Министерство образования Новосибирской области ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

РАЗРАБОТКА СИСТЕМЫ ОТСЛЕЖИВАНИЯ ПОСЕЩЕНИЯ АУДИТОРИЙ СТУДЕНТАМИ

Пояснительная записка дипломного проекта

НАТКиГ.100700.010.000ПЗ

Выполнил:

студент группы ПР-20.102К

Лазарев Д.А.

2024

**Содержание**

[Введение 3](#_Toc165686329)

[1 Исследовательский раздел 5](#_Toc165686330)

[1.1 Описание предметной области 5](#_Toc165686331)

[1.2 Определение функциональных задач пользователей 8](#_Toc165686332)

[1.3 Анализ аналогов и прототипов 16](#_Toc165686333)

[1.4 Проектирование информационной системы 19](#_Toc165686334)

[1.5 Анализ программных ресурсов, необходимых в работе 20](#_Toc165686335)

[2 Технологический раздел 24](#_Toc165686336)

[2.1 Структура базы данных 24](#_Toc165686337)

[2.2 Разработка форм выходных документов 29](#_Toc165686338)

[2.3 Разработка структуры программного продукта 30](#_Toc165686339)

[2.4 Описание процедур и функций 37](#_Toc165686340)

[3 Инструкции пользователям системы 42](#_Toc165686341)

## Введение

В современном образовании эффективное управление учебным процессом и контроль посещаемости студентов играют ключевую роль в обеспечении качественного образования и достижении учебных целей. Однако, традиционные методы отслеживания посещаемости часто оказываются неэффективными и трудоемкими как для преподавателей, так и для администрации учебных заведений.

В этом контексте разработка приложения для отслеживания посещаемости студентов становится актуальной и востребованной задачей. Переход к цифровизации и использование современных информационных технологий позволяют создать инновационный инструмент, который сможет упростить и автоматизировать процесс учета посещаемости, обеспечивая при этом надежность, точность и удобство использования.

Цель данной курсовой работы состоит в разработке и реализации такого приложения, которое позволит эффективно отслеживать посещаемость студентов в учебных заведениях. В рамках работы будут проанализированы существующие подходы к отслеживанию посещаемости, выявлены основные требования к приложению, разработана его структура и интерфейс, а также проведена оценка эффективности и возможных преимуществ от внедрения данного решения в учебный процесс.

Результаты этой работы могут быть полезны как для преподавателей и администрации учебных заведений, так и для самих студентов, способствуя оптимизации учебного процесса, повышению его качества и обеспечению более эффективного контроля за посещаемостью.

Информационная система (далее – ИС) состоит из трех частей:

* база данных;
* серверная часть;
* десктопное приложение.

Десктопное приложение планируется использовать как основной инструмент для управления информационной системой, оно позволит полностью отображать информацию о перемещении студентов внутри стен колледжа, такую как: точное время и дату входа и выхода в аудиторию, сам номер аудитории. Также преподавателю будут доступны инструменты для удобного поиска студента и генерации отчетности, а также интуитивно понятный интерфейс для использования всего функционала приложения.

Десктопное приложение также выступает в роли посредника между базой данных и пользователем, предоставляющим функционал для взаимодействия клиента с ИС. Десктопное приложение должно обрабатывать все входные и выходные данные, которые поступают в системы путем QR кода. Студенты не могут как то пользоваться информационной системой и использовать её функционал, а лишь могут отправлять запросы косвенным способом, а именно с помощью QR кодов, заранее подготовленным преподавателям, воспользовавшимися инструментов внутри десктопного приложения.

Актуальность разрабатываемого программного продукта заключается в том, что он предназначен для замены используемых бумажных журналов, упрощения создания отчетной документации.

Можно выделить следующие задачи дипломного проекта:

* исследование предметной области;
* выбор средств разработки;
* разработка дизайн-проекта;
* разработка программной части;
* разработка базы данных;
* документирование.

## 1 Исследовательский раздел

## Описание предметной области

Предметная область отслеживания посещаемости студентов в учебных заведениях является одной из ключевых сфер управления учебным процессом. Её значимость обусловлена не только тем, что регулярное посещение занятий способствует более эффективному усвоению учебного материала и повышению успеваемости, но и тем, что это позволяет преподавателям и администрации учебных заведений оценить уровень активности студентов, выявить проблемные ситуации и своевременно предоставить необходимую поддержку.

Традиционные методы учета посещаемости, такие как ручное ведение журналов преподавателями, часто оказываются недостаточно эффективными и подвержены ошибкам. Этот процесс также требует значительных временных затрат на обработку данных, что может отвлекать преподавателей от основной образовательной деятельности.

Однако в настоящее время с развитием информационных технологий и цифровизации учебного процесса появляются новые возможности для автоматизации и улучшения процесса отслеживания посещаемости студентов. Создание приложения, позволяющего студентам и преподавателям эффективно отслеживать посещаемость занятий, становится все более актуальным.

Заинтересованными сторонами в данной предметной области являются преподаватели, администрация учебных заведений, студенты и их родители. Для них важно иметь доступ к достоверной информации о посещении занятий и возможность эффективно управлять этим процессом. Также необходимо учитывать вопросы приватности данных, безопасности и удобства использования приложения.

Обоснование актуальности проекта по созданию приложения для отслеживания посещаемости студентов в учебных заведениях основывается на нескольких ключевых аспектах.

Регулярное посещение занятий является ключевым элементом успешного обучения. Отслеживание посещаемости помогает образовательным учреждениям и преподавателям оценить уровень учебной активности студентов, выявить проблемные ситуации и своевременно оказать необходимую поддержку.

Приложение для отслеживания посещаемости может значительно упростить и автоматизировать процесс регистрации студентов на занятиях. Это позволит преподавателям и администрации учебных заведений сосредоточиться на более важных аспектах образовательного процесса.

В условиях быстрого развития информационных технологий и мобильных приложений использование современных технологий становится неотъемлемой частью образовательного процесса. Приложения для отслеживания посещаемости представляют собой пример инновационного подхода к управлению образованием.

Приложение для отслеживания посещаемости также может способствовать улучшению коммуникации между преподавателями и студентами, предоставляя им информацию о текущем состоянии посещаемости и связанных с этим вопросах.

Таким образом, создание приложения для отслеживания посещаемости студентов является актуальным и востребованным проектом, который может значительно облегчить и оптимизировать управление учебным процессом, повысить его эффективность и поддержать активное участие студентов в образовательном процессе.

## Определение функциональных задач пользователей

Серверная часть ИС будет находиться на удаленом сервере частной компании, с которой учебное заведение заключит договор, все же фотоизображения будут храниться на каждом компьютере индивидуально. Сотрудник может взаимодействовать с серверной частью в любое время.

Серверная часть обладает следующим набором функций:

* просмотр записей посещений, включая поиск по имени студента и фильтрацию по дате посещений и по группе студента;
* добавление студента;
* формирование отчета за определенный период времени индивидуально для каждого студента в формате Excel ;
* отображение подробной информации о студенте;

Разработка мобильного приложения для ИС неподразумевается, т.к. большого функционала в ней не планируется, вместо этого студенту понадобится любое приложение или встроенное приложение для сканирования QR кода .

Следовательно, мобильная часть ИС обладает следующим набором функций:

* Сканирования QR кода и передачи набора данных о студенте.

На рисунке 1 представлена диаграмма прецедентов для ИС.

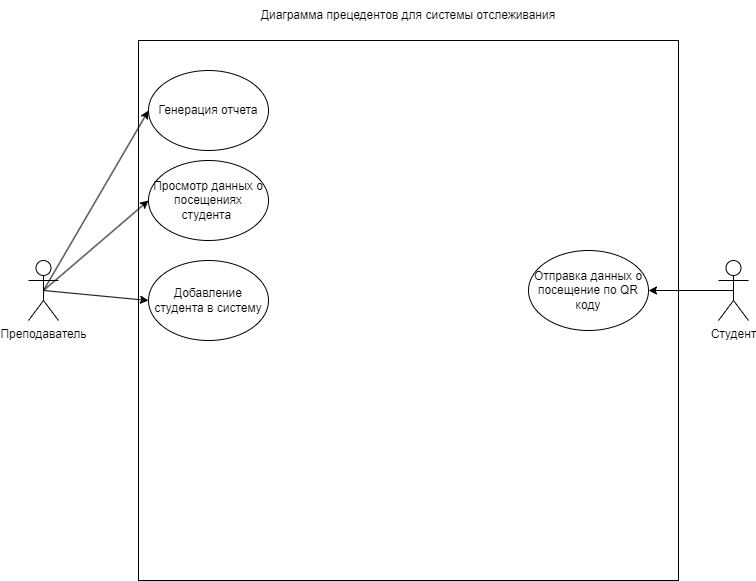


Рисунок 1 – диаграмма прецедентов ИС

В системе участвуют два актера. Преподаватель, взаимодействующий с серверной частью ИС и студент, который взаимодействует с веб частью.

Спецификации прецедентов указаны в таблицах 1–10, альтернативные потоки обозначены A*n*, где *n* – номер потока.

Таблица 1 – Спецификация для прецедента «Генерация отчета»

|  |  |
| --- | --- |
| **Раздел** | Описание |
| **Наименование прецедента** | Генерация отчета |
| **Краткое описание** | Позволяет преподавателю сгенерировать отчет о студенте |
| **Действующие лица** | Преподаватель |
| **Предусловия** | Система запущена, студент добавлен в систему |
| **Основной поток** | 1 Система открывает окно с полями для выбора дат  2 Сотрудник заполняет необходимые поля и нажимает кнопку «Отчет»  А1. Обязательные поля не заполнены |

Окончание таблицы 1

|  |  |
| --- | --- |
| **Альтернативный поток** | А1. Обязательные поля не заполнены   1. Система информирует пользователя о необходимости заполнить определенные поля 2. Поток возвращается на 2 этап основного потока |
| **Постусловие** | Результатом будет генерация отчета о студенте |

Таблица 2 – Спецификация для прецедента «Просмотр данных о посещениях студента»

|  |  |
| --- | --- |
| **Раздел** | Описание |
| **Наименование прецедента** | Просмотр данных о посещениях студента |
| **Краткое описание** | Позволяет преподавателю просматривать данные о посещениях студента |
| **Действующие лица** | Преподаватель |
| **Предусловия** | У системы есть доступ к интернету |
| **Основной поток** | 1 Система запускается |
| **Альтернативный поток** |  |
| **Постусловие** | Результатом просмотр информации о студенте. |

Таблица 3 – Спецификация для прецедента «Добавление студента в систему»

|  |  |
| --- | --- |
| **Раздел** | Описание |
| **Наименование прецедента** | Добавление студента в систему |
| **Краткое описание** | Позволяет преподавателю добавить нового студента |
| **Действующие лица** | Преподаватель |
| **Предусловия** | Система запущена |
| **Основной поток** | 1 Система открывает окно с полями для ввода данных   1. Сотрудник заполняет необходимые поля и нажимает кнопку «Добавить»   А1. Обязательные поля не заполнены |
| **Альтернативный поток** | А1. Обязательные поля не заполнены   1. Система информирует пользователя о необходимости заполнить определенные поля 2. Поток возвращается на 2 этап основного потока |
| **Постусловие** | Результатом будет добавление записи с данными студента |

Таблица 4 – Спецификация для прецедента «Отправка данных о посещении по QR коду»

|  |  |
| --- | --- |
| **Раздел** | Описание |
| **Наименование прецедента** | Отправка данных о посещении по QR коду |
| **Краткое описание** | Позволяет студенту данные о посещении по QR коду |
| **Действующие лица** | Студент |
| **Предусловия** | Система запущена, у студента есть доступ в интернет |
| **Основной поток** | 1 Студент сканирует QR код и переходит по ссылке |
| **Альтернативный поток** |  |
| **Постусловие** | Результатом будет отправка данных в систему |

## 1.3 Анализ аналогов и прототипов

«МОЙ КЛАСС» — коммерческий продукт представленный в виде CRM системы, предназначенный для автоматизации учебного процесса. Сервис реализует функции обработки данных о посещениях, управления учебного процесса, осуществления процесса коммуникации между учниками и преподавателями.

«МОЙ КЛАСС» предоставляет большое количество возможностей управления учебным процессом, таких как:

* доступ к материалам занятий;
* оповещение и рассылка;
* отчеты и аналитика;
* интеграция с мессенджерами.

«МОЙ КЛАСС» предоставляет удобный интерфейс обслуживания для любого компьютера или смартфона. Интерфейс контроля посещаемости системы представлен на рисунке 2

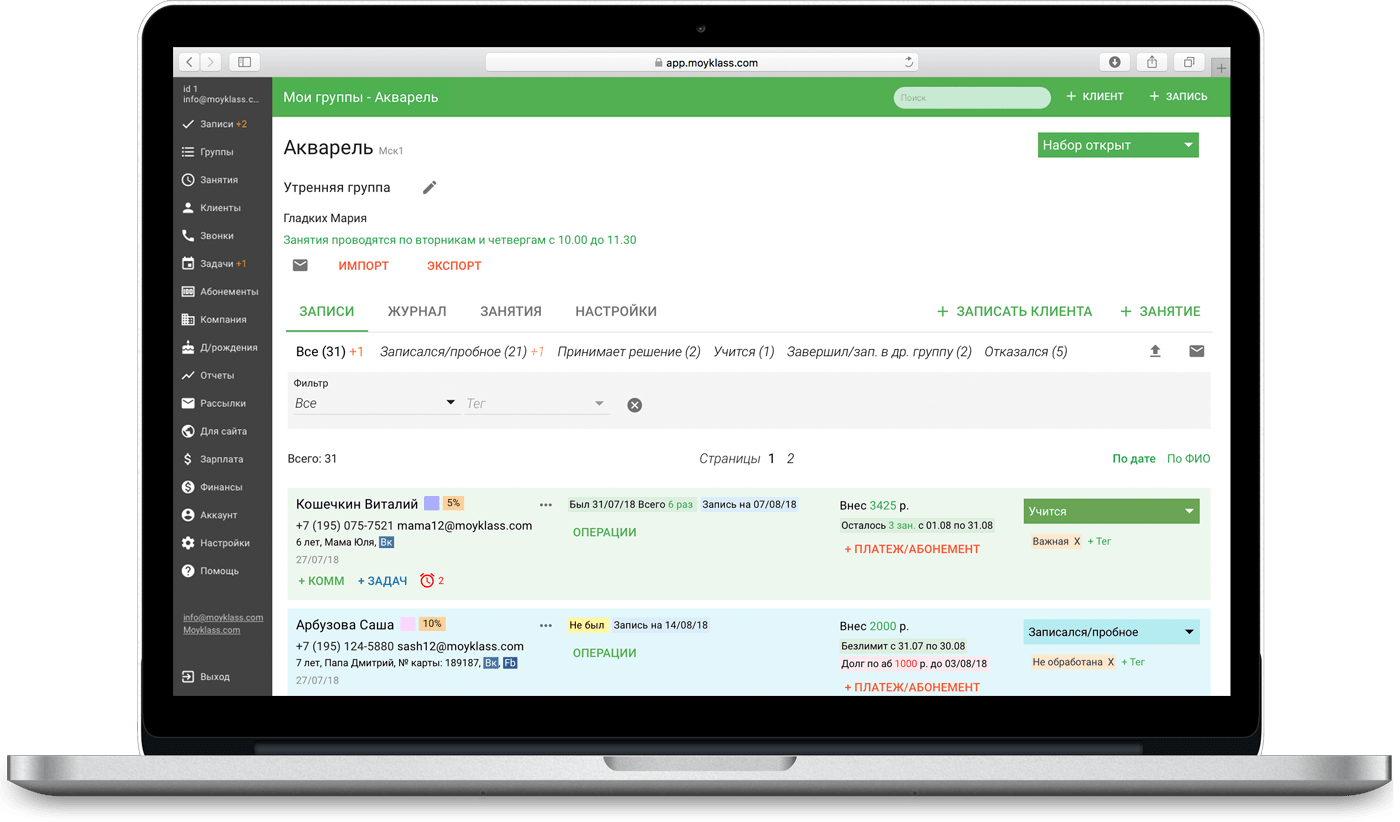


Рисунок 2 – интерфейс контроля посещаемости системы «МОЙ КЛАСС»

«Дневник.ру» – это веб приложение, включающее все функции электронного дневника. Сервис содержит основные функции: просмотр расписания, оценок, информации об учащемся. В программе можно установить коммуникацию между преподавателем и учащемся путем встроенного чата. Также в системе предусмотрен собственный файлообменник.

Основные функции, предоставляемые системой:

* просмотр посещаемости;
* широкий функционал как учащегося, так и для преподавателя;
* оценивание и посещаемости;
* обмен данными большинства форматов;
* печать отчетности об учебном процессе.

На рисунке 3 представлен интерфейс преподавателя в сервисе «Дневник.ру»



Рисунок 3 – интерфейс преподавателя

В таблице 12 представлено сравнение двух представленных выше систем по контролю учебного процесса

Таблица 12 – сравнение прототипов

|  |  |  |
| --- | --- | --- |
| Критерии\ПО | МОЙ КЛАСС | Дневник.ру |
| Функциональность | Складской учет; Розничные продажи, Хранение информации о клиентах; Обработка заказов; Контроль движения продукции; Управление финансами. | Складской учет; Кассовая программа; Информация о товаре (Карточка товара); Дисконтная система; Печать этикеток и ценников; Отчеты по категориям и группам. |
| Удобство | Доступно на смартфоне, планшете и компьютере. Удобный интерфейс. Интеграции с популярными сервисами для бизнеса. | Товароучетная система заполняется при помощи сканирования штрих-кода продукции со смартфона. База данных содержит 13,000,000 штрихкодов и товаров. Импорт товаров из Excel |

Окончание таблицы 12

|  |  |  |
| --- | --- | --- |
| Безопасность | Круглосуточная поддержка; Резервное копирование данных; Шифрование данных; Система разграничения доступа. | Передаваемые данные шифруются и передаются по защищенным каналам; Использование TLS соединения, использующее асимметричную криптографию для аутентификации, симметричное шифрование для конфиденциальности и коды аутентичности сообщений для сохранения целостности сообщений. |
| Стабильность | Хостинг в России; Ежедневное обслуживание и проверка серверов | Надежные дата-центры с гарантированным каналом связи и бесперебойным питанием |
| Доступность | Предоставляется бесплатный тестовый период на 14 дней, после предлагается подписка с ежемесячной платой, зависимой от тарифа. | Предоставляется бесплатный период 14 дней, после предлагается подписка с подарками при оплате сразу за 3/6/12 месяцев. Предоставляются тарифы с разными возможностями и стоимостью |

Все данные о прототипах, представленные в таблице, взяты с официальных сайтов о данной продукции.

## 1.4 Проектирование информационной системы

На рисунке 4 представлена диаграмма с общим функционалом визуального интерфейса серверной части ИС.

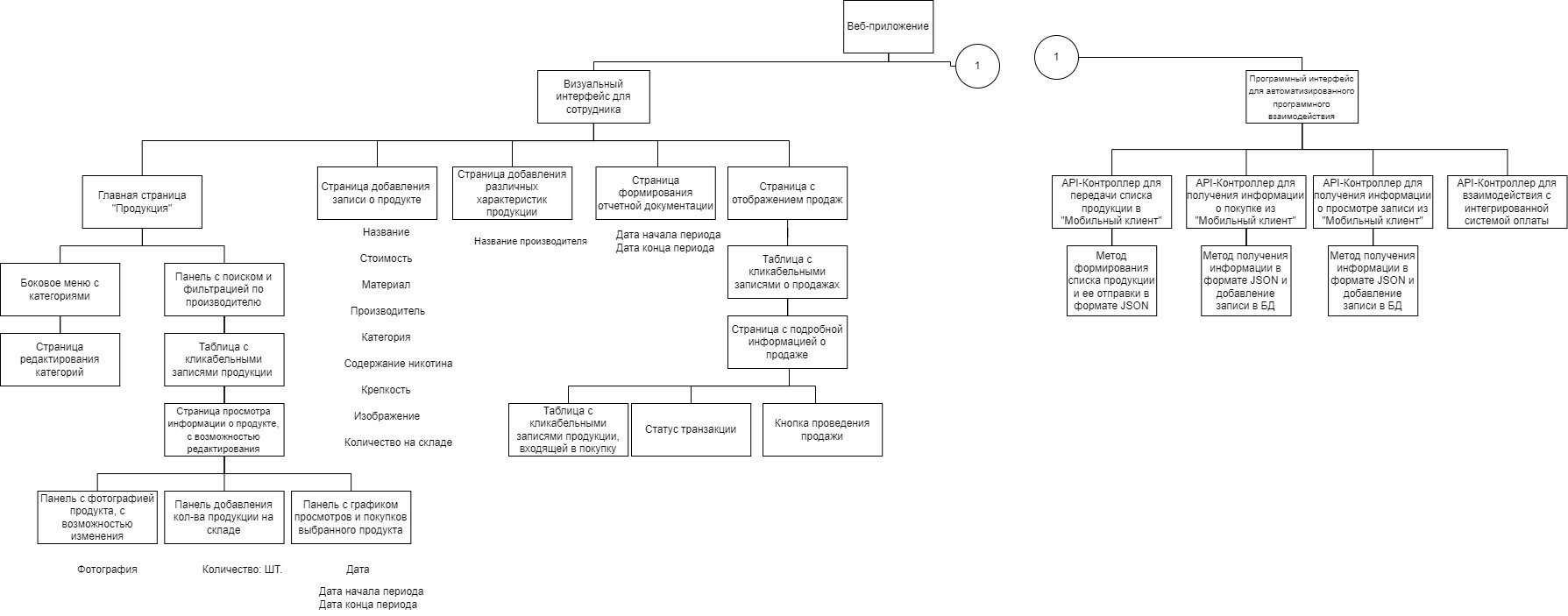


Рисунок 4 – общий функционал визуального интерфейса серверной части

На рисунке 5 представлена диаграмма с общим функционалом интерфейса автоматизированного программного взаимодействия.

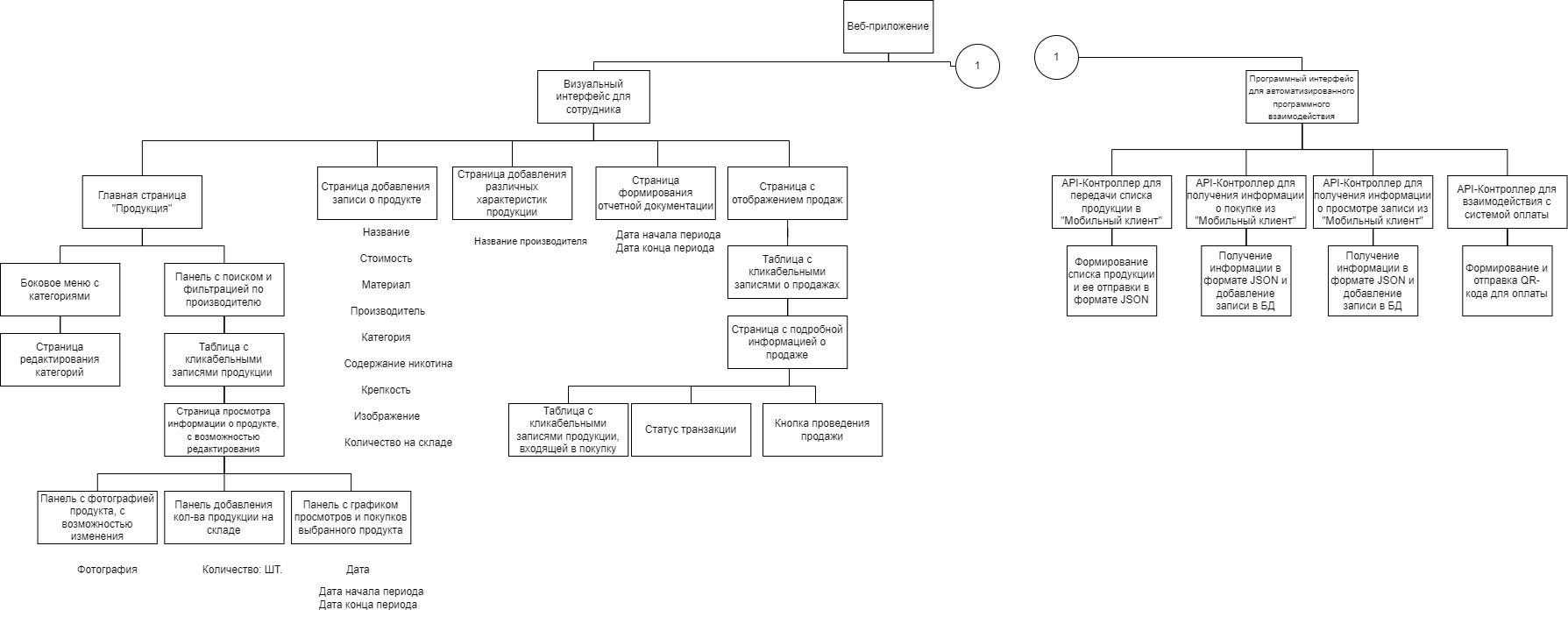


Рисунок 5 – интерфейс программного взаимодействия

На рисунке 6 представлена диаграмма с общим функционалом мобильной части ИС.

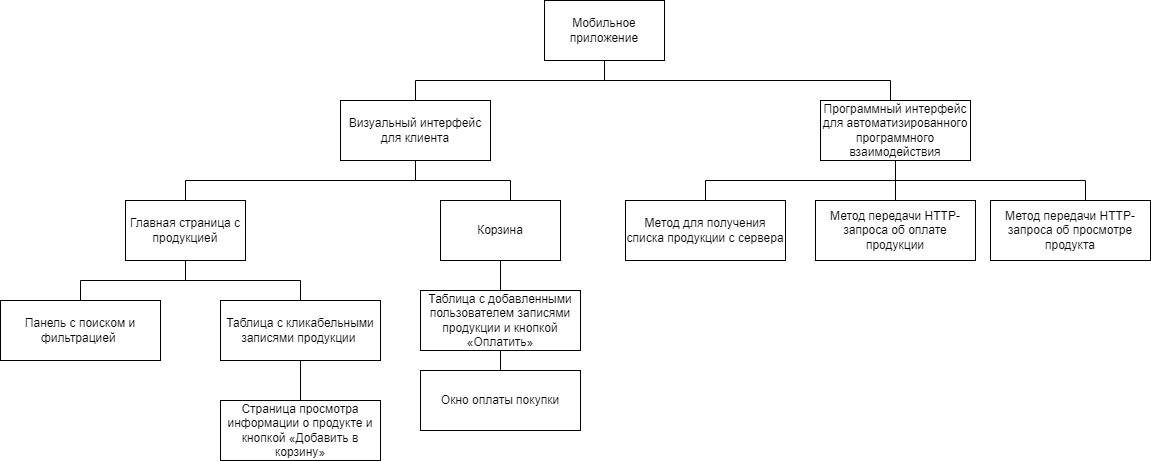


Рисунок 6 – общий функционал мобильной части ИС

## 1.5 Анализ программных ресурсов, необходимых в работе

Для разработки базы данных было решено использовать реляционную систему управления базами данных (далее – СУБД) – «MySQL». Данная система является хорошим решением для малых и средних приложений, а также свободно распространяется по лицензии «GPL» при условии, что программный код разрабатываемого продукта является открытым и свободным в использовании.

Данное СУБД обладает графическим интерфейсом, который позволяет наиболее удобно разрабатывать и конфигурировать базу данных – «MySQL Workbrench». Для работы с базами данных также используется язык SQL, который изучался в колледже. Описанные характеристики вполне доказывают обоснованность использования данной системы.

Для разработки веб-части информационной системы было решено использовать «ASP.NET Core» – технологию для создания веб-приложений на платформе .NET. Данная технология развивается компанией Microsoft, в качестве языков программирования могут использоваться C# и F#.

Данная технология позволяет разрабатывать логическую часть на языке C#, который является основным языком разработки в колледже, и верстку визуальных представлений на языках HTML и СSS, а также обладает функцией включения блоков кода C# в документ разметки HTML – функция Razor Pages, что предоставляет удобство при создании функций, таблиц и элементов управления.

Также стоит обратить внимание на используемые в технологии паттерн «ASP.NET Core MVC», позволяющий строить приложение вокруг трех основных компонентов: модели, представления и контроллера, и «ASP.NET Core Web API» – реализацию паттерна REST, при котором для каждого типа HTTP-запроса предназначен отдельный ресурс, что облегчает разработку API-интерфейса веб-части информационной системы.

Отдельно можно выделить следующие особенности ASP.NET Core:

* работа поверх платформы .NET и полное задействование ее функционала;
* кроссплатформенный фреймворк, приложения на котором могут быть развернуты на всех основных популярных информационных системах: Windows, Mac OS, Linux;
* фреймворк является модульным, что позволяет загружать все необходимые компоненты веб-приложения как отдельные модули через пакетный менеджер NuGet;
* в качестве инструментария разработки можно использовать такую среду разработки с богатым функционалом, как Visual Studio.

Выбор данной технологии обусловлен ее растущей популярностью, удобностью применяемых шаблонов проектирования и положительными особенностями, описанными выше.

Для обработки некоторых функций на стороне клиента предполагается использовать фреймворк JavaScript – JQuerry, который представляет набор функций JavaScript и фокусируется на взаимодействии JavaScript и HTML.

Для разработки мобильной части информационной системы было решено использовать новый кроссплатформенный фреймворк от компании Microsoft для создания мобильных и десктопных приложений – .NET Multi-platform App UI. Данная технология использует язык программирования C# и язык разметки XAML. С помощью данного фреймворка можно разрабатывать приложения под такие операционные системы как Android, iOS, macOS и Windows, используя при этом один и тот же код.

.NET Multi-platform App UI представляет эволюцию другого фреймворка – Xamarin Forms, имеющего более чем 10-летнюю историю разработки.

Основополагающим фактором выбора данного фреймворка является то, что при создании приложений мы можем использовать платформу .NET и язык программирования C#, а также вести разработку в среде VisualStudio.

Можно заключить, что выбор вышеописанных технологий обусловлен не только их растущей популярностью и востребованностью в мире, но также и удобностью использования – все части информационной системы, за исключением базы данных, разрабатываются в одной среде разработки VisualStudio, их логика разрабатывается на языке C# поверх платформы .NET, что позволит использовать различные дополнительные библиотеки как, например Entity Framework, которая позволяет облегчить взаимодействие с базой данных.

## 2 Технологический раздел

## Структура базы данных

База данных включает в себя 13 сущностей. На рисунке 7 представлена ER-диаграмма базы данных.

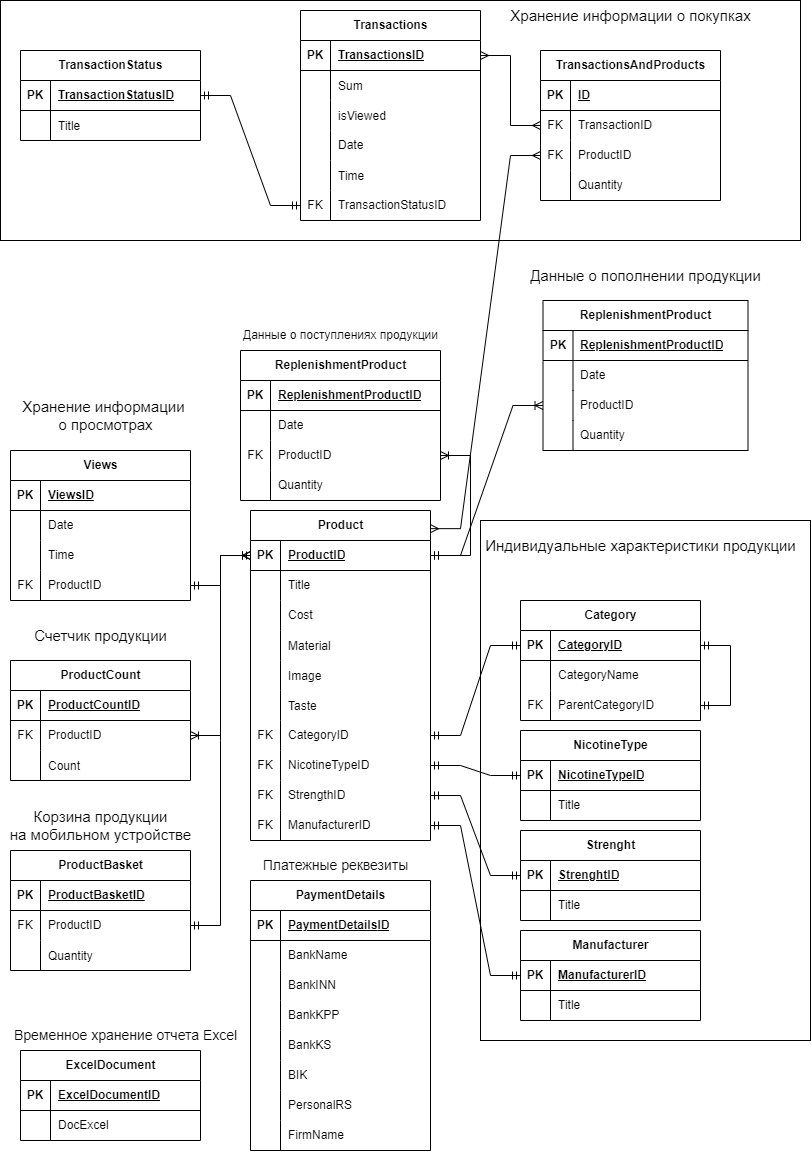


Рисунок 7 – ER-диаграмма базы данных

Сущность Manufacturer предназначена для хранения в базе данных информации о производителях продукции.

Таблица 13 – Сущность Manufacturer

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| ID | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| Title | Nvarchar | No | Название производителя |

Сущность Category предназначена для хранения в базе данных информации об имеющихся категориях и подкатегориях продукции.

Таблица 14 – Сущность Category

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| ID | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| CategoryName | Nvarchar | No | Название категории |
| ParentCategoryID | Int | Yes | Внешний ключ к родительской категории. При наличии значения в данном поле – экземпляр является подкатегорией |
| CategoryLevel | Int | No | Уровень категории. От 1 до 3 |

Сущность NicotineType предназначена для хранения информации о количестве никотина в определенной продукции.

Таблица 15 – Сущность NicotineType

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| ID | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| Title | Nvarchar | No | Количество содержания никотина |

Сущность Strenght предназначена для хранения информации о крепкости продукции определенного вида.

Таблица 16 – Сущность Strenght

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| ID | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| Title | Nvarchar | No | Крепкость |

Сущность ProductCount предназначена для хранения информации о количестве определенного продукта.

Таблица 17 – Сущность ProductCount

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| ID | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| ProductID | Int | No | Внешний ключ к таблице с информацией о продукте |
| Count | Int | Yes | Счетчик количества продукции |

Сущность Transactions предназначена для хранения информации о проведенных покупках продукции.

Таблица 18 – Сущность Transactions

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| ID | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| Date | Date | No | Дата проведения покупки |
| Time | Time | No | Время проведения покупки |
| Sum | Float | No | Информация о сумме покупки |
| TransactionStatusID | Int | No | Статус покупки |
| isViewed | Boolean | No | Статус просмотра транзакции |

Сущность Views предназначена для хранения информации о просмотрах записи о продукции.

Таблица 19 – Сущность Views

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| 1 | 2 | 3 | 4 |
| ID | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| Date | Date | No | Дата просмотра записи |

Окончание таблицы 19

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Time | Time | No | Время просмотра записи |
| ProductID | Int | No | Внешний ключ к таблице с информацией о продукте |

Сущность TransactionsAndProduct является навигационной таблицей, предназначенной для хранения информации о продукции, которая входила в покупку и соответствующей ей транзакции.

Таблица 20 – Сущность TransactionsAndProduct

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| ID | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| TransactionID | Int | No | Внешний ключ к таблице транзакций |
| ProductID | Int | No | Внешний ключ к таблице продукции |
| Quantitly | Int | No | Количество приобретенной продукции определенного вида |

Сущность Product предназначена для хранения информации о продукции.

Таблица 21 – Сущность Product

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| ID | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| Name | Nvarchar | No | Название продукции |
| Cost | Float | No | Стоимость продукции |
| Material | Nvarchar | No | Материал продукции |
| Image | Bit | Yes | Изображение продукции, если имеется |
| Taste | Nvarchar | Yes | Вкус продукции, если имеется |
| CategoryID | Int | No | Внешний ключ к таблице категории |
| NicotineTypeID | Int | Yes | Внешний ключ к таблице содержания никотина, если имеется |
| StrenghtID | Int | Yes | Внешний ключ к таблице крепкости продукции, если имеется |
| ManufacturerID | Int | No | Внешний ключ к таблице производителей продукции |

Сущность TransactionStatus предназначена для хранения информации о статусах покупки: Оплачена, Не оплачена, Отменена.

Таблица 22 – Сущность TransactionStatus

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| ID | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| Title | Nvarchar | No | Название статуса |

Сущность ReplenishmentProduct предназначена для хранения информации о поступлениях продукции.

Таблица 23 – Сущность ReplenishmentProduct

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| ID | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| Date | Datetime | No | Дата и время поступления продукции |
| ProductID | Int | No | Внешний ключ к таблице поставляемого продукта |
| Quantity | Int | No | Количество поставленного продукта |

Сущность ExcelDocument предназначена для временного хранения сгенерированного отчетного документа Excel.

Таблица 24 – Сущность ExcelDocument

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| ID | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| DocExcel | Varbinary | No | Документ Excel, записанный в виде массива байтов |

Сущность PaymentDetails предназначена для хранения платежных реквизитов, по которым генерируется QR-код для оплаты покупок.

Таблица 25 – Сущность PaymentDetails

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| 1 | 2 | 3 | 4 |
| ID | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| BankName | Nvarchar | No | Название банка |
| BankINN | Nvarchar | No | ИНН банка |
| BankKPP | Nvarchar | No | КПП банка |
| BankKS | Nvarchar | No | Корреспондентский счет банка |

Окончание таблицы 25

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| BIK | Nvarchar | No | БИК банка |
| PersonalRS | Nvarchar | No | Персональный расчетный счет |
| FirmName | Nvarchar | No | Название фирмы |

Сущность ProductBasket предназначена для хранения продуктов, добавленных пользователем в корзину на мобильном устройстве.

Таблица 26 – Сущность ProductBasket

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| ID | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| ProductID | Int | No | Внешний ключ к таблице продукции |
| Quantity | Int | No | Количество продукции |

## Разработка форм выходных документов

В ИС в качестве выходного документа можно выделить автоматически генерируемый отчетный документ для использования внутри магазина.

Данный документ создается в серверной части ИС. Для его генерации сотруднику необходимо указать период времени, за который необходимо собрать данные и сопоставить их в отчетный документ. На рисунке 8 показано окно выбора временного промежутка на странице работы с документацией.

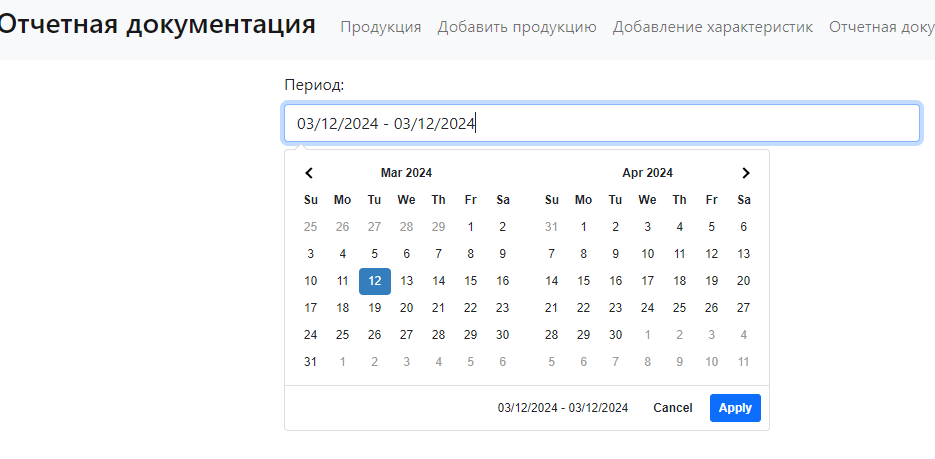


Рисунок 8 – выбор временного промежутка для генерации отчета

Сам же метод генерации документа получает в качестве входных параметров две переменные – дату старта промежутка и дату конца промежутка. Если выбрана одна дата, то переменные равны между собой и отчет генерируется за один день. После указания дат система собирает необходимые данные из базы данных и передает их в метод генерации документа.

Для работы с Excel документом используется библиотека EPPlus версии 4.5.3.3. Данная версия полностью свободна для любого вида использования. Листинг кода методов представлен в листинге 1 приложения А.

Отчетный документ включает в себя такие параметры, как:

* общее число продаж;
* сумма продаж;
* остаток продукции;
* поступление продукции за период;
* список оставшейся продукции (название, остаток, стоимость);
* список поступившей продукции (название, стоимость, количество поступления, дата поступления).

На рисунке 9 представлен образец содержания отчетного документа.

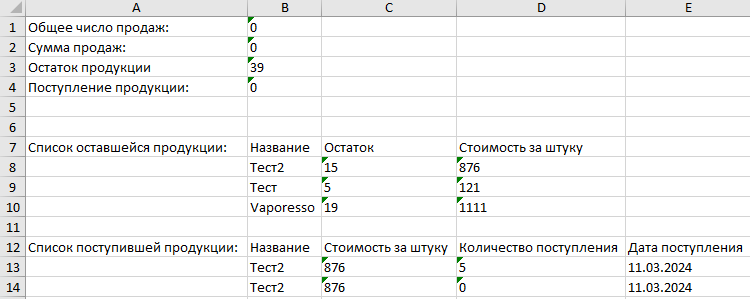


Рисунок 9 – образец содержания отчетного документа

Отчет является внутренним и позволяет получить основную информацию о движении продукции в магазине.

## Разработка структуры программного продукта

Для реализации архитектуры серверной части ИС было решено использовать шаблон проектирования MVC – Model-View-Controller. Данный шаблон предполагает разделение пользовательского интерфейса (представления), данных (модели) и логики приложения (контроллера) и помогает добиться разделения задач.

Шаблон MVC имеет следующую логику взаимодействия: запросы направляются в контроллер, который отвечает за работу с моделью и обеспечивает выполнение действий с данными и / или их извлечение. В представлении же отображается конечная страница на основе данных из модели. На рисунке 10 изображена схема работы шаблона MVC.

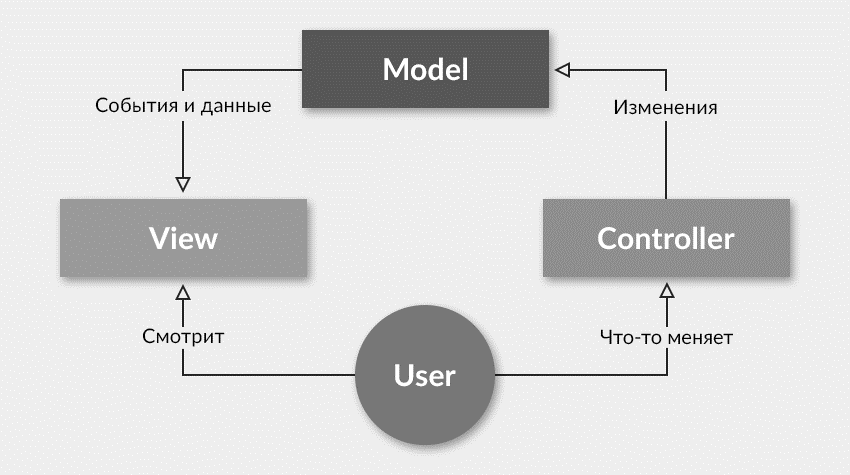


Рисунок 10 – схема работы шаблона MVC

Взаимодействие между серверной и мобильной частью происходит посредством реализации API-интерфейсов четырех видов: GET, POST, PUT и DELETE. API интерфейс вида GET позволяет мобильному приложению получать с сервера список продукции в текстовом формате обмена данными, основанном на языке JavaScript – JSON. Запросы вида POST и PUT позволяют мобильному приложению передавать определенные данные на сервер, в нашем случае передается информация о просмотрах и покупках товаров, осуществленных в мобильном приложении. DELETE же удаляет запись. При получении QR-кода для оплаты, из мобильного приложения не только происходит запрос, но также передаются определенные данные о продукции, а также итоговая сумма, после чего происходит отправка сгенерированного кода в виде массива байтов. Программный код API-интерфейсов представлен в листинге 2 приложения А.

Файлы серверной части ИС представлены в таблице 26

Таблица 26 – Файлы, входящие в серверную часть программного продукта

|  |  |  |
| --- | --- | --- |
| Наименование | Назначение | Примечание |
| 1 | 2 | 3 |
| Properties/launchSettings.json | Набор шаблонов веб-приложения | Содержит список источников конфигурации по умолчанию от наиболее до наименее приоритетных |
| wwwroot/css/bootstrap.min.css | Фреймворк с набором готовых CSS стилей | Содержит набор различных CSS стилей, применяемых при верстке HTML страниц |
| wwwroot/css/daterangepicker.css | Фреймворк с набором готовых CSS стилей | Содержит набор различных CSS стилей, необходимых для построения временного графика |
| wwwroot/css/DocsSheet.css | Набор CSS стилей | Содержит набор стилей, необходимых для страниц, работающих с документацией |
| wwwroot/css/homeStyle.css | Набор CSS стилей | Содержит набор стилей, необходимых для основных страниц |
| wwwroot/js | Папка, содержащая javascript файлы | В папке находятся файлы со скриптами на языке JavaScript |
| wwwroot/js/datepicker | Папка, содержащая javascript файлы | В папке находятся файлы со скриптами на языке JavaScript, необходимые для работы выборщика временных промежутков |
| wwwroot/js/diagram | Папка, содержащая javascript файлы | В папке находятся файлы со скриптами на языке JavaScript, необходимые для работы временного графика |
| API | Папка содержащая API-контроллеры | В папке находятся C# классы, реализующие технологию Web API |
| API/ProductController.cs | Класс, реализующий API-контроллер | Реализует отправку JSON, содержащего информацию о продукции |
| API/QrController | Класс, реализующий API-контроллер | Реализует принятие параметров, включающих стоимость и отправляет сгенерированный QR-код для оплаты в виде массива байтов |
| API/ViewController | Класс, реализующий API-контроллер | Реализует принятие параметров даты, времени и идентификатора просмотренной продукции |

Продолжение таблицы 26

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Controllers | Папка, содержащая в себе классы контролера. Одна из составляющих паттерна MVC | Содержит в себе классы контроллера |
| Controllers/HomeController.cs | Класс контроллера | Класс, реализующий взаимодействие между представлениями проекта и моделью данных, работает со страницами информации о товарах |
| Controllers/DocsController.cs | Класс контроллера | Класс, реализующий взаимодействие между представлениями проекта и моделью данных, работает со страницами генерации документации и ее импорта в систему |
| Models | Папка, содержащая классы модели данных проекта | Содержит в себе классы модели данных |
| Models/ViewModels | Папка, содержащая классы моделей представления | Содержит в себе классы модели представления, которая собирает в себе определенные данные из нескольких моделей данных и формирует их в необходимом виде для передачи в представление |
| ViewModels/PaymentManufacturer.cs | Класс модели представления | Содержит в себе данные из модели данных о производителе и модели с платежными реквезитами |
| ViewModels/ProductsAttributesCategories.cs | Класс модели представления | Содержит в себе данные из модели данных о продуктах, аттрибутах и категориях |

Продолжение таблицы 26

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| ViewModels/ProductsTransactionsWithStatus.cs | Класс модели представления | Содержит в себе данные из модели данных о продуктах, транзакциях и параметр статуса |
| ViewModels/ProductsViewsTransactions.cs | Класс модели представления | Содержит в себе данные из модели данных о продуктах, просмотрах и транзакицях |
| Models/НазваниеКласса | Классы модели данных | Классы содержат в себе описание атрибутов одноименных сущностей, полученных из базы данных. В связи с однотипностью и большим количеством подобных классов не имеет смысла описывать каждый подробно. |
| Models/VapeshopContext.cs | Класс контекста базы данных | Содержит в себе ссылку на экземпляр класса Product, а также дополнительные списки, расширяющие набор данных. Необходим для передачи наиболее полной информации в страницу представления |
| Models/ProductDTO.cs | Класс модели данных о продуктах для отправки в виде JSON | Содержит в себе описание атрибутов сущности Product, но исключает внешние ключи, заменяя их определенными атрибутами из связных таблиц |

Продолжение таблицы 26

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Utils | Папка, содержащая дополнительные классы – утилиты | Содержит в себе вспомогательные файлы и классы, необходимые для валидации и работы с базой данных |
| Utils/ValidatewPhoto.cs | Класс валидации фотографии | Содержит в себе логику проведения валидации фотографии – устанавливает, является ли загруженный в систему файл фотографией |
| Utils/stub.jpg | Фотография-заглушка | Если фотография не была загружена в систему, или загруженный файл не является фотографией, то используется данная фотография-заглушка |
| Utils/Recorder.cs | Класс рекордера | Содержит в себе методы, для взаимодействующие с базой данных – получение различных данных. Необходим, чтобы исключить «избавить» основной код от однотипных методов |
| Utils/ExcelExtensions.cs | Класс работы с документами Excel | Содержит методы для генерации Excel документа, а также считывания импортированного в систему файла. |
| Utils/Форма.xlsx | Файл Excel | Форма для заполнения списком продукции с целью дальнейшей загрузки информации в систему |
| Views | Папка, содержащая страницы представления, элемент MVC | Содержит страницы представления, которые определяют внешний вид приложения и на основе которых потом формируется веб-страница |
| Views/Home | Папка, содержащая страницы представления, работающие с контроллером Home | Содержит страницы типа HTML, необходимые для определения внешнего вида, , содержат в себе код на языках разметки HTML и CSS, а также блоки кода на C# |
| Views/Docs | Папка, содержащая страницы представления, работающие с контроллером Docs | Содержит страницы типа HTML, необходимые для определения внешнего вида, содержат в себе код на языках разметки HTML и CSS, а также блоки кода на C# |

Окончание таблицы 26

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Views/Shared/\_Layout.cshtml | HTML мастер-страница представления | Необходима для определения внешнего вида элементов страницы, которые будут находится на всех остальных страницах, содержит в себе код на языках разметки HTML и CSS |
| appsettings.json | Файл конфигурации приложения | Содержит в себе информацию о параметрах проекта |
| Program.cs | Основной класс сборки приложения | Содержит в себе методы, реализующие сборку и работу проекта, настраивает маршрутизацию |

Мобильная часть ИС реализована с использованием паттерна MVVM – Model-View-ViewModel. Данный шаблон похож на используемый в серверной части MVC. Он также позволяет отделить логику приложения от визуальной части (представления) и задает общую архитектуру приложения. На рисунке 11 представлена структура шаблона MVVM.



Рисунок 11 – структура шаблона MVVM

Файлы мобильной части ИС представлены в таблице 27

Таблица 27 – Файлы мобильной части ИС

|  |  |  |
| --- | --- | --- |
| Наименование | Назначение | Примечание |
| 1 | 2 | 3 |
| Properties/launchSettings.json | Набор шаблонов для приложения | Содержит список источников конфигурации по умолчанию от наиболее до наименее приоритетных |
| Models | Папка, содержащая классы модели данных проекта | Содержит в себе классы модели данных |

Окончание таблицы 27

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Models/Product.cs | Класс модели данных о продуктах | Содержит в себе описание атрибутов сущности Product, необходим для получения данных из запроса |
| Models/ViewDTO.cs | Класс модели данных о просмотре продукции | Используется для организации данных о просмотрах для передачи на сервер |
| Resources | Папка, содержащая различные ресурсы приложения | Содержит в себе различные ресурсы, необходимые для разработки дизайна приложения, такие как шрифты, изображения, иконки и т.п. |
| ViewModels | Папка, содержащая в себе классы моделей представления. Один из элементов паттерна MVVM | Содержит в себе классы моделей представления |
| ViewModels/ProductViewModel.cs | Класс модели представления | Содержит в себе различные методы, поля и функции, необходимые для взаимодействия между представлением и моделью с информацией о продукции |
| ViewModels/InfoViewModel.cs | Класс модели представления | Содержит в себе различные методы, поля и функции, необходимые для взаимодействия между представлением и моделью |
| Views | Папка, содержащая страницы отображения | Содержит страницы представления, которые определяют внешний вид приложения |

## Описание процедур и функций

Процедуры и функции, реализованные в серверной части ИС представлены в таблице 28

Таблица 28 – Процедуры и функции серверной части ИС

|  |  |
| --- | --- |
| Имя процедуры | Назначение |
| 1 | 2 |
| <ActionResult<IEnumerable<ProductDTO>>> Get()  в API/ProductController.cs | Асинхронная функция, являющаяся реализацией HTTP-GET запроса. Обращается через контекст БД к сущности Product, получает все имеющиеся данные в виде списка экземпляров класса Product, после чего передает данные в специальный класс для отправки – ProductDTO и отправляет их получателю в виде JSON |
| <ActionResult<byte[]>>Get(string sum)  в API/QrController.cs | Асинхронная функция, являющаяся реализацией HTTP-GET запроса. Получает из запроса мобильного приложения сумму, на которую был набран товар, после чего генерирует QR-код для оплаты и отправляет его в мобильное приложение в качестве ответа в виде массива байтов |
| Async void Post([FromBody] View DTO)  в API/ViewController.cs | Асинхронный метод, являющийся реализацией HTTP-POST запроса. Получает из запроса мобильного приложения JSON файл, содержащий параметры экземпляра класса, отвечающего за единичный просмотр продукции, после чего записывает его в базу данных. |
| Put([FromBody] ProductToB p)  в API/BusketController.cs | Асинхронный метод, являющийся реализацией HTTP-PUT запроса. Получает из тела запроса модель данных о продукте добавленном в корзину мобильного приложения и записывает в базу данных |
| Delete()  в API/BusketController.cs | Асинхронный метод, являющийся реализацией HTTP-DELTE запроса. Удаляет запись из корзины |
| IActionResult ExcelDocCreate()  в Controllers/DocsController.cs | Метод, возвращающий HTML страницу ExcelDocCreate. |
| IActionResult AllTransactions()  в Controllers/DocsController.cs | Метод, возвращающий HTML страницу AllTransactions. Передает в страницу данные о всех транзакциях. |
| IActionResult InfoTransactions(int IdTransact)  в Controllers/DocsController.cs | Метод, возвращающий HTML страницу InfoTransactions. Получает идентификатор транзакции, о которой необходимо вывести всю информацию, после чего получает информацию о статусе транзакции, продуктах, входящих в транзакцию; представляет полученные данные в виде модели отображения и передает данную модель на страницу. |
| <IActionResult> excelToDatabase(IFormFile excel)  в Controllers/DocsController.cs | Асинхронный HTTP-POST метод, получающий из страницы отображения файл Excel, после чего отправляющий его в метод, возвращающий восстановленные данные из Excel файла в виде списка экземпляров модели данных, записывает полученную информацию в базу данных |
| <IActionResult> SetAccept(int idTransaction)  в Controllers/DocsController.cs | Асинхронный HTTP-POST метод, вызываемый при просмотре транзакции и изменяющий ее статус в базе данных |
| <IActionResult>CreateDocument (DateOnly dateStart, dateEnd)  в Controllers/DocsController.cs | Асинхронный HTTP-POST метод, получающий из представления дату начала и конца промежутка, за который необходимо сгенерировать отчетный документ. После получения временных промежутков, собирает данные из базы и передает их в виде созданной модели данных в функцию генерации документа. Полученный документ записывает в базу данных и передает идентификатор в метод загрузки файла. |
| <IActionResult>Download(int id)  в Controllers/DocsController.cs | Асинхронный метод, получающий идентификатор записи созданного отчетного документа и производящий его скачивание на компьютер. |
| IActionResult getExcelForm()  в Controllers/DocsController.cs | Метод, позволяющий скачать заранее подготовленную форму, используемую для заполнения списком продукции |
| IActionResult Index  в Controllers/HomeController.cs | Метод, возвращающий HTML страницу Index. Передает в страницу модель отображения с необходимыми данными. Реализует работу отображения данных, их поиска и фильтрации |
| IActionResult Menu()  в Controllers/HomeController.cs | Метод, возвращающий HTML страницу Menu, которая является частичным представлением и отображается одновременно со страницей Index |
| IActionResult AddManufacturer()  в Controllers/HomeController.cs | Метод, возвращающий HTML страницу AddManufacturer |
| IActionResult UpdateCategory()  в Controllers/HomeController.cs | Функция, возвращающая HTML страницу UpdateCategory. Передает в отображение модель данных, содержащую категории продукции в виде списка при помощи словаря ViewBag |
| IActionResult AddProduct()  в Controllers/HomeController.cs | Метод, возвращающий HTML страницу AddProduct. Передает в страницу данные об атрибутах сущности Product в виде списка при помощи словаря ViewBag |
| IActionResult AboutProduct(int id)  в Controllers/HomeController.cs | Метод возвращающий HTML страницу AboutProduct, передает в страницу основные данные о выбранном продукте, полученные из базы данных |
| Delete(int id)  в Controllers/HomeController.cs | Метод, реализующий поиск выбранного объекта по ID и его удаление из базы данных |
| GetDataAboutViews(int idProduct, DateOnly dateStart, dateEnd)  в Controllers/HomeController.cs | HTTP-POST функция, возвращающая полученную информацию о просмотрах и покупках выбранной продукции на страницу в виде JSON файла |
| \_Quantity(int idProduct, int quantity)  в Controllers/HomeController.cs | HTTP-POST функция, получающая со страницы данные о пополнении количества определенной продукции на складе и записывающая данную информацию в базу данных, возвращает обновленное количество продукции на страницу в виде файла JSON |
| UpdateProduct()  в Controllers/HomeController.cs | HTTP-POST функция, получающая из страницы обновления продукта обновленную модель данных и записывающая ее в базу данных |
| UpdatePayment()  в Controllers/HomeController.cs | HTTP-POST функция, получающая из страницы настроек обновленную модель данных о платежных реквизитах и записывающая ее в базу данных |
| CreateManufacturer()  в Controllers/HomeController.cs | HTTP-POST функция, получающая со страницы настроек модель данных о новом производителе продукции и записывающая ее в базу данных |
| CreateProduct()  в Controllers/HomeController.cs | HTTP-POST функция, получающая со страницы добавления продукции модель данных с новой записью о продукте и записывающая ее в базу данных |
| CreateCategory()  в Controllers/HomeController.cs | HTTP-POST функция, получающая со страницы создания категории модель данных с новой записью о категории и записывающая ее в базу данных |
| RenameCategory()  в Controllers/HomeController.cs | HTTP-POST функция, получающая со страницы обновления категории модель данных с обновленной записью о категории и записывающая ее в базу данных |
| GetParentCategory()  в Controllers/HomeController.cs | HTTP-POST функция, передающая на страницу данные о родительских категориях для определенного уровня категорий в виде файла JSON |
| byte[] Generate()  в Utils/ExcelExtensions.cs | Функция, возвращающая сгенерированный Excel документ в виде массива байтов. Получает модель с необходимыми данными, после чего записывает их в документ формата xlsx |
| bool columnValidate()  в Utils/ExcelExtensions.cs | Булева функция, проверяющая импортированный Excel документ на правильность столбцов, при правильности возвращает True |
| IEnumerable<T> FromSheetToModel()  в Utils/ExcelExtensions.cs | Функция, получающая импортированный в систему документ Excel, считывающая его и преобразующая в коллекцию. |

## 3 Инструкции пользователям системы

Серверная часть ИС имеет следующий набор функций и задач:

* просмотр записей продукции, включая поиск по названию и сортировку по категориям;
* добавление записи о продукте;
* редактирование и удаление записи о продукте;
* формирование графика о просмотрах и покупках продукции;
* отображение подробной информации о продукте;
* формирование внутреннего отчетного документа в Excel;
* добавление записей продукции списком при помощи импорта Excel файла;
* отображение списка проведенных продаж и вывод списка продукции, проведенной по ним;
* заполнение платежных данных;
* добавление и редактирование категорий, по которым происходит фильтрация записей продукции.

Ниже приведено описание пользовательских операций для выполнения основных задач.

**Задача 1: Создание, редактирование и удаление записи продукции**

Условия, при соблюдении которых возможно выполнение задачи:

* устройство включено;
* веб-приложение доступно и запущено;
* веб-приложение исправно функционирует.

Подготовительные действия:

* открытие окна веб-приложения в браузере;
* информация о добавляемом продукте известна пользователю.

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 45 секунд.

Основные действия в требуемой последовательности:

1. В открывшемся главном окне, на верхней навигационной панели необходимо нажать на вкладку «Добавить продукцию», чтобы перейти на страницу добавления записи продукции. На рисунке 11 представлена часть главного окна с верхней навигационной панелью.

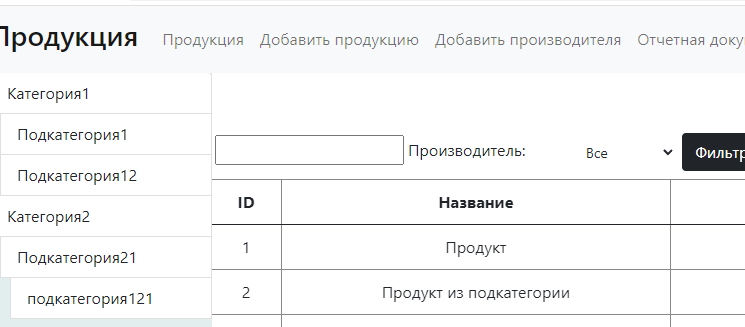


Рисунок 11 – часть главного окна и вкладка «Добавить продукцию»

1. Дождаться перехода на страницу добавления продукции, после чего начать ввод данных. Необходимо заполнить все поля, кроме фотографии. Если не добавлять фотографию, то автоматически будет вставлено изображение-заглушка. На рисунке 12 представлена страница добавления данных продукции с заполненными полями.

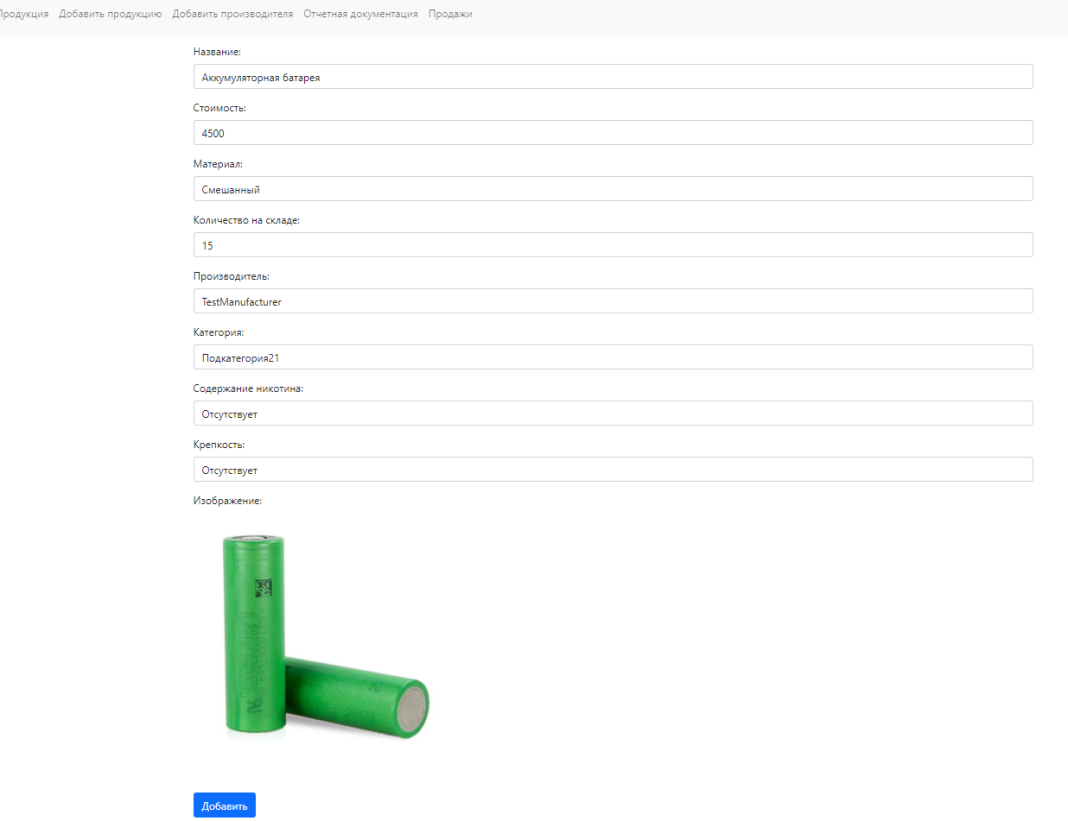


Рисунок 12 – страница добавления продукции

1. После заполнения полей необходимо нажать кнопку добавить, после чего произойдет переход на главную страницу, на которой будет находится добавленная запись. Главная страница с добавленной записью изображена на рисунке 13

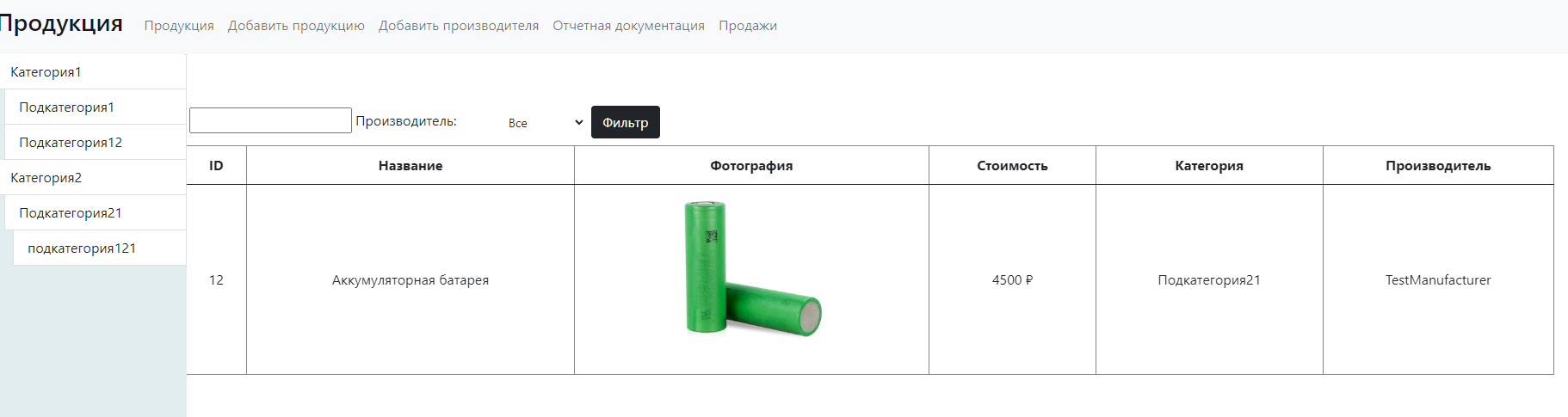


Рисунок 13 – главная страница с добавленной записью

1. Для редактирования и удаления записи необходимо выбрать запись – навести курсор мыши на неё и она будет подсвечена оранжевым цветом. Выбор записи показан на рисунке 14

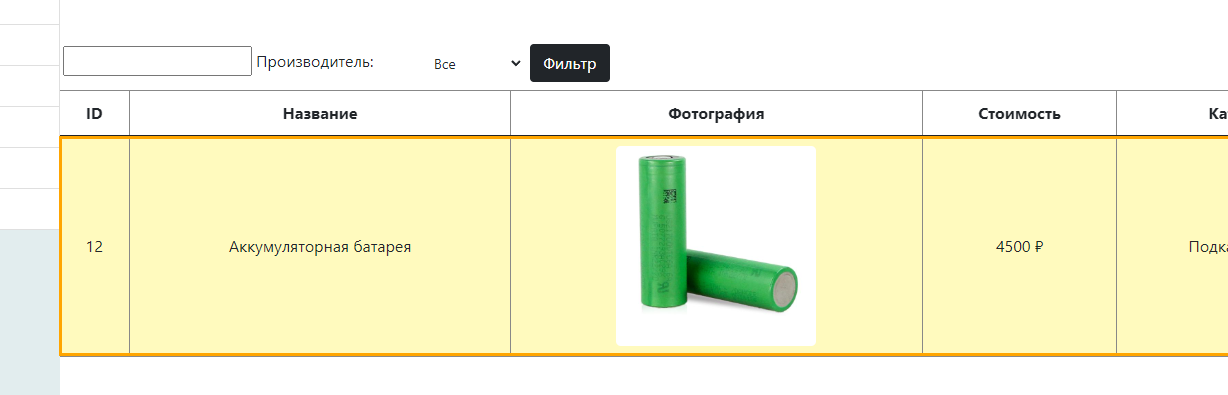


Рисунок 14 – выбранная запись

1. После выбора записи необходимо нажать левую кнопку мыши (ЛКМ), после чего произойдет переход на страницу с характеристиками записи. Страница с характеристиками записи представлена на рисунке 15

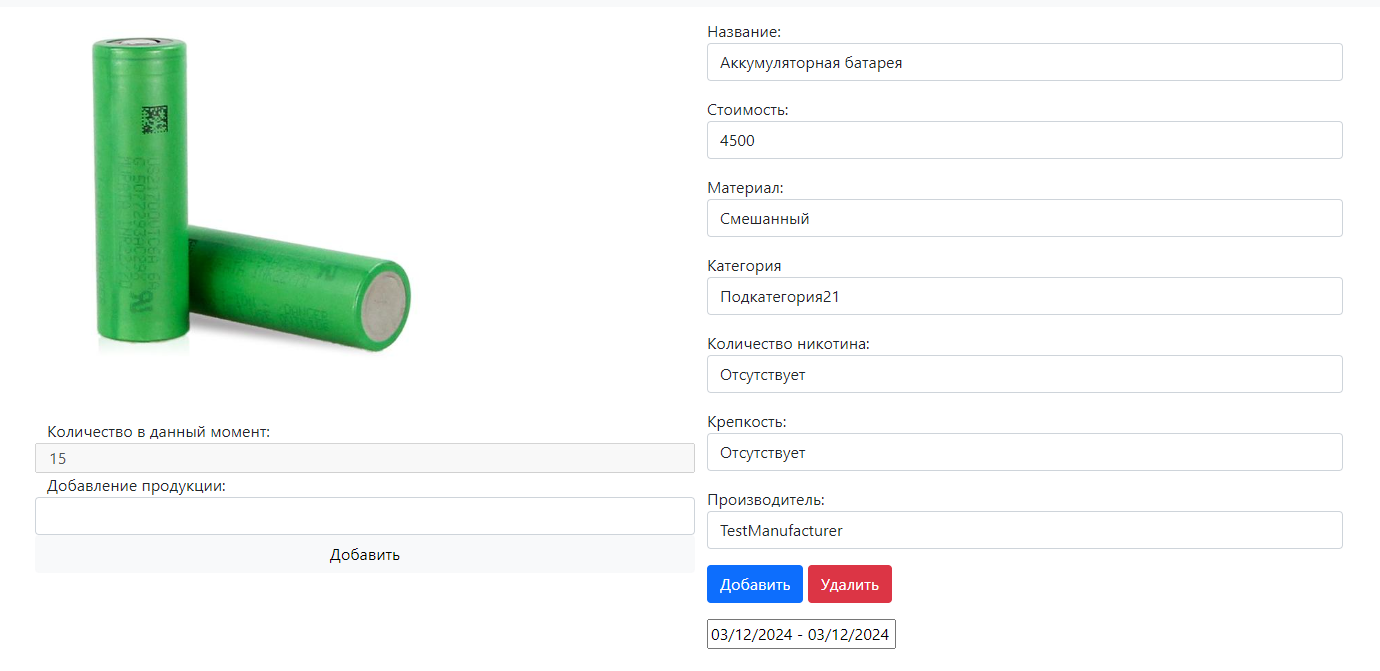


Рисунок 15 – страница с характеристиками записи

1. На данной странице пользователь может изменять атрибуты продукции, для сохранения которых необходимо нажать на кнопку «Добавить» синего цвета, для изменения фотографии – нажать на фотографию, после чего откроется диалоговое окно для выбора файла фотографии, после выбора также нажать кнопку «Добавить» синего цвета. Для пополнения продукции на складе необходимо заполнить поле «Добавление продукции» и нажать кнопку «Добавить» серого цвета.
2. После нажатия кнопки «Добавить» синего цвета происходит изменение записи продукции и переход на главную страницу.
3. Для удаления записи необходимо нажать на кнопку «Удалить», после чего произойдет удаление записи и переход на главную страницу.

**Задача 2: Формирование графика просмотров и продаж продукта**

Условия, при соблюдении которых возможно выполнение задачи:

* Устройство включено;
* веб-приложение доступно и запущено;
* веб-приложение исправно функционирует;
* наличие записей продуктов.

Подготовительные действия:

* Открытие окна веб-приложения в браузере.

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 35 секунд.

Основные действия в требуемой последовательности:

1. В открывшемся главном окне, в списке продукции необходимо выбрать запись продукции, информацию о которой необходимо узнать. На рисунке 16 показан выбор продукции.

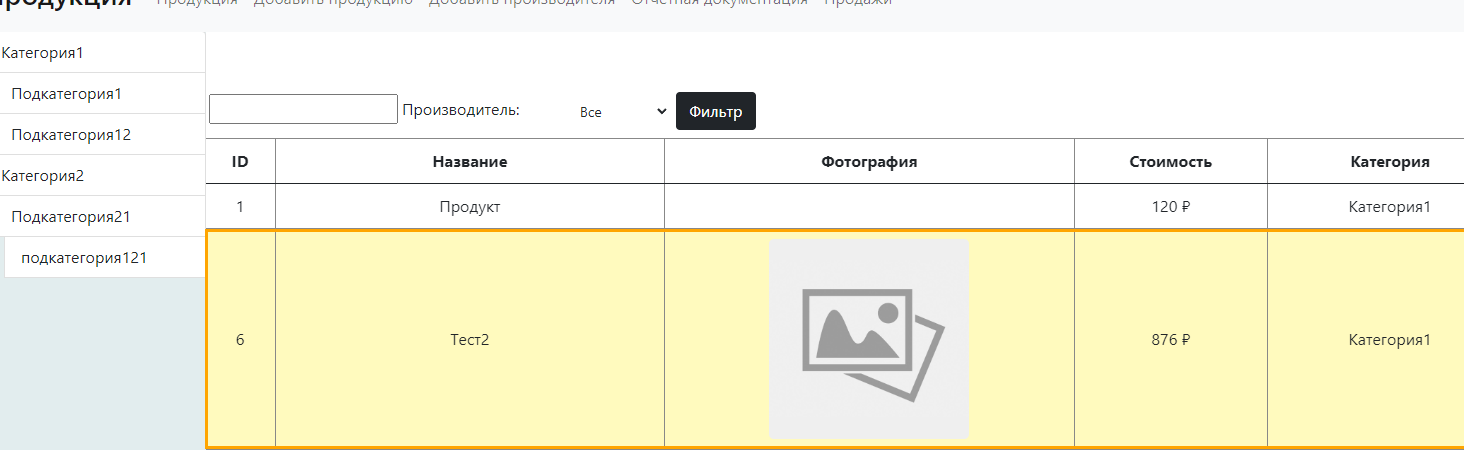


Рисунок 16 – выбранная запись

1. После выбора записи необходимо нажать левую кнопку мыши (ЛКМ), после чего произойдет переход на страницу с характеристиками записи. Там в поле выбора промежутка времени необходимо выбрать дату начала и конца промежутка, либо же выбрать одну дату двумя нажатиями ЛКМ на число. Выбор временного промежутка записи представлен на рисунке 17

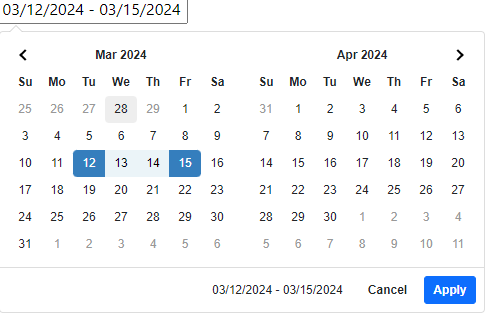


Рисунок 17 – выбор временного промежутка

1. После выбора временного промежутка необходимо нажать на клавишу «Apply», после чего будет сформирован график просмотров и покупок продукции. График показан на рисунке 18

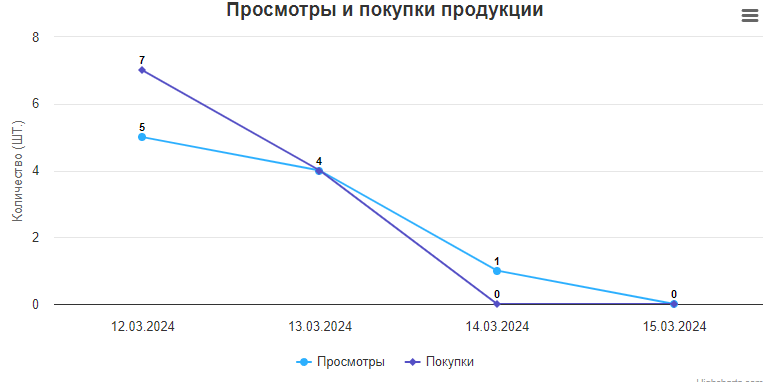


Рисунок 18 – график просмотров и покупок продукции

**Задача 3: Формирование отчетного документа Excel**

Условия, при соблюдении которых возможно выполнение задачи:

* устройство включено;
* веб-приложение доступно и запущено;
* веб-приложение исправно функционирует;
* наличие записей продуктов.

Подготовительные действия:

* открытие окна веб-приложения в браузере.

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 20 секунд.

Основные действия в требуемой последовательности:

1. В открывшемся главном окне, на верхней навигационной панели необходимо нажать на вкладку «Отчетная документация», чтобы перейти на страницу создания отчета. На рисунке 19 представлено часть главного окна с верхней навигационной панелью.

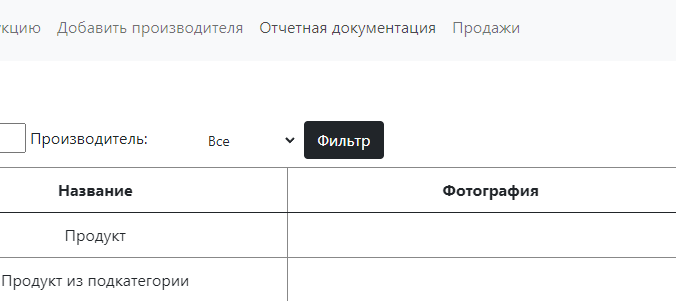


Рисунок 19 – часть навигационной панели и вкладка «Отчетная документация»

1. После нажатия на вкладку произойдет переход на страницу работы с документами, на которой нужно выбрать временной период, за который необходимо собрать информацию и нажать клавишу «Apply», после чего произойдет скачивание Excel документа. Страница отчетной документации представлена на рисунке 20

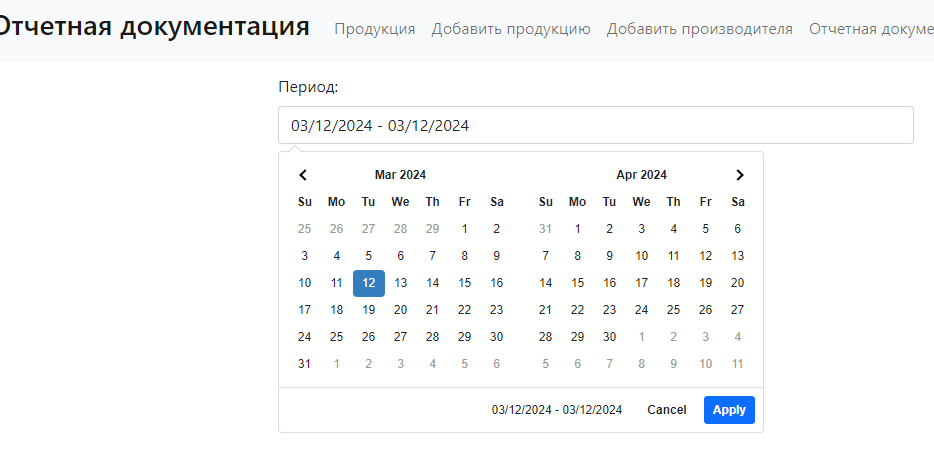


Рисунок 20 – страница отчетной документации и окно выбора периода

**Задача 4: Загрузка продукции списком через импорт Excel**

Условия, при соблюдении которых возможно выполнение задачи:

* устройство включено;
* веб-приложение доступно и запущено;
* веб-приложение исправно функционирует;
* наличие Excel документа с записями продуктов.

Подготовительные действия:

* открытие окна веб-приложения в браузере.

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 120 секунд.

Основные действия в требуемой последовательности:

1. В открывшемся главном окне, на верхней навигационной панели необходимо нажать на вкладку «Отчетная документация», чтобы перейти на страницу создания отчета.
2. После нажатия на вкладку произойдет переход на страницу работы с документами, после чего необходимо нажать на кнопку «Получить форму заполнения» для того, чтобы скачать форму, в которую необходимо вставить данные о продуктах. На рисунке 21 представлена страница работы с документами.

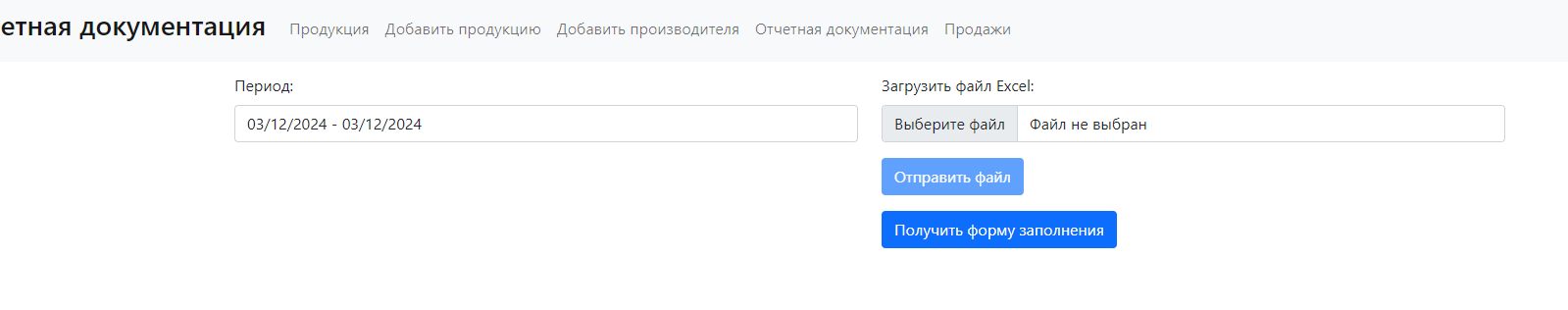


Рисунок 21 – страница работы с документами и поле для загрузки файла

На рисунке 22 представлена скачанная форма заполнения.

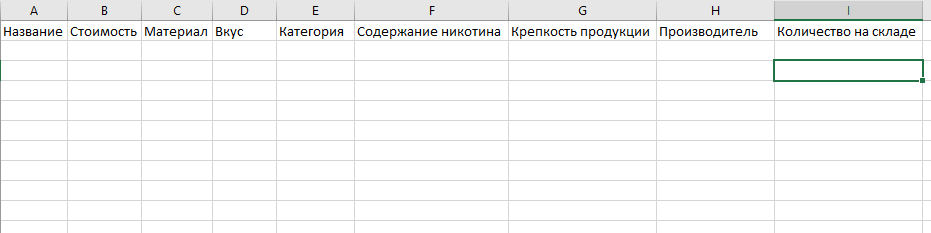


Рисунок 22 – форма заполнения

1. Необходимо заполнить и сохранить документ, закрыть его. После чего в веб-приложении выбрать этот файл. После успешной проверки его расширения нажать на кнопку «Отправить файл», которая станет доступной.
2. После отправления файла произойдет запись информации из Excel файла в базу данных и переход на главную страницу, на которой уже будут отображены добавленные записи.

Важно заметить, что записи, которые не прошли проверку не добавляются.

**Задача 5: Контроль проводимых транзакций**

Условия, при соблюдении которых возможно выполнение задачи:

* устройство включено;
* веб-приложение доступно и запущено;
* веб-приложение исправно функционирует;
* транзакции проводились.

Подготовительные действия:

* открытие окна веб-приложения в браузере.

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 40 секунд.

Основные действия в требуемой последовательности:

1. В открывшемся главном окне, на верхней навигационной панели необходимо нажать на вкладку «Продажи», чтобы перейти на страницу просмотра транзакций. На рисунке 23 изображена часть навигационной панели с вкладкой «Продажи».

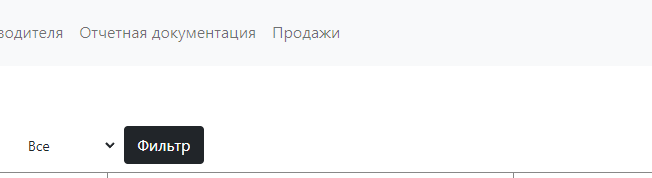


Рисунок 23 – часть навигационной панели с вкладкой «Продажи»

1. После нажатия на вкладку произойдет переход на страницу со всеми транзакциями. Записи имеют цветовую индикацию: Не просмотрена или отменена – красный; Просмотрена, но не оплачена – Желтый; Просмотрена и оплачена – синий. Страница с транзакциями изображена на рисунке 24

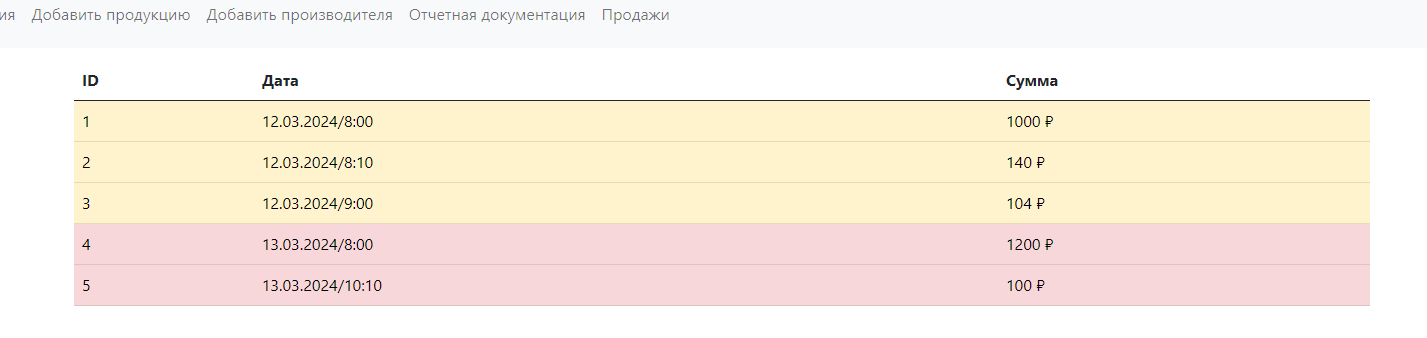


Рисунок 24 – страница с транзакциями

1. При нажатии на любую запись происходит переход к подробной информации о транзакции, туда входят записи купленных продуктов, информация о статусе транзакции (Не оплачена; Оплачена; Отменена), а также кнопка подтверждения транзакции. При нажатии на запись продукта происходит переход на страницу информации об этом продукте. Страница с подробной информацией о транзакции представлена на рисунке 25

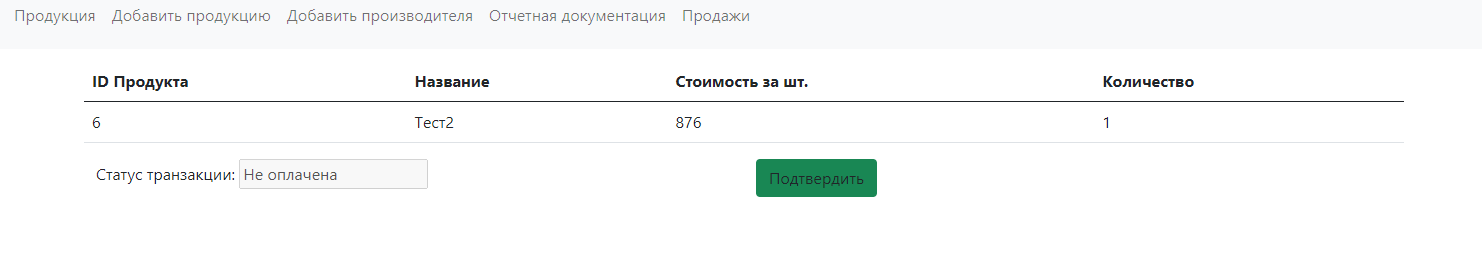


Рисунок 25 – страница с подробной информацией о транзакции