**Содержание**

[ВВЕДЕНИЕ 4](#_Toc9371923)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc9371924)

[2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 12](#_Toc9371925)

[2.1 Построение ER-диаграммы. 12](#_Toc9371926)

[2.2 Построение схемы реляционной модели базы данных. 12](#_Toc9371927)

[2.3 Описание физической модели базы данных. 15](#_Toc9371928)

[2.4 Эскиз интерфейса 19](#_Toc9371929)

[3 ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ 20](#_Toc9371930)

[3.1 Описание структуры базы данных 20](#_Toc9371931)

[3.2 Описание интерфейса пользователя и задач автоматизации 24](#_Toc9371932)

[ВЫВОДЫ 35](#_Toc9371933)

[Список литературы 36](#_Toc9371934)

# ВВЕДЕНИЕ

База данных — это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов.

Базы данных активно используются для динамических сайтов со значительными объемами данных — часто это интернет-магазины, порталы, корпоративные сайты.

Существует большое количество разновидностей баз данных, отличающихся по различным свойствам и критериям. К основным типам баз данных относятся:

• Иерархическая; • Сетевая;

• Объектно-ориентированная; • Реляционная. Самыми распространенными являются реляционные базы данных. Реляционная база данных – это набор данных с предопределенными связями между ними. Эти данные организованны в виде набора таблиц, состоящих из столбцов и строк. В таблицах хранится информация об объектах, представленных в базе данных. В каждом столбце таблицы хранится определенный тип данных, в каждой ячейке – значение атрибута. Каждая стока таблицы представляет собой набор связанных значений, относящихся к одному объекту или сущности. Каждая строка в таблице может быть помечена уникальным идентификатором, называемым первичным ключом, а строки из нескольких таблиц могут быть связаны с помощью внешних ключей.

Для работы с базами данных используются специальные программные средства — системы управления базами данных (СУБД). СУБД – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Основные функции СУБД:

• управление данными во внешней памяти (на дисках);

• управление данными в оперативной памяти с использованием дискового кэша;

• журнализация изменений, резервное копирование и восстановление базы данных после сбоев;

• поддержка языков БД (язык определения данных, язык манипулирования данными).

Для оперирования данными в реляционной базе данных с помощью СУБД используется специальный язык SQL.

SQL (Structured Query Language — Структурированный язык запросов) — язык управления базами данных для реляционных баз данных.

Виды СУБД

По характеру работы СУБД делятся на однопользовательские и многопользовательские. Однопользовательские базы данных подразумевают работу одновременно только с одним пользователем, а с многопользовательскими соответственно могут работать одновременно сразу несколько пользователей. Многопользовательские базы данных в свою очередь делятся на базы данных с последовательным и параллельным доступом.

В настоящее время в основном применяются многопользовательские СУБД. Наиболее популярные из них MS SQL Server, Oracle и MySQL.

MySQL — это самая популярная система управления базами данных при веб-разработке. Большинство сайтов и интернет-порталов, разрабатываются с применением этой СУБД.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Каждый владелец бизнеса знает, что для правильного функционирования системы нужны не только файлы с кодом страниц, но и базы данных. Создав базу данных для базы отдыха тем самым увеличим продуктивность работы сотрудников в разы, избавив их от лишней работы.

БД должна хранить информацию о следующих объектах: клиенты, сотрудники, услуги, бронь, номера домов, информация о домах.

БД должна помогать операторам бронировать номера для клиентов, осуществлять хранение и поиск нужной информации и предоставлять возможность получать разнообразные отчёты. Содержать некоторые готовые запросы. Данная БД должна стать доступным средством для обслуживания клиентов и для ускоренной обработки запросов пользователя. Для хранения информации о всех клиентах, сотрудниках, жилых домов данной базы отдыха. Система строится по следующим правилам:

* Клиент бронирует номер через администратора.
* Бронировать номера можно минимум на день.
* Оплата складывается из цен на услуги и выбора места проживания.
* В заказе должно быть указано место проживания и неограниченное количество услуг.
* Постоялец может себе выбрать в качестве места проживания отдельный дом.
* При отъезде перемещаем постояльца и бронирование из базы данных в архив.
* Все клиенты сохраняются в БД и никогда не удаляются.
* функции администратора:

1) разного рода изменение информации о сотрудниках

2) разного рода изменение информации об услугах

3) разного рода изменение информации об домах и их типах

БД будет хранить информацию о таких объектах:

* Номера домов:
  1. ID дома: Число, 2 символа, обязательное, уникальное, положительное.
  2. ID типа дома: Внешний ключ.
  3. Номер сектора: Число, 2 символа, обязательное.
  4. Расстояние до водоема, м: Число, 3 символа, обязательное.
* Типы домов:
  1. ID типа дома: Число, 2 символа, обязательное, уникальное, положительное.
  2. Название: Текст, 30 символов, обязательное.
  3. Количество комнат: Число, обязательное.
  4. Общая площадь: вещественное, положительное(3,2), обязательное.
  5. Количество кроватей: Число, обязательное.
  6. Стоимость аренды(1 день): вещественное, положительное, обязательное.
  7. Наличие кухни : логическое(0,1), обязательное.
  8. Описание: Текст, 100 символов.
* Услуги:
  1. ID услуги: Число, 2 символов, обязательное, уникальное, положительное.
  2. Название: Текст, 30 символов, обязательное.
  3. Цена: Денежный, 6 символов, 2 после запятой, обязательное.
  4. Описание: Текст, 100 символов.
* Клиенты:
  1. ID: Число, 7 символов, обязательное, уникальное, положительное.
  2. ФИО: Текст, 60 символов, обязательное.
  3. Паспорт: Текст, 10 символов, обязательное, уникальное.
  4. Контактный телефон: Числовое(12), обязательное.
* Сотрудники:
  1. ID: Число, 7 символов, обязательное, уникальное, положительное.
  2. Имя: Текст, 30 символов, обязательное.
  3. Фамилия: Текст, 30 символов, обязательное.
  4. Пол: М/Ж, список, обязательное.
  5. Дата рождения: дата, обязательное, сотрудники не могут быть младше 18 лет.
  6. Контактный телефон: Числовое(12), обязательное.
  7. ID должности: внешний ключ.
  8. Дата приёма на работу: Дата, обязательное, Больше 2015 года.
* Бронь:
  1. ID бронировки: Число, 7 символов, обязательное, уникальное, положительное.
  2. ID сотрудника: Внешний ключ.
  3. ID клиента: Внешний ключ.
  4. ID дома: Внешний ключ.
  5. Дата заключения: Дата, обязательное, вычисляется по текущей дате.
  6. Дата въезда: Дата, обязательное, не раньше текущей даты.
  7. Дата выезда: Дата, обязательное, не раньше или равно даты въезда.

Список дополнительных услуг:

1. Прокат инвентаря
   1. Рыбалка
      * Удочки
      * Палатка
      * Сетка
   2. Спортинвентарь
      * Ракетки для тенниса
      * Волейбольный мяч
      * Коньки
      * Клюшка и шайба
      * Лыжи, сноуборд
2. Баня
   1. Русская
   2. Турецкая
3. Дополнительная аренда
   1. Детская коляска
   2. Мангал
   3. Шампура х5

Система создаётся для обслуживания следующих групп пользователей:

1. Администрация;
2. Менеджеры;

**1. Функциональные возможности системы**

Система «База отдыха» предоставляет следующие возможности для пользователей:

* Занесение новых клиентов, сотрудников, информации о домах, услуг
* Возможность работы с системой только для авторизованных пользователей.
* Изменение информации о клиентах, заказах, сотрудниках, информации о домах, услуг.
* Выбор клиентам определенных услуг.
* Просмотр информации о любом клиенте, заказе, услуге, доме и т.д.
* Просмотр архивной информации
* Вывод нужной информации на печать
* Удобный поиск нужной информации
* Формирование чека об оплате
* Просмотр статистики заказов
* Выселение клиентов досрочно.

**2. Готовые запросы**

* Поиск номеров по разным параметрам (количество комнат, кроватей, диапазону цен, типу и т.д.)
* Формирование списка занятых и свободных номеров по указанной дате.
* Формирование списка постояльцев, услуг, домов и их типов, заказов
* Формирование архивных таблиц об клиентах, заказах, сотрудниках
* Обеспечение бронирования номеров на определенную дату
* Формирование финансовых отчетов
* Отмена брони

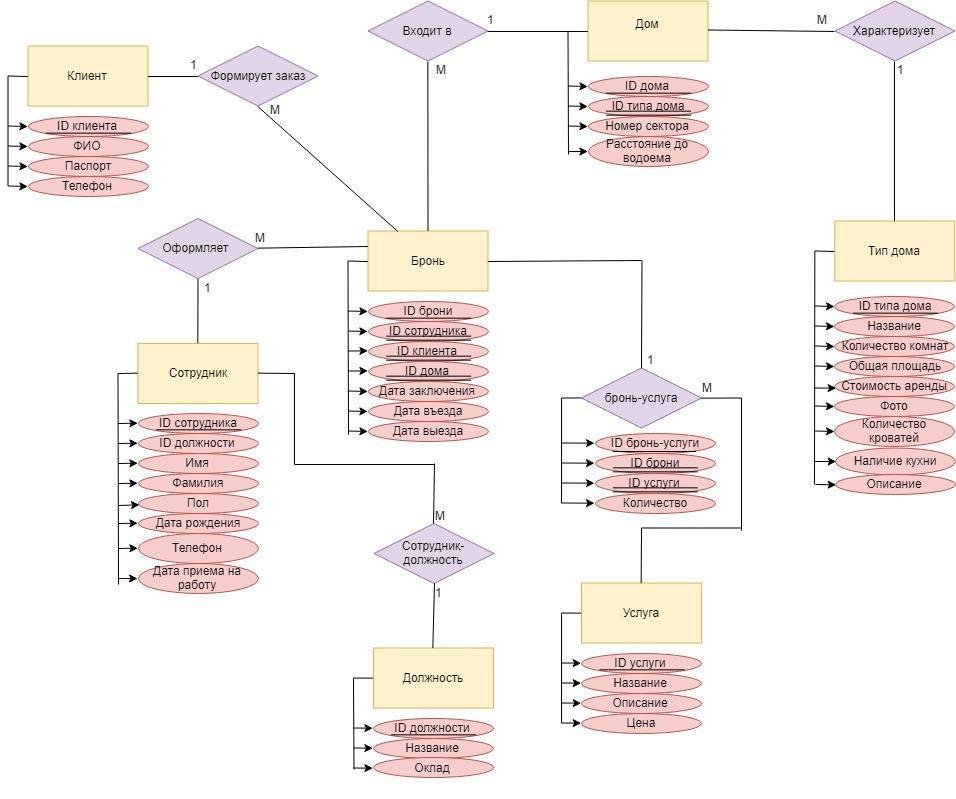
В базе данных необходимо контролировать следующую информацию:

* Сотрудники должны быть старше 18 и младше 65 лет
* Номер телефона при вводе должен совпадать с маской 380\_ \_ \_ \_ \_ \_ \_ \_ \_
* Паспорт состоит полностью из цифр или первые два символа – буквы, остальное – символы.
* Секторы расположения домов не должны совпадать
* При формировании заказа с «старым» клиентом об этом нужно дать знать системе
* Формировать заказ на текущий день можно только до 12:00 (время заезда постояльцев) или на любой будущий день при наличии свободных домов.
* При досрочной отмене брони клиенту возвращается сумма в размере количества не прожитых дней \* стоимость аренды (с учетом был ли заказ отменен до 18:00(время выезда постояльцев))
* Номера паспортов клиентов не могут совпадать

# 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(таблицы 2.1-2.11), которая отвечает всем требованиям нормализации.

Таблица 2.1 – Отношение КЛИЕНТЫ

|  |  |  |  |
| --- | --- | --- | --- |
| ID клиента | ФИО | Паспорт | Телефон |
|  |  |  |  |

Таблица 2.2 – Отношение БРОНЬ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID брони | ID сотрудника | ID клиента | ID дома | Дата заключения | Дата въезда | Дата выезда |
|  |  |  |  |  |  |  |

Таблица 2.3 – Отношение СОТРУДНИК

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID сотрудника | Имя | Фамилия | Пол | День рождения | Телефон | ID должности | Дата приема на работу |
|  |  |  |  |  |  |  |  |

Таблица 2.4 – Отношение БРОНЬ-УСЛУГА

|  |  |  |
| --- | --- | --- |
| ID брони | ID услуги | Количество |
|  |  |  |

Таблица 2.5 – Отношение УСЛУГА

|  |  |  |  |
| --- | --- | --- | --- |
| ID услуги | Название | Цена | Описание |
|  |  |  |  |

Таблица 2.6 – Отношение ДОМ

|  |  |  |  |
| --- | --- | --- | --- |
| ID дома | ID типа дома | Номер сектора | Расстояние до водоёма |
|  |  |  |  |

Таблица 2.7 – Отношение ТИП ДОМА

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID типа дома | Название | Количество комнат | Общая площадь | Количество кроватей | Стоимость аренды | Кухня | Описание |
|  |  |  |  |  |  |  |  |

Таблица 2.8 – Отношение ДОЛЖНОСТЬ

|  |  |  |
| --- | --- | --- |
| ID должности | Название | Оклад |
|  |  |  |

Таблица 2.9 – Отношение ПОЛЬЗОВАТЕЛИ

|  |  |  |
| --- | --- | --- |
| ID пользователя | Логин | Пароль |
|  |  |  |

Таблица 2.10 – Отношение АРХИВ - СОТРУДНИК

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID сотрудника | Имя | Фамилия | Телефон | Дата приема на работу | Дата увольнения | Причина увольнения |
|  |  |  |  |  |  |  |

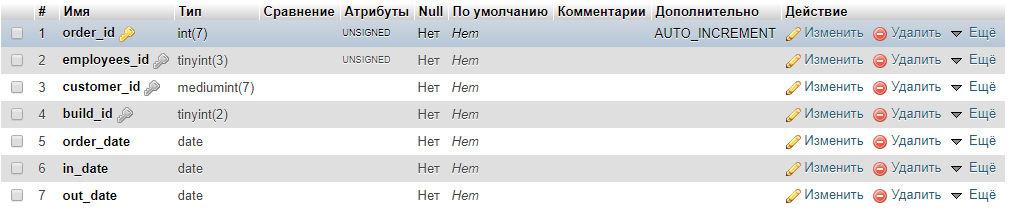
Таблица 2.11 – Отношение АРХИВ - БРОНЬ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID брони | ID сотрудника | ID клиента | ID дома | Стоимость услуг | Дата заключения | Дата въезда | Дата выезда | Причина | Выполнено |
|  |  |  |  |  |  |  |  |  |  |

## 2.3 Описание физической модели базы данных.

База данных «База отдыха» была разработана с помощью СУБД MySQL phpMyAdmin 4.8.4

При создании БД были в режиме конструктора были созданы 11 таблиц и схема данных. Скриншоты таблиц, в режиме конструктора, представлены на рисунках 2.2-2.14.

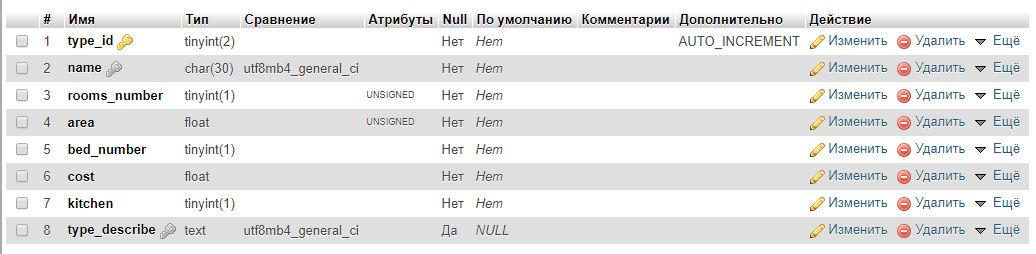
На рисунке 2.1 представлена таблица «Заказ», в которую включены следующие поля и характеристики: ключевое поле - код заказа, код сотрудника, код дома, дата заказа, дата въезда, дата выезда.

*Рисунок 2.1 – Структура таблицы «Заказ»*

На рисунке 2.2 представлена таблица «Дома», в которую включены следующие поля и характеристики: ключевое поле – код дома, код типа дома, номер сектора, дистанция до водоёма.

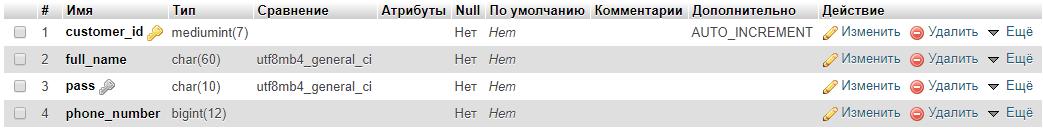


*Рисунок 2.2 – Структура таблицы «Дома»*

На рисунке 2.3 представлена таблица «Тип дома», в которую включены следующие поля и характеристики: ключевое поле – код типа дома, название, количество комнат, площадь, количество кроватей, стоимость, кухня, описание.

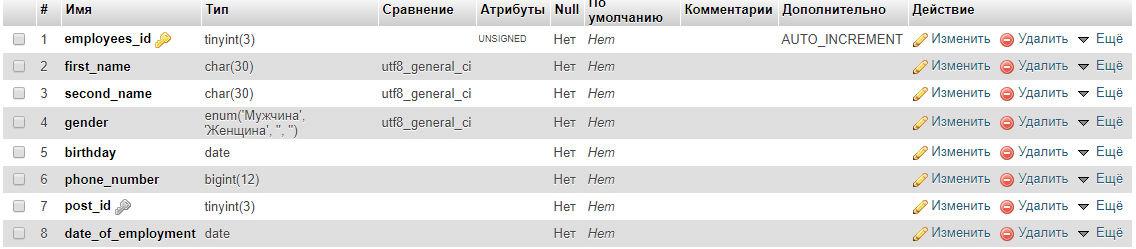
*Рисунок 2.3 – Структура таблицы «Тип дома»*

На рисунке 2.4 представлена таблица «Клиент», в которую включены следующие поля и характеристики: ключевое поле – код клиента, ФИО, паспорт, номер телефона.



*Рисунок 2.4 – Структура таблицы «Клиент»*

На рисунке 2.5 представлена таблица «Сотрудник», в которую включены следующие поля и характеристики: ключевое поле – код сотрудника, имя, фамилия, пол, день рождения, номер телефона, код должности, дата устройства на работу



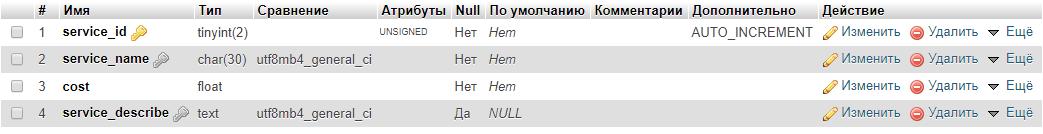
*Рисунок 2.5 – Структура таблицы «Сотрудник»*

На рисунке 2.6 представлена таблица «Должность», в которую включены следующие поля и характеристики: ключевое поле – код должности, название, оклад, доступ админа.

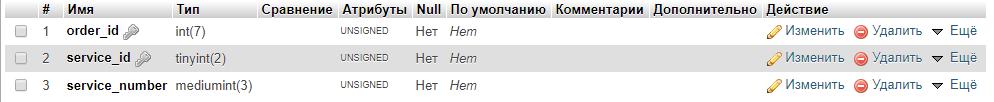


*Рисунок 2.6 – Структура таблицы «Должность»*

На рисунке 2.7 представлена таблица «Услуги», в которую включены следующие поля и характеристики: ключевое поле – код услуги, название, цена, описание.



*Рисунок 2.7 – Структура таблицы «Услуги»*

На рисунке 2.8 представлена таблица «Заказ-услуги», в которую включены следующие поля и характеристики: ключевое поле – код заказа/ код услуги, количество.

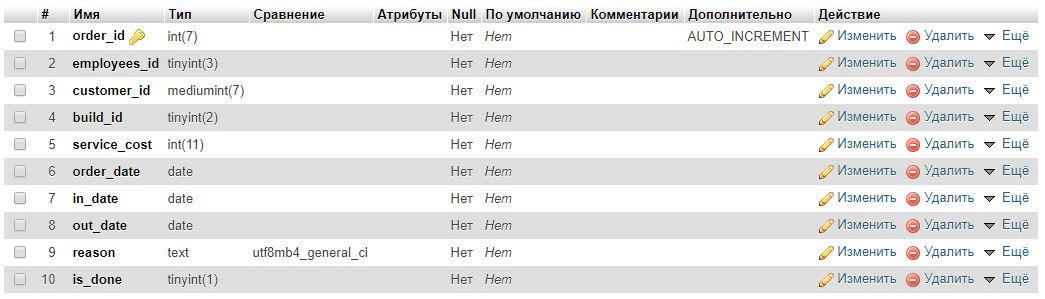
*Рисунок 2.8 – Структура таблицы «Заказ-услуги»*

На рисунке 2.9 представлена таблица «Пользователи», в которую включены следующие поля и характеристики: ключевое поле – код пользователя, логин, пароль.



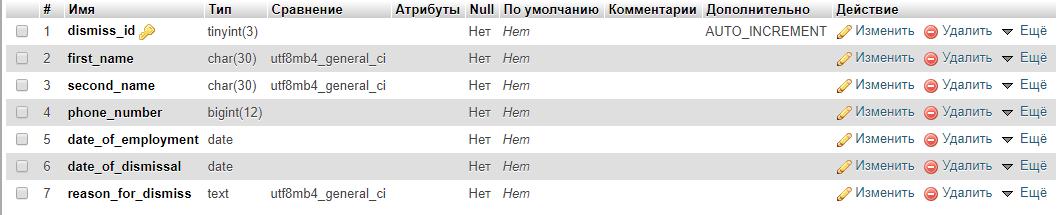
*Рисунок 2.9 – Структура таблицы «Пользователи»*

На рисунке 2.10 представлена таблица «Архив – заказы», в которую включены следующие поля и характеристики: ключевое поле – код заказа, код сотрудника, код клиента, код дома, цена за услуги, дата заказа, дата въезда, дата выезда, причина, состояние.



*Рисунок 2.10 – Структура таблицы «Архив – заказы»*

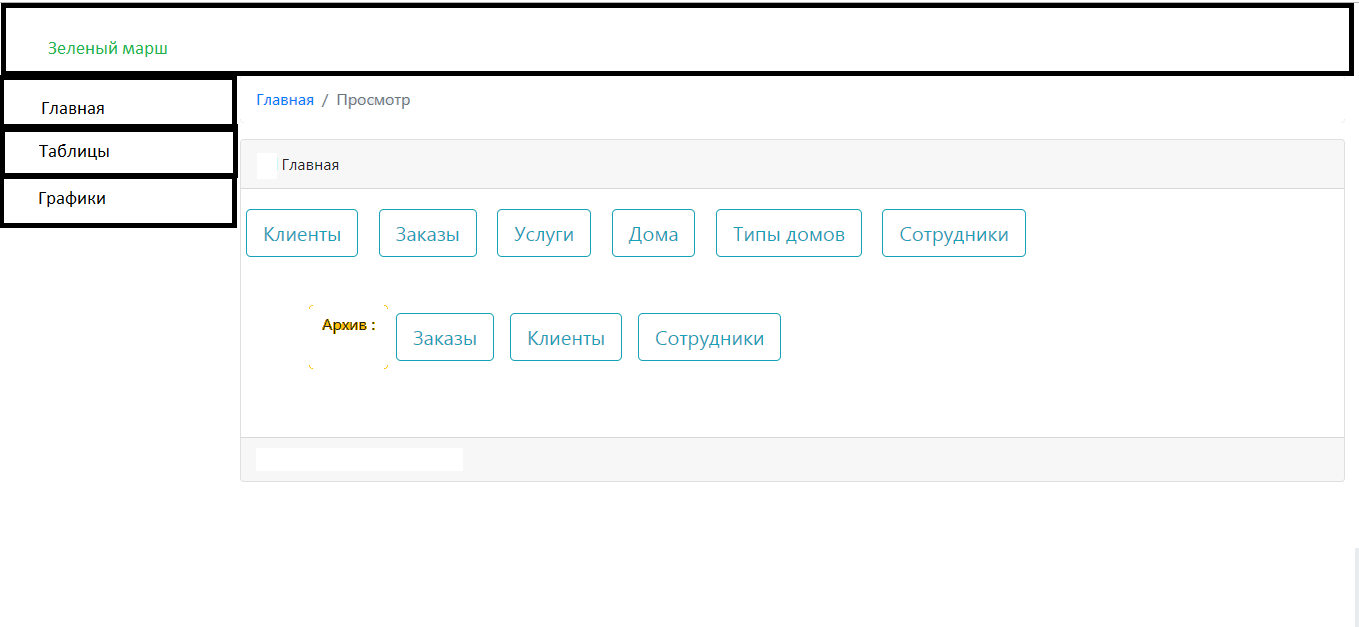
На рисунке 2.11 представлена таблица «Архив – сотрудники», в которую включены следующие поля и характеристики: ключевое поле – код сотрудника, имя, фамилия, номер телефона, дата приема на работу, дата увольнения, причина увольнения.



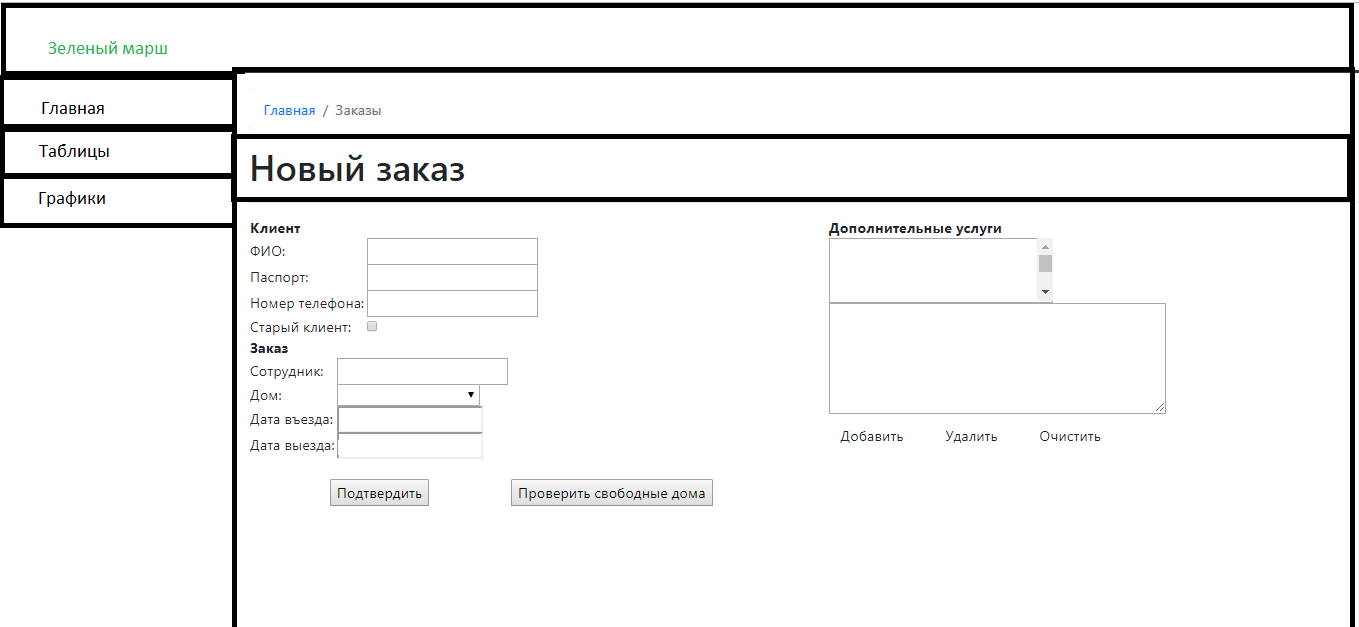
*Рисунок 2.11 – Структура таблицы «Архив – сотрудники»*

## 2.4 Эскиз интерфейса

Эскиз интерфейса - это быстрый набросок будущего приложения. Он дает возможность сгруппировать объекты и правильно распределить их на рабочем полотне. Интерфейс приложения для работы с БД будет разработан в виде веб страницы с более-менее одинаковым стилем. Далее на рисунках 2.12 – 2.13 представлены эскизы.



*Рисунок 2.12 – Эскиз интерфейса страницы “Главное меню”*

**

*Рисунок 2.13 – Эскиз интерфейса страницы “Добавления заказа”*

# 3 ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ

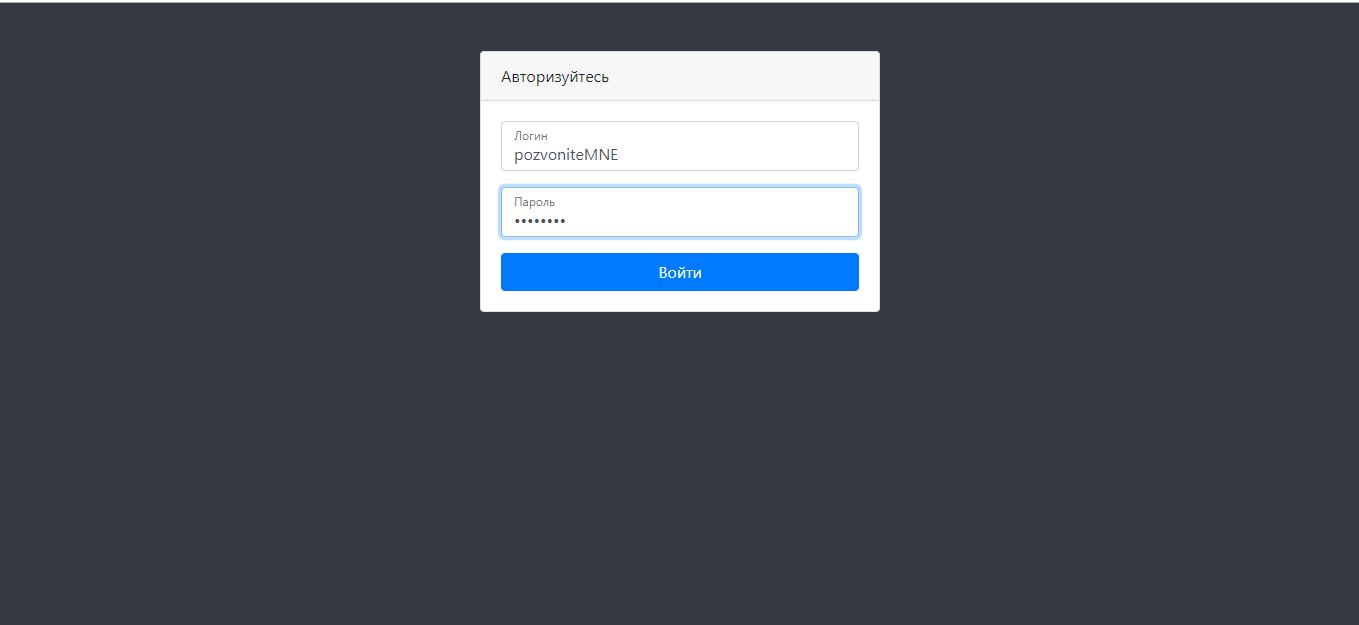
## 3.1 Описание структуры базы данных

В соответствии с физической моделью БД, была создана структура таблиц СУБД MySQL. Далее представлен текст из консоли MySQL-сервера, показывающий, в каком виде создавались и хранятся таблицы данной БД:

|  |  |
| --- | --- |
| **book** | CREATE TABLE `book` (  `order\_id` int(7) unsigned NOT NULL AUTO\_INCREMENT,  `employees\_id` tinyint(3) unsigned NOT NULL,  `customer\_id` mediumint(7) NOT NULL,  `build\_id` tinyint(2) NOT NULL,  `order\_date` date NOT NULL,  `in\_date` date NOT NULL,  `out\_date` date NOT NULL,  PRIMARY KEY (`order\_id`),  KEY `employees\_id` (`employees\_id`),  KEY `customer\_id` (`customer\_id`),  KEY `build\_id` (`build\_id`),  CONSTRAINT `book\_ibfk\_1` FOREIGN KEY (`customer\_id`) REFERENCES `customers` (`customer\_id`),  CONSTRAINT `book\_ibfk\_3` FOREIGN KEY (`build\_id`) REFERENCES `build` (`build\_id`),  CONSTRAINT `book\_ibfk\_4` FOREIGN KEY (`employees\_id`) REFERENCES `employees` (`employees\_id`) ) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8mb4 |
| **book\_arch** | CREATE TABLE `book\_arch` (  `order\_id` int(7) NOT NULL AUTO\_INCREMENT,  `employees\_id` tinyint(3) NOT NULL,  `customer\_id` mediumint(7) NOT NULL,  `build\_id` tinyint(2) NOT NULL,  `service\_cost` int(11) NOT NULL,  `order\_date` date NOT NULL,  `in\_date` date NOT NULL,  `out\_date` date NOT NULL,  `reason` text NOT NULL,  `is\_done` tinyint(1) NOT NULL,  PRIMARY KEY (`order\_id`) ) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8mb4 |
| **build** | CREATE TABLE `build` (  `build\_id` tinyint(2) NOT NULL AUTO\_INCREMENT,  `type\_id` tinyint(2) NOT NULL,  `sector\_number` tinyint(2) DEFAULT NULL,  `sector\_distance` smallint(3) DEFAULT NULL,  PRIMARY KEY (`build\_id`),  UNIQUE KEY `sector\_number` (`sector\_number`),  KEY `type\_id` (`type\_id`),  CONSTRAINT `build\_ibfk\_1` FOREIGN KEY (`type\_id`) REFERENCES `build\_type` (`type\_id`) ) ENGINE=InnoDB AUTO\_INCREMENT=10 DEFAULT CHARSET=utf8mb4 |
| **build\_type** | CREATE TABLE `build\_type` (  `type\_id` tinyint(2) NOT NULL AUTO\_INCREMENT,  `name` char(30) NOT NULL,  `rooms\_number` tinyint(1) unsigned NOT NULL,  `area` float unsigned NOT NULL,  `bed\_number` tinyint(1) NOT NULL,  `cost` float NOT NULL,  `kitchen` tinyint(1) NOT NULL,  `type\_describe` text,  PRIMARY KEY (`type\_id`),  FULLTEXT KEY `Name` (`name`),  FULLTEXT KEY `type\_describe` (`type\_describe`) ) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8mb4 |
| **customers** | CREATE TABLE `customers` (  `customer\_id` mediumint(7) NOT NULL AUTO\_INCREMENT,  `full\_name` char(60) NOT NULL,  `pass` char(10) NOT NULL,  `phone\_number` bigint(12) NOT NULL,  PRIMARY KEY (`customer\_id`),  UNIQUE KEY `pass` (`pass`) ) ENGINE=InnoDB AUTO\_INCREMENT=14 DEFAULT CHARSET=utf8mb4 |
| **employees** | CREATE TABLE `employees` (  `employees\_id` tinyint(3) unsigned NOT NULL AUTO\_INCREMENT,  `first\_name` char(30) CHARACTER SET utf8 NOT NULL,  `second\_name` char(30) CHARACTER SET utf8 NOT NULL,  `gender` enum('Мужчина','Женщина','','') CHARACTER SET utf8 NOT NULL,  `birthday` date NOT NULL,  `phone\_number` bigint(12) NOT NULL,  `post\_id` tinyint(3) NOT NULL,  `date\_of\_employment` date NOT NULL,  PRIMARY KEY (`employees\_id`),  KEY `post\_id` (`post\_id`),  CONSTRAINT `employees\_ibfk\_1` FOREIGN KEY (`post\_id`) REFERENCES `post` (`post\_id`) ) ENGINE=InnoDB AUTO\_INCREMENT=22 DEFAULT CHARSET=utf8mb4 |
| **employees\_arch** | CREATE TABLE `employees\_arch` (  `dismiss\_id` tinyint(3) NOT NULL AUTO\_INCREMENT,  `first\_name` char(30) NOT NULL,  `second\_name` char(30) NOT NULL,  `phone\_number` bigint(12) NOT NULL,  `date\_of\_employment` date NOT NULL,  `date\_of\_dismissal` date NOT NULL,  `reason\_for\_dismiss` text NOT NULL,  PRIMARY KEY (`dismiss\_id`) ) ENGINE=InnoDB AUTO\_INCREMENT=27 DEFAULT CHARSET=utf8mb4 |
| **post** | CREATE TABLE `post` (  `post\_id` tinyint(3) NOT NULL AUTO\_INCREMENT,  `post\_name` char(30) NOT NULL,  `salary` float NOT NULL,  `is\_admin` tinyint(1) NOT NULL,  PRIMARY KEY (`post\_id`) ) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8mb4 |
| **service** | CREATE TABLE `service` (  `service\_id` tinyint(2) unsigned NOT NULL AUTO\_INCREMENT,  `service\_name` char(30) NOT NULL,  `cost` float NOT NULL,  `service\_describe` text,  PRIMARY KEY (`service\_id`),  FULLTEXT KEY `service\_name` (`service\_name`),  FULLTEXT KEY `service\_describe` (`service\_describe`) ) ENGINE=InnoDB AUTO\_INCREMENT=14 DEFAULT CHARSET=utf8mb4 |
| **service\_book** | CREATE TABLE `service\_book` (  `order\_id` int(7) unsigned NOT NULL,  `service\_id` tinyint(2) unsigned NOT NULL,  `service\_number` mediumint(3) unsigned NOT NULL,  KEY `order\_id` (`order\_id`),  KEY `service\_id` (`service\_id`),  CONSTRAINT `service\_book\_ibfk\_2` FOREIGN KEY (`service\_id`) REFERENCES `service` (`service\_id`),  CONSTRAINT `service\_book\_ibfk\_3` FOREIGN KEY (`order\_id`) REFERENCES `book` (`order\_id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 |
| **users** | CREATE TABLE `users` (  `user\_id` tinyint(3) unsigned NOT NULL,  `login` char(32) NOT NULL,  `password` char(32) NOT NULL,  PRIMARY KEY (`user\_id`),  CONSTRAINT `users\_ibfk\_1` FOREIGN KEY (`user\_id`) REFERENCES `employees` (`employees\_id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 |

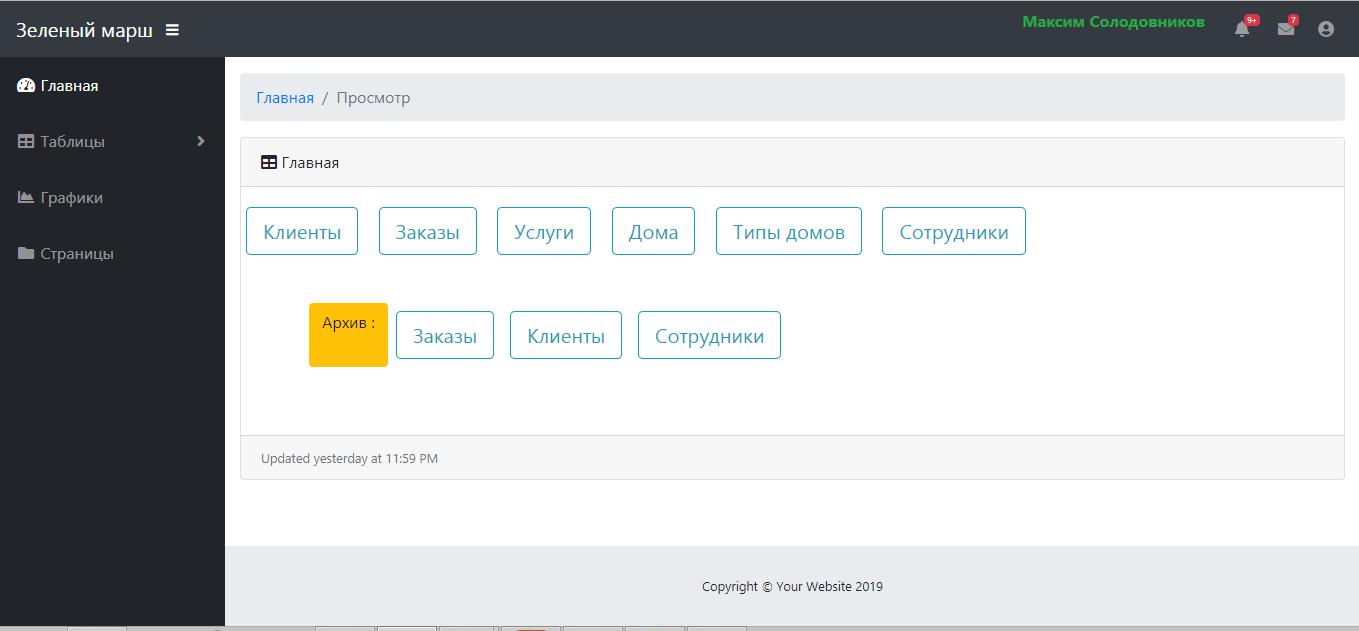
## 3.2 Описание интерфейса пользователя и задач автоматизации

Для начала работы с системой следует перейти по ссылке полученной у администратора и авторизоваться.(Рис 5.1)



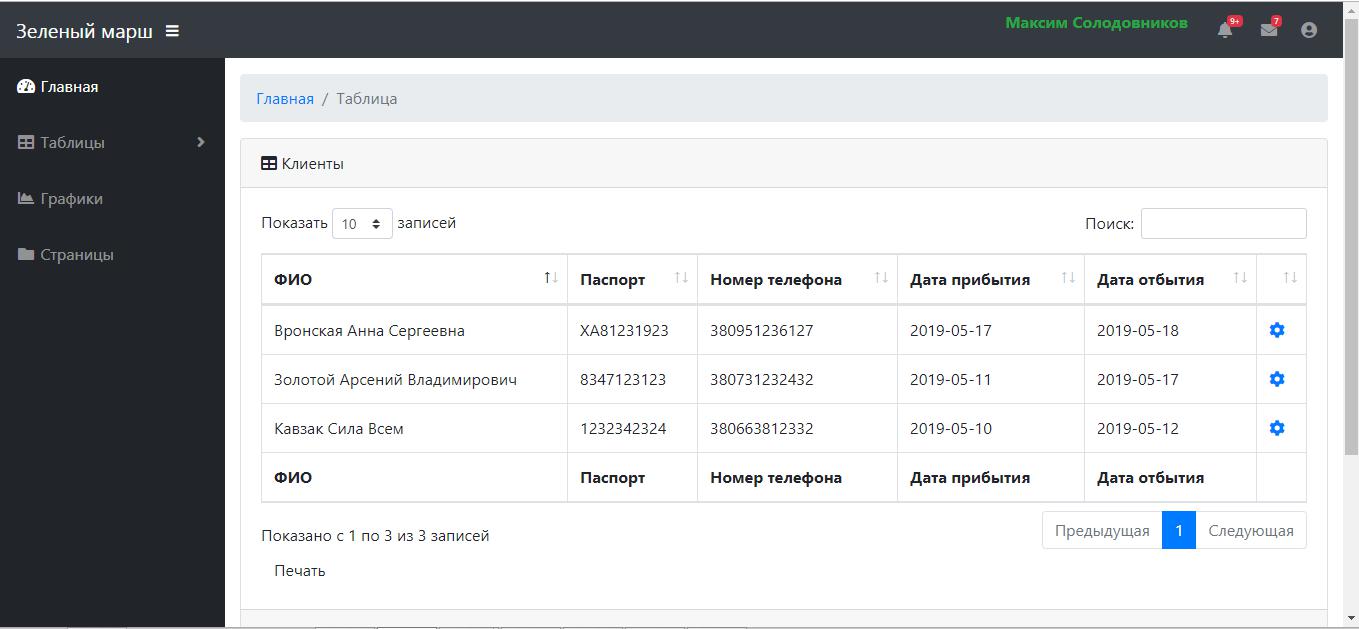
*Рисунок 5.1 – Окно авторизации*

Данная страница содержит два поля для ввода логина и пароля соответственно. При подборе подходящей комбинации логин-пароль мы сможем увидеть главную страницу приглашающую нас начать работу(Рис 5.2) иначе система оповестит о неправильном вводе логина или пароля и попросит повторить попытку.

*Рисунок 5.2 – Окно главного меню*

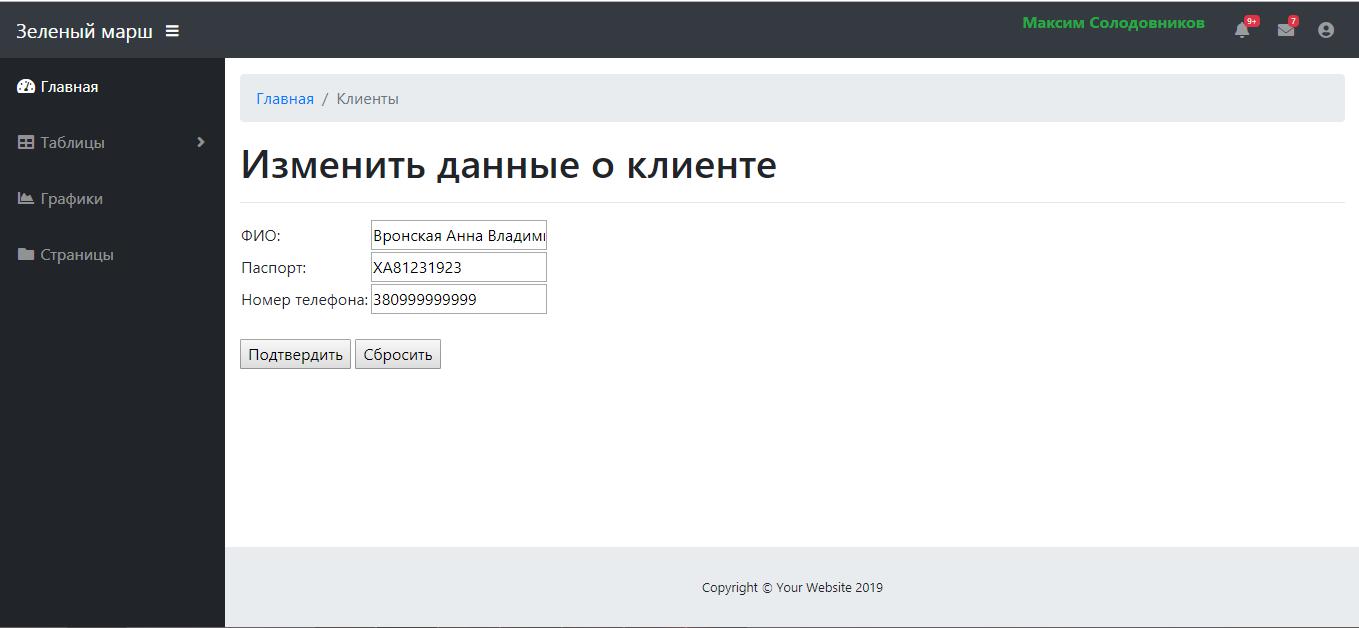
Страница «Главная» представляет нам выбор: начать работу с действующих таблиц или работать с архивом. Для примера рассмотрим все возможные функции доступные для работы с некоторыми таблицами.

Первая таблица – Клиенты (Рис 5.3), при выборе её, выводит список действующих клиентов.

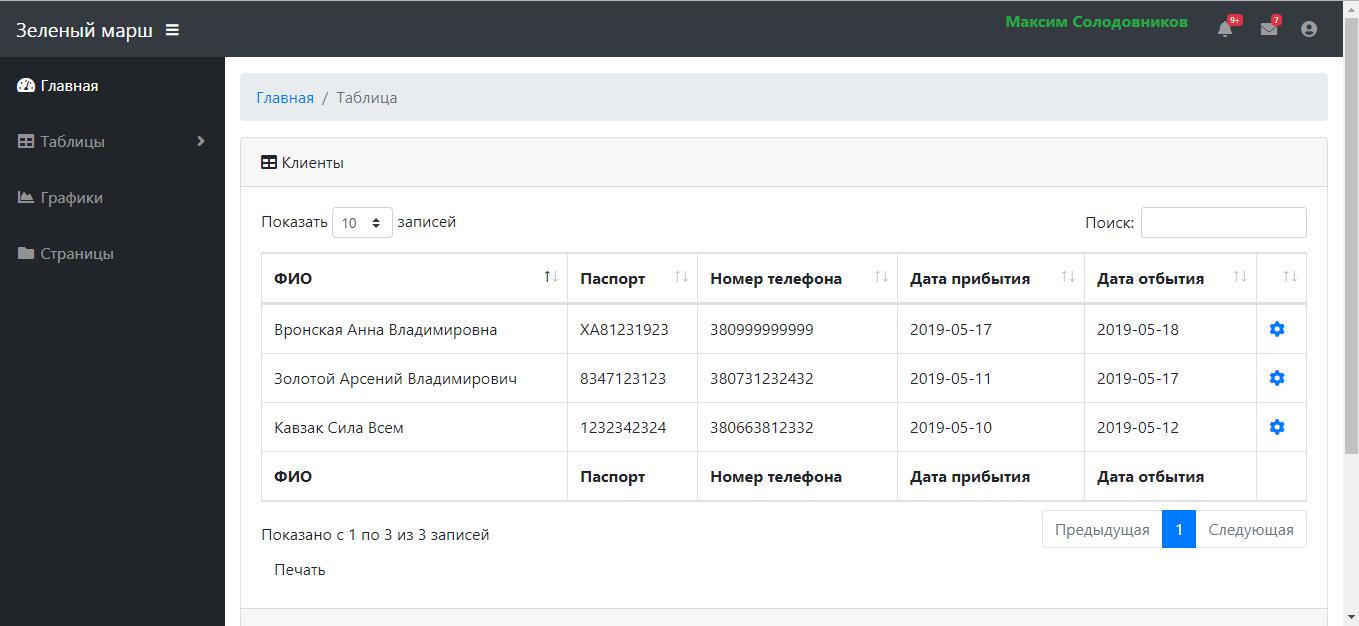


*Рисунок 5.3 – Окно вывода таблицы «Клиенты»*

Из доступных функций, помимо различных фильтров, есть возможность изменять данные о клиенте при нажатии на шестеренку напротив необходимого клиента. Для примера изменим данные о клиенте Вронская Анна Сергеевна(Рис 5.4)

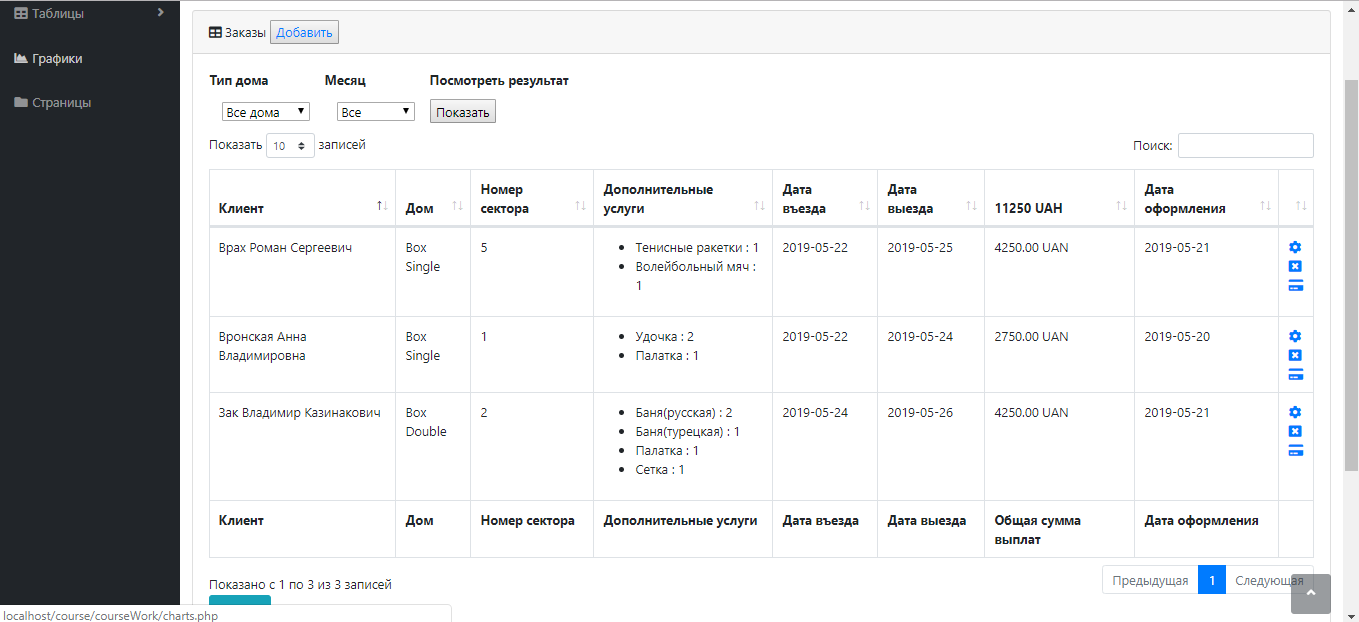


*Рисунок 5.4 – Окно изменения данных о клиенте*

* данной форме мы изменили отчество Сергеевна на Владимировна и поменяли номер телефона на 380999999999, сохраним результаты и посмотрим что получится(Рис 5.5) 

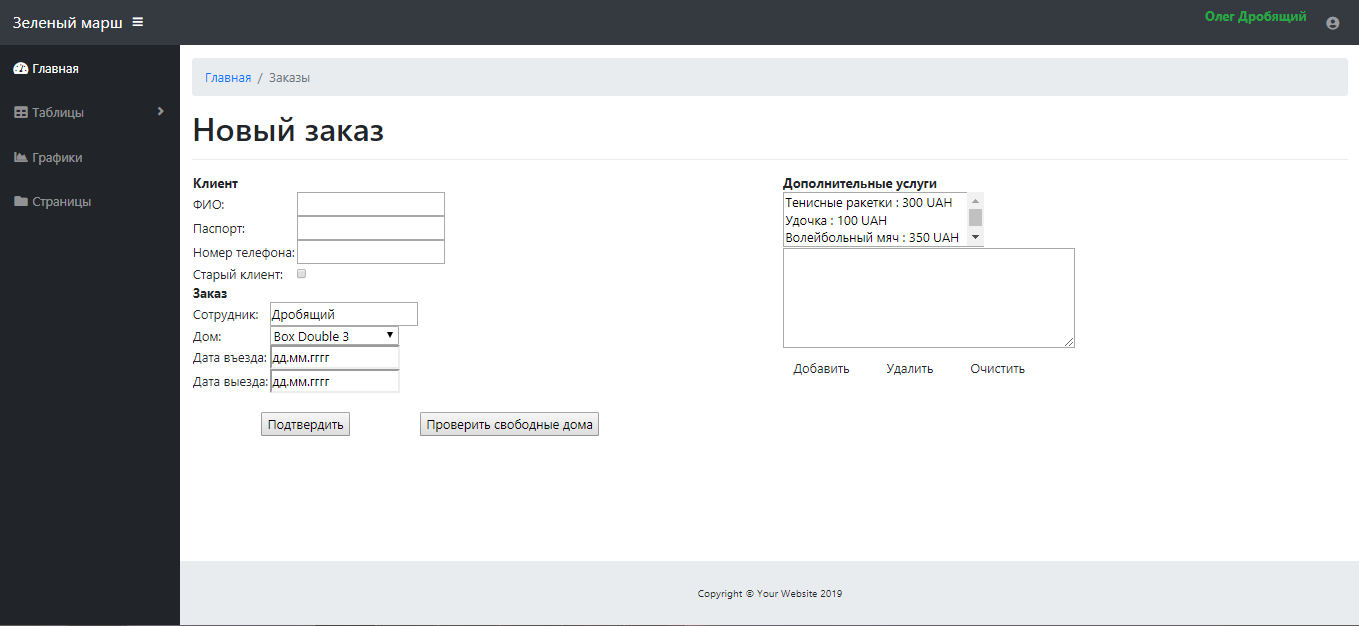
*Рисунок 5.5 – Окно вывода таблицы «Клиенты»*

Далее перейдем во вкладку заказы и увидим существующие на данный момент заказы(Рис 5.6).



*Рисунок 5.6 – Окно вывода таблицы «Заказы»*

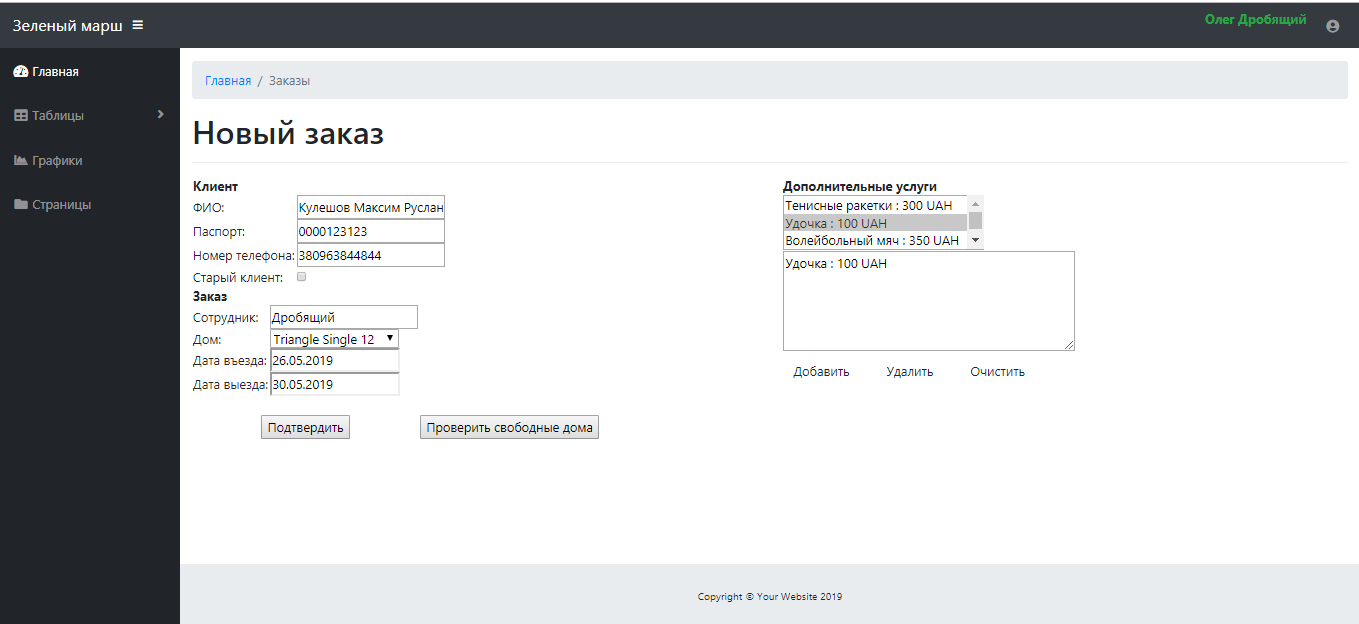
Сверху, рядом с надписью «Заказы», есть кнопка добавить, нажав на которую мы попадаем на форму создания нового заказа.(Рис 5.7)



*Рисунок 5.7 – Окно создания нового заказа*

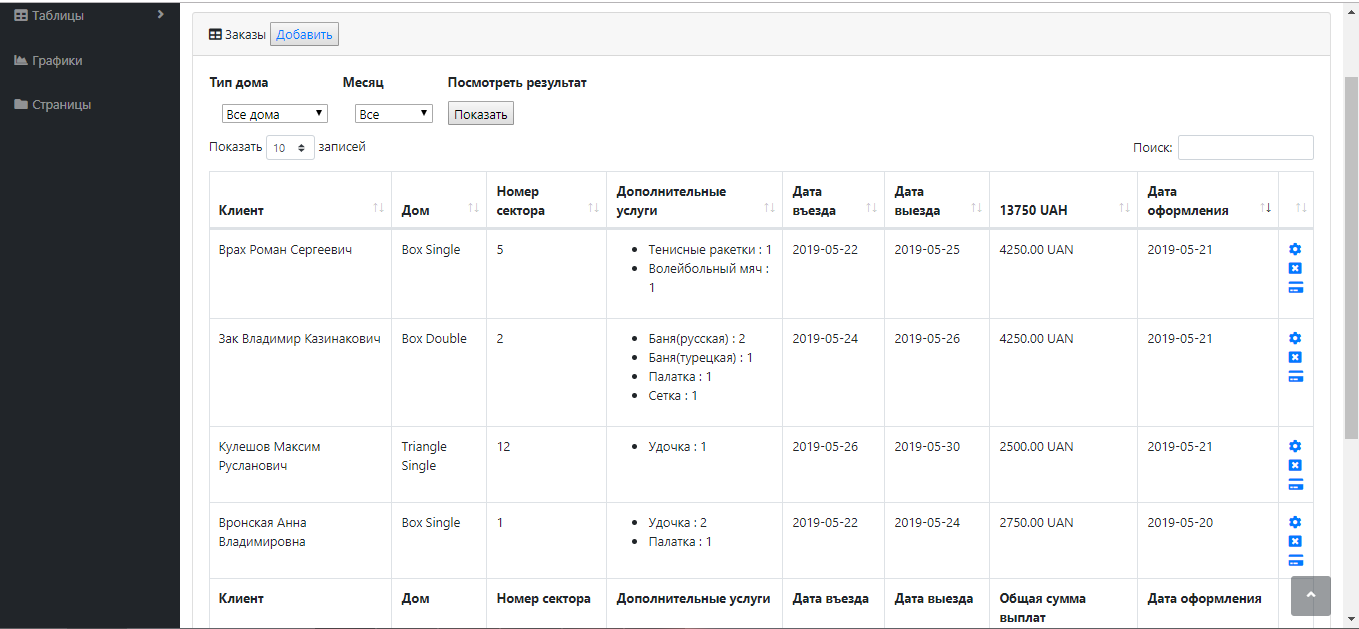
Данное окно предлагает нам следующий функционал:

1. Добавление в БД нового клиента или выбор «старого»
2. Возможность просмотра свободных домов на определенную дату при выборе даты въезда, даты выезда и при нажатии на кнопку «Проверить свободные дома»
3. Добавление неограниченного количества дополнительных услуг
4. Формирование нового заказа в случае ввода верных данных



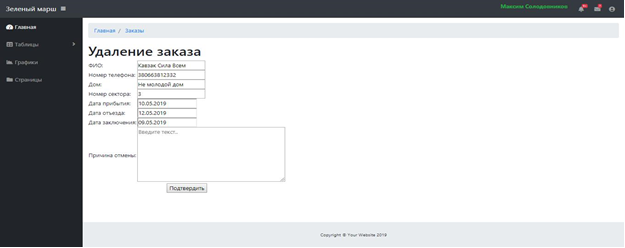
*Рисунок 5.8 – Окно с заполненной формой создания нового заказа*

Заполнив данными форму(Рис 5.8) нажимаем кнопку подтвердить для создания нового заказа. Сразу же после нажатия кнопки нас перенаправляет на страницу вывода информации о действующих заказах с только что оформленным заказом(Рис 5.9)

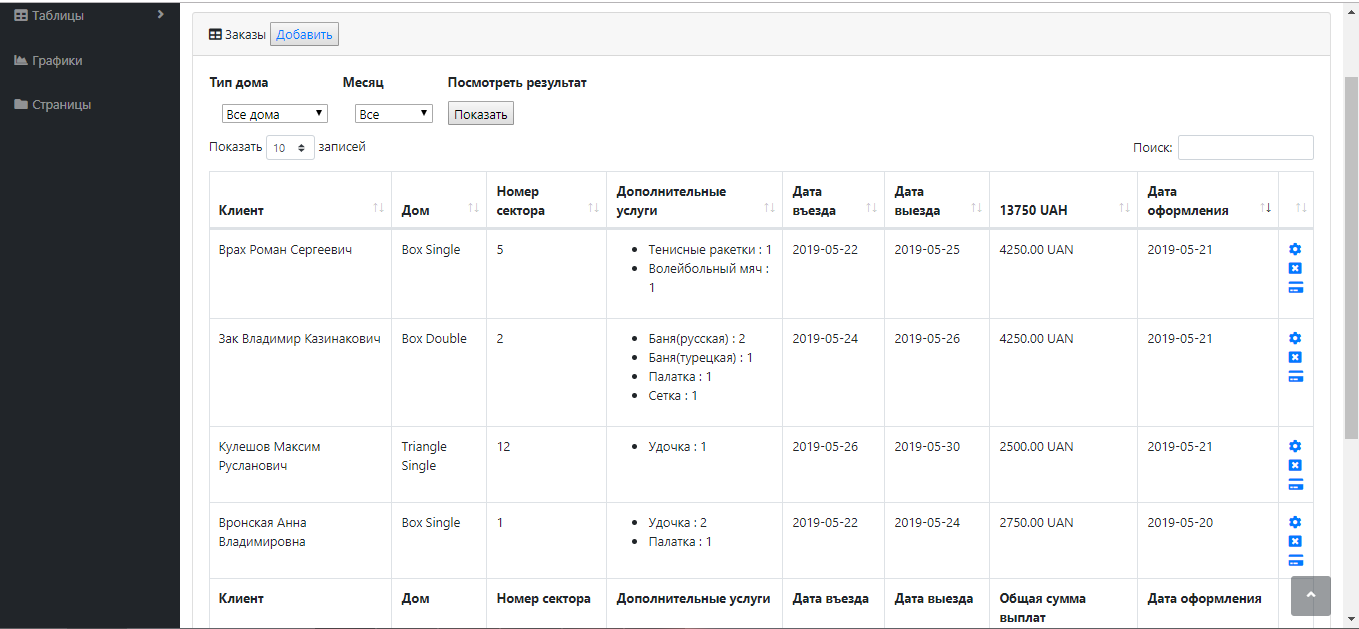


*Рисунок 5.9 – Окно вывода таблицы «Заказы»*

Справа возле каждой строчки мы можем наблюдать три кнопки для взаимодействия с определенной информацией: шестеренка для изменения данных о заказе, крестик – удаление или отмена заказа, карта – посчитать и распечатать чек. Рассмотрим удаление(Рис 5.10).

*Рисунок 5.10– Окно удаления действующего заказа*

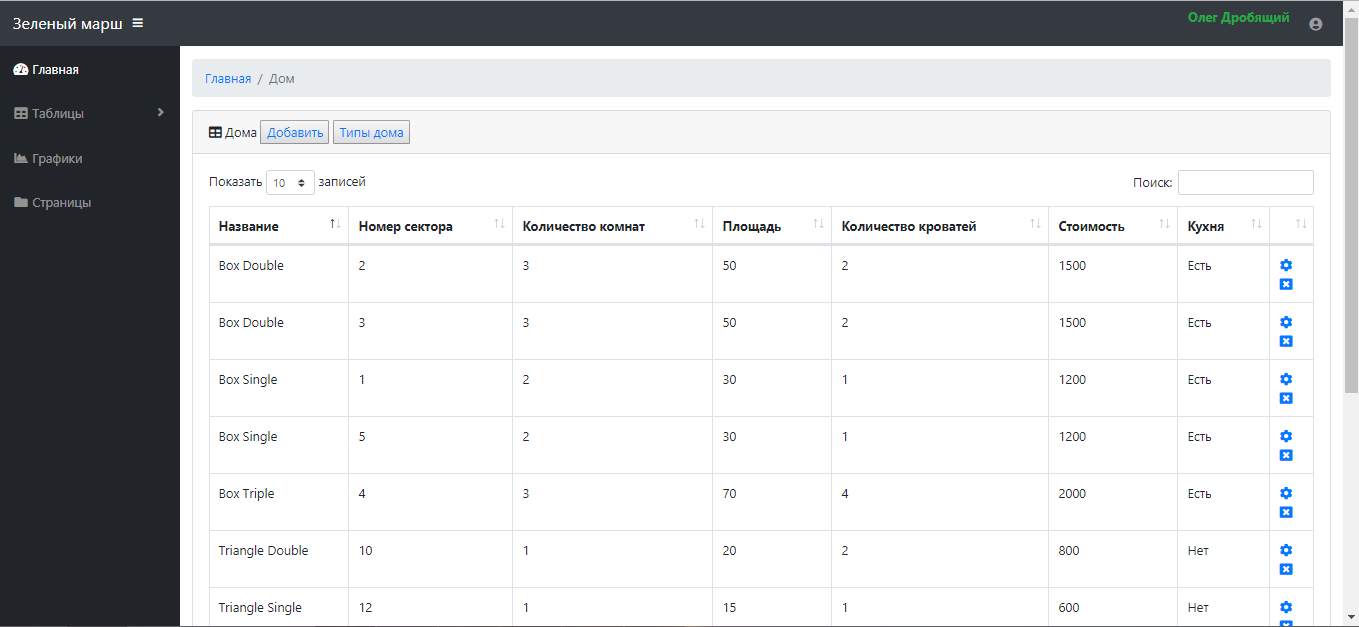
При нажатии на крестик на определенной строке, вся информация выбранного объекта переносится на форму(Рис 5.10) где следует указать причину отмены заказа, после чего можно удалить заказ нажав кнопку «Подтвердить» После нажатия кнопки нас возвращает на окно вывода действующих заказов, где мы можем продолжить свою работу дальше(Рис 5.11).



*Рисунок 5.11 – Окно вывода таблицы «Заказы»*

В итоге при работы с таблицей «Заказы» мы имеем такие возможности:

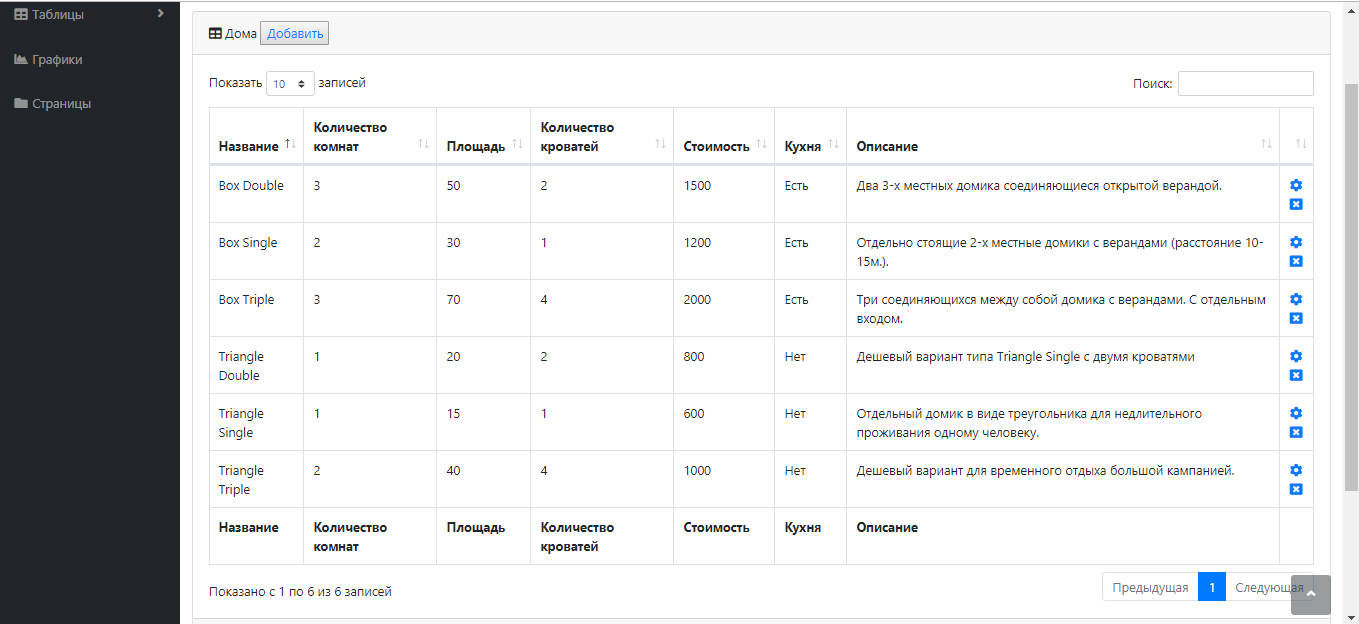
* 1. Просмотр данных
  2. Добавление данных
  3. Удаление данных
  4. Изменение данных
  5. Вывод данных на печать



*Рисунок 5.12 – Окно вывода таблицы «Дома»*

На данной форме(Рис 5.12) выводится информация о всех домах и доступны следующие функциональные возможности:

1. Добавление новых домов
2. Редактирование существующих домов
3. Удаление существующих домов при условии, что они отсутствуют в заказах
4. Просмотр «Типов домов»(Рис 5.13)

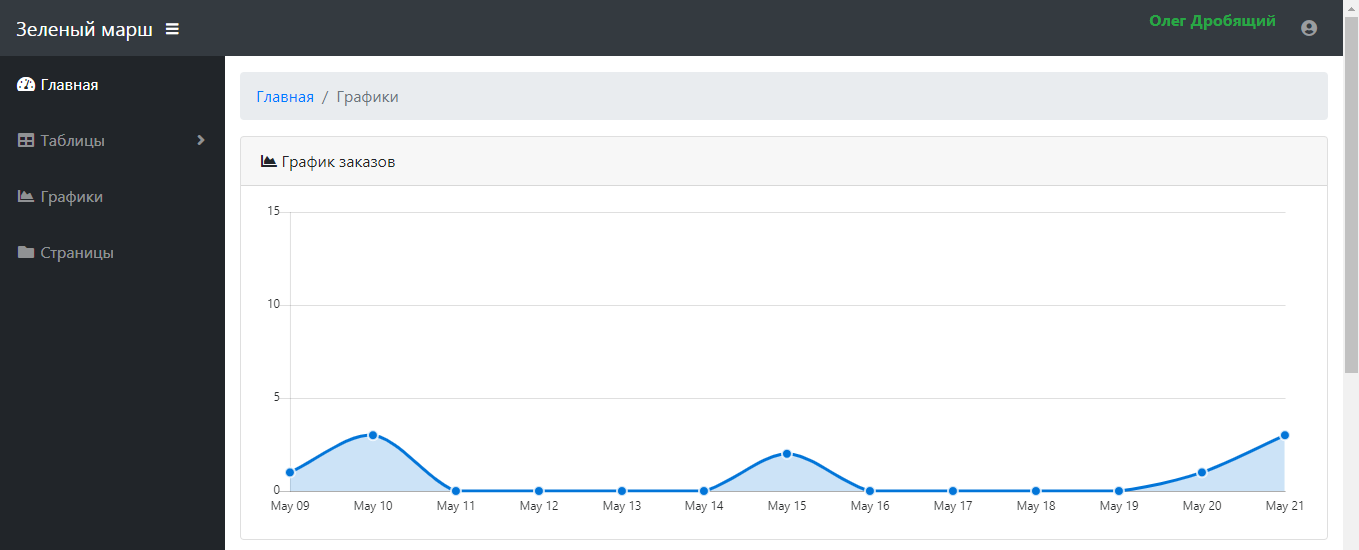


*Рисунок 5.13 – Окно вывода таблицы «Типы домов»*

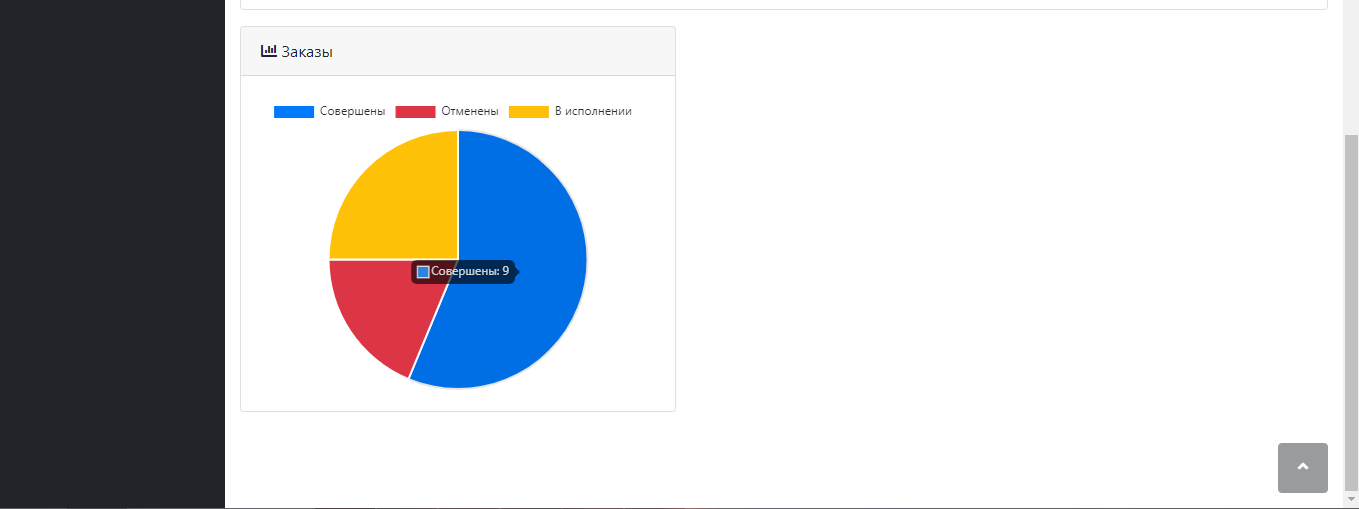
На форме «Типы домов» доступна информация о существующих типов домов, так же доступен следующий функционал:

1. Добавление нового типа дома
2. Изменение существующего типа дома
3. Удаление существующего типа дома при условии, что на данный момент нет домов с таким типов в таблице «Дома»

Далее рассмотрим графики(Рис 5.15 и Рис 5.16). Первый выводит количество заказов за сегодня и прошедшие 12 дней, второй – соотношение выполненных, отмененных и действующих заказов.



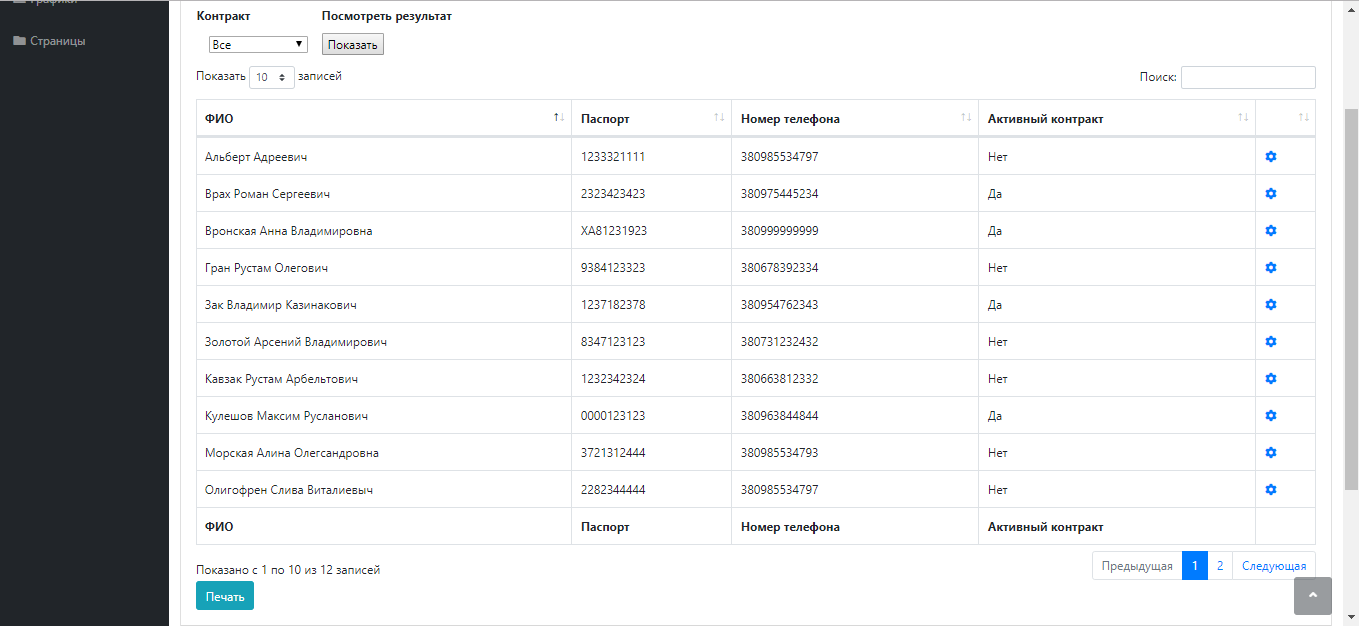
*Рисунок 5.15 – Окно графика отношений День(количество заказов)*



*Рисунок 5.16 – Окно графика отношений заказов.*

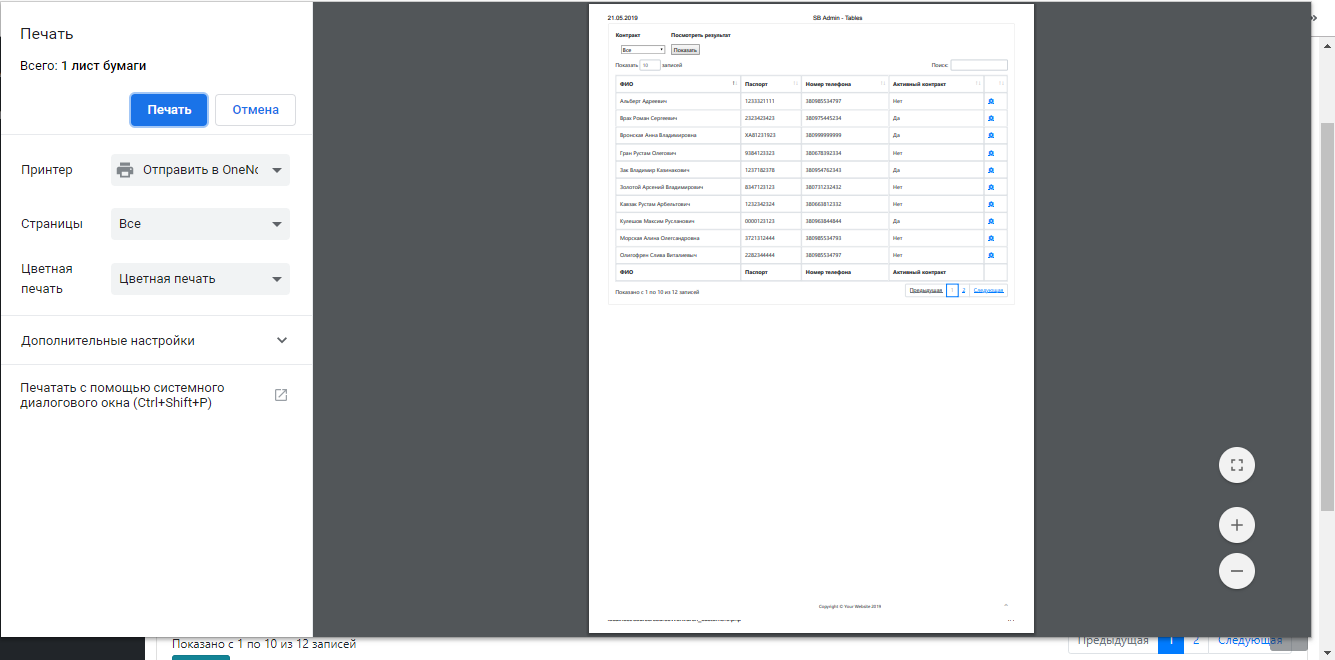
Перейдем к информации которую необходимо отправлять на печать:

На рисунке 5.17 изображен архив таблицы «Клиенты», на которой доступна функция «Печать».



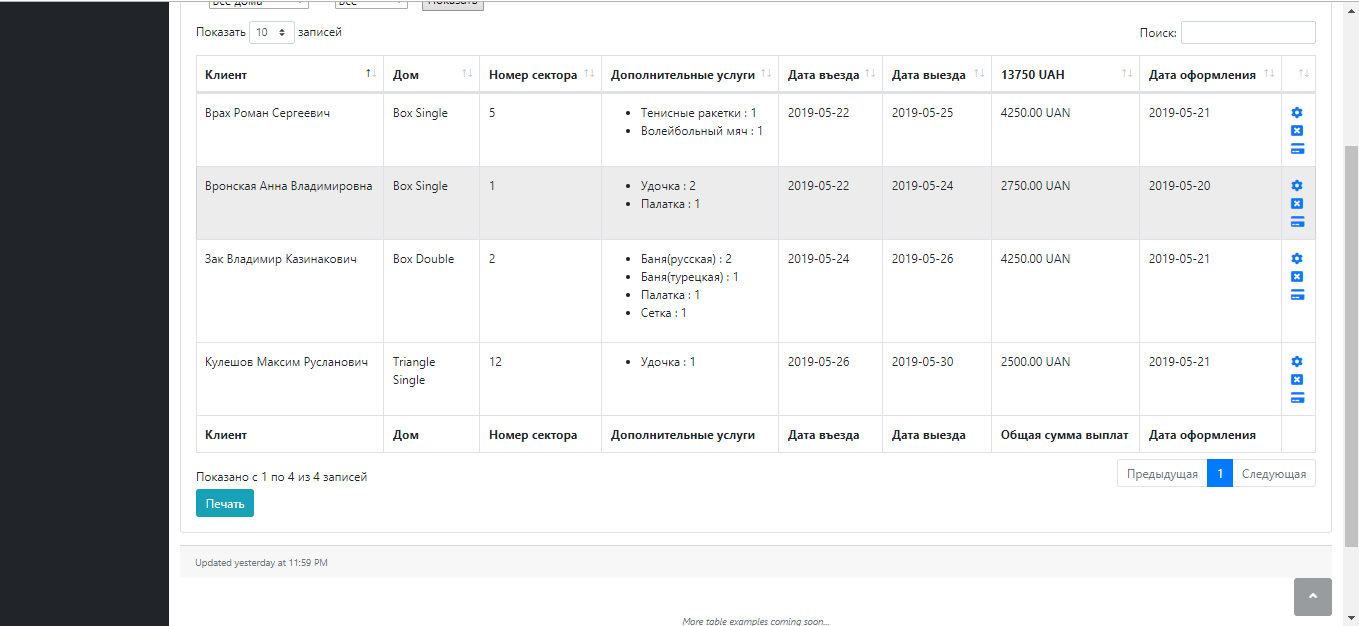
*Рисунок 5.17 – Окно вывода таблицы «Архив-клиенты»*

После нажатия кнопки печати нам предлагается выбрать нужные атрибуты и приступить непосредственно к печати.(Рис 5.18)



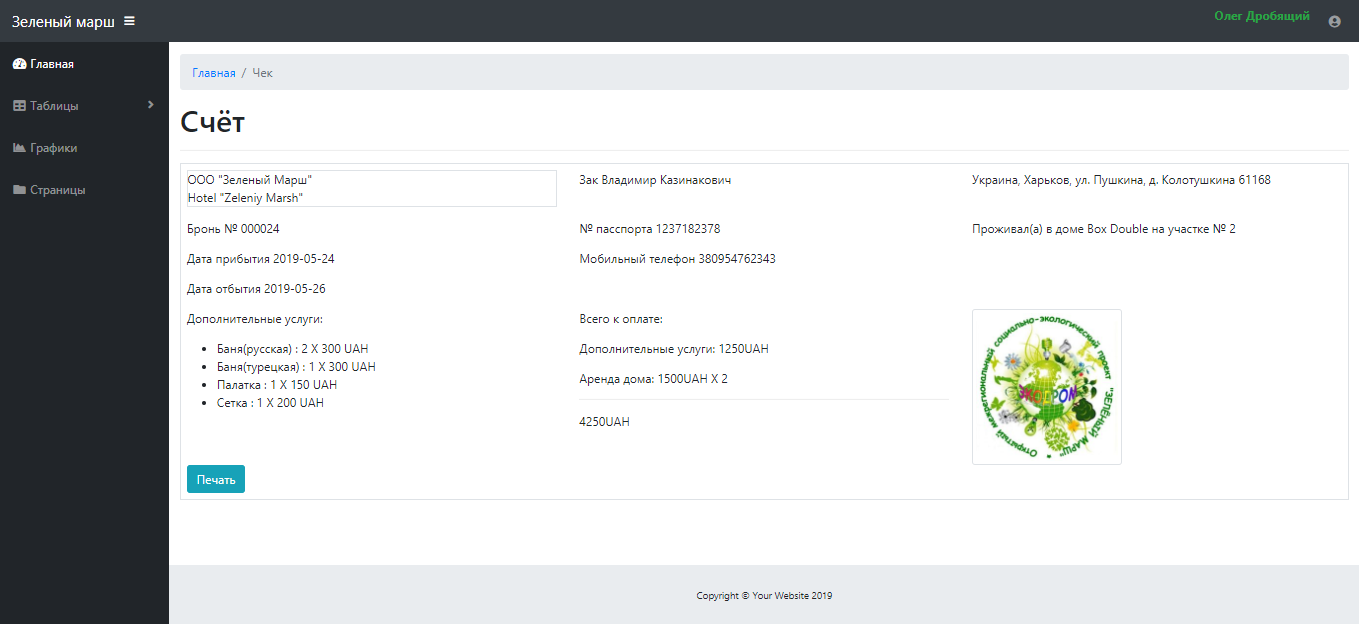
*Рисунок 5.18 – Окно для печати таблицы «Архив-клиенты».*

Перейдя во вкладку заказы мы можем наблюдать список действующих заказов(Рис 5.19), где у каждого столбца есть функция обозначенная «картой», которая формирует чек об оплате



*Рисунок 5.19 – Окно для печати таблицы «Архив-клиенты».*

Выбрав функцию формирования чека мы видим следующую картину(Рис 5.20)



*Рисунок 5.20 – Окно для печати таблицы «Архив-клиенты».*

# ВЫВОДЫ

В результате выполнения курсового проекта, были использованы новые для меня технологии разработки программного продукта такие как HTML,CSS, PHP, JavaScript. Был получен опыт создания базы данных, связывания всех компонентов воедино и сформировалось представление общей картины декомпозиции элементов и их синтеза. В ходе изобретения программного продукта, он кардинально изменялся в течении всего времени: вначале продукт создавался по методологии программирования «сломя голову» т.е. просто хаотичное написание кода без какой либо конкретной цели. После знакомства с новыми доступными возможностями был создан некий план по будущей реализации имеющихся идей. И наконец после создания основной части работы осталось провести тестирования и исправлять возникающие баги. К концу разработки ПО, возник ряд дополнительных идей, которые требуются для большей функциональности приложения. Все тонкости были учтены и предусмотрены для безопасного использования программного продукта. Для будущей реализации осталось добавить: свою кухню, разграничение уровней доступа, возможность нанимать тренера

# Список литературы

* Скляр, Дэвид. С43 Изучаем РНР 7: руководство по созданию интерактивных веб-сайтов. : Пер. с англ. — СпБ. : ООО “Альфа-книга”, 2017. — 464 с. : ил. — Парал. тит. англ. ISBN 978-5-9908462-3-4 (рус.)
* Кузнецов, М. В. К89 Самоучитель РНР 7 / М. В. Кузнецов, И. В. Симдянов. - СПб.: БХВ-Петербург, 2018. -448 с.: ил. - (Самоучитель) ISBN 978-5-9775-3817-
* Фрэйн Б. Ф86 HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. 2-е изд. — СПб.: Питер, 2017. — 272 с.: ил. — (Серия «Библиотека программиста»). ISBN 978-5-496-02271-