dizainy". При необходимости она может быть перемещена или удалена.

this.dizainyTableAdapter.Fill(this.proizvodstvoBanerovDataSet.Dizainy);

}

private void First\_Click(object sender, EventArgs e)

{

dizainyBindingSource.MoveFirst();

}

private void Save\_Click(object sender, EventArgs e)

{

this.Validate();

this.dizainyBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Delete\_Click(object sender, EventArgs e)

{

dizainyBindingSource.RemoveCurrent();

}

private void Add\_Click(object sender, EventArgs e)

{

dizainyBindingSource.AddNew();

}

private void Previous\_Click(object sender, EventArgs e)

{

dizainyBindingSource.MovePrevious();

}

private void Last\_Click(object sender, EventArgs e)

{

dizainyBindingSource.MoveLast();

}

private void Next\_Click(object sender, EventArgs e)

{

dizainyBindingSource.MoveNext();

}

private void Back\_Click(object sender, EventArgs e)

{

Menu frm = new Menu();

frm.Show();

this.Hide();

}

}

}

Форма «Доставка»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace UP\_ProizvodstvoBannerov

{

public partial class Dostavka : Form

{

public Dostavka()

{

InitializeComponent();

}

private void dostavkaBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.dostavkaBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Dostavka\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "proizvodstvoBanerovDataSet.Dostavka". При необходимости она может быть перемещена или удалена.

this.dostavkaTableAdapter.Fill(this.proizvodstvoBanerovDataSet.Dostavka);

}

private void First\_Click(object sender, EventArgs e)

{

dostavkaBindingSource.MoveFirst();

}

private void Save\_Click(object sender, EventArgs e)

{

this.Validate();

this.dostavkaBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Delete\_Click(object sender, EventArgs e)

{

dostavkaBindingSource.RemoveCurrent();

}

private void Add\_Click(object sender, EventArgs e)

{

dostavkaBindingSource.AddNew();

}

private void Previous\_Click(object sender, EventArgs e)

{

dostavkaBindingSource.MovePrevious();

}

private void Last\_Click(object sender, EventArgs e)

{

dostavkaBindingSource.MoveLast();

}

private void Next\_Click(object sender, EventArgs e)

{

dostavkaBindingSource.MoveNext();

}

private void Back\_Click(object sender, EventArgs e)

{

Menu frm = new Menu();

frm.Show();

this.Hide();

}

}

}

Форма «Клиенты»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace UP\_ProizvodstvoBannerov

{

public partial class Klienty : Form

{

public Klienty()

{

InitializeComponent();

}

private void klientyBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.klientyBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Klienty\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "proizvodstvoBanerovDataSet.Klienty". При необходимости она может быть перемещена или удалена.

this.klientyTableAdapter.Fill(this.proizvodstvoBanerovDataSet.Klienty);

}

private void First\_Click(object sender, EventArgs e)

{

klientyBindingSource.MoveFirst();

}

private void Save\_Click(object sender, EventArgs e)

{

this.Validate();

this.klientyBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Back\_Click(object sender, EventArgs e)

{

Menu frm = new Menu();

frm.Show();

this.Hide();

}

private void Add\_Click(object sender, EventArgs e)

{

klientyBindingSource.AddNew();

}

private void Previous\_Click(object sender, EventArgs e)

{

klientyBindingSource.MovePrevious();

}

private void Last\_Click(object sender, EventArgs e)

{

klientyBindingSource.MoveLast();

}

private void Next\_Click(object sender, EventArgs e)

{

klientyBindingSource.MoveNext();

}

private void Delete\_Click(object sender, EventArgs e)

{

klientyBindingSource.RemoveCurrent();

}

}

}

Форма «Оплата»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace UP\_ProizvodstvoBannerov

{

public partial class Oplata : Form

{

public Oplata()

{

InitializeComponent();

}

private void oplatyBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.oplatyBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Oplata\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "proizvodstvoBanerovDataSet.Oplaty". При необходимости она может быть перемещена или удалена.

this.oplatyTableAdapter.Fill(this.proizvodstvoBanerovDataSet.Oplaty);

}

private void First\_Click(object sender, EventArgs e)

{

oplatyBindingSource.MoveFirst();

}

private void Save\_Click(object sender, EventArgs e)

{

this.Validate();

this.oplatyBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Delete\_Click(object sender, EventArgs e)

{

oplatyBindingSource.RemoveCurrent();

}

private void Add\_Click(object sender, EventArgs e)

{

oplatyBindingSource.AddNew();

}

private void Previous\_Click(object sender, EventArgs e)

{

oplatyBindingSource.MovePrevious();

}

private void Last\_Click(object sender, EventArgs e)

{

oplatyBindingSource.MoveLast();

}

private void Next\_Click(object sender, EventArgs e)

{

oplatyBindingSource.MoveNext();

}

private void Back\_Click(object sender, EventArgs e)

{

Menu frm = new Menu();

frm.Show();

this.Hide();

}

}

}

Форма «Ошибки»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace UP\_ProizvodstvoBannerov

{

public partial class Oshibki : Form

{

public Oshibki()

{

InitializeComponent();

}

private void oshibkiBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.oshibkiBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Oshibki\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "proizvodstvoBanerovDataSet.Oshibki". При необходимости она может быть перемещена или удалена.

this.oshibkiTableAdapter.Fill(this.proizvodstvoBanerovDataSet.Oshibki);

}

private void First\_Click(object sender, EventArgs e)

{

oshibkiBindingSource.MoveFirst();

}

private void Save\_Click(object sender, EventArgs e)

{

this.Validate();

this.oshibkiBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Delete\_Click(object sender, EventArgs e)

{

oshibkiBindingSource.RemoveCurrent();

}

private void Add\_Click(object sender, EventArgs e)

{

oshibkiBindingSource.AddNew();

}

private void Previous\_Click(object sender, EventArgs e)

{

oshibkiBindingSource.MovePrevious();

}

private void Last\_Click(object sender, EventArgs e)

{

oshibkiBindingSource.MoveLast();

}

private void Next\_Click(object sender, EventArgs e)

{

oshibkiBindingSource.MoveNext();

}

private void Back\_Click(object sender, EventArgs e)

{

Menu frm = new Menu();

frm.Show();

this.Hide();

}

}

}

Форма «Отзывы»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace UP\_ProizvodstvoBannerov

{

public partial class Otzyvy : Form

{

public Otzyvy()

{

InitializeComponent();

}

private void otyvyBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.otyvyBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Otzyvy\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "proizvodstvoBanerovDataSet.Otyvy". При необходимости она может быть перемещена или удалена.

this.otyvyTableAdapter.Fill(this.proizvodstvoBanerovDataSet.Otyvy);

}

private void First\_Click(object sender, EventArgs e)

{

otyvyBindingSource.MoveFirst();

}

private void Save\_Click(object sender, EventArgs e)

{

this.Validate();

this.otyvyBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Back\_Click(object sender, EventArgs e)

{

Menu frm = new Menu();

frm.Show();

this.Hide();

}

private void Add\_Click(object sender, EventArgs e)

{

otyvyBindingSource.AddNew();

}

private void Previous\_Click(object sender, EventArgs e)

{

otyvyBindingSource.MovePrevious();

}

private void Last\_Click(object sender, EventArgs e)

{

otyvyBindingSource.MoveLast();

}

private void Next\_Click(object sender, EventArgs e)

{

otyvyBindingSource.MoveNext();

}

private void Delete\_Click(object sender, EventArgs e)

{

otyvyBindingSource.RemoveCurrent();

}

}

}

Форма «Продукты»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace UP\_ProizvodstvoBannerov

{

public partial class Produkty : Form

{

public Produkty()

{

InitializeComponent();

}

private void produktyBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.produktyBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Produkty\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "proizvodstvoBanerovDataSet.Produkty". При необходимости она может быть перемещена или удалена.

this.produktyTableAdapter.Fill(this.proizvodstvoBanerovDataSet.Produkty);

}

private void First\_Click(object sender, EventArgs e)

{

produktyBindingSource.MoveFirst();

}

private void Save\_Click(object sender, EventArgs e)

{

this.Validate();

this.produktyBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Delete\_Click(object sender, EventArgs e)

{

produktyBindingSource.RemoveCurrent();

}

private void Add\_Click(object sender, EventArgs e)

{

produktyBindingSource.AddNew();

}

private void Previous\_Click(object sender, EventArgs e)

{

produktyBindingSource.MovePrevious();

}

private void Last\_Click(object sender, EventArgs e)

{

produktyBindingSource.MoveLast();

}

private void Back\_Click(object sender, EventArgs e)

{

Menu frm = new Menu();

frm.Show();

this.Hide();

}

private void Next\_Click(object sender, EventArgs e)

{

produktyBindingSource.MoveNext();

}

}

}

Форма «Заказы»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace UP\_ProizvodstvoBannerov

{

public partial class Zakazy : Form

{

public Zakazy()

{

InitializeComponent();

}

private void zakazyBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.zakazyBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Zakazy\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "proizvodstvoBanerovDataSet.Zakazy". При необходимости она может быть перемещена или удалена.

this.zakazyTableAdapter.Fill(this.proizvodstvoBanerovDataSet.Zakazy);

}

private void First\_Click(object sender, EventArgs e)

{

zakazyBindingSource.MoveFirst();

}

private void Save\_Click(object sender, EventArgs e)

{

this.Validate();

this.zakazyBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.proizvodstvoBanerovDataSet);

}

private void Delete\_Click(object sender, EventArgs e)

{

zakazyBindingSource.RemoveCurrent();

}

private void Add\_Click(object sender, EventArgs e)

{

zakazyBindingSource.AddNew();

}

private void Previous\_Click(object sender, EventArgs e)

{

zakazyBindingSource.MovePrevious();

}

private void Last\_Click(object sender, EventArgs e)

{

zakazyBindingSource.MoveLast();

}

private void Next\_Click(object sender, EventArgs e)

{

zakazyBindingSource.MoveNext();

}

private void Back\_Click(object sender, EventArgs e)

{

Menu frm = new Menu();

frm.Show();

this.Hide();

}

}

}

**3.3. Руководство пользователя**

**1. ВВЕДЕНИЕ**

**1.1. Область применения**

Данное руководство предназначено для использования при:

1. Организации производства баннеров.
2. Работа с программным обеспечением для дизайна и печати баннеров.

3. Обучении сотрудников, вовлеченных в процесс производства.

**1.2. Краткое описание возможностей**

Производство баннеров включает в себя процессы, связанные с проектированием, печатью, отделкой и монтажом баннеров. Основные возможности системы:

1. Создание дизайна баннеров.
2. Печать баннеров различных форматов.
3. Отделка и монтаж баннеров в зависимости от требований заказчика.

**1.3. Уровень подготовки пользователя**

Пользователь должен иметь базовые навыки работы с графическими редакторами, знать основы проектирования и печати, а также разбираться в материалах, используемых для изготовления баннеров.

**1.4. Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю**

* 1. Общая информация о производстве баннеров.
  2. Технические характеристики и выбор материалов.
  3. Руководство пользователя по графическим редакторам.
  4. Методические рекомендации по подготовке макетов и печати.

**2. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ СИСТЕМЫ ПРОИЗВОДСТВО БАННЕРОВ**

Система производства баннеров предназначена для автоматизации процессов проектирования, печати и монтажа баннеров, а также управления заказами и клиентскими взаимодействиями.

1. Хранение информации о клиентах, заказах, материалах.
2. Регистрация дизайнов и технических заданий.
3. Мониторинг статуса заказов.

**Условия применения системы «Производство баннеров»**

Работа с системой возможна при соблюдении следующих условий:

1. Аппаратные и программные требования:

Программное обеспечение для работы с графикой (Adobe Illustrator, CorelDRAW и др.).

Система должна поддерживать графический дизайн и печать на широкоформатных принтерах.

1. Операционная система: Windows 10/11 или аналогичная.
2. Операционная среда:

Наличие установленного драйвера принтера.

Программное обеспечение для управления печатью и резкой.

1. Технические средства:

Широкоформатный принтер и принтеры для отделки.

Оборудование для монтажа баннеров.

**3. ПОДГОТОВКА К РАБОТЕ**

**3.1. Состав и содержание дистрибутивного носителя данных**

Для работы с программой «Производство баннеров» необходимо:

1. Программное обеспечение (Adobe Illustrator, CorelDRAW).
2. Шаблоны для проектирования баннеров.
3. Электронные материалы и графика.

**3.2. Порядок загрузки данных и программ**

Перед началом работы выполните следующие шаги:

1. Убедитесь, что все необходимые программы и драйвера установлены на компьютере.
2. Откройте графический редактор и выберите шаблон для создания баннера.
3. Введите требуемую информацию для авторизации в системе (если требуется).
4. Убедитесь, что настройки принтера и параметры печати соответствуют требованиям передачи данных.

**3.3. Порядок проверки работоспособности**

Чтобы проверить доступность программы и выполнения заказа, выполните:

1. Откройте приложение (графический редактор).
2. Создайте тестовый проект или откройте готовый.
3. Убедитесь, что программа правильно работает и отображает необходимые элементы.
4. Проверьте параметры печати, предварительно просматривая макет.

**4. ОПИСАНИЕ ОПЕРАЦИЙ**

**4.1. Создание дизайна баннера**

Условия выполнения: Доступ к приложению и наличие шаблона.

Подготовительные действия:

1. Вход в приложение.
2. Сбор информации о размере и дизайне баннера.

Основные действия:

1. Нажать «Создать новый проект».
2. Выбрать подходящий шаблон.
3. Заполнить необходимые данные и добавить графику.
4. Нажать кнопку «Сохранить».

Заключительные действия:

1. Просмотреть созданный макет.
2. При необходимости отредактировать дизайн и сохранить изменения.

**4.2. Подготовка к печати**

Условия выполнения: Готовый макет и доступ к принтеру.

Подготовительные действия:

1. Вход в приложение.
2. Открыть форму печати.

Основные действия:

1. Указать параметры печати (формат, плотность).
2. Нажать кнопку «Печать» и проверить, идет ли печать.

**5. АВАРИЙНЫЕ СИТУАЦИИ**

Сбой приложения (краш, фатальная ошибка):

Причины: Ошибки в коде, перегрузка системы.

Устранение: Просмотр логов приложения, оптимизация.

Проблемы с производительностью:

Причины: Долгое время обработки печати.

Устранение: Оптимизация шаблонов и графики.

Ошибки пользователей:

Причины: Некорректный ввод данных при создании макета.

Устранение: Валидация графики и подсказки.

**6. РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ**

**6.1. Общие рекомендации**

* 1. Ознакомьтесь с интерфейсом приложения.
  2. Создайте тестовые дизайны для практики.

**6.2. Контрольный пример**

Создайте несколько макетов баннеров, заполнив их различной графикой и текстом.

**6.3. Правила запуска и выполнения**

Запуск приложения:

1. Запустите графический редактор.
2. Введите свои данные для входа, если требуется.

Выполнение операций:

1. Следуйте интуитивно понятному интерфейсу.
2. Проверяйте данные на актуальность, чтобы избежать ошибок при печати.

# ГЛАВА 4. РАЗРАБОТКА 1С

# . Разработка подсистем

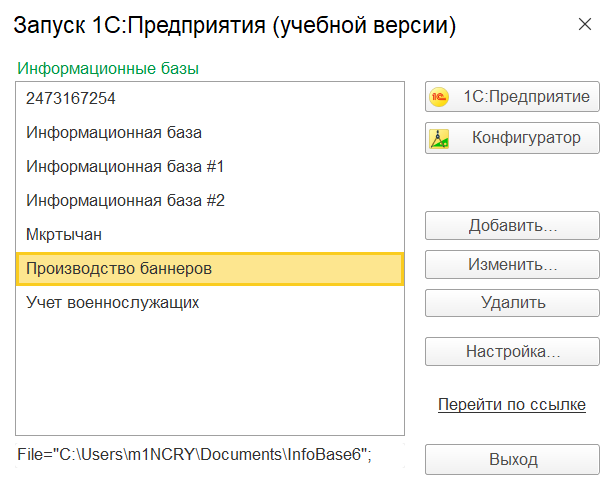
****

Рис. 39 Информационная база

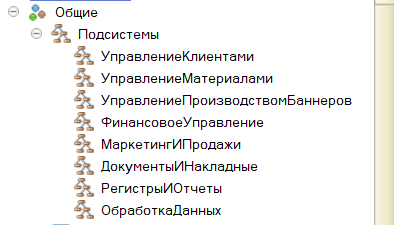
****

Рис. 40 Список подсистем

* 1. **Разработка справочников**

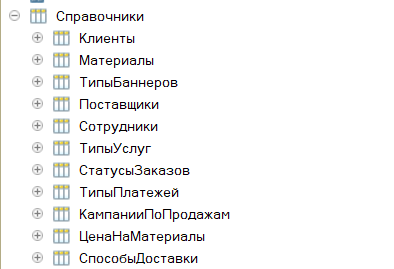
****

Рис. 41 Создание справочников

* 1. **Разработка документов и регистров**

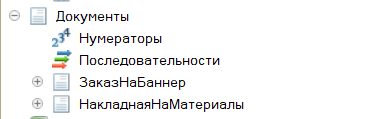
****

Рис. 42 Создание документов

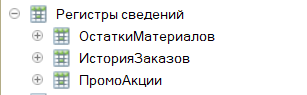
****

Рис. 43 Создание регистров сведений

* 1. **Разработка отчетов**

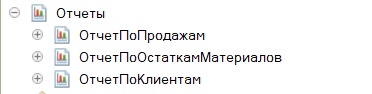
****

Рис. 44 Создание отчётов

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Начало работы с SQL - <https://cloud.vk.com/databases/mysql>
2. SQL для начинающих - <https://htmlacademy.ru/blog/php/sql>
3. Создания приложения - <https://learn.microsoft.com/ru-ru/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022>
4. Отчеты в 1С - <https://www.1s-up.ru/otchety-v-1s/>
5. Регистр сведений - <https://www.1s-up.ru/registry-svedenij-1c/>
6. Тихомиров А. В., Кузнецов С. И. Основы SQL и реляционных баз данных. - М.: Издательство БХВ-Петербург, 2021. - 320 с.
7. Иванова Е. Р., Сидоров П. Н. Проектирование баз данных с использованием SQL. - СПб.: Питер, 2020. - 450 с
8. Тихомиров А. В., Кузнецов С. И. Основы SQL и реляционных баз данных. - М.: Издательство БХВ-Петербург, 2021. - 320 с.
9. Иванова Е. Р., Сидоров П. Н. Проектирование баз данных с использованием SQL. - СПб.: Питер, 2020. - 450 с