

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Системотехніки
(повна назва)

АТЕСТАЦІЙНА РОБОТА (ПРОЕКТ)
Пояснювальна записка

Бакалавр
(освітньо-кваліфікаційний рівень)

ГЮИК 502520.001 ПЗ
(позначення документа)

Система Internet-продажу кондитерських виробів
(тема)

Виконав: студент 4 курсу, групи СІ-12-1
напряму підготовки (спеціальності)
6.050201 – системна інженерія
(шифр і назва напряму, спеціальності)

Ахмад Ф. Х.
(прізвище, ініціали)
Керівник Безкоровайний В. В.
(прізвище, ініціали)
Рецензент _____
(прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Гребеннік І. В.
(прізвище, ініціали)

Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій

Кафедра Системотехніки

Освітньо-кваліфікаційний рівень Бакалавр

Напрямок підготовки 6.050201 – системна інженерія

(шифр і назва)

Спеціальність _____

(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«____» _____ 20 ____ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ (ПРОЕКТ)

студентові Ахмаду Фаді Хадєду

(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) Система Internet-продажу кондитерських виробів
затверджена наказом по університету від «20» травня 2016 р. № 570 Ст

2. Термін подання студентом роботи (проекту) 10.06.2016 р.

3. Вихідні дані до роботи (проекту)

Об'єкт – система Internet-продажу кондитерських виробів. Функція – автоматизація віддаленого продажу кондитерських виробів. Технічне забезпечення: IBM-сумісний персональний комп'ютер. Перелік використовуваних програмних засобів: базове програмне забезпечення – операційна система Ubuntu 14.04; спеціальне програмне забезпечення – php framework yii2, СУБД MySQL, WEB-сервер Apache2.

4. Зміст пояснювальної записки (перелік питань, що потрібно розробити)

Вступ. Аналіз сучасного стану проблеми автоматизації торгових технологій. Торгові технології як об'єкт автоматизації. Огляд і аналіз сучасних торгових технологій. Постановка задачі розробки системи. Розробка структури системи. Побудова функціональної моделі системи. Проектування структури бази даних. Опис і аналіз використовуваної технології. Шаблон проектування WEB-додатку. Вибір і обґрунтування програмних засобів для реалізації системи. Обґрунтування вибору СУБД. Обґрунтування вибору мови програмування. Обґрунтування вибору середовища розробки. Обґрунтування вибору WEB-сервера. Функціональна реалізація системи. Розробка бази даних. Опис процесу розробки додатку. Текст програми. Настанова користувача. Опис контрольного прикладу. Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів) Креслення або плакати на аркушах формату А4: потокові діаграми функціональної моделі системи; концептуальна схема шаблону MVC; схема бази даних системи; екранні форми програмного засобу.

6. Консультанти розділів роботи (проекту)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Безкоровайний В.В.		

7. Дата видачі завдання 14.03.2016 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи (проекту)	Термін виконання етапів проекту (роботи)	Примітка
1	Отримання завдання на атестаційну роботу	14.03.16	
2	Аналіз сучасного стану проблеми автоматизації торгових технологій	31.03.16	
3	Обґрунтування проектних рішень	15.04.16	
4	Розробка структури системи	29.04.16	
5	Розробка програмного забезпечення	11.05.16	
6	Проведення машинних експериментів	20.05.16	
7	Оформлення пояснювальної записки	27.05.16	
8	Оформлення графічної частини та презентації	30.06.16	
9	Подання роботи керівникові	03.06.16	
10	Подання роботи на рецензування	08.06.16	
11	Подання роботи до екзаменаційної комісії	10.06.16	

Студент _____
(підпис)

Ахмад Ф. Х.
(прізвище, ініціали)

Керівник роботи (проекту) _____
(підпис)

проф. Безкоровайний В. В.
(посада, прізвище, ініціали)

РЕФЕРАТ

Атестаційна робота: пояснювальна записка: 115 с., 34 рис., 4 додатки, 33 джерела, графічний матеріал – 7 аркушів.

Об'єкт розробки – система Internet-продажу кондитерських виробів.

Мета атестаційної роботи – автоматизація процесів пошуку товару, оформлення замовлень і ведення обліку на підприємстві, а також надання можливості виконувати дані процеси – за допомогою мережі Internet.

Методи розробки і апаратура – технології програмування за допомогою засобів yii2 framework. Розробка реалізована на IBM-сумісному персональному комп'ютері (тактова частота процесора – 2.27 ГГц, об'єм оперативної пам'яті – 4 Гб, обсяг жорсткого диска – 500 Гб).

Результати атестаційної роботи – система Internet-продажу кондитерських виробів, що реалізована як WEB-додаток, має користувальницький інтерфейс і дозволяє проводити: пошук товару, оформлення замовлень і ведення обліку на підприємстві. Взаємодія з системою здійснюється за допомогою мережі Internet. Система має підказки та покажчики, а також захист від несанкціонованого доступу.

Область застосування – компанії, що займаються реалізацією кондитерських виробів.

Практичне застосування розробленої системи дозволяє скоротити час реалізації і пошуку необхідного товару, а також спростити й прискорити процес формування замовлень, що сприяє підвищенню конкурентоспроможності компанії на ринку товарів і послуг.

КОНДИТЕРСЬКІ ВИРОБИ, АВТОМАТИЗАЦІЯ ПРОДАЖІВ, INTERNET-МАГАЗИН, WEB-ДОДАТОК, БАЗА ДАНИХ, ПРОГРАМНИЙ ПРОДУКТ.

THE ABSTRACT

Attestation work: explanatory note: 115 pages, 34 figures, 4 annexes, 33 sources, 7 pages of graphic material.

Productive formation is Internet-selling confectionery system.

The objective of the work is automation of searching of goods, ordering and accounting at the enterprise, as well as providing the opportunity to to make these processes remotely via the Internet.

A method of development is programming development technology yii2 framework. Development is implemented on IBM-compatible personal computer with 2.27 GHz of CPU clock frequency, 4 GB RAM and 500 GB ROM.

The results of the work are a system of Internet-sales of confectionery products that implemented as a WEB-application has a user interface and allows you to: searching of goods, ordering and accounting at the company. The interaction with the system is performed via Internet network. The system has the tips and pointers, as well as protection against unauthorized access.

The application area is a company that engages in the sale of confectionery.

Practical application of the developed system can reduce the time of implementation and find the necessary goods, as well as to simplify and express formation of orders process, thereby increasing the company's competitiveness.

CONFECTIONERY, SALES FORCE AUTOMATION, ONLINE STORE,
WEB-APPLICATION, DATABASE, SOFTWARE.

РЕФЕРАТ

Аттестационная работа: пояснительная записка: 115 с., 34 рис., 4 приложения, 33 источника, графический материал – 7 листов.

Объект разработки – система Internet-продажи кондитерских изделий.

Цель аттестационной работы – автоматизация процессов поиска товара, оформления заказов и ведения учета на предприятии, а также предоставление возможности производить данные процессы удаленно – с помощью сети Internet.

Методы разработки и аппаратура – технологии программирования с помощью средств yii2 framework. Разработка реализована на IBM-совместимом персональном компьютере (тактовая частота процессора – 2.27 ГГц, объем оперативной памяти – 4 Гб, объем жесткого диска – 500 Гб).

Результаты аттестационной работы – система Internet-продажи кондитерских изделий, реализованная как WEB-приложение, имеет пользовательский интерфейс и позволяет производить: поиск товара, оформление заказов и ведения учета на предприятии. Взаимодействие с системой осуществляется с помощью сети Internet. Система имеет подсказки и указатели, а также защиту от несанкционированного доступа.

Область применения – компании, занимающиеся реализацией кондитерских изделий.

Практическое применение разработанной системы позволяет сократить время реализации и поиска необходимого товара, а также упростить и ускорить процесс формирования заказов, что способствует повышению конкурентоспособности компании на рынке товаров и услуг.

КОНДИТЕРСКИЕ ИЗДЕЛИЯ, АВТОМАТИЗАЦИЯ ПРОДАЖ, INTERNET-МАГАЗИН, WEB-ПРИЛОЖЕНИЕ, БАЗА ДАННЫХ, ПРОГРАММНЫЙ ПРОДУКТ.

СОДЕРЖАНИЕ

Перечень условных обозначений, символов, единиц, сокращений и терминов	9
Введение	10
1 Анализ современного состояния проблемы автоматизации торговых технологий	12
1.1 Торговые технологии как объект автоматизации	12
1.2 Обзор и анализ современных торговых технологий	14
1.2.1 Обзор и анализ сайта кондитерской фабрики «Сладкоежка»	14
1.2.2 Обзор и анализ Internet-магазина компании «Комус»	15
1.2.3 Обзор и анализ Internet-супермаркета «Осьминог»	16
1.2.4 Обзор и анализ системы «Sales Force Automation»	16
1.2.5 Обзор и анализ системы автоматизации продаж «NetSuite»	18
1.2.6 Сравнение характеристик и возможностей аналогичных систем ..	20
1.3 Постановка задачи разработки системы	20
2 Разработка структуры системы Internet-продажи кондитерских изделий ...	22
2.1 Построение функциональной модели системы	22
2.2 Проектирование структуры базы данных	26
2.3 Описание и анализ используемой технологии	28
2.4 Шаблон проектирования WEB-приложения	30
3 Выбор и обоснование программных средств для реализации системы	36
3.1 Обоснование выбора СУБД	36
3.2 Обоснование выбора языка программирования для разрабатываемой системы	38
3.3 Обоснование выбора framework	42
3.4 Обоснование выбора WEB-сервера	43
4 Функциональная реализация системы	45
4.1 Описание процесса разработки приложения	45
4.1.1 Разработка базы данных системы	45

4.1.2 Разработка WEB-приложения	48
4.2 Анализ эффективности и результаты разработки системы	53
Выводы	64
Перечень ссылок	65
Приложение А Текст программы	68
Приложение Б Руководство пользователя	88
Приложение В Графические материалы аттестационной работы	106
Приложение Г Ведомость аттестационной работы	114

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ, ЕДИНИЦ,
СОКРАЩЕНИЙ И ТЕРМИНОВ

CRM	– Customer Relationship Management.
HTML	– Hypertext Markup Language.
IDEF0	– Методология функционального моделирования.
LAMP	– Linux & Apache & MySQL & PHP.
MVC	– Model View Controller.
PHP	– Personal Home Pages.
SFA	– Sales Force Automation.
SQL	– Structured Query Language.
WEB	– World Electronic Broadcast.
БД	– база данных.
СУБД	– система управления базами данных.

ВВЕДЕНИЕ

Деятельность любой фирмы или холдинга (группы компаний) оценивается качеством и количеством продаваемых товаров и услуг. Ведь основной источник прибыли и прогресса кроется в продажах. Продажи представляют собой сложный, многоступенчатый процесс, зависящий от множества факторов, имеющих множество показателей. Управлять продажей (отделом продаж) – значит управлять огромным объемом информации, большим количеством людей и каналов сбыта. Это трудоемкая деятельность, строящаяся на учете заказов, учете клиентов и учете продаж, а также умении анализировать полученные данные и принимать на их основе правильные управленческие решения.

Зачастую, руководитель (генеральный или коммерческий директор, директор по продажам или маркетингу) не может отследить всю «воронку продаж», выявить в ней проблемные места, понять причины падения или роста. А это значит, что он не может повлиять на увеличение объемов продаж, т. е. не может эффективно управлять этими процессами. Если компания растет и расширяется, ей следует задуматься об автоматизации процессов продаж, ведения учета и контроля товара, а также планирования грузоперевозок.

Автоматизация продаж может быть простой – реализованной как часть компьютерной программы, или же полной – реализованной как пакет программ, установленный в операционной системе компании [1]. И то, и другое предназначено для достижения одинаковых результатов и увеличения итогов компании. Если компания хочет достичь желаемого роста, ей необходимо автоматизировать продажи.

Следует также отметить, что характерной особенностью нашего времени является стремительный рост конкуренции между производителями продукции и, как следствие, номенклатурой товаров, предлагаемых потребителю. В таких условиях усложняется задача выбора товара по комплексному показателю «цена-качество». Одним из путей решения этой задачи является создание информационных технологий на основе поисковых систем, которые автоматизируют поиск, выбор и покупку товаров по установленным признакам. Это привело к появлению «предприятий новой экономики», сбыт продукции которых производится «электронными» покупателями с использованием сети Internet. Особое значение такая технология торговли имеет для фирм, реализующих продукцию

с установленным малым сроком реализации, примером которой являются кондитерские изделия.

Система Internet-продажи кондитерских изделий предназначена для решения двух основных задач: выбора товара (клиентом) и планирования процесса его доставки.

Целью разработки системы является автоматизация процессов поиска товара, оформления заказов и ведения учета на предприятии, а также предоставление возможности использовать систему удаленно – с помощью сети Internet.

Практическое применение предлагаемой системы Internet-продажи кондитерских изделий позволит:

- сократить время поиска необходимого товара, как клиентам, так и сотрудникам фирмы, а также его реализации;
- уменьшить затраты на доставку товаров покупателям за счет оптимизации маршрутов развозки;
- упростить и ускорить процесс формирования заказов сотрудниками.

За счет единой (общей) базы товаров и заказов, разрабатываемая система будет способствовать приданию структурной целостности организации.

1 АНАЛИЗ СОВРЕМЕННОГО СОСТОЯНИЯ ПРОБЛЕМЫ АВТОМАТИЗАЦИИ ТОРГОВЫХ ТЕХНОЛОГИЙ

1.1 Торговые технологии как объект автоматизации

Продажа, как сложный, многоступенчатый процесс, зависящий от множества факторов, имеющий множество показателей, заключается в выполнении менеджером определенных действий.

Процесс активных продаж товаров и услуг условно разделяют на этапы продаж и обычно выделяют пять классических этапов [2]:

- установление контакта;
- выявление потребностей;
- презентация товара;
- работа с возражениями;
- завершение сделки.

В отделе продаж информация постоянно поступает в таком количестве, что ни один среднестатистический специалист и ни один руководитель отдела продаж не в состоянии без использования специальных средств, записей или расчетов обработать эту информацию в нужном объеме и при этом учесть каждую деталь, ничего не упустив.

В процессе работы отдела продаж возникает немало проблем, справиться с которыми без качественного структурирования задач и их последующего выполнения невозможно. Основными проблемами могут быть:

- потеря клиентов;
- утечка важной информации;
- недостаточно четкое и рациональное распределение времени специалистов отдела продаж.

Для решения таких проблем следует применить системы автоматизации, позволяющие [2]:

- создать качественную базу сотрудников и клиентов;
- отслеживать стадии осуществления продаж, сложность работы организации в целом;
- фиксировать важную информацию;
- отслеживать каждую сделку;

- анализировать работу отдела продаж, выявлять его проблемные зоны для их дальнейшего устранения;
- оптимизировать рабочее время сотрудников отдела, существенно облегчить им работу с клиентами;
- облегчить процесс руководства отделом продаж, сделать его максимально эффективным.

Введение систем автоматизации продаж. Автоматизация продаж может быть нацелена на отдельные департаменты или области компании. Компании могут использовать автоматизированную систему продаж в самом отделе продаж или в отделе приема и обработки заказов.

Обычно в маленьких и средних компаниях автоматизация продаж применяется как основной способ, при помощи которого можно увеличить товарооборот.

Использование программного обеспечения автоматизации продаж в отделе сбыта дает возможность каждому продавцу оставаться на вершине своей повседневной деятельности, такой как коммерческие звонки и целевые потребители.

Если правильно использовать программу обеспечения автоматизации продаж, то она будет поддерживать продавца на связи с его потребителями и клиентами на протяжении всего процесса продаж, также она является отличным средством для развития связей [2]. С другой стороны, у продавца (или других сотрудников компании) пропадет необходимость совершения постоянных звонков друг другу, например, для произведения заказов или ведения и контроля учета. Также, пропадает необходимость ведения большинства записей и прочей бумажной работы.

Прежде всего, для использования автоматизированного способа продаж, необходимо определить желаемые результаты работы системы – это очень важный анализ для конечного решения разработки любой автоматизированной системы продаж. Например, если цель компании – управление связями с клиентами, то будет достаточно разработать основное программное обеспечение автоматизации продаж для управления связями [1].

С другой стороны, если компания захочет подключить дополнительные отделы, такие как: бухгалтерия, отдел закупок, маркетинг и др., с целью получения максимальной пользы от автоматизации продаж, впоследствии завер-

шенная система инсталляции должна быть тщательно продумана [1]. Такой тип автоматизации продаж придаст законченный вид целого процесса операций компании. Однако такой способ достаточно трудоемкий, требует много времени на разработку, а также внедрение и обслуживание системы.

Следует также упомянуть о том, что для достижения большей эффективности и результатов компании, целесообразно включить в систему возможность Internet-продажи, что поспособствует автоматизации поиска, выбора и покупке товаров. Это позволит клиенту совершать заказы (покупки) в любом месте, а также минуя очереди.

Еще одним немаловажным этапом, позволяющим увеличить доход и улучшить работу фирмы, является рациональное планирование маршрутов грузоперевозки и доставки товара. Нахождение оптимального маршрута в заданных условиях позволит уменьшить затраты ресурсов и времени на грузоперевозку товара.

1.2 Обзор и анализ современных торговых технологий

1.2.1 Обзор и анализ сайта кондитерской фабрики «Сладкоежка»

Компания «Сладкоежка» с 2006 года выпускает широкий ассортимент кондитерской продукции. Фабрика находится в станице Тбилисской Краснодарского края и работает по всей России, а также за ее пределами. Компания нацелена на сотрудничество с крупными торговыми организациями и оптовыми базами [3].

Для расширения круга клиентов, компания реализовала WEB-сайт, содержащий информацию об изготавливаемой продукции, вакансиях на фирме, а также контактные данные.

WEB-приложение предоставляет клиентам достаточно полное представление о самой компании, а также об изготавливаемой ею продукции.

Следует отметить, что WEB-приложение реализует также обратную связь: клиент может обратиться с вопросом к производителю через Internet.

Однако WEB-сайт имеет и ряд недостатков:

- существенным недостатком приложения является невозможность производить заказы с самого WEB-сайта, т. е. компания, вероятней всего,

предоставляет продукцию лишь торговым организациям или оптовым базам, сотрудничающим с ней – не позволяет клиентам самостоятельно производить заказы через Internet;

- компания, как следует из описания, занимается реализацией только собственно изготовленной продукции, а разрабатываемая система, в свою очередь, нацелена на продажу продукции, поставляемой от разных производителей; это усложняет процесс исполнения заказа (доставки товара) и ведения учета.

Однако стоит заметить, что WEB-сайт компании «Сладкоежка» имеет достаточно удобный пользовательский интерфейс, что способствует привлечению большего количества клиентов, а также позволяющего им быстро и легко пользоваться данным сайтом.

1.2.2 Обзор и анализ Internet-магазина компании «Комус»

Компания «Комус» занимает лидирующие позиции на российском рынке товаров для офиса и компьютерной техники и является одним из крупнейших поставщиков бумаги, картона и пластиковой упаковки в России. Компания является эксклюзивным дистрибьютором торговых марок «Kores», «Sax», «Bantex» и официальным дистрибьютором 110 ведущих производителей 15 стран Америки, Европы и Азии [4].

Следует взять на рассмотрение 4 весьма удобные формата покупок, предоставляемые клиентам компанией «Комус» [4]:

- заказы по телефону контакт-центра «Комус», что является предпочтительным для клиентов, которым необходима доставка товара, индивидуальный подход, помощь оператора, а также экономия времени;
- Internet-магазин, что является удобным для клиентов, предпочитающим удобный, быстрый, автоматический сервис, возможность заказа 24 часа в сутки и много информации о товарах;
- сеть розничных магазинов «Комус», если клиенту необходим товар именно «сегодня» или он предпочитает лично выбрать товар «с полки»;
- сервис самовывоза, позволяющий клиентам самостоятельно, в удобное для них время, а также с экономией времени на доставке, получить свой заказ.

1.2.3 Обзор и анализ Internet-супермаркета «Осьминог»

Internet-магазин (Internet-супермаркет) «Осьминог» предоставляет клиентам следующее [5]:

- более 15 000 товаров в 3 десятках категорий;
- доставка товаров по городу (Харьков); дата и время доставки определяются совместно клиентом и оператором магазина во время подтверждения заказа;
- отправка заказов в другие населенные пункты почтово-курьерской службой.

Следует отметить, что компания «Осьминог» предоставляет лишь услуги Internet-магазина, однако они могут быть полезны при разработке клиентской части системы Internet-продажи кондитерских изделий.

1.2.4 Обзор и анализ системы «Sales Force Automation»

«Sales Force Automation» (SFA) – система автоматизации продаж, осуществляющая автоматическую регистрацию всех этапов продаж компании. SFA включает в себя систему отслеживания контактов с клиентами. SFA легко интегрируется с Customer Relationship Management (CRM), и может служить основой для этой системы. Наиболее развитые SFA системы предоставляют клиенту возможность смоделировать продукт, удовлетворяющий его потребности On-line. Это стало популярным в автомобильной промышленности. Покупатель может, воспользовавшись подобной функцией, выбрать наиболее подходящий цвет и интерьер автомобиля [6].

Система управления продажами «Sales Force Management System» (SFMS) предоставляет информацию обо всех клиентах в виде таблиц, диаграмм, графиков. Менеджеру не придется тратить время на их составление. При работе с SFMS менеджер владеет полной информацией о заказах, что дает ему возможность своевременно реагировать на изменение ситуации, давать необходимые рекомендации подчиненным, проверять запас продукции и корректировать систему скидок.

Функции системы следующие [6]:

- управление контактами – хранение и обновление информации о

клиентах, история контактов, информация о точках циклических продаж и нуждаемости клиента в продукции компании;

- управление деятельностью – составление и ведение делового дневника торгового представителя;
- управление связью – передача данных посредством модема или мобильного телефона, а также обеспечение сохранности и репликации информации;
- прогнозирование – планирование продаж, а также поиск и анализ прогнозов различных исследовательских организаций и маркетинговых исследований собственных подразделений;
- управление возможностями – внедрение различных схем, привлекающих потенциальных клиентов;
- управление заказами – оперативный доступ к информации о наличии товаров на складе, а также размещение заказа на изготовление и доставку требуемого продукта;
- управление документацией – разработка стандартов, отчетов, информационных и рекламных материалов.
- анализ продаж – использование статистических данных для улучшения стратегии продаж.
- управление информацией о продукции – хранение и обновление данных о характеристиках товаров, ценовом диапазоне, наличии альтернативных продуктов.
- энциклопедия маркетинга – обновление информации о продуктах, новинках, ценах, рекламных акциях, системах скидок, а также результаты исследований и информация о конкурентах.

Аппаратные платформы. Ноутбуки на базе операционной системы Windows или планшетные компьютеры, снабженные сенсорным дисплеем, удобны для торговых представителей, которым необходимо делать презентации или знакомить клиента с некой визуальной информацией, а также, если менеджер работает в условиях офиса и не совершает многочисленных выездов. КПК на базе одной из мобильных операционных систем, таких как «Windows Mobile», «Symbian», «Palm OS», незаменимы для сотрудника, работающего на выезде [6].

Удобно, если КПК будет оснащен GSM/GPRS, GPS модулями, фотокаме-

рой. Это даст возможность собрать и систематизировать данные, передать их практически из любого места, сфотографировать требуемые объекты. А посредством GPS осуществляется контроль над передвижением торгового представителя. Преимущества данной аппаратной платформы очевидны: портативность, функциональность, эргономичность, сравнительно низкая стоимость, длительность работы без подзарядки [6].

В целом, система «Sales Force Automation» предоставляет множество удобных функций для работы с клиентом, документацией и товаром, что является незаменимым в любой системы автоматизации продаж. Однако недостатками системы можно выделить следующее:

- из всех возможностей, предоставляемой данной системой множество, вероятней всего, не будут использоваться, что делает систему более сложной для работы;
- необходимость установки программного обеспечения на каждое из устройств сотрудников;
- не предоставляет возможности клиенту производить заказы самостоятельно.

1.2.5 Обзор и анализ системы автоматизации продаж «NetSuite»

Система «NetSuite» предоставляет возможность полностью управлять процессом продаж, включая поступление заказов на закупку, послепродажное обслуживание и другие мероприятия, а также возможности организации дополнительных (пакетных и повторных) продаж.

В отличие от конкурирующих продуктов, система автоматизации продаж «NetSuite» обеспечивает обзор полной картины истории покупок клиента, что, в свою очередь, позволяет предлагать клиентам лучшее обслуживание, повышать уровень удовлетворенности потребителей, а соответственно, и их лояльность, а в итоге – увеличивать прибыль компании [7].

Показатель будущих доходов. На панели инструментов системы автоматизации продаж можно наблюдать все контакты (всех претендентов, кандидатов), сделки – в соответствии с заданными параметрами и территориями. Можно также проследить несколько контактов в пределах одной компании по отдельности, проникая в самую суть каждого контакта, каждого разговора и вза-

имодействия для определения перспективности каждого потенциального клиента. Также на панели инструментов возможно отображение таких возможностей, как целевые показатели объёма продаж, заказы, закрытые продажи; возможности совершения сделки, планируемая сумма сделки и средняя сумма для каждой транзакции [7].

Управление заказами. С помощью одного клика мыши торговый представитель компании может превратить целевые показатели объёма продаж в заказы. Следовательно, торговый персонал сможет тратить больше времени на продажу продуктов и оказание услуг, а не на перебирание бумаг.

Именно поэтому системы автоматизации продаж и управления взаимоотношениями с клиентами (CRM) «NetSuite» устраняют необходимость излишнего ввода данных и сводят к минимуму вероятность допущения ошибок при вводе информации [7].

Стандартное и расширенное прогнозирование. Приложения «NetSuite» и «NetCRM» имеют опции автоматизации и планирования продаж.

Стандартное прогнозирование строится на основе средневзвешенного измерения цен и заказов. При расширенном прогнозировании возможно использование и более сложных инструментов – таких, как категории прогноза и диапазон сделок [7].

Дополнительные продажи. Система автоматизации процесса продаж позволяет торговому персоналу анализировать предыдущие покупки клиента и автоматически рекомендует продукты для дополнительных (пакетных, повторных) продаж. Работа системы в реальном времени позволяет вовремя организовывать высокоэффективные маркетинговые кампании по продвижению продуктов, в том числе и дополнительных продаж [7].

Работа с клиентами. Обладание полной информацией о клиенте, включая историю продаж, обслуживания, выставления счетов, поставок и т. д., обеспечивают высокую вероятность установления более качественных и долгосрочных отношений [7].

Таким образом, можно сказать, что система «NetSuite» предоставляет достаточно много возможностей для работы с клиентами и заказами, однако к этим возможностям для системы Internet-продажи кондитерских изделий стоило бы добавить больше информации о товарах, а также предоставить возможность клиенту самостоятельно производить заказы с помощью Internet.

1.2.6 Сравнение характеристик и возможностей аналогичных систем

Проведя анализ систем-аналогов, можно отметить, что такие системы как «Sales Force Automation» и «NetSuite» предоставляет множество удобных функций для работы с клиентом, документацией и товаром и заказами, но не предоставляют возможности клиенту самостоятельно просматривать каталоги товаров, а также производить заказы с помощью Internet. Однако это реализовано у систем компаний «Комус» и «Осьминог», а также кондитерской фабрики «Сладкоежка».

Для системы Internet-продажи кондитерских изделий можно взять за основу возможности систем автоматизации продаж «Sales Force Automation» и «NetSuite», такие как:

- управление контактами;
- управление заказами;
- анализ продаж;
- управление информацией о продукции;
- дополнительные продажи;
- работа с клиентами.

Включив в систему возможность создания заказов клиентами через Internet, как это предоставляют компании «Комус» и «Осьминог», а также предоставить удобный пользовательский интерфейс для клиентов и достаточно полную информацию о товарах, как представлено на сайте кондитерской фабрики «Сладкоежка», можно преступить к проектированию структуры системы.

1.3 Постановка задачи разработки системы

Для развивающейся компании, занимающейся продажей кондитерских изделий, необходимо произвести автоматизацию процессов поиска товаров, формирования заказов, оформления продаж, сбора и обработки статистики продаж.

Для этого необходимо разработать приложение, удовлетворяющее следующим условиям:

- включать в себя базу данных содержащую следующую информацию:

- 1) о товарах, а также наличии их в магазинах или складах;
- 2) о сотрудниках (контактные данные, должности и т. п.);
- 3) о клиентах (контактные данные, совершенные заказы);
- 4) о поставщиках товара;
- 5) о заказах;

— позволять лишь сотрудникам фирмы просматривать полный список товара в базе данных (для ведения учета и выполнения заказов);

— предоставлять возможность формирования заказов внутри сети (самими сотрудниками организации), а также клиентам (покупателям) с использованием сети Internet;

— предоставлять возможность для администратора (руководителя) редактировать список товаров, доступных клиенту для осуществления заказа с помощью Internet;

— группировать заказы по дате и, при подтверждении администратором, отправлять в отдел грузоперевозок для выполнения доставки заказов.

2 РАЗРАБОТКА СТРУКТУРЫ СИСТЕМЫ INTERNET-ПРОДАЖ КОНДИТЕРСКИХ ИЗДЕЛИЙ

2.1 Построение функциональной модели системы

Для более точного представления структуры системы, построена функциональная модель (стандарт IDEF0), отображающая структуру и функции системы [8].

В IDEF0, модель – искусственный объект, представляющий собой отображение (образ) системы и ее компонентов; система представляет собой совокупность взаимосвязанных частей, выполняющих некоторую полезную работу [8].

Методология IDEF0 основана на следующих основных концепциях:

- блочное моделирование и его графическое представление;
- строгость и формализм.
- итеративное моделирование.

Функциональные модели являются структурным изображением функций системы или среды, а также информации и объектов, связывающих эти функции.

На рисунке 2.1 представлено определение единственной работы контекстной диаграммы как «Internet-продажа кондитерских изделий». Далее следует определить входные и выходные данные, а также механизмы и управление.

Из диаграммы следует, что процесс Internet-продажи кондитерских изделий заключается в том, что поступившие в систему денежные средства и заявки клиентов при помощи механизмов (системы автоматизации и подразделениями предприятия) и управления (документации) позволяют получить готовую продукцию на выходе системы.

Далее необходимо провести декомпозицию контекстной диаграммы, описав последовательность «Internet-продажи кондитерских изделий». Это будут следующие блоки (рис. 2.2):

- подача заявки;

- перевозка продукции;
- реализация продукции.

Декомпозиция «Подачи заявки» (рис. 2.3): первым этапом при подаче заявки является «получение и анализ заявки», которая впоследствии должна быть подтверждена; на основе подтвержденной заявки и будет формироваться заказ.

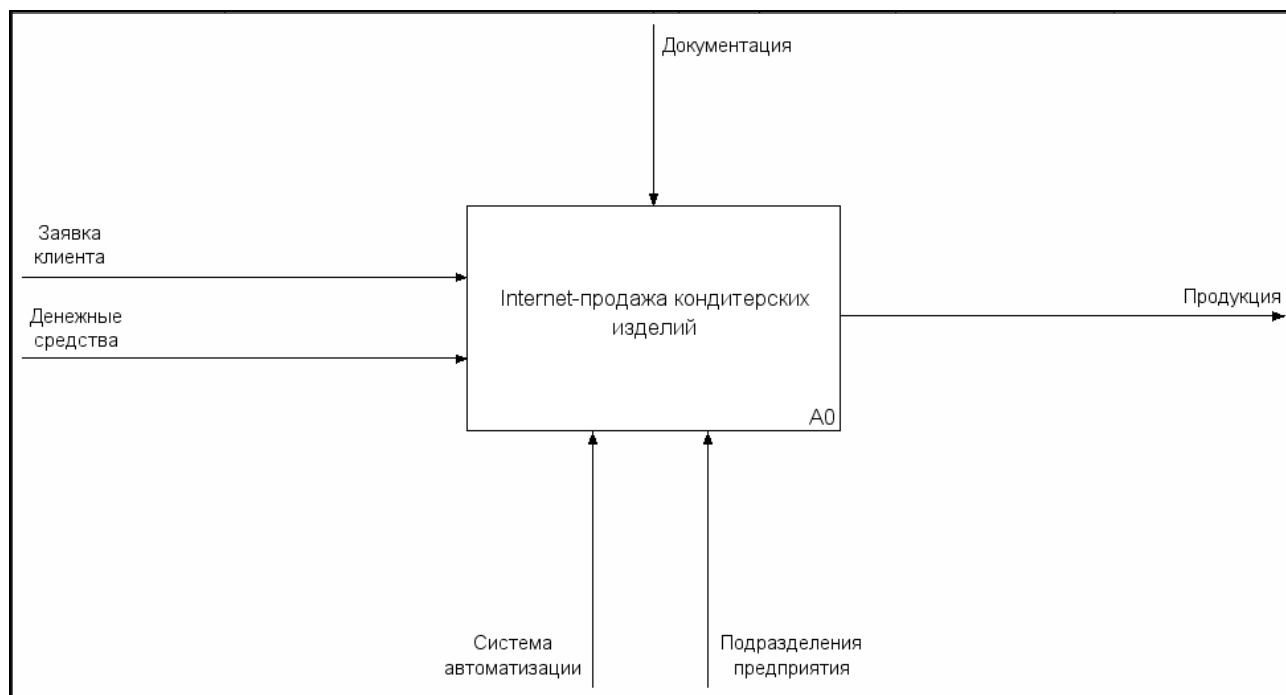


Рисунок 2.1 – Контекстная диаграмма системы

Затем, производится выполнение сформированного заказа – перевозка продукции, заключающаяся в следующих работах:

- заказ продукции у изготовителя (поставщика);
- формирование маршрута;
- перевозка продукции;
- регистрация продукции на складе.

Следовательно, можно произвести декомпозицию «перевозки продукции» (рис. 2.4).

Получив и зарегистрировав продукт на складе, под «управлением» указания на отгрузку, производится реализация продукции, которая, в свою очередь, состоит из (рис. 2.5):

- отгрузки со склада;
- регистрации на пункте выдачи;
- передачи продукции клиенту.

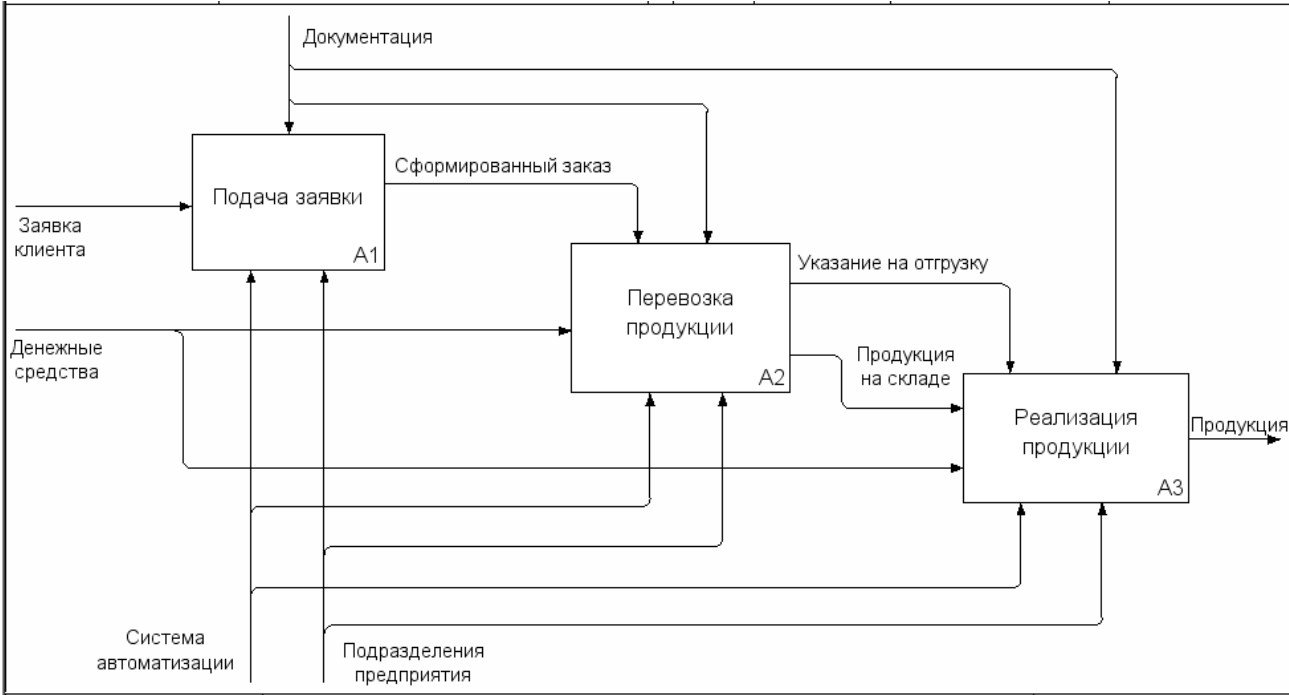


Рисунок 2.2 – Декомпозиция «Internet-продажи кондитерских изделий»



Рисунок 2.3 – Декомпозиция «Поддача заявки»

Построенная функциональная модель позволяет увидеть, какие работы подлежат автоматизации; а также те работы, которые могут быть исполнены только человеком (сотрудником компании).



Рисунок 2.4 – Декомпозиция «Перевозки продукции»

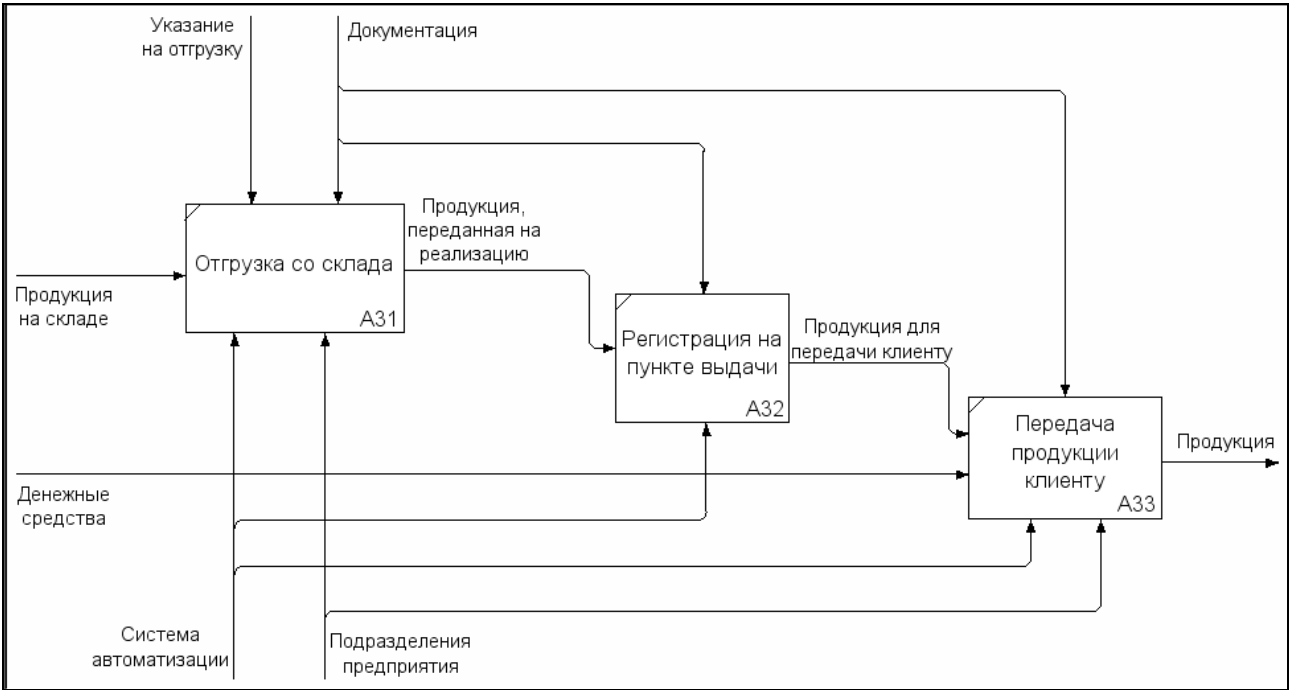


Рисунок 2.5 – Декомпозиция «Реализации продукции»

2.2 Проектирование структуры базы данных

Руководствуясь построенной функциональной моделью, а также постановкой задачи, для получения представления о данных, которые будут использоваться в системе Internet-продажи кондитерских изделий, стоит обозначить приблизительные задачи, которые система должна выполнять (предоставлять). Задачи ее следующие:

- добавление, редактирование и удаление списка товаров (администратором);
- отображение товаров;
- добавление, редактирование, удаление и отображение списка поставщиков (администратором);
- классификация, фильтрация и поиск товаров;
- регистрация сотрудников и клиентов;
- оформление заказов сотрудниками фирмы;
- отображение заказов (администратору и сотрудникам);
- редактирование и подтверждение заказов администратором;
- добавление товаров в корзину покупок;
- отображение и редактирование содержимого корзины покупок;
- оформление заказов посетителями;
- и т. д.

Проведя анализ списка задач, можно вывести сущности, которые имеют важные роли в системе:

- товары;
- категории товаров
- пользователи
- категории пользователей (администратор, сотрудники, клиенты и др.);
- поставщики;
- заказы

Определившись с сущностями системы, следует установить (определить) связи между ними, что позволит получить представление о структуре базы данных (рис. 2.6).

Между «заказом» и «товаром» существует связь «многие-ко-многим», так

как каждый заказ содержит 1 или более товаров, а каждый товар может быть связан с любым количеством заказов. Связь «многие-ко-многим» в реляционных базах данных создается с помощью трех таблиц: две таблицы – источники данных («order» – заказ и «product» – товары) и одна – соединительная («orderproduct»). В результате, заказы и товары имеют связь «один-ко-многим» с соединительной таблицей, а вместе они образуют связь «многие-ко-многим» между «заказами» и «товарами».

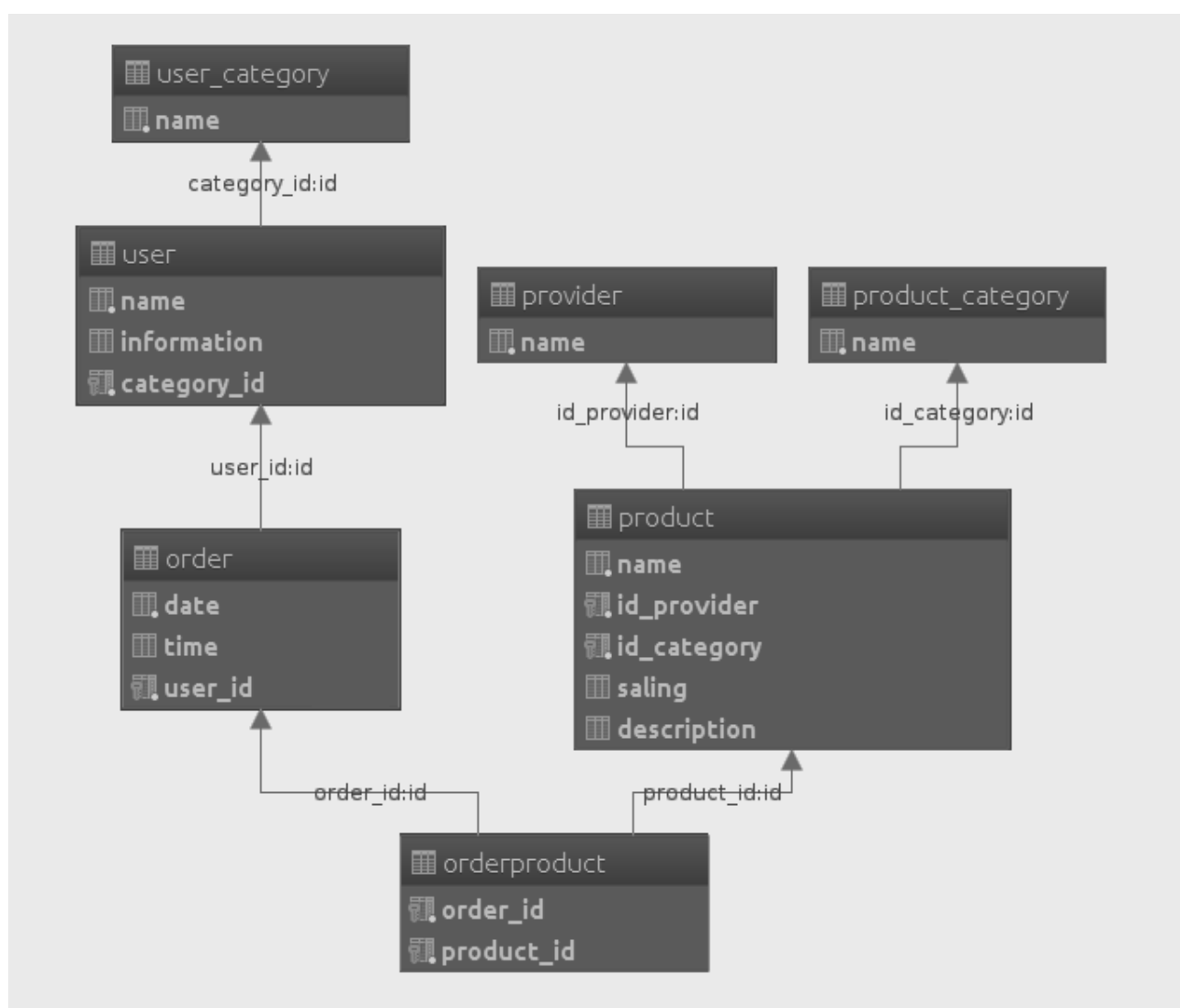


Рисунок 2.6 – Структура базы данных системы

«Пользователи» и «заказы» имеют связь «один-ко-многим». Каждая запись о пользователе может быть связана с множеством записей о заказах («заказами») и наоборот: каждая запись о заказе (конкретный заказ) может быть

связана только с одной записью о пользователе. Другими словами, пользователь может выполнить множество заказов, в то время как заказ может быть выполнен лишь одним пользователем.

Один товар может поставляться лишь одним поставщиком и может относиться лишь к одной категории. В свою очередь, поставщик может поставлять множество товаров, а также одна категория может включать множество товаров. Из этого следует, что между «поставщиками» и «товарами» связь «один-ко-многим». Такая же связь между «категориями» и «товарами».

Руководствуясь полученной структурой базы данных, можно приступить к выбору паттерна (шаблона) проектирования, а также выбору программных средств, а впоследствии – к реализации базы данных и самой системы Internet-продажи кондитерских изделий.

2.3 Описание и анализ используемой технологии

Исходя из того, что система Internet-продажи кондитерских изделий должна решать две основные задачи: выбор товара и планирование процесса его доставки, а также, произведя оценку и анализ существующих систем автоматизации и Internet-продажи товаров, следует разработать систему как программу, которая может быть доступна практически на любом современном электронном устройстве – WEB-приложение.

Известно, что WEB-приложение имеет ряд преимуществ [9]:

- не требуют установки на компьютер сотрудников объемного программного обеспечения, для полноценной работы нужен только браузер и доступ в Internet;
- WEB-разработки не требуют специальной настройки и администрирования, их администраторами являются разработчики;
- для работы WEB-приложений требуется минимальная аппаратная платформа;
- обновление WEB-приложений происходит автоматически;
- WEB-приложения обеспечивают высокую мобильность – возможность ими пользоваться везде, где есть доступ в Internet.

Таким образом, любой сотрудник или клиент, имеющий устройство с браузером и доступом к сети, сможет воспользоваться данной системой.

Также, система должна быть способна удовлетворять основную цель разработки – автоматизация процессов поиска товаров, формирования заказов, продаж, а в дальнейшем – формирование оптимальных маршрутов грузоперевозки и доставки товара.

Первая задача решается с помощью "клиент-серверного" WEB-приложения [9]. При этом WEB-приложение находится и выполняется на сервере, клиент при этом получает только результаты работы.

Работа приложения основывается на получении запросов на товар от пользователей (клиентов), их обработке и выдачи результатов относительно выбранных товаров. Передача запросов и результатов их обработки производится через Internet. Отображение результатов запросов, а также прием данных от клиентов и их передача на сервер реализуется браузером (рис. 2.7). Браузер отображает данные, полученные из Internet, в виде страницы, описанной на языке hypertext markup language (HTML) [9].

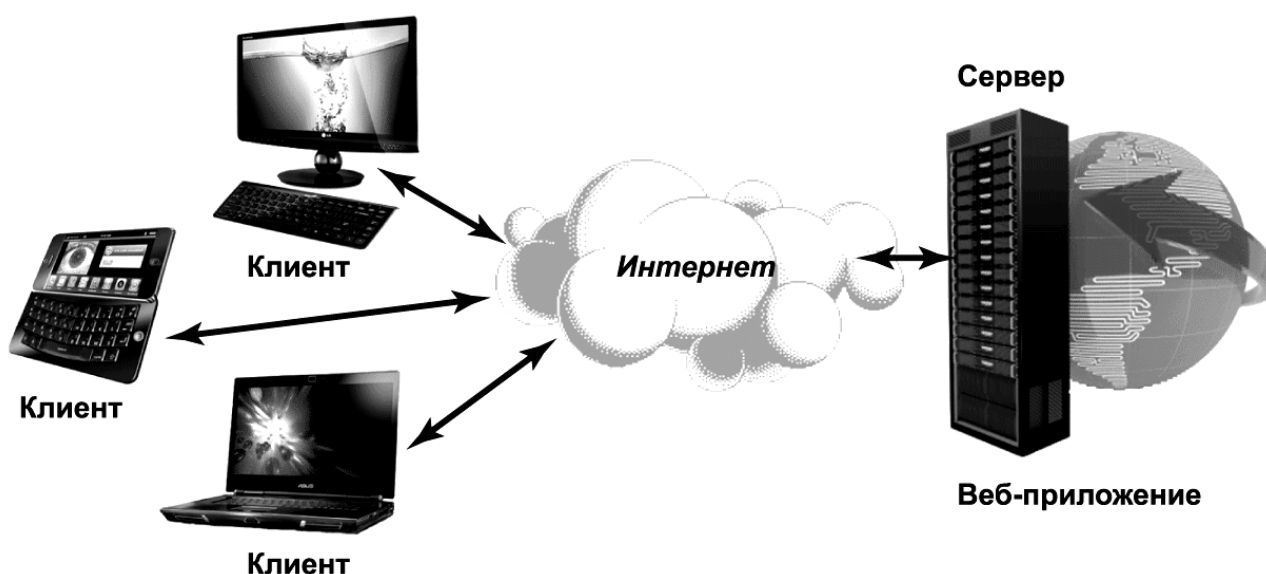


Рисунок 2.7 – Архитектура WEB-приложения

На стороне сервера WEB -приложение выполняется специальным программным обеспечением (WEB -сервером), который принимает запросы клиентов, обрабатывает их, формирует ответ в виде страницы, описанной на языке HTML, и передает его клиенту.

В разрабатываемой системе, клиенты с помощью WEB -приложения будут производить заказ товаров и, при желании, производить оплату покупок.

Информация о поступивших заказах сохраняется в базе данных. На ее основе производится планирование маршрутов доставки товара [10, 11].

Суть задачи планирования маршрутов состоит в следующем:

- из базы данных системы поступает информация:
 - 1) о множестве выполненных покупок, требующих доставки товара (место доставки, объем товара, срок доставки);
 - 2) о множестве имеющихся транспортных средств, их функциональные и стоимостные характеристики;
- необходимо определить:
 - 1) требуемое количество транспортных средств;
 - 2) их типы;
 - 3) множества обслуживаемых ими покупателей;
 - 4) а также технологию (очередность) развозки товара.
- при этом желательными являются минимизация затрат на доставку товаров и требование выполнения заданных ограничений на время их доставки [10].

В системе могут наблюдаться существенные колебания спроса на продукцию, что требует решения задач, значительно различающихся по размерности. С учетом того, для решения задачи предлагается использовать интерактивную технологию, включающую множество методов, различающихся по точности и временной сложности.

2.4 Шаблон проектирования WEB-приложения

Архитектуры WEB-приложений могут быть самыми различными. Не секрет, что на современном рынке существует большое количество технологий и программных продуктов, связанных с Internet. Когда Internet попал в поле зрения крупных корпораций, многие устремились занять свое место в этой области. Все, что можно было найти "на полке", либо модифицировалось, либо просто продавалось для использования в WEB и Internet. В результате появился широкий спектр продуктов и технологий, которые могут быть частью архитектуры WEB-приложений [12].

WEB-приложения развивались на основе совершенствования и усложнения функций WEB-узлов. По сути, WEB-приложение расширяет функции

WEB-узла, предоставляя его клиентам возможность использования бизнес-логики и, следовательно, изменения данных на сервере. Это определение WEB-приложения указывает, что в нем, как минимум, имеется три важных архитектурных компонента [12]:

- клиентский browser;
- WEB-сервер;
- сервер приложений.

Зачастую WEB-приложение использует также сервер баз данных.

Для реализации системы Internet-продажи кондитерских изделий целесообразней было бы использовать один из паттернов WEB-представления. Например [12]:

- Transform View (Преобразователь);
- Template View (Шаблонизатор);
- Application Controller (Контроллер приложения);
- Two Step View (Двухшаговая шаблонизация);
- Page Controller (Контроллер страницы);
- Front Controller (Контроллер входа / Единая точка входа);
- Model View Controller (MVC, Модель-Вид-Контроллер).

Transform View преобразует записи в HTML по одной.

Template View подразумевает заполнение HTML-шаблонов информацией при помощи маркеров, указанных в шаблоне.

Application Controller – единая точка управления отображением и выполнением приложения.

Two Step View преобразует данные в HTML в два шага: сначала формирует логическую структуру, а позже – заполняет её отформатированными данными.

Page Controller – объект, обрабатывающий запрос к отдельной странице или действию. В результате, контроллер страницы (Page Controller) – паттерн, в котором один контроллер отвечает за отображение одной логической страницы. Это может быть как отдельная страница, хранящаяся на WEB-сервере, так и отдельный объект, который отвечает за страницу.

Front Controller – один контроллер обрабатывает все запросы к WEB-сайту. Паттерн Front Controller объединяет всю обработку запросов, пропуская запросы через единственный объект-обработчик. Этот объект содержит общую

логику поведения, которая может быть изменена в реальном времени при помощи декораторов. После обработки запроса контроллер обращается к конкретному объекту для отработки конкретного поведения.

Model View Controller разделяет работу WEB-приложения на три отдельные функциональные роли:

- модель данных (model);
- пользовательский интерфейс (view);
- управляющую логику (controller).

Таким образом, изменения, вносимые в один из компонентов, оказывают минимально возможное воздействие на другие компоненты. В данном паттерне модель не зависит от представления или управляющей логики, что делает возможным проектирование модели как независимого компонента и, например, создавать несколько представлений для одной модели. Впервые этот шаблон был применён во framework, разрабатываемом для языка Smalltalk в конце 1970-х годов. С этого момента он играет основополагающую роль в большинстве framework с пользовательским интерфейсом. Он в корне изменил взгляд на проектирование приложений.

Создание приложений, генерирующих HTML, зачастую, гораздо более сложно, чем кажется. Несмотря на то, что современные языки программирования стали лучше справляться с обработкой текста, создание и конкатенация строк всё ещё представляется проблемой. Если надо вывести немного информации – это не так страшно, но если надо сгенерировать целую HTML-страницу – появляется много работы с текстом. Также, возникает проблема в случае с динамическими страницами, которые, например, берут данные из базы данных (БД) и наполняют ими HTML. Страницы выглядят по-разному каждый раз, и использование обычного HTML-редактора не подходит [13].

Если WEB-приложение состоит из множества страниц, необходим единый вид и единая структура сайта. Если каждая страница выглядит по своему, получится сайт, который будет непонятным для пользователя. Также возможна ситуация, когда нужно сделать глобальные изменения на всём сайте (например, поменять заголовки), но при использовании Template View или Transform View возникают трудности, потому что код представления дублируется от страницы к странице и надо исправлять его во всех файлах.

Когда выполняются запросы к БД, приложение получает данные, однако

этого не достаточно, чтобы отобразить «нормальную» WEB-страницу. Задача вида (view) в паттерне MVC – формировать данные в WEB-страницу.

Исходя из вышесказанного, принято решение использования паттерна MVC для реализации системы Internet-продажи кондитерских изделий.

Шаблон MVC описывает простой способ построения структуры приложения, целью которого является отделение бизнес-логики от пользовательского интерфейса. В результате, приложение легче масштабируется, тестируется, сопровождается и, конечно же, реализуется.

Концептуальная схема шаблона MVC представлена на рис. 2.8 [13].

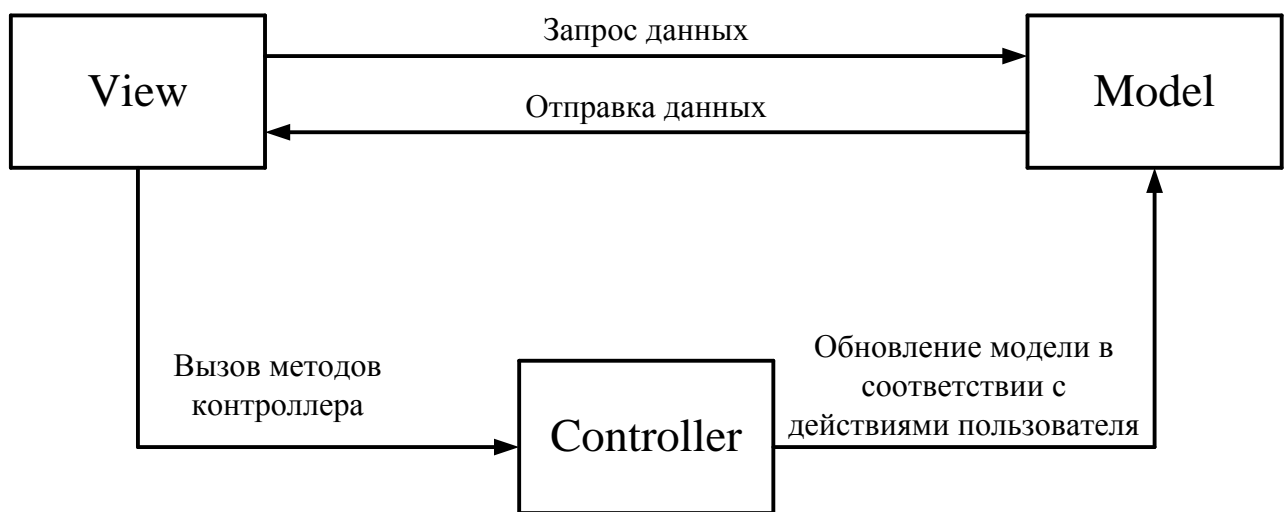


Рисунок 2.8 – Концептуальная схема шаблона MVC

В архитектуре MVC модель предоставляет данные и правила бизнес-логики, представление отвечает за пользовательский интерфейс, а контроллер обеспечивает взаимодействие между моделью и представлением.

Типичную последовательность работы MVC-приложения можно описать следующим образом [13]:

- при входе пользователя на WEB-ресурс, скрипт инициализации создает экземпляр приложения и запускает его на выполнение;
- при этом отображается вид, например, главной страницы сайта;
- приложение получает запрос от пользователя и определяет запрошенные контроллер и действие; в случае главной страницы, выполняется действие по умолчанию – index;
- приложение создает экземпляр контроллера и запускает метод дей-

ствия, в котором, к примеру, содержатся вызовы модели, считывающие информацию из базы данных;

– после этого, действие формирует представление с данными, полученными из модели, и выводит результат пользователю.

Модель. Содержит бизнес-логику приложения и включает методы выборки (это могут быть методы ORM), обработки (например, правила валидации) и предоставления конкретных данных, что зачастую включает в себя много кода, что вполне нормально [13].

Модель не должна напрямую взаимодействовать с пользователем. Все переменные, относящиеся к запросу пользователя должны обрабатываться в контроллере.

Модель не должна генерировать HTML или другой код отображения, который может изменяться в зависимости от нужд пользователя. Такой код должен обрабатываться в видах.

Одна и та же модель, например: модель аутентификации пользователей может использоваться как в пользовательской, так и в административной части приложения. В таком случае, можно вынести общий код в отдельный класс и наследоваться от него, определяя в наследниках специфичные для подприложений методы.

Вид. Используется для задания внешнего отображения данных, полученных из контроллера и модели.

Виды содержат HTML-разметку и небольшие вставки PHP-кода для обхода, форматирования и отображения данных [13].

Не должны напрямую обращаться к базе данных. Этим должны заниматься модели.

Не должны работать с данными, полученными из запроса пользователя. Эту задачу должен выполнять контроллер.

Может напрямую обращаться к свойствам и методам контроллера или моделей, для получения готовых к выводу данных.

Виды обычно разделяют на общий шаблон, содержащий разметку, общую для всех страниц (например, шапку и подвал) и части шаблона, которые используют для отображения данных выводимых из модели или отображения форм ввода данных.

Контроллер. Связующее звено, соединяющее модели, виды и другие

компоненты в рабочее приложение. Контроллер отвечает за обработку запросов пользователя. Контроллер не должен содержать SQL-запросов. Их лучше держать в моделях. Контроллер не должен содержать HTML и другой разметки. Её стоит выносить в виды [13].

В хорошо спроектированном MVC-приложении контроллеры обычно очень «тонкие» и содержат только несколько десятков строк кода. Логика контроллера довольно типична и большая ее часть выносится в базовые классы.

Модели, наоборот, содержат большую часть кода, связанную с обработкой данных, т. е. структура данных и бизнес-логика, содержащаяся в них, обычно довольно специфична для конкретного приложения.

3 ВЫБОР И ОБОСНОВАНИЕ ПРОГРАММНЫХ СРЕДСТВ ДЛЯ РЕАЛИЗАЦИИ СИСТЕМЫ

3.1 Обоснование выбора СУБД

Базы данных – это специально разработанное хранилище для различных типов данных. Каждая база данных, имеет определённую модель (реляционная, объектно-ориентированная и др.), которая обеспечивает удобный доступ к данным. Системы управления базами данных (СУБД) – специальные приложения (или библиотеки) для управления базами данных различных размеров и форм [14].

Взаимодействие с базой данных происходит при помощи СУБД, которая расшифровывает запросы и производит операции с информацией в базе данных. Правильнее рассматривать запросы к СУБД с точки зрения их взаимодействия с базой данных из WEB-приложения.

Существуют следующие разновидности СУБД (баз данных) [15]:

- иерархические;
- реляционные;
- объектно-ориентированные;
- гибридные.

Базы данных с иерархической моделью основаны на древовидной структуре хранения информации, состоящей из объектов (данных) различных уровней. В этом смысле иерархические базы данных очень напоминают файловую систему компьютера [14].

Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка (объект более близкий к корню) к потомку (объект более низкого уровня), при этом возможна ситуация, когда объект-предок не имеет потомков или имеет их несколько, тогда как у объекта-потомка обязательно только один предок.

В реляционных СУБД (РСУБД) данные собраны в таблицы, которые в свою очередь состоят из столбцов и строк, на пересечении которых расположены ячейки. Запросы к таким базам данных возвращает таблицу, которая повторно может участвовать в следующем запросе. Данные в одних таблицах, как

правило, связаны с данными других таблиц, откуда и произошло название "реляционные" [15, 16].

В объектно-ориентированных базах данных данные хранятся в виде объектов. С объектно-ориентированными базами данных удобно работать, применяя объектно-ориентированное программирование. Эти системы управления базами данных обрабатывают данные как абстрактные объекты, наделенные свойствами и использующие методы взаимодействия с другими объектами окружающего мира. Однако на сегодняшний день такие базы данных еще не достигли популярности реляционных, поскольку пока значительно уступают им в производительности [15].

Гибридные СУБД совмещают в себе возможности реляционных и объектно-ориентированных баз данных.

В WEB-приложениях, как правило, используются реляционные базы данных. Эта модель данных отображает информацию в наиболее простой для пользователя форме, основана на развитом математическом аппарате, который позволяет достаточно лаконично описать основные операции над данными. Базы данных с реляционной моделью позволяют создавать языки манипулирования данными не процедурного типа, а также производить манипулирование данными на уровне выходной БД и возможность изменения данных [15].

Одна из самых распространенных полноценных серверных СУБД, которая будет использоваться для реализации системы Internet-продажи кондитерских изделий – MySQL. MySQL очень функциональная, свободно распространяемая СУБД, которая успешно работает с различными сайтами и WEB-приложениями. Несмотря на то, что в ней не реализован весь SQL (языка структурированных запросов) функционал, MySQL предлагает довольно много инструментов для разработки приложений. Так как это серверная СУБД, приложения для доступа к данным, в отличие от, например, SQLite, работают со службами MySQL [16].

Преимущества MySQL [17]:

- простота в работе – установить MySQL довольно просто; дополнительные приложения позволяет довольно легко работать с БД;
- богатый функционал – MySQL поддерживает большинство функционала SQL;
- безопасность – большое количество функций обеспечивающих без-

опасность, которые поддерживается по умолчанию;

- масштабируемость – MySQL легко работает с большими объемами данных и легко масштабируется;
- скорость работы – упрощение некоторых стандартов позволяет MySQL значительно увеличить производительность;
- кроссплатформенность – в настоящее время существуют версии программы для большинства распространенных компьютерных платформ, что устраняет необходимость в использовании определенной операционной системы (ОС); это позволяет свободно выбирать, с чем работать, например с Linux или Windows, но даже в случае замены ОС не произойдет потери данных и даже не понадобятся дополнительные инструменты для их переноса;
- возможность работы нескольких пользователей, что является немаловажным при работе (или разработке) WEB-приложений.

Суммируя вышесказанное, можно сделать вывод, что возможности СУБД MySQL вполне удовлетворяют все потребности большинства Internet-проектов, таких как система Internet-продажи кондитерских изделий, поэтому данная СУБД будет использоваться в разрабатываемой системе.

3.2 Обоснование выбора языка программирования для разрабатываемой системы

Существует множество языков программирования, предназначенных для выполнения различных задач. Каждый из них характеризуется уникальным набором операторов и особым синтаксисом.

Стоит взять на рассмотрение три наиболее популярных языка, применяемых в WEB-разработке [18]:

- Personal Home Pages (PHP);
- Ruby;
- Python.

PHP. PHP является самым популярным языком для Internet. Это язык программирования, исполняемый на стороне WEB-сервера, спроектированный Расмусом Лердорфом (Rasmus Lerdorf) в качестве инструмента создания динамических и интерактивных WEB-сайтов. Этот язык оказался достаточно гибким и мощным, поэтому приобрёл большую популярность и используется в

проектах любого масштаба: от простого блога до крупнейших WEB-приложений в Internet- [5, 19].

Преимущества PHP [18, 19, 20]:

- является свободным программным обеспечением, распространяемым под особой лицензией (PHP license);
- легок в освоении на всех этапах;
- язык поддерживается большим сообществом пользователей и разработчиков;
- имеет развитую поддержку баз данных;
- имеется огромное количество библиотек и расширений языка;
- может использоваться в изолированной среде;
- предлагает нативные средства организации WEB-сессий, программный интерфейс расширений;
- является довольно полной заменой проприетарной среды Active Server Pages (ASP) от Microsoft;
- может быть развёрнут почти на любом сервере;
- портирован под большое количество аппаратных платформ и операционных систем.

Недостатки PHP [19, 20, 21]:

- не подходит для создания «десктопных» приложений или системных компонентов;
- имеет слабые средства для работы с исключениями;
- глобальные параметры конфигурации влияют на базовый синтаксис языка, что затрудняет настройку сервера и разворачивание приложений;
- объекты передаются по значению, что смущает многих программистов, привыкших к передаче объектов по ссылке, как это делается в большинстве других языков.

Ruby. Динамический императивный объектно-ориентированный язык программирования, разработанный Юкихио Матсумото. Ruby был создан под влиянием таких языков, как Perl, Eiffel и Smalltalk [20].

Он характеризуется динамической типизацией и автоматическим управлением памятью. Язык Ruby используется в WEB-разработке в составе открытого WEB-framework Rails, чаще называемого Ruby on Rails (RoR) [22].

Преимущества Ruby [20, 22]:

- открытая разработка;
- работает на многих платформах;
- относится к языкам программирования сверхвысокого уровня (VHLL), то есть обладает высоким уровнем абстракции и предметным подходом в реализации алгоритмов;
- реализует концептуально чистую объектно-ориентированную парадигму;
- предоставляет продвинутые методы манипуляции строками и текстом;
- легко интегрирует в свои программы высокопроизводительные серверы баз данных;
- имеется простой программный интерфейс для создания многопоточных приложений;
- имеет продвинутые средства для работы с массивами;
- возможности языка можно расширить при помощи библиотек, написанных на C или Ruby;
- зарезервированные слова могут являться идентификаторами, если это не создаёт неоднозначности для парсера;
- встроенный отладчик.

Недостатки Ruby [20, 21]:

- обучение языку выше начального уровня может оказаться непрос-тым;
- информационных ресурсов, посвящённых Ruby, явно недостаточно;
- менее производителен по сравнению со многими другими языками, применяемыми в WEB-разработке;
- относительно медленно разрабатывается и развивается.

Python. Язык широко применяется как интерпретируемый язык для скриптов различного назначения (хотя существуют и трансляторы языка Python) [20, 23].

Как и Ruby, Python имеет целью приблизить синтаксис реальной программы, написанной на нём, к описывающему задачу псевдокоду, что позволяет программисту уменьшить объём программы. Идея создания данного языка возникла в конце 1980-х и была реализована Гвидо ван Россумом [20].

Элегантный дизайн и эффективный, дисциплинирующий синтаксис этого

языка облегчают программистам совместную работу над кодом. Python – мультипарадигмальный язык программирования. Он позволяет совмещать процедурный подход к написанию кода с объектно-ориентированным и функциональным [20].

Преимущества Python [19, 20]:

- открытая разработка;
- довольно прост в изучении, особенно на начальном этапе;
- особенности синтаксиса стимулируют программиста писать хорошо читаемый код;
- предоставляет средства быстрого прототипирования и динамической семантики;
- множество полезных библиотек и расширений языка можно легко использовать в своих проектах благодаря предельно унифицированному механизму импорта и программным интерфейсам;
- механизмы модульности хорошо продуманы и могут быть легко использованы;
- абсолютно всё в Python является объектами в смысле ООП, но при этом объектный подход не навязывается программисту.

Недостатки Python [19, 20]:

- не слишком удачная поддержка многопоточности;
- на Python создано не так уж много качественных программных проектов по сравнению с другими универсальными языками программирования, например, с Java;
- отсутствие коммерческой поддержки средств разработки;
- изначальная ограниченность средств для работы с базами данных;
- benchmark-и показывают меньшую производительность Python по сравнению с основными Java VM, что создаёт этому языку репутацию медленного.

Подытожить можно следующим:

- PHP – лучший язык для создания динамических WEB-страниц;
- Python – универсальный язык программирования, при помощи которого можно делать любые приложения в диапазоне от WEB-сайтов и «desktopных» приложений до роботов и системных сервисов;
- Ruby – наиболее высокоуровневый язык, позволяющий разработчи-

ку уделять меньше внимания деталям интерфейса и организации хранения данных, чтобы сосредоточиться на прикладной задаче.

Каждый из этих языков имеет преимущества и предоставляет достаточно возможностей для реализации системы Internet-продажи кондитерских изделий. Однако был выбран язык PHP, в связи с тем, что этот язык ориентирован именно на WEB-разработку, по сравнению с универсальным языком Python, умеющего ограниченный набор средств для работы с базами данных, а также, в свою очередь, превосходящего Ruby в количестве информационных ресурсов, а также скорости разработки и развития.

Можно также добавить, что поскольку в мире большинство сайтов написано на PHP, это свидетельствует о том, что данный язык по многим параметрам опережает своих конкурентов и имеет довольно много библиотек и инструментов разработки, а также будет востребован ещё длительное время.

3.3 Обоснование выбора framework

Framework представляет собой программную платформу, определяющую структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. Зачастую, его также называют «каркасом» [18, 24, 25].

Можно также говорить о каркасном подходе как о подходе к построению программ, где любая конфигурация программы строится из двух частей: первая, постоянная часть – каркас, не меняющийся от конфигурации к конфигурации и несущий в себе гнезда, в которых размещается вторая; вторая, переменная часть – сменные модули (или точки расширения) [24].

Одним из главных преимуществ при использовании каркасных приложений является то, что такие приложения имеют стандартную структуру. Каркасы приложения стали популярны с появлением графических интерфейсов пользователя, которые имели тенденцию к реализации стандартной структуры для приложений. С их использованием стало гораздо проще создавать средства для автоматического создания графических интерфейсов, так как структура внутренней реализации кода приложения стала известна заранее [25].

Для обеспечения каркаса обычно используются техники объектно-ориентированного программирования (например, части приложения могут

наследоваться от базовых классов framework).

Одним из PHP framework является yii framework, главное преимущество которого – гибкость [26]. Yii framework позволяет решать разные задачи без надобности искать какие-либо пути обхода. Он предоставляет разработчику все инструменты для создания, как небольших сайтов, так и сложных порталов. WEB-приложение, обеспечивающее работу системы Internet-продажи кондитерских изделий, будет разработано на базе yii framework (точнее на его второй версии – yii 2). Данный framework приложений основанный на композитном паттерне Модель-Вид-Контроллер, что удовлетворяет выбор структуры приложения.

Yii 2 является высокопроизводительным framework, используемым для быстрой разработки WEB-приложений. Он хорошо спроектирован, имеет хорошую поддержку, а также легок в изучении и сопровождении [26].

Данный framework подходит для построения, как приложений командной строки, так и WEB-приложений, однако состав его возможностей делает его наиболее полезным при разработке именно WEB-приложений. Он содержит несколько средств для автоматической генерации исходного кода, включая создатель полноценных Create-Read-Update-Delete (CRUD) интерфейсов. Он в значительной степени полагается на соглашения, выраженные в его настройках по умолчанию [19, 26].

3.4 Обоснование выбора WEB-сервера

WEB-сервер представляет собой сервер, принимающий HTTP-запросы от клиентов, обычно WEB-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными.

WEB-сервером называют как программное обеспечение, выполняющее функции WEB-сервера, так и непосредственно компьютер, на котором это программное обеспечение работает [27].

Клиент, которым обычно является WEB-браузер, передаёт WEB-серверу запросы на получение ресурсов, обозначенных URL-адресами. В качестве ресурсов выступают HTML-страницы, изображения, файлы, медиа-поток или другие данные, которые необходимы клиенту. В ответ WEB-сервер передаёт

клиенту запрошенные данные. Этот обмен происходит по протоколу HTTP.

Также принято решение выполнять реализацию WEB-приложения на Ubuntu (операционной системе, дистрибутиве, основанной на Debian GNU/Linux). Поэтому, был выбран один из самых популярных Ubuntu WEB-серверов – Apache2 [27, 28].

Стоит отметить, что лучшим решением для реализации WEB-приложений на Ubuntu является установка набора (комплекса) серверного программного обеспечения LAMP, названного по первым буквам входящих в его состав компонентов:

- Linux (операционная система Linux);
- Apache (WEB-сервер);
- MySQL (СУБД);
- PHP (язык программирования, используемый для создания WEB-приложений) [27, 29].

Одними из преимуществ LAMP являются значительная гибкость в выборе различных баз данных, WEB серверов и языков сценариев, а также простота в установке и использовании [29, 30].

В данном случае, также немаловажным является то, что LAMP удовлетворяем выбранным ранее СУБД и языку программирования. Установив данный комплекс серверного программного обеспечения можно приступить к реализации WEB-приложения – системы Internet-продажи кондитерских изделий.

4 ФУНКЦИОНАЛЬНАЯ РЕАЛИЗАЦИЯ СИСТЕМЫ

4.1 Описание процесса разработки приложения

После завершения установки комплекса серверного программного обеспечения, для того, чтобы «развернуть» yii-приложение, необходимо скачать исполняемый файл Composer – менеджер зависимостей для PHP, или установить плагин Composer Assets Manager и затем установить новый проект через Composer [26].

Для установки расширенного шаблона проекта yii, в терминале Ubuntu были выполнены следующие команды (рис. 4.1).

```
$ curl -sS https://getcomposer.org/installer | php  
$ php composer.phar global \  
    require "fxp/composer-asset-plugin:1.0.0-beta3"  
$ php composer.phar create-project --prefer-dist -- stability=dev \  
    yiisoft/yii2-app-advanced Magaz
```

Рисунок 4.1 – Установка расширенного шаблона yii2-приложения

После инициализации проекта можно приступить к разработке БД, которая будет использоваться приложением. Стоит отметить, что в файле «common/config/main-local.php» необходимо указать конфигурации для работы с базой данных.

После того, как база данных создана (пустая), необходимо запустить «миграции», позволяющие обновлять структуру базы данных от одной версии до другой (обычно более новой) [31].

4.1.1 Разработка базы данных системы

Для создания миграции, в терминале необходимо прописать команду «php yii migrate/create» и указать название миграции [32].

В результате выйдет, например, такая команда (в которой будет производиться создание таблицы «shop»):

```
$ php yii migrate/create 'create_table_shop'.
```

Framework сам генерирует php-файл, в который необходимо написать код миграции, в данном случае, для создания таблицы «shop» (рис. 4.2).

```
<?php

use yii\db\Schema;
use yii\db\Migration;

class m160602_233040_create_table_magaz extends Migration
{
    public function up()
    {
        $tableOptions = null;
        if ($this->db->driverName === 'mysql') {
            // http://stackoverflow.com/questions/766809/whats-the-difference-between-utf8-general-ci-and-utf8-unicode-ci
            $tableOptions = 'CHARACTER SET utf8 COLLATE utf8_unicode_ci ENGINE=InnoDB';
        }
        $this->createTable('{{%shop}}', [
            'id' => $this->primaryKey(),
            'name' => $this->string(32)->notNull()->unique(),
            'info' => $this->text(),
            'created_at' => $this->integer()->notNull(),
            'updated_at' => $this->integer()->notNull(),
            'status' => $this->smallInteger()->defaultValue(10),
        ], $tableOptions);

    }

    public function down()
    {
        $this->dropTable('{{%shop}}');
    }
}
```

Рисунок 4.2 – Код класса миграции для создания таблицы «shop»

Для создания таблицы «shop» необходимо произвести миграцию (рис. 4.3) – команда:

```
$ php yii migrate .
```

```
blackghoul@woodenpc:/var/www/html/old_magaz/magaz$ php yii migrate
Yii Migration Tool (based on Yii v2.0.6)

Total 1 new migration to be applied:
    m160602_233040_create_table_magaz

Apply the above migration? (yes|no) [no]:y
*** applying m160602_233040_create_table_magaz ***
> create table {{%shop}} ... done (time: 0.124s)
*** applied m160602_233040_create_table_magaz (time: 0.176s) ***
Migrated up successfully.
blackghoul@woodenpc:/var/www/html/old_magaz/magaz$
```

Рисунок 4.3 – Создание таблицы «shop»

Таким же образом, создаются все остальные миграции, описывающие все изменения в БД. Созданная в результате база данных для системы Internet-продажи кондитерских изделий представлена на рисунке 4.4.

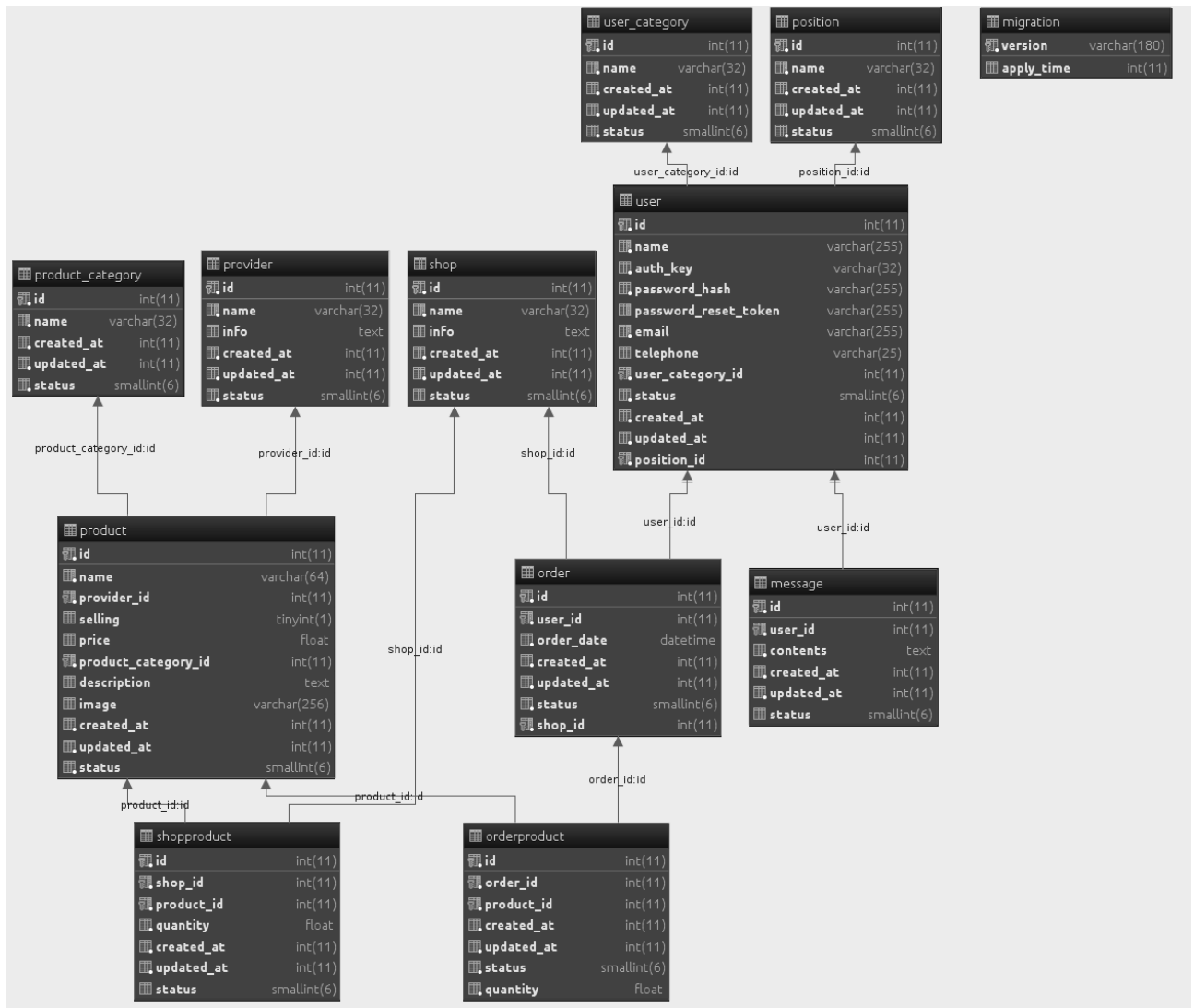


Рисунок 4.4 – Схема базы данных

От созданной на этапе проектирования базы данных, результирующая БД отличается тем, что в нее добавлены две новые таблицы – «position», «shop», «shopproduct», а также «messages» (не считая автоматически создаваемую yii, таблицу «migration», содержащую историю миграций). Также, в таблицы добавлены некоторые поля, такие как «created_at», «updated_at», «status» и пр., служащие для более организованной работы приложения.

4.1.2 Разработка WEB-приложения

Создав базу данных, можно приступить к разработке приложения. Стоит отметить, что созданный yii проект (advanced project) можно разделить на 3 основные части:

- frontend (клиентская);
- backend (администраторская);
- common (общая – общие файлы-модели, использующиеся как на frontend, так и на backend).

Безусловно, разработку стоит начать с backend и common частей, для создания возможности быстрого и удобного редактирования клиентской части.

Исходя из того, что предусматривается MVC-паттерном, вся суть работы приложения заключается в том, что пользователь, переходя по страницам (ссылкам) приложения, заставляет его определить необходимый контроллер (Controller), а также действие (Action) контроллера, выполняющего какие-либо операции в зависимости от запроса или условий. Обычно, действия выполняют отображения страниц вида (View), вызовы методов моделей и т. п. Изначально, в проекте yii существуют Контроллеры, выполняющие шаблонные действия, такие как отображение главной страницы сайта.

«Точками входа» в приложение являются файлы «index.php», расположенные в каталогах «backend/web/» и «frontend/web/». В них подключаются конфигурации, а также вызывается метод run(), запускающий приложение.

Файл «index.php» содержит следующий код:

```
<?php
defined('YII_DEBUG') or define('YII_DEBUG', true);
defined('YII_ENV') or define('YII_ENV', 'dev');

require(__DIR__ . '/../vendor/autoload.php');
require(__DIR__ . '/../vendor/yiisoft/yii2/Yii.php');
require(__DIR__ . '/../common/config/bootstrap.php');
require(__DIR__ . '/../config/bootstrap.php');

$config = yii\helpers\ArrayHelper::merge(
    require(__DIR__ . '/../common/config/main.php'),
    require(__DIR__ . '/../common/config/main-local.php'),
    require(__DIR__ . '/../config/main.php'),
    require(__DIR__ . '/../config/main-local.php')
);
```



```
$application = new yii\web\Application($config);
$application->run();
```

В конфигурациях указывается файл вида, который будет отображаться по умолчанию (вместе с другими view-файлами) – «backend/views/layouts/main.php». Здесь описана навигационная панель (NavBar), содержащая ссылки, доступные на любой странице WEB-сайта:

```
<?php NavBar::begin([
    'brandLabel' => 'Магазин',
    'brandUrl' => '../old_magaz/magaz/frontend/web',
    'options' => [
        'class' => 'navbar-default navbar-fixed-top',
    ],
]);
if (Yii::$app->user->isGuest) {
    $menuItems[] = ['label' => 'Войти', 'url' => ['/site/login']];
} else {
    $menuItems = [
        ['label' => 'Заказы', 'url' => ['/orders/list']],
        ['label' => 'Добавить заказ', 'url' => ['/orders/make_order']],
        ['label' => 'Учет', 'url' => ['/shops/products']],
        ['label' => 'Товар', 'url' => ['/product/list']],
    ];
    $menuItems[] = [
        'label' => ' Меню ',
        'items' => [
            '<li class="dropdown-header"></li>', [
                'label' => 'Поставщики',
                'url' => ['/providers/list'],
            ],
            '<li class="dropdown-header"></li>', [
                'label' => 'Пользователи',
                'url' => ['/user/list'],
            ],
            '<li class="dropdown-header"></li>', [
                'label' => 'Категории товаров',
                'url' => ['/product_categories/list'],
            ],
            '<li class="dropdown-header"></li>', [
                'label' => 'Магазины (помещения)',
                'url' => ['/shops/list'],
            ],
            '<li class="dropdown-header"></li>', [
                'label' => 'Должности сотрудников',
                'url' => ['/position/list'],
            ],
        ],
    ];
}
```

```

    ],
    '<li class="dropdown-header"></li>', [
        'label' => 'Выйти (' . Yii::$app->user->identity['name'] . ')',
        'url' => ['/site/logout'],
        'linkOptions' => ['data-method' => 'post']
    ]
    ]
    ];
}
echo Nav::widget([
    'options' => ['class' => 'navbar-nav navbar-right'],
    'items' => $menuItems,
]);
NavBar::end() ?>

```

Таким образом, к примеру,

```
'label' => 'Товар', 'url' => ['/product/list']
```

отобразит на панели пункт под названием «Товар», а также, будучи выбранным, заставит приложение перейти на контроллер «Product», действие «list» (для отображения списка товара). Для этого создан класс контроллер с названием «ProductController», а также описан метод «actionList» (файл «backend/controllers/ProductController.php»):

```

public function actionList()
{
    if (Yii::$app->user->isGuest)
        return $this->redirect('../site/login.html');
    $model = new Product();
    if (isset($_GET['delete']) && $_GET['delete'] == 1)
        Yii::$app->session->setFlash('success', 'Удаление произведено успешно.');
```

```

    if (isset($_GET['added']) && $_GET['added'] == 1)
        Yii::$app->session->setFlash('success', 'Новый товар успешно добавлен.');
```

```

    if (isset($_GET['deleted']))
        Yii::$app->session->setFlash('error', 'Такого товара не существует. Или, возможно, он удален.');
```

```

    if (!isset($_POST['id']) && !isset($_POST['add']))
        return $this->render('list', [
            'model' => $model
        ]);
    else {
        if (isset($_POST['id'])) {
            $model->id = $_POST['id'];
            return $this->redirect(['edit_product',
                'id' => $model->id,
            ]);
        }
    }
}

```

```

        });
    }
    if (isset($_POST['add'])) {
        return $this->redirect(['add_product']);
    }
}
}

```

Действие `actionList()` первым делом проверяет, авторизован ли пользователь, а затем перенаправляет на страницу авторизации, в случае неудачи. В массив `$_POST` или `$_GET` записывается значение нажатой пользователем кнопки, описанной во `view`-файле «`list.php`» (для `$_POST`), или запросы с других страниц `WEB`-приложения, передаваемые с помощью `$_GET`. Поэтому далее, проверяются глобальные массивы (`$_POST` и `$_GET`) с целью, в данном случае, перенаправить пользователя на страницы редактирования или добавления товара в БД. Так как изначально `$_POST` пуст – действие «`list`» (`actionList()`) просто отобразит файл вида «`list.php`», находящегося в «`backend/views/product/`».

В этом файле описана `HTML`-разметка, создающая таблицу (приложение А), а также производится вывод данных, полученных из модели «`Product`», которая, в свою очередь, связана с таблицей «`product`» в базе данных.

Для указания данной связи в модели «`Product`» необходимо описать, унаследованный от класса-предка «`Model`», метод «`tableName()`»:

```

/**
 * @inheritdoc
 */
public static function tableName()
{
    return 'product';
}

```

В модели «`Product`», также необходимо описать переменные, заносимые в поля таблицы, а также указать правила валидации полей ввода данных в приложении – `HTML`-форм (метод «`rules()`»):

```

/**
 * Product model
 *
 * @property integer $id
 * @property string $name

```

```

* @property integer $provider_id
* @property integer $product_category_id
* @property boolean $selling
* @property float $price
* @property string $description
* @property string $image
* @property integer $created_at
* @property integer $updated_at
* @property integer $status
*/

```

.....

```

public function rules()
{
    return [
        [['name', 'description', 'price', 'image'], 'trim'],
        [['name', 'provider_id', 'product_category_id'], 'required'],
        ['name', 'unique', 'targetClass' => '\common\models\Product',
            'message' => 'Уже занято такое название товара.'],
        ['name', 'string', 'min' => 4, 'max' => 32],
        [['description', 'image'], 'string', 'min' => 0, 'max' => 256],
        ['selling', 'in', 'range' => [0, 1]],
        ['selling', 'default', 'value' => '0'],
        [['file'], 'file'],
        ['priceKop', 'string', 'min' => 0, 'max' => 2],
        [['priceKop', 'priceGrn'], 'default', 'value' => '0'],
        ['order', 'string'],
        [['delete_accept', 'delete_image'], 'in', 'range' => [0, 1]],
        ['status', 'default', 'value' => self::STATUS_ACTIVE],
        ['status', 'in', 'range' => [self::STATUS_ACTIVE, self::STATUS_DELETED]],
    ];
}

```

Занесения в БД поступающих данных реализуется следующим образом:

```

public function updateProduct($product)
{
    if ($product->priceGrn < 0 || $product->priceKop < 0)
        return false;
    $this->name = $product->name;
    $this->provider_id = $product->provider_id;
    $this->selling = $product->selling;
    $this->price = $product->priceGrn + $product->priceKop / 100;
    $this->description = $product->description;
    $this->image = $product->image;
    $this->product_category_id = $product->product_category_id;
    return $this->save() ? true : false;
}

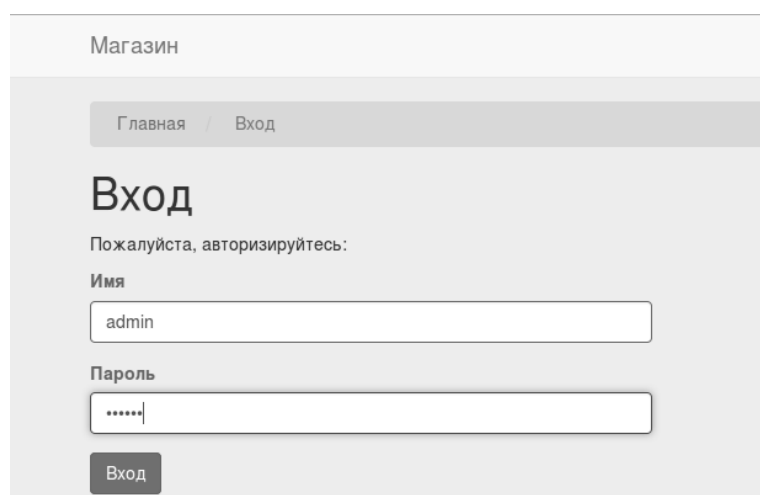
```

Аналогичным образом описываются методы для удаления товара (точнее – присвоения статуса «удален»), удаления изображения, а также изменения категории товара или поставщика данного товара, что позволяет WEB-приложению предусмотреть внешние связи между таблицами в БД. То есть, при попытке удаления какого-нибудь «поставщика товара» в том случае, если имеется «поставляемый им товар» – связь, приложение предупредит пользователя об этой связи, а также предложит удалить все поставляемые этим поставщиком «товары» или присвоить им нового «поставщика». В противном случае, «поставщик» не будет удален.

За исключением некоторых особенностей, подобным обзором для реализации системы, создаются другие компоненты – файлы вида, контроллеры и модели. Стоит обратить внимание на то, что большинство моделей расположены в «common/Models» по той причине, что они используются как во frontend, так и в backend.

4.2 Анализ эффективности и результаты разработки системы

Чтобы начать работу в системе (backend), необходимо выполнить авторизацию – войти в систему (рис. 4.5). При этом, войти в систему могут только пользователи с уровнем доступа «Администратор (полный)» или «Сотрудник (чтение)».



Магазин

Главная / Вход

Вход

Пожалуйста, авторизуйтесь:

Имя

Пароль

Вход

Рисунок 4.5 – Вход в систему

Выполнив вход в систему, пользователю отображается главная страница,

а также навигационная панель (NavBar) WEB-сайта (рис. 4.6). Выбрав в меню панели, например, пункт «поставщик», приложение перенаправит пользователя на страницу со списком поставщиков (рис. 4.7).



Рисунок 4.6 – Панель навигации



Рисунок 4.7 – Страница «Поставщики»

На странице из базы данных выводится в виде таблицы информация о поставщиках. Информацию можно добавлять, редактировать или удалять. На рисунках 4.8-4.9 представлены результаты редактирования информации.

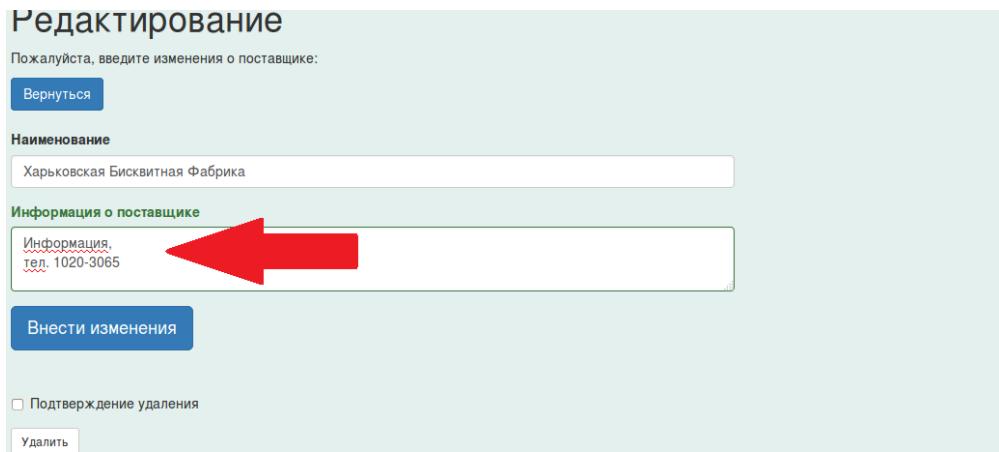


Рисунок 4.8 – Редактирование информации о поставщике

При добавлении нового «поставщика» отображается страница «Добавить поставщика», представленная на рисунке 4.10.



Рисунок 4.9 – Изменение в таблице «Поставщики»

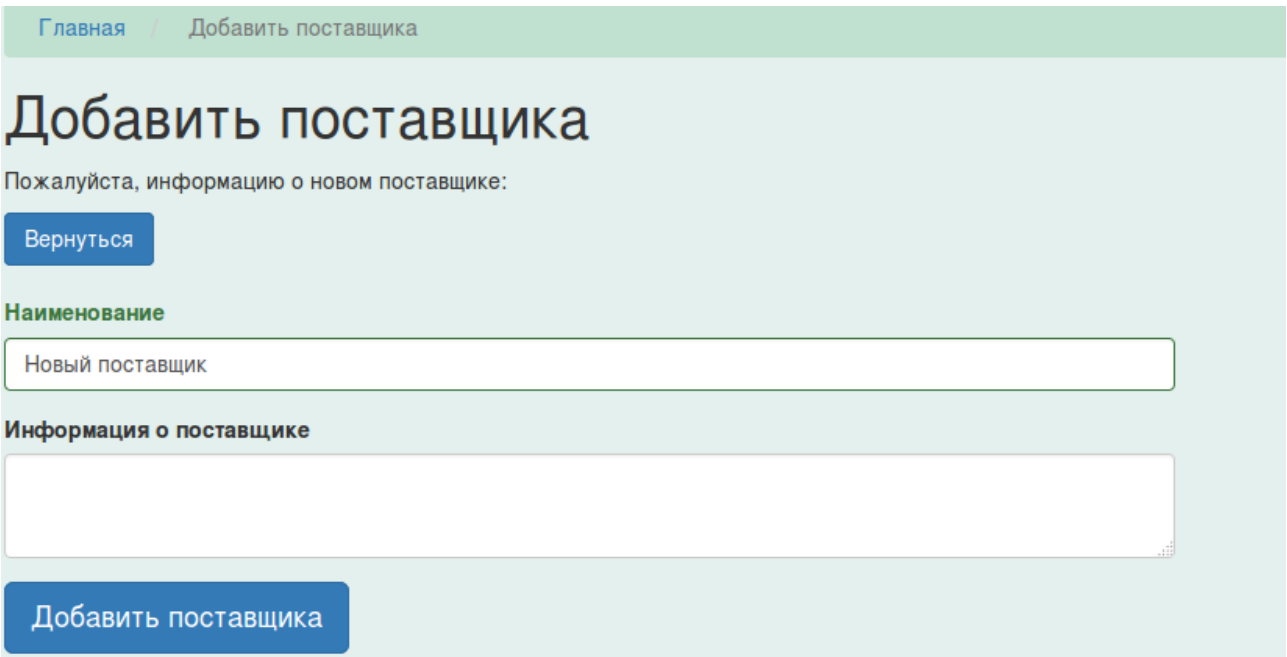


Рисунок 4.10 – Страница «Добавить поставщика»

Результат добавления нового «поставщика» представлен на рисунке 4.11. Для проверки, запись «Новый поставщик» была удалена (рис 4.12).

Удаление записей в системе производится через их редактирование. Также, для уменьшения риска случайного удаления записи, действие должно быть подтверждено.

Таким же образом, в базу данных, для дальнейшего проведения анализа работоспособности системы, были добавлены два поставщика – «Тестовый поставщик» и «Тестовый поставщик 2».

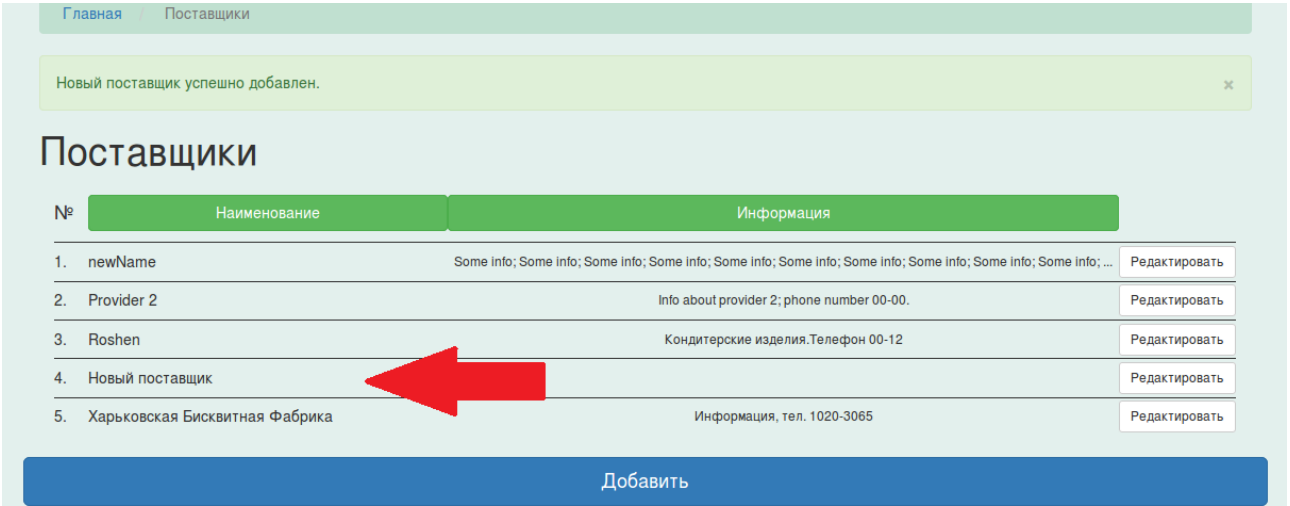


Рисунок 4.11 – Добавление нового «поставщика»

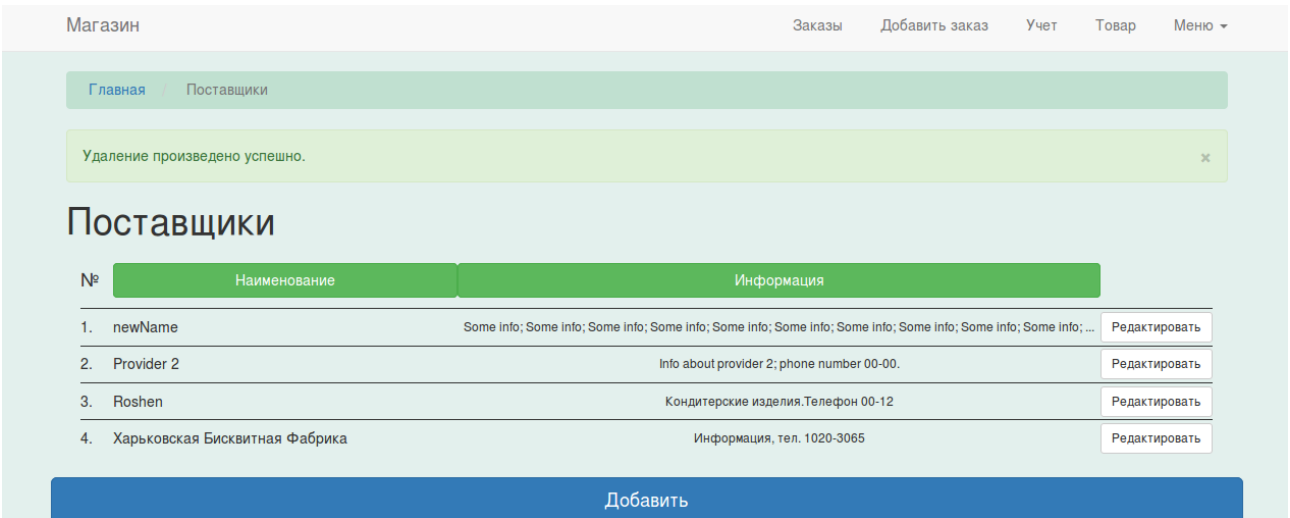


Рисунок 4.12 – Результат удаления записи «Новый поставщик»

Стоит добавить, что действия этих трех страниц (списка, добавления и редактирования поставщиков) описаны в одном контроллере – «ProvidersController». Действия следующие:

- actionList() – список;
- actionEdit() – редактирование;
- actionAdd() – добавление.

Такой же принцип сохраняется и на других страницах приложения.

Далее, произведен переход на страницу «Товар», на которой отображается информация о товаре (рис. 4.13).



Рисунок 4.13 – Страница списка товара

На рисунке 4.14 представлена работа описанных в модели «Product» правил валидации (метод rules()) при добавлении «товар» в базу данных.

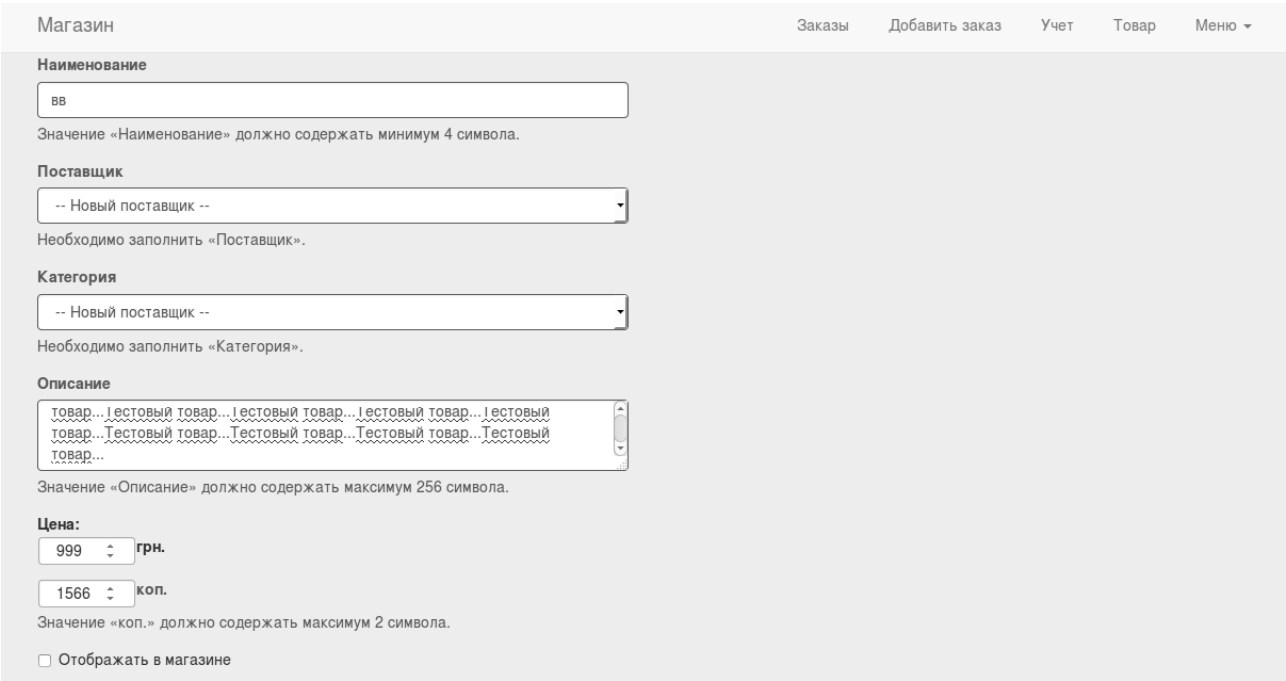


Рисунок 4.14 – Валидация данных о товаре

Заполнив поля в соответствии с правилами валидации, а также загрузив файл изображения, «товар» добавлен в базу данных системы (рис. 4.15-4.16).

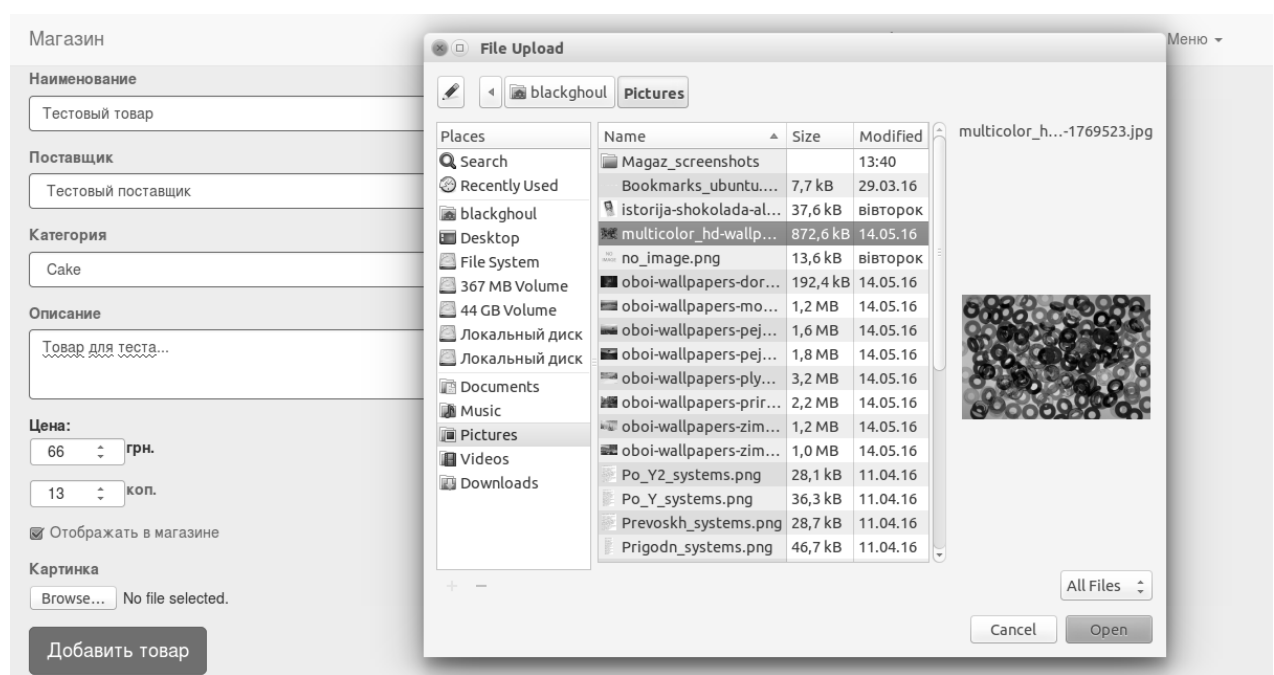


Рисунок 4.15 – Поля, заполненные в соответствии с правилами валидации

Новый товар успешно добавлен.

Товар

№	Название	Поставщик	Цена	Описание	Категория	В магазине	
1.	Product	newName	26.30		Шоколад 100 г	✗	Редактировать
2.	product 2	newName	36.20		Cookies	✗	Редактировать
3.	Декстер	Roshen	26.05	Хавает хаку.	Шоколад 200 г	✓	Редактировать
4.	Еще товар	Provider 2	13.66	Текс описания товара... Те...	Cake	✗	Редактировать
5.	Молочный шоколад "Ал...	Roshen	12.35	12/24 шт. в блоке.	Шоколад 100 г	✓	Редактировать
6.	Печенье "Ромашка"	Харьковская Бисквитн...	25.95	Вкусная печенюха	Cookies	✓	Редактировать
7.	Тестовый товар	Тестовый поставщик	66.13	Товар для теста...	Cake	✓	Редактировать
8.	Товар_1	Provider 2	9.10	Some description; Some des...	Ice cream	✗	Редактировать
9.	Харьковский вафельны...	Харьковская Бисквитн...	200.00		Cake	✓	Редактировать

Рисунок 4.16 – Добавление нового «товара»

Теперь, когда «Тестовый поставщик» используется в добавленном «Тестовом товаре», то есть id «Тестового поставщика» указано в связанной с таблицей «provider» с помощью внешнего ключа таблице «product», разработанное приложение предусматривает данную связь и при попытке удаления «Тестового поставщика» предупреждает пользователя о связи, предоставив ему следу-

ющие варианты:

- удаление всех «продуктов», поставляемых удаляемым поставщиком;
- назначение «продуктам», поставляемым удаляемым поставщиком, другого поставщика.

На рисунке 4.17 представлен процесс удаления «Тестового поставщика» с присвоением, поставляемым им товарам, другого поставщика – в данном случае, «Тестового поставщика 2».

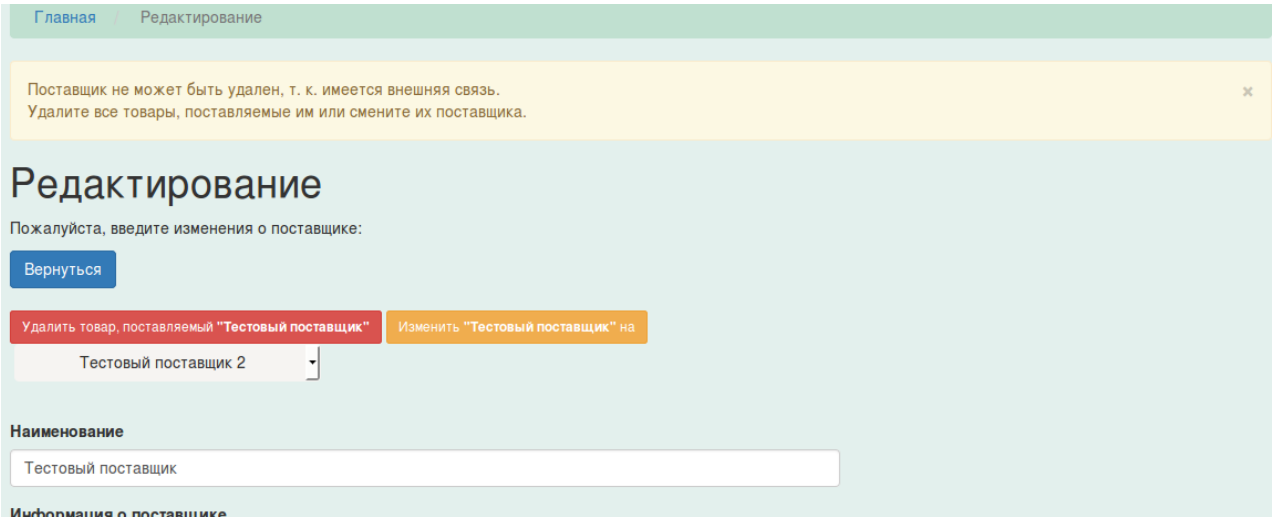


Рисунок 4.17 – Удаление «Тестового поставщика»

В результате удаления, «Тестовый товар» использует id «Тестового поставщика 2» (рис. 4.18).

Товар						
№	Название	Поставщик	Цена	Описание	Категория	В магазине
1.	Product	newName	26.30		Шоколад 100 г	✗ Редактировать
2.	product 2	newName	36.20		Cookies	✗ Редактировать
3.	Декстер	Roshen	26.05	Хакает хавку.	Шоколад 200 г	✓ Редактировать
4.	Еще товар	Provider 2	13.66	Текс описания товара... Те...	Cake	✗ Редактировать
5.	Молочный шоколад "Ал...	Roshen	12.35	12/24 шт. в блоке.	Шоколад 100 г	✓ Редактировать
6.	Печенье "Ромашка"	Харьковск... Бисквитн...	25.95	Вкусная печенюха	Cookies	✓ Редактировать
7.	Тестовый товар	Тестовый поставщик 2	66.13	Товар для теста...	Cake	✓ Редактировать
8.	Товар_1	Provider 2	9.10	Some description; Some des...	Ice cream	✗ Редактировать
9.	Харьковский вафельны...	Харьковская Бисквитн...	200.00		Cake	✓ Редактировать

Рисунок 4.18 – Проверка результатов удаления «Тестового поставщика»

Одним из назначений разработанной системы является ведение учета товара. Перейдя на страницу «Учет», выбрав из списка магазинов (складов) необходимый, приложение отображает список всего товара, а также список товара, имеющегося в выбранном магазине или складе, и предоставляет возможность его редактирования (рис. 4.19-4.21).

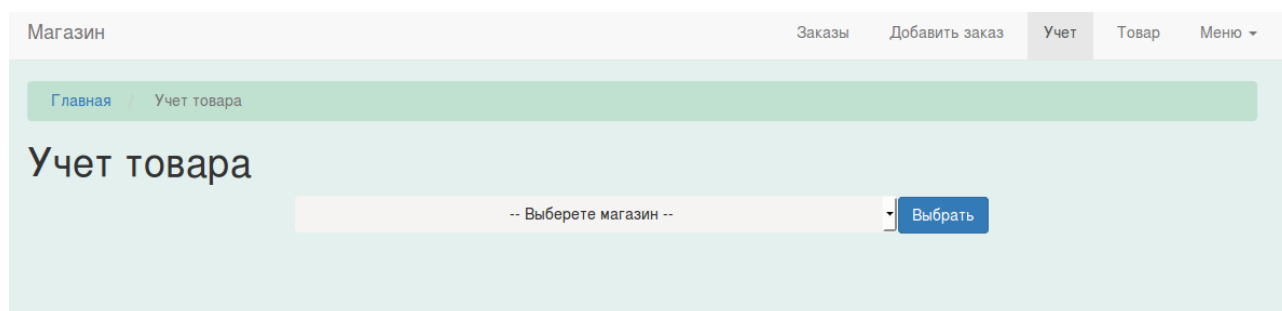


Рисунок 4.19 – Страница ведения учета товара

№	Название	Поставщик	Цена	Категория	Кол-во	
Товара на сумму = 0 \$						
Отсутствует в Магазин_1						
1.	Product	newName	26.30	Шоколад 100 г...		Добавить
2.	product 2	newName	36.20	Cookies		Добавить
3.	Декстер	Roshen	26.05	Шоколад 200 г...		Добавить
4.	Еще товар	Provider 2	13.66	Cake		Добавить
5.	Молочный шоколад "Ал...	Roshen	12.35	Шоколад 100 г...		Добавить
6.	Печенье "Ромашка"	Харьковская Бискупити	25.95	Cookies		Добавить

Рисунок 4.20 – Добавление «товара» в «Магазин_1»

№	Название	Поставщик	Цена	Категория	Кол-во ↓↑	
Имеется в Магазин_1						
1.	Product	newName	26.30	Шоколад 100 г...	5	Удалить - 5
2.	product 2	newName	36.20	Cookies	4	Удалить
3.	Еще товар	Provider 2	13.66	Cake	4	Удалить
4.	Декстер	Roshen	26.05	Шоколад 200 г...	1	Удалить
Товара на сумму = 356.91						
Весь товар						
1.	Product	newName	26.30	Шоколад 100 г...		Добавить

Рисунок 4.21 – Редактирование списка товара в «Магазин_1»

В администраторской части WEB-приложения отображается список всех

пользователей системы, что предоставит сотрудникам возможность просматривать необходимую информацию о клиентах или других сотрудниках, а также предоставит администратору возможность редактировать некоторую информацию о пользователях в системе и назначить им требуемых уровень доступа (рис 4.22-4.23).

Магазин

Заказы

Добавить заказ

Учет

Товар

Меню ▾

Главная / Пользователи

Пользователи

№	Имя	e-mail	Телефон	Уровень доступа	Должность	
1.	admin	admin@admin.com	(666) 777-1313	Администратор (полный)	Администратор	Редактировать
2.	admin2	admin2@admin.com	(666) 666-7777	Администратор (полный)	Администратор	Редактировать
3.	client	client@client.com	(989) 898-9898	Клиент (нет доступа)	Клиент	Редактировать
4.	client2	client2@client.com	(012) 345-6789	Клиент (нет доступа)	Клиент	Редактировать
5.	client3	client3@client.com	(987) 654-3210	Клиент (нет доступа)	Клиент	Редактировать
6.	client4	client4@client.com	(068) 654-1231	Клиент (нет доступа)	Клиент	Редактировать
7.	client5	client5@client.com	(012) 345-6765	Клиент (нет доступа)	Клиент	Редактировать
8.	client6	client6@client.com	(012) 345-6798	Клиент (нет доступа)	Клиент	Редактировать
9.	dexter	dada@mai.hjj	(063) 297-2761	Клиент (нет доступа)	Водитель	Редактировать
10.	Seller3	seller3@seller.com	(014) 785-2036	Администратор (полный)	Продавец	Редактировать

Рисунок 4.22 – Список пользователей

Магазин

Заказы Добавить заказ Учет Товар Меню ▾

Редактирование

Пожалуйста, введите изменения о пользователе client3:

[Вернуться](#)

email: [client3@client.com](#)

Должность

Клиент ▾

Уровень доступа

Клиент (нет доступа) ▾

Телефон

(987) 654-3210

[Внести изменения](#)

Рисунок 4.23 – Редактирование информации о пользователе

Для создания заказа администраторам или сотрудникам, необходимо перейти на страницу «Добавить заказ» и, указав дату и магазин, нажать кнопку

«Готово» (рис. 4.24-4.25). После этого, заказ будет отправлен. При этом если кнопка «Готово» нажата не будет (например, пользователь не точно определился с датой, пунктом доставки или содержанием заказа, ведь редактировать отправленные заказы может только администратор), заказ останется сформированным, однако будет отображаться лишь данному пользователю.

Магазин

ЗаказыДобавить заказУчетТоварМеню

Дата доставки

11-06-2016

Магазин

Магазин_2

Удалить

Подтверждение удаления

№	Название	Поставщик	Цена	Категория	Кол-во
Товара на сумму = 0 \$					
Весь товар					
1.	Product	Харьковская Бисквитн...	26.30	Шоколад 100 г...	Добавить
2.	product 2	newName	36.20	Cookies	Добавить
3.	Декстер	Roshen	26.05	Шоколад 200 г...	Добавить
4.	Еще товар	Provider 2	13.66	Cake	Добавить
5.	Молочный шоколад "Ал...	Roshen	12.35	Шоколад 100 г...	Добавить
6.	Печенье "Ромашка"...	Харьковская Бисквитн...	25.95	Cookies	Добавить
7.	Тестовый товар...	Тестовый поставщик 2...	66.13	Cake	Добавить
8.	Товар_1	Provider 2	9.10	Ice cream	Добавить
9.	Тофикс	Тестовый поставщик...	90.00	Sweets	Добавить

Рисунок 4.24 – Процесс формирования заказа. Инициализация заказа

Магазин

ЗаказыДобавить заказУчетТоварМеню

Главная

Добавить заказ

Пожалуйста, сформируйте заказ и нажмите кнопку "готово".

Дата доставки

11-06-2016

Магазин

Магазин_2

Готово

Удалить

Подтверждение удаления

№	Название	Поставщик	Цена	Категория	Кол-во	
Присутствует в заказе						
1.	Молочный шоколад "Ал...	Roshen	12.35	Шоколад 100 г...	1	Удалить
2.	Печенье "Ромашка"...	Харьковская Бисквитн...	25.95	Cookies	1	Удалить
3.	Харьковский вафельны...	Харьковская Бисквитн...	200.00	Cake	1	Удалить
Товара на сумму = 238.3 \$						
Весь товар						
1.	Product	Харьковская Бисквитн...	26.30	Шоколад 100 г...	Добавить	

Рисунок 4.25 – Процесс формирования заказа. Добавление товара

Завершив формирование заказа (на 11.06.2016), а нажав кнопку «Готово» заказ отображается в системе и помечается как «Поступивший заказ», другими словами – готов к выполнению. Нажав кнопку «Выполнить», администратор отправляет заказ на выполнение – в подсистему грузоперевозок. Выполняющиеся заказы можно лишь просматривать, но невозможно редактировать. Заказы в системе на 11.06.2016 представлены на рисунке 4.26.

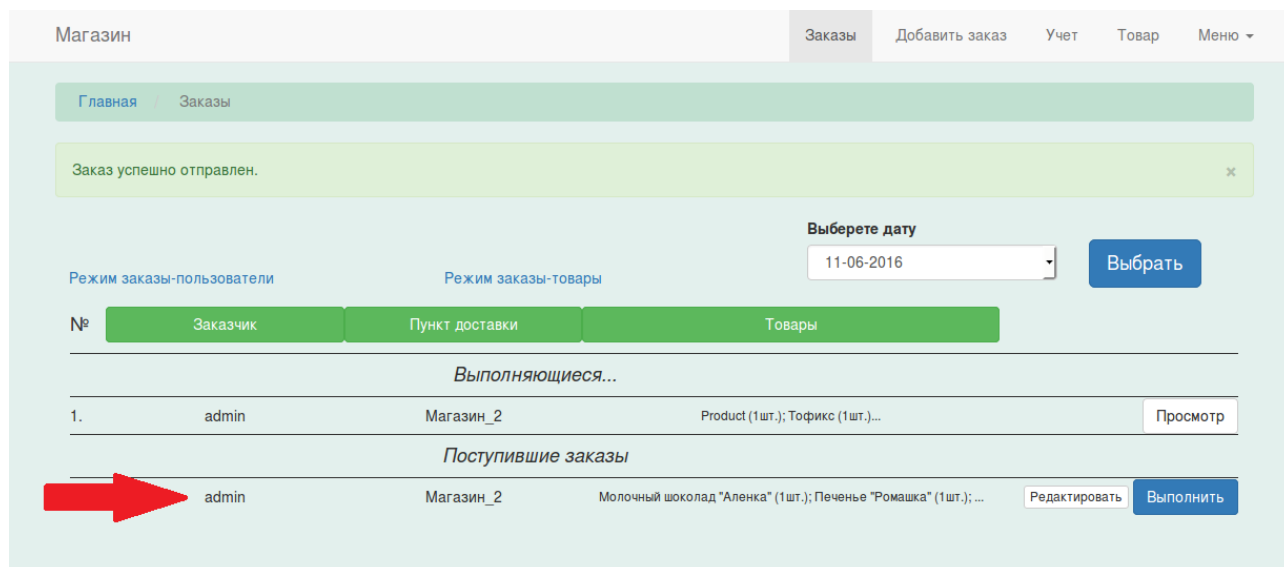


Рисунок 4.26 – Заказы на 11.06.2016

ВЫВОДЫ

В результате выполнения задания аттестационной работы разработана система Internet-продаж кондитерских изделий, позволяющая автоматизировать процессы: поиска товаров; формирования заказов; оформления продаж; сбора и обработки статистики продаж.

Система Internet-продажи кондитерских изделий отличается от аналогичных систем тем, что она объединяет в себе функции систем автоматизации продаж и систем Internet-коммерций, что позволяет:

- объединить всю информацию о товарах, заказах, поставщиках и пользователях (клиенты и сотрудники) организации в единой базе;
- пользоваться системой без необходимости установки дополнительных программных модулей или библиотек;
- использовать систему в любом удобном месте, как клиентам, так и сотрудникам организации.

За счет общей базы данных, разработанная система придает структурную целостность организации, а также позволяет руководителю или сотруднику быстро и удобно просматривать, анализировать и обрабатывать информацию, полученную от клиентов или других сотрудников.

Практическое применение разработанной системы позволяет сократить время реализации, а также поиска необходимого товара, как клиентам, так и сотрудникам фирмы, упростить и ускорить процесс формирования заказов. Используя разработанную систему, компании смогут повысить свою конкурентоспособность на рынке товаров и услуг.

Таким образом, данная разработка является актуальной, так как она способствует сокращению времени реализации и поиска товара, а также облегчению и ускорению процесса формирования заказов.

В дальнейшем, планируется разработка и введение подсистемы планирования процесса доставки товара, а также нахождения оптимального маршрута в заданных условиях. Это позволит минимизировать затраты на доставку товаров и выполнять заданные ограничения на время доставки товара.

По теме аттестационной работы подготовлен доклад на XX Юбилейный Международный молодежный форум «Радиоэлектроника и молодежь в XXI веке» [33].

ПЕРЕЧЕНЬ ССЫЛОК

1. Преимущества автоматизированной системы продаж [Электронный ресурс] – Режим доступа: [www / URL: http://business.damotvet.ru/sales-management/655957.htm](http://business.damotvet.ru/sales-management/655957.htm) – 05.05.2016 г. – Загл. с экрана.
2. Автоматизация отдела продаж, или как добиться взрывного роста [Электронный ресурс] / Москва – Режим доступа: [www / URL: http://class365.ru/uchet-tovarov-uslug/avtomatizatsiya-otdela-prodazh](http://class365.ru/uchet-tovarov-uslug/avtomatizatsiya-otdela-prodazh) – 06.05.2016 г. – Загл. с экрана.
3. Кондитерская фабрика «сладкоежка» [Электронный ресурс] / Краснодарский край – Режим доступа: [www / URL: http://sladkoezhka93.ru](http://sladkoezhka93.ru) – 19.05.2016 г. – Загл. с экрана.
4. «Комус» – гипермаркет для дома и офиса [Электронный ресурс] / Москва – Режим доступа: [www / URL: http://www.komus.ru/catalog/1031/](http://www.komus.ru/catalog/1031/) – 20.05.2016 г. – Загл. с экрана.
5. «Осьминог» – Интернет магазин [Электронный ресурс] / Харьков – Режим доступа: [www / URL: http://8nog.com.ua](http://8nog.com.ua) – 20.05.2016 г. – Загл. с экрана.
6. TADVISTER. Государство. Бизнес. IT [Электронный ресурс] / Москва – Режим доступа: [www / URL: http://www.tadviser.ru/index.php/](http://www.tadviser.ru/index.php/) Статья: Sales_Force_Automation_(SFA)_- _система_автоматизации_продаж – 21.05.2016 г. – Загл. с экрана.
7. Systematic software solutions [Электронный ресурс] / Москва – Режим доступа: [www / URL: http://www.netsuite.ru/ns_2.html](http://www.netsuite.ru/ns_2.html) – 23.05.2016 г. – Загл. с экрана.
8. Integration definition for function modeling (idef0) – Draft federal information processing standards publication, 1993 – 183 с.
9. Олищук А.В. Разработка WEB-приложений на PHP 5. Профессиональная работа / А.В. Олищук, А.Н. Чаплыгин. – М.: Вильямс, 2006. – 352 с.
10. Петров Э.Г. Территориально распределенные системы обслуживания / Э.Г. Петров, В.П. Пискалова, В.В. Бескоровайный. – Киев: Техника, 1992. – 208 с.
11. Бескоровайный В.В. Синтез логической схемы системного проектирования территориально распределенных объектов / В. В. Бескоровайный // Радиоэлектроника и информатика. – 2002. – №. 3. – С. 94 – 96.

12. Использование UML при проектировании Web-приложений [Электронный ресурс] – Режим доступа: [www / URL: http://alice.pnzgu.ru/~dvn/uproc/books/uml_web/index.htm](http://www.alice.pnzgu.ru/~dvn/uproc/books/uml_web/index.htm) – 25.05.2016 г. – Загл. с экрана.

13. Адам Фримен. ASP.NET MVC 4 с примерами на C# 5.0 для профессионалов, 4-е издание = Pro ASP.NET MVC 4, 4th edition. / А. Фримен – Москва: Вильямс, 2013. – 688 с.

14. Singh P. K. Database Management System Concepts / P. K. Singh – New Delhi, 2009 – 91 с.

15. Кузнецов С. Д. Основы баз данных, 2-е изд. / С. Д Кузнецов – Москва: Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2007. – 484 с.

16. Дейт К. Дж. Введение в системы баз данных = Introduction to Database Systems, 8-е изд. / К. Дж. Дейт – Москва: Вильямс, 2006. – 1328 с.

17. Кузнецов М. Р. MySQL 5 / М. Р. Кузнецов, И. В. Симдянов – Санкт-Петербург: БХВ-Петербург, 2010. – 1007 с.

18. Стив Макконнелл. Совершенный код. Практическое руководство по разработке программного обеспечения / С. Макконнелл – Москва: Русская Редакция – 896 с.

19. Мэтт Зандстра. PHP: объекты, шаблоны и методики программирования, 3-е издание = PHP Objects, Patterns and Practice, Third Edition. / М. Зандстра – Москва: «Вильямс», 2010. – 560 с.

20. What programming languages should I learn for web development? [Электронный ресурс] – Режим доступа: [www / URL: https://www.quora.com/What-programming-languages-should-I-learn-for-web-development/](http://www.quora.com/What-programming-languages-should-I-learn-for-web-development/) 23.05.2016 г. – Загл. с экрана.

21. Learn to code [Электронный ресурс] – Режим доступа: [www / URL: http://www.webdesigndegreecenter.org/learn-to-code/](http://www.webdesigndegreecenter.org/learn-to-code/) 22.05.2016 г. – Загл. с экрана.

22. Dave Thomas. Programming Ruby 1.9 & 2.0: The Pragmatic Programmers' Guide (The Facets of Ruby), 4th Edition / D. Thomas, Chad Fowler, Andy Hunt Raleigh: Pragmatic Bookshelf – 2013. – 888 с.

23. Фаронов В. В Создание приложений с помощью C# / В. В. Фаронов. – Москва: Эскимо, 2008. – 576 с.

24. Методы и технологии реинжиниринга ИС [Электронный ресурс] – Режим доступа: [www / URL: http://citforum.ru/SE/project/isr/](http://citforum.ru/SE/project/isr/) – 31.05.2016 г. – Загл. с экрана.
25. Горбунов-Посадов М. М. Расширяемые программы / М. М. Горбунов-Посадов. – Москва: Полиптих, 1999. – 336 с
26. Сафронов М. Разработка веб-приложений в Yii 2 / М. Сафронов. – Москва: ДМК Пресс – 2015. – 392 с.
27. Web Server Survey [Электронный ресурс] – Режим доступа: [www / URL: http://news.netcraft.com/archives/2011/08/05/august-2011-web-server-survey-3.html](http://news.netcraft.com/archives/2011/08/05/august-2011-web-server-survey-3.html) – 01.06.2016 г. – Загл. с экрана.
28. Тестируем ОС в качестве веб-сервера [Электронный ресурс] – Режим доступа: [www / URL: https://habrahabr.ru/post/165141/](https://habrahabr.ru/post/165141/) – 01.06.2016 г. – Загл. с экрана.
29. Как установить Linux, Apache, MySQL, PHP (LAMP) в Ubuntu 14.04 [Электронный ресурс] – Режим доступа: [www / URL: https://www.digitalocean.com/community/tutorials/linux-apache-mysql-php-lamp-ubuntu-14-04-ru](https://www.digitalocean.com/community/tutorials/linux-apache-mysql-php-lamp-ubuntu-14-04-ru) – 02.06.2016 г. – Загл. с экрана.
30. Элизабет Нарамор. Профессиональное программирование с LAMP (Linux, Apache, MySQL, PHP5) / Э. Нарамор, Д. Гернр – Оренбург: Алекс-Софт, 2006. – 448 с.
31. Версионная миграция структуры базы данных: основные подходы [Электронный ресурс] – Режим доступа: [www / URL: https://habrahabr.ru/post/121265/](https://habrahabr.ru/post/121265/) – Загл. с экрана.
32. Yii PHP framework: best for developing Web 2.0 applications [Электронный ресурс] – Режим доступа: [www / URL: http://www.yiiframework.com](http://www.yiiframework.com) – 02.06.2016 г. – Загл. с экрана.
33. Ахмад Ф. Х. Система электронной продажи кондитерских изделий / Ф. Х. Ахмад // 20-й Ювілейний Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті»: зб. матеріалів форуму. Т. 6. – Харків: ХНУ-РЕ. 2016 – с. 66–67.

Приложение А

Текст программы

УТВЕРЖДЕН

ГЮИК. 502520.001 - 01 12 01 ЛУ

Система Internet-продажи кондитерских изделий

Текст программы

ГЮИК 502520.001 - 01 12 01

ЛИСТОВ 19

«УТВЕРЖДАЮ»

Руководитель
аттестационной работы
проф. Бескоровайный В.В.

Система Internet-продажи кондитерских изделий

Текст программы

ЛИСТ УТВЕРЖДЕНИЯ

ГЮИК 502520.001 - 01 12 01 – ЛУ

РАЗРАБОТАЛ:

ст. гр. СИ-12-1

Ахмад Ф. Х.

backend/web/index.php

```
<?php
defined('YII_DEBUG') or define('YII_DEBUG', true);
defined('YII_ENV') or define('YII_ENV', 'dev');

require(__DIR__ . '/../vendor/autoload.php');
require(__DIR__ . '/../vendor/yiisoft/yii2/Yii.php');
require(__DIR__ . '/../common/config/bootstrap.php');
require(__DIR__ . '/../config/bootstrap.php');

$config = yii\helpers\ArrayHelper::merge(
    require(__DIR__ . '/../common/config/main.php'),
    require(__DIR__ . '/../common/config/main-local.php'),
    require(__DIR__ . '/../config/main.php'),
    require(__DIR__ . '/../config/main-local.php')
);

$application = new yii\web\Application($config);
$application->run();
```

backend/views/layouts/main.php

```
<?php

/* @var $this \yii\web\View */
/* @var $content string */

use backend\assets\AppAsset;
use yii\helpers\Html;
use yii\bootstrap\Nav;
use yii\bootstrap\NavBar;
use yii\widgets\Breadcrumbs;
use common\widgets\Alert;

AppAsset::register($this);
$this->registerCssFile('/css/site.css');
$this->registerCssFile('/css/main.css');
$this->registerCss('body {
```

```

background: #E3F0ED;}
li {
padding-left: 0.5em;
text-indent: 0em;}
li:before {
font-family: FontAwesome;
margin-right: 10px;}
.breadcrumb {
background: #C0E0D0;}
.panel-footer {
footer .container {
background-color:#fff;});
?>
<?php $this->beginPage() ?>
<!DOCTYPE html>
<html lang="<?= Yii::$app->language ?>">
<head>
    <meta charset="<?= Yii::$app->charset ?>">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <?= Html::csrfMetaTags() ?>
    <title><?= Html::encode($this->title) ?></title>
    <?php $this->head() ?>
</head>
<body>
<?php $this->beginBody() ?>

<div class="wrap">
    <?php
    NavBar::begin([
        'brandLabel' => 'Магазин',
        'brandUrl' => '../old_magaz/magaz/frontend/web',
        'options' => [
            'class' => 'navbar-default navbar-fixed-top',
        ],
    ]);
    if (Yii::$app->user->isGuest) {
        $menuItems[] = ['label' => 'Войти', 'url' => ['/site/login']];
    } else {
        $menuItems = [
            ['label' => 'Заказы', 'url' => ['/orders/list']],
            ['label' => 'Добавить заказ', 'url' => ['/orders/make_order']],

```



```

        ['label' => 'Учет', 'url' => ['/shops/products']],
        ['label' => 'Товар', 'url' => ['/product/list']],
    ];
    $menuItems[] = [
        'label' => ' Меню ',
        'items' => [
            '<li class="dropdown-header"></li>', [
                'label' => 'Поставщики',
                'url' => ['/providers/list'],
            ],
            '<li class="dropdown-header"></li>', [
                'label' => 'Пользователи',
                'url' => ['/user/list'],
            ],
            '<li class="dropdown-header"></li>', [
                'label' => 'Категории товаров',
                'url' => ['/product_categories/list'],
            ],
            '<li class="dropdown-header"></li>', [
                'label' => 'Магазины (помещения)',
                'url' => ['/shops/list'],
            ],
            '<li class="dropdown-header"></li>', [
                'label' => 'Должности сотрудников',
                'url' => ['/position/list'],
            ],
            '<li class="dropdown-header"></li>', [
                'label' => 'Выйти ( ' . Yii::$app->user->identity['name'] . ' )',
                'url' => ['/site/logout'],
                'linkOptions' => ['data-method' => 'post']
            ]
        ]
    ];
}

echo Nav::widget([
    'options' => ['class' => 'navbar-nav navbar-right'],
    'items' => $menuItems,
]);
NavBar::end();
?>
<div class="container">

```

```

<?= Breadcrumbs::widget([
    'links' => isset($this->params['breadcrumbs']) ? $this->params['breadcrumbs']: [],
]) ?>

<?= Alert::widget() ?>

<?= $content ?>

</div>
</div>

<div class="container">
    <p class="pull-right">My Company, <?= date('M Y') ?> <span class="glyphicon glyphicon-gift"></p>
</div>

<?php $this->endBody() ?>
</body>
</html>
<?php $this->endPage()

```

backend/views/product/list.php

```

<?php
/**
 * Created by PhpStorm.
 * User: blackghoul
 * Date: 14.05.16
 * Time: 16:22
 */

/* @var $this yii\web\View */
/* @var $form yii\bootstrap\ActiveForm */
/* @var $model \common\models\Product */

use yii\helpers\Html;
use yii\bootstrap\ActiveForm;
use common\models\MyQuery;

$this->title = 'Товар';
$this->params['breadcrumbs'][] = $this->title;

?>
<div class="product-list">

```

```

<h1><?= Html::encode($this->title) ?></h1>
<p></p>
<div class="row">
    <div>
        <?php
        isset($_POST['order']) ? $model->order = $_POST['order'] : $model->order = 'name'; ?>
        <table border="0" width="95%" align="center" rules="rows">
            <tr>
                <td><h4>№</h4></td>
                <td><h6><?php ActiveForm::begin();
                    if ($model->order == 'name ') {
                        $val = 'name desc';
                        $type = '<span class="glyphicon glyphicon-sort-by-alphabet">';
                    } elseif ($model->order == 'name desc') {
                        $val = 'name ';
                        $type = '<span class="glyphicon glyphicon-sort-by-alphabet-alt">';
                    } else {
                        $val = 'name ';
                        $type = "";
                    } ?>
                    <?= Html::submitButton('Название ' . $type,
                        ['class' => 'btn btn-block btn-success',
                         'name' => 'order', 'value' => $val]) ?>
                </h6></td>
                <td><h6>
                    <?php ActiveForm::begin();
                    if ($model->order == 'provider') {
                        $val = 'provider desc';
                        $type = '<span class="glyphicon glyphicon-sort-by-alphabet">';
                    } elseif ($model->order == 'provider desc') {
                        $val = 'provider';
                        $type = '<span class="glyphicon glyphicon-sort-by-alphabet-alt">';
                    } else {
                        $val = 'provider';
                        $type = "";
                    } ?>
                    <?= Html::submitButton('Поставщик ' . $type,
                        ['class' => 'btn btn-block btn-success',
                         'name' => 'order', 'value' => $val]) ?>
                </h6></td>
            </tr>
        </table>
    </div>
</div>

```

```

<?php ActiveForm::begin();
if ($model->order == 'price') {
    $val = 'price desc';
    $type = '<span class="glyphicon glyphicon-sort-by-order">';
} elseif ($model->order == 'price desc') {
    $val = 'price';
    $type = '<span class="glyphicon glyphicon-sort-by-order-alt">';
} else {
    $val = 'price';
    $type = "";
} ?>

<?= Html::submitButton('Цена ' . $type,
    ['class' => 'btn btn-block btn-success',
    'name' => 'order', 'value' => $val]) ?>
</h6></td>
<td><h6>
    <?php ActiveForm::begin();
    if ($model->order == 'description') {
        $val = 'description desc';
        $type = '<span class="glyphicon glyphicon-sort-by-alphabet">';
    } elseif ($model->order == 'description desc') {
        $val = 'description';
        $type = '<span class="glyphicon glyphicon-sort-by-alphabet-alt">';
    }
    else{
        $val = 'description';
        $type = "";
    } ?>

    <?= Html::submitButton('Описание ' . $type,
        ['class' => 'btn btn-block btn-success',
        'name' => 'order', 'value' => $val]) ?>

    </h6></td>
<td><h6>
    <?php ActiveForm::begin();
    if ($model->order == 'category') {
        $val = 'category desc';
        $type = '<span class="glyphicon glyphicon-sort-by-alphabet">';
    } elseif ($model->order == 'category desc'){

```

```

        $val = 'category';
        $type = '<span class="glyphicon glyphicon-sort-by-alphabet-alt">';
    }else{
        $val = 'category';
        $type = "";
    } ?>

    <?= Html::submitButton('Категория ' . $type,
        ['class' => 'btn btn-block btn-success',
        'name' => 'order', 'value' => $val]) ?>

    </h6></td>
<td><h4>
    <?php ActiveForm::begin();
    if ($model->order == 'popularity desc') {
        $val = 'popularity';
        $type = '<span class="glyphicon glyphicon-sort-by-attributes-alt">';
    } elseif ($model->order == 'popularity') {
        $val = 'popularity ';
        $type = '<span class="glyphicon glyphicon-sort-by-attributes">';
    } else {
        $val = 'popularity desc';
        $type = "";
    } ?>

    <?= Html::submitButton('Попул. ' . $type,
        ['class' => 'btn btn-block btn-success',
        'name' => 'order', 'value' => $val]) ?>

    </h4></td>
<td><h6><?php ActiveForm::begin();
    if ($model->order == 'selling desc') {
        $val = 'selling';
        $type = '<span class="glyphicon glyphicon-sort-by-attributes-alt">';
    } elseif ($model->order == 'selling') {
        $val = 'selling ';
        $type = '<span class="glyphicon glyphicon-sort-by-attributes">';
    } else {
        $val = 'selling desc';
        $type = "";
    } ?>

```

```

        <?= Html::submitButton('B I-мар. ' . $type,
            ['class' => 'btn btn-block btn-success',
             'name' => 'order', 'value' => $val]) ?></h6></td>

<td><h4></h4></td>
<?php ActiveForm::end();
$rows = MyQuery::selectProduct($model->order) ?>
</tr>
<?php $i = 0;
foreach ($rows as $value):
    mb_strlen($value['name'], 'UTF-8') > 20 ?
        $value['name'] = mb_substr($value['name'], 0, 20, 'UTF-8') . '...' :
        $value['name'];
    mb_strlen($value['provider'], 'UTF-8') > 20 ?
        $value['provider'] = mb_substr($value['provider'], 0, 20, 'UTF-8') . '...' :
        $value['provider'];
    mb_strlen($value['description'], 'UTF-8') > 20 ?
        $value['description'] = mb_substr($value['description'], 0, 26, 'UTF-8') . '...' :
        $value['description'];
    mb_strlen($value['category'], 'UTF-8') > 20 ?
        $value['category'] = mb_substr($value['category'], 0, 20, 'UTF-8') . '...' :
        $value['category']; ?>
<tr>
    <?php echo '<td width="3%"><h5>' . ++$i . '</h5></td>';
    echo '<td width="17%"><h5>' . $value['name'] . '</h5></td>';
    echo '<td width="16%"><h5>' . $value['provider'] . '</h5></td>';
    echo '<td width="9%"><h5>' . number_format($value['price'], 2) . '</h5></td>';
    echo '<td width="17%"><h6>' . $value['description'] . '</h6></td align="center">';
    echo '<td width="16%"><h5>' . $value['category'] . '</h5></td>';
    echo '<td width="9%" align="center"><h5>' . $value['popularity'] . '</h5></td>';
    if ($value['selling'])
        echo '<td width="8%" align="center"><span class="glyphicon glyphicon-ok"></td>';
    else
        echo '<td width="8%" align="center"><span class="glyphicon glyphicon-remove"></td>';
    echo '<td width="8 % ">';
    ActiveForm::begin(['id' => 'id']) ?>
    <?= Html::submitButton('Редактировать', ['class' => 'btn btn-default btn-sm',
        'name' => 'id', 'value' => $value['id']]); ?>
    <?php ActiveForm::end();
    echo '</td>' ?>

```

```

        </tr>
        <?php endforeach; ?>
    </table>
    <?php
    ActiveForm::begin(['id' => 'id']); ?>
    <br>
    <?= Html::submitButton('Добавить', ['class' => 'btn btn-primary btn-lg btn-block',
        'name' => 'add', 'value' => '1']); ?>
    <?php ActiveForm::end(); ?>
    <br>
</div>
</div>
</div>

```

backend/controllers/ProductController.php

```

<?php

namespace backend\controllers;

use common\models\Product;
use common\models\ProductForm;
use Yii;
use yii\base\InvalidParamException;
use yii\web\BadRequestHttpException;
use yii\web\Controller;
use yii\filters\VerbFilter;
use yii\filters\AccessControl;
use yii\web\UploadedFile;

/**
 * Product controller
 */
class ProductController extends BehaviorsController
{
    /**
     * @return mixed
     */

    public function actionList()
    {
        if (Yii::$app->user->isGuest)

```

```

        return $this->redirect('../site/login.html');
$model = new Product();
if (isset($_GET['delete']) && $_GET['delete'] == 1)
    Yii::$app->session->setFlash('success', 'Удаление произведено успешно.');
```

```

if (isset($_GET['added']) && $_GET['added'] == 1)
    Yii::$app->session->setFlash('success', 'Новый товар успешно добавлен.');
```

```

if (isset($_GET['deleted']))
    Yii::$app->session->setFlash('error', 'Такого товара не существует. Или, возможно, он удален.');
```

```

if (!isset($_POST['id']) && !isset($_POST['add']))
    return $this->render('list', [
        'model' => $model
    ]);
else {
    if (isset($_POST['id'])) {
        $model->id = $_POST['id'];
        return $this->redirect(['edit_product',
            'id' => $model->id,
        ]);
    }
    if (isset($_POST['add'])) {
        return $this->redirect(['add_product']);
    }
}
}

/**
 * @return mixed
 */

public function actionEdit_product()
{
    if (Yii::$app->user->isGuest)
        return $this->redirect('../site/login.html');
    if (!isset($_GET['id']))
        return $this->redirect('list');
    $model = ($model = Product::findOne($_GET['id'])) ? $model : new Product();
    if ($model->status == $model::STATUS_DELETED) {
        return $this->redirect(['list', 'deleted' => '1']);
    }
    if ($model->load(Yii::$app->request->post()) && isset($_POST['del']) && $_POST['del'] == 1) {
        if ($model->load(Yii::$app->request->post()) && $model->delete_accept) {

```



```

        if ($model->deleteProduct())
            return $this->redirect(['list', 'delete' => '1']);
    } else
        Yii::$app->session->setFlash('error', 'Подтвердите удаление!');
    }
}

if ($model->load(Yii::$app->request->post()) && isset($_POST['edit_product-button'])) {
    if ($model->validate()) {
        $model->file = UploadedFile::getInstance($model, 'file');
        if (isset($model->file)) {
            $model->file->saveAs('uploads/' . $model->id .
                str_replace("\", ", (str_replace("'", ", str_replace(' ', '_ ', $model->name)))) .
                ' ' . $model->file->extension);

            $model->image = 'uploads/' . $model->id .
                str_replace("\", ", (str_replace("'", ", str_replace(' ', '_ ', $model->name)))) .
                ' ' . $model->file->extension;
        }
        if ($model->validate()) {
            if ($model->updateProduct($model)) {
                Yii::$app->session->setFlash('success', 'Редактирование произведено успешно.');
```

return \$this->render('edit_product', ['model' => \$model]);

```

            } else
                Yii::$app->session->setFlash('error', 'Не удалось внести изменения!
                <br>Пожалуйста, перепроверьте заполненные поля и попробуйте снова.');
```

}

```

        }
    }
}

if ($model->load(Yii::$app->request->post()) && isset($_POST['delete-image-button'])) {
    if ($model->load(Yii::$app->request->post()) && $model->delete_image) {
        if ($model->validate()) {
            if ($model->deleteImage()) {
                Yii::$app->session->setFlash('success', 'Редактирование произведено успешно.');
```

return \$this->render('edit_product', ['model' => \$model]);

```

            }
        }
    } else
        Yii::$app->session->setFlash('error', 'Подтвердите удаление изображения!');
    }
}

if (isset($_POST['back'])) {
    return $this->redirect(['list']);
}

return $this->render('edit_product', [

```

```

        'model' => $model
    );
}

/**
 * @return mixed
 */

public function actionAdd_product()
{
    if (Yii::$app->user->isGuest)
        return $this->redirect('../site/login.html');
    $model = new ProductForm();
    if ($model->load(Yii::$app->request->post())) {
        $model->file = UploadedFile::getInstance($model, 'file');
        if (isset($model->file)) {
            $model->file->saveAs('uploads/' . $model->id .
                str_replace("\", ", (str_replace("'", ", str_replace(' ', '_ ', $model->name)))) .
                ' ' . $model->file->extension);
            $model->image = 'uploads/' . $model->id .
                str_replace("\", ", (str_replace("'", ", str_replace(' ', '_ ', $model->name)))) .
                ' ' . $model->file->extension;
        }
        if ($model->validate()) {
            if ($product = $model->addProduct()) {
                return $this->redirect(['list', 'added' => '1']);
            } else
                Yii::$app->session->setFlash('error', 'Не удалось внести изменения!
                <br>Пожалуйста, перепроверьте заполненные поля и попробуйте снова.');
```

common/models/Product.php

```
<?php
/**
 * Created by PhpStorm.
 * User: blackghoul
 * Date: 14.05.16
 * Time: 16:41
 */

namespace common\models;

use Yii;
use yii\base\NotSupportedException;
use yii\behaviors\TimestampBehavior;
use yii\db\ActiveRecord;
use yii\db\Query;

/**
 * Product model
 *
 * @property integer $id
 * @property string $name
 * @property integer $provider_id
 * @property integer $product_category_id
 * @property boolean $selling
 * @property float $price
 * @property string $description
 * @property string $image
 * @property integer $created_at
 * @property integer $updated_at
 * @property integer $status
 * @property integer $popularity
 */
class Product extends ActiveRecord
{
    const STATUS_DELETED = 0;
    const STATUS_ACTIVE = 10;

    public $delete_accept;
    public $delete_image;
```

```
public $file;
public $priceGrn;
public $priceKop;
public $order;

/**
 * @inheritdoc
 */
public static function tableName()
{
    return 'product';
}

public function attributeLabels()
{
    return [
        'name' => 'Наименование',
        'provider_id' => 'Поставщик',
        'product_category_id' => 'Категория',
        'selling' => 'Отображать в магазине',
        'price' => 'Цена',
        'description' => 'Описание',
        'file' => 'Картинка',
        'delete_аccept' => 'Подтверждение удаления',
        'delete_image' => 'Удаление изображения',
        'priceGrn' => 'грн.',
        'priceKop' => 'коп.',
    ];
}

/**
 * @inheritdoc
 */
public function behaviors()
{
    return [
        TimestampBehavior::className(),
    ];
}

/**
```

```

* @inheritdoc
*/
public function rules()
{
    return [
        [['name', 'description', 'price', 'image'], 'trim'],
        [['name', 'provider_id', 'product_category_id'], 'required'],
        ['name', 'unique', 'targetClass' => '\common\models\Product',
            'message' => 'Уже занято такое название товара.'],
        ['name', 'string', 'min' => 4, 'max' => 32],
        [['description', 'image'], 'string', 'min' => 0, 'max' => 256],
        ['selling', 'in', 'range' => [0, 1]],
        ['selling', 'default', 'value' => '0'],
        [['file'], 'file'],
        ['priceKop', 'string', 'min' => 0, 'max' => 2],
        [['priceKop', 'priceGrn'], 'default', 'value' => '0'],
        ['order', 'string'],
        [['delete_accept', 'delete_image'], 'in', 'range' => [0, 1]],
        ['status', 'default', 'value' => self::STATUS_ACTIVE],
        ['status', 'in', 'range' => [self::STATUS_ACTIVE, self::STATUS_DELETED]],
    ];
}

/**
 * @inheritdoc
 */
public static function findIdentity($id)
{
    return static::findOne(['id' => $id, 'status' => self::STATUS_ACTIVE]);
}

/**
 * Finds product by name
 *
 * @param string $name
 * @return static|null
 */
public static function findByProductName($name)
{
    return static::findOne(['name' => $name, 'status' => self::STATUS_ACTIVE]);
}

```

```
/**
 * @inheritdoc
 */
public function getId()
{
    return $this->getPrimaryKey();
}

/**
 * update product.
 *
 * @return Product|null the saved model or null if saving fails
 */
public function updateProduct($product)
{
    if ($product->priceGrn < 0 || $product->priceKop < 0)
        return false;
    $this->name = $product->name;
    $this->provider_id = $product->provider_id;
    $this->selling = $product->selling;
    $this->price = $product->priceGrn + $product->priceKop / 100;
    $this->description = $product->description;
    $this->image = $product->image;
    $this->product_category_id = $product->product_category_id;
    return $this->save() ? true : false;
}

public function deleteProduct()
{
    if (strlen($this->name) >= 7)
        $this->name = substr($this->name, 0, 7);
    $this->name = $this->name . '_deleted_' . date('D' . 'M' . 'Y') . rand(0, 9999);
    $this->status = self::STATUS_DELETED;
    return $this->save() ? true : false;
}

public function updateProvider($new_provider_id)
{

```

```
$this->provider_id = $new_provider_id;
return $this->save() ? true : false;
}

public function updateCategory($new_category_id)
{
    $this->product_category_id = $new_category_id;
    return $this->save() ? true : false;
}

public function deleteImage()
{
    if (isset($this->image) && $this->image != 'uploads/no_image.png')
        unlink($this->image);
    $this->image = 'uploads/no_image.png';
    return $this->save() ? true : false;
}

public function getPopularity()
{
    return $this->popularity;
}

public function incrementPopularity()
{
    $this->popularity = $this->getPopularity() + 1;
    return $this->save() ? true : false;
}

public static function updatePopularity($id)
{
    return self::findIdentity($id)->incrementPopularity() ? true : false;
}
}
```

Приложение Б

Руководство пользователя

УТВЕРЖДЕН

ГЮИК.502520.001 ИЗ – ЛУ

Система Internet-продажи кондитерских изделий

Руководство пользователя

ГЮИК.502520.001 ИЗ

ЛИСТОВ 17

Харьковский национальный университет радиоэлектроники

«УТВЕРЖДАЮ»

Руководитель
аттестационной
работы
проф. Бескоровайный В.В.

Система Internet-продажи кондитерских изделий

Руководство пользователя

ЛИСТ УТВЕРЖДЕНИЯ

ГЮИК.502520.001 ИЗ – ЛУ

РАЗРАБОТАЛ:

ст. гр. СИ-12-1

Ахмад Ф. Х.

СОДЕРЖАНИЕ

Введение	4
1 Назначение и условия применения	4
2 Подготовка к работе.....	5
3 Описание операций	7
3.1 Администраторская (backend) часть	7
3.2 Клиентская (frontend) часть	11
4 Аварийные ситуации.....	13
5 Контрольный пример оформления заказа	15

ВВЕДЕНИЕ

Данная программная система применяется для автоматизации поиска товаров, формирования заказов и оформления продаж, а также сбора и обработки статистики продаж.

В возможности системы входит:

- регистрация и авторизация пользователей системы (администраторы, сотрудники, клиенты); данная функция используется для идентификации пользователей и ведения личного учета совершенных заказов и купленных товаров;

- добавление, редактирование и удаление в базе данных информации (в администраторской части приложения):

- 1) о товарах, а также наличии их в магазинах или складах;
- 2) о сотрудниках (контактные данные, должности и т. п.);
- 3) о клиентах (контактные данные, совершенные заказы);
- 4) о поставщиках товара;
- 5) о заказах;

Программный комплекс не требует установок и доступен каждому пользователю ПК или мобильного устройства, владеющий Internet.

1 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

Программное приложение позволяет сотрудникам организации и клиентам просматривать или редактировать (для пользователей с соответствующим уровнем доступа, администраторам) информацию о товарах, поставщиках товара, заказах, а также сотрудниках и клиентах (администраторам).

Разработанная система функционирует посредством браузера. Для этого рекомендуется использовать персональный компьютер, имеющий доступ к Internet, со следующими характеристиками:

- 512 Мб ОЗУ;
- операционная система Windows XP, Vista, 7, 8 или 10, Ubuntu 10.04 +, Debian 6 +, OpenSuSE 11.3 +, Mac OS X 10.6 или более поздней версии либо Fedora Linux 14;
- процессор Intel Pentium 4 / Athlon 64 или более поздней версии.

Система Internet-продажи кондитерских изделий также доступна на мобильных устройствах, которые имеют доступ к Internet и поддерживают браузер.

2 ПОДГОТОВКА К РАБОТЕ

Разработанное программное обеспечение доступно любому пользователю, имеющему доступ к Internet, а также не требует предварительной установки и не поставляется с дистрибутивным носителем данных.

Для того чтобы получить возможность использовать систему, необходимо установить на персональном компьютере или мобильном устройстве браузер, а также произвести настройки доступа к Internet.

Для регистрации в системе, необходимо перейти на страницу регистрации, выбрав на навигационной панели «Регистрация» (в клиентской части WEB-приложения), и ввести запрашиваемые приложением данные (рис. 2.1).

My Company

Регистрация

Для регистрации, заполните, пожалуйста, следующие поля:

Логин

Электронная почта

Телефон

Пароль

Повторите пароль

[Регистрация](#)

Рисунок 2.1 – Страница регистрации

Зарегистрированные пользователи не имеют доступа к администраторской части приложения. Для назначения пользователю определенного уровня доступа, администратор должен перейти к редактированию пользователей (страница «Пользователи» в администраторской части) и присвоить соот-

ветствующий уровень доступа необходимому пользователю (рис. 2.2). Стоит отметить, что в системе изначально присутствует один администратор и три уровня доступа пользователей – «клиент (нет доступа)», «сотрудник (чтение)» и «администратор (полный)».

Рисунок 2.2 – Изменение уровня доступа зарегистрированного пользователя

Для перехода на другие страницы, в верхней части приложения представлена навигационная панель, в элементах которой описаны ссылки для перехода на страницы приложения, такие как «Заказы», «Добавить заказ», «Товары», «Поставщики» и пр. (рис 2.3).

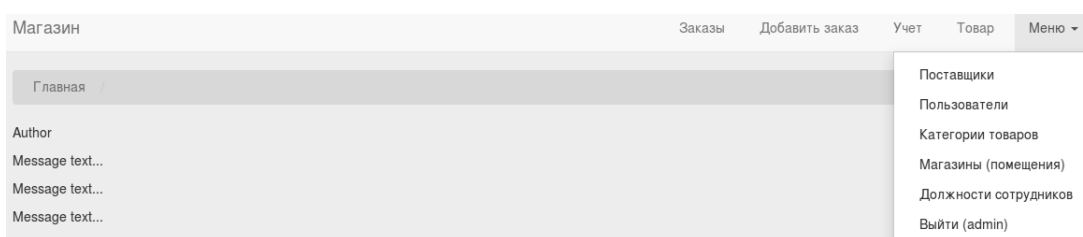
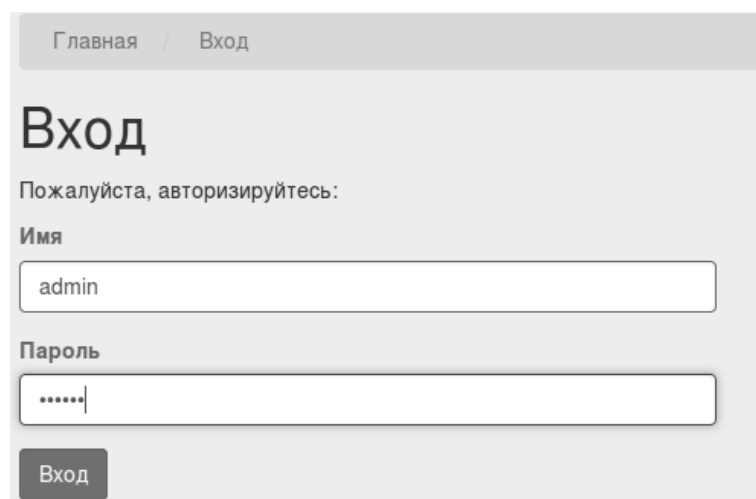


Рисунок 2.3 – Панель навигации

3 ОПИСАНИЕ ОПЕРАЦИЙ

3.1 Администраторская (backend) часть

Выполнив вход в систему (рис. 3.1), пользователю отображается главная страница, а также навигационная панель. Выбрав в меню панели, например, пункт «Поставщик», приложение перенаправит пользователя на страницу со списком поставщиков (рис. 3.2).



Главная / Вход

Вход

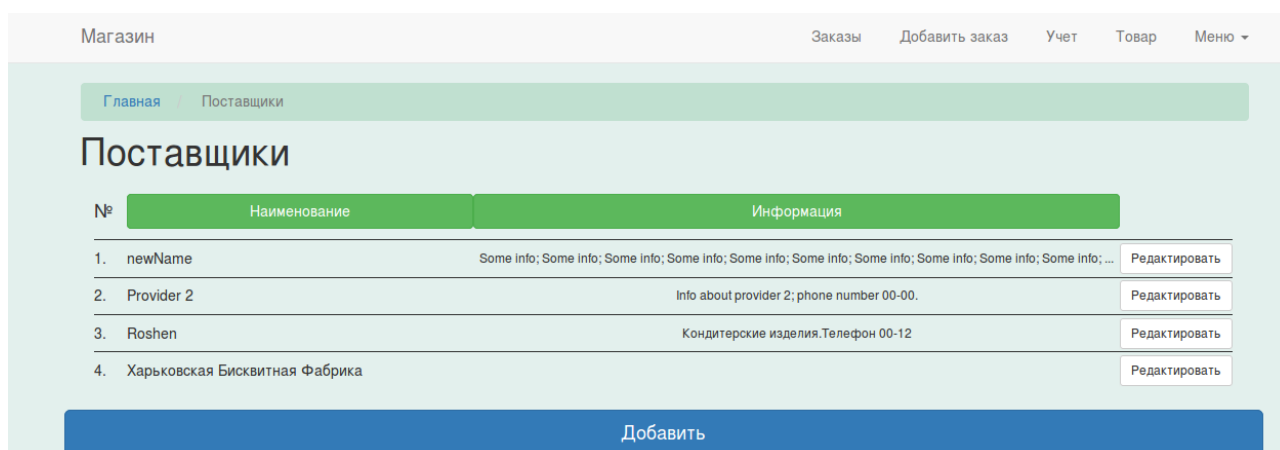
Пожалуйста, авторизуйтесь:

Имя

Пароль

Вход

Рисунок 3.1 – Вход в систему



Магазин

Заказы Добавить заказ Учет Товар Меню

Главная / Поставщики

Поставщики

№	Наименование	Информация	
1.	newName	Some info; Some info; Some info; Some info; Some info; Some info; Some info; Some info; Some info; Some info; ...	Редактировать
2.	Provider 2	Info about provider 2; phone number 00-00.	Редактировать
3.	Roshen	Кондитерские изделия.Телефон 00-12	Редактировать
4.	Харьковская Бисквитная Фабрика		Редактировать

Добавить

Рисунок 3.2 – Страница «Поставщики»

Администратор имеет возможность добавлять, редактировать или удалять информацию о поставщиках. Процесс редактирования информации представлен на рисунке 3.3. Если отредактированная информация успешно заноситься в

базу данных, пользователь будет перенаправлен на страницу списка поставщиков и оповещен об успешно проведенном редактировании (рис. 3.4). Подобным образом производится работа и на других страницах WEB-приложения, таких как: «Товар», «Пользователи», «Должности сотрудников», «Магазины» и «Категории товара».

Редактирование

Пожалуйста, введите изменения о поставщике:

[Вернуться](#)

Наименование

Харьковская Бисквитная Фабрика

Информация о поставщике

Информация, тел. 1020-3065

[Внести изменения](#)

☐ Подтверждение удаления

[Удалить](#)

Рисунок 3.3 – Редактирование информации о поставщике

Главная / Поставщики

Редактирование произведено успешно.

Поставщики

№	Наименование	Информация	
1.	newName	Some info; Some info; Some info; Some info; Some info; Some info; Some info; Some info; Some info; Some info; ...	Редактировать
2.	Provider 2	Info about provider 2; phone number 00-00.	Редактировать
3.	Roshen	Кондитерские изделия. Телефон 00-12	Редактировать
4.	Харьковская Бисквитная Фабрика	Информация, тел. 1020-3065	Редактировать

[Добавить](#)

Рисунок 3.4 – Результаты успешно заверщенного редактирования

В системе присутствует возможность вести учет товара. Для этого необходимо перейти на страницу «Учет» (рис. 3.5).

Выбрав из списка магазинов (складов) необходимый, приложение отоб-

ражает список всего товара, а также список товара, имеющегося в выбранном магазине или складе, и предоставляет возможность его редактирования (рис. 3.6-3.7).

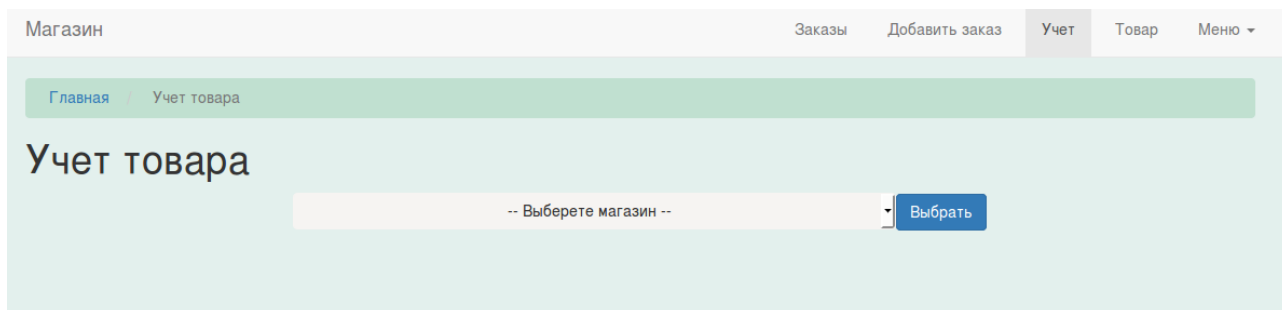


Рисунок 3.5 – Страница ведения учета товара

№	Название	Поставщик	Цена	Категория	Кол-во
Товара на сумму = 0 \$					
Отсутствует в Магазин_1					
1.	Product	newName	26.30	Шоколад 100 г...	Добавить
2.	product 2	newName	36.20	Cookies	Добавить
3.	Декстер	Roshen	26.05	Шоколад 200 г...	Добавить
4.	Еще товар	Provider 2	13.66	Cake	Добавить
5.	Молочный шоколад "Ал...	Roshen	12.35	Шоколад 100 г...	Добавить
6.	Печенье "Ромашка"	Харьковская Кондитерская	25.95	Cookies	Добавить

Рисунок 3.6 – Список товара в «Магазин_1»

№	Название	Поставщик	Цена	Категория	Кол-во	
Имеется в Магазин_1						
1.	Product	newName	26.30	Шоколад 100 г...	5	Удалить - 5
2.	product 2	newName	36.20	Cookies	4	Удалить
3.	Еще товар	Provider 2	13.66	Cake	4	Удалить
4.	Декстер	Roshen	26.05	Шоколад 200 г...	1	Удалить
Товара на сумму = 356.9\$						
Весь товар						
1.	Product	newName	26.30	Шоколад 100 г...		Добавить

Рисунок 3.7 – Редактирование списка товара в «Магазин_1»

Для создания, заказа администраторам или сотрудникам, необходимо перейти на страницу «Добавить заказ» и, указав дату и магазин, нажать кнопку «Готово» (рис. 3.8-3.9). После этого, заказ будет отправлен. При этом если

кнопка «Готово» нажата не будет (например, пользователь не точно определился с датой, пунктом доставки или содержанием заказа, т. к. редактировать отправленные заказы может только администратор), заказ останется сформированным, однако будет отображаться лишь данному пользователю.

Магазин

ЗаказыДобавить заказУчетТоварМеню

Дата доставки

11-06-2016

Магазин

Магазин_2

Удалить

☐ Подтверждение удаления

№

Название

Поставщик

Цена

Категория

Кол-во

Товара на сумму = 0 \$

Весь товар

1.

Product

Харьковская Бисквитн...

26.30

Шоколад 100 г...

Добавить

2.

product 2

newName

36.20

Cookies

Добавить

3.

Декстер

Roshen

26.05

Шоколад 200 г...

Добавить

4.

Еще товар

Provider 2

13.66

Cake

Добавить

5.

Молочный шоколад "Ал...

Roshen

12.35

Шоколад 100 г...

Добавить

6.

Печенье "Ромашка"...

Харьковская Бисквитн...

25.95

Cookies

Добавить

7.

Тестовый товар...

Тестовый поставщик 2...

66.13

Cake

Добавить

8.

Товар_1

Provider 2

9.10

Ice cream

Добавить

9.

Тофикс

Тестовый поставщик...

90.00

Sweets

Добавить

Рисунок 3.8 – Процесс формирования заказа. Инициализация заказа

Магазин

ЗаказыДобавить заказУчетТоварМеню

Главная

Добавить заказ

Пожалуйста, сформируйте заказ и нажмите кнопку "готово".

Дата доставки

11-06-2016

Магазин

Магазин_2

Готово

Удалить

☐ Подтверждение удаления

№

Название

Поставщик

Цена

Категория

Кол-во

Присутствует в заказе

1.

Молочный шоколад "Ал...

Roshen

12.35

Шоколад 100 г...

1

Удалить

2.

Печенье "Ромашка"...

Харьковская Бисквитн...

25.95

Cookies

1

Удалить

3.

Харьковский вафельны...

Харьковская Бисквитн...

200.00

Cake

1

Удалить

Товара на сумму = 238.3 \$

Весь товар

1.

Product

Харьковская Бисквитн...

26.30

Шоколад 100 г...

Добавить

Рисунок 3.9 – Процесс формирования заказа. Добавление товара

После формирования заказа и нажатия кнопки «Готово» заказ отображается в системе, отмечается как «Поступившие заказы» и считается готовым к выполнению. Нажав кнопку «Выполнить», администратор отправляет заказ на

выполнение – в подсистему грузоперевозок.

Выполняющиеся заказы можно лишь просматривать, но невозможно редактировать. Заказы в системе группируются по дате, заказы на 11.06.2016 представлены на рисунке 3.10.

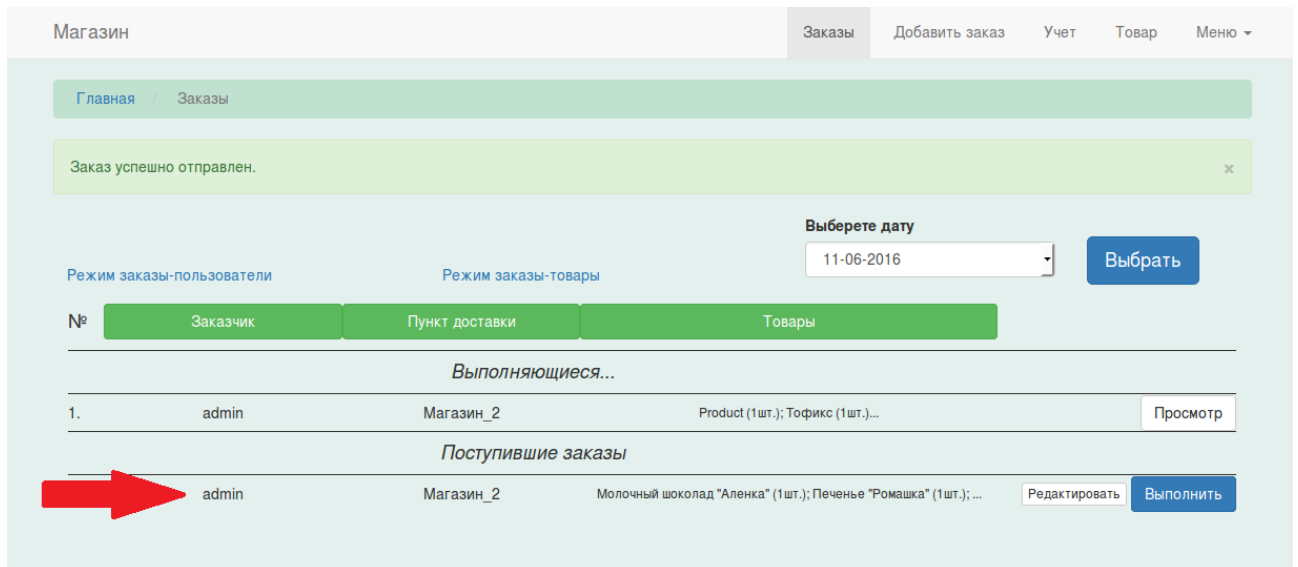


Рисунок 3.10 – Заказы на 11.06.2016

3.2 Клиентская (frontend) часть

В клиентской части приложения главной страницей является страница со списком товара, представленного клиенту (рис. 3.11).

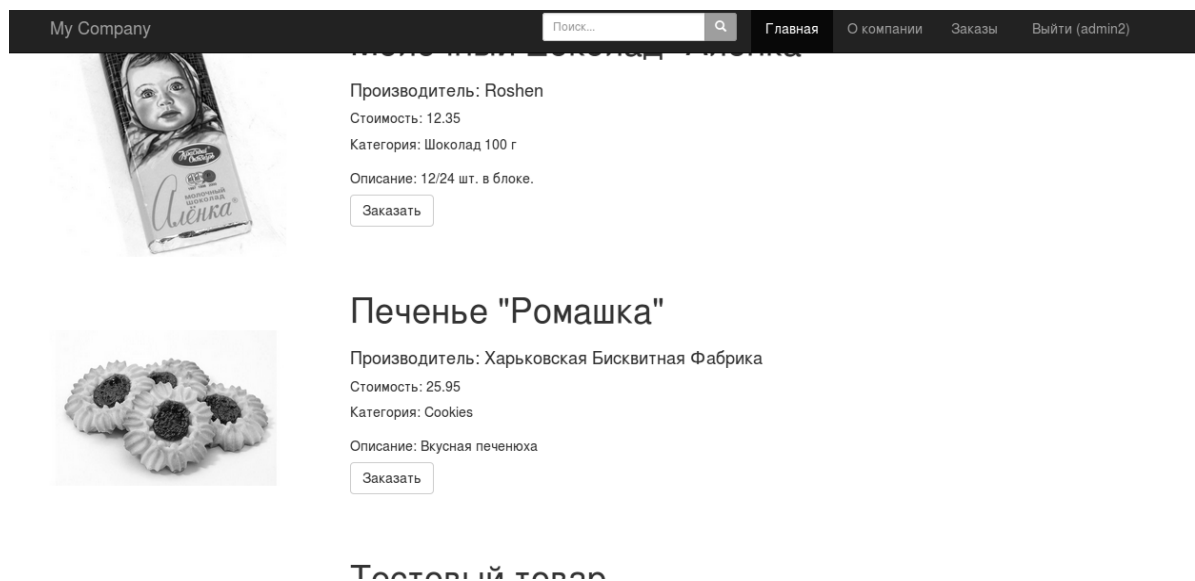


Рисунок 3.11 – Клиентская часть приложения (главная страница)

В основном, операция оформления заказа не отличается от подобной операции в администраторской части приложения, за исключением того, что клиент может просматривать только созданные им заказы. Стоит заметить, что в клиентской части многие функции ограничены.

На странице главной странице реализованы операции фильтрации, сортировки и поиска товара. Например, отправив запрос, введенный в строку поиска, приложение отобразит пользователю товары, в названиях, описаниях или других полях которых встретиться значение данного запроса (рис. 3.12).

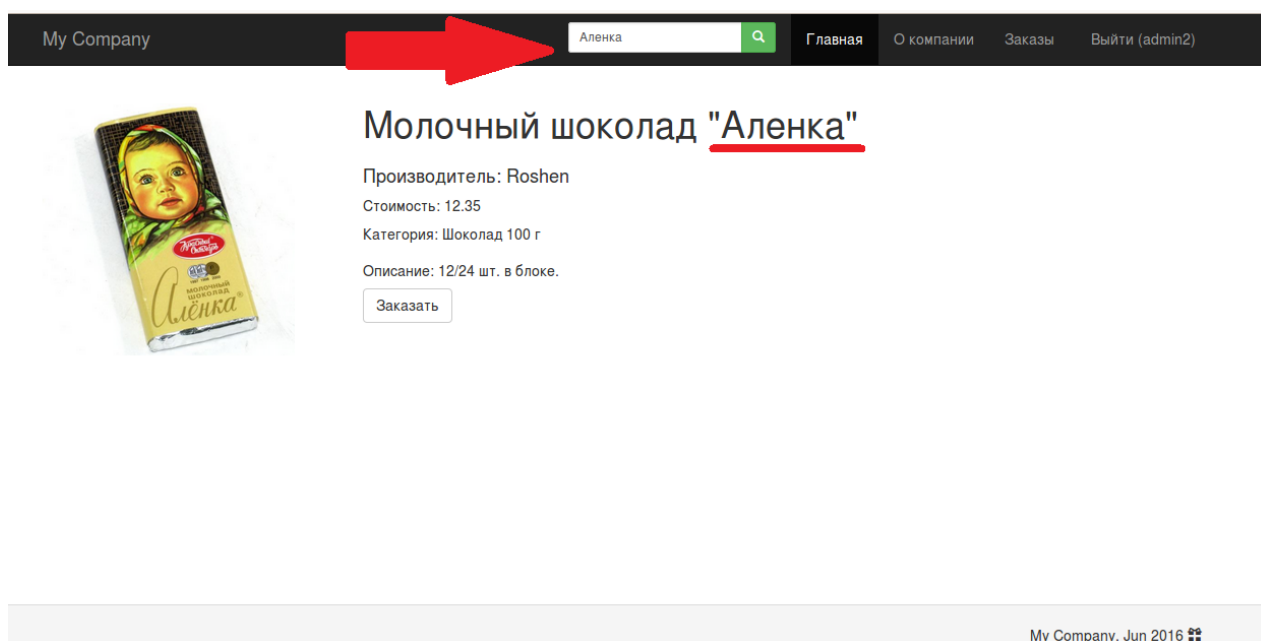


Рисунок 3.12 – Результаты поиска по запросу «Аленка»

В случае если товар по запросу поиска не будет найден, пользователь получит оповещение, представленное на рисунке 3.13.

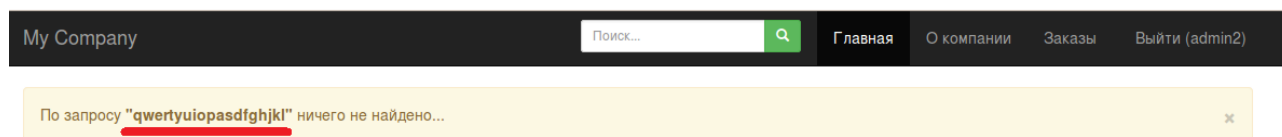


Рисунок 3.12 – Результаты поиска по запросу «Аленка»

4 АВАРИЙНЫЕ СИТУАЦИИ

При переходе на несуществующую страницу пользователю отображается страница ошибки (рис. 4.1).

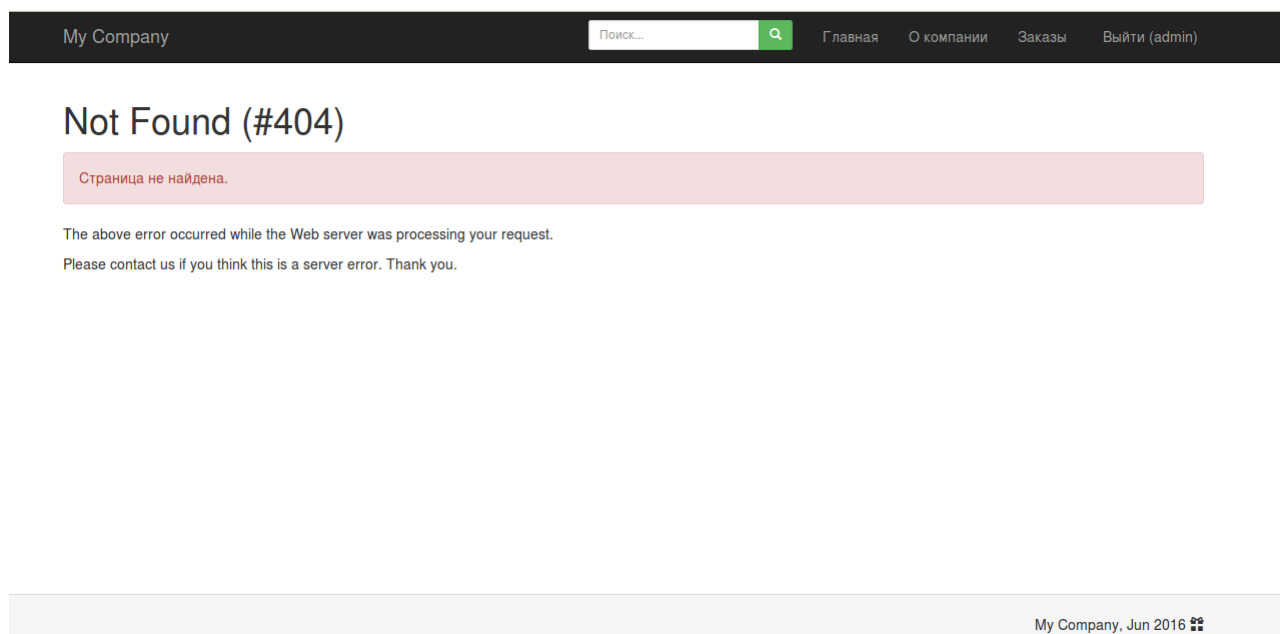


Рисунок 4.1 – Страница ошибки

При попытке пользователя, не имеющего соответствующего уровня доступа, входа в администраторскую часть системы, приложение выведет сообщение об ошибке и препятствует входу в систему (рис. 4.2).

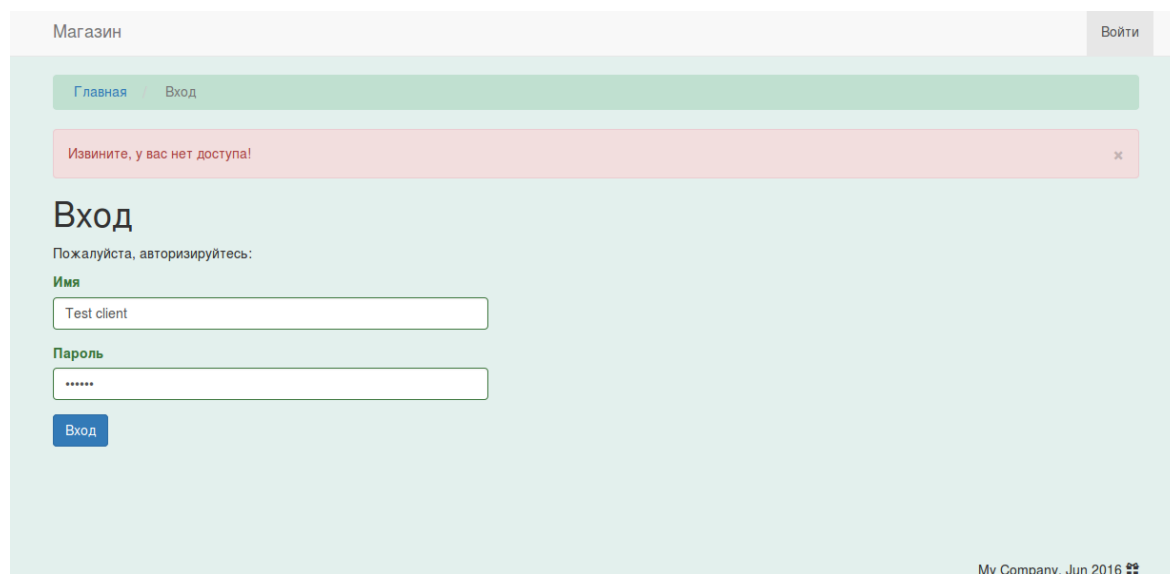


Рисунок 4.2 – Попытка входа в систему, не имея доступа

Разработанное приложение также предусматривает связи с помощью внешних ключей и при попытке удаления какой-то записи предупреждает пользователя о связи, предоставив ему следующие варианты:

- удаление всех записей, указывающих на удаляемую запись;
- назначение записям указание на другую запись.

На рисунке 4.3 представлен процесс удаления «Тестового поставщика» с присвоением, поставляемым им товарам, другого поставщика – в данном случае, «Тестового поставщика 2».

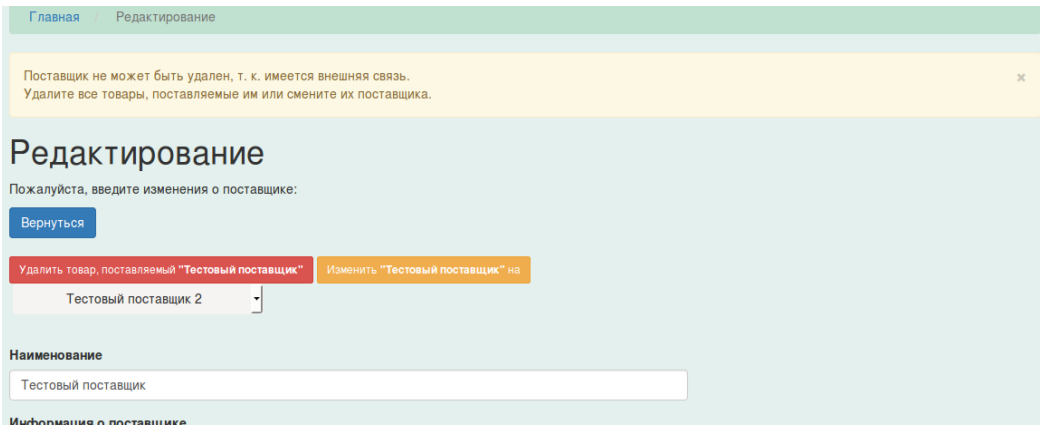


Рисунок 4.3 – Удаление записи, связанной внешним ключом

При некорректном заполнении полей, приложение выдаст ошибку и подскажет пользователю, как их необходимо заполнить (рис. 4.4).

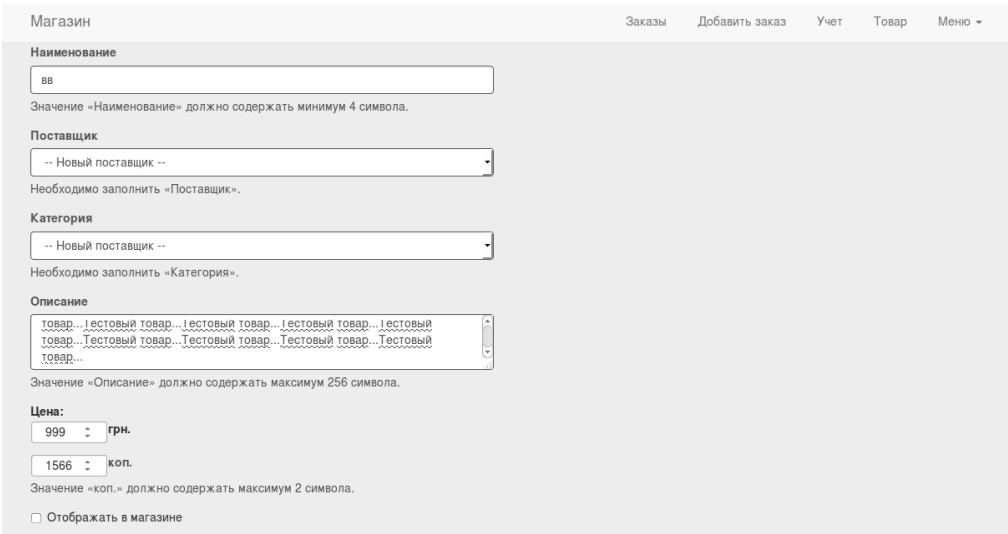


Рисунок 4.4 – Валидация введенных пользователем данных

5 КОНТРОЛЬНЫЙ ПРИМЕР ОФОРМЛЕНИЯ ЗАКАЗА

Для оформления заказа администратором или сотрудником, необходимо перейти на страницу «Добавить заказ» и, указав дату и магазин, нажать кнопку «Готово» (рис. 5.1-5.2). После этого, заказ будет отправлен.

Магазин

Заказы

Добавить заказ

Учет

Товар

Меню

Дата доставки

11-06-2016

Магазин

Магазин_2

Удалить

Подтверждение удаления

№	Название	Поставщик	Цена	Категория	Кол-во
Товара на сумму = 0 \$					
Весь товар					
1.	Product	Харьковская Бисквитн...	26.30	Шоколад 100 г...	Добавить
2.	product 2	newName	36.20	Cookies	Добавить
3.	Декстер	Roshen	26.05	Шоколад 200 г...	Добавить
4.	Еще товар	Provider 2	13.66	Cake	Добавить
5.	Молочный шоколад "Ал...	Roshen	12.35	Шоколад 100 г...	Добавить
6.	Печенье "Ромашка"...	Харьковская Бисквитн...	25.95	Cookies	Добавить
7.	Тестовый товар...	Тестовый поставщик 2...	66.13	Cake	Добавить
8.	Товар_1	Provider 2	9.10	Ice cream	Добавить
9.	Тофикс	Тестовый поставщик...	90.00	Sweets	Добавить

Рисунок 5.1 – Процесс оформления заказа. Инициализация заказа

Магазин

Заказы

Добавить заказ

Учет

Товар

Меню

Главная

Добавить заказ

Пожалуйста, сформируйте заказ и нажмите кнопку "Готово".

Дата доставки

11-06-2016

Магазин

Магазин_2

Готово

Удалить

Подтверждение удаления

№	Название	Поставщик	Цена	Категория	Кол-во
Присутствует в заказе					
1.	Молочный шоколад "Ал...	Roshen	12.35	Шоколад 100 г...	1 Удалить
2.	Печенье "Ромашка"...	Харьковская Бисквитн...	25.95	Cookies	1 Удалить
3.	Харьковский вафельны...	Харьковская Бисквитн...	200.00	Cake	1 Удалить
Товара на сумму = 238.3 \$					
Весь товар					
1.	Product	Харьковская Бисквитн...	26.30	Шоколад 100 г...	Добавить

Рисунок 5.2 – Процесс оформления заказа. Добавление товара

При этом если кнопка «Готово» нажата не будет (например, пользователь не точно определился с датой, пунктом доставки или содержанием заказа, а редактировать отправленные заказы может только администратор), заказ останется сформированным, однако будет отображаться лишь данному пользователю (рис. 5.3).

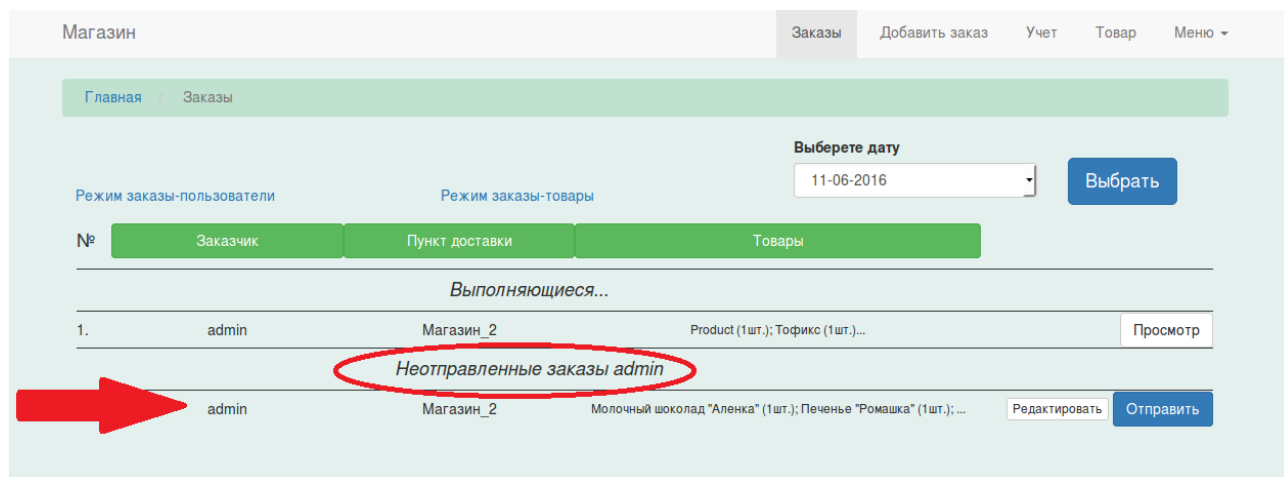


Рисунок 5.3 – Неотправленные заказы

Завершив оформление заказа (нажав кнопку «Готово» или «отправить», находящуюся напротив заказа, администратором), заказ будет отображаться в системе, и помечаться как «Поступивший заказ» — готовый к выполнению (рис. 5.4).

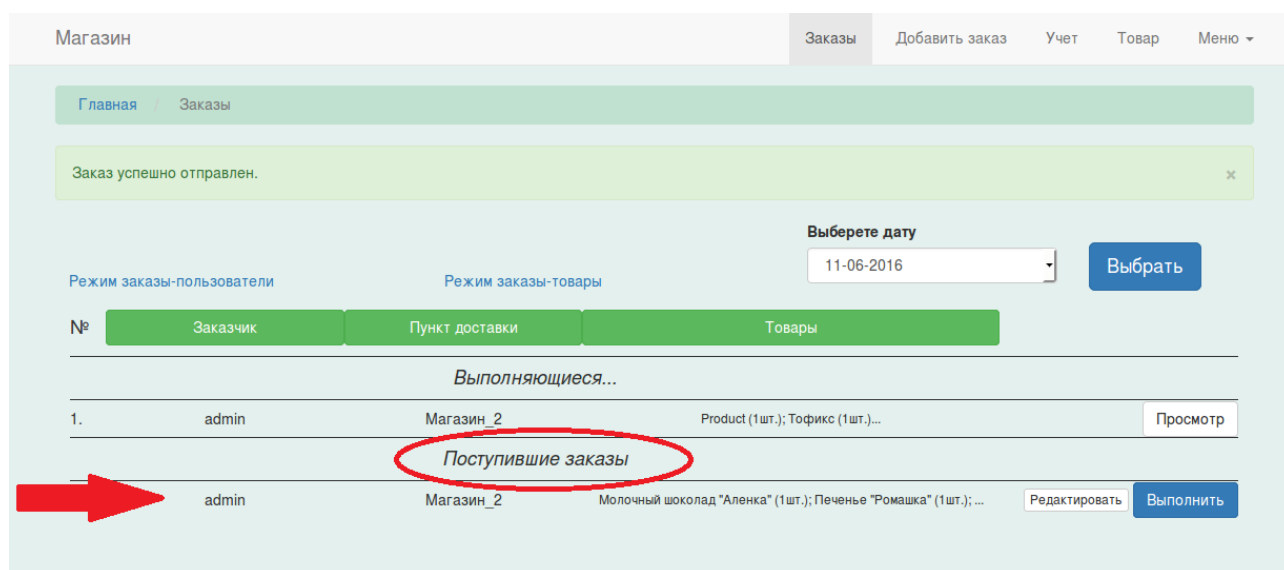


Рисунок 5.4 – Отправка заказа

Нажав кнопку «Выполнить», администратор отправляет заказ на выполнение – в подсистему грузоперевозок (рис. 5.5). Выполняющиеся заказы можно лишь просматривать, но невозможно редактировать.

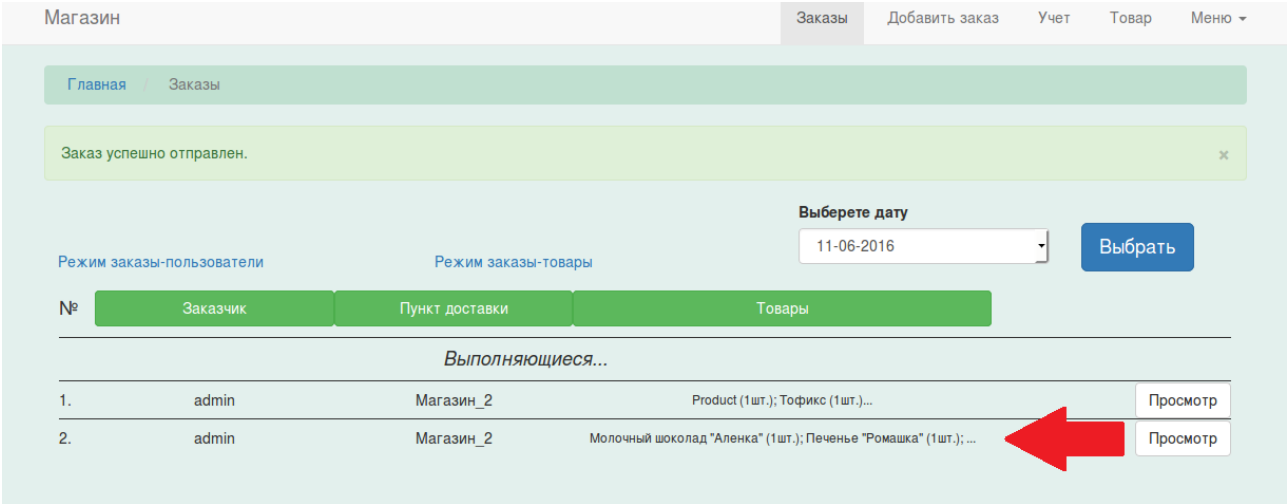


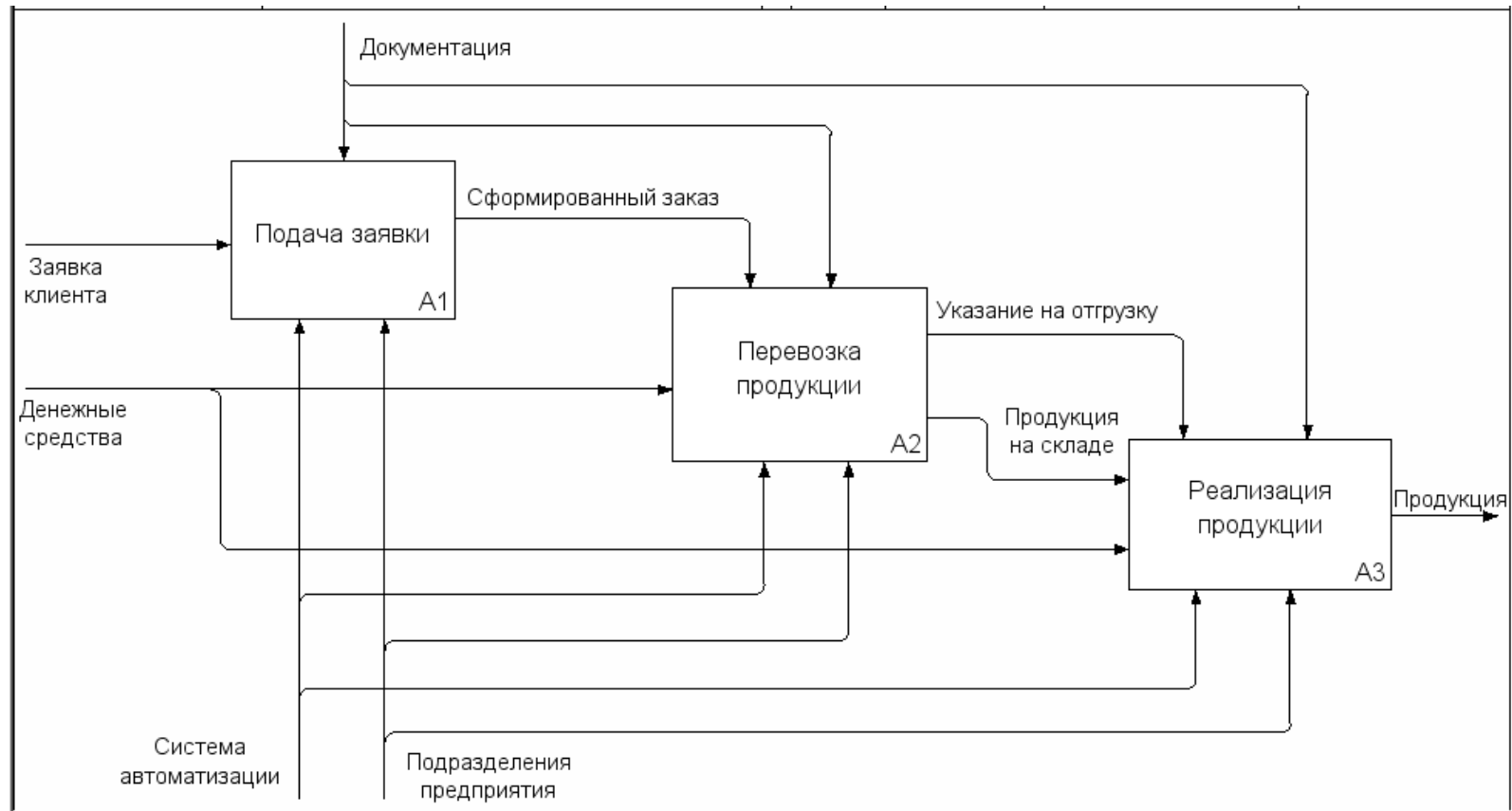
Рисунок 5.5 – Выполнение заказа

Подобным образом, за исключением некоторых ограничений, заказ оформляется и в клиентской части приложения.

Приложение В

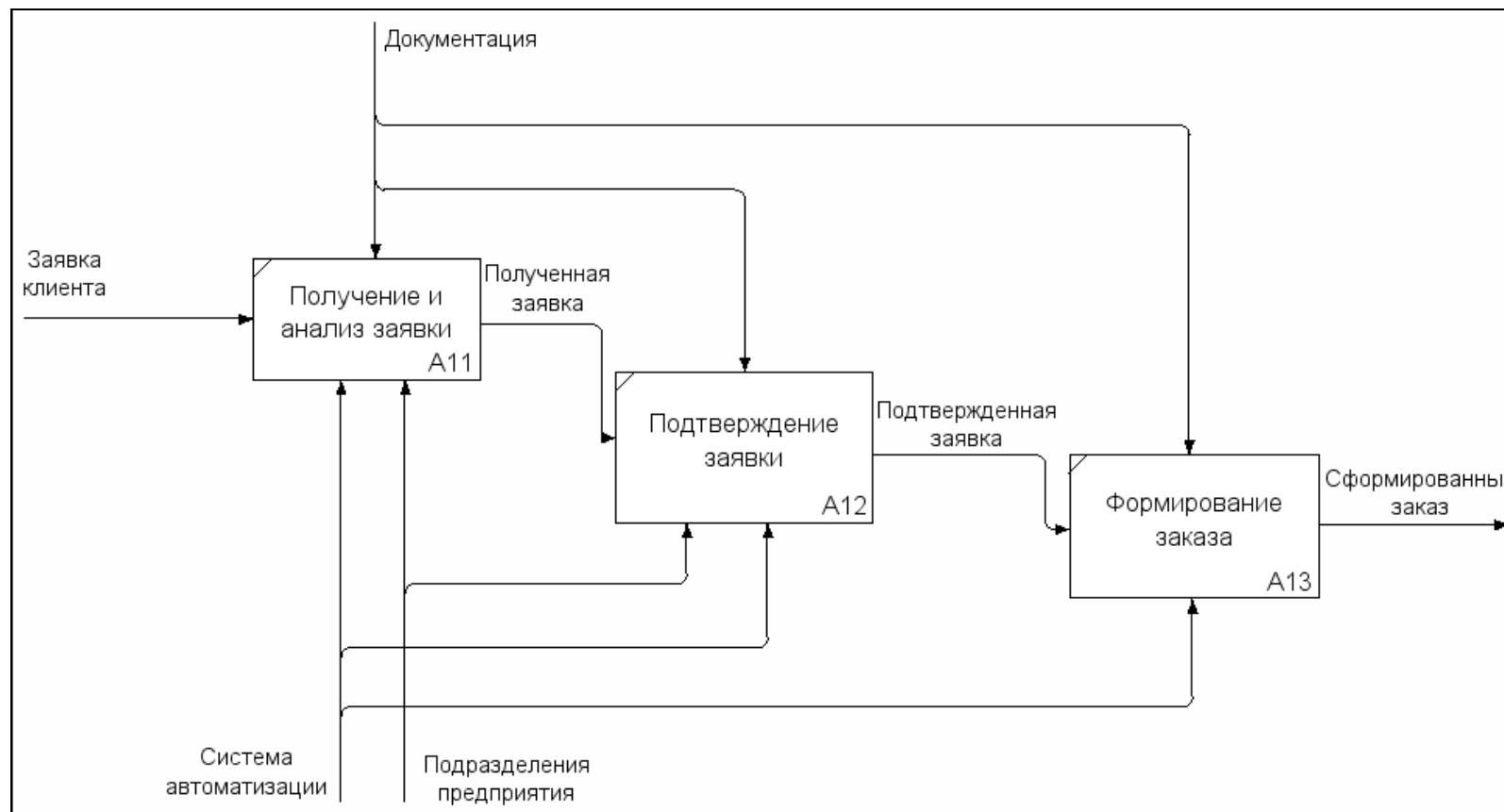
Графические материалы аттестационной работы

Диаграмма первого уровня функциональной модели системы



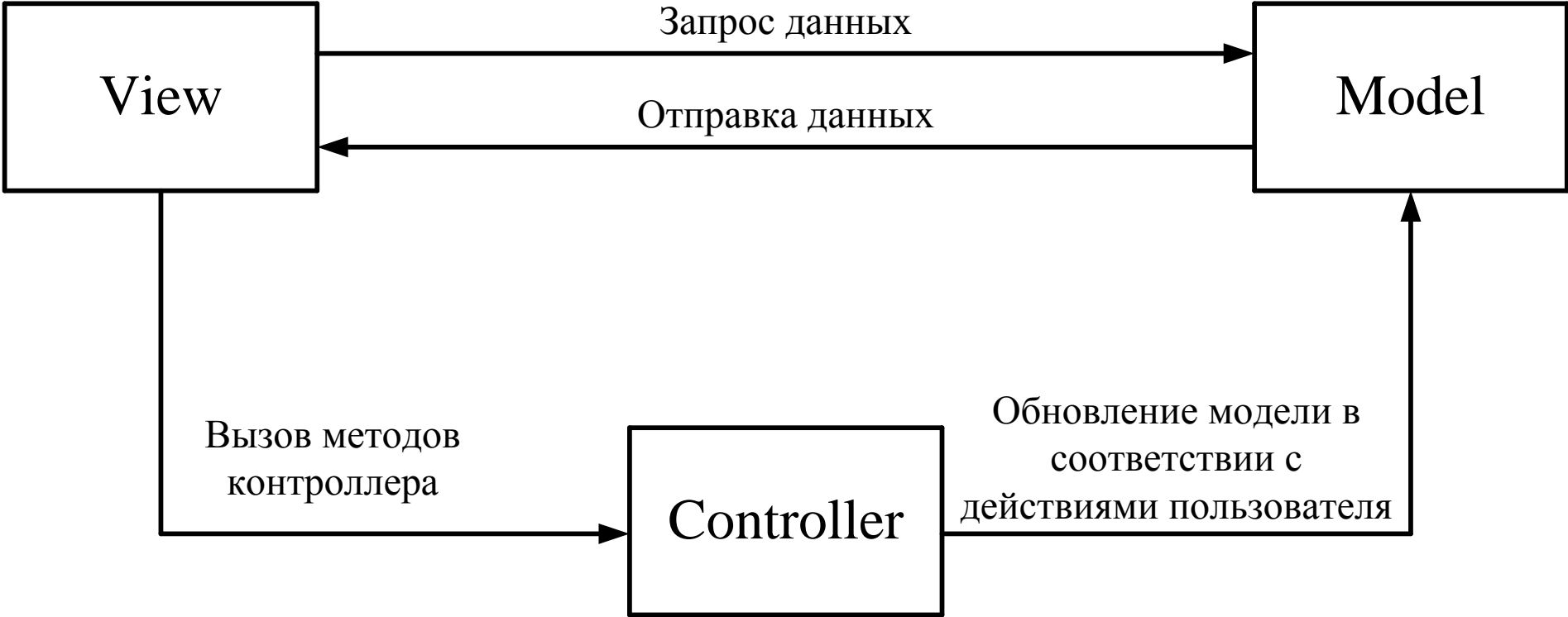
Разраб.	Ахмад Ф. Х.			Система Internet-продажи кондитерских изделий	
Провер.	Бескоровайный В.В.				
Н. Контр.	Бескоровайный В.В.				
				СИ-12-1	Лист 1
Утверд.	Гребенник И.В.			СТ	Листов 1

Декомпозиция «Поддачи заявки»



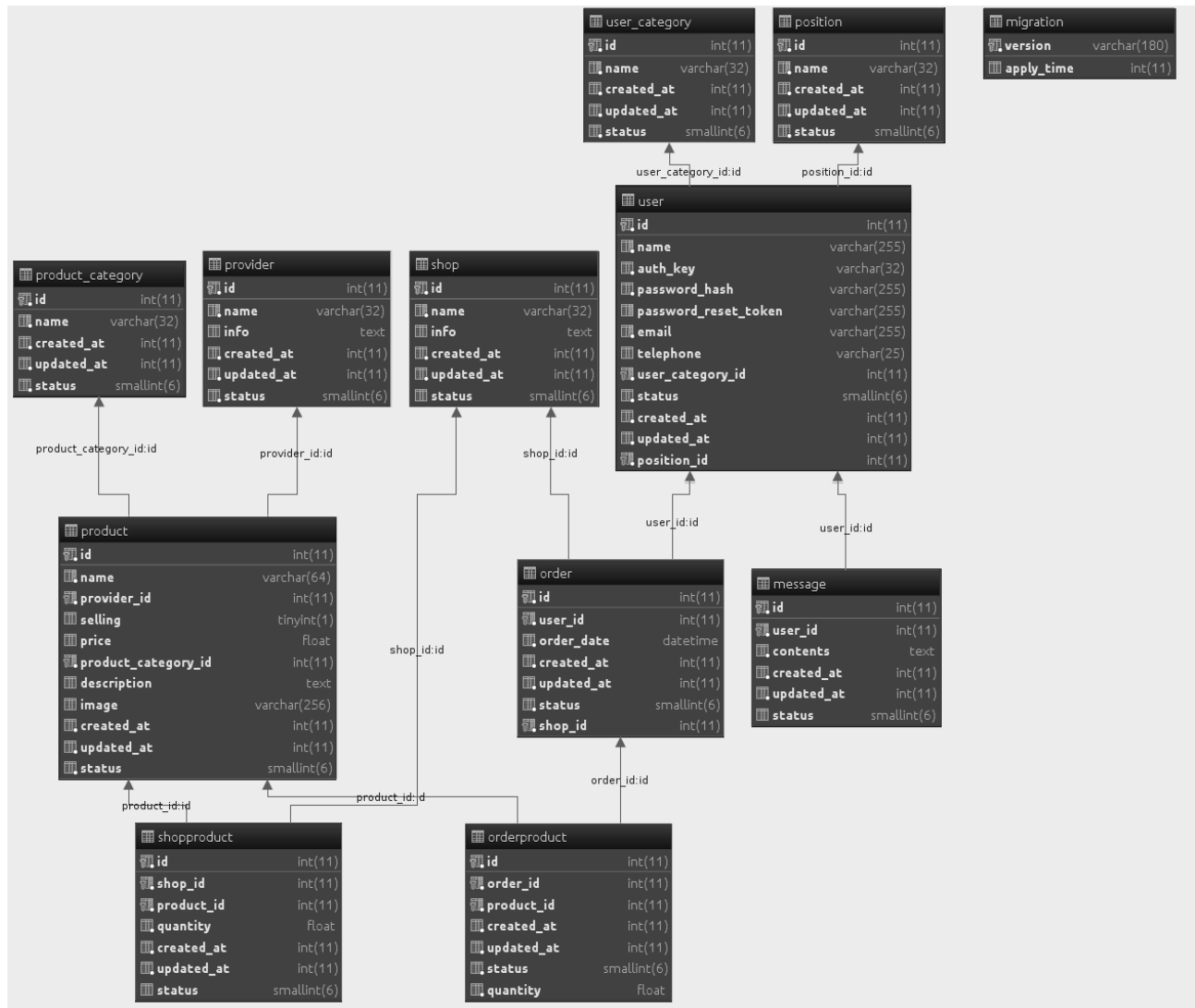
Разраб.	Ахмад Ф. Х.			Система Internet-продажи кондитерских изделий	
Провер.	Бескоровайный В.В.				
Н. Контр.	Бескоровайный В.В.				
				СИ-12-1	Лист 1
Утверд.	Гребенник И.В.			СТ	Листов 1

Концептуальная схема шаблона MVC



Разраб.	Ахмад Ф. Х.			Система Internet-продажи кондитерских изделий	
Провер.	Бескоровайный В.В.				
Н. Контр.	Бескоровайный В.В.				
				СИ-12-1	Лист 1
Утверд.	Гребенник И.В.			СТ	Листов 1

Схема базы данных системы



Разраб.	Ахмад Ф. Х.			Система Internet-продажи кондитерских изделий	
Провер.	Бескоровайный В.В.				
Н. Контр.	Бескоровайный В.В.				
				СИ-12-1	Лист 1
Утверд.	Гребенник И.В.			СТ	Листов 1

Экранная форма «Добавление/редактирование товара»

Магазин
Заказы
Добавить заказ
Учет
Товар
Меню

Редактирование произведено успешно.

Редактирование

Пожалуйста, введите изменения о товаре:

Вернуться

Наименование

Поставщик

Категория

Описание

Цена:

грн.

коп.

☒ Отображать в магазине

Картинка

No file selected.

☐ Удаление изображения

☐ Подтверждение удаления

Разраб.	Ахмад Ф. Х.			Система Internet-продажи кондитерских изделий	
Провер.	Бескорвайный В.В.				
Н. Контр.	Бескорвайный В.В.				
				СИ-12-1	Лист 1
Утверд.	Гребенник И.В.			СТ	Листов 1

Клиентская часть приложения



Производитель: Roshen

Стоимость: 12.35

Категория: Шоколад 100 г

Описание: 12/24 шт. в блоке.

Заказать



Печенье "Ромашка"

Производитель: Харьковская Бисквитная Фабрика

Стоимость: 35.95

Категория: Cookies

Описание: Песочное печенье с джемом.

Заказать



Харьковский вафельный торт

Производитель: Харьковская Бисквитная Фабрика

Стоимость: 200.00

Категория: Cake

Описание: Сувенирный торт состоит из отдельных долек, покрытых шоколадной глазурью, декорированных фигурными украшениями из шоколада и фруктами из помады.

Заказать

Разраб.	Ахмад Ф. Х.			Система Internet-продажи кондитерских изделий	
Провер.	Бескоровайный В.В.				
Н. Контр.	Бескоровайный В.В.				
				СИ-12-1	Лист 1
Утверд.	Гребенник И.В.			СТ	Листов 1

Экранная форма «оформление заказа»

[Главная](#) / [Добавить заказ](#)

Пожалуйста, сформируйте заказ и нажмите кнопку "готово".

Дата доставки

09-06-2016

Магазин

Магазин_1

Готово

Удалить

☐ Подтверждение удаления

№	Название	Поставщик	Цена	Категория	Кол-во	
Присутствует в заказе						
1.	Product	Харьковская Бисквитн...	26.30	Шоколад 100 г...	1	<div>Удалить</div>
Товара на сумму = 26.3 \$						
Весь товар						
1.	Product	Харьковская Бисквитн...	26.30	Шоколад 100 г...		<div>Добавить</div>
2.	product 2	newName	36.20	Cookies		<div>Добавить</div>
3.	Декстер	Roshen	26.05	Шоколад 200 г...		<div>Добавить</div>

Разраб.	Ахмад Ф. Х.			Система Internet-продажи кондитерских изделий	
Провер.	Бескоровайный В.В.				
Н. Контр.	Бескоровайный В.В.				
				СИ-12-1	Лист 1
Утверд.	Гребенник И.В.			СТ	Листов 1

Приложение Г
Ведомость аттестационной работы

№	Обозначение				Наименование			Доп. сведения				
					Текстовые документы							
1.	ГЮИК 502520.001 ПЗ				Пояснительная записка			115 с.				
2.	ГЮИК. 502520.001-01 12 01				Текст программы			19 с.				
3.	ГЮИК 502520.001 ИЗ				Руководство пользователя			17 с.				
					Графические документы							
4.					Диаграмма первого							
					уровня функциональной							
					модели системы			Включено в ПЗ				
5.					Декомпозиция							
					«Подачи заявки»			Включено в ПЗ				
6.					Концептуальная схема							
					шаблона MVC			Включено в ПЗ				
7.					Схема базы данных							
					системы			Включено в ПЗ				
8.					Экранная форма							
					«Добавление /							
					редактирование товара»			Включено в ПЗ				
9.					Клиентская часть							
					приложения			Включено в ПЗ				
10.					Экранная форма							
					«оформление заказа»			Включено в ПЗ				
					ГЮИК 502520.001 ДЗ							
Изм.	Лист	№ докум	Подпись	Дата								
Разраб.		Ахмад Ф. Х.			Система Internet-продажи кондитерских изделий				Лист	Листов		
Проверил.		Бескоровайный В.В.								1	1	
Н. Контр.		Бескоровайный В.В.						ХНУРЭ Кафедра СТ				
Утверд.		Гребенник И.В.										