|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_«Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_ «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОМУ ПРОЕКТУ***

***НА ТЕМУ:***

***Магазин электронной техники «E-Tech»***

Студент \_\_\_\_\_ИУ7-66б\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_**Гасанзаде М.А.\_\_

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсового проекта **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_**Филиппов М.В.**\_\_**

(Подпись, дата) (И.О.Фамилия)

Консультант **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2020 г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_ИУ7\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_И.В. Рудаков

(И.О.Фамилия)

« \_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**ЗАДАНИЕ**

**на выполнение курсового проекта**

по дисциплине \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Базы данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент группы \_ ИУ7-66б\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Гасанзаде Мухаммедали Алиназим оглы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Тема курсового проекта \_\_\_\_Магазин электронной техники «E-Tech»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Направленность КП (учебный, исследовательский, практический, производственный, др.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_учебный\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Источник тематики (кафедра, предприятие, НИР) \_\_\_\_\_\_\_\_\_\_\_\_\_\_кафедра\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

График выполнения проекта: 25% к \_\_\_ нед., 50% к \_\_\_ нед., 75% к \_\_ нед., 100% к \_\_\_ нед.

***Задание:*** разработать web-приложение (интернет-магазин), занимающийся продажей электронной техники и комплектующих. Организовать поддержку добавления, хранения, редактирования, и

удаления информации о клиентах/работниках/товарах. Поиск по фильтру/названию товара, а также позволять вводить информацию о проданном товаре в журнал продаж.

***Оформление курсового проекта:***

Расчетно-пояснительная записка на 40-50 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Расчетно-пояснительная записка должна содержать постановку введение, аналитическую часть, конструкторскую часть, технологическую часть, экспериментально-исследовательский раздел, заключение, список литературы, приложения.

На защиту проекта должна быть представлена презентация, состоящая из 15-20 слайдов. На слайдах должны быть отражены: постановка задачи, использованные методы и алгоритмы, расчетные соотношения, структура комплекса программ, таблица классов, интерфейс, характеристики разработанного ПО, результаты проведенных исследований.

Дата выдачи задания «20» апреля 2020 г.

**Руководитель курсового проекта**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Филиппов М.В.\_\_

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_**Гасанзаде М.А.**\_**

(Подпись, дата) (И.О.Фамилия)

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 4](#_Toc41478854)

[I. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ. 5](#_Toc41478855)

[1.1. Описание предметной области 5](#_Toc41478856)

[1.2. Выделение понятий предметной области в составление глоссария 9](#_Toc41478857)

[1.3. Предварительная требования к информационной системе 12](#_Toc41478858)

[1.4. Составление диаграммы прецедентов 14](#_Toc41478859)

[1.5. Описание вариантов пользования 15](#_Toc41478860)

[1.6. Составление подробного словесного описания вариантов использования 17](#_Toc41478861)

[1.7. Упрощенная диаграмма последовательности 20](#_Toc41478862)

[1.8. Составление диаграммы классов 23](#_Toc41478863)

[1.9. Разработка схемы базы данных 24](#_Toc41478864)

[II. КОНСТРУКТОРСКАЯ ЧАСТЬ 26](#_Toc41478865)

[2.1. Техническое задание 26](#_Toc41478866)

[2.1.1. Введение 26](#_Toc41478867)

[2.1.2. Основания для разработки 26](#_Toc41478868)

[2.1.3. Назначение разработки 26](#_Toc41478869)

[2.1.4. Требования к программе или программному изделию 26](#_Toc41478870)

[2.1.5. Требования к программной документации 27](#_Toc41478871)

[2.1.6. Технико-экономические показатели 27](#_Toc41478872)

[2.1.7. Стадии и этапы разработки 27](#_Toc41478873)

[2.1.8. Порядок контроля; 28](#_Toc41478874)

[2.2. Архитектура программного обеспечения 28](#_Toc41478875)

[2.3. Выбор среды и языка разработки серверной части 28](#_Toc41478876)

[2.4. Используемые инструменты и технологии веб-приложения 30](#_Toc41478877)

[2.5. Реализация 30](#_Toc41478878)

[2.5.1. Организация просмотра истории заказов 38](#_Toc41478879)

[ЗАКЛЮЧЕНИЕ 41](#_Toc41478880)

[ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА 42](#_Toc41478881)

# ВВЕДЕНИЕ

**Актуальность** – В настоящее время базы данных повсеместно используются не только в большом, но и в малом бизнесе, т.к. значительно упрощают контроль и счёт товаров. Хоть и ранее, они использовались лишь крупными корпорациями и в вычислительных средах. База данных облегчает работу с огромной информацией, необходимой для организации интернет-магазина.

**Цель данной работы** – Решить проблемы предметной области, такие как:

1. Проблема надежности хранения данных.
2. Использование большого объёма бумажных документов.
3. Трата большого количества времени на заполнение документов и их поиск.

**Возможности системы –** Основными возможностями системы будут являться:

* ввод и хранение информации о товарах в магазине;
* ввод и хранение журналов продаж;
* предоставление отчетов о поступлении товаров, продаже товаров, а также о выручке и прибыли.

# I. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ.

## 1.1. Описание предметной области

Магазин электронной техники “E-Tech” занимается следующим видом деятельности:

* Продажа электронной техники и комплектующих

Предметной областью информационной системы является деятельность магазина, которая выступает посредником между **производителем и покупателем (клиентом)**.

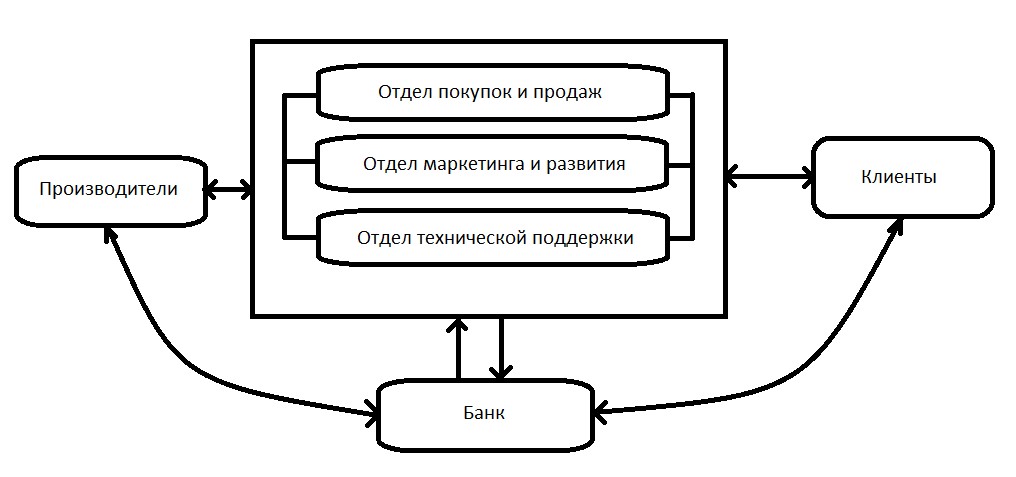


Рис. 1 – Схема работы магазина

**Цели магазина:**

* Достичь наибольшей популярности среди других магазинов в этой области;
* Обеспечить максимальный комфорт клиентам при пользовании данным ресурсом;
* Обеспечить всем производителям равные условия для конкуренции;
* Создать общую систематизированную базу электронной техники и комплектующих;

**Взаимодействие с производителем:**

**Отдел покупок и продаж** регистрирует нового производителя в **каталоге «Производители»**. Далее происходит рассмотрение предложения о сотрудничестве от производителя на продажу электронной техники (комплектующих) и принятие решении о проверке **товара**. Составляется **акт** об электронной техники (**комплектующих**), где **производитель (поставщик)** указывает название, описание, стоимость товара.

При отказе производителя (поставщика) от новой цены, оформляется акт об отказе **(заносится в журнал регистрации актов)** и его товар не принимается.

При согласии, **Отдел покупок и продаж** оформляет **накладную**, закрепляющий право производителя за его товаром. Оформляется **договор** об оказании **услуг**, где ставится подпись производителя товара и представителя Отдела покупок и продаж, указывается дата, реквизиты обеих сторон.

**Отдел покупок и продаж** заносит в **журнал регистрации актов** реквизиты производителя и передает копии акта о характеристиках электронной техники (комплектующих), договор об оказании услуг в **Отдел по маркетингу и развитию.**

**Отдел по маркетингу и развитию** занимается созданием **каталога «Товары»**, где содержится описание электронной техники, стоимость, информация о **производителе**. Дополнительно приобретается таргетированная реклама в социальных сетях с целью привлечения новых клиентов. Оформляется **акт** о выполненных работах, ставится подпись Представителя Отдела маркетинга и развития, указывается дата завершенных работ и копия документа передается в Отдел технической поддержки.

**Отдел технической поддержки** при получении акта о выполненных работах вносит данные о новом товаре в **журнал доступных товаров**, указывая дату внесения товара. Составляется **акт** о завершении работы с подписями представителей Отдела маркетинга и технической поддержки, который передается в Отдел покупок и продаж. Копия данного документа отсылается по почте производителю.

**Отдел покупок и продаж** передает реквизиты производителя и магазина в **Банк**, где заключается договор о взаимовыручке. Одна копия договора отсылается производителю, вторая передается в магазин. Происходит создание *«денежного коридора»*. Товар выставляется в магазине и становится доступен для всех верифицированных пользователей.

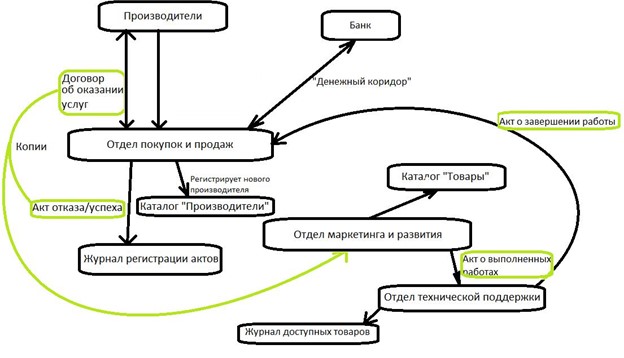


Рис 2. – Схема работы магазина (техническая часть)

**Взаимодействие с клиентом:**

Перед просмотром и выбором товара клиент обязан пройти **верификацию**.

При верификации клиент указывает о себе личную информацию - Фамилию, Имя, Отчество (необязательно), контактный телефон, дату рождения, электронную почту. Также при верификации необходимо поставить роспись «Соглашения об обработке персональных данных». После заполнения всех данных продавец магазина проверяет их корректность и достоверность.

Данные пользователя, прошедшего проверку, вносятся в журнал учета клиентов. После **верификации**, клиенту становятся доступны все возможности, а именно – покупка/заказ/предзаказ товаров, консультация в **Отделе технической поддержки**.

Если пользователь умышленно заполняет данные некорректно, продавец обязан внести клиента в «черный список» на 30 дней, с указанием даты внесения. С этого момента клиенту будет отказано в обслуживании на определенный срок. Повторное внесение в «черный список» влечет за собой пожизненный отказ в обслуживании клиента.

При покупке товара пользователь отправляет *платежное поручение* в Банк, где указывается сумма и реквизиты магазина. Клиент оплачивает товар, ему возвращается чек в электронном виде от Банка.

**Задачи Отдела покупок и продаж:**

* Рассмотрение заявок и регистрация новых производителей в каталоге «Производители»;
* Назначение новой цены товару производителя, исходя из расчетов прибыли с продаж данного продукта;
* Передача реквизитов производителей и магазина Банку.

**Задачи Отдела технической поддержки:**

* Проверка данных продукта и производителя;
* Осуществление гарантийного ремонта/возврата средств;
* Оформление запроса в Банк о возврате денежных средств клиенту;

**Задачи Отдела маркетинга и развития:**

* Оформление каталога с указанием всех необходимых данных о товаре;
* Размещение рекламы о товаре производителя;
* Продвижение магазина, реклама;

**Документы, участвующие при регистрации производителя и его товара:**

* **Журнал** регистрации актов;
* **Акт** о технических характеристиках;
* **Договор** об оказании услуг;
* Акт о выполненных работах;
* Акт о завершении работы;
* Договор о взаимовыручке;

**Документы, участвующие при регистрации клиента (пользователя):**

* Документ «Соглашение об обработке персональных данных»;

## 1.2. Выделение понятий предметной области в составление глоссария

Понятия предметной области можно выделить из Описания взаимодействия с производителем электронной техники, а также из Взаимодействие с клиентом:

1. ***Покупатель, клиент, пользователь*** – частное лицо или организация, которая покупает товар в магазине и использует для расчетов за покупку Банк.
2. ***Магазин (Отдел покупок и продаж, Отдел маркетинга и развития, Отдел технической поддержки)*** – сервис по продаже электронной техники и комплектующих, которые разработаны и поддерживаются сторонними компаниями.
3. ***Производитель, издатель, поставщик*** – имеет товар, который продает, используя услуги магазина, расчеты и вычеты производятся через Банк.
4. ***Банк*** – здесь рассматривается, как организатор расчетов между объектами. Принимает платежное поручение от пользователя для передачи средств.

Банк создает «денежный коридор» между магазином и производителем.

1. ***Продукт, товар***– здесь рассматривается, как результат деятельности производителя (электронной техники и/ или комплектующие).
2. ***Пользовательские данные***– данные, которые передает пользователь при авторизации в магазине. К ним относится Фамилия, Имя, Отчество (необязательно), контактный телефон, дата рождения, банковская карта, пароль.
3. ***Каталог «Производители»*** – картотека производителей, где у каждого производителя указывается название, продукт, закрепленный за ним, дата обращения в магазин.
4. ***Каталог «Товары»*** – картотека товаров, где у каждого производителя указывается название, технические характеристики и дата добавления товара в магазин.
5. ***Заявка*** – заявление о предоставлении прав на продажу товара.
6. *«****Денежный коридор****»* – это организованный поток денежных средств между такими объектами как магазин и производитель, поддержку которого обеспечивает Банк.
7. ***Администратор*** – лицо, наделенное рядом полномочий для поддержания порядка на страницах сайта.
8. ***Верификация*** – введение пользователем своих личных данных для пользования услугами магазина.
9. ***Запрет*** – это способ регулирования поведения и действий клиента, опирающийся на соглашения между магазином и клиентом.
10. ***Платежное поручение*** *–* запрос пользователя на перевод денежных средств от своего лица в бюджет магазина через Банк.
11. ***Акт*** *— это* документ, который подтверждает какой-то установленный факт, произошедшее событие или имеющее место действие. Составляют его несколько лиц.
12. ***Договор*** *–* это письменное соглашение, решение о будущих действиях и обязательствах, которое принимается между магазином и производителем.
13. ***Администратор базы данных*** — лицо, отвечающее за выработку требований к базе данных, её проектирование, реализацию, эффективное использование и сопровождение, включая управление учётными записями пользователей БД и защиту от несанкционированного доступа. Не менее важной функцией администратора БД является поддержка целостности базы данных.

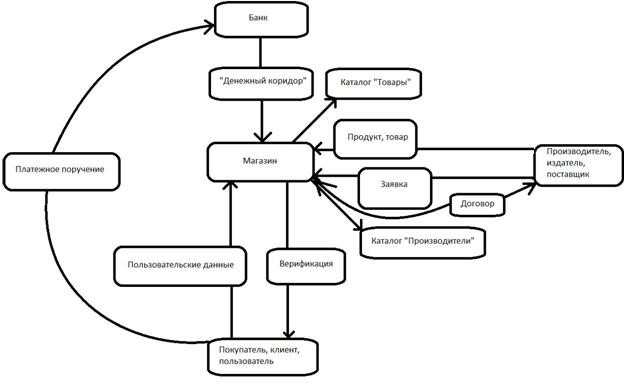


Рис. 3 - Диаграмма основных понятий предметной области и связей между ними

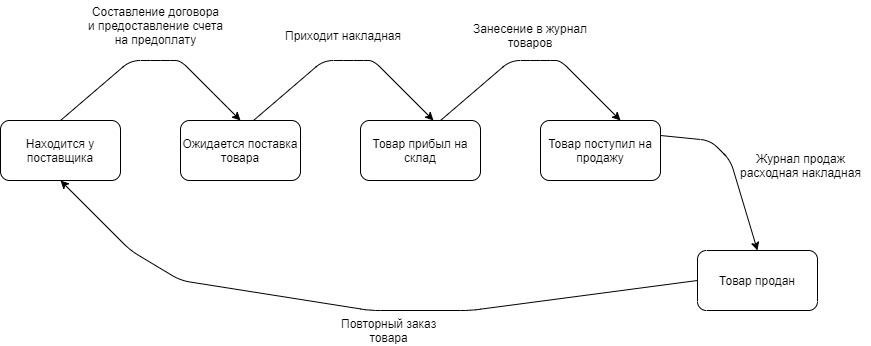


Рис. 4 - Схема движение товара

## **1.3. Предварительная требования к информационной системе**

На основе анализа предметной области можно сформулировать следующие предварительные требования к информационной системе:

**Функциональные требования**

FR010. Система должна позволять вводить, хранить, редактировать и удалять информацию о клиентах.

FR020. Система должна позволять вводить, хранить, редактировать и удалять информацию о работниках магазина электронной техники.

FR030. Система должна позволять вводить, хранить, редактировать и удалять информацию о товарах и услугах.

FR040. Система должна позволять вводить информацию о проданном товаре в журнал продаж.

FR050. Система должна предоставлять информацию о результатах работы магазина электронной техники за выбранный период времени: о поступивших товарах, проданных товарах, проведенных приемах пациентов.

FR060. Система должна позволять вводить информацию о поступившем товаре в журнал регистрации принятых товаров.

FR070. Система должна предоставлять возможность просмотра каталога товаров и услуг.

FR080. Система должна обеспечивать поиск товара.

FR090. Поиск должен производиться по всем параметрам каждого вида объектов.

FR100. Система должна предоставлять возможность фильтрации, сортировки и настройки отображения данных.

FR110. Система должна автоматически создавать согласие на обработку персональных данных.

FR120. Система должна позволять работать с производителем по предоплате, а значит создавать следующие документы: заявку на поставку, **договор поставки товара,** счет-фактура, счет на предоплату.

FR130. Система должна позволять вести журнал регистрации входящих документов и журнал наличия товара.

FR140. Система должна сохранять информацию обо всех поступлениях товаров в приходной накладной и обо всех их продажах (расходах) в **расходной накладной.**

FR150. Система должна генерировать следующие отчеты: приход, расход и выручка. При этом общей для всех отчетов должна быть функция составления отчета за какой-то конкретный период.

FR160. Система должна выполнять проверку правильности вводимых данных при регистрации и редактировании информации о товарах, клиентах и производителя.

FR170. Система должна правильно назначать новую цену на товары от производителя, учитывая налог НДС и прибыль магазина с данных продаж;

FR180. Система должна хранить данные о всех производителях.

FR190. Система должна хранить данные о всех клиентах, приобретавших товар в магазине, а также дату покупки.

FR200. Система должна учитывать все виды товаров и неограниченное количество самих товаров;

FR210. Система должна отображать все новые товары;

FR210. Система должна отображать все виды конкретного электронной техники и/или комплектующих в определенной рубрике, форуме;

**Варианты пользования**

**Список действующих лиц**

Администратор – обслуживает, проектирует ИС, добавляет новых пользователей и управляет правами каждого из пользователей, редактирует записи (добавляет, обновляет, удаляет)

Продавец –добавляет информацию о покупках клиентов, ведёт учет клиентов, а также просматривает состояние заказа или регистрирует новый.

## 1.4. Составление диаграммы прецедентов



Рис. 5 – Диаграмма прецендентов Администратора



Рис. 6 – Диаграмма прецендентов Продавца

## 1.5. Описание вариантов пользования

|  |  |  |
| --- | --- | --- |
| **Участник** | **Вариант пользования** | **Описание** |
| **Администратор** | Создание/удаление/ редактирование всех БД | Задача состоит в разработке структуры таблиц базы данных. Кроме таблиц администраторы разрабатывают и другие объекты базы данных, предназначенные, с одной стороны, для автоматизации работы с базой, а с другой стороны - для  ограничения функциональных возможностей работы с базой. |
| Добавление новых пользователей ИС | Задача состоит в добавлении лиц, отвечающих за выработку требований к базе данных, эффективное использование и сопровождение. |
| Назначение прав | Задача состоит в назначении уровня доступа пользователей ко всем функциям информационной системы. |
| **Администратор** | Добавление/удаление производителя | Задача состоит в добавлении или удалении производителей, с которыми сотрудничает магазин. |
|  | Добавление/удаление товара | Задача состоит в добавлении или удалении товаров, которые поставляет производитель для реализации в магазине. |
| Изменение некорректных данных | Задача состоит в проверке информации на достоверность, а также в её изменении. |
| **Продавец магазина** | Просмотр/добавление заказа на электронной  техники | Задача состоит в проверке наличия электронной техники в магазине, а также в оформлении заказа на необходимый товар. |
| Добавление/изменение данных о клиенте | Задача состоит в занесении данных о клиенте, а также их изменении, в случае ошибки. |
| Добавление информации о купленном товаре | Задача состоит в добавлении информации о приобретенном товаре клиентами, и ведении отчетности покупок. |

## 1.6. Составление подробного словесного описания вариантов использования

1. **Вход пользователя в ИС магазина**

|  |  |
| --- | --- |
| **Пользователь** | **Система** |
| 1) Переходит на веб-страницу магазина. | Отображает главную страницу магазина. |
| 2) Видит перед собой главную страницу магазина, в панели навигации необходимо нажать кнопку «Авторизация». | Отображает форму авторизации с полями «Логин» и «Пароль», а также предлагает «Вход для администратора». |
| 3) Вводит логин и пароль,  нажимает на кнопку  «Авторизоваться». | Проверяет совпадение сочетания логин/пароль в базе данных пользователей. В случае успеха |
|  | перенаправляет пользователя на главную страницу.  *Альтернативный вариант:*  В случае несовпадения выводит ошибку «Имя пользователя или пароль неверны», пользователю будет предложено ввести их заново. |

1. **Добавление заказа на электронные техники продавцом**

|  |  |
| --- | --- |
| **Пользователь** | **Система** |
| 1) Переходит на вкладку  «Товары» в панели навигации. | Отображает раздел «Товары» и предоставляет пользователю инструменты для поиска товара. |
| 2) Заполняет корзину необходимыми товарами по требованию клиента, нажимая напротив товаров кнопку «В корзину». | Добавляет все товары в корзину. |
| 3) Предлагает пользователю подтвердить заказ, нажимая на кнопку «Корзина», где можно удалить товар или увеличить количество какого-либо товара. | Отображает все товары, добавленные в корзину. Также предлагает пользователю вернуться к выбору товаров по кнопке «Вернуться в магазин», очистить всю корзину при помощи кнопки «Очистить корзину», или нажать кнопку «Заказать». |
| 4) Нажимает на кнопку  «Заказать», клиент отправляет заказ на товар. | Отображает форму, где необходимо ввести данные о клиенте, либо выбрать уже существующего клиента в базе данных. |

1. **Добавление данных о клиенте продавцом**

|  |  |
| --- | --- |
| **Администратор** | **Система** |
| 1) Переходит на «Панель управления» вводя URL localhost/admin. | Отображается панель управления. |
| 2) Во вкладке «Клиенты» заполняет данные о клиенте и нажимает кнопку «Добавить». | Считывает все поля, заполненные продавцом, и заносит в базу данных сведения о клиенте. |

1. **Добавление данных о товаре администратором**

|  |  |
| --- | --- |
| **Администратор** | **Система** |
| 1) Переходит на «Панель управления» вводя URL localhost/admin. | Отображается панель управления. |
| 2) Во вкладке «Управление товарами» нажимает кнопку «Добавить». | Отображает форму с полями, необходимыми для добавления нового товара. |
| 3) Заполняет все поля для товара и нажимает на кнопку «Сохранить». | Сохраняет новый товар в базе данных и отображает его на вкладке «Товары». |

1. **Добавление данных о производителе администратором**

|  |  |
| --- | --- |
| **Администратор** | **Система** |
| 1) Переходит на «Панель управления» вводя URL localhost/admin. | Отображается панель управления. |
| 2) Во вкладке «Управление производителями» нажимает кнопку «Добавить». | Отображает форму с полями, необходимыми для добавления нового производителя. |
| 3) Заполняет все поля для производителя и нажимает на кнопку «Сохранить». | Сохраняет нового производителя в базе данных и делает его доступным при добавлении нового товара. |

1. **Изменение некорректных данных администратором**

|  |  |
| --- | --- |
| **Администратор** | **Система** |
| 1) Переходит на «Панель управления» вводя URL localhost/admin. | Отображается панель управления. |
| 2) В зависимости от данных, которые необходимо изменить, переходит на нужную вкладку («Управление товарами/заказами/производителями») | Отображает форму с данными о товарах/заказах/производителях. |
| 3) Напротив некорректной записи нажимает «Изменение» | Отображает форму с полями для изменения данных о товаре/заказе/производителе. |
| 4) Изменяет некорректные данные и нажимает «Сохранить» | Удаляет прежнюю запись, и добавляет новую с указанными изменениями. |

Далее опираясь на диаграмму прецедентов и словесное описание вариантов использования можно составить диаграмму последовательности, которая предназначена для моделирования взаимодействия объектов Системы во времени, а также обмена данными между ними

## 1.7. Упрощенная диаграмма последовательности

Построим диаграммы последовательности для каждого прецедента и выделим интерфейсные, управленческие и объектные формы в каждом. Диаграмма последовательности состоит из последовательных откликов системы на различные действия клиента, взаимодействия отдельных элементов системы между собой (интерфейс с контроллером, контроллер с объектными формами) и вариантами ответа на запросы.

Между экранными формами подразумевается наличие контроллера.

1. **Регистрация покупки.**

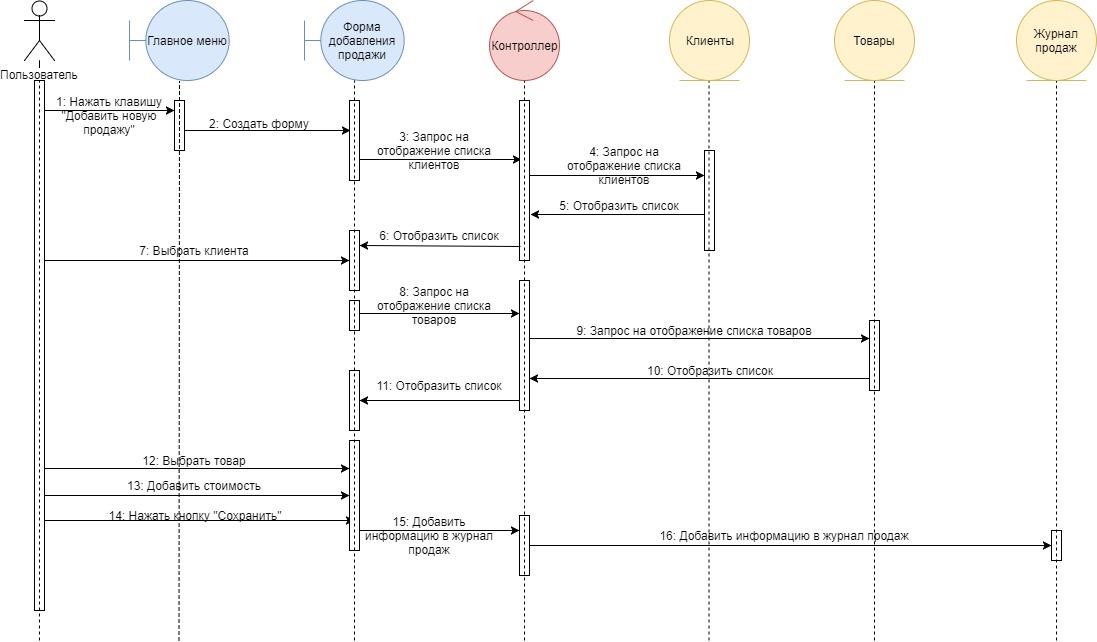


Рис. 7 – Регистрация покупки

1. **Авторизация.**

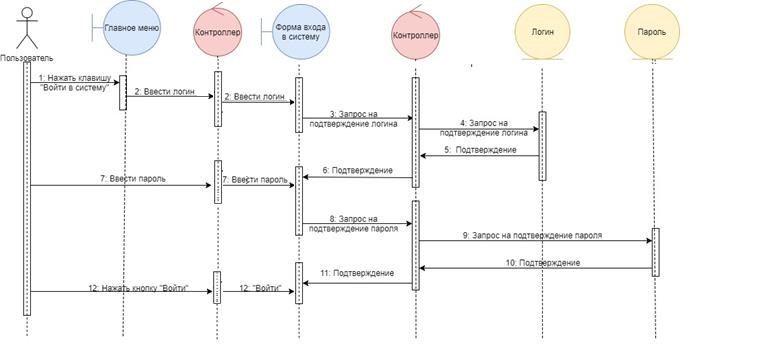


Рис. 8 - Авторизация

1. **Поиск необходимого товара.**

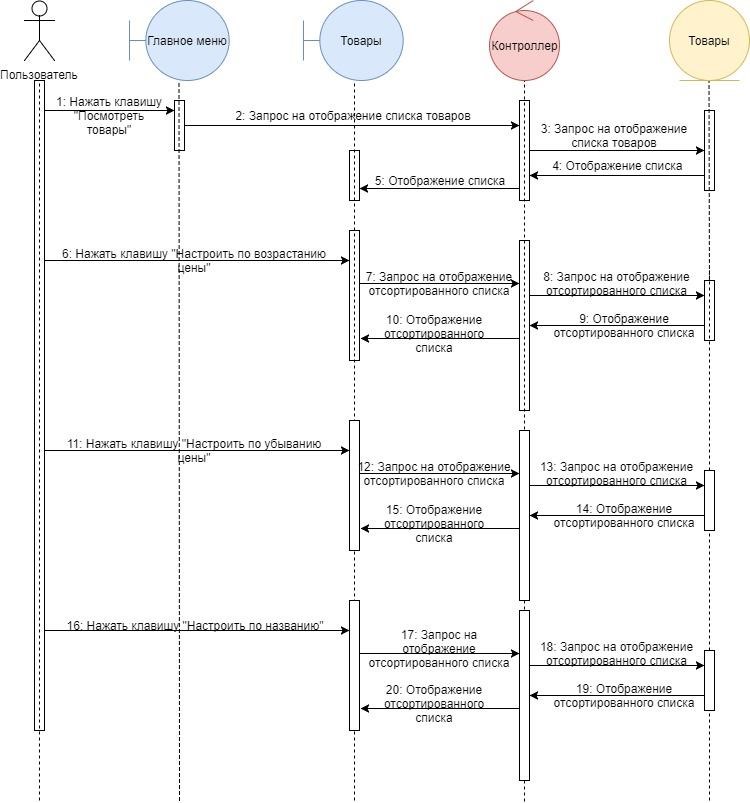


Рис. 9 - Поиск необходимого товара

1. **Добавление нового товара.**

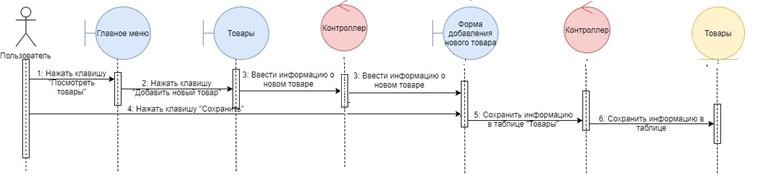


Рис. 10 - Добавление нового товара.

1. **Добавление нового клиента.**

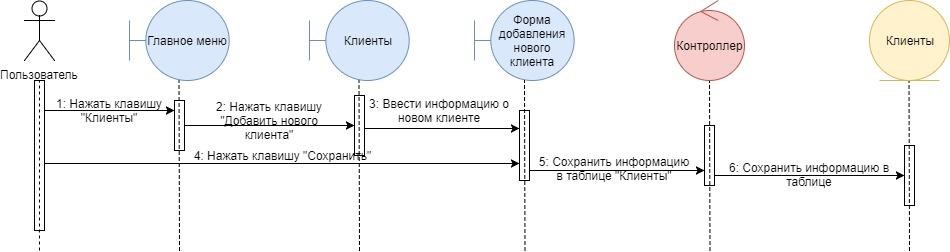


Рис. 11 - Добавление нового клиента.

1. **Изменение информации о товаре.**

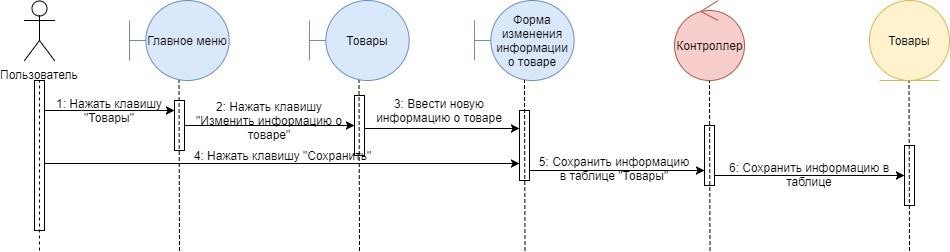


Рис. 12 - Изменение информации о товаре.

## 1.8. Составление диаграммы классов

Для представления используемых объектов и для конкретизации их параметров и выполняемых функций мы построим диаграмму классов, в которой каждый из них определенным образом взаимодействует друг с другом. При входе в систему клиент может попасть на конкретные классы, а на другие попасть только после взаимодействия с первыми. В связи с этим была спроектирована предварительная диаграмма классов, которая в ходе разработки и тестирования программного обеспечения может дополняться или изменяться.

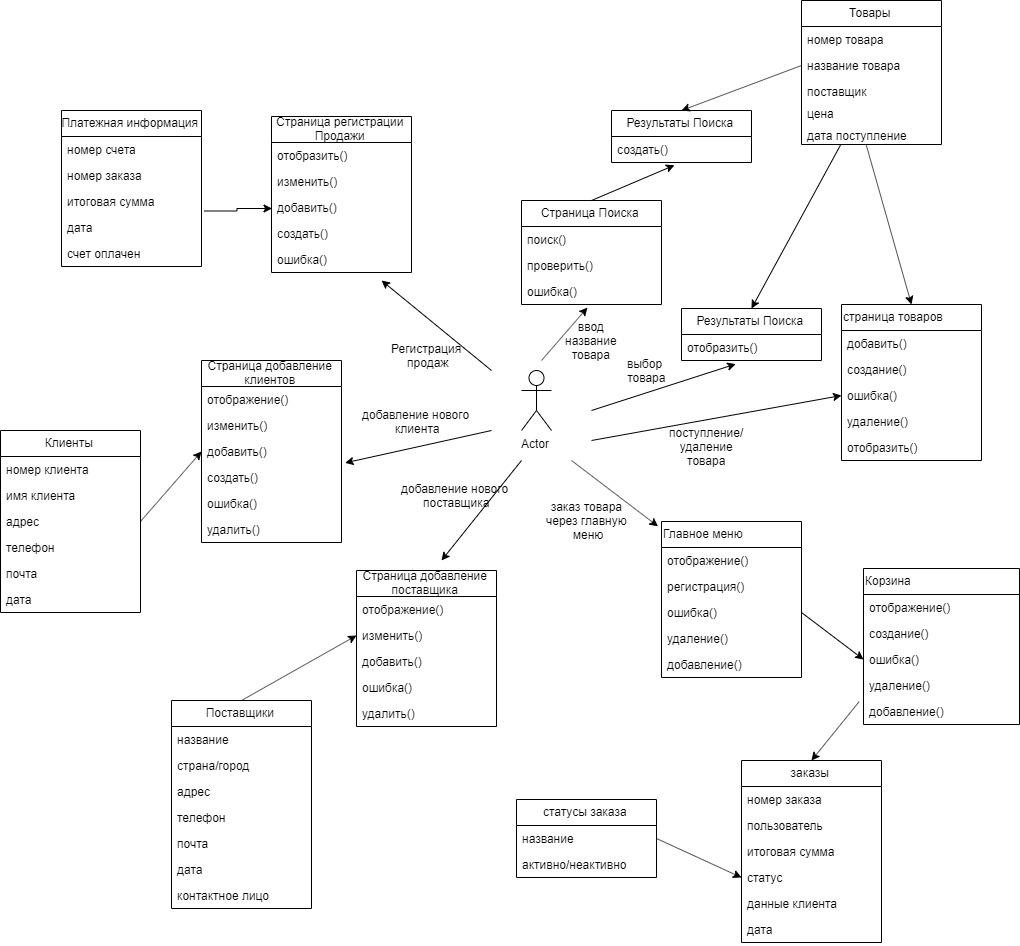


Рис. 13 - Диаграмма классов

## 1.9. Разработка схемы базы данных

Составление базы данных основано на предыдущих исследованиях и учета особенностей каждого класса. Каждая таблица приведена к третьей нормальной форме. На схеме также указаны первичные и вторичные ключи взаимодействия.

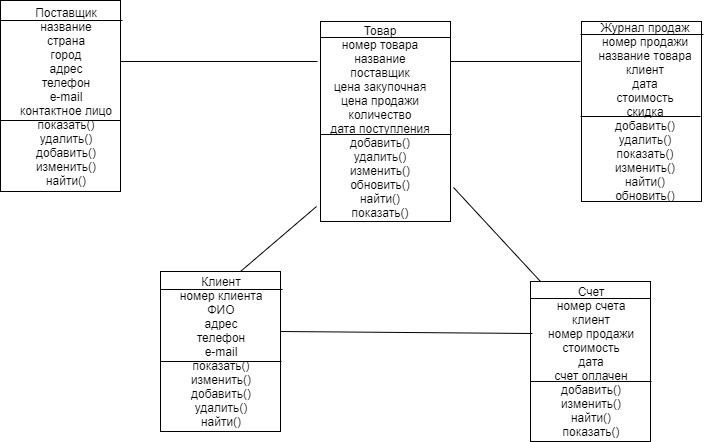


Рис. 14 - Упрощенная схема базы данных

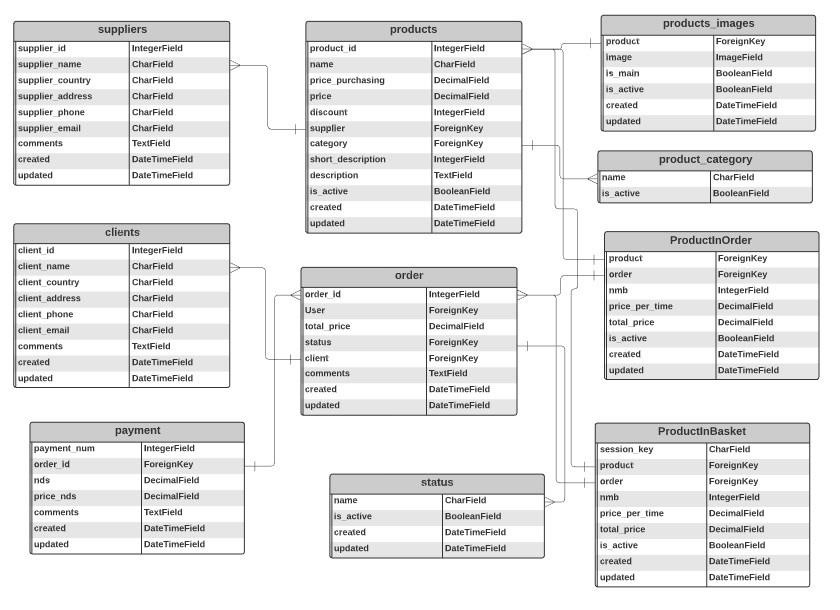


Рис. 15 - Схема базы данных (MySQL)

# II. КОНСТРУКТОРСКАЯ ЧАСТЬ

## 2.1. Техническое задание

### 2.1.1. Введение

Магазин электронной техники существует для оптимизации взаимодействия клиента и предприятия магазина и являет его онлайн интерфейсом.

Цели данной информационной системы:

* использования в процессе создания магазина электронной техники передовых информационных технологий
* обеспечения удобного централизованного механизма для контроля и анализа купли/продажи
* сокращения времени оформления товаров

### 2.1.2. Основания для разработки

Документ разработан на основании курсового проекта в данной предметной области.

### 2.1.3. Назначение разработки

Система предназначена для автоматизации купли/продажи, а также для контроля за выполнением заказов.

### 2.1.4. Требования к программе или программному изделию

Пользовательский интерфейс должен быть разработан как Web-приложение. Экранные формы выполняются в виде динамических или статических Web-страниц.

**Требования к системе:**

1. Система должна правильно назначать новую цену на товары от производителя, учитывая налог НДС и прибыль магазина с данных продаж;
2. Система должна хранить данные о всех производителях.
3. Система должна хранить данные о всех клиентах, приобретавших товар в магазине, а также дату покупки.
4. Система должна проверять правильность и корректность данных, введённых администратором при регистрации нового товара/производителя;
5. Система должна хранить отчет о доходах за определенный промежуток времени;
6. Система должна предоставить средства поиска информации;
7. Система должна учитывать все виды товаров и неограниченное количество самих товаров;
8. Система должна отображать все новые товары;
9. Система должна отображать все виды конкретной техники и/или комплектующих в определенной рубрике, форуме;

### 2.1.5. Требования к программной документации

По окончанию проектирования курсового проекта необходимо предоставить следующие документы:

* Перечень оказываемых услуг, описание вариантов использования и ответы системы на действия клиента
* Исходные коды
* Протокол тестирования

### 2.1.6. Технико-экономические показатели

Используемые программы и инструменты должны находиться в открытом доступе и давать возможность работы с базовыми функциями на бесплатной основе.

### 2.1.7. Стадии и этапы разработки

1. Разработка архитектуры программы
2. Анализ технологий разработки программного обеспечения
3. Выбор СУБД, среды и языка разработки серверной части, инструментов веб-приложения
4. Реализация программного обеспечения
5. Отладка и тестирование программы

### 2.1.8. Порядок контроля;

Проверка системы осуществляется в 3 этапа:

1. Проверка логической структуры исходного кода
2. Тестирование базовых функций приложения
3. Правка интерфейсных объектов

## 2.2. Архитектура программного обеспечения

В соответствии с техническим заданием будущее приложение должно иметь трехзвенную систему: веб-приложение (представление), базу данных (модель) и логику на серверной части (контроллер). Основная цель применения этой концепции состоит в отделении бизнес-логики (модели) от её визуализации (представления, вида). За счёт такого разделения повышается возможность повторного использования кода. Наиболее полезно применение данной концепции в тех случаях, когда пользователь должен видеть те же самые данные одновременно в различных контекстах и/или с различных точек зрения.

## 2.3. Выбор среды и языка разработки серверной части

При выборе языка будем руководствоваться следующими факторами:

* Платформ независимый
* Многопоточный
* Динамический
* Высокоуровневый

Под динамическим в данном случае понимается, что программа может выполнять обширное количество задач во время обработки информации, которая может быть использовано для проверки и разрешения доступа к объектам на время выполнения.

Всем этим критериям удовлетворяет язык Python3. Основным достоинством является то, что с помощью данного языка можно «разбить» реализацию: интерфейс, работу с файлами и элементами формы организовать с помощью встроенных средств.

При выборе среды разработки рассматриваются и другие возможные среды, которые помогут автоматизировать процесс разработки. Для разработки данной программы необходимо обеспечение средой следующих возможностей:

* Удобные инструменты для отладки и поиска ошибок, в случае их возникновения;
* Разработка более гибкой и надежной программы путем обработки различных исключительных ситуаций, возникающих в результате некорректной работы программы;
* Использование всплывающих подсказок во время написания кода программы, что обеспечивает значительное экономию времени и повышения уровня продуктивности.

Среда разработки **PyCharm** поддерживает все эти возможности, а также многие другие, такие как:

* Инструменты для запуска тестов и анализа покрытия кода, включая поддержку всех популярных фреймворков для тестирования;
* Инструменты для работы с базами данных и SQL файлами, включая удобный клиент и редактор для схемы базы данных;
* Окно классов для перехода по исходному коду по типам, а не файлам;
* Обозреватель документов для просмотра и поиска документации по продуктам на локальном компьютере или в Интернете;
* Умное автодополнение, инструменты для анализа качества кода, удобная навигация, расширенные рефакторинги и форматирование;
* Окно Свойства для настройки свойств и событий элементов управления в пользовательском интерфейсе.

Среду разработки **PyCharm** стоит рассматривать как интеллектуальную среду разработки, понимающую код. В процессе его написания программистом она занимается построением синтаксического дерева, определением особенностей размещенных ссылок, анализом возможных путей исполнения операторов и передачи данных.

## 2.4. Используемые инструменты и технологии веб-приложения

Для облегчения разработки веб-приложения нужно использовать фреймворки, которые являются набором шаблонов или заготовок, облегчающий разработку и объединение разных модулей программного проекта.

Необходимо провести анализ существующих фреймворков, с целью выявления наиболее подходящего для данного проекта.

Одними из популярных фреймворков являются PHP-фреймворки Zend Framework и Symfony или Django, написанный на Python.

Основным критерием выбора является язык разработки серверной части, который должен корректно взаимодействовать с Фреймворком.

Анализируя различные инструменты, для разработки веб-приложения был выбран универсальный фреймворк с открытым исходным кодом Django написанный на Python3.

## 2.5. Реализация

Для работы с базой данных Django использует собственную технологию программирования, в которой модель данных описывается классами Python, и по ней генерируется схема базы данных. В качестве среды разработки выбран редактор **PyCharm**.

В терминале PyCharm с помощью команды python manage.py runserver запускается сервер, выделяется адрес для localhost. В данном случае был выделен адрес 127.0.0.1:8000.

Информационная система составляется в первую очередь для администратора магазина электронной техники, поэтому в таблицу «Авторизация» Администратор записывает email и пароль администратора.

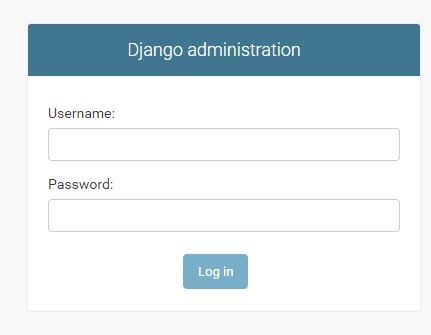


Рис. 16 - Интерфейс авторизации

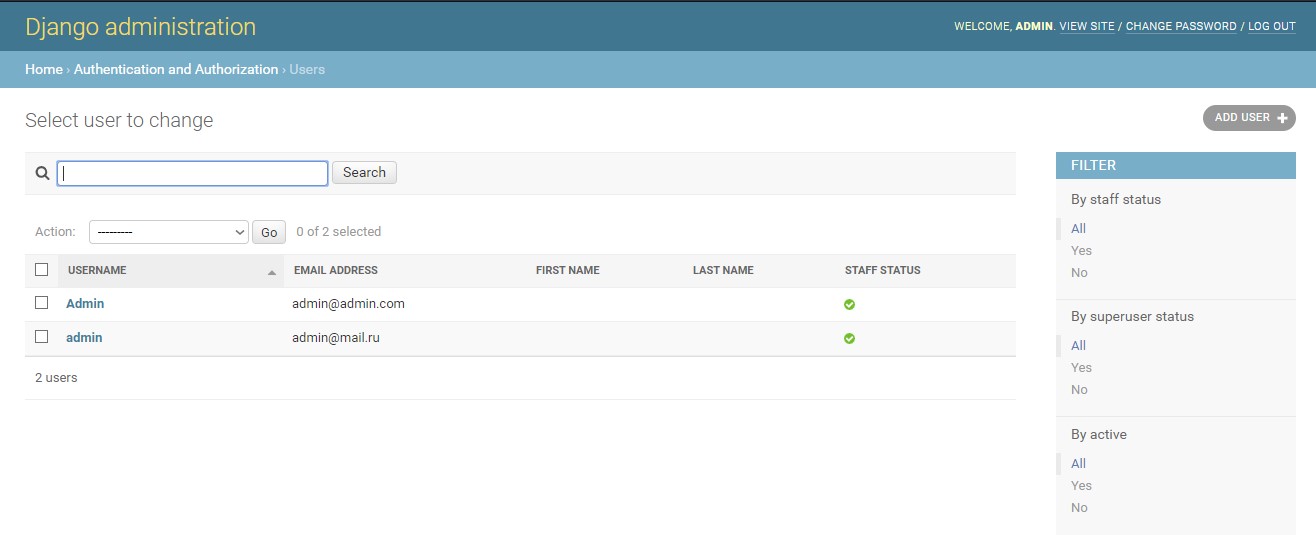


Рис. 16.1 - Таблица «Авторизация».



Рис. 17.1 - Главная страница сайта (начало).

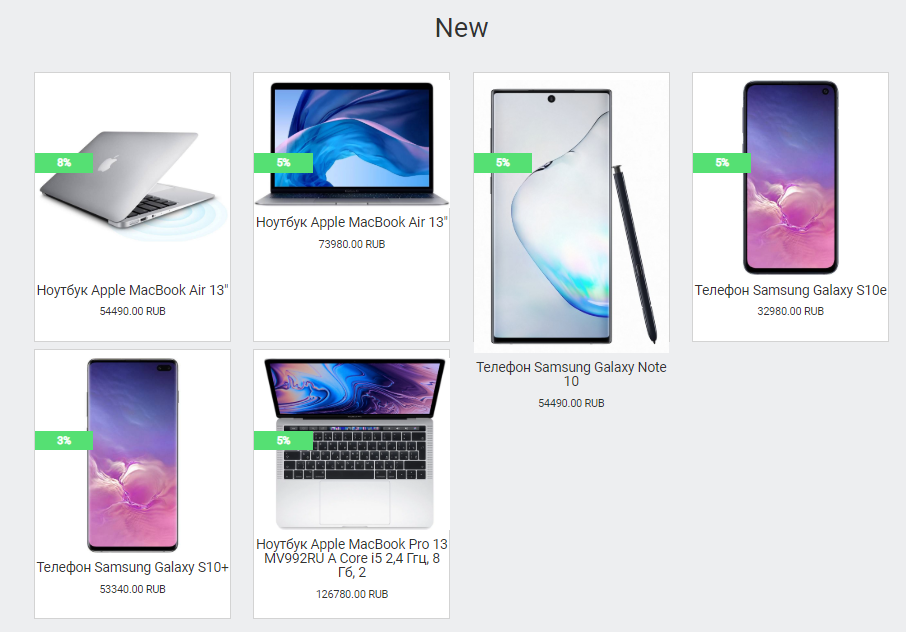


Рис. 17.2 - Главная страница сайта (продолжение), новые товары.

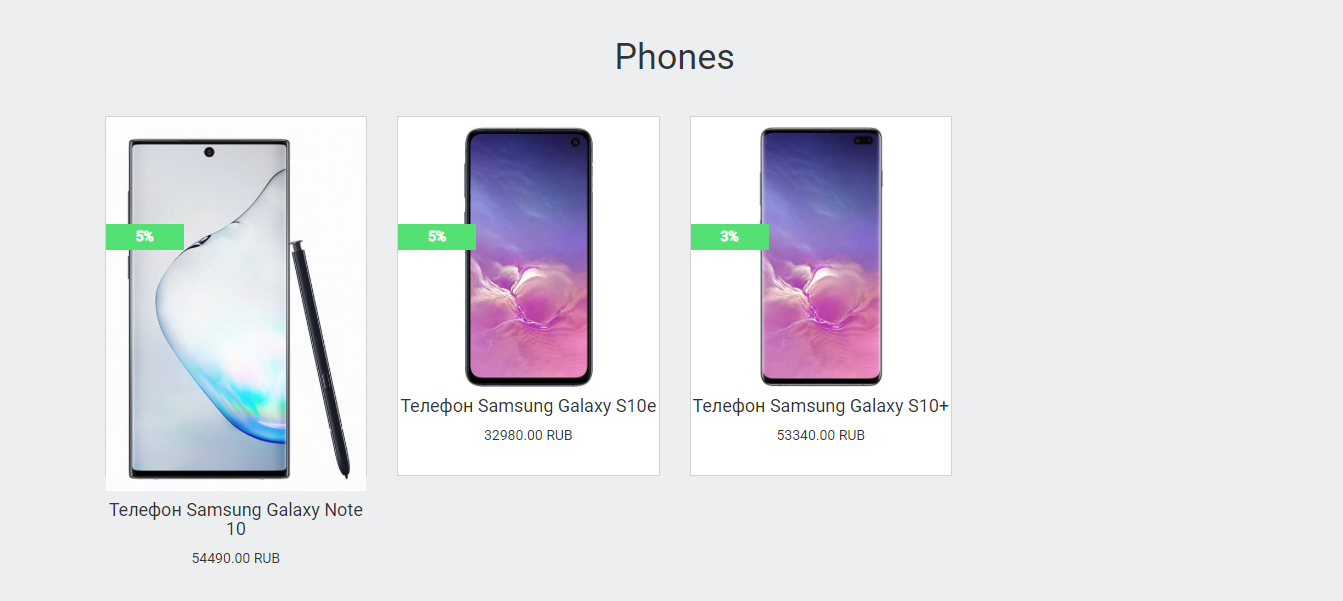


Рис. 17.3 - Главная страница сайта (продолжение) группа смартфонов.

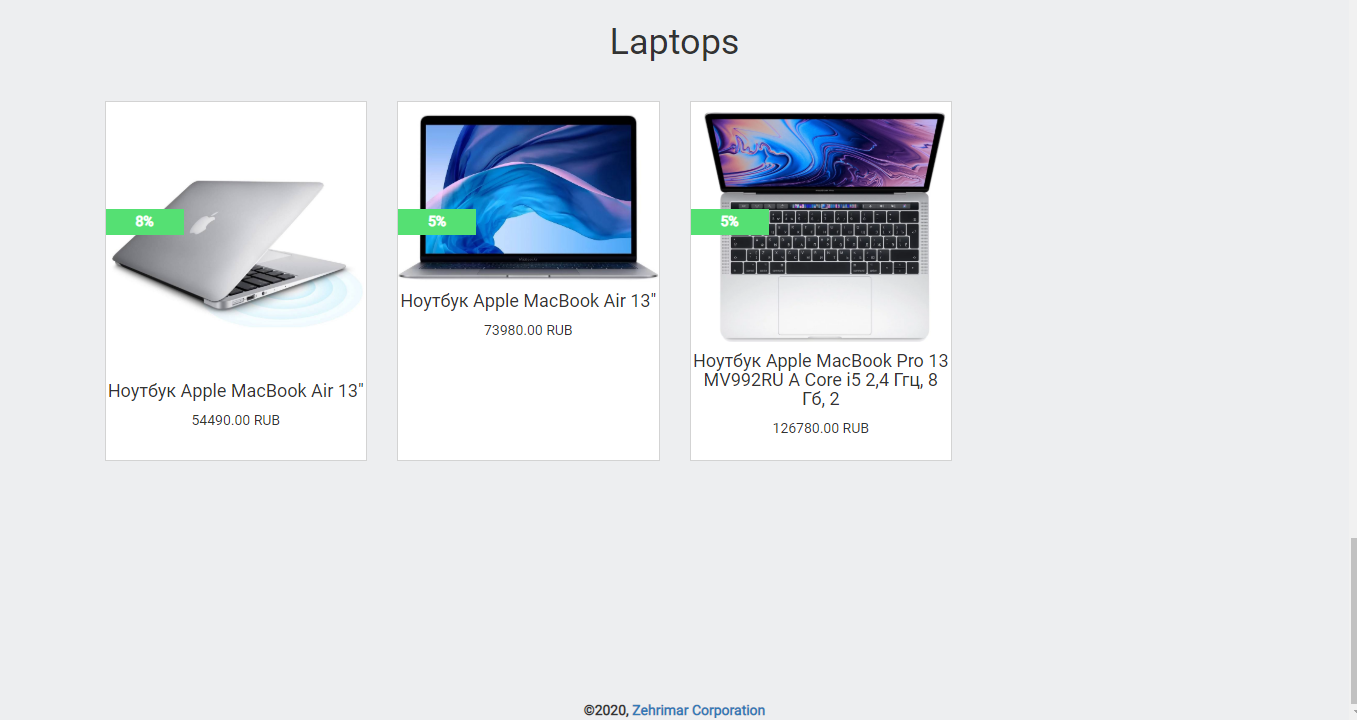


Рис. 17.4 - Главная страница сайта (конец) группа ноутбуки.

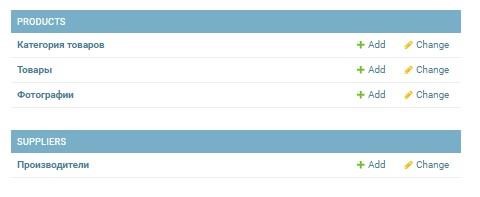
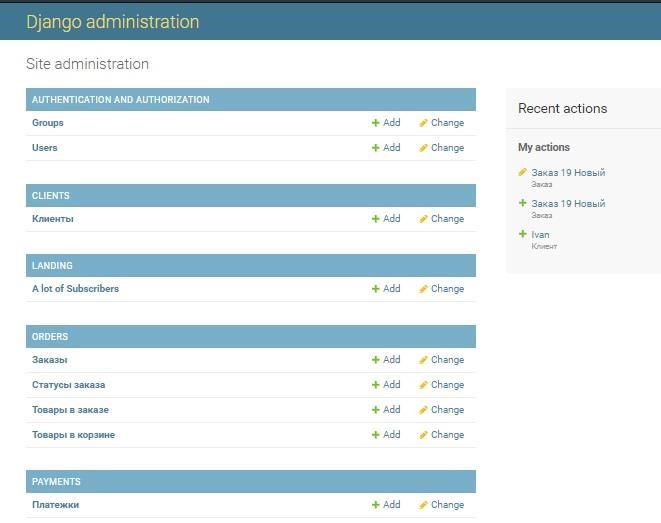


Рис. 18 - Список таблиц базы данных.

**Листинг 1.** Ниже отображена часть кода, ответственная за заказы.

**class** **Product(**models**.**Model**):**

name **=** models**.**CharField**(**max\_length**=**64**,** blank**=True,** null**=True,** default**=None)** price\_purchasing **=** models**.**DecimalField**(**max\_digits**=**10**,** decimal\_places**=**2**,** default**=**0**)**

price **=** models**.**DecimalField**(**max\_digits**=**10**,** decimal\_places**=**2**,** default**=**0**)** discount **=** models**.**IntegerField**(**default**=**0**)**

supplier **=** models**.**ForeignKey**(**

Supplier**,** blank**=True,** null**=True,** default**=None,** on\_delete**=**models**.**CASCADE**)**

category **=** models**.**ForeignKey**(**

ProductCategory**,** blank**=True,** null**=True,** default**=None,** on\_delete**=**models**.**CASCADE**)**

short\_description **=** models**.**TextField**(**blank**=True,** null**=True,** default**=None)** description **=** models**.**TextField**(**blank**=True,** null**=True,** default**=None)** is\_active **=** models**.**BooleanField**(**default**=True)**

created **=** models**.**DateTimeField**(**auto\_now\_add**=True,** auto\_now**=False)** updated **=** models**.**DateTimeField**(**auto\_now\_add**=False,** auto\_now**=True)**

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

**return** "%s, %s" **%** **(**self**.**price**,** self**.**name**)**

**class** **Meta:**

verbose\_name **=** 'Товар' verbose\_name\_plural **=** 'Товары'

Каждый пункт обрабатывается отдельной моделью, берущую данные из БД. При пустом списке элементов, пункт не отображается.

**Листинг 2.** Ниже отображена часть кода, ответственная за заказы.

**class** **ProductInOrder(**models**.**Model**):** order **=** models**.**ForeignKey**(**

Order**,** blank**=True,** null**=True,** default**=None,** on\_delete**=**models**.**CASCADE**)**

product **=** models**.**ForeignKey**(**

Product**,** blank**=True,** null**=True,** default**=None,** on\_delete**=**models**.**CASCADE**)**

nmb **=** models**.**IntegerField**(**default**=**1**)**

# price\*nmb

price\_per\_item **=** models**.**DecimalField**(**max\_digits**=**10**,** decimal\_places**=**2**,** default**=**0**)** total\_price **=** models**.**DecimalField**(**max\_digits**=**10**,** decimal\_places**=**2**,** default**=**0**)**

is\_active **=** models**.**BooleanField**(**default**=True)**

created **=** models**.**DateTimeField**(**auto\_now\_add**=True,** auto\_now**=False)** updated **=** models**.**DateTimeField**(**auto\_now\_add**=False,** auto\_now**=True)**

**def** \_\_str\_\_**(**self**):** **return** "%s" **%** self**.**product**.**name

**class** **Meta:**

verbose\_name **=** 'Товар в заказе' verbose\_name\_plural **=** 'Товары в заказе'

**def** save**(**self**,** **\***args**,** **\*\***kwargs**):** price\_per\_item **=** self**.**product**.**price self**.**price\_per\_item **=** price\_per\_item **print** **(**self**.**nmb**)**

self**.**total\_price **=** **int(**self**.**nmb**)** **\*** price\_per\_item

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

*@disable\_for\_loaddata* **def** product\_in\_order\_post\_save**(**sender**,** instance**,** created**,** **\*\***kwargs**):** order **=** instance**.**order

all\_products\_in\_order **=** ProductInOrder**.**objects**.filter(**order**=**order**,** is\_active**=True)** order\_total\_price **=** 0 **for** item **in** all\_products\_in\_order**:** order\_total\_price **+=** item**.**total\_price

instance**.**order**.**total\_price **=** order\_total\_price instance**.**order**.**save**(**force\_update**=True)**

post\_save**.**connect**(**product\_in\_order\_post\_save**,** sender**=**ProductInOrder**)**

Каждый пункт обрабатывается отдельной моделью, берущую данные из БД. При пустом списке элементов, пункт не отображается.

**Листинг 3.** Ниже отображена часть кода, ответственная за заказы.

**class** **ProductInBasket(**models**.**Model**):**

session\_key **=** models**.**CharField**(**max\_length**=**128**,** blank**=True,** null**=True,** default**=None)** order **=** models**.**ForeignKey**(**Order**,** blank**=True,** null**=True,** default**=None,** on\_delete**=**models**.**CASCADE**)**

product **=** models**.**ForeignKey**(**

Product**,** blank**=True,** null**=True,** default**=None,** on\_delete**=**models**.**CASCADE**)**

nmb **=** models**.**IntegerField**(**default**=**1**)**

# price\*nmb

price\_per\_item **=** models**.**DecimalField**(**max\_digits**=**10**,** decimal\_places**=**2**,** default**=**0**)** total\_price **=** models**.**DecimalField**(**max\_digits**=**10**,** decimal\_places**=**2**,** default**=**0**)**

is\_active **=** models**.**BooleanField**(**default**=True)**

created **=** models**.**DateTimeField**(**auto\_now\_add**=True,** auto\_now**=False)** updated **=** models**.**DateTimeField**(**auto\_now\_add**=False,** auto\_now**=True)**

**def** \_\_str\_\_**(**self**):** **return** "%s" **%** self**.**product**.**name

**class** **Meta:**

verbose\_name **=** 'Товар в корзине' verbose\_name\_plural **=** 'Товары в корзине' **def** save**(**self**,** **\***args**,** **\*\***kwargs**):** price\_per\_item **=** self**.**product**.**price self**.**price\_per\_item **=** price\_per\_item

self**.**total\_price **=** **int(**self**.**nmb**)** **\*** price\_per\_item

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

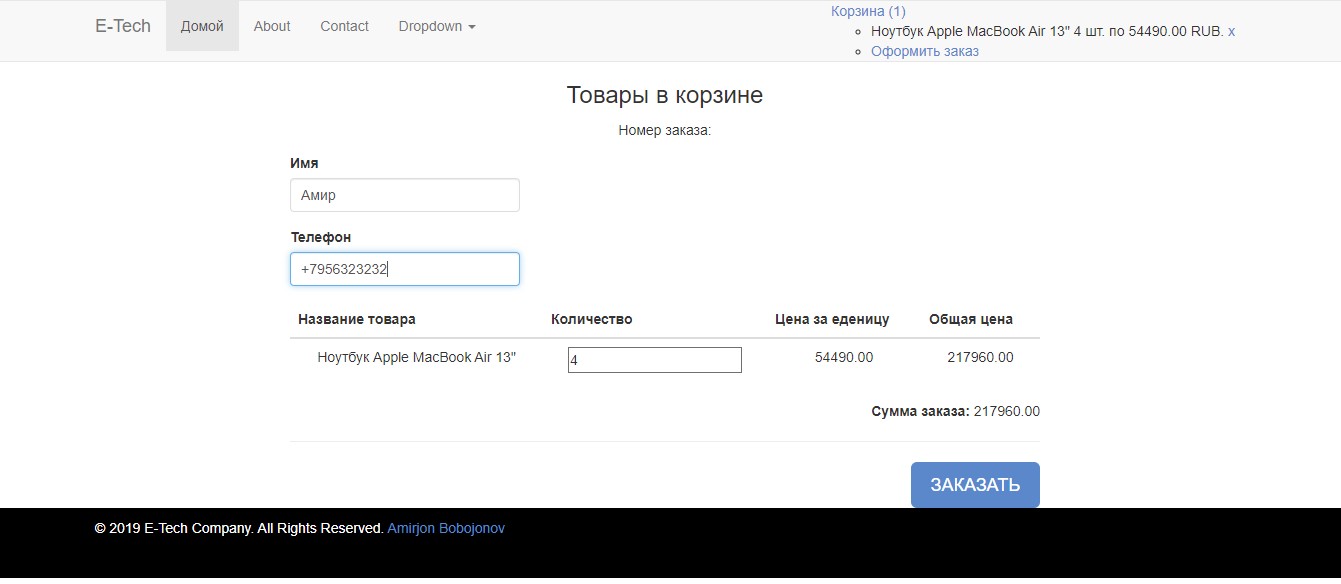


Рис. 19 - Корзина.

Каждый пункт обрабатывается отдельной моделью, берущую данные из БД. При пустом списке элементов, пункт не отображается.

**Листинг 4.** Ниже отображена часть кода, ответственная за заказы.

**class** **Status(**models**.**Model**):** name **=** models**.**CharField**(**max\_length**=**24**,** blank**=True,** null**=True,** default**=None)** is\_active **=** models**.**BooleanField**(**default**=True)**

created **=** models**.**DateTimeField**(**auto\_now\_add**=True,** auto\_now**=False)** updated **=** models**.**DateTimeField**(**auto\_now\_add**=False,** auto\_now**=True)**

**def** \_\_str\_\_**(**self**):** **return** "Статус %s" **%** self**.**name **class** **Meta:** verbose\_name **=** 'Статус заказа' verbose\_name\_plural **=** 'Статусы заказа'

### 2.5.1. **Организация** просмотра истории заказов

Нажав на вкладку «Заказы», пользователь может просмотреть историю заказов.

Каждый пункт обрабатывается отдельной моделью, берущую данные из БД. При пустом списке элементов, пункт не отображается.

**Листинг 5.** Ниже отображена часть кода, ответственная за заказы.

**class** **Order(**models**.**Model**):**

user **=** models**.**ForeignKey**(**User**,** blank**=True,** null**=True,**

default**=None,** on\_delete**=**models**.**CASCADE**)**

# total price for all products in order

total\_price **=** models**.**DecimalField**(**

max\_digits**=**10**,** decimal\_places**=**2**,** default**=**0**)**

client **=** models**.**ForeignKey**(**

Client**,** blank**=True,** null**=True,** default**=None,** on\_delete**=**models**.**CASCADE**)**

comments **=** models**.**TextField**(**blank**=True,** null**=True,** default**=None)** status **=** models**.**ForeignKey**(**Status**,** on\_delete**=**models**.**CASCADE**)** created **=** models**.**DateTimeField**(**auto\_now\_add**=True,** auto\_now**=False)** updated **=** models**.**DateTimeField**(**auto\_now\_add**=False,** auto\_now**=True)**

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

**return** "Заказ %s %s" **%** **(**self**.id,** self**.**status**.**name**)**

**class** **Meta:**

verbose\_name **=** 'Заказ' verbose\_name\_plural **=** 'Заказы' **def** save**(**self**,** **\***args**,** **\*\***kwargs**):** **super(**Order**,** self**).**save**(\***args**,** **\*\***kwargs**)**

**Листинг 6.** Ниже отображена часть кода, ответственная за заказы.

**class** **ProductCategory(**models**.**Model**):**

name **=** models**.**CharField**(**max\_length**=**64**,** blank**=True,** null**=True,** default**=None)** is\_active **=** models**.**BooleanField**(**default**=True)**

**def** \_\_str\_\_**(**self**):** **return** "%s" **%** self**.**name

**class** **Meta:**

verbose\_name **=** 'Категория товара'

verbose\_name\_plural **=** 'Категория товаров'

**Листинг 7.** Ниже отображена часть кода, ответственная за заказы.

**class** **ProductImage(**models**.**Model**):**

product **=** models**.**ForeignKey**(**

Product**,** blank**=True,** null**=True,** default**=None,** on\_delete**=**models**.**CASCADE**)**

image **=** models**.**ImageField**(**upload\_to**=**'products\_images/'**)**

is\_main **=** models**.**BooleanField**(**default**=False)** is\_active **=** models**.**BooleanField**(**default**=True)**

created **=** models**.**DateTimeField**(**auto\_now\_add**=True,** auto\_now**=False)**

updated **=** models**.**DateTimeField**(**auto\_now\_add**=False,** auto\_now**=True)**

**def** \_\_str\_\_**(**self**):** **return** "%s" **%** self**.id**

**class** **Meta:**

verbose\_name **=** 'Фотография'

verbose\_name\_plural **=** 'Фотографии'

**Листинг 8.** Ниже отображена часть кода, ответственная за заказы.

**class** **Supplier(**models**.**Model**):**

supplier\_name **=** models**.**CharField**(**max\_length**=**64**,** blank**=True,** null**=True,** default**=None)** supplier\_country **=** models**.**CharField**(**max\_length**=**64**,** blank**=True,** null**=True,** default**=None)**

supplier\_address **=** models**.**CharField**(**

max\_length**=**128**,** blank**=True,** null**=True,** default**=None)**

supplier\_phone **=** models**.**CharField**(**

max\_length**=**48**,** blank**=True,** null**=True,** default**=None)**

supplier\_email **=** models**.**EmailField**(**blank**=True,** null**=True,** default**=None)** comments **=** models**.**TextField**(**blank**=True,** null**=True,** default**=None)** created **=** models**.**DateField**(**auto\_now\_add**=True,** auto\_now**=False)** updated **=** models**.**DateField**(**auto\_now\_add**=False,** auto\_now**=True)**

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

**return** "%s" **%** self**.**supplier\_name

**class** **Meta:**

verbose\_name **=** 'Производитель' verbose\_name\_plural **=** 'Производители'

**Листинг 9.** Ниже отображена часть кода, ответственная за заказы.

**class** **Payment(**models**.**Model**):**

payment\_name **=** models**.**CharField**(**max\_length**=**64**,** blank**=True,** null**=True,** default**=None)** order\_name **=** models**.**ForeignKey**(**Order**,** blank**=True,** null**=True,** default**=None,** on\_delete**=**models**.**CASCADE**)**

nds **=** models**.**DecimalField**(**max\_digits**=**10**,** decimal\_places**=**2**,** default**=**0**)** price\_nds **=** models**.**DecimalField**(**max\_digits**=**10**,** decimal\_places**=**2**,** default**=**0**)** comments **=** models**.**TextField**(**blank**=True,** null**=True,** default**=None)** created **=** models**.**DateField**(**auto\_now\_add**=True,** auto\_now**=False)** updated **=** models**.**DateField**(**auto\_now\_add**=False,** auto\_now**=True)**

**def** \_\_str\_\_**(**self**):** **return** "%s" **%** self**.**payment\_name

**class** **Meta:**

verbose\_name **=** 'Платежка'

verbose\_name\_plural **=** 'Платежки'

Содержимое каждой таблицы можно просматривать, добавлять новые элементы, удалять элементы, искать нужную информацию.

# ЗАКЛЮЧЕНИЕ

В рамках курсового проекта была реализована система интернет-магазина. С помощью разработанной программы можно существо сократить время на оформление услуг магазина электронной техники, уменьшить нагрузку на персонал и частично избавиться от бумажных документов.

Были успешно реализованы следующие задачи:

* Автоматизация оформления заказов
* Сохранение информации о клиентах – добавление и удаление
* Минимизировать присутствие рабочего персонала для контроля товаров
* Автоматизирована регистрация входящих документов и наличия товара с помощью журнала событий

Также при тестировании программы выявлены основные недочеты, связанные с вводом неправильных данных, проведен анализ основных проблем, которые могут возникнуть в процессе разработки и поддержки программного обеспечения.

# ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

1. Гаврилова Ю.М. Лекция №1 «[06-04-2020-Лекция 01. Основные понятия и определения.pdf](ftp://eufs.bmstu.ru/4d25b186-bd19-11e6-80c8-005056960017/06-04-2020-%D0%9B%D0%B5%D0%BA%D1%86%D0%B8%D1%8F%2001.%20%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D1%8B%D0%B5%20%D0%BF%D0%BE%D0%BD%D1%8F%D1%82%D0%B8%D1%8F%20%D0%B8%20%D0%BE%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F.pdf)» (дата обращения 23.04.2020)
2. Гаврилова Ю.М. Лекция №2 «[06-04-2020-Лекция 02. Реляционная модель данных.pdf](ftp://eufs.bmstu.ru/4d25b186-bd19-11e6-80c8-005056960017/06-04-2020-%D0%9B%D0%B5%D0%BA%D1%86%D0%B8%D1%8F%2002.%20%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F%20%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85.pdf)» (дата обращения 24.04.2020)
3. Гаврилова Ю.М. Лекция №3 «[06-04-2020-Лекция 03. Семантическое моделирование данных.pdf](ftp://eufs.bmstu.ru/4d25b186-bd19-11e6-80c8-005056960017/06-04-2020-%D0%9B%D0%B5%D0%BA%D1%86%D0%B8%D1%8F%20%2003.%20%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5%20%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85.pdf)» (дата обращения 25.04.2020)
4. Гаврилова Ю.М. Лекция №4 «[06-04-2020-Лекция 04. Теория проектирования реляционных баз данных.pdf](ftp://eufs.bmstu.ru/4d25b186-bd19-11e6-80c8-005056960017/06-04-2020-%D0%9B%D0%B5%D0%BA%D1%86%D0%B8%D1%8F%20%2004.%20%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F%20%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%20%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B1%D0%B0%D0%B7%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85.pdf)» (дата обращения 26.04.2020)
5. Гаврилова Ю.М. Лекция №5 «[06-04-2020-Лекция 05. Введение в SQL.pdf](ftp://eufs.bmstu.ru/4d25b186-bd19-11e6-80c8-005056960017/06-04-2020-%D0%9B%D0%B5%D0%BA%D1%86%D0%B8%D1%8F%20%2005.%20%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B2%20SQL.pdf)» (дата обращения 27.04.2020)
6. Django Software Foundation «<https://www.djangoproject.com/>» (дата обращения 27.03.2020)
7. Python «<https://www.python.org/>» (дата обращения 05.05.2020)
8. W3Schools Online Web Tutorials «<http://w3schools.com/>» (дата обращения 03.05.2020)
9. Animate.css «<https://animate.style/>» (дата обращения 02.05.2020)
10. WebReference «<https://webref.ru/layout/html5-css3/list/use>» (дата обращения 03.05.2020)
11. Online diagram «<https://www.draw.io/>» (дата обращения 02.05.2020)
12. Python Dersleri. Python teknolojileri hakkında dökümanlar. «<https://www.pythondersleri.com/p/django-egitim-serisi.html>» (дата обращения 02.05.2020)
13. Sqlite ile Veritabanı Programlama «<https://python-istihza.yazbel.com/standart_moduller/sqlite.html>» (дата обращения 04.05.2020)