**Содержание**

ВВЕДЕНИЕ ………………………………………………………………….........4 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ………..5 2 ПРОЕКТИРОВАНИЕ…………………………………………………………...9

2. 1 Определение основных сущностей предметной области…………...9

2. 2 Построение схемы реляционной базы данных……………………..11

2. 3 Описание физической модели данных……………………………...14

2. 4 Эскиз интерфейса…………………………………………………….17 3 ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ……………………………25

3. 1 Описание структуры базы данных………………………………….25

3. 2 Описание задач автоматизации и интерфейса пользователя……...27 ВЫВОДЫ………………………………………………………………………...38 ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ……………………………...39

**ВВЕДЕНИЕ**

**База данных** – это совокупность связанных данных, отражающая состояние объектов и взаимосвязь между собой. Создание базы данных, её поддержка и обеспечение доступа пользователей к ней осуществляется с помощью специального программного инструмента – система управления базами данных (СУБД).

**СУБД** – это комплекс языковых и программных средств, необходимых для создания, обработки и коллективного использования баз данных.

**Основные функции СУБД**:

1.Непосредственное управление данными во внешней памяти;

2.Управление буферами оперативной памяти;

3.Управление транзакциями;

4.Журнализация;

5.Поддержкаязыков БД.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Создание базы данных для кампании, занимающейся дизайном интерьера-это очень удобное решение для оптимизации работы и управления заведением.

В базе данных необходимо обеспечить хранение и обработку информации о следующих объектах: сотрудники, клиенты, объекты, заказы, поставка товара.

Приложение должно облегчить процесс формирования заказов и их корректировку , в случае необходимости.

Разрабатываемой системе необходимо иметь ряд второстепенных функций для нормального функционирования организации, а именно :простое добавление информации о сотрудниках, возможность контролировать наличие объектов и расчет прибыли.

В базе данных будет хранится информация о следующих объектах:

1. Клиент (Заказчик):
   1. Фамилия: текст 20 символов.
   2. Имя: текст 20 символов.
   3. Отчество: текст 20 символов.
   4. Адрес: текст 150 символов.
   5. Телефон: текст не более 10 символов, уникальное поле.
2. Сотрудники:
   1. Фамилия: текст 20 символов.
   2. Имя: текст 20 символов.
   3. Отчество: текст 20 символов.
   4. Дата рождения: дата в числовом формате, не может быть меньше 18 лет(разность сегодняшней даты и даты рождения).
   5. Телефон: текст не более 10 символов, уникальное поле.
   6. Адрес: текст 150 символов.
3. Покупки (Закупка на склад):
   1. Код сотрудника: целое число.
   2. Дата: дата в числовом формате.
4. Список покупок:
   1. Код объекта: целое число.
   2. Количество: целое число, положительное.
   3. Код покупки: целое число.
5. Объект:
   1. Название: текст 30 символов.
   2. Код типа: целое число.
   3. Цена: вещественное число, положительное, 8 цифр до запятой и 2 после.
   4. Количество на складе: целое число, положительное.
   5. Описание: текст 300 символов.
6. Заказ:
   1. Код клиента: целое число.
   2. Код сотрудника: целое число.
   3. Дата принятия: дата в числовом формате.
   4. Дата сдачи: дата в числовом формате.
   5. Статус: выбор из: Принят, Ведутся работы, Готов к сдаче, Сдан
   6. Код шаблона: целое число.
   7. Цена: вещественное число, положительное.
7. Шаблоны:
   1. Название: текст (100).
   2. Описание: текст (300).
8. Список шаблонов:
   1. Код объекта: целое число.
   2. Количество: целое число, положительное.
   3. Код шаблона: целое число.

В приложение должны быть реализованы следующие функции:

1. Формирование списка заказов на указанную дату
2. Формирование списка не выполненных заказов по сотруднику, дате, клиенту
3. Поиск шаблонов дизайна
4. Формирование финансовых отчётов
5. Формирование договора о предоставлении услуг
6. Сортировка объектов, заказов, сотрудников по различным параметрам: дата, кол-во, название и т. д.
7. Поиск объектов по различным фильтрам: цена, название, тип и т.д.
8. Контроль за правильностью вводимых данных
9. Возможность добавления объектов, типов объектов, сотрудников и клиентов.
10. Возможность добавления и изменения информации о объектах, типах объектов, сотрудниках и клиентах.
11. Формирование списков сотрудников, объектов , заказов

В программе необходимо предусмотреть следующее:

* Контроль за вводимыми данными
* Невозможно оформить заказ (покупку) датой, которая больше сегодняшней.
* Невозможно принять на работу сотрудника, возраст которого менее 18 лет
* Шаблон интерьера должен состоять минимум из 2-х объектов
* Организация скидки постоянным клиентам
* Организация возможности заказа на интерьер сразу нескольких помещений

В программе на печать должна отправляться следующая информация:

1. Отчётная документация о доходах и статистике
2. Документ о предоставлении услуг

Строки в таблицах будут приблизительно в таком количестве: сотрудники – 15, клиенты – 30, объекты – 50, шаблоны интерьера – 5-7, количество строк может редактироваться со временем.

**2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

**2.1 Построение ER-диаграммы.**

На основе выполненного анализа предметной области была построена ER-диаграмма (рисунок 2.1).

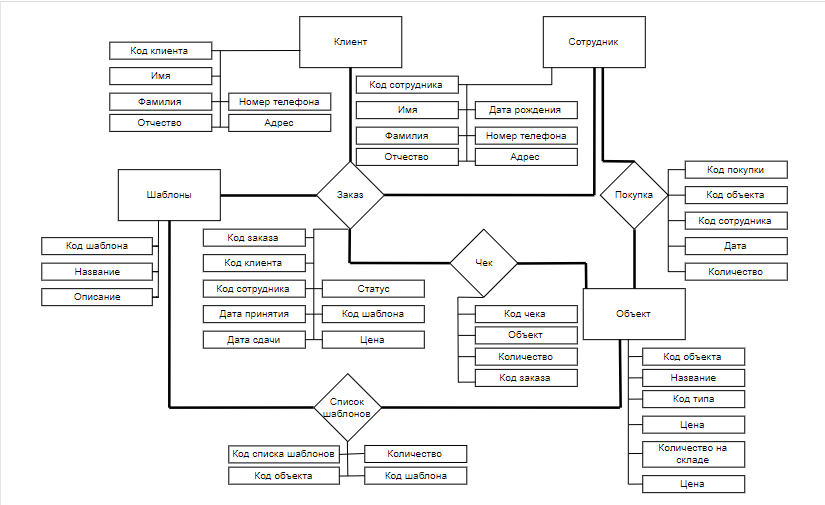


Рисунок 2.1- ER-диаграмма

**2.2 Построение схемы реляционной модели базы данных.**

На основании проведенного анализа предметной области и построенной ER- диаграммы создана реляционная модель базы данных (рисунок 2.2), которая отвечает всем требованиям нормализации отношений.

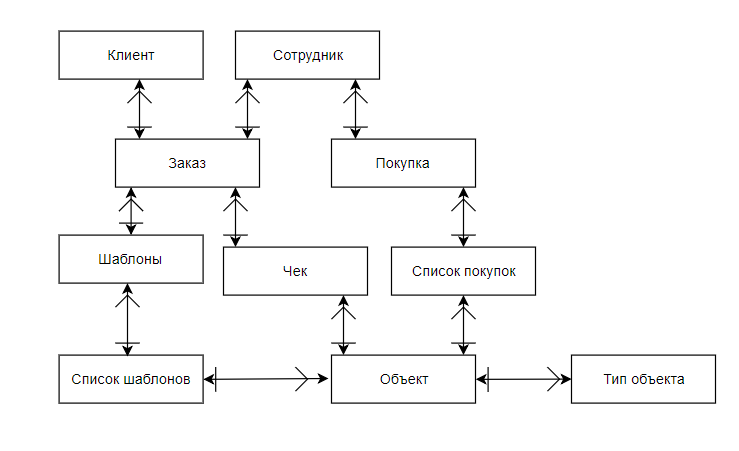


Рисунок 2.2 –Реляционная модель БД

Спроектированные отношения, представленные в таблицах 2.1-2.10

Таблица 2.1- Отношение ОБЪЕКТЫ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код объекта | Название | Код типа | Цена | Кол-во на складе | Описание |
|  |  |  |  |  |  |

Таблица 2.2 – Отношение ЗАКАЗ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Код заказа | Код клиента | Код сотрудника | Дата принятия | Дата сдачи | Статус | Код шаблона | Цена |
|  |  |  |  |  |  |  |  |

Таблица 2.3 – Отношение КЛИЕНТ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код клиента | Фамилия | Имя | Отчество | Адрес | Номер телефона |
|  |  |  |  |  |  |

Таблица 2.4 – Отношение ТИП ОБЪЕКТА

|  |  |
| --- | --- |
| Код типа | Название типа |
|  |  |

Таблица 2.5 – Отношение ЧЕК

|  |  |  |  |
| --- | --- | --- | --- |
| Код чека | Код объекта | Кол-во | Код заказа |
|  |  |  |  |

Таблица 2.6 – Отношение СПИСОК ШАБЛОНОВ

|  |  |  |  |
| --- | --- | --- | --- |
| Код списка шаблонов | Код объекта | Кол-во | Код шаблона |
|  |  |  |  |

Таблица 2.7 – Отношение ШАБЛОНЫ

|  |  |  |
| --- | --- | --- |
| Код шаблона | Название | Описание |
|  |  |  |

Таблица 2.8 – Отношение ПОКУПКИ (На склад)

|  |  |  |
| --- | --- | --- |
| Код покупки | Код сотрудника | Дата |
|  |  |  |

Таблица 2.9 – Отношение СПИСОК ПОКУПОК (На склад)

|  |  |  |  |
| --- | --- | --- | --- |
| Код списка покупок | Код объекта | Кол-во | Код покупки |
|  |  |  |  |

Таблица 2.10 – Отношение СОТРУДНИКИ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Код сотрудника | Фамилия | Имя | Отчество | Дата рождения | Номер телефона | Адрес |
|  |  |  |  |  |  |  |

**2.3 Описание физической модели базы данных.**

База данных будет разрабатываться в СУБД MySQL.

С учетом особенностей выбранной СУБД спроектирована физическая модель для созданной ранее реляционной модели (рисунок 2.2). Физические модели БД представленной в таблицах 2.11-2.20

Таблица 2.11-Физическая модель таблицы «Объект (objects)»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип | Ограничения | Имя |
| Id | Int | PK,AI,NN | Код объекта |
| Имя столбца | Тип | Ограничения | Имя |
| Names | Varchar(40) | NN | Название |
| Type | Int | FK,NN | Код типа |
| Cost | Decimal(8,2) | NN, US | Цена |
| Count | Int | NN,US | Кол-во на складе |
| Description | Varchar(300) | -//- | Описание |

Таблица 2.12 – Физическая модель таблицы «Заказ (orders)»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип | Ограничения | Имя |
| Id | Int | PK,AI,NN | Код заказа |
| Customer | Int | FK,NN | Код клиента |
| Employee | Int | KK,NN | Код сотрудника |
| DateOrder | Date | NN | Дата принятия |
| DateGive | Date | NN | Дата сдачи |
| Stattus | Enum (Принят, Ведутся работы, Готов к сдаче, Сдан) | NN | Статус |
| Shablon | Int | FK,NN | Код шаблона |
| Cost | Decimal(8,2) | NN | Цена |

Таблица 2.13 – Физическая модель таблицы «Клиент (customer)»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип | Ограничения | Имя |
| Id | Int | PK,AI,NN | Код клиента |
| Fname | Varchar(30) | NN | Фамилия |
| Lname | Varchar(30) | NN | Имя |
| Sname | Varchar(30) | NN | Отчество |
| Adres | Varchar(100) | NN | Адрес |
| Phone | Int | NN | Номер телефона |

Таблица 2.14 – Физическая модель таблицы «Тип объекта (objecttype)»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип | Ограничения | Имя |
| Id | Int | PK,AI,NN | Код типа |
| Names | Varchar(40) | NN | Название типа |

Таблица 2.15 – Физическая модель таблицы «Чек (checks)»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип | Ограничения | Имя |
| Id | Int | PK,AI,NN | Код чека |
| Object | Int | FK,NN | Код объекта |
| Count | Int | NN,US | Кол-во |
| Order | Int | FK,NN | Код заказа |

Таблица 2.16 – Физическая модель таблицы «Список шаблонов (shablonlist)»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип | Ограничения | Имя |
| Id | Int | PK,AI,NN | Код списка шаблонов |
| Objects | Int | FK,NN | Код объекта |
| Count | Int | NN,US | Кол-во |
| Shablon | Int | FK,NN | Код шаблона |

Таблица 2.17 – Физическая модель таблицы «Шаблоны (shablons)»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип | Ограничения | Имя |
| Id | Int | PK,AI,NN | Код шаблона |
| Names | Varchar(40) | NN | Название |
| Description | Varchar(300) | -//- | Описание |

Таблица 2.18 – Физическая модель таблицы «Покупки (buy)»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип | Ограничения | Имя |
| Id | Int | PK,AI,NN | Код покупки |
| Employee | Int | FK,NN | Код сотрудника |
| Date | Date | NN | Дата |

Таблица 2.19 – Физическая модель таблицы «Список покупок (buylist)»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип | Ограничения | Имя |
| Id | Int | PK,AI,NN | Код списка покупок |
| Objects | Int | FK,NN | Код объекта |
| Counts | Int | NN,US | Кол-во |
| Buy | Int | FK,NN | Код покупки |

Таблица 2.20 – Физическая модель таблицы «Сотрудники (buylist)»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип | Ограничения | Имя |
| Id | Int | PK,AI,NN | Код сотрудника |
| Fname | Varchar(30) | NN | Фамилия |
| Lname | Varchar(30) | NN | Имя |
| Sname | Varchar(30) | NN | Отчество |
| Born | Date | NN | Дата рождения |
| Phone | Int | NN | Номер телефона |
| Adres | Varchar(100) | NN | Адрес |

**2. 4 Эскиз интерфейса**

Эскиз интерфейса - это визуальный набросок будущего приложения. Он дает возможность сгруппировать объекты и правильно распределить функции. От удобства интерфейса зависит то, насколько клиент «привяжется» к системе, насколько ему удобно будет работать.

Интерфейс приложения для работы с БД будет разработан в виде форм с кнопками, полями и переключателями. Далее на рисунках 2.3 - 2.7 представлены эскизы окон, с которыми будет работать пользователь во время эксплуатации приложения.

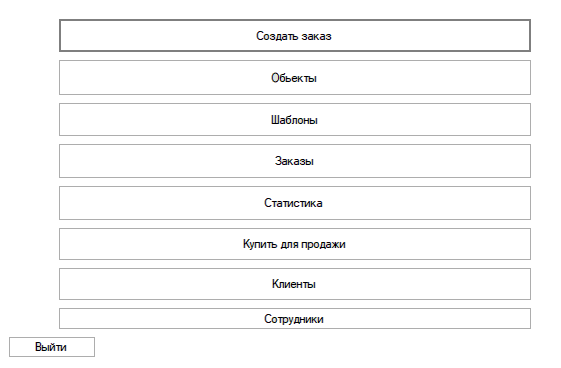


Рисунок 2.3 – Эскиз интерфейса «Меню»

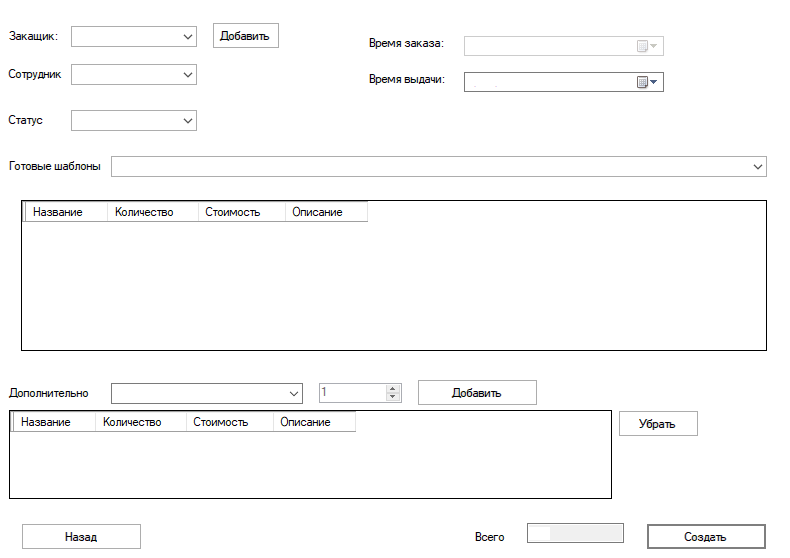


Рисунок 2.4 – Эскиз интерфейса «Создать заказ»

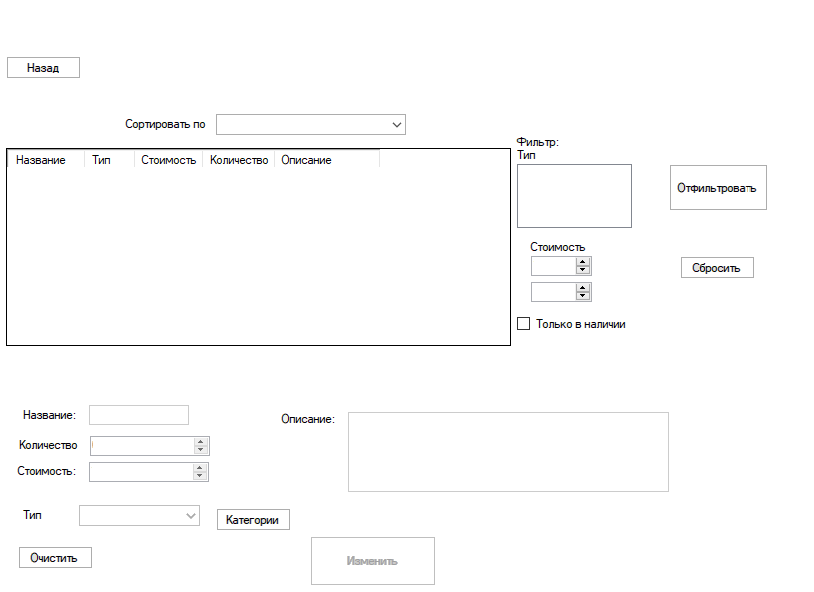


Рисунок 2.5 – Эскиз интерфейса «Объекты»



Рисунок 2.6 – Эскиз интерфейса «Шаблоны»

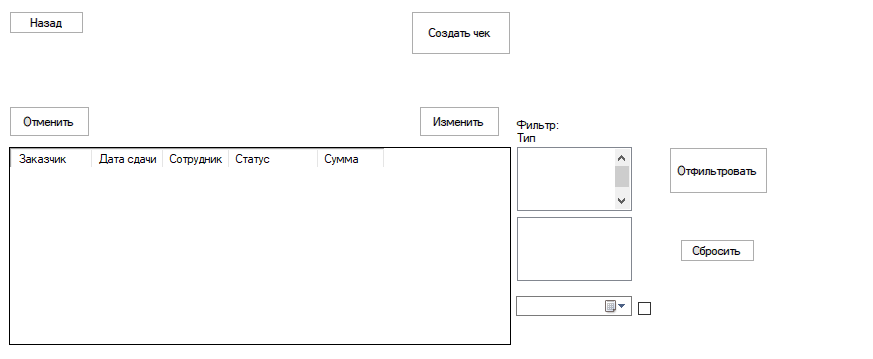


Рисунок 2.7 – Эскиз интерфейса «Заказы»

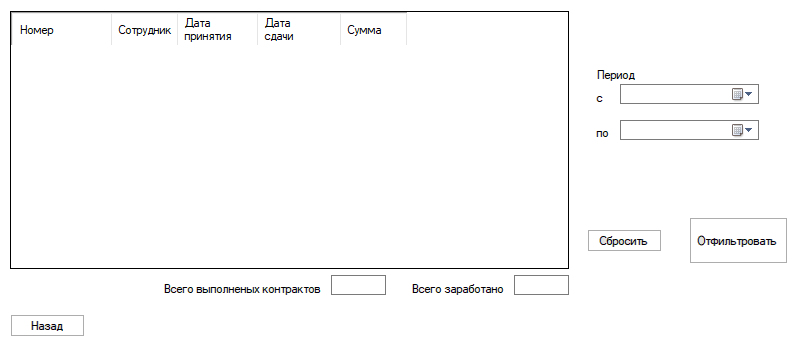


Рисунок 2.8 – Эскиз интерфейса «Статистика»



Рисунок 2.9 – Эскиз интерфейса «Закупка»

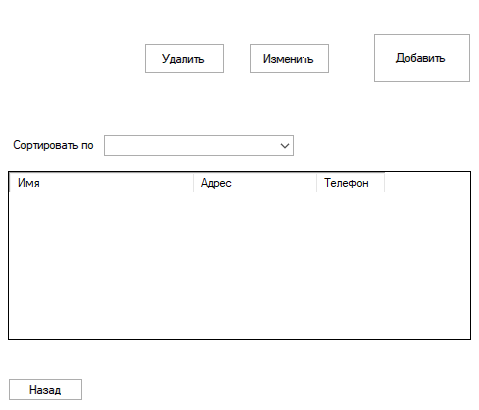


Рисунок 2.10 – Эскиз интерфейса «Клиенты»

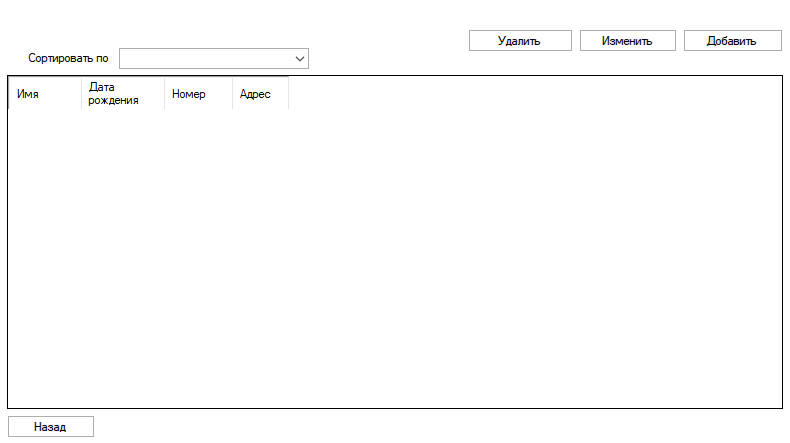


Рисунок 2.11 – Эскиз интерфейса «Сотрудники»

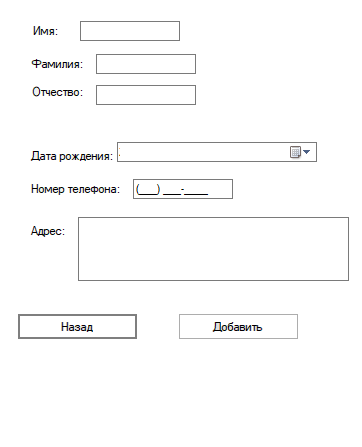


Рисунок 2.12 – Эскиз интерфейса «Добавить сотрудника»

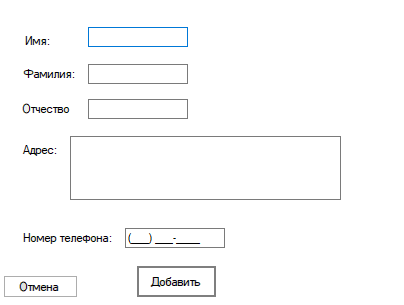


Рисунок 2.13 – Эскиз интерфейса «Добавить клиента»

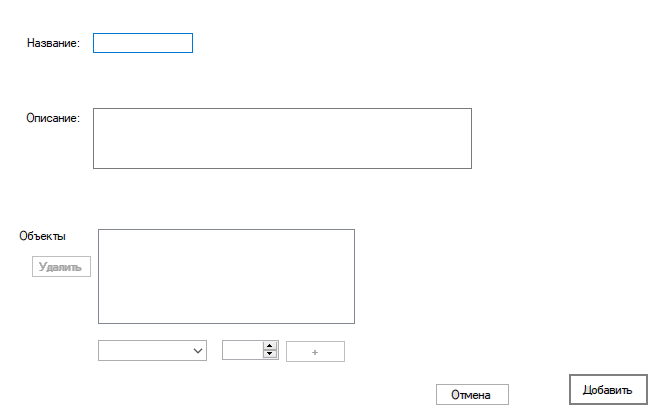


Рисунок 2.14 – Эскиз интерфейса «Добавить шаблон»

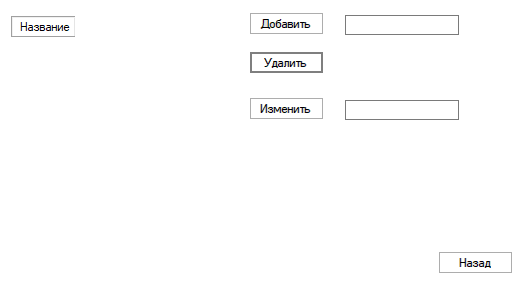


Рисунок 2.15 – Эскиз интерфейса «Категории»

**3 ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ**

**3. 1 Описание структуры базы данных**

CREATE DATABASE IF NOT EXISTS `interior` /\*!40100 DEFAULT CHARACTER SET utf8 \*/;

USE `interior`;

-- MySQL dump 10.13 Distrib 8.0.13, for Win64 (x86\_64)

-- Host: 127.0.0.1 Database: interior

-- Server version 8.0.13

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

SET NAMES utf8 ;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

-- Table structure for table `buy`

DROP TABLE IF EXISTS `buy`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

SET character\_set\_client = utf8mb4 ;

CREATE TABLE `buy` (

`Id` int(11) NOT NULL AUTO\_INCREMENT,

`Employeer` int(11) NOT NULL,

`Date` date NOT NULL,

`Enable` int(11) NOT NULL DEFAULT '1',

PRIMARY KEY (`Id`),

KEY `EmployeerBuy\_idx` (`Employeer`),

CONSTRAINT `EmployeerBuy` FOREIGN KEY (`Employeer`) REFERENCES `employeers` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=27 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

-- Dumping data for table `buy`

LOCK TABLES `buy` WRITE;

/\*!40000 ALTER TABLE `buy` DISABLE KEYS \*/;

INSERT INTO `buy` VALUES (1,1,'2019-02-23',1),(2,1,'2019-02-23',1),(3,1,'2019-02-23',1),(4,1,'2019-02-23',1),(5,1,'2019-02-23',1),(6,1,'2019-02-23',1),(7,1,'2019-02-23',1),(8,1,'2019-02-23',1),(9,1,'2019-02-23',1),(10,1,'2019-02-24',1),(11,1,'2019-03-01',1),(12,2,'2019-03-01',1),(13,1,'2019-03-01',1),(14,2,'2019-03-01',1),(15,1,'2019-03-01',1),(16,1,'2019-03-01',1),(17,2,'2019-03-01',1),(18,1,'2019-03-01',1),(19,1,'2019-03-01',1),(20,2,'2019-03-01',1),(21,1,'2019-03-01',1),(22,2,'2019-03-01',1),(23,1,'2019-03-01',1),(24,1,'2019-03-04',1),(25,1,'2019-03-04',1),(26,1,'2019-03-04',1);

/\*!40000 ALTER TABLE `buy` ENABLE KEYS \*/;

UNLOCK TABLES;

-- Table structure for table `buylist`

DROP TABLE IF EXISTS `buylist`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

SET character\_set\_client = utf8mb4 ;

CREATE TABLE `buylist` (

`Id` int(11) NOT NULL AUTO\_INCREMENT,

`Object` int(11) NOT NULL,

`Counts` int(11) NOT NULL,

`Buy` int(11) NOT NULL,

PRIMARY KEY (`Id`),

KEY `Object\_idx` (`Object`),

KEY `Buy\_idx` (`Buy`),

CONSTRAINT `Buy` FOREIGN KEY (`Buy`) REFERENCES `buy` (`id`),

CONSTRAINT `Object` FOREIGN KEY (`Object`) REFERENCES `objects` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=14 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

-- Dumping data for table `buylist`

LOCK TABLES `buylist` WRITE;

/\*!40000 ALTER TABLE `buylist` DISABLE KEYS \*/;

INSERT INTO `buylist` VALUES (1,1,3,9),(2,2,2,14),(3,2,3,17),(4,1,3,18),(5,1,2,19),(6,2,3,20),(7,2,1,20),(8,1,2,21),(9,2,5,22),(10,1,4,23),(11,1,7,24),(12,1,10,25),(13,1,1,26);

/\*!40000 ALTER TABLE `buylist` ENABLE KEYS \*/;

UNLOCK TABLES;

-- Table structure for table `checks`

DROP TABLE IF EXISTS `checks`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

SET character\_set\_client = utf8mb4 ;

CREATE TABLE `checks` (

`Id` int(11) NOT NULL AUTO\_INCREMENT,

`Object` int(11) NOT NULL,

`Count` int(11) NOT NULL,

`Order` int(11) NOT NULL,

PRIMARY KEY (`Id`),

KEY `Object\_idx` (`Object`),

KEY `Order\_idx` (`Order`),

CONSTRAINT `ObjectOrder` FOREIGN KEY (`Object`) REFERENCES `objects` (`id`),

CONSTRAINT `Order` FOREIGN KEY (`Order`) REFERENCES `orders` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

-- Dumping data for table `checks`

LOCK TABLES `checks` WRITE;

/\*!40000 ALTER TABLE `checks` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `checks` ENABLE KEYS \*/;

UNLOCK TABLES;

-- Table structure for table `customer`

DROP TABLE IF EXISTS `customer`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

SET character\_set\_client = utf8mb4 ;

CREATE TABLE `customer` (

`Id` int(11) NOT NULL AUTO\_INCREMENT,

`Fname` varchar(45) NOT NULL,

`Lname` varchar(45) NOT NULL,

`Sname` varchar(45) NOT NULL,

`Adres` varchar(150) NOT NULL,

`Phone` varchar(15) NOT NULL,

`Enable` int(11) NOT NULL DEFAULT '1',

PRIMARY KEY (`Id`),

UNIQUE KEY `Phone\_UNIQUE` (`Phone`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

-- Dumping data for table `customer`

LOCK TABLES `customer` WRITE;

/\*!40000 ALTER TABLE `customer` DISABLE KEYS \*/;

INSERT INTO `customer` VALUES (1,'Виталий','Слива','Витальевич','Купянск','3121312321',1),(2,'Славик','Лиман','Витальевич','Салтовка возле метро','1111134234',1),(3,'Егор','Демченко','Егорович','Салтовка','1111111111',1),(4,'Макси','Солодовников','Максимович','м. Героев труда','1234222443',1);

/\*!40000 ALTER TABLE `customer` ENABLE KEYS \*/;

UNLOCK TABLES;

-- Table structure for table `employeers`

DROP TABLE IF EXISTS `employeers`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

SET character\_set\_client = utf8mb4 ;

CREATE TABLE `employeers` (

`Id` int(11) NOT NULL AUTO\_INCREMENT,

`Fname` varchar(45) NOT NULL,

`Lname` varchar(45) NOT NULL,

`Sname` varchar(45) NOT NULL,

`Born` date NOT NULL,

`Phone` varchar(15) NOT NULL,

`Adres` varchar(150) NOT NULL,

`Enable` int(11) NOT NULL DEFAULT '1',

PRIMARY KEY (`Id`),

UNIQUE KEY `Phone\_UNIQUE` (`Phone`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

-- Dumping data for table `employeers`

LOCK TABLES `employeers` WRITE;

/\*!40000 ALTER TABLE `employeers` DISABLE KEYS \*/;

INSERT INTO `employeers` VALUES (1,'Влад','Влад','Сергеевич','2000-05-12','1111111111','Салтовка',1),(2,'Слива','Слива','Витальевич','2000-12-09','3445678976','Купянск',1);

/\*!40000 ALTER TABLE `employeers` ENABLE KEYS \*/;

UNLOCK TABLES;

-- Table structure for table `objects`

DROP TABLE IF EXISTS `objects`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

SET character\_set\_client = utf8mb4 ;

CREATE TABLE `objects` (

`Id` int(11) NOT NULL AUTO\_INCREMENT,

`Names` varchar(100) NOT NULL,

`Type` int(11) NOT NULL,

`Cost` int(11) NOT NULL,

`Count` int(11) NOT NULL,

`Description` varchar(200) DEFAULT NULL,

`Enable` int(11) NOT NULL DEFAULT '1',

PRIMARY KEY (`Id`),

KEY `ObjectType\_idx` (`Type`),

CONSTRAINT `ObjectType` FOREIGN KEY (`Type`) REFERENCES `objecttype` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

-- Dumping data for table `objects`

LOCK TABLES `objects` WRITE;

/\*!40000 ALTER TABLE `objects` DISABLE KEYS \*/;

INSERT INTO `objects` VALUES (1,'Диван',1,2001,0,'Белый, кожанный',1),(2,'Стол кухоный',2,300,9,'Дубовый, черный',1),(3,'Стол игровой',2,300,3,'Качественный',1),(4,'Кровать',3,300,9,'Удобная и стильна',1),(5,'Кресло',1,100,1,'Деревянное',1),(6,'Диван 2',1,300,4,'Компактные',1);

/\*!40000 ALTER TABLE `objects` ENABLE KEYS \*/;

UNLOCK TABLES;

-- Table structure for table `objecttype`

DROP TABLE IF EXISTS `objecttype`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

SET character\_set\_client = utf8mb4 ;

CREATE TABLE `objecttype` (

`Id` int(11) NOT NULL AUTO\_INCREMENT,

`Names` varchar(70) NOT NULL,

`Enable` int(11) NOT NULL DEFAULT '1',

PRIMARY KEY (`Id`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

-- Dumping data for table `objecttype`

LOCK TABLES `objecttype` WRITE;

/\*!40000 ALTER TABLE `objecttype` DISABLE KEYS \*/;

INSERT INTO `objecttype` VALUES (1,'Диваны',1),(2,'Столы',1),(3,'Кровати',1);

/\*!40000 ALTER TABLE `objecttype` ENABLE KEYS \*/;

UNLOCK TABLES;

-- Table structure for table `orders`

DROP TABLE IF EXISTS `orders`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

SET character\_set\_client = utf8mb4 ;

CREATE TABLE `orders` (

`Id` int(11) NOT NULL AUTO\_INCREMENT,

`Customer` int(11) NOT NULL,

`Employeer` int(11) NOT NULL,

`DateOrder` date NOT NULL,

`DateGive` date NOT NULL,

`Status` enum('Принят','Ведутся работы','Готов к сдаче','Сдан','Отменен') NOT NULL,

`Shablon` int(11) NOT NULL,

`Cost` int(11) NOT NULL,

`Enable` int(11) NOT NULL DEFAULT '1',

PRIMARY KEY (`Id`),

KEY `Customer\_idx` (`Customer`),

KEY `Employeer\_idx` (`Employeer`),

KEY `ShablonOrder\_idx` (`Shablon`),

CONSTRAINT `Customer` FOREIGN KEY (`Customer`) REFERENCES `customer` (`id`),

CONSTRAINT `Employeer` FOREIGN KEY (`Employeer`) REFERENCES `employeers` (`id`),

CONSTRAINT `ShablonOrder` FOREIGN KEY (`Shablon`) REFERENCES `shablons` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=10 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

-- Dumping data for table `orders`

LOCK TABLES `orders` WRITE;

/\*!40000 ALTER TABLE `orders` DISABLE KEYS \*/;

INSERT INTO `orders` VALUES (2,1,1,'2019-02-25','2019-02-25','Ведутся работы',4,2000,1),(5,4,1,'2019-03-01','2019-03-10','Принят',3,900,1),(6,1,1,'2019-03-04','2019-03-11','Принят',8,40020,1),(7,3,1,'2019-03-04','2019-03-11','Принят',1,600,1),(8,2,1,'2019-03-04','2019-03-11','Принят',8,20010,1),(9,1,1,'2019-03-04','2019-03-11','Принят',2,2876,1);

/\*!40000 ALTER TABLE `orders` ENABLE KEYS \*/;

UNLOCK TABLES;

-- Table structure for table `shablonlist`

DROP TABLE IF EXISTS `shablonlist`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

SET character\_set\_client = utf8mb4 ;

CREATE TABLE `shablonlist` (

`Id` int(11) NOT NULL AUTO\_INCREMENT,

`Objects` int(11) NOT NULL,

`Count` int(11) NOT NULL,

`Shablon` int(11) NOT NULL,

`Enable` int(11) NOT NULL DEFAULT '1',

PRIMARY KEY (`Id`),

KEY `Objects\_idx` (`Objects`),

KEY `Shablons\_idx` (`Shablon`),

CONSTRAINT `Objects` FOREIGN KEY (`Objects`) REFERENCES `objects` (`id`),

CONSTRAINT `Shablons` FOREIGN KEY (`Shablon`) REFERENCES `shablons` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=23 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

-- Dumping data for table `shablonlist`

LOCK TABLES `shablonlist` WRITE;

/\*!40000 ALTER TABLE `shablonlist` DISABLE KEYS \*/;

INSERT INTO `shablonlist` VALUES (8,1,1,4,1),(9,3,1,5,1),(10,1,1,5,1),(11,2,1,5,1),(15,3,1,3,1),(16,2,1,3,1),(17,1,1,2,1),(18,3,1,2,1),(20,1,10,8,1),(21,2,2,1,1),(22,3,1,1,1);

/\*!40000 ALTER TABLE `shablonlist` ENABLE KEYS \*/;

UNLOCK TABLES;

-- Table structure for table `shablons`

DROP TABLE IF EXISTS `shablons`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

SET character\_set\_client = utf8mb4 ;

CREATE TABLE `shablons` (

`Id` int(11) NOT NULL AUTO\_INCREMENT,

`Names` varchar(45) NOT NULL,

`Description` varchar(150) NOT NULL,

`Enable` int(11) NOT NULL DEFAULT '1',

PRIMARY KEY (`Id`)

) ENGINE=InnoDB AUTO\_INCREMENT=10 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

-- Dumping data for table `shablons`

LOCK TABLES `shablons` WRITE;

/\*!40000 ALTER TABLE `shablons` DISABLE KEYS \*/;

INSERT INTO `shablons` VALUES (1,'Кухня стандарт','Топовый выбор для мам и студентов',1),(2,'Спальня','Ну такое',1),(3,'Спальня 2','Отличная для детей',1),(4,'Заказ Номер2','Это специальный зкаказ2',0),(5,'Заказ Номер3','Это специальный зкаказ3',0),(6,'Заказ Номер4','Это специальный зкаказ4',0),(7,'Заказ Номер5','Это специальный зкаказ5',0),(8,'Тестовый','Проверка работоспособности',0),(9,'Заказ Номер8','Это специальный зкаказ8',0);

/\*!40000 ALTER TABLE `shablons` ENABLE KEYS \*/;

UNLOCK TABLES;

-- Dumping events for database 'interior'

-- Dumping routines for database 'interior'

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

**3. 2 Описание задач автоматизации и интерфейса пользователя**

Чтобы приступить к работе с приложение, для начала нужно запустить файл CourseWork. При открытии данного файла, появится окно Главной страницы (Меню), которая имеет следующий вид:

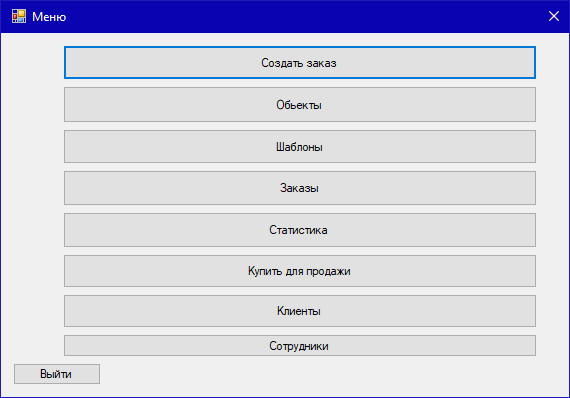


Рисунок 3.1 – Страница «Главная(Меню)»

Эта страница (Рисунок 3.1) содержит следующие компоненты:

- Пункт меню «Создать заказ», при переходе на который имеем следующее:

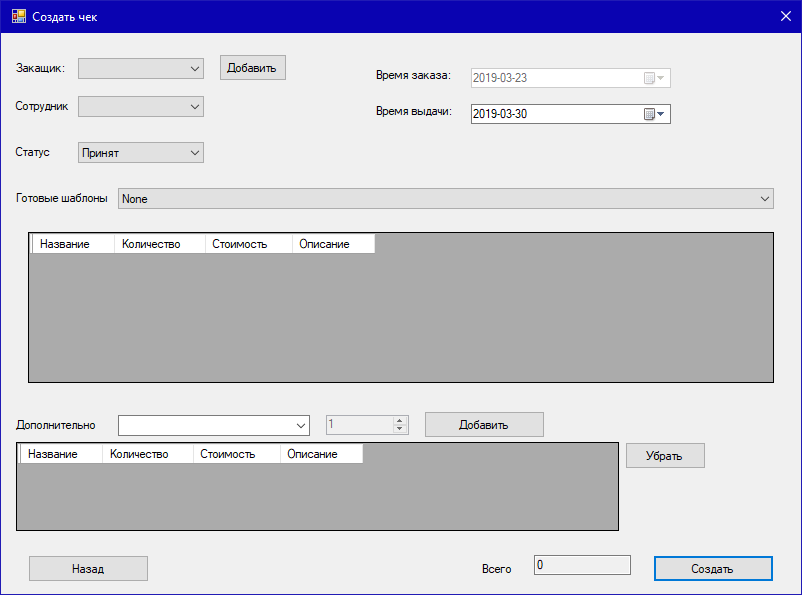


Рисунок 3.2 – Страница «Создать заказ»

На данной странице (Рисунок 3.2) в верхней части формы расположены поля для выбора заказчика, сотрудника, статуса , даты заказа и даты выдачи, а также возможность выбора готового шаблона. На данной форме расположено две области для вывода информации об объектах, которые входят в состав шаблона, и которые мы можем добавить в заказ дополнительно, с помощью поля для выбора объекта и его количества, при нажатии на кнопку «Добавить». Также с помощью кнопки «Убрать» возможно удалить лишние объекты, которые были заранее выделены. Внизу формы располагается поле для вывода общей суммы заказа, и 2 кнопки: «Назад», которая выполняет функцию возврата на предыдущую страницу и «Создать» - для создания заказа.

- Пункт меню «Объекты», при переходе на который имеем следующее:

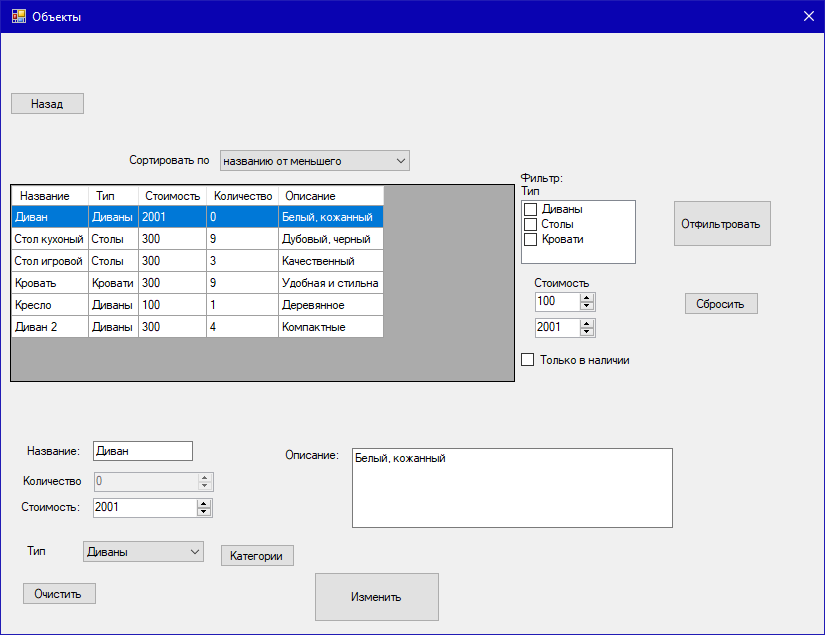


Рисунок 3.3 – Страница «Объекты»

На данной странице (Рисунок 3.3) расположены кнопки: «Назад» - для возврата в предыдущее окно; «Отфильтровать» и поля для фильтрации, необходимые для выбора различный объектов, по какому-либо условию, а также поле для сортировки; «Сбросить» - для сбрасывания фильтров; Поля для изменения выбранного объекта, кнопка «Очистить» для очистки полей изменения объекта, кнопка «Изменить», которая применяет внесённые изменения, и кнопка «Категория» - при нажатии на которую мы переходим на страницу «Добавить категорию» (Рисунок 3.4). А также поле для вывода информации об объектах.

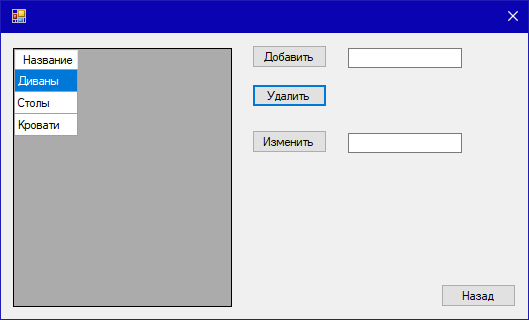


Рисунок 3.4 - Страница «Добавить категорию»

На данной странице (Рисунок 3.4) расположены 3 кнопки: «Добавить» и поле для ввода – необходимо для добавления новой категории; «Удалить» - для удаления выбранной категории; «Изменить» и поле для ввода – необходимы для изменения выбранной категории. На странице также расположено поле для вывода списка категорий.

- Пункт меню «Шаблоны», при переходе на который имеем следующее:

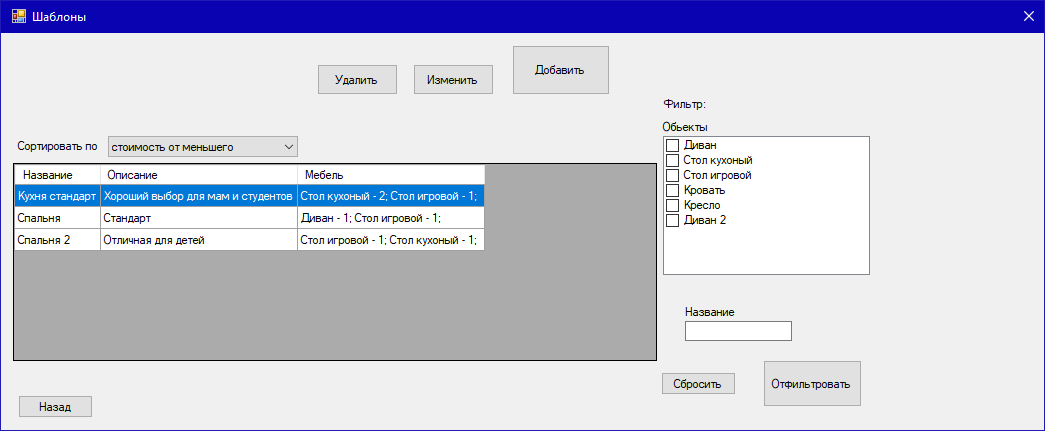


Рисунок 3.5 - Страница «Шаблоны»

На данной странице (Рисунок 3.5) расположены следующие компоненты: Поле для вывода информации о шаблонах, и компонентах, которые входят в состав шаблона; Поля для фильтрации данных; И кнопки: «Удалить» - для удаления выбранного шаблона, «Назад» - для перехода на предыдущую страницу, «Сбросить» - для сброса фильтров, «Отфильтровать» - для применения фильтров, «Изменить» - для изменения шаблона, при нажатии на которую мы попадаем страницу «Изменить шаблон» (Рисунок 3.6), «Создать» - для создания шаблона, при нажатии на которую мы попадаем на страницу «Добавить шаблон» (Рисунок 3.7).

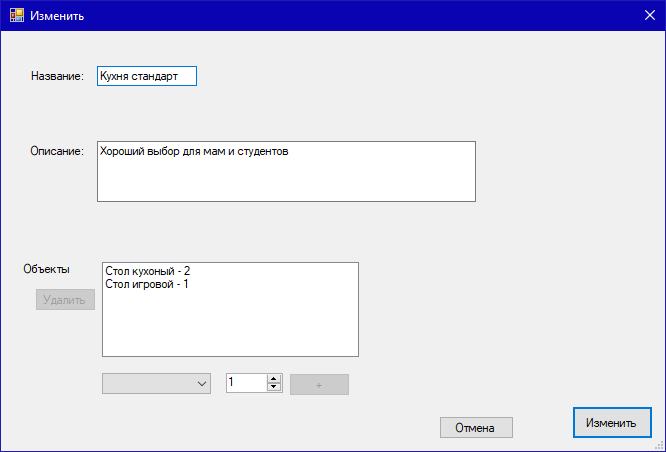


Рисунок 3.6 - Страница «Изменить шаблон»

На данной странице (Рисунок 3.6) находятся поля для ввода информации о шаблоне, и поля для добавления или удаления объектов в шаблон, а также кнопки «Отмена» и «Изменить» - необходимые для отмены изменения шаблона и применения изменения.

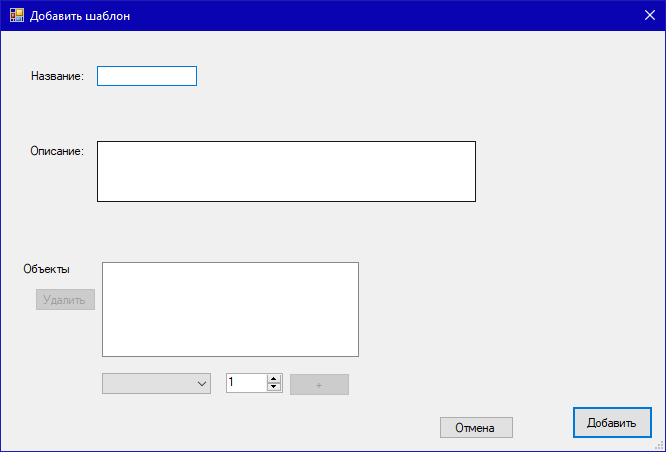


Рисунок 3.7 - Страница «Добавить шаблон»

На данной странице (Рисунок 3.7) находятся поля для ввода информации о шаблоне, и поля для добавления или удаления объектов в шаблон, а также кнопки «Отмена» и «Добавить» - необходимые для отмены добавления шаблона и добавления шаблона.

- Пункт меню «Заказы», при переходе на который имеем следующее:

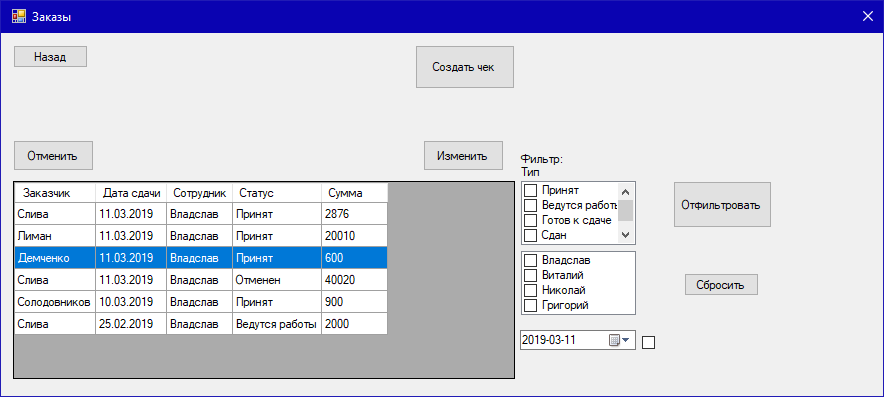


Рисунок 3.8 - Страница «Добавить шаблон»

На данной странице (Рисунок 3.8) расположены следующие компоненты: Поле для вывода информации о заказах; Поля для фильтрации данных с кнопками «Отфильтровать» и «Сбросить» - для применения и сброса фильтров соответственно; Кнопки «Назад» - для возврата на предыдущую страницу, «Создать чек» - для создания текстового файла, необходимого для печати чека, «Отменить» - для изменения статуса заказа на отменён, «Изменить» - необходима для изменения заказа, при нажатии на которую происходит переход на страницу «Изменить заказ» (Рисунок 3.9).

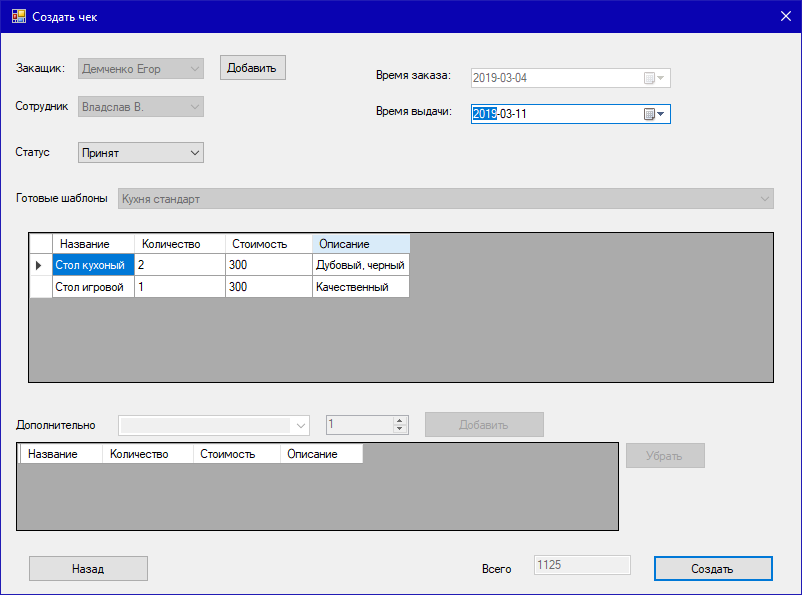


Рисунок 3.9 - Страница «Изменить заказ»

На данной странице все элементы расположены точно так же, как и на странице «Создать заказ» (Рисунок 3.2).

- Пункт меню «Статистика», при переходе на который имеем следующее:

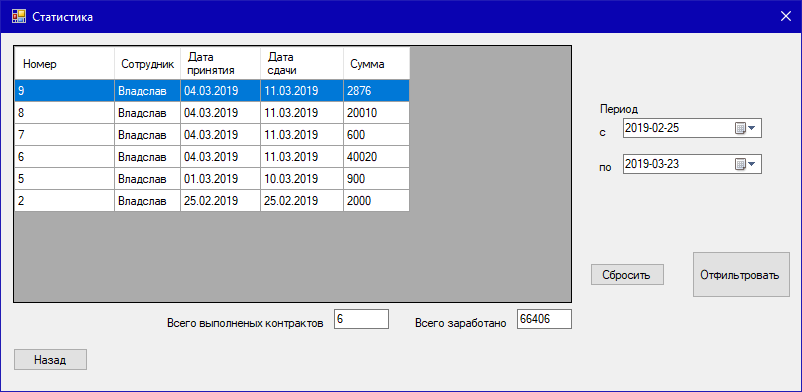


Рисунок 3.10 - Страница «Статистика»

На данной странице (Рисунок 3.10) расположены следующие компоненты: Поле для вывода статистики (Заработанные суммы за каждый заказ) и общей суммы заработка; Поля с фильтрами, необходимые для фильтрации данных; Кнопки: «Назад» - для возвращения на предыдущую страницу, «Сбросить» и «Отфильтровать» - необходимые для сброса и применения фильтров.

- Пункт меню «Закупка», при переходе на который имеем следующее:

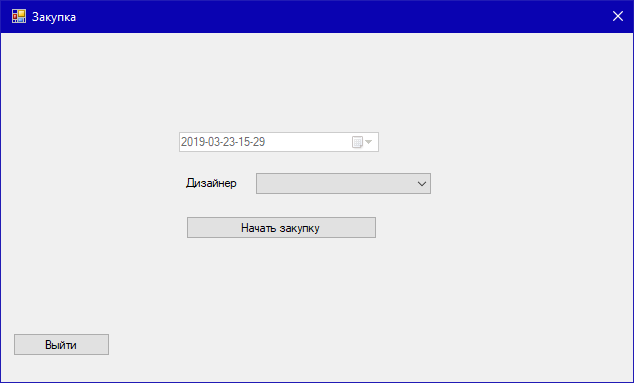


Рисунок 3.11 - Страница «Закупка»

На данной странице (Рисунок 3.11) расположены следующие компоненты: Выбор даты и дизайнера; Кнопки: «Выйти» - для возвращения в меню, «Начать закупку» - при нажатии на эту кнопку мы переходим на страницу «Закупка 2» (Рисунок 3.12)

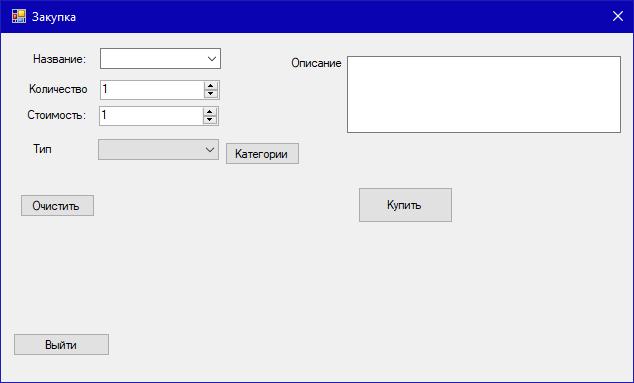


Рисунок 3.12 – Страница «Закупка 2»

Данная страница (Рисунок 3.12) содержит следующие компоненты: Поля для ввода информации об объекте: Кнопки: «Очистить»- для очистки полей ввода, «Купить» - для оформления заказа, «Выйти» - для возвращения на предыдущую страницу, «Категории» - при нажатии на которую происходит переход на страницу «Добавить категорию» (Рисунок 3.4).

- Пункт меню «Клиенты», при переходе на который имеем следующее:

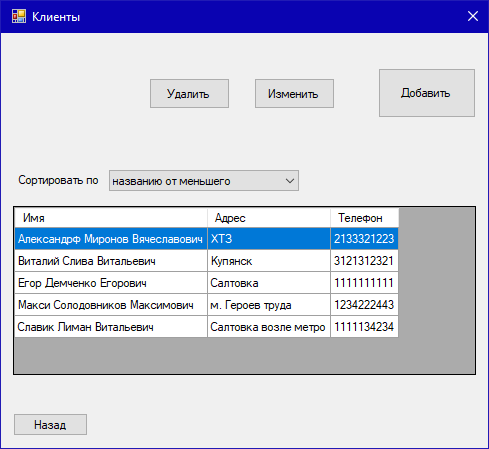


Рисунок 3.13 – Страница «Клиенты»

Данная страница (Рисунок 3.13) содержит следующие компоненты: Поле для вывода информации о клиентах; Поле для сортировки; Кнопки: «Назад» - для возвращения на предыдущую страницу, «Удалить» - для удаления выбранного клиента, «Изменить» - для изменения информации выбранного клиента, при нажатии на которую происходит переход на страницу «Изменить клиента» (Рисунок 3.14), «Добавить» - для добавления клиента, при нажатии на которую происходит переход на страницу «Добавить клиента» (Рисунок 3.15).

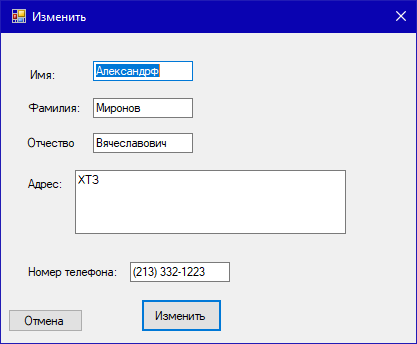


Рисунок 3.14 – Страница «Изменить клиента»

Данная страница (Рисунок 3.14) содержит следующие компоненты: Поля для изменения информации о клиенте; Кнопки: «Отмена» - необходимая для отмены изменения и возврата на предыдущую страницу, «Изменить» - для применения изменений.

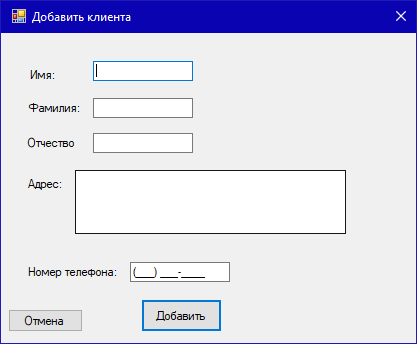


Рисунок 3.15 – Страница «Добавить клиента»

Данная страница (Рисунок 3.15) содержит следующие компоненты: Поля для ввода информации о клиенте; Кнопки: «Отмена» - необходимая для отмены добавления и возврата на предыдущую страницу, «Добавить» - для добавления.

- Пункт меню «Сотрудники», при переходе на который имеем следующее:

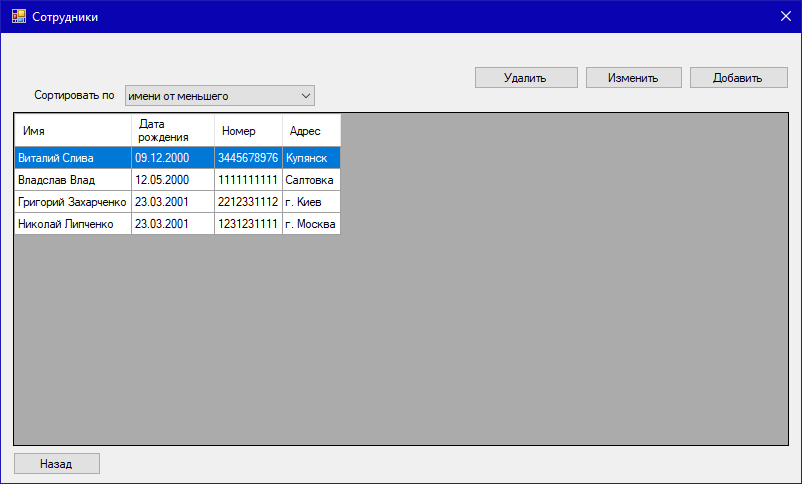


Рисунок 3.16 – Страница «Сотрудники»

Данная страница (Рисунок 3.13) содержит следующие компоненты: Поле для вывода информации о сотрудниках; Поле для сортировки; Кнопки: «Назад» - для возвращения на предыдущую страницу, «Удалить» - для удаления выбранного сотрудника, «Изменить» - для изменения информации выбранного сотрудника, при нажатии на которую происходит переход на страницу «Изменить сотрудника» (Рисунок 3.16), «Добавить» - для добавления сотрудника, при нажатии на которую происходит переход на страницу «Добавить сотрудника» (Рисунок 3.17).

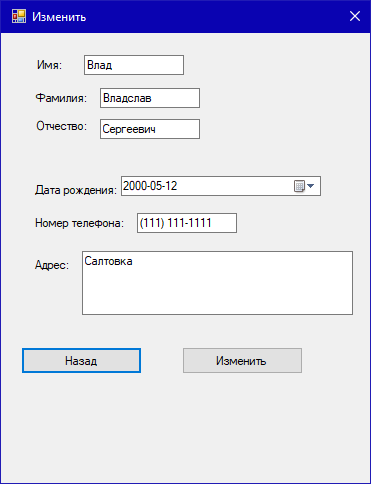


Рисунок 3.16 – Страница «Изменить сотрудника»

Данная страница (Рисунок 3.16) содержит следующие компоненты: Поля для изменения информации о сотруднике; Кнопки: «Назад» - необходимая для отмены изменения и возврата на предыдущую страницу, «Изменить» - для применения изменений.

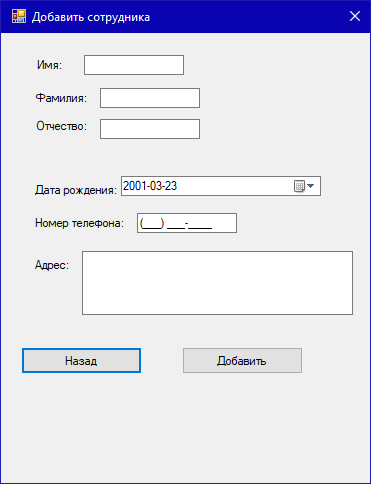


Рисунок 3.17 – Страница «Добавить сотрудника»

Данная страница (Рисунок 3.17) содержит следующие компоненты: Поля для ввода информации о сотруднике; Кнопки: «Назад» - необходимая для отмены добавления и возврата на предыдущую страницу, «Добавить» - для добавления.

**ВЫВОДЫ**

Результатом выполнения курсовой работы стало разработанная база данных и программа (приложение), которое может использоваться в кампании, занимающейся дизайном.

В данной курсовой работе были успешно реализованы все поставленные задачи, были подробно рассмотрены: задача курсового проекта, а также порядок выполнения задания.

Программа приложение имеет интуитивно понятный интерфейс, что способствует быстрому внедрению в программу без трудностей, и не будет надоедать или утомлять пользователя при длительной работе.

Программа имеет большое количество различных программных возможностей и ориентирована на пользователя, не имеющего большого опыта работы с программными продуктами такого типа. Данный программный продукт подходит не только для неопытных пользователей, но и для пользователей, которые имели опыт работы с такими программами, так как интерфейс прост в использовании и не требует длительного привыкания.

**ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1.Державний стандарт України. ДСТУ 300895 Документація. Звіти в сфері науки і техніки. Структура і правила оформлення.

2.Теория и практика построения баз данных. 8-е изд./ Д. Кренке.

–СПб.: Питер, 2003.–800с.

3.Герберт Шилдт – С#4. 0. Полное руководство – Вильямс, 2011–1056с.

4.Глушаков С. В., Ломотько Д. В. Базы данных. (Учебный курс) .– Харьков, 2002.–504с.

5.Мартин Грабер. SQL. – К. :Ид-во «ЛОРИ» , 2003.–644с.

6.Гарсиа Молина, Гектор, Ульман, Джеффри, Уидом, Дженнифер, Системы баз данных. Полный курс. : Пер. с англ. – М. : Издательство дом «Вильямс» , 2003.–1088с.

7.Аткинсон, Леон. MySQL. Библиотека профессионала. : Пер.с англ. – М. : Издательский дом «Вильяме», 2002.–624с