**Министерство образования Московской области**

**ГОУ ВО МО «Государственный гуманитарно-технологический университет» (ГГТУ)**

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**О Т Ч Ё Т**

**ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

по ПМ.11 Разработка, администрирование и защита баз данных

Обучающегося \_\_\_\_\_\_\_\_Демин Артем Дмитриевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ф.И.О.

Курс \_\_\_\_3\_\_\_\_\_ группа \_\_\_ИСП.18.1А\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность 09.02.07 Информационные системы и программирование \_\_\_

Место практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «IS-ART»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Период практики с 01.06.2021 г. по 28.06.2021 г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководители практики

от колледжа \_\_\_\_\_\_\_\_\_\_Пронина Алла Юрьевна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Селиверстова Ольга Михайловна \_\_\_\_\_\_\_\_\_\_\_\_\_

от организации\_\_\_\_\_\_ Колесников Кирилл Николаевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

МП

г. Ликино-Дулево

2021 г

**Содержание**

[1. Общие сведения о компании 3](#_Toc75480987)

[1.1 Структура организации 4](#_Toc75480988)

[Анализ материально-технической базы 5](#_Toc75480989)

[2.1 Состав программного обеспечения 5](#_Toc75480990)

[2.2 Состав технических средств 7](#_Toc75480991)

[2.3 Структура локальной сети предприятия 12](#_Toc75480992)

[3.1.1 Назначение разработки 13](#_Toc75480993)

[3.2. Проектирование предметной области 14](#_Toc75480994)

[3.2.1 Разработка диаграммы «Сущность-связь» 14](#_Toc75480995)

[3.2.2 Нормализация БД 14](#_Toc75480996)

[3.3 Разработка БД 17](#_Toc75480997)

[3.4 Создание таблиц 17](#_Toc75480998)

[3.5 Заполнение таблиц 19](#_Toc75480999)

[3.6 Реализация запросов 20](#_Toc75481000)

[3.7 Интерфейс приложения 20](#_Toc75481001)

[3.8 Средства защиты и администрирование БД 21](#_Toc75481002)

[3.9 Реализация прав пользователей 23](#_Toc75481003)

[Заключение 25](#_Toc75481004)

[Список используемой литературы 26](#_Toc75481005)

## **1.** **Общие сведения о компании**

Студия IS ART специализируется на создании сайтов по всей России и всеми сопровождающимися услугами. За всё время работы компания успела собрать большие базу довольных клиентов и портфолио работ. Компания предлагает выполнение работ по разработке сайтов по всей России под ключ, таким образом, клиент получает полностью готовый под его нужды проект с минимальным участием.

Для этих целей, грамотные специалисты, помогают преобразовать пожелания клиента по созданию сайта в грамотное техническое задание, с помощью которого будет создан именно тот сайт, который полностью бы охватывал все потребности.

Сферы деятельности:

* Создание и обслуживание сайтов;
* Разработка интернет-магазинов;
* SEO и трафик;
* Продвижение сайтов;
* Разработка ПО;
* Веб-программирование;
* Интернет ресурсы и сервисы;
* Интернет-агентство.

## **1.1 Структура организации**

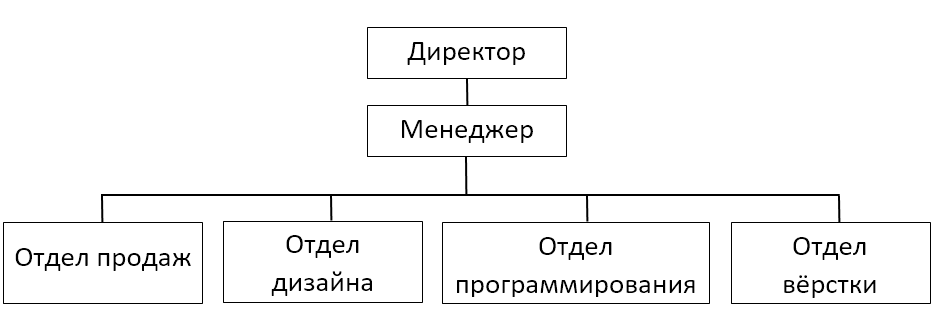


Рисунок . «Схема организации»

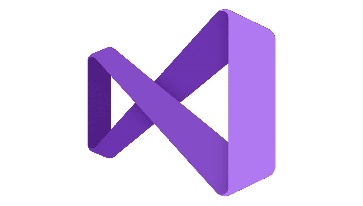
## **Анализ материально-технической базы**

## **2.1 Состав программного обеспечения**

**Microsoft Office** - Офисный пакет приложений, созданных корпорацией Microsoft для операционных систем Microsoft Windows, Windows Phone, Android, macOS, iOS. В состав этого пакета входит программное обеспечение для работы с различными типами документов: текстами, электронными таблицами, базами данных и др. Microsoft Office является сервером OLE-объектов и его функции могут использоваться другими приложениями, а также самими приложениями Microsoft Office. Поддерживает скрипты и макросы, написанные на VBA.

 **Windows 10** — операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT. После Windows 8.1 система получила номер 10, минуя 9. Серверный аналог Windows 10 — Windows Server 2016.

**Kaspersky Endpoint Security** - Антивирусное программное обеспечение, разрабатываемое Лабораторией Касперского. Предоставляет пользователю защиту от вирусов, троянских программ, шпионских программ, руткитов, adware, а также от неизвестных компоненту «файловый антивирус» угроз с помощью проактивной защиты, включающей компонент HIPS. Первоначально, в начале 1990-х, именовался -V, затем - AntiViral Toolkit Pro.

**Microsoft Visual Studio** — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

 **Django** — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation.

 **Django Rest Framework** (DRF) — это библиотека, которая работает со стандартными моделями Django для создания гибкого и мощного API для проекта.

## **2.2 Состав технических средств**

1. Принтер HP LaserJet 1018

|  |  |
| --- | --- |
| Принтер HP LaserJet 1018  **Основные характеристики** | |
| Размещение: | настольный |
| Устройство: | принтер |
| Тип печати: | черно-белый |
| Технология печати: | лазерный |
| **Печать** | |
| Максимальный формат: | A4 |
| Максимальное разрешение для черно белой печати: | 600x600 dpi |
| Скорость печати: | 12 стр/мин (ч/б А4) |
| Время выхода первого отпечатка: | 10 c (ч/б) |
| **Расходные материалы** | |
| Плотность бумаги: | 60-163 г/м2 |
| Печать на: | пленках, этикетках, глянцевой бумаге, конвертах, матовой бумаге |
| Ресурс ч/б картриджа/тонера: | 2000 страниц |
| Количество картриджей: | 1 |

2. Корпус Zalman T7

|  |  |
| --- | --- |
| https://c.dns-shop.ru/thumb/st1/fit/500/500/2a6f58207005578fa9bc3e75f28ce34f/7b2838cdc3eedff7582f4492a52a90519efbb8be173a2873b2a0786492357570.jpg  **Общие параметры** | |
| Модель | ZALMAN T7 |
| Год релиза | 2019 |
| **Форм-фактор и габариты** | |
| Типоразмер корпуса | Mid-Tower |
| Ориентация материнской платы | вертикально |
| Длина | 384 мм |
| Ширина | 202 мм |
| Высота | 438 мм |
| Вес | 3.2 кг |
| **Совместимость** | |
| Форм-фактор совместимых плат | Micro-ATX, Standard-ATX, Mini-ITX |
| Размещение блока питания | нижнее |
| Форм-фактор совместимых блоков питания | ATX |
| Горизонтальные слоты расширения | 7 |
| Максимальная длина блока питания | 150 мм |
| Вертикальные слоты расширения | нет |
| Количество отсеков 2.5" накопителей | 6 шт |
| Максимальная длина устанавливаемой видеокарты | 305 мм |
| Число внутренних отсеков 3.5" | 2 шт |
| Число внешних отсеков 3.5" | нет |
| Число отсеков 5.25" | нет |
| Максимальная высота процессорного кулера | 160 мм |
| **Охлаждение** | |
| Вентиляторы в комплекте | 2 x 120 мм |
| Поддержка фронтальных вентиляторов | 3 x 120 мм |
| Поддержка тыловых вентиляторов | 1 x 120 мм |
| Поддержка верхних вентиляторов | 2 x 120 мм |
| Поддержка нижних вентиляторов | 2 x 120 мм |
| Поддержка боковых вентиляторов | нет |
| Возможность установки системы жидкостного охлаждения | нет |
| Верхний монтажный размер радиатора СВО | 240 мм |
| **Внешний вид** | |
| Основной цвет | черный |
| Материал корпуса | пластик , сталь |
| Толщина металла | 0.4 мм |
| Наличие окна на боковой стенке | слева |
| Материал окна | оргстекло |
| Материал фронтальной панели | пластик |
| Тип подсветки | нет |
| Цвет подсветки | нет |
| Источник подсветки | нет |
| Разъем подключения подсветки | нет |
| Способ управления подсветкой | нет |
| **Разъемы и интерфейсы лицевой панели** | |
| Расположение I/O панели | сверху |
| Типы USB-портов на панели | USB 3.2 Gen1 Type-A, USB 2.0 Type-A |
| Разъемы | 2 x 3.5 Jack, 1 x USB 3.2 Gen1 Type-A, 2 x USB 2.0 Type-A |
| Встроенный кард-ридер | нет |

3. Сервер Dell PowerEdge R740 (210-AKXJ-167)

|  |  |
| --- | --- |
| https://static.onlinetrade.ru/img/items/b/server_dell_poweredge_r740_210_akxj_167__1235530_1.png  **Общие** | |
| Тип корпуса | 2U |
| Код процессора | Silver 4116 |
| Количество ядер | 12 |
| Установленный объем ОЗУ | 384 (12 х 32) Гб |
| Установлено блоков питания | 2 шт |
| Тип процессора | Intel Xeon |
| Частота процессора | 2100 МГц |
| Установлено процессоров | 1 |
| Количество слотов ОЗУ | 24 шт |
| Мощность блока питания | 750 Вт |

4. Wi-Fi роутер TP-LINK Archer C6

|  |  |
| --- | --- |
| Wi-Fi роутер TP-LINK Archer C6 — купить по выгодной цене на Яндекс.Маркете  **Параметры Wi-Fi** | |
| Wi-Fi | есть |
| Стандарт Wi-Fi | 5 (802.11ac), 4 (802.11n) |
| Класс Wi-Fi | AC1200 |
| Максимальная скорость по частоте 2.4 ГГц | 300 Мбит/с |
| Максимальная скорость по частоте 5 ГГц | 867 Мбит/с |
| Одновременная работа в двух диапазонах | есть |
| Многопотоковая передача данных | MU-MIMO |
| Мощность передатчика | 20 dBm |
| Тип и количество антенн | внешняя несъемная x4, внутренняя x1 |
| Коэффициент усиления антенны | 5 dBi |
| Безопасность соединения | WPA-PSK/WPA2-PSK , WPA/WPA2-Enterprise , WEP |
| **Порты** | |
| Количество LAN портов | Количество LAN портов |
| 4 | 4 |
| Скорость передачи по проводному подключению | Скорость передачи по проводному подключению |
| 1000 Мбит/с | 1000 Мбит/с |
| Количество SFP портов | Количество SFP портов |

## **2.3 Структура** **локальной сети предприятия**

На предприятии: «IS ART» данные передаются, в основном, с помощью интернета. Локальной сетью объединены только 2 компьютера, остальная информация передается посредством интернета и сервиса «GitHub».

Топология общая шина предполагает использование одного кабеля, к которому подключаются все компьютеры сети. Отправляемое какой-либо рабочей станцией сообщение распространяется на все компьютеры сети. Каждая машина проверяет кому адресовано сообщение, — если сообщение адресовано ей, то обрабатывает его.

Шина самой своей структурой допускает идентичность сетевого оборудования компьютеров, а также равноправие всех абонентов. При таком соединении компьютеры могут передавать информацию только по очереди, — последовательно — потому что линия связи единственная. В противном случае пакеты передаваемой информации будут искажаться в результате взаимного наложения (т. е. произойдет конфликт, коллизия). Таким образом, в шине реализуется режим полудуплексного (half duplex) обмена (в обоих направлениях, но по очереди, а не одновременно (т. е. последовательно, а не параллельно)).

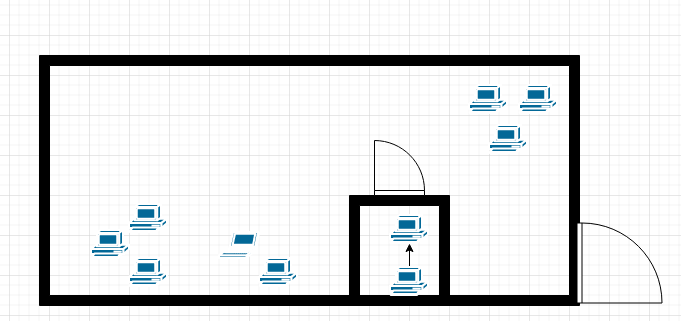


Рисунок . «Схема офиса компании»

# **3. Выполнение индивидуального задания**

## **3.1 Разработка технического задания**

# **3.1.1 Назначение разработки**

Приложение автоматизирует работу менеджера компании IS-ART с проектами и подчиненными сотрудниками. Выполняет следующие функции: Добавление нового сотрудника, добавление проекта, добавление этапа, добавление задачи, просмотр проектов, этапов, задач и сотрудников.

**3.1.2 Требования к функциональным характеристикам:**

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

* Добавление проекта
* Добавление этапов проекта
* Добавление задач этапов
* Добавление нового сотрудника
* Занесение проектов в архив
* Просмотр проектов, этапов, задач и сотрудников.

**Требования к организации входных данных:**

Входные данные программы должны быть организованы в виде вводимого в специальную форму текста, соответствующего определенному шаблону.

**Требования к организации выходных данных:**

Выходные данные программы должны быть организованы в виде ответа на сервер формата json.

**Требования к численности и квалификации персонала:**

Количество персонала, требуемое для работы программы, составляет всех сотрудников – менеджер, программист, дизайнер и др.

Сотрудник не обязан быть опытным пользователем ПК, но ему необходимо знать базовые вещи в работе с компьютером.

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

Минимальные характеристики компьютера

Процессор От 2 ГГц;

Память От 2 Гб;

Разрешение экрана От 1600x800;

Размер экрана От 5″;

Устройства ввода Клавиатура, мышь

Дисковое пространство От 2 Гб

Оптимальные характеристики компьютера

|  |  |
| --- | --- |
| Процессор | Intel Core i5 3ГГц |
| Память | От 8 гб |
| Разрешение экрана | От 1920x1080 |
| Устройства ввода | Клавиатура, мышь |
| Дисковое  пространство | От 1 Гб |
| Дополнительно | Постоянное подключение к  сети, в которой находится база  данных |

**Требования к информационной и программной совместимости:**

Системные программные средства, используемые программой, должны быть представлены операционной системы Windows7 и более поздними версиями.

**Требования к программной документации:**

В ходе разработки программы должны быть подготовлены следующие программные документы: текст программы, описание программы, программа и методика испытаний, руководство пользователя, руководство программиста, технико-экономическое обоснование.

# **3.2. Проектирование предметной области**

# **3.2.1 Разработка диаграммы «Сущность-связь»**

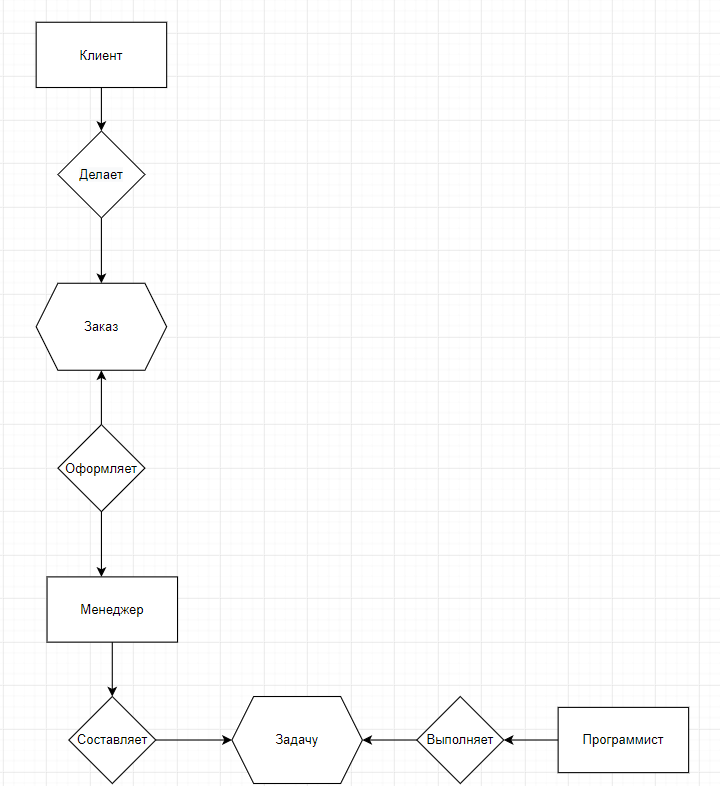


Рисунок . «Сущность-связь»

# **3.2.2 Нормализация БД**

Нормализация базы данных - это процесс структурирования реляционной базы данных в соответствии с серией так называемых нормальных форм, чтобы уменьшить избыточность данных и улучшить целостность данных. Впервые он был предложен Эдгаром Ф. Коддом как часть его реляционной модели.

Нормализация предполагает организацию столбцов (атрибутов) и таблиц (отношений) базы данных, чтобы гарантировать, что их зависимости должным образом обеспечиваются ограничениями целостности базы данных. Это достигается путем применения некоторых формальных правил в процессе синтеза (создание нового дизайна базы данных) или декомпозиции (улучшение существующего дизайна базы данных).

Первая нормальная форма гласит, что таблица базы данных – это представление сущности вашей системы, которую вы создаете. Примеры сущностей: заказы, клиенты, заказ билетов, отель, товар и т.д. Каждая запись в базе данных представляет один экземпляр сущности. Например, в таблице клиентов каждая запись представляет одного клиента.

Для того, чтобы база данных была нормализована согласно второй нормальной форме, она должна быть нормализована согласно первой нормальной форме. Вторая нормальная форма связана с избыточностью данных.

Третья нормальная форма связана с транзитивными зависимостями. Транзитивные зависимости между полями базы данных существует тогда, когда значения не ключевых полей зависят от значений других не ключевых полей. Чтобы база данных была в третьей нормальной форме, она должна быть во второй нормальной форме.

**1 Нормальная форма**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Имя исполнителя | Пароль исполнителя | Роль исполнителя | Название проекта | Описание проекта | Название Этапа | Статус Этапа | Название Задачи | Статус Задачи |
| Alex | Hesh | Manager | Краска | Some Text | First | False | Написать ТЗ | False |
| Sem | Hesh | Manager | Краска | Some Text | Second | False | Обговорить дизайн с клиентом | False |

Таблица «1НФ»

**2 Нормальная форма**

Таблица 2 «Исполнитель»

|  |  |  |  |
| --- | --- | --- | --- |
| Код исполнителя | Имя исполнителя | Пароль исполнителя | Роль исполнителя |
| 1 | Alex | Hesh | Manager |
| 2 | Sem | Hesh | Executor |

Таблица 3 «Проект»

|  |  |  |
| --- | --- | --- |
| Код проекта | Название проекта | Описание проекта |
| 1 | Краска | Some text |
| 2 | Стой-дом | Some text |

Таблица 4 «Этап»

|  |  |  |
| --- | --- | --- |
| Код этапа | Название этапа | Состояние этапа |
| 1 | First | False |
| 2 | Second | False |

Таблица 5 «Задача»

|  |  |  |
| --- | --- | --- |
| Код задачи | Название задачи | Состояние задачи |
| 1 | Написать ТЗ | False |
| 2 | Обговорить дизайн с клиентом | False |

**3 Нормальна форма**

Таблица 6 «Этапы»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код этапа | Название этапа | Состояние этапа | Код исполнителя | Код проекта |
| 1 | First | False | 1 | 1 |
| 2 | Second | False | 2 | 2 |

Таблица 7 «Задачи»

|  |  |  |  |
| --- | --- | --- | --- |
| Код задачи | Название задачи | Состояние задачи | Код этапа |
| 1 | Написать ТЗ | False | 1 |
| 2 | Обговорить дизайн с клиентом | False | 2 |

# **3.3 Разработка БД**

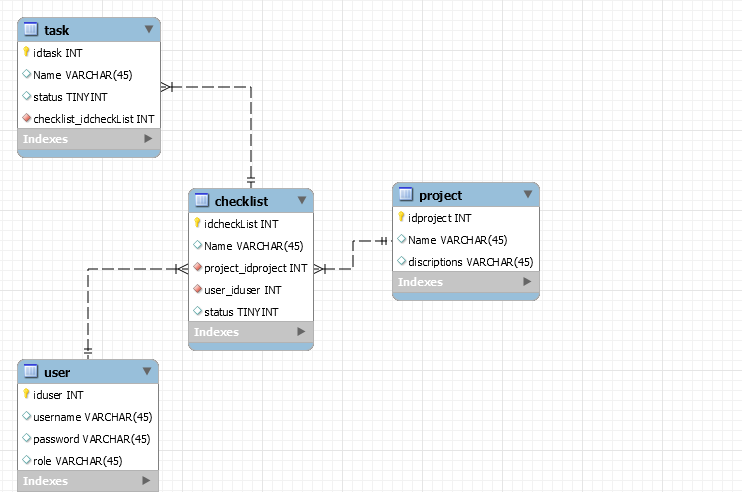


Рисунок . «Схема данных»

# **3.4 Создание таблиц**

Для работы с таблицами сервера были разработаны классы, которые имеют полный список атрибутов таблиц.

**Проект**

*class Project(models.Model):*

*projectName = models.CharField("Название", max\_length=150)*

*description = models.CharField("Опсианеи", max\_length=150, default='SOME STRING')*

*def \_\_str\_\_(self):*

*return f'{self.id}\_{self.projectName}'*

*class Meta:*

*verbose\_name = "Проект"*

*verbose\_name\_plural = "Проекты"*

**Этапы**

*class CheckList(models.Model):*

*stepNumber = models.CharField("Номер", max\_length=100)*

*project = models.ForeignKey(Project, related\_name="check\_lists", on\_delete=models.SET\_NULL, null=True)*

*user = models.ForeignKey(CostumUser, related\_name="ch\_user", on\_delete=models.SET\_NULL, null=True)*

*CheckListStatus = models.BooleanField(default=False)*

*def \_\_str\_\_(self):*

*return f'{self.id}\_{self.stepNumber}'*

*class Meta:*

*verbose\_name = "Чек-лист"*

*verbose\_name\_plural = "Чек-листы"*

**Задачи**

*class Task(models.Model):*

*taskName = models.CharField("Задача", max\_length=100)*

*taskStatus = models.BooleanField(default=False)*

*checkList = models.ForeignKey(CheckList, verbose\_name="Чек лист", on\_delete=models.CASCADE,*

*related\_name="task\_list")*

*def \_\_str\_\_(self):*

*return f'{self.id}'*

*class Meta:*

*verbose\_name = "Задача"*

*verbose\_name\_plural = "Задачи"*

**Пользователь**

*class CostumUser(AbstractUser):*

*role = models.CharField("Роль", max\_length=100)*

*def save(self, \*args, \*\*kwargs):*

*super(CostumUser, self).save(\*args, \*\*kwargs)*

*permissions = []*

*if self.role == "Executor":*

*permissions = [*

*Permission.objects.get(name='Can view Чек-лист'),*

*Permission.objects.get(name='Can change Задача'),*

*]*

*elif self.role == "Meneger":*

*permissions = [*

*Permission.objects.get(name='Can view Проект'),*

*Permission.objects.get(name='Can add Проект'),*

*Permission.objects.get(name='Can change Проект'),*

*Permission.objects.get(name='Can delete Проект'),*

*Permission.objects.get(name='Can view Чек-лист'),*

*Permission.objects.get(name='Can add Чек-лист'),*

*Permission.objects.get(name='Can change Чек-лист'),*

*Permission.objects.get(name='Can delete Чек-лист'),*

*Permission.objects.get(name='Can view Задача'),*

*Permission.objects.get(name='Can add Задача'),*

*Permission.objects.get(name='Can change Задача'),*

*Permission.objects.get(name='Can delete Задача'),*

*Permission.objects.get(name='Can view user'),*

*Permission.objects.get(name='Can add user'),*

*Permission.objects.get(name='Can delete user'),*

*Permission.objects.get(name='Can change user'),*

*]*

*self.user\_permissions.set(permissions)*

# **3.5 Заполнение таблиц**

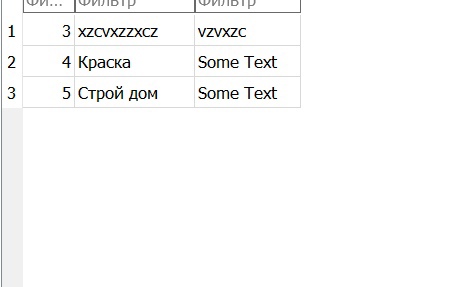


Рисунок . «Заполнение таблицы проекты»

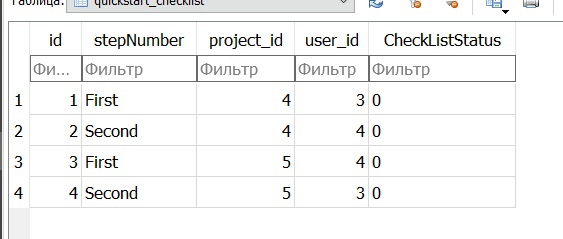


Рисунок . «Заполнение таблицы этапы»

# **3.6 Реализация запросов**

**Вывод проектов:**

*class ProjectViewSet(viewsets.ModelViewSet):*

*pagination\_class = None*

*permission\_classes = [*

*permissions.IsAuthenticated,*

*permissions.DjangoModelPermissions,*

*]*

*serializer\_class = ProjectSerializer*

*def get\_queryset(self):*

*user = self.request.user*

*queryset = Project.objects.all()*

*to\_be\_deleted = []*

*for p in queryset:*

*check\_lists\_status = []*

*for cl in p.check\_lists.all():*

*if all(t.taskStatus for t in cl.task\_list.all()):*

*check\_lists\_status.append(True)*

*else:*

*check\_lists\_status.append(False)*

*if len(CheckList.objects.all()) != 0:*

*if all(check\_lists\_status):*

*to\_be\_deleted.append(p.id)*

*queryset.filter(id\_\_in=to\_be\_deleted).delete()*

*if user.role == "Executor":*

*queryset = Project.objects.filter(id\_\_in=CheckList.objects.values("project\_id").filter(*

*user\_id\_\_in=CostumUser.objects.filter(id=self.request.user.id)))*

*return queryset*

**Вывод этапов:**

*class CheckListViewSet(viewsets.ModelViewSet):*

*pagination\_class = None*

*queryset = CheckList.objects.all()*

*permission\_classes = [permissions.IsAuthenticated, permissions.DjangoModelPermissions]*

*serializer\_class = CheckListSerializer*

*def get\_queryset(self):*

*queryset = CheckList.objects.all()*

*to\_be\_deleted = []*

*for cl in queryset:*

*if len(cl.task\_list.all()) != 0:*

*if all(t.taskStatus for t in cl.task\_list.all()):*

*to\_be\_deleted.append(cl.id)*

*queryset.filter(id\_\_in=to\_be\_deleted).delete()*

*user = self.request.user*

*if user.role == "Meneger" or user.role == "Admin":*

*queryset = CheckList.objects.all()*

*elif user.role == "Executor":*

*queryset = CheckList.objects.filter(user=self.request.user)*

*return queryset*

**Вывод пользователей:**

*class UserViewSet(viewsets.ModelViewSet):*

*"""*

*API endpoint that allows users to be viewed or edited.*

*"""*

*queryset = CostumUser.objects.all().order\_by('-date\_joined')*

*pagination\_class = None*

*serializer\_class = UserSerializer*

*permission\_classes = [permissions.IsAuthenticated, permissions.DjangoModelPermissions]*

*def get\_queryset(self):*

*queryset = CostumUser.objects.all().order\_by('-date\_joined')*

*user = self.request.user*

*if (user.role == "Executor"):*

*queryset = CostumUser.objects.filter(id=self.request.user.id)*

*return queryset*

# **3.7 Интерфейс приложения**

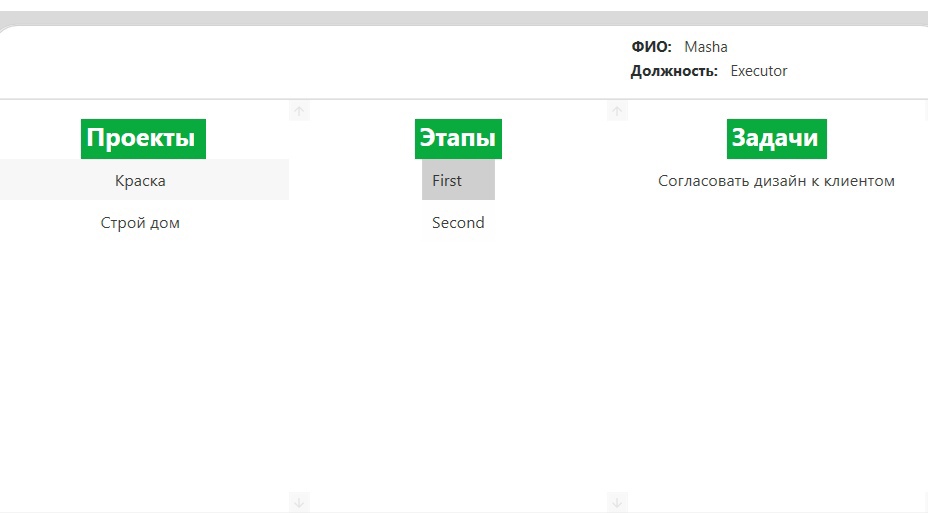


Рисунок . «Меню пользователя»

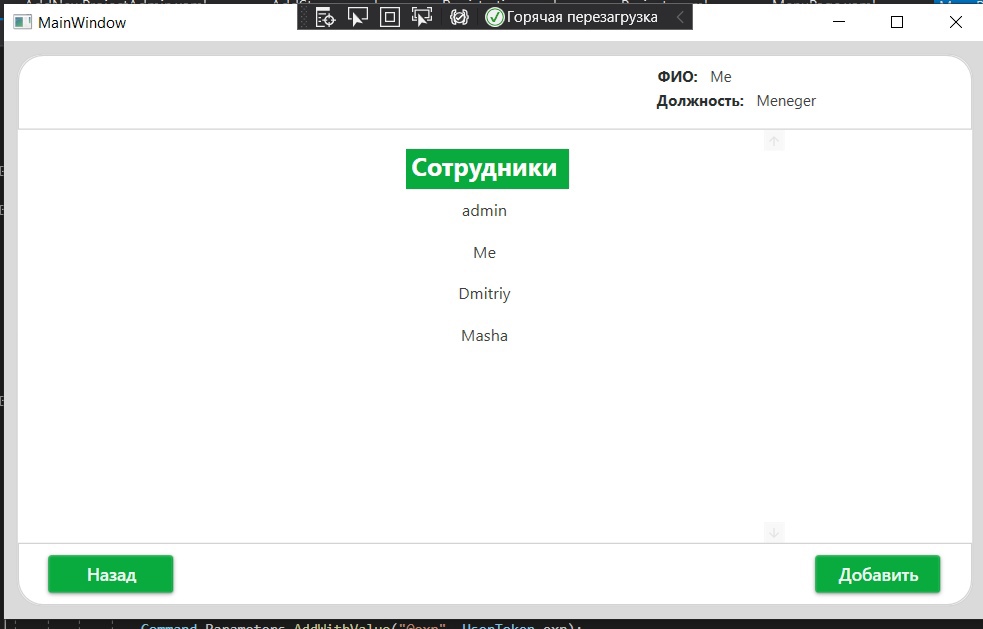


Рисунок . «Окно просмотра пользователей»

# **3.8 Средства защиты и администрирование БД**

Методы защиты баз данных: защита паролем, шифрование, разграничение прав доступа

Методы защиты баз данных в различных СУБД несколько отличаются друг от друга. Анализ современных СУБД показывает, что они условно делятся на две группы: основные и дополнительные.

**К** **основным** средствам зашиты относится:

* зашита паролем:
* шифрование данных и программ:
* разграничение прав доступа к объектам базы данных:
* защита полей и записей таблиц БД.

Защита паролем представляет собой простой и эффективный способ зашиты БД от несанкционированного доступа. Пароли устанавливаются пользователями или администраторами БД. Учет и хранение паролей выполняется самой СУБД. Обычно, пароли хранятся в определенных системных файлах СУБД в зашифрованном виде. После ввода пароля пользователю СУБД предоставляются все возможности по работе с БД. Парольная зашита является достаточно слабым средством, особенно если пароль не шифруется. Основной ее недостаток состоит в том, что все пользователи, использующие одинаковый пароль, с точки зрения вычислительной системы неразличимы. Неудобство парольной зашиты для пользователя состоит в том. что пароль надо запоминать или записать. При небрежном отношении к записям пароль может стать достоянием других. Более мощным средством зашиты данных от просмотра является их шифрование. Шифрование — это преобразование читаемого текста в нечитаемый текст. При помощи некоторого алгоритма: применяется для зашиты уязвимых данных. Процесс дешифрования восстанавливает данные в исходное состояние. В целях контроля использования основных ресурсов СУБД во многих системах имеются средства установления прав доступа к объектам БД. Права доступа определяют возможные действия над объектами. Владелец объекта (пользователь, создавший объект). а также администратор БД имеют все права. Остальные пользователи к разным объектам могут иметь различные уровни доступа. Разрешение на доступ к конкретным объектам базы данных сохраняется в файле рабочей группы. Файл рабочей группы содержит данные о пользователях группы и считывается во время запуска. Файл содержит следующую информацию: имена учетных записей пользователей, пароли пользователей, имена групп. в которые входят пользователи. По отношению к таблицам могут предусматриваться следующие права доступа:

* просмотр (чтение) данных:
* изменение (редактирование) данных:
* добавление новых записей:
* добавление и удаление данных:
* изменение структуры таблицы.

К данным, имеющимся в таблице, могут применяться меры защиты по отношению к отдельным полям и отдельным записям. Защита данных в полях таблиц предусматривает следующие уровни прав доступа:

* полный запрет доступ:
* только чтение:
* разрешение всех операций (просмотр. ввод новых значений, удаление и изменение).

К **дополнительным** средствам защиты БД можно отнести такие, которые нельзя прямо отнести к средствам зашиты, но которые непосредственно влияют на безопасность данных. Их составляют следующие средства:

* встроенные средства контроля значений данных в соответствии с типами:
* повышения достоверности вводимых данных:
* обеспечения целостности связей таблиц.

# **3.9 Реализация прав пользователей**

В приложении защита выполнена в виде логина и пароля. Пароль дополнительно проходит хеширование. Если введенные данные совпадают с данными в БД, пользователь войдёт в аккаунт.

Код выдачи прав пользователю:

*class CostumUser(AbstractUser):*

*role = models.CharField("Роль", max\_length=100)*

*def save(self, \*args, \*\*kwargs):*

*super(CostumUser, self).save(\*args, \*\*kwargs)*

*permissions = []*

*if self.role == "Executor":*

*permissions = [*

*Permission.objects.get(name='Can view Чек-лист'),*

*Permission.objects.get(name='Can change Задача'),*

*]*

*elif self.role == "Meneger":*

*permissions = [*

*Permission.objects.get(name='Can view Проект'),*

*Permission.objects.get(name='Can add Проект'),*

*Permission.objects.get(name='Can change Проект'),*

*Permission.objects.get(name='Can delete Проект'),*

*Permission.objects.get(name='Can view Чек-лист'),*

*Permission.objects.get(name='Can add Чек-лист'),*

*Permission.objects.get(name='Can change Чек-лист'),*

*Permission.objects.get(name='Can delete Чек-лист'),*

*Permission.objects.get(name='Can view Задача'),*

*Permission.objects.get(name='Can add Задача'),*

*Permission.objects.get(name='Can change Задача'),*

*Permission.objects.get(name='Can delete Задача'),*

*Permission.objects.get(name='Can view user'),*

*Permission.objects.get(name='Can add user'),*

*Permission.objects.get(name='Can delete user'),*

*Permission.objects.get(name='Can change user'),*

*]*

*self.user\_permissions.set(permissions)*

Код авторизации:

*class UserSerializer(serializers.ModelSerializer):*

*password = serializers.CharField(*

*write\_only=True,*

*required=True,*

*style={'input\_type': 'password', 'placeholder': 'Password'}*

*)*

*password\_confirm = serializers.CharField(*

*write\_only=True,*

*required=True,*

*style={'input\_type': 'password', 'placeholder': 'Password confirm'}*

*)*

*def create(self, validated\_data):*

*password = validated\_data.get('password')*

*validated\_data['password'] = make\_password(password)*

*del validated\_data['password\_confirm']*

*return super(UserSerializer, self).create(validated\_data)*

*class Meta:*

*model = CostumUser*

*fields = ['id', 'username', 'password', 'password\_confirm', 'role']*

# **Заключение**

Производственную практику проходил в организации «IS-ART». Данная организация занимается создание и продвижением сайтов.

Проведен анализ материально-технической базы, который показал, что в организации находится 6 компьютеров. Установлено следующее программное обеспечение: Операционная система Windows 10. MS Office, Postman, Photoshop, Figma.

Материально-техническая база соответствует поставленной задаче.

В соответствии с постановкой задачи было разработано техническое задание. Согласно техническому заданию разработана Приложение автоматизирует работу сотрудника отдела продаж с клиентами и заказами. Выполняет следующие функции:

* добавление проектов;
* внесение проектов в архив;
* редактирование записей;

В техническом задании отражены требования к составу технических средств, информационно-программной совместимости и требования к надёжности которые должны быть соблюдены во время разработки, внедрении и эксплуатации приложения.

На основании собранной информации в программе Visio спроектирован пользовательский интерфейс программы и функциональная схема.

Для разработки приложения использовались следующие средства: Microsoft Visual Studio, PyCharm.

Приложение обладает удобным графическим интерфейсом, имеется главное меню с основным функционалом.

Для отладки и тестирование приложения использовались встроенные программные средства языка Python.

# **Список используемой литературы**

1. Илюшечкин, В. М. Основы использования и проектирования баз данных: учебник для среднего профессионального образования /В. М. Илюшечкин. — испр. и доп. — Москва: Издательство Юрайт, 2019. — 213 с. — (Профессиональное образование). — ISBN 978-5-534-01283-5. — Текст: электронный //ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/437670
2. Нестеров, С. А. Базы данных: учебник и практикум для среднего профессионального образования / С. А. Нестеров. — Москва: Издательство Юрайт, 2020. — 230 с. — (Профессиональное образование). — ISBN 978-5-534-11629-8. — URL: https://urait.ru/bcode/457142
3. Федорова Г.Н. Основы проектирования баз данных: учебник для студентов СПО /Г.Н. Федорова.- 2-е изд., стер.– Москва: Академия, 2018.– 219с.
4. Федорова Г.Н. Разработка, администрирование и защита баз данных.: учебник для среднего профессионального образования / Федорова Г.Н — Москва : Издательство Академия, 2020. — 288 с. — (Профессиональное образование).
5. Маркин, А. В. Программирование на SQL : учебное пособие для среднего профессионального образования / А. В. Маркин. — Москва : Издательство Юрайт, 2020. — 435 с. — (Профессиональное образование). — ISBN 978-5-534-11093-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/456926
6. Стружкин, Н. П. Базы данных: проектирование: учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва: Издательство Юрайт, 2020. — 477 с. — (Профессиональное образование). — ISBN 978-5-534-11635-9. — URL: https://urait.ru/bcode/457135
7. Гордеев, С. И. Организация баз данных в 2 ч. Часть 1: учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2020. — 310 с. — (Профессиональное образование). — ISBN 978-5-534-11626-7. — URL: https://urait.ru/bcode/457145
8. Гордеев, С. И. Организация баз данных в 2 ч. Часть 2: учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2020. — 513 с. — (Профессиональное образование). — ISBN 978-5-534-11625-0. — URL: https://urait.ru/bcode/457146
9. Стасышин, В. М. Базы данных: технологии доступа: учебное пособие для среднего профессионального образования / В. М. Стасышин, Т. Л. Стасышина. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 164 с. — (Профессиональное образование). — ISBN 978-5-534-09888-4. — URL : https://urait.ru/bcode/45586
10. Интернет-ресурсы
11. ЭБС «Университетская библиотека online» - http://biblioclub.ru/
12. ЭБС «ЮРАЙТ» - https://urait.ru/