

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ
НАУК

КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ

**АВТОМАТИЗИРОВАННАЯ СИСТЕМА УЧЁТА ЗАКАЗОВ
ООО «РАССВЕТ»**

Выпускная квалификационная работа
обучающегося по направлению подготовки
02.03.01 «Математика и компьютерные науки»
очной формы обучения, группы 07001403
Миронова Николая Максимовича

Научный руководитель
к.т.н., доцент
Бурданова Е. В.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛАВА 1. АНАЛИЗ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ ООО «РАССВЕТ». АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УЧЁТА	6
1.1. Общая характеристика предприятия ООО «Рассвет»	6
1.2. Структура предприятия ООО «Рассвет»	9
1.3. Теоретические аспекты автоматизированных систем учета	12
1.4. Этапы создания автоматизированной системы учета заказов и основные требования к системе.....	14
ГЛАВА 2. ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА ЗАКАЗОВ.....	19
2.1. Проектирование базы данных автоматизированной системы	19
2.2. Разработка алгоритмов работы автоматизированной системы.....	24
2.3. Выбор средств разработки автоматизированной системы	31
ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА ЗАКАЗОВ	40
3.1. Создание основных программных модулей автоматизированной системы.....	40
3.2. Организация внешнего вида страниц веб-приложения	46
3.3. Организация учета заказов пользователей	47
ГЛАВА 4. ТЕСТИРОВАНИЕ РАЗРАБОТАННОЙ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА ЗАКАЗОВ	51
4.1. Тестирование функций модуля пользователя.....	51
4.2. Тестирование функций модуля администрирования	61
ЗАКЛЮЧЕНИЕ	68
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	70
ПРИЛОЖЕНИЕ 1	72
ПРИЛОЖЕНИЕ 2	78
ПРИЛОЖЕНИЕ 3	86
ПРИЛОЖЕНИЕ 4	89

ВВЕДЕНИЕ

В нынешнем информационном мире влияние сети Интернет на окружающую жизнь трудно переоценить. Интернет проник во все сферы нашего мира: политика, социальная сфера, медицина, обучение, в том числе и торговля различными товарами и услугами. Отдельным пунктом необходимо отметить также востребованность автоматизированных систем учёта заказов для компаний, занимающихся определёнными видами торговли товарами и услугами. На сегодняшний день любая компания должна иметь автоматизированную систему, которая способна эффективно управлять ресурсами компании, вести точный и безошибочный учет поставляемой продукции, предоставлять клиентам компании в удобном и быстром режиме приобретать продукцию компании, обеспечивать информационную поддержку существующего бизнеса. При помощи автоматизированной системы решаются такие важные и необходимые задачи для любой компании, как представление её в сети Интернет, расширение потенциальной аудитории потребителей товаров, производимых компанией, учет производимой продукции, эффективное управление всеми ресурсами компании.

Разработка автоматизированной системы учета заказов для предприятия ООО «Рассвет» является актуальной и востребованной темой. Автоматизированная система предоставляет в распоряжение компании эффективный инструмент управления, дает возможность потенциальным и существующим клиентам легко приобретать продукцию компании, получать актуальную информацию о товарах и услугах компании, ее деловых интересах, что может помочь найти новых заказчиков и партнеров по бизнесу, а, следовательно, способствует увеличению объема продаж и рентабельности компании.

Целью данной выпускной квалификационной работы является разработка автоматизированной системы учета заказов для предприятия ООО «Рассвет».

Для достижения поставленной цели необходимо выделить и решить следующие частные задачи:

1. Изучение структуры и деятельности предприятия ООО «Рассвет».
2. Изучение соответствующих теме литературных источников.
3. Изучение теоретических аспектов, этапов разработки и внедрения автоматизированных систем учета.
4. Проектирование автоматизированной системы учета, включающее в себя проектирование базы данных и разработку алгоритмов работы автоматизированной системы.
5. Изучение программного инструментария, применяемого для разработки и внедрения автоматизированных систем учета.
6. Программная реализация автоматизированной системы с использованием актуальных средств разработки приложений.
7. Тестирование корректной работы всех функций автоматизированной системы учёта.

Объектом исследования выпускной квалификационной работы является предприятие ООО «Рассвет».

Предметом исследования является инструмент эффективного управления деятельностью предприятия в виде автоматизированной системы.

Практическая ценность заключается в том, что задачи, разработанные в процессе выполнения ВКР, должны быть реализованы на исследуемом предприятии. Эту существенно повысит качество обслуживания, конкурентоспособность данного предприятия и увеличит эффективность его работы в целом.

Выпускная квалификационная работа «Автоматизированная система учета заказов ООО «Рассвет» включает в себя введение, заключение, главу 1,

главу 2, главу 3, главу 4, список использованной литературы, приложение 1, приложение 2, приложение 3, приложение 4.

В 1 главе производится анализ исследуемого предприятия и изучение теоретических аспектов автоматизированных систем учета.

Во 2 главе описываются основные этапы проектирования автоматизированной системы учета.

В 3 главе описываются этапы программной реализации автоматизированной системы учета.

В 4 главе описывается процесс тестирования основных функций автоматизированной системы учета.

Приложение 1 включает в себя SQL код базы данных, сгенерированный утилитой phpMyAdmin.

Приложение 2 включает в себя программный код web приложения на языке программирования php с использованием html.

Приложение 3 включает в себя разработанные макеты страниц веб-приложения.

Приложение 4 включает в себя таблицу типов полей таблиц базы данных.

Выпускная квалификационная работа содержит 91 страницу, 53 рисунка, 1 таблицу.

ГЛАВА 1. АНАЛИЗ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ ООО «РАССВЕТ». АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УЧЁТА

1.1. Общая характеристика предприятия ООО «Рассвет»

Общество с ограниченной ответственностью «Рассвет» - это торговое предприятие, которое специализируется на товарах, предназначенных для строительства, ремонта и обустройства жилых и не жилых помещений, благоустройства участков и общественных территорий. Предприятие осуществляет свою торгово-хозяйственную деятельность на потребительском рынке города Белгород. Основным видом деятельности предприятия является торговля промышленными, строительными и бытовыми товарами, подразделяющиеся на товары повседневного спроса и товары длительного пользования.

Основная цель предприятия ООО «Рассвет» заключается в содействии экономическому развитию страны, сосредоточенном в результате удовлетворения потребности людей в товарах народного потребления, в обеспечении прибыли предприятия, справедливого отношения к сотрудникам предприятия в рамках нормативно-правовых документов.

Конкурентное преимущество предприятия состоит в более низких ценах на товары, хорошей и качественной рекламе, в грамотно обученных кадрах. Предприятие предоставляет возможность в профессиональном росте, получении дополнительной подработки, осуществляет для своих сотрудников обучающие семинары по маркетингу, психологии, применяет систему поощрений и стимулирования.

Не менее важной целью предприятия ООО «Рассвет» является занятие лидирующих позиций на потребительском рынке строительных товаров, что достигается путем производства продукции с учетом индивидуальных пожеланий клиентов и экономической обстановки в строительной сфере, с

использованием передовой техники, соответствующей экологическим стандартам.

В рамках проводимого исследования предприятия необходимо отметить, что данное предприятие является юридическим лицом и имеет самостоятельный отчетный баланс, расчетный счет в Сбербанке РФ, круглую печать, штампы, бланки с указанием своего полного наименования, собственный зарегистрированный товарный знак.

Одним из существенных показателей будущей получаемой прибыли от продаж является дата создания и начала деятельности предприятия. Торговое предприятие ООО «Рассвет» было создано 21 мая 2017 года в соответствии с действующим законодательством Российской Федерации. В своей деятельности предприятие руководствуется Уставом и Учредительным договором, а также соответствующими нормативно-правовыми актами в сфере осуществления предпринимательской деятельности в городе Белгород и Белгородской области.

Данное предприятие отвечает по своим обязательствам находящимися в его распоряжении денежными средствами. Предприятие от своего имени приобретает имущественные и личные неимущественные права, исполняет обязанности, выступает истцом и ответчиком в суде.

Предприятие является самостоятельным хозяйствующим субъектом, созданным для реализации товаров, выполнения работ и оказания услуг в целях удовлетворения общественных потребностей и получения прибыли.

Предприятие осуществляет свою деятельность на основе полного хозяйственного расчета, самофинансирования и самокупаемости. Также данное предприятие самостоятельно отвечает за полученные результаты от своей производственной деятельности и выполнение обязательств перед заказчиками, бюджетами, банками и другими контрагентами.

Предприятие заключает сделки (договоры, контракты) со многими предприятиями города Белгород, городами Белгородской области на поставку своей продукции.

Источниками формирования финансовых результатов предприятия являются прибыль, амортизационные отчисления. На предприятии созданы следующие фонды: уставный, резервный и другие фонды общего и специального назначения.

Предприятие имеет самостоятельный баланс, в котором отражается его имущество. Имущество предприятия состоит из основных и оборотных средств, а также иных материальных и финансовых ценностей.

Управление основной деятельностью предприятия осуществляется генеральным директором предприятия. Директор самостоятельно определяет структуру управления предприятия, формирует штаты и стратегию дальнейшего развития предприятия.

Предприятие обеспечивает гарантированный законом минимальный размер заработной платы, условия труда и меры социальной защиты работников. Размер заработной платы работников предприятия установлен согласно штатному расписанию.

Предприятие ведёт бухгалтерскую и статистическую отчётность в порядке, установленном законодательством РФ.

Предприятие осуществляет свою деятельность в соответствии с разрабатываемым и утверждаемым планом экономического и социального развития.

Основные виды деятельности предприятия в соответствии с учредительными документами:

1. Оптовая и розничная торговля, предполагающие как наличную, так и безналичную формы оплаты.
2. Осуществление сбора и распространение коммерческой и экономической информации.

Уставный капитал ООО «Рассвет» составляет 10 тыс. руб.

Режим работы данного предприятия: по будням с 10.00 до 20.00, по выходным с 10.00 до 15.00, без перерыва на обед.

Анализируемое торговое предприятие имеет достаточно выгодное географическое расположение, оно находится в районе Старый город города Белгорода, на улице Корочанская, которая является одной из центральных и оживленных улиц города, что способствует привлечению многочисленных клиентов, а также повышению качества доставки товаров.

1.2. Структура предприятия ООО «Рассвет»

Структура предприятия - это его внутреннее строение, характеризующее состав подразделений, связи, подчиненность и взаимодействие между ними. Совокупность производственных подразделений (отделов, участков, обслуживающих хозяйств и служб) прямо или косвенно участвующих в производственном процессе, их количество и состав определяют организационную структуру предприятия[3].

Схема организационной структуры предприятия ООО «Рассвет», представленная на рис. 1.1, показывает, что предприятие включает в себя следующие подразделения: отдел закупок, отдел продаж, склад, отдел бухгалтерии, отдел web-проектов, которые непосредственно подчиняются генеральному директору. В ООО «Рассвет» четко прослеживается линейно-функциональная структура организации деятельности предприятия.

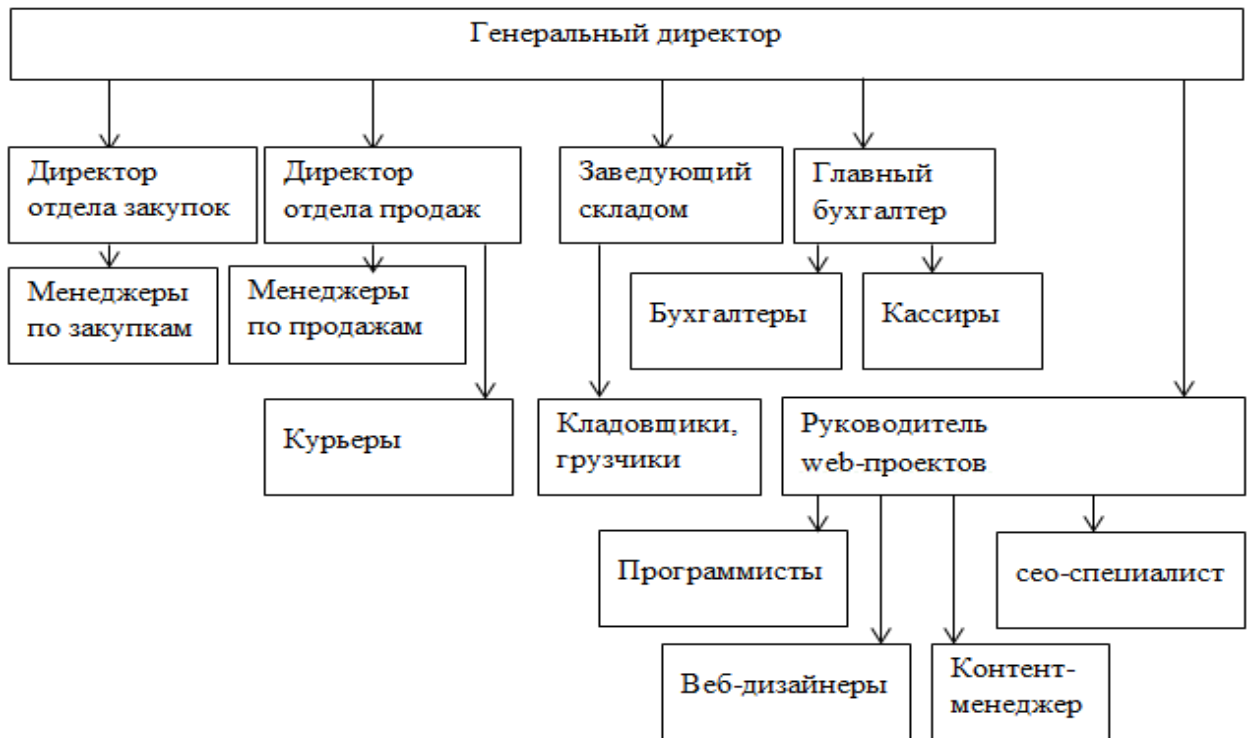


Рис.1.1. Схема организационной структуры предприятия

Линейно-функциональная структура заключается в сочетании горизонтальных и вертикальных связей, т.е. выделении подразделений (служб, отделов, отдельных работников) по линейным и функциональным признакам[11].

Для данного предприятия такой тип организационной структуры является наиболее приемлемым, так как предприятие, выпускающее ограниченную номенклатуру продукции для обеспечения своего функционирования, непременно требует решения стандартных управленческих задач[11].

Всего на предприятии работает 24 человека.

Руководствуясь представленной выше схемой организационной структуры предприятия, следует отметить, что исполнительным органом является генеральный директор. Он осуществляет общее руководство, несет персональную ответственность за выполнение договорных обязательств, за соблюдение финансовой и расчетной дисциплины, действующего

законодательства, норм по охране труда, за ритмичную и четкую работу организации в целом. Это высококвалифицированный работник, имеющий инженерно-экономическое образование, определенный опыт предпринимательской деятельности. Генеральный директор определяет основные направления ассортиментной, сбытовой, кадровой политики, осуществляет общее руководство организацией на принципах единоначалия и ответственности.

В непосредственном подчинении генерального директора находятся: отдел продаж, отдел закупок, бухгалтерия, склад, отдел, занимающийся web-проектами.

Во главе каждого отдела предприятия стоит директор, координирующий деятельность соответствующих подразделений. У каждого директора подразделения в подчинении находятся сотрудники, занимающие определенные должности.

В подчинении директора отдела по закупкам находятся менеджеры по закупкам, осуществляющие анализ экономических показателей, оптимизацию производственных расходов, закупку товаров и их учет.

В подчинении директора отдела по продажам находятся менеджеры по продажам и курьеры. Менеджеры занимаются анализом состояния потребительского рынка, организацией рабочего пространства на предприятии, консультацией и обслуживанием клиентов. Курьеры осуществляют доставку товаров клиентам.

Не менее важным является отдел хранения поставленной продукции (склад). Важно отметить, что склад предприятия находится на территории реализации продукции предприятия, что существенно сокращает затраты на транспортировку товаров для дальнейшей реализации. Главой этого подразделения является заведующий складом. В его обязанности входит приём, хранение и учёт поставленной продукции. В подчинении заведующего складом находятся кладовщики и грузчики.

В подчинении главного бухгалтера находятся бухгалтеры и кассиры. В обязанности бухгалтеров входит расчет окладов, начисление заработной платы сотрудникам, выполнение расчетов налоговых отчислений, себестоимости продукции, проведение счетов от поставщиков. В обязанности кассиров входит ведение денежной кассы на предприятии, прием и выдача денег, работа с пластиковыми банковскими картами и электронными деньгами.

В подчинении руководителя web-проектов находятся: контент-менеджеры, ответственные за наполнение автоматизированной системы учета продукцией предприятия; программисты, занимающиеся технической поддержкой системы; веб-дизайнеры, отвечающие за внешний вид системы; SEO-специалисты, отвечающие за продвижение и рекламу предприятия в сети Интернет.

В целом структура предприятия создает достаточную организационную основу для результативной коммерческой деятельности.

1.3. Теоретические аспекты автоматизированных систем учета

Цель автоматизации информационных процессов - повышение производительности и эффективности труда работников, улучшение качества информационной продукции и услуг, повышение сервиса и оперативности обслуживания пользователей. Автоматизация базируется на использовании средств вычислительной техники и необходимого программного обеспечения[9].

Основными задачами автоматизации информационных процессов являются[9]:

1. Сокращение трудозатрат при выполнении традиционных информационных процессов и операций.
2. Устранение рутинных операций.

3. Ускорение процессов обработки и преобразования информации.
4. Расширение возможностей осуществления статистического анализа и повышение точности учетно-отчетной информации.
5. Повышение оперативности и качественного уровня обслуживания пользователей.
6. Модернизация или полная замена элементов традиционных технологий.
7. Облегчение возможностей широкого обмена информацией.
8. Для решения представленных задач существуют автоматизированные системы.

Автоматизированная система учета (АСУ) – человеко-машинная система, в которой с помощью современных и актуальных информационно-математических методов сбора, передачи, хранения и обработки информации решаются задачи по управлению производственными процессами и ресурсами предприятия[10].

Обязательными элементами этой системы являются: человек, технические средства, информационно-математические методы, совокупность задач и функций управления, представленных в формализованном виде, программные средства[10].

В зависимости от вида деятельности различают следующие разновидности автоматизированных систем АС[9]:

1. АСУ технологическими процессами (АСУ ТП).
2. АСУ предприятиями (АСУП), производством (АСУП) и т. д.
3. САП (системы автоматизированного проектирования).
4. САПР (системы автоматизированного проектирования и расчета).
5. САПР ТП (системы автоматизированного проектирования технологических процессов) и тому подобное.
6. АС обработки и передачи информации.
7. АИПС (автоматизированная информационно-поисковая система).
8. САМ (АС технологической подготовки производства).

9. Автоматизированные системы контроля и испытаний.

10. АС, объединяющие функции перечисленных выше систем.

АС реализуют информационную технологию в виде определенной последовательности информационно связанных функций, задач или процедур, выполняемых в автоматизированном (интерактивном) или автоматическом режимах[9].

Автоматизированная система учета производственного назначения осуществляет сбор информации с объекта управления, передает, преобразует и обрабатывает ее, формирует управляющие команды и выполняет их на управляемом объекте, т. е. осуществляет только те функции, которые поддаются автоматизации. Участие человека в этой системе заключается в определении целей и критериев управления, их корректировки при изменении условий, в выполнении функции ведения работы автоматизированной системы, а в случае необходимости, в изменении алгоритма их работы, в принятии общих решений по управлению автоматизированной системой в измененных или сложных ситуациях[10].

1.4. Этапы создания автоматизированной системы учета заказов и основные требования к системе

Автоматизированная система учета заказов (АСУЗ) в целом и каждая из ее подсистем должны соответствовать общим техническим требованиям, установленным ГОСТом, а также требованиям, содержащимся в техническом задании (ТЗ) на ее создание.

Основными этапами создания автоматизированной системы учета заказов для предприятия ООО «Рассвет» являются:

1. Этап формирования требований. На данном этапе производится исследование объекта, формирование требований в АСУЗ

2. Этап разработки концепции АСУЗ. Данный этап состоит в разработке различных вариантов концепции АСУЗ в соответствии с требованиями, сформированными на первом этапе.

3. Этап разработки технического задания. Включает в себя разработку и утверждение технического задания в соответствии с разработанным вариантом концепции АСУЗ.

4. Этап разработки технорабочего проекта. Включает в себя разработку программ и проектных решений по системе и её частям (программная реализация), разработку и оформление документации на поставку изделий для комплектования АСУЗ и (или) технических требований (технических заданий) на их разработку.

5. Этап ввода системы в эксплуатацию. Включает в себя такие пункты, как подготовка объекта автоматизации к вводу АСУЗ в действие, подготовку персонала, комплектацию АСУЗ поставляемыми товарами, пусконаладочные работы, проведение предварительных испытаний, проведение опытной эксплуатации, проведение приемочных испытаний.

6. Этап сопровождения АСУЗ. Включает в себя выполнение работ в соответствии с гарантийными обязательствами и послегарантийное обслуживание АСУЗ.

Основными требованиями к автоматизированной системе учета заказов для предприятия ООО «Рассвет» являются:

1. Удобство и простота общения с системой, в том числе настройка системы под конкретного пользователя и эргономичность конструкции.

2. Безопасность системы.

3. Надежность системы.

4. Производительность системы.

5. Оперативность ввода, обработки, размножения и поиска информации.

6. Возможность оперативного обмена информацией между персоналом организации, с различными лицами и организациями за ее пределами.

7. Возможность модернизации, усовершенствования и развития системы.

8. Обеспечение заданного среднего срока службы с учётом проведения восстановительных работ, включая ремонт различных компонентов автоматизированной системы.

9. Совместимость с другими автоматизированными системами.

10. Обеспечение расширяемости системы, т.е. возможность добавления новых модулей в уже существующую систему.

11. Учёт международных и промышленных стандартов разработки автоматизированных систем.

12. Непротиворечивость автоматизированной системы общегосударственным нормативно-правовым актам.

Для модуля клиента автоматизированной системы основными требованиями являются:

1. Дружественный и понятный интерфейс модуля.

2. Возможность регистрации и авторизации клиента в системе.

3. Предоставление одинакового функционала как зарегистрированному пользователю, так и не зарегистрированному.

4. Показ каталога товаров предприятия с неограниченным числом категорий, подкатегорий, товаров.

5. «Живой» и обычный поиск товаров с помощью поисковой формы.

6. Фильтрация товаров по цене, по производителю.

7. Показ цены, описания товаров, их характеристик и наличие на складе.

8. Возможность проведения таких операций с отдельными товарами, как добавление товара в избранное, сравнение товаров по их характеристикам, добавление товара в «корзину» для последующего оформления заказа, изменение количества товара, удаление товара из «корзины».

9. Возможность составления и отправки отзывов о работе предприятия и системы в целом.

10. Предоставление выбора способов доставки и оплаты заказа.

11. Учет общего веса заказа, количества товаров, способа доставки, способа оплаты, цены заказа.

Для модуля администратора автоматизированной системы основными требованиями являются:

1. Дружественный интерфейс модуля.

2. Предоставление возможности работы с системой сотрудникам различных подразделений предприятия путём создания системных профилей (например, профиль администратора, профиль курьера и т.п.).

3. Делегирование прав и полномочий в зависимости от сотрудника, работающего с системой.

4. Возможность работы с отдельными компонентами системы.

5. Учёт количества компонентов системы.

6. Предоставление функций ввода, изменения, удаления, сортировки и поиска информации.

Принимая во внимание все результаты, полученные в ходе исследования деятельности предприятия ООО «Рассвет», а также учитывая изученные теоретические аспекты, этапы создания автоматизированных систем учёта и основные требования к системе, было принято организовать автоматизированную систему учёта заказов в виде web-приложения, разделенного на два основных модуля: модуль пользователя и модуль администратора. Такая организация позволит максимально эффективно использовать все имеющиеся ресурсы предприятия, исключить затраты на приобретение дорогостоящего программного обеспечения и оборудования, предоставить клиентам предприятия удобный и простой способ взаимодействия с ресурсами предприятия, т.к. подобная организация системы не потребует установки дополнительного программного

обеспечения на устройства клиента. Для корректной работы с системой будет достаточно использования Web-браузера.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА ЗАКАЗОВ

2.1. Проектирование базы данных автоматизированной системы

Этап проектирования базы данных представляет собой процесс создания схемы базы данных и определение необходимых ограничений целостности[6].

Основными задачами проектирования базы данных являются:

1. Обеспечение хранения в базе данных всей необходимой информации для корректной работы автоматизированной системы учета заказов (АСУЗ).
2. Обеспечение возможности извлечения данных по всем требуемым запросам.
3. Сокращение избыточности и дублирования данных.
4. Обеспечение целостности базы данных.

Основными стадиями проектирования базы данных являются:

1. Введение и определение информационных объектов (сущностей) АСУЗ.
2. Выделение характеристик (атрибутов) этих объектов.
3. Определение связей между сущностями.
4. Описание ограничений целостности атрибутов сущностей.

Принимая во внимание результаты анализа предприятия ООО «Рассвет» и требования к АСУЗ, следует ввести следующие информационные объекты: объект категория товаров, объект подкатегория товаров, объект товар, объект поставщик, объект пользователь, объект заказ, объект избранный товар, объект список товаров в заказе, объект подразделение предприятия, объект сотрудник, объект должность.

Данные объекты объединяются в схему, изображенную на рис. 2.1.

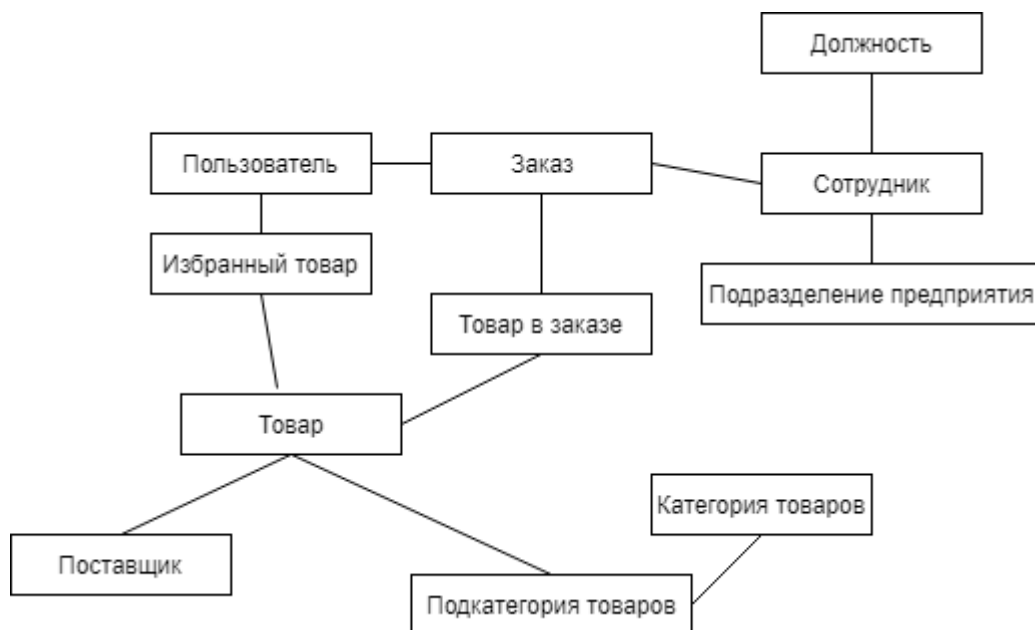


Рис.2.1. Схема информационных объектов базы данных

Далее необходимо наделить каждый информационный объект характеристиками.

Объект категория товаров должен обладать следующими характеристиками: код категории, название категории.

Объект подкатегория товаров должен обладать такими характеристиками, как код подкатегории, код категории, в которую входит данная подкатегория, название подкатегории.

Объект товар характеризуется такими характеристиками, как код товара, код подкатегории, которой принадлежит товар, код поставщика товара, название товара, артикул товара, цена товара, название цены товара, т.к. у разных товаров могут быть различные единицы цены (например, шт./м², руб./кг. и т.п.), описание товара, компания-производитель товара, страна, в которой произведен товар, физические характеристики, такие как: вес брутто, длина, ширина, высота; оставшееся количество на складе, изображения товара для его отображения в системе.

Объект поставщик должен обладать следующими характеристиками: код поставщика, название поставщика, юридический адрес поставщика, телефон.

Объект пользователь характеризуется такими характеристиками как код пользователя, электронная почта пользователя, пароль пользователя для доступа к системе, имя и фамилия пользователя, адрес проживания для возможности автоматического заполнения формы доставки курьером, телефон.

Объект заказ включает в себя следующие характеристики: код заказа, код пользователя, сформировавшего заказ, код сотрудника, ответственного за заказ, статус заказа, общая стоимость заказа, дата оформления заказа, дата исполнения заказа, метод доставки заказа, метод оплаты заказа, комментарий к заказу.

Объект список избранных товаров включает в себя характеристики код пользователя, код товара.

Объект подразделение компании характеризуется кодом подразделения, названием подразделения.

Объект сотрудники характеризуется кодом сотрудника, кодом подразделения, к которому принадлежит сотрудник, имя и фамилия сотрудника, адрес проживания сотрудника, телефон, дата начала работы, оклад.

Объект должности характеризуется кодом должности, названием должности.

Все информационные объекты определены и охарактеризованы.

Следующим шагом проектирования базы данных является определение связей между информационными объектами, а также описание ограничений целостности, то есть требования к допустимым значениям характеристик информационных объектов.

Анализ работы предприятия показал, что за каждой категорией товаров закреплен определенный список подкатегорий для удобства навигации по товарам, то есть выявляем первую связь: категории-подкатегории. Данный вид связи является «один-ко-многим»: в одной категории может быть несколько подкатегорий, у каждой подкатегории может быть только одна

категория. Атрибут код категории должен являться первичным ключом, атрибут название должен быть уникальным.

Продолжим определение связей с сущности подкатегории. Данную сущность требуется связать с сущностью товары, т.к. товары расположены в определенных подкатегориях. Данный вид связи также является «один-ко-многим». Атрибут код подкатегории должен быть первичным ключом, атрибут название подкатегории должен быть уникальным.

Следующей сущностью для определения связи примем сущность поставщики. Поставщики поставляют товары предприятию, т. е. налицо связь с сущностью товары, которая имеет тип связи «один-ко-многим». Атрибут код поставщика должен быть первичным ключом, атрибуты название и телефон должны быть уникальными.

Сущность пользователь связывается типом связи «один-ко-многим» сразу с двумя сущностями: это сущность избранные товары пользователя для возможности сохранения понравившихся товаров, а также сущность заказ, содержащая в себе характеристику код пользователя, сделавшего данный заказ. Атрибут сущности пользователь код пользователя должен быть первичным ключом, атрибуты электронная почта, адрес проживания, телефон должны быть уникальными.

Дальнейшей сущностью для установления связи выберем сущность заказ. Помимо уже полученной связи с сущностью пользователь, данную сущность необходимо связать с сущностью товары в заказе, являющую в себе не что иное, как список товаров в заказе и их количество. Связь – «один-ко-многим». Атрибут код заказа должен быть первичным ключом.

Сущность подразделение связываем с сущностью сотрудники при помощи связи «один-ко-многим», т.к. в одном подразделении может работать несколько сотрудников. Атрибут код подразделения должен быть первичным ключом, название подразделения должно быть уникальным.

Сущность должность следовало бы связать с сущностью сотрудник, однако анализ предприятия показал, что некоторые сотрудники занимают

более одной должности. Прямое связывание будет создавать избыточность данных в базе, поэтому следует ввести вспомогательную сущность Должность_Сотрудник и связать её с сущностями должность и сотрудник. Связь – «один-ко-многим». С помощью вспомогательной сущности исключается возможность избыточности данных, а также аномалий, возникающих при работе с сущностью. Атрибут код должности сущности должность должен быть первичным ключом, название должности должно быть уникальным.

Сущность сотрудники уже имеет практически все необходимые связи с другими сущностями, однако все же следует организовать еще одну связь с сущностью заказ, т. к. за состояние заказа отвечает определенный сотрудник предприятия. Связь - «один-ко-многим». Атрибут код сотрудника должен быть первичным ключом, атрибуты телефон и адрес проживания должны быть уникальными.

Все сущности связаны в единую модель информационной системы, определены типы связей между сущностями, описаны ограничения целостности. Проектирование базы данных для автоматизированной системы учёта заказов выполнено. Результат проектирования модели базы данных приведён на рисунке 2.2.

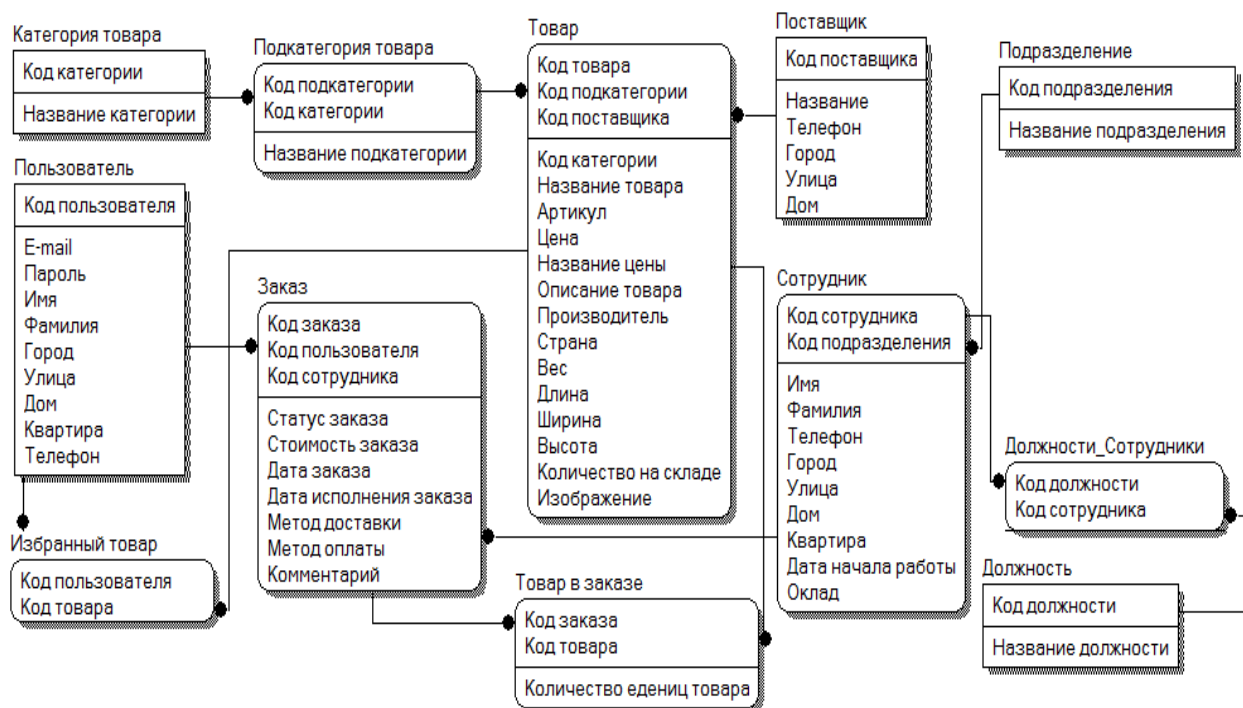


Рис. 2.2. Разработанная схема базы данных.

2.2. Разработка алгоритмов работы автоматизированной системы

Этап разработки алгоритмов работы автоматизированной системы учета заказов включает в себя необходимость выбора вычислительной архитектуры, на которой будет базироваться будущая автоматизированная система[9]. Исходя из имеющихся результатов анализа деятельности предприятия ООО «Рассвет», а также учитывая наличие спроектированной базы данных, самым эффективным вариантом вычислительным архитектуры будет архитектура «клиент-сервер».

В рамках многоуровневого представления автоматизированных систем можно выделить три группы функций, ориентированных на решение различных подзадач:

1. Функции ввода и отображения данных (обеспечивают взаимодействие с пользователем).
2. Прикладные функции, характерные для данной предметной области.

3. Функции управления ресурсами (файловой системой, базой данных и т.д.).

Выполнение этих функций в основном обеспечивается программными средствами, которые можно представить в виде следующих взаимосвязанных компонентов:

1. Компонент представления, отвечающий за пользовательский интерфейс.

2. Прикладной компонент, реализующий алгоритм решения конкретной задачи.

3. Компонент управления ресурсом обеспечивает доступ к необходимым ресурсам и данным.

Автоматизированная система представляет все эти компоненты как на различных уровнях (ОС, служебное ПО и утилиты, прикладное ПО), так и на уровне приложений. Так же и сеть — она представляет все эти компоненты, но, в общем случае, распределенные между узлами. Задача сводится к обеспечению сетевого взаимодействия между этими компонентами. Решением этой задачи является клиент-серверная архитектура.

Архитектура клиент-сервер (англ. client-server) — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Фактически клиент и сервер — это программное обеспечение. Обычно эти программы расположены на разных вычислительных машинах и взаимодействуют между собой через вычислительную сеть посредством сетевых протоколов, но они могут быть расположены также и на одной машине. Программы-серверы ожидают от клиентских программ запросы и предоставляют им свои ресурсы в виде данных (например, работа с базами данных) или в виде сервисных функций (например, работа с электронной почтой). Поскольку одна программа-сервер может выполнять запросы от множества программ-клиентов, её размещают на специально выделенной вычислительной машине, настроенной особым

образом, как правило, совместно с другими программами-серверами, поэтому производительность этой машины должна быть высокой. Из-за особой роли такой машины в сети, специфики её оборудования и программного обеспечения, её также называют сервером, а машины, выполняющие клиентские программы, соответственно, клиентами[10].

Основными преимуществами такой архитектуры являются:

1. Учитывая, что все вычисления автоматизированной системы будут производиться исключительно на стороне сервера, можно отметить, что требования к компьютерам, на которых установлен клиент, существенно снижены.

2. Как правило, в сети с такой архитектурой все необходимые данные хранятся на сервере, что является хорошим показателем безопасности автоматизированной системы в целом, так как обычно сервер защищён гораздо лучше, чем клиент.

3. На сервере проще обеспечивать контроль за полномочиями, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа.

4. Упрощение обслуживания автоматизированной системы.

5. Позволяет объединить различные клиенты. Использовать ресурсы одного сервера часто могут клиенты с разными аппаратными платформами, операционными системами и т. п.

Однако, как и у любой другой вычислительной архитектуры, у архитектуры «клиент-сервер» наблюдаются следующие недостатки:

1. Неработоспособность сервера может сказаться на работе всей автоматизированной системы в целом вплоть до полной её неработоспособности.

2. Поддержка работы автоматизированной работы требует наличия в штате предприятия нескольких специалистов по обеспечению работы сервера – программистов и системных администраторов, что влечёт за собой дополнительные расходы предприятия.

Практические реализации такой архитектуры называются клиент-серверными технологиями. Каждая технология определяет собственные или использует имеющиеся правила взаимодействия между клиентом и сервером, которые называются протоколом обмена (протоколом взаимодействия).

В настоящее время многие приложения, содержащие в основе своей работы рассматриваемую вычислительную архитектуру, используют одну из двух её разновидностей. Различают двухуровневую и многоуровневую архитектуры. Трёхуровневая архитектура является частным случаем многоуровневой архитектуры.

Двухзвенная архитектура используется в клиент-серверных системах, где сервер отвечает на клиентские запросы напрямую и в полном объеме, при этом используя только собственные ресурсы. Т.е. сервер не вызывает сторонние сетевые приложения и не обращается к сторонним ресурсам для выполнения какой-либо части запроса.

Многоуровневые архитектуры, в частности трёхуровневая архитектура, как правило, основываются на двухуровневой архитектуре, однако к существующей двухуровневой архитектуре прибавляется третий уровень.

В большинстве автоматизированных систем, основанных на трёхуровневой архитектуре, третьим уровнем становится сервер приложений, т.е. компоненты распределяются следующим образом[10]:

1. Представление данных — на стороне клиента.
2. Прикладной компонент — на выделенном сервере приложений.
3. Управление ресурсами — на сервере БД, который и представляет запрашиваемые данные.

Трёхуровневая архитектура может быть расширена до многоуровневой путем выделения дополнительных серверов, каждый из которых будет представлять собственные сервисы и пользоваться услугами прочих серверов разного уровня.

Сравнивая данные архитектуры, важно отметить, что двухуровневая архитектура более простая, так как все запросы, поступающие с клиента,

обслуживаются всего лишь одним сервером. Из-за этого двухуровневая архитектура менее надежна, так как предъявляет повышенные требования к общей вычислительной мощности сервера. На фоне этого трехуровневая архитектура является более сложной, однако благодаря распределению функций автоматизированной системы между серверами второго и третьего уровней, эта архитектура обеспечивает высокий уровень гибкости и масштабируемости автоматизированной системы, обладает повышенной безопасностью, обеспечивает высокую производительность автоматизированной системы в целом, т.к. основные задачи будут распределены между серверами разных уровней.

Подводя промежуточный итог всему вышесказанному, принимая во внимание все преимущества и недостатки подобной архитектуры, а также результаты сравнения архитектур разных уровней, следует сделать вывод, что алгоритмы работы автоматизированной системы учёта заказов будут базироваться на клиент-серверной трёхуровневой архитектуре.

Также этап разработки алгоритмов функционирования автоматизированной системы включает в себя проектирование и составление блок-схем основных алгоритмов.

Основываясь на требованиях к автоматизированной системе учета заказов, была разработана основная блок-схема приложения, показывающая процесс формирования заказа клиентом. Данная блок-схема приведена на рис. 2.3.

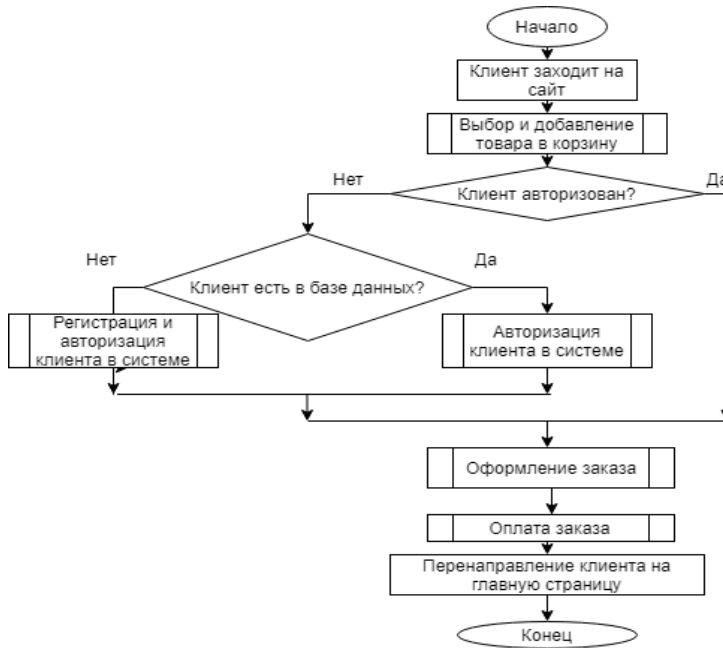


Рис. 2.3. Блок-схема формирования заказа клиентом

Данная блок-схема включает в себя несколько процедур, такие как добавление товара в корзину, регистрация и авторизация клиента, оформление заказа, оплата заказа. Для эффективной разработки также необходимо составить блок-схемы данных процедур.

Первой процедурой, согласно блок-схеме на рис.2.3, является процедура добавления товара в корзину. Блок-схема данной процедуры приведена на рис.2.4.

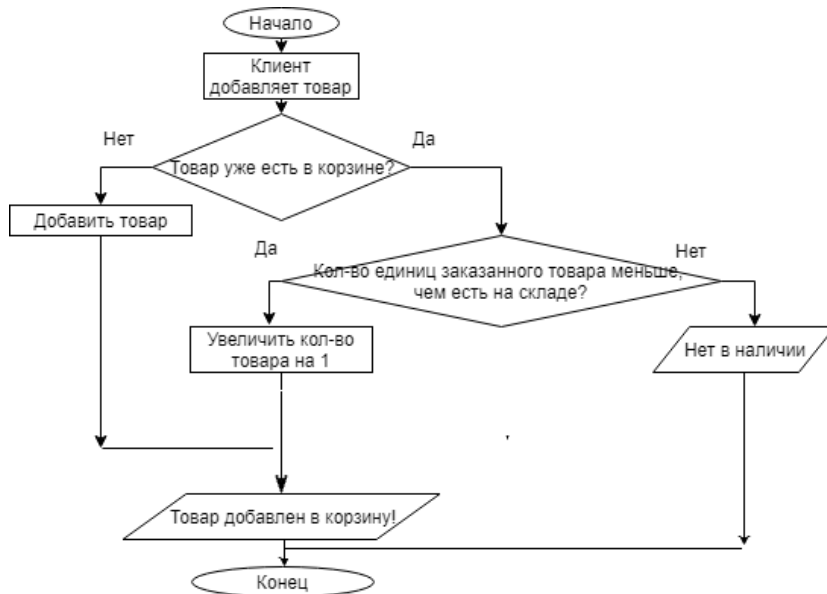


Рис.2.4. Блок-схема процедуры добавления в корзину

Следующей процедурой является процедура регистрации клиента (см. рис. 2.5) в системе с последующей авторизацией (см. рис.2.6).



Рис.2.5. Блок-схема процедуры регистрации клиента

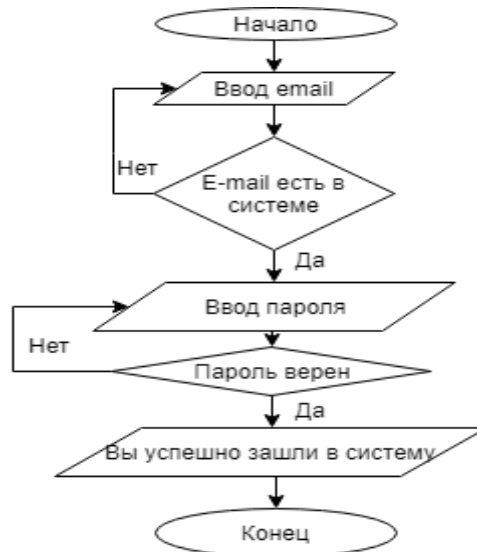


Рис.2.6. Блок-схема процедуры авторизации клиента

Следующей процедурой является процедура оформления заказа. Блок-схема данной процедуры представлена на рис. 2.7



Рис.2.7. Блок-схема процедуры оформления заказа

2.3. Выбор средств разработки автоматизированной системы

Для создания дружелюбного и интуитивно понятного интерфейса автоматизированной системы был выбран язык разметки гипертекста, а именно его самая актуальная на момент разработки системы версия, именуемая HTML5.

HTML5 (англ. HyperText Markup Language, version 5) — язык для структурирования и представления содержимого всемирной паутины. Это пятая версия HTML. Хотя стандарт был завершён (рекомендованная версия к использованию) только в 2014 году, ещё с 2013 года большинством браузеров оперативно осуществлялась поддержка, а разработчиками — использование рабочего стандарта[5].

Для обоснования выбора именно этой версии следует отметить, что во всемирной паутине долгое время использовались стандарты HTML 4.01,

XHTML 1.0 и XHTML 1.1. Веб-страницы на практике оказывались сверстаны с использованием смеси особенностей, представленных различными спецификациями, включая спецификации программных продуктов, например веб-браузеров, а также сложившихся общеупотребительных приёмов. Учитывая эти проблемы, HTML5 был создан как единый язык разметки, который мог бы сочетать синтаксические нормы HTML и XHTML. Он расширяет, улучшает и рационализирует разметку документов, а также добавляет единый API для сложных веб-приложений.

В HTML5 реализовано множество новых синтаксических особенностей. Например, элементы `<video>`, `<audio>` и `<canvas>`, а также возможность использования SVG и математических формул. Эти новшества разработаны для упрощения создания и управления графическими и мультимедийными объектами в сети без необходимости использования сторонних API и плагинов. Другие новые элементы, такие как `<section>`, `<article>`, `<header>` и `<nav>`, разработаны для того, чтобы обогащать семантическое содержимое документа (страницы). Новые атрибуты были введены с той же целью, хотя ряд элементов и атрибутов был удалён. Некоторые элементы были изменены, переопределены или стандартизированы. API и DOM стали основными частями спецификации HTML5. HTML5 также определяет некоторые особенности обработки ошибок вёрстки, поэтому синтаксические ошибки должны рассматриваться одинаково всеми совместимыми браузерами.

Для графического оформления компонентов интерфейса автоматизированной системы учета заказов были выбраны каскадные таблицы стилей.

CSS (англ. Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки HTML[5].

CSS используется создателями веб-приложений для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления

внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.

Для придания интерактивности пользовательскому интерфейсу, а также для осуществления возможности асинхронных запросов к серверу автоматизированной системы был выбран скриптовый язык JavaScript в связке с технологией AJAX и JavaScript-библиотекой JQuery[5].

JavaScript используется в клиентской части веб-приложений: клиент-серверных программ, в которых клиентом является браузер, а сервером — веб-сервер, имеющих распределённую между сервером и клиентом логику. Обмен информацией в веб-приложениях происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются кроссплатформенными сервисами.

JavaScript также используется в AJAX, популярном подходе к построению интерактивных пользовательских интерфейсов веб-приложений, заключающемся в «фоновом» асинхронном обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью и интерфейс веб-приложения становится быстрее, чем это происходит при традиционном подходе (без применения AJAX).

jQuery это библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный программный интерфейс для работы с AJAX.

При организации работы серверной части автоматизированной системы выбор пал на связку технологий серверного языка программирования PHP версии 5.3 с веб-сервером Apache. Язык PHP, и веб-сервер являются свободно распространяемыми программными продуктами, а также на момент разработки автоматизированной системы учета данные технологии были задействованы в большинстве веб-приложений, что показывает их высокую надёжность, удобство и стабильность работы.

PHP – это широко используемый язык сценариев общего назначения с открытым исходным кодом. Другими словами, PHP – это язык программирования, специально разработанный для написания веб-приложений (сценариев), исполняющихся на веб-сервере[4].

Аббревиатура PHP означает Hypertext Preprocessor (Препроцессор Гипертекста). PHP достаточно прост для изучения и предоставляет возможность быстрого создания динамически генерируемых веб-страниц.

Важным преимуществом языка PHP перед такими языками, как языки Perl и Си заключается в возможности создания HTML документов с внедренными командами PHP.

Значительным отличием PHP от какого-либо кода, выполняющегося на стороне клиента, например, JavaScript, является то, что PHP-скрипты выполняются на стороне сервера.

PHP позволяет создавать качественные веб-приложения за очень короткие сроки, получая продукты, легко модифицируемые и поддерживаемые в будущем.

Язык PHP постоянно совершенствуется и ему обеспечено доминирование в области языков веб-программирования, по крайней мере, в ближайшее время.

Главным фактором языка PHP является практичность. PHP должен предоставить программисту средства для быстрого и эффективного решения поставленных задач. Практический характер PHP обусловлен следующими важными характеристиками:

1. Традиционностью (код PHP очень похож на тот, который встречается в типичных программах на Си или Pascal).

2. Простотой (php-код может быть встроен непосредственно в html -код страниц, которые, в свою очередь, будут корректно обрабатываться PHP-интерпретатором).

3. Эффективностью («движок» PHP является транслирующим интерпретатором, что позволяет обрабатывать сценарии с достаточно высокой скоростью).

4. Безопасностью (PHP предоставляет в распоряжение разработчиков и администраторов гибкие и эффективные средства безопасности, которые условно делятся на две категории: средства системного уровня и средства уровня приложения).

5. Гибкостью (PHP может интегрироваться и в JavaScript, WML, XML и другие языки; PHP не содержит кода, ориентированного на конкретный web-сервер в целом является платформенно-независимым языком).

6. Бесплатностью (PHP распространяется бесплатно, причем, с открытыми исходными кодами).

Таким образом, язык PHP является наилучшим выбором для разработки серверной части автоматизированной системы учёта заказов.

При проектировании базы данных в графическом виде было принято решение задействовать программный комплекс ERwin.

ERwin является достаточно популярным, производительным, простым в применении средством для построения и конструирования схем баз

данных, завоевавшим широкое признание по всему информационно-технологическому миру. Программный комплекс ERwin предоставляет высокий показатель продуктивности труда при разработке и сопровождении автоматизированных систем с использованием баз данных[7].

На протяжении всего процесса проектирования базы данных – начиная с логического моделирования требований к информационным объектам, их атрибутам и бизнес-правилам, которые в совокупности определяют базу данных автоматизированной системы, заканчивая процессом оптимизации физической модели в соответствии с заданными требованиями - ERwin позволяет наглядно отобразить структуру, связи, типы, основные элементы базы данных.

ERwin облегчает проектирование баз данных. Для этого достаточно создать графическую ER модель (объект-отношение), удовлетворяющую всем требованиям к данным и ввести бизнес-правила для создания логической модели, которая отображает все элементы, атрибуты, отношения и группировки. Пользователь может расширить возможности Erwin, воспользовавшись уникальной поддержкой пользовательских свойств для ввода в модель любой дополнительной информации, значимой для вашей деятельности.

ERwin поддерживает все наиболее популярные реляционные СУБД, включая Oracle, Microsoft SQL Server, Sybase, DB2 и Informix. Одна и та же модель может быть использована для создания нескольких баз данных или для переноса приложения с платформы одной СУБД на другую.

При создании базы данных на сервере следует выбрать одну из существующих систем управления базами данных (СУБД). Для этого необходимо выделить основные требования к СУБД, с точки зрения создания автоматизированной системы учета заказов:

1. Поддержка реляционной модели баз данных.
2. Поддержка стандарта SQL (Structured Query Language).
3. Достаточное быстродействие.

4. Достаточная надежность.

Выбор СУБД был сужен до двух вариантов: MySQL и PostgreSQL, так как они наиболее совместимы со связкой Apache + PHP. Необходимо сравнить данные продукты, чтобы выбрать наиболее подходящий для разрабатываемой автоматизированной системы.

MySQL обладает следующими преимуществами перед PostgreSQL:

1. MySQL обычно намного превосходит PostgreSQL по скорости работы. Кроме того, в MySQL 4.0 реализован кэш запросов. Он позволяет во много раз увеличить скорость обработки запросов для веб-приложений, на которых преобладают неоднократно повторяющиеся запросы на чтение.

2. По качеству поддержки MySQL также превосходит PostgreSQL. Компания MySQL AB предоставляет высококачественную коммерческую техническую поддержку MySQL с момента появления этой системы на рынке, а у PostgreSQL до самого последнего времени поддержки не было.

3. MySQL оснащен большим количеством API для других языков и поддерживается большим количеством существующих программ, нежели PostgreSQL, что обеспечивает лучшую гибкость и расширяемость автоматизированной системы.

4. Репликация данных MySQL надежна, протестирована и используется в таких крупных проектах, как: Yahoo Finance, Mobile.de, Slashdot[13].

5. Литературы о MySQL вышло значительно больше, нежели о PostgreSQL. Кроме того, литература к PostgreSQL преимущественно на иностранном языке, что вызывает дополнительные трудности с интерпретацией некоторых спорных моментов. Все возможности MySQL детально описаны в документации, так как это является обязательным условием включения новых возможностей в код.

6. MySQL поддерживает больше стандартных функций ODBC, чем PostgreSQL.

7. MySQL обладает значительно более мощной реализацией запросов вида ALTER TABLE[13].

8. В MySQL предусмотрена возможность создания таблиц без транзакций, что необходимо приложениям, требующим максимально возможной скорости работы[13].

9. MySQL может работать с двумя поддерживаемыми транзакции обработчиками таблиц, а именно - InnoDB и BerkeleyDB. Так как все системы поддержки транзакций в разных условиях работают по-разному, это дает разработчику возможность найти наилучшее решение для условий, в которых будет работать его система.

10. В MySQL реализован полнотекстовый поиск, который упрощает выполнение операций, связанных с нахождением нужной строки текста в базе данных, снижая нагрузку на основное приложение.

11. В MySQL имеется возможность работы с несколькими базами через одно соединение (разумеется, в зависимости от привилегий пользователя)[13].

12. Система MySQL с самого начала разрабатывалась в расчете на многопоточность, а PostgreSQL использует процессы. Переключение контекстов и доступ к общим данным несколькими потоками осуществляется значительно быстрее, нежели отдельными процессами. Таким образом MySQL в многопользовательских приложениях получает неплохое преимущество в производительности.

13. MySQL реализована значительно более мощная система привилегий, нежели в PostgreSQL. В то время как PostgreSQL обеспечивает лишь привилегии INSERT, SELECT и UPDATE/DELETE над базой или таблицей, MySQL предоставляет возможность определения полного набора разнообразных привилегий на уровне базы, таблицы и столбца. Кроме того, MySQL позволяет задавать привилегии для комбинаций хост/пользователь.

14. В MySQL используется протокол связи между клиентом и сервером со сжатием данных, что увеличивает производительность системы в условиях низкоскоростных каналов связи.

15. Обновление MySQL проходит достаточно просто и не вызывает лишних трудностей. При модернизации MySQL нет нужды в сохранении и восстановлении данных, что приходится делать при установке большинства обновлений PostgreSQL.

Принимая во внимание перечисленные особенности, можно сделать вывод о том, что система управления базой данных MySQL наиболее удачно подходит для реализации проектируемой автоматизированной системы учета заказов, беря в расчёт как конкурентные преимущества над другими СУБД, так и особенности будущей системы, которые также накладывают свои ограничения на использование программных средств. Данная СУБД является свободно распространяемой. Она отлично подходит и массово используется для сопровождения Web-приложений, так как наиболее тесно взаимодействует с языком программирования PHP, применяемого для разработки данной системы, и обладает необходимыми инструментами, облегчающими создание и администрирование базы данных.

ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА ЗАКАЗОВ

3.1. Создание основных программных модулей автоматизированной системы

В результате анализа деятельности и организационной структуры предприятия ООО «Рассвет», было принято решение разбить автоматизированную систему учёта заказов на два основных программных модуля: модуль пользователя (клиента) и модуль администратора, предназначенного для управления и администрирования автоматизированной системой в целом.

Исходя из требований к автоматизированной системе учета заказов, выявленных в первой главе, в модуле пользователя следует выделить следующие функциональные подмодули:

1. Подмодуль поиска товаров в системе.
2. Подмодуль вывода категорий, подкатегорий и товаров определенной подкатегории.
3. Подмодуль фильтрации товаров по заданным критериям.
4. Подмодуль вывода основной информации об определенном товаре.
5. Подмодуль «Избранное», предназначенный для хранения избранных товаров пользователя.
6. Подмодуль «Сравнение товаров», предназначенный для сравнения товаров по их характеристикам.
7. Подмодуль «Корзина», предназначенный для осуществления заказов пользователем.
8. Подмодуль «Регистрация и вход», предназначенный для осуществления регистрации клиентов предприятия в системе, а также для входа в систему.

9. Для зарегистрированных пользователей выделяется специальный модуль «Личный кабинет», предназначенный для отслеживания клиентами их заказов, для возможности добавления контактной информации пользователя, просмотра избранных товаров, смены пароля.

10. Информационный подмодуль, предоставляющий необходимую информацию пользователям системы.

В модуле администрирования системы следует выделить следующие подмодули:

1. Подмодуль «Вход в панель управления», осуществляющий вход в рабочую зону, состав которой зависит от сотрудника.

2. Подмодуль «Категории» для работы с категориями товаров, предоставляющий возможность просмотра, добавления, изменения, удаления и поиска категорий.

3. Подмодуль «Подкатегории» для работы с подкатегориями товаров, обладающий аналогичным функционалом.

4. Подмодуль «Товары» для работы с товарами, предоставляющий возможность добавления, изменения, удаления, поиска, сортировки товаров, а также возможность быстрого редактирования цены количества на складе определённого товара.

5. Подмодуль «Поставщики» для работы с поставщиками товаров.

6. Подмодуль «Пользователи» для работы с пользователями системы.

7. Подмодуль «Заказы» для работы с заказами.

8. Подмодуль «Сотрудники» для работы с сотрудниками предприятия.

9. Подмодуль «Должности» для работы с должностями предприятия.

10. Подмодуль «Выход», осуществляющий выход из рабочей зоны.

Для организации работы всех вышеперечисленных подмодулей необходимо иметь постоянный доступ к базе данных, спроектированной во второй главе. Для этого следует выполнить процедуру программирования базы данных на стороне SQL-сервера MySQL.

Программирование на стороне MySQL сервера включает в себя следующие этапы:

1. Создание таблиц, соответствующих информационным объектам (сущностям) и их характеристикам (атрибутам).
2. Определение типов и длин атрибутов сущностей.

Создание таблиц осуществляется на основе спроектированной во второй главе базы данных, где каждая таблица представляет собой информационный объект (сущность), а столбцы таблицы – характеристики объектов (атрибуты).

Следующим шагом определим типы и длины характеристик информационных объектов. Результаты этого шага представлены в таблицу типов в приложении 4.

Фрагмент SQL кода создания таблиц и определения типов характеристик представлен на рис. 3.1.

```
CREATE TABLE IF NOT EXISTS `orders` (  
  `id_order` int(11) NOT NULL AUTO_INCREMENT,  
  `id_user` int(11) NOT NULL,  
  `id_worker` int(11) NOT NULL,  
  `status` text NOT NULL,  
  `cost` float NOT NULL,  
  `date_order` date NOT NULL,  
  `Date_delivery` date DEFAULT NULL,  
  `delivery_method` text NOT NULL,  
  `payment_method` text NOT NULL,  
  `comment` text NOT NULL,  
  PRIMARY KEY (`id_order`),  
  KEY `id_user` (`id_user`),  
  KEY `id_worker` (`id_worker`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=15 ;
```

Рис. 3.1. Пример создания таблицы «Заказы»

После завершения всех этапов программирования базы данных на стороне sql-сервера можно переходить к программированию подмодулей, определённых в начале данной главы.

Так как все подмодули оперируют информацией, хранящейся в базе данных, важно обеспечить им возможность подключения к ней. Для этого создадим php-файл, содержащий в себе программный код, обеспечивающий

подключение к базе данных. Так как этот файл единственный и будет использоваться во всех модулях приложения, это обеспечит дополнительную гибкость автоматизированной системе, так как при изменении данных подключения изменения потребует всего лишь один файл.

Для обеспечения требования работы подмодулей как с зарегистрированными, так и с не зарегистрированными пользователями, а также для удобного и понятного интерфейса в процессе программирования подмодулей модуля пользователя был использован механизм сессий, входящий в состав языка php.

Сессии предназначены для хранения сведений о пользователях при использовании нескольких компонентов автоматизированной системы. При использовании сессий данные сохраняются во временных файлах на сервере.

Использование сессий очень удобно и оправдано в таких приложениях как автоматизированная система учета заказов так как, во-первых, необходимо сохранять информацию о пользователях на протяжении нескольких страниц, а, во-вторых, своевременно предоставлять пользователю новую информацию. Фрагмент кода с использованием сессий показан на рис. 3.2.

```
<?php
if (empty($_SESSION['products']))
{
    echo 'Ваша корзина пуста';
}
else {
    echo '<p>Товаров в корзине: ' . $_SESSION['products_incart'] . '<p>';
    echo '<p>Итоговая сумма: ' . $_SESSION['cart_coast'] . ' руб</p>';
    echo '<a class="cart_interface_a" href="cart_interface.php">Перейти к корзине</a>';
}
?>
```

Рис. 3.2. Использование сессий

Данный фрагмент кода проверяет наличие товаров в корзине, а также выводит общую стоимость товаров, если таковые имеются.

Помимо сессий, в программный код подмодулей была внедрена технология AJAX, описанная во второй главе. С помощью данной технологии повышается удобность пользования автоматизированной

системой учета заказов, так как пропадает необходимость постоянно перезагружать страницы web-приложения при использовании какого-либо подмодуля, что соответствует одному из пунктов требований к автоматизированной системе. Данная технология обращается к серверу, используя асинхронные запросы, получает необходимую информацию и предоставляет её пользователю или администратору. Технология AJAX реализована как в подмодулях модуля пользователя, так и в подмодулях модуля администрирования. Фрагмент кода с использованием AJAX показан на рис. 3.3.

```
function add_to_cart(product_id) {  
    $.post("backend/add_to_cart.php", {product_id: product_id}, update_cart);  
    $('.it_in_cart').text("Товар добавлен в корзину").fadeIn(800).delay(1000).fadeOut(800);  
}
```

Рис.3.3. Использование AJAX библиотеки JQuery

Данный фрагмент кода представляет собой функцию на языке Javascript, которая осуществляет добавление товара в корзину. Технология AJAX здесь представлена функцией \$.post() с такими параметрами, как url-адрес, которому будет передан асинхронный запрос, данные в виде объекта, пользовательская функция, которая будет вызвана после ответа сервера.

Следует также отметить, что среди требований было требование к безопасности системы. Так как в системе подразумевается использование подмодулей регистрации и авторизации пользователя, а также множество различных форм, которые могут быть использованы злоумышленниками в качестве средства для взлома автоматизированной системы учета заказов в целом и её базы данных в частности, важно использовать средства, позволяющие предотвратить несанкционированный доступ.

Для обеспечения требования безопасности и исключения подобных ситуаций, были задействованы следующие возможности языков javascript и php:

1. Проверка введенных данных в форму на корректность с помощью регулярных выражений. Фрагмент кода с использованием регулярных выражений представлен на рис. 3.4.

```
var rv_email = /^[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]+\.[a-zA-Z]{1,4}+$/;
if((flag == 1) && !(rv_email.test(login)))
{
    $(".error").css("display","block").html("Некорректный e-mail!");
}
```

Рис.3.4. Использование регулярных выражений

Данный фрагмент кода, написанный на языке javascript, является частью функции регистрации пользователя и представляет собой проверку введенного e-mail на корректность.

2. Использование следующих встроенных функций языка php: функция crypt, производящая шифрование поступившей информации на основе алгоритма создания хэшей DES, а также функции trim(), htmlspecialchars() и stripslashes(), позволяющие исключить запрещенные символы, если таковые будут содержаться в поступившей информации. Фрагмент кода с использованием этих функций представлен на рис. 3.5.

```
session_start();
$login = trim(htmlspecialchars(stripslashes($_POST['login'])));
$password = trim(htmlspecialchars(stripslashes($_POST['password'])));
$name = trim(htmlspecialchars(stripslashes($_POST['name'])));
$surname = trim(htmlspecialchars(stripslashes($_POST['surname'])));
$хash = crypt($password);
```

Рис.3.5. Использование встроенных функций для обеспечения безопасности

Данный фрагмент кода, написанный на языке php, является частью функции регистрации пользователя. В данном фрагменте производится обработка введенной информации и шифрование пароля.

Результаты создания модуля пользователя и модуля администратора автоматизированной системы учета заказов будут представлены в главе 4, совместно с результатами тестирования работы этих модулей.

3.2. Организация внешнего вида страниц веб-приложения

Для организации внешнего вида страниц веб-приложения были разработаны макеты следующих страниц: главная страница модуля пользователя, главная страница модуля администрирования, страница показа списка товаров определённой подкатегории, страница показа определенного товара, внешний вид подмодулей «Избранное», «Сравнение товаров», «Корзина», «Регистрация и вход», «Личный кабинет». С макетами можно ознакомиться в приложении 3.

Основываясь на разработанных макетах модулей автоматизированной системы, было принято решение использовать метод блочной вёрстки страниц, используя доступные средства технологий HTML5 и CSS3. На данный момент блочная верстка является наиболее популярным способом вёрстки страниц веб-приложения. Такая верстка обладает рядом преимуществ:

1. Отделение стиля элементов от кода HTML.
2. Возможность наложения одного слоя на другой – такая возможность во многом облегчает позиционирование элементов.
3. Высокая скорость загрузки страницы.
4. Лучшее индексирование поисковыми системами.
5. Легкость создания визуальных эффектов (выпадающих меню, списков, всплывающих подсказок).

Фрагмент кода блочной вёрстки представлен на рис. 3.6.

```
<div class="search-block">
  <div class="logo"></div>
  <div class="search-field">
    <p class="slogan">Всегда низкие цены!</p>
    <form action="search_good.php" method="POST">
      <input type="text" name="search-field" id="search" class="search-input">
      <input type="submit" name="" value=" ">
      <ul class="search_result" id="search_result"></ul>
    </form>
  </div>
  <div class="contact-phone">
    <p class="slogan">Телефон: <span>(4722) 54-98-59</span></p>
    <p class="slogan">Email: <a href="mailto:pkfsvet@yandex.ru">pkfsvet@yandex.ru</a></p>
  </div>
</div>
```

Рис. 3.6. Использование блочной вёрстки при организации внешнего вида страниц

Немаловажным фактором при организации внешнего вида страниц веб-приложений является выбор цветов. При правильном выборе цветов и их комбинации можно увеличить эффективность работы автоматизированной системы. Так как основная деятельность предприятия ООО «Рассвет» заключается в торговле промышленными и строительными товарами, можно предположить, что основными пользователями веб-приложения будут являться мужчины. Учитывая этот фактор, а также специфику отрасли, были выбраны следующие основные цвета: коричневый, бледно-оранжевый и белый.

3.3. Организация учета заказов пользователей

Одним из важнейших факторов эффективной работы автоматизированной системы учета заказов является правильная организация процедуры учета заказов пользователей системы.

Для решения этой задачи и был введён подмодуль «Корзина». Данный подмодуль предоставляет пользователям удобный и понятный интерфейс с товарами, которые они намереваются приобрести. Подмодуль «Корзина» условно разбивается на несколько компонентов. Первый компонент предназначен для показа выбранных товаров, их цены и количества. Данные характеристики являются динамически обновляемыми, что оказывает положительное влияние на гибкость системы. Используя этот компонент, можно выбрать нужное количество товаров, посмотреть подробное описание товара или же удалить его. Предусмотрено, что клиент не может выбрать количество товара больше, чем есть на складе. При удалении всех товаров работа с подмодулем «Корзина» автоматически завершается. Фрагмент кода данного компонента представлен на рис. 3.7.

```

<?php
if (empty($_SESSION['products']))
{
    echo "<p class='special-offer' style='padding-top: 20px;'>Ваша корзина пуста<p><br>";
    echo "<a class='product_delete' href='index.php'>Вернуться к покупкам</a>";
}
else
{
    echo "<form action='confirm_order.php' type='POST'><div class='cart_interface clearfix'>";
    echo "<p class='special-offer' style='padding-left: 10px;'>Корзина покупок:</p>";
    $_SESSION['weight'] = 0;
    foreach ($_SESSION['products'] as $key=>$value)
    {
        echo "<div class='product_in_cart clearfix'>\t";
        $q="SELECT * FROM `goods` WHERE id_good=".$key;
        $res = $dbn->query($q);
        $product = $res->fetch(PDO::FETCH_ASSOC);
        $_SESSION['weight'] += $_SESSION['products'][$key]['count'] * $product['weight'];
        echo "<img class='product_in_cart_image' src='img/goods/' . $product['image'] . ' width='100px' height='100px' >";
        echo "\n";
        echo "<a href='show_good.php?id_good=' . $product['id_good'] . ' class='product_in_cart_name'> . $product['good'] . '</a>";
    }

    Количество:
    <input
        type="number"
        name = "product_count_<?php echo $key?>"
        min="1" max="<?php echo $product['quantity'];?>"
        id="product_count_<?php echo $key?>"
        value="<?php echo $_SESSION['products'][$key]['count'];?>"
        onchange="update_product_count(<?php echo $key?>, $('#product_count_<?php echo $key?>').val())"
    ><br>
    <!--<span onclick="update_product_count(<?php //echo $key?>, $('#product_count_<?php //echo $key?>').val())">обновить</span-->
    <p style="padding-top: 5px;">
    Общая стоимость:
    <?php
        echo ($_SESSION['products'][$key]['count'] * $_SESSION['products'][$key]['coast']);
        echo " руб";
    >
    </p>
    <br><a class="product_delete" onclick="remove_from_cart(<?php echo $key?>)">Удалить товар</a>

```

Рис.3.7. Программная реализация показа товаров в корзине на языке рhp

Следующие два компонента позволяют выбрать способ получения заказа и способ оплаты заказа. В способе получения заказа учитывается, что общий вес заказа не должен превышать 20 килограмм, в противном случае возможность выбора способа доставки курьером будет недоступна. Фрагмент кода данных компонентов представлен на рис.3.8.

```

<div class="cart_result clearfix">
    <p class='special-offer' style="padding-bottom: 10px;">
        Способ получения заказа:
        <a target="_blank" style="color: #5C392C;" href="delivery.php">[?]</a>
    </p>
    <input class="radio" type="radio" id="delivery-method_1" name="delivery" value="1" checked>
    <label for="delivery-method_1">Пункт выдачи в магазине</label><br>
    <?php
        if($_SESSION['weight'] > 20)
        {
            ?>
            <input class="radio" type="radio" id="delivery-method_2" name="delivery" value="2" disabled="disabled">
            <label for="delivery-method_2">Доставка курьером недоступна, Ваш заказ превышает 20 кг. !</label><?php
        }
        else
        {
            ?>
            <input class="radio" type="radio" id="delivery-method_2" name="delivery" value="2" >
            <label for="delivery-method_2">Доставка курьером</label> <?php
        }
    ?>
</div>

```

Рис.3.8. Программная реализация выбора способа доставки заказа

Последний компонент отображает итоги заказа, такие как количество товаров, общий вес заказа и итоговую сумму. После завершения покупок пользователь подтверждает заказ, заполняя необходимые контактные данные, комментарий к заказу, если потребуется оповестить сотрудников предприятия дополнительной информацией. Также данным подмодулем предусмотрена возможность оформить сделанный заказ на другого человека, родственника или друга в случае, если сам клиент по личным обстоятельствам не имеет возможности получить заказ в ближайшее время.

После подтверждения заказа работа клиента с подмодулем «Корзина» завершается, сам заказ попадает в базу данных автоматизированной системы, а именно в таблицы «Заказы» и «Товары в заказе». Каждый заказ получает уникальный идентификатор, идентификатор пользователя, сделавшего заказ, идентификатор сотрудника, ответственного за заказ, итоги заказа, статус «В обработке» и дату подтверждения заказа. В таблицу «Товары заказа» попадают идентификаторы товаров и их количество. В дальнейшем, используя модуль администрирования, менеджер по продажам собирает заказ, присваивает ему статус «Поступил в пункт выдачи заказов», либо «Поступил в отдел доставки», в зависимости от выбранного способа доставки пользователем.

В пункте выдачи заказ может храниться до 14 дней, на 13 день, если клиент не забрал заказ, система автоматически высылает электронное письмо на указанный адрес, если таковой существует, а в модуле администратора формируется оповещение о том, что необходимо оповестить клиента об истечении срока хранения заказа. В случае если клиент не забирает заказ из пункта выдачи в течение 14 дней, заказ приобретает статус «Просрочен».

После получения и оплаты клиентом заказ приобретает статус «Выполнен», в таблице «Товары» происходит автоматическое обновление атрибута «Количество на складе» в сторону уменьшения на величину количества заказанного товара.

Все данные о заказе хранятся в базе данных согласно Приказа Минкультуры России от 31.07.2007 N 1182 (ред. от 28.04.2011) «Об утверждении Перечня типовых архивных документов, образующихся в научно-технической и производственной деятельности организаций, с указанием сроков хранения».

Подводя итоги реализации автоматизированной системы учёта заказов для ООО «Рассвет», следует отметить, что большинство подмодулей системы были реализованы в полной мере, однако на данный момент подмодуль модуля пользователя «Оплата», отраженный в блок-схеме второй главы (см. рис. 2.7.), отвечающий за процедуру оплаты заказа, не был реализован, так как отделом бухгалтерии предприятия ещё не были подготовлены основные финансовые документы, необходимые для подключения платёжных систем, которые требуются для полноценной работы данного подмодуля.

ГЛАВА 4. ТЕСТИРОВАНИЕ РАЗРАБОТАННОЙ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА ЗАКАЗОВ

Согласно этапам разработки автоматизированной системы, определённым в главе 1, необходимо провести тестирование на основе разработанного приложения с целью выявления возможных ошибок в работе системы и их последующего устранения.

Для проведения тестирования работы функций автоматизированной системы важно наполнить её базу данных тестовыми данными. Тестовые данные будут выбраны и определены случайно. Пример заполненной таблицы приведен на рис. 4.1.













4	Семена цветов		Астра многолетняя микс	<input type="text" value="16"/> руб/шт	Agropi	Россия	<input type="text" value="55"/> шт.	 
5	Семена цветов		Ромашка "Нивяник"	<input type="text" value="16.3"/> руб/шт	Аэлита	Россия	<input type="text" value="55"/> шт.	 
6	Семена цветов		Цинния изящная	<input type="text" value="30"/> руб/шт	Гавриш	Россия	<input type="text" value="110"/> шт.	 
7	Электропилы		Электропила Makita UC405	<input type="text" value="9560"/> руб.	Makita	Япония	<input type="text" value="20"/> шт.	 

Рис. 4.1. Фрагмент заполненной тестовыми данными таблицы товаров

4.1. Тестирование функций модуля пользователя

После заполнения всех таблиц базы данных автоматизированной системы учёта заказов можно переходить к тестированию модуля пользователя. Следует отметить, что тестирование модуля будет производиться путём проверки корректной работы созданных программных подмодулей, определённых в главе 3.

Первым подмодулем в списке является подмодуль поиска товаров в системе. Как следует из списка требований к системе, поиск должен быть «живым», т.е. выдавать результаты уже в процессе ввода поискового запроса в поисковую форму, а также поддерживать стандартный алгоритм поиска. Результат тестирования «живого» поиска представлен на рис. 4.2.

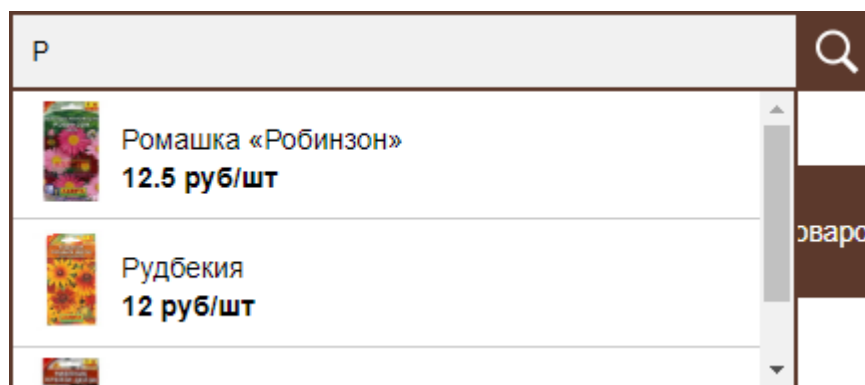


Рис. 4.2. Результат работы «живого поиска»

Можно сделать вывод, что функция «живого» поиска работает корректно. Протестируем стандартный поиск (рис. 4.3.).

Результаты поиска по запросу Ромашка

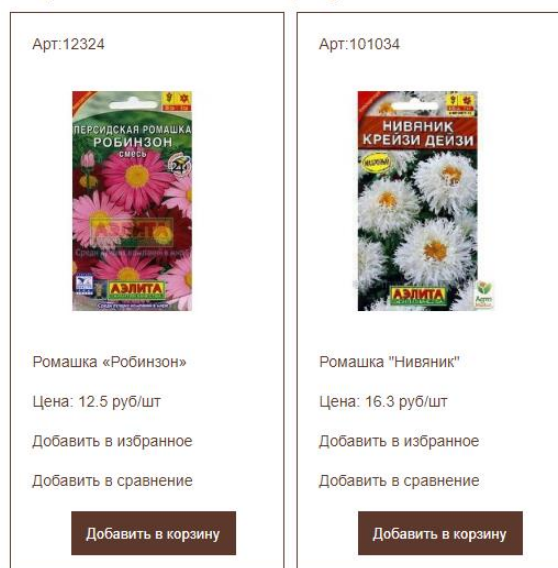


Рис. 4.3 Результат работы стандартного поиска

Подводим промежуточный итог: тестирование первого подмодуля показало, что ошибок в работе нет.

Следующим подмодулем является подмодуль вывода категорий, подкатегорий и товаров определенной категории. Подмодуль должен корректно выводить тестовые категории, подкатегории и товары в подкатегории из базы данных системы. Результаты тестирования представлены на рис. 4.4. и рис 4.5.

Благоустройство участков	»	Семена цветов
Внешняя отделка	»	Семена овощей
Внутренняя отделка	»	Семена газонных трав
Кровля	»	Горшки
Окна и двери	»	Плитка
Осветительное оборудование	»	Садовые фигуры
Строительный инструмент	»	Растения плодовые
Сантехника	»	Растения декоративные

Рис 4.4. Категории и соответствующие им подкатегории.

Товары категории Семена цветов:

<p>Арт:12324</p>  <p>Персидская ромашка РОБИНСОН</p> <p>Ромашка «Робинзон» Цена: 12.5 руб/шт Добавить в избранное Добавить в сравнение Добавить в корзину</p>	<p>Арт:33312</p>  <p>РУДБЕКИЯ ОСЕННИЕ ЦВЕТЫ</p> <p>Рудбекия Цена: 12 руб/шт Добавить в избранное Добавить в сравнение Добавить в корзину</p>	<p>Арт:1000234</p>  <p>ЛАВАНДА ЮЖАНКА</p> <p>Лаванда «Южанка» Цена: 24 руб/шт Добавить в избранное Добавить в сравнение Добавить в корзину</p>
--	---	---

Рис.4.5. Товары определенной категории

Подводим промежуточный итог: тестирование второго подмодуля не выявило никаких ошибок в его работе.

Следующим подмодулем является подмодуль фильтрации товаров по заданным критериям. В качестве заданных критериев используется

выбранный диапазон цены и производитель товара. В качестве небольшого дополнения, подмодуль предоставляет функцию сортировки товаров по возрастанию или убыванию цены. Результаты тестирования представлены на рис. 4.6. и рис. 4.7.

Фильтры:

По возрастанию цены ▾

Диапазон цены

от 15

до 26

Производитель

Азлита [3]

Agroni [2]

Гавриш [1]

Применить фильтры

Рис. 4.6. Подмодуль фильтрации товаров

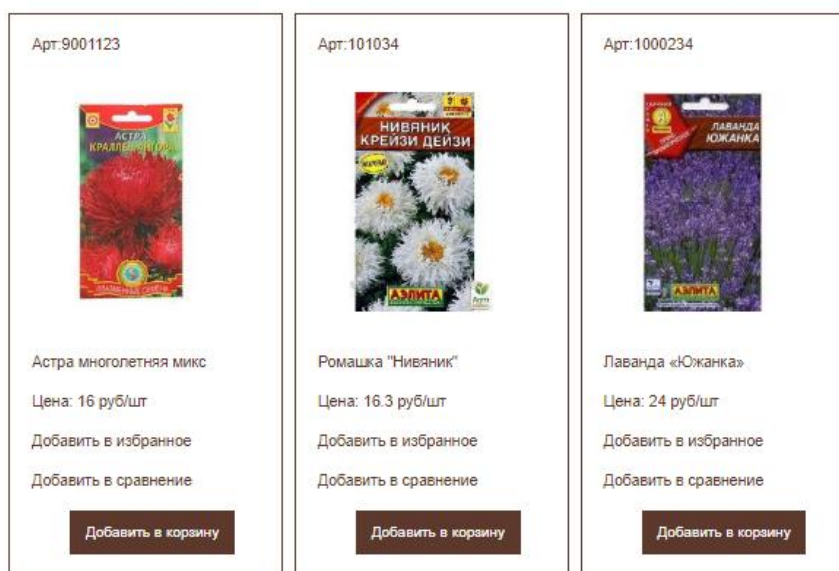



Рис. 4.7. Результаты фильтрации товаров.

Легко видеть, что подмодуль фильтрации товаров по заданным критериям работает успешно: был осуществлен вывод только тех товаров, цена которых попала в диапазон [15- 26], а также производителями которых

были компании «Agroni» и «Аэлита». При этом товары отсортированы по возрастанию цены. Подводим промежуточный итог: подмодуль фильтрации товаров работает корректно.

Следующим в списке подмодулем является подмодуль вывода основной информации о товаре. Его работа заключается в отображении всей информации о товаре, которая содержится в базе данных автоматизированной системы. Результат тестирования можно наблюдать на рис. 4.8.



Ромашка "Нивяник"

(Артикул: 101034)

16.3 руб/шт

Количество на складе: 55 шт

[Добавить в избранное](#)

[Добавить в сравнение](#)

[Добавить в корзину](#)

Данные о цене и наличии товаров обновятся в режиме реального времени. Пожалуйста, уточняйте точную стоимость и наличие товаров в магазине. Они могут отличаться от опубликованных на сайте.
Внимание! Действительный цвет и текстура товаров могут незначительно отличаться от изображений, представленных на сайте.

Описание продукта

Ромашка Нивяник махровой обладает высокими тонкими тепло-всплывающими стеблями и многочисленными расположенными крупными листьями. Растение вырастает до 70 сантиметров. Привычное сочетание белых махровых цветов и желтых трубочек присутствует и здесь. Растение предпочитает яркий солнечный свет и плодородную почву. Цветет в теплое (летнее) время года. Высаживается в групповых композициях, в миксбордерах и рабатках. Отлично подходит для среза и букетов.

Характеристики продукта

Название продукта:	Ромашка "Нивяник"
Артикул:	101034
Компания-производитель:	Аэлита
Страна-производитель:	Россия
Вес:	0.03 кг
Категория продукта:	Семена цветов
Длина:	10 см.
Ширина:	5 см.
Высота:	0.2 см.

Рис. 4.8. Результат вывода основной информации о товаре

Подводим промежуточный итог: тестирование четвертого подмодуля не выявило ошибок в его работе.

Следующим подмодулем является подмодуль «Избранное». Работа данного подмодуля состоит в хранении товаров, помеченных пользователем как избранные. Данный подмодуль позволяет хранить неограниченное число избранных товаров, однако он также не позволяет добавить в себя товар, который уже имеется в списке. Также этот подмодуль позволяет удалять

товары из «Избранного», добавлять их в подмодуль «Корзина» или просматривать их, т.е. происходит непосредственный вызов четвертого подмодуля вывода основной информации о товаре. Результат тестирования работы можно наблюдать на рис. 4.9.

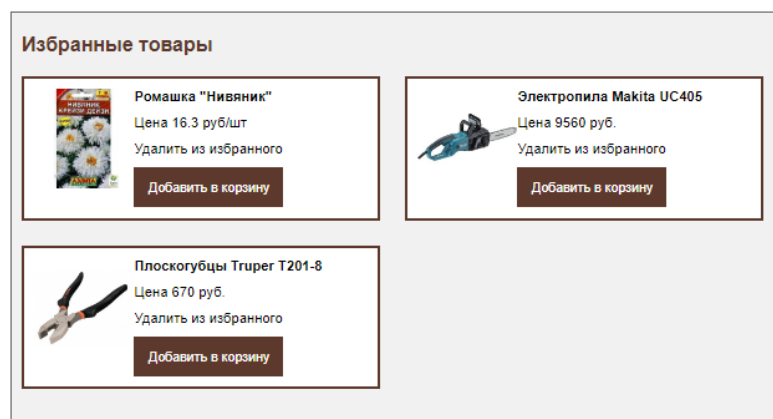


Рис. 4.9. Результат работы подмодуля «Избранное»

Убеждаемся, что пятый подмодуль работает без нареканий.

Следующим подмодулем является подмодуль «Сравнения товаров», позволяющий проводить сравнение характеристик товаров. Его функционал сход с функционалом подмодуля «Избранное», также присутствуют функции удаления, просмотра, добавления в подмодуль «Корзина». Результат тестирования данного подмодуля представлен на рис. 4.10.




	Ромашка «Робинзон»	Рудбекия	Лаванда «Южанка»	Астра много микс
Цена	12.5 руб/шт	12 руб/шт	24 руб/шт	16 руб
Вес	0.015 кг	0.021 кг	0.02 кг	0.035
Производитель	Аэлита	Аэлита	Agrofi	Agrofi
Страна	Россия	Россия	Россия	Росси

Рис. 4.10. Результат работы подмодуля «Сравнение товаров»

Делаем вывод, что шестой подмодуль выполняет свои функции в полном объеме, не вызывая при этом ошибок.

На очереди тестирования один из самых важных подмодулей автоматизированной системы учёта заказов: подмодуль «Корзина». Данный подмодуль представляет собой ни что иное, как список товаров, или заказ; обладает множеством функций, таких как изменение количества товара в заказе, удаление товара из заказа, вызов подмодуля вывода основной информации о товаре, выбор способа доставки и оплаты товара, автоматический подсчёт итогов заказа, переход к оформлению заказа. Результаты тестирования приведены на рис. 4.11.

Корзина покупок:

 <p>Электропила Makita UC405 Количество: <input type="text" value="1"/> Общая стоимость: 9660 руб Удалить товар</p>	 <p>Молоток KRAFTOOL 20280-560 Количество: <input type="text" value="3"/> Общая стоимость: 2370 руб Удалить товар</p>
 <p>Плоскогубцы Truper T201-8 Количество: <input type="text" value="2"/> Общая стоимость: 1340 руб Удалить товар</p>	

Способ получения заказа: [?]

Пункт выдачи в магазине
 Доставка курьером

Способ оплаты заказа: [?]

Наличный расчет
 Банковская карта

Итоги заказа:

Количество товаров: 3
Вес заказа: 9 кг.
Итоговая стоимость: 13270 рублей

Оформить заказ

Рис. 4.11. Результат работы подмодуля «Корзина»

После нажатия пользователем на кнопку «Оформить заказ», подмодуль переходит в режим оформления заказа, показанный на рис. 4.13. Данный режим включает в себя возможность авторизации или регистрации

пользователя для более удобной работы с системой, но не исключает ситуации, если пользователь не имеет желания проходить регистрацию и предоставляет оформить заказ без регистрации (рис.4.12).

Оформление заказа

Вход

Введите E-mail:

Введите пароль:

Войти и продолжить

[Назад к корзине](#)

Регистрация

Введите E-mail:

Введите Ваше имя:

Введите Вашу Фамилию:

Введите пароль:

Подтвердите пароль:

Зарегистрироваться и продолжить

[Продолжить без регистрации](#)

Рис. 4.12. Режим «Оформление заказа» подмодуля «Корзина»

Оформление заказа

Данные покупателя

ИМЯ ФАМИЛИЯ

E-mail Телефон

Получить бюджет другой человек?

Данные заказа

Итоговая сумма заказа: 13270 рублей

Итоговый вес заказа: 5 кг.

Способ доставки: Пункт выдачи в магазине

Способ оплаты: Наличный расчет

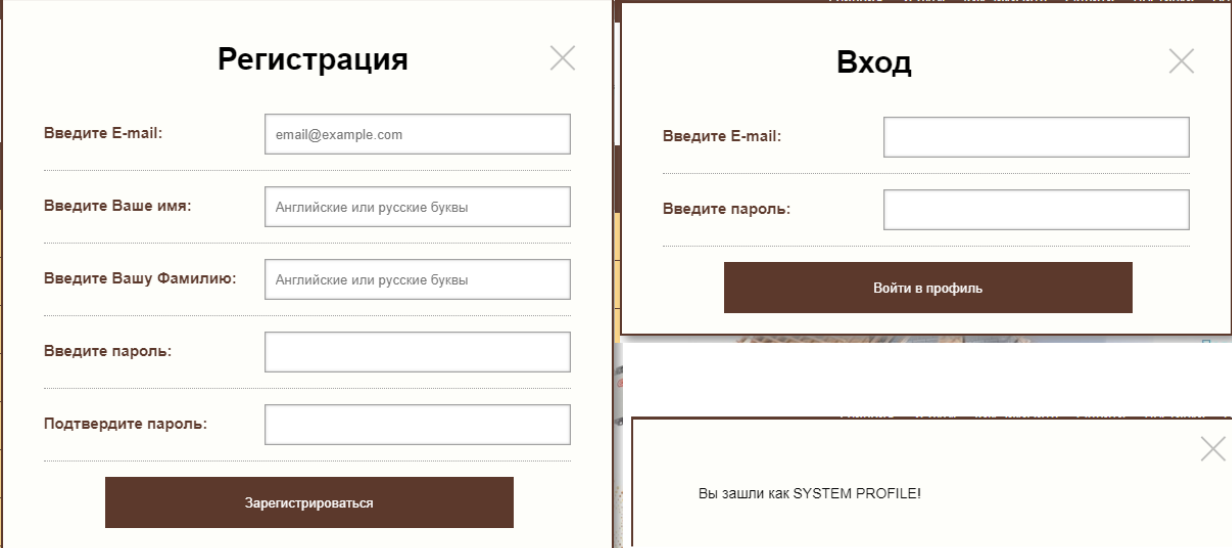
Комментарий к заказу:

Подтвердить заказ

Рис. 4.13. Режим «Подтверждение заказа» подмодуля «Корзина»

Процесс тестирования подмодуля «Корзина» прошёл без ошибок, можно переходить к тестированию следующего подмодуля.

Одним из последних подмодулей является подмодуль «Регистрация и вход», позволяющий осуществлять регистрацию в системе, а также вход в систему для более удобной работы с ней. Результат тестирования представлен на рис. 4.14.



The image shows two overlapping windows from a web application. The left window is titled "Регистрация" (Registration) and contains the following fields: "Введите E-mail:" with the value "email@example.com", "Введите Ваше имя:" with the hint "Английские или русские буквы", "Введите Вашу Фамилию:" with the hint "Английские или русские буквы", "Введите пароль:", and "Подтвердите пароль:". A dark button at the bottom is labeled "Зарегистрироваться". The right window is titled "Вход" (Login) and contains "Введите E-mail:" and "Введите пароль:" fields, with a dark button labeled "Войти в профиль". Below the login window, a separate box displays the message "Вы зашли как SYSTEM PROFILE!".

Рис. 4.14. Результат работы подмодуля «Регистрация и вход»

В процессе тестирования работы данного подмодуля ошибок не обнаружено.

Девятым по счёту подмодулем автоматизированной системы является специальный подмодуль «Личный кабинет», доступный только зарегистрированным пользователям. Его функции состоят в показе пользователю всей необходимой информации сделанных им заказов, возможности добавления контактных данных, просмотра избранных товаров, смены пароля. Результат работы данного подмодуля приведён на рис. 4.15. и рис. 4.16.

Личный кабинет пользователя SYSTEM PROFILE

История ваших заказов ▲

Номер заказа	Сумма заказа	Статус заказа	Дата заказа	Дата выполнения	Способ доставки	Способ оплаты
12 [Подробнее]	10230 руб.	Доставлен	2018-04-25	2018-05-04	Пункт выдачи в магазине	Наличные
14 [Подробнее]	4835 руб.	В обработке	2018-04-25	В процессе исполнения	Доставка курьером	Банковская карта

Избранные товары ▲

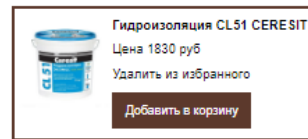
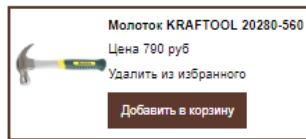


Рис. 4.15. Результат работы подмодуля «Личный кабинет», показ истории заказов и списка избранного

Личные данные ▲

Имя	<input type="text" value="SYSTEM"/>
Фамилия	<input type="text" value="PROFILE"/>
Телефонный номер	<input type="text" value="1111111111"/>
Электронная почта	<input type="text" value="admin"/>
Город	<input type="text" value="TEST_CITY"/>
Улица	<input type="text" value="TEST_STREET"/>
Дом	<input type="text" value="TEST_HOUSE"/>
Квартира	<input type="text"/>
<input type="button" value="Внести изменения"/>	

Сменить пароль ▲

Введите старый пароль	<input type="text"/>
Введите новый пароль	<input type="text"/>

Рис. 4.16. Результат работы подмодуля «Личный кабинет», личные данные и смена пароля

В процессе тестирования работы данного подмодуля ошибок не обнаружено.

Последним в списке является особый информационный подмодуль, компоненты которого распределены по всей автоматизированной системе. Например, с помощью данного подмодуля при оформлении заказа клиент может ознакомиться с существующими способами доставки и оплаты заказа более подробно, узнать об услугах предприятия ООО «Рассвет», прочитать инструкцию заказа товаров и т.д. Большинство компонентов сосредоточены на первой навигационной панели (рис. 4.17.)

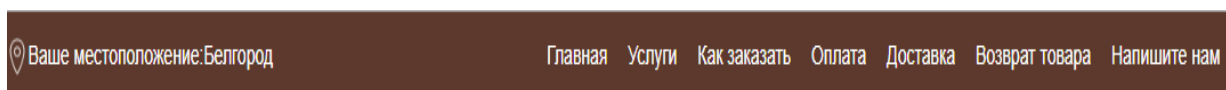


Рис. 4.17. Компоненты информационного подмодуля

Тестирование данного подмодуля не вывило ошибок в его работе.

На этом подмодуле тестирование функций модуля пользователя можно считать завершённым.

Как конечный результат, стоит отметить, что ни в одном из подмодулей не было найдено ошибок в их работе, из чего можно сделать вывод, что все функции модуля пользователя работают корректно.

4.2. Тестирование функций модуля администрирования

Тестирование функций модуля администрирования веб-приложения также будем производить, руководствуясь списком подмодулей, определённым в главе 3.

Первым подмодулем для тестирования является подмодуль «Вход в панель управления». Особенность данного модуля заключается в том, что он предоставляет разный уровень доступа к автоматизированной системе учёта заказов, зависящий от сотрудника предприятия, который начинает работу с

системой в данный момент. Так, введем тестовых сотрудников и назначим им права:

1. Администратор. Пользователь, обладающий всеми правами в отношении автоматизированной системы. В его возможности, помимо общедоступных функций (поиск и сортировка), входят функции добавления, обновления и удаления данных из любой таблицы базы данных. Состав рабочей зоны этого пользователя представлен на рис. 4.18.

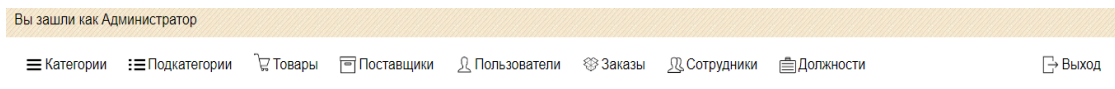


Рис. 4.18. Вид рабочей зоны администратора

2. Кладовщик. Человек, работающий с товарами. В его полном распоряжении могут находиться таблицы категорий, подкатегорий, товаров и поставщиков, остальные таблицы ему не доступны. Состав рабочей зоны этого пользователя представлен на рис. 4.19.

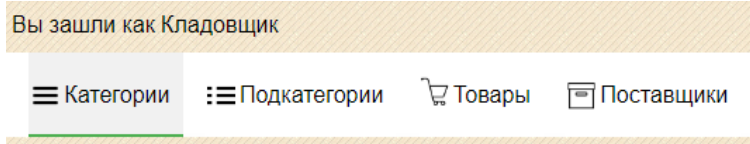


Рис. 4.19. Вид рабочей зоны кладовщика

3. Менеджер по продажам. В часть обязанностей этого пользователя входит обязанность осуществлять сбору заказов. Следовательно, предоставим ему права просматривать такие таблицы как товары, заказы, пользователи. Также предоставим ему возможность просмотра списка товаров определенного заказа. Состав рабочей зоны этого пользователя представлен на рис. 4.20.

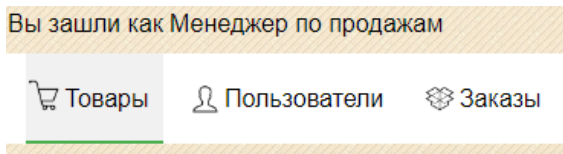


Рис. 4.20. Вид рабочей зоны менеджера по продажам

4. Курьер. Основная обязанность этого пользователя – развозить заказы по адресам заказчиков. Предоставим ему возможность просматривать таблицы заказов и пользователей. Состав рабочей зоны этого пользователя представлен на рис. 4.21.

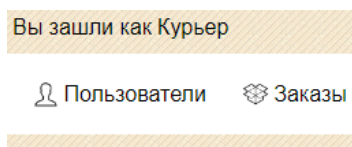


Рис. 4.21. Вид рабочей зоны курьера

Для полного тестирования функций модуля администрирования будет достаточно использования пользователя Администратор. Общий вид первого подмодуля «Вход в панель управления» приобретает вид, представленный на рис. 4.22.

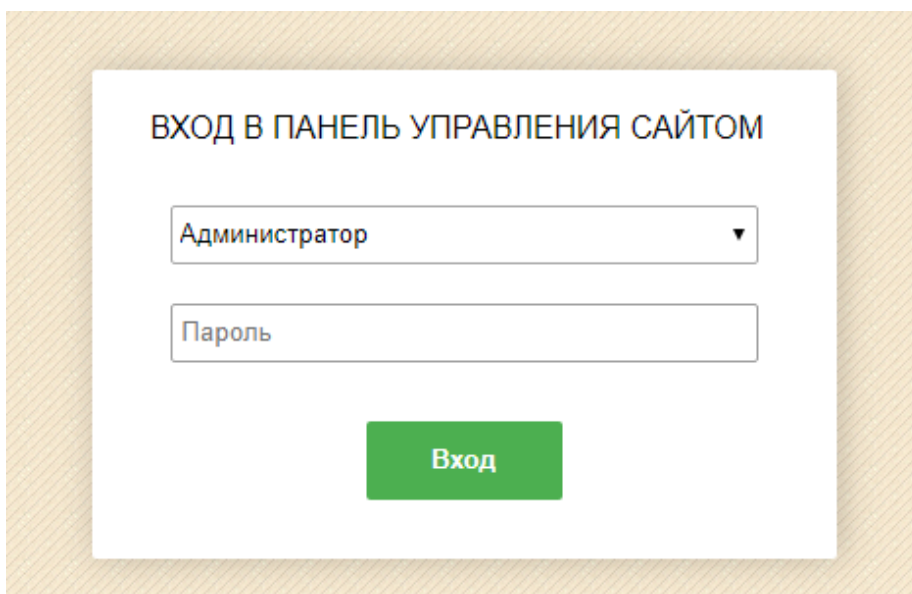


Рис.4.22. Общий вид подмодуля «Вход в панель управления»

Тестирование данного подмодуля показало, что вход для всех тестовых пользователей осуществляется корректно, с правильным распределением прав.

Следующим подмодулем для тестирования, руководствуясь списком, будет подмодуль «Категории». Он предоставляет возможность просмотра, поиска, добавления, обновления и удаления категорий товаров (рис .4.23).

Номер категории	Название категории	Действия
1	Благоустройство участков	✎ ■
2	Внешняя отделка	✎ ■
3	Внутренняя отделка	✎ ■
4	Кровля	✎ ■
5	Окна и двери	✎ ■

Рис. 4.23. Результат вывода таблицы «Категории»

В ходе тестирования подмодуля «Категории» ошибок выявлено не было. Добавление, редактирование, удаление, просмотр и поиск данных работают корректно.

На очереди тестирования подмодуль «Подкатегории». Данный подмодуль также предоставляет возможность просмотра, добавления, обновления, удаления и поиска данных. Процесс тестирования аналогичен. Результат тестирования представлен на рис.4.24.

Номер подкатегории	Название категории	Название подкатегории	Действия
1	Благоустройство участков	Семена цветов	✎ ■
2	Благоустройство участков	Семена овощей	✎ ■
3	Благоустройство участков	Семена газонных трав	✎ ■

Рис. 4.24. Результат тестирования подмодуля «Подкатегории»

В ходе тестирования подмодуля «Подкатегории» ошибок выявлено не было. Добавление, редактирование, удаление, просмотр и поиск данных работают корректно.

Подмодуль «Товары» обладает большим функционалом, нежели рассмотренные ранее подмодули. Помимо стандартных функций добавления, обновления и т.д. данный подмодуль позволяет производить сортировку товаров по цене, названию или уникальному идентификатору, а также динамично изменять цену и наличие товара. Протестируем эти возможности. Результат тестирования представлен на рис. 4.25.

Товары магазина
Всего товаров: 11 шт.

Добавить новый товар По возрастанию цены Поиск по названию...

Номер товара	Название подкатегории	Изображение продукта	Название продукта	Цена продукта	Производитель продукта	Страна продукта	В наличии	Действия
1	Семена цветов		Ромашка «Робинзон»	12,5 руб/шт	Аэлита	Россия	101 шт.	
2	Семена цветов		Рудбекия	12 руб/шт	Аэлита	Россия	50 шт.	

Изменения внесены

Все права защищены

Рис. 4.25. Результат тестирования подмодуля «Товары»

В процессе тестирования данного подмодуля ошибок не обнаружено. Стандартные функции и дополнительные возможности, описанные ранее, работают корректно.

Следующим подмодулем для проведения тестирования выберем подмодуль «Поставщики». Функциональные возможности схожи с подмодулями «Категории» и «Подкатегории». Результат тестирования представлен на рис. 4.26.

Поставщики магазина
Всего поставщиков: 3 шт.

Добавить нового поставщика Поиск по названию...

Номер поставщика	Название поставщика	Телефон поставщика	Адрес поставщика	Действия
1	ООО НТЦ БИО	9103456788	г. Белгород, ул. Буденного, д. 41	
2	КУБАЛА РУС	8915671134	г. Москва, ул. Парковая, д. 6	
4	АргонГруп	8910333675	г. Валуйки, ул. Свердлова, д. 13	

Рис. 4.26. Результат тестирования подмодуля «Поставщики»

В процессе тестирования данного подмодуля ошибок не обнаружено. Стандартные функции, описанные ранее, работают корректно.

Подмодуль «Пользователи» предоставляет функционал для работы с пользователями системы. Обладает следующими функциями: добавление, изменение, удаление, поиск пользователей. Результат тестирования представлен на рис. 4.27.

Пользователи системы
Всего пользователей: 4 шт.

[Добавить нового пользователя](#)

Номер пользователя	E-mail (логин)	Password	Имя	Фамилия	Телефон	Адрес	Действия
31	admin	saEZ6MIWYV9nQ	SYSTEM	PROFILE	11111111111	Нет данных	
32	merchant	saEZ6MIWYV9nQ	SYSTEM	PROFILE	0	Нет данных	
33	collector	saEZ6MIWYV9nQ	SYSTEM	PROFILE	0	Нет данных	
34	courier	saEZ6MIWYV9nQ	SYSTEM	PROFILE	0	Нет данных	

Рис. 4.27. Результат тестирования подмодуля «Пользователи»

В процессе тестирования данного подмодуля ошибок не обнаружено.

Следующий подмодуль «Заказы» несколько отличается от других подмодулей. Данный подмодуль не обладает возможностью добавления заказа, но обладает функцией просмотра списка товаров в заказе. По клику на надпись [Подробнее] подмодуль подгружает информацию о товарах в заказе из базы данных системы и выводит её в отдельной таблице поверх остальных. Протестируем эти возможности, а также стандартные функции обновления и удаления. Результат тестирования подмодуля представлен на рис. 4.28.

Вы зашли к...

Товаров в заказе №12:2

Категория

Плоскогубцы Truper T201-8
Цена за всё: 670 руб.
Количество: 1

Электропила Makita UC405
Цена за всё: 9560 руб.

Заказы
Всего за...

Пользователи Заказы Сотрудники Должности Выход

Номер заказа	E-mail (логин) заказчика	ФИ сотрудника	Статус	Дата заказа	Дата доставки	Цена	Метод доставки	Метод оплаты	Комментарий к заказу	Действия
12 [Подробнее]	admin	Кирилл Жингалов	Доставлен	2018-04-25	2018-05-04	10230 руб.	Пункт выдачи в магазине	Наличные	Без комментариев	
14 [Подробнее]	admin	Сергей Васильев	В обработке	2018-04-25	В обработке	4835 руб.	Доставка курьером	Банковская карта	Подвезти с 18:00	

Рис. 4.28. Результат тестирования подмодуля «Заказы»

В процессе тестирования данного подмодуля ошибок не обнаружено.

Последними двумя подмодулями для проведения тестирования являются подмодули «Сотрудники» и «Должности». Так как они обладают идентичным функционалом (добавление, обновление, удаление), проведем

их тестирование в связке. Результат тестирования соответственно показан на рис. 4.29. и рис. 4.30.

Сотрудники магазина
Всего сотрудников: 7 шт.

[Добавить нового сотрудника](#)

Номер сотрудника	ФИ сотрудника	Телефон сотрудника	Адрес	Дата принятия	Должность	Оклад	Действия
1	Сергей Васильев	89136889032	г. Белгород, ул. Буденного, д. 13Б, кв. 2	2017-10-06	Кладовщик	25000	
2	Кирилл Жингалов	89203109785	г. Белгород, ул. Белгородского полка, д. 45, кв. 14	2017-02-18	Менеджер по заказам	20000	
4	Николай Фомичев	89103900198	г. Белгород, ул. Преображенская, д. 71, кв. 31	2018-05-21	Администратор системы	25000	
3	Александр Володин	8910231456	г. Белгород, ул. Костюкова, д. 45, кв. 12	2017-11-07	Курьер	15000	

Рис. 4.29. Результат тестирования подмодуля «Сотрудники»

Должности
Всего должностей: 5 шт.

[Добавить новую должность](#)

Номер должности	Название должности	Действия
1	Кладовщик	
2	Менеджер по заказам	
3	Администратор системы	
4	Курьер	

Рис. 4.30. Результат тестирования подмодуля «Должности»

На этом подмодуле тестирование функций модуля администрирования можно считать завершенным.

Как промежуточный результат, стоит отметить, что ни в одном из подмодулей не было найдено ошибок в их работе, из чего можно сделать вывод, что все функции модуля администрирования работают корректно.

Подводя конечный итог, можно сказать, что тестирование модуля пользователя и модуля администрирования не выявило ошибок в их работе и позволило полностью удостовериться в корректном функционировании автоматизированной системы учёта заказов, как со стороны обычного пользователя, так и со стороны сотрудника предприятия.

ЗАКЛЮЧЕНИЕ

В результате работы над ВКР была достигнута основная цель – разработка автоматизированной системы учёта заказов для предприятия ООО «Рассвет».

Для достижения поставленной цели были выполнены следующие задачи:

1. Изучена структура и деятельность предприятия ООО «Рассвет», дана общая характеристика предприятию.

2. Изучены соответствующие теме литературные источники.

3. Изучены основные теоретические аспекты создания автоматизированных систем, требования к разработке.

4. Спроектирована автоматизированная система учета, а также база данных для хранения требуемой информации, определены основные алгоритмы работы автоматизированной системы.

5. Изучен программный инструментарий в соответствии с требованиями к системе.

6. Реализована автоматизированная система учёта заказов.

7. В полном объёме проведено комплексное тестирование обоих модулей автоматизированной системы учета заказов.

Последующее внедрение автоматизированной системы учёта заказов с подключенными и настроенными платежными системами должно повысить эффективность использования предприятием имеющихся ресурсов, повысить доходы предприятия, ускорить оборот средств, повысить показатели реализации товаров, сократить расходы, привлечь внимание к предприятию как новых клиентов, так и поставщиков, позволить разгрузить рабочий процесс, сделать сотрудничество с предприятием более продуктивным. Данный проект имеет достаточно короткий срок окупаемости, не требует большого количества ресурсов для обслуживания в

будущем, дополнительного обучения кадров, обладает минимальным набором требований для использования в дальнейшей деятельности.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Зуев Б. М. Организация основного производства предприятий строительных материалов, изделий и конструкций / Б.М. Зуев. – Москва: Проспект науки, 2008. – 224 с.
2. Новицкий, Н.И. Организация, планирование и управление производством / Н.И. Новицкий, В.П. Пашуто.-Москва: Финансы и статистика, 2008. – 576 с.
3. Толстой, А.Д. Технологические процессы и оборудование предприятий строительных материалов. Учебное пособие / А.Д. Толстой, В.С. Лесовик.-Санкт-Петербург: Лань, 2015. – 336 с.
4. Ульман, Л. PHP и MySQL. Создание интернет-магазинов/ Л.Ульман. – Москва: Вильямс, 2017. – 544 с.
5. Прохоренок, Н.А. HTML, Javascript, PHP, MySQL Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. - Санкт-Петербург: БХВ-Петербург, 2015. – 752 с.
6. Грас, Д. Data Science. Наука о данных с нуля/ Д.Грас. – Санкт-Петербург: БХВ-Петербург, 2018. – 336 с.
7. Конноли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика/ Т. Конноли, К. Бегг.- Москва: Вильямс, 2017. – 1440 с.
8. Форта, Б. Освой самостоятельно SQL за 10 минут/ Б.Форта. – Москва: Вильямс, 2017. – 288 с.
9. Смирнов, Ю. А. Технические средства автоматизации и управления. Учебное пособие / Ю. А. Смирнов.- Санкт-Петербург: Лань, 2017. – 456 с.
10. Щербаков, В.В. Автоматизация бизнес-процессов / В.В. Щербаков, А.В. Мерзляк.-Санкт-Петербург: Питер, 2016. – 464 с.

11. Степанов, В. И. Информационные технологии управления продажами и маркетингом / В. И. Степанов.- Санкт-Петербург: LAP Lambert Academic Publishing, 2013. – 284 с.

12. Остроух, А. М. Информационные системы строительных предприятий / А. М. Остроух.- Санкт-Петербург: LAP Lambert Academic Publishing, 2011. – 364 с.

13. Дюбуа, П. MySQL. Сборник рецептов / П. Дюбуа.- Санкт-Петербург: Символ-Плюс, 2004. – 1056 с.

14. Айзенберг, Б. Тестирование и оптимизация веб-сайтов / Б. Айзенберг.- Москва: Вильямс, 2009. – 336 с.

15. Жуков, Ю. Е. Основы веб-хакинга. Нападение и защита. / Ю. Е. Жуков.- Санкт-Петербург: Питер, 2011. – 176 с.

16. Силва, С. Администрирование веб-серверов. / С. Силва.- Санкт-Петербург: КУДИЦ-Образ, 2004. – 688 с.

ПРИЛОЖЕНИЕ 1

SQL КОД БАЗЫ ДАННЫХ, СГЕНЕРИРОВАННЫЙ УТИЛИТОЙ
PHPMYADMIN

```
CREATE TABLE IF NOT EXISTS `categories` (  
  `id_category` int(11) NOT NULL AUTO_INCREMENT,  
  `category` text NOT NULL,  
  PRIMARY KEY (`id_category`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=9 ;  
CREATE TABLE IF NOT EXISTS `claims` (  
  `id_claim` int(11) NOT NULL AUTO_INCREMENT,  
  `claim` text NOT NULL,  
  PRIMARY KEY (`id_claim`),  
  UNIQUE KEY `id_claim` (`id_claim`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;  
CREATE TABLE IF NOT EXISTS `favorite_list` (  
  `id_favorite_list` int(11) NOT NULL,  
  `id_good` int(11) NOT NULL,  
  KEY `id_favorite_list` (`id_favorite_list`,`id_good`),  
  KEY `id_good` (`id_good`),  
  KEY `id_favorite_list_2` (`id_favorite_list`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
CREATE TABLE IF NOT EXISTS `goods` (  
  `id_good` int(11) NOT NULL AUTO_INCREMENT,  
  `id_subcategory` int(11) NOT NULL,  
  `id_provider` int(11) NOT NULL,  
  `good` text NOT NULL,  
  `articul` bigint(20) NOT NULL,  
  `price` float NOT NULL,  
  `price_name` text NOT NULL,  
  `description` text NOT NULL,  
  `producer` text NOT NULL,  
  `country` text NOT NULL,
```



```

`weight` float NOT NULL,
`length` float NOT NULL,
`width` float NOT NULL,
`height` float NOT NULL,
`quantity` int(11) NOT NULL,
`image` text NOT NULL,
`on_offer` int(1) NOT NULL,
PRIMARY KEY (`id_good`),
KEY `id_subcategory` (`id_subcategory`),
KEY `id_provider` (`id_provider`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=12 ;
CREATE TABLE IF NOT EXISTS `goods_in_order` (
`id_order` int(11) NOT NULL,
`id_good` int(11) NOT NULL,
`quantity` int(11) NOT NULL,
KEY `id_order` (`id_order`),
KEY `id_good` (`id_good`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE IF NOT EXISTS `illiquid` (
`id_good` int(11) NOT NULL,
`reason` text NOT NULL,
`date` date NOT NULL,
`quantity` int(11) NOT NULL,
`id_reason` int(11) NOT NULL AUTO_INCREMENT,
PRIMARY KEY (`id_reason`),
KEY `id_good` (`id_good`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=9 ;
CREATE TABLE IF NOT EXISTS `orders` (
`id_order` int(11) NOT NULL AUTO_INCREMENT,
`id_user` int(11) NOT NULL,
`id_worker` int(11) NOT NULL,
`status` text NOT NULL,
`cost` float NOT NULL,
`date_order` date NOT NULL,
`Date_delivery` date DEFAULT NULL,

```

```

`delivery_method` text NOT NULL,
`payment_method` text NOT NULL,
`comment` text NOT NULL,
PRIMARY KEY (`id_order`),
KEY `id_user` (`id_user`),
KEY `id_worker` (`id_worker`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=15 ;
CREATE TABLE IF NOT EXISTS `positions` (
`id_position` int(11) NOT NULL AUTO_INCREMENT,
`name` text NOT NULL,
PRIMARY KEY (`id_position`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=6 ;
CREATE TABLE IF NOT EXISTS `provider` (
`id_provider` int(11) NOT NULL AUTO_INCREMENT,
`name` text NOT NULL,
`phone_number` decimal(10,0) NOT NULL,
`city` text NOT NULL,
`street` text NOT NULL,
`house` text NOT NULL,
PRIMARY KEY (`id_provider`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;
CREATE TABLE IF NOT EXISTS `provider_claim` (
`id_provider` int(11) NOT NULL,
`id_claim` int(11) NOT NULL,
KEY `id_provider` (`id_provider`,`id_claim`),
KEY `id_claim` (`id_claim`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE IF NOT EXISTS `subcategories` (
`id_subcategory` int(11) NOT NULL AUTO_INCREMENT,
`id_category` int(11) NOT NULL,
`subcategory` text NOT NULL,
PRIMARY KEY (`id_subcategory`),
KEY `id_category` (`id_category`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=65 ;
CREATE TABLE IF NOT EXISTS `subdivisions` (

```

```

`id_subdivision` int(11) NOT NULL AUTO_INCREMENT,
`name` text NOT NULL,
PRIMARY KEY (`id_subdivision`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
CREATE TABLE IF NOT EXISTS `users` (
`id_user` int(11) NOT NULL AUTO_INCREMENT,
`id_favorite_list` int(11) NOT NULL,
`login` text NOT NULL,
`password` text NOT NULL,
`name` text NOT NULL,
`surname` text NOT NULL,
`city` text NOT NULL,
`street` text NOT NULL,
`house` text NOT NULL,
`flat` int(11) DEFAULT NULL,
`phone_number` decimal(11,0) NOT NULL,
PRIMARY KEY (`id_user`),
KEY `id_favorite_list` (`id_favorite_list`),
KEY `id_favorite_list_2` (`id_favorite_list`),
KEY `id_favorite_list_3` (`id_favorite_list`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=36 ;
CREATE TABLE IF NOT EXISTS `workers` (
`id_worker` int(11) NOT NULL AUTO_INCREMENT,
`name` text NOT NULL,
`surname` text NOT NULL,
`phone` decimal(11,0) NOT NULL,
`city` text NOT NULL,
`street` text NOT NULL,
`house` text NOT NULL,
`flat` int(11) NOT NULL,
`date_of_receipt` date NOT NULL,
`salary` int(11) NOT NULL,
`id_subdivision` int(11) NOT NULL,
PRIMARY KEY (`id_worker`),
UNIQUE KEY `id_worker` (`id_worker`),

```

```

KEY `id_worker_2` (`id_worker`),
KEY `id_subdivision` (`id_subdivision`),
KEY `id_subdivision_2` (`id_subdivision`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=8 ;
CREATE TABLE IF NOT EXISTS `workers_positions` (
`id_position` int(11) NOT NULL,
`id_worker` int(11) NOT NULL,
KEY `id_position` (`id_position`,`id_worker`),
KEY `id_worker` (`id_worker`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `favorite_list`
ADD CONSTRAINT `favorite_list_ibfk_1` FOREIGN KEY (`id_good`) REFERENCES
`goods` (`id_good`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `favorite_list_ibfk_2` FOREIGN KEY (`id_favorite_list`) REFERENCES
`users` (`id_favorite_list`) ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE `goods_in_order`
ADD CONSTRAINT `goods_in_order_ibfk_1` FOREIGN KEY (`id_order`) REFERENCES
`orders` (`id_order`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `goods_in_order_ibfk_2` FOREIGN KEY (`id_good`) REFERENCES
`goods` (`id_good`) ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE `illiquid`
ADD CONSTRAINT `illiquid_ibfk_1` FOREIGN KEY (`id_good`) REFERENCES `goods`
(`id_good`) ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE `orders`
ADD CONSTRAINT `orders_ibfk_2` FOREIGN KEY (`id_user`) REFERENCES `users`
(`id_user`) ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE `provider_claim`
ADD CONSTRAINT `provider_claim_ibfk_2` FOREIGN KEY (`id_provider`) REFERENCES
`provider` (`id_provider`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `provider_claim_ibfk_3` FOREIGN KEY (`id_claim`) REFERENCES
`claims` (`id_claim`) ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE `subcategories`
ADD CONSTRAINT `subcategories_ibfk_1` FOREIGN KEY (`id_category`) REFERENCES
`categories` (`id_category`) ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE `workers_positions`

```

```
ADD CONSTRAINT `workers_positions_ibfk_1` FOREIGN KEY (`id_worker`)
REFERENCES `workers` (`id_worker`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `workers_positions_ibfk_2` FOREIGN KEY (`id_position`)
REFERENCES `positions` (`id_position`) ON DELETE CASCADE ON UPDATE CASCADE;
```

ПРИЛОЖЕНИЕ 2

ПРОГРАММНЫЙ КОД РАЗРАБОТАННЫХ МОДУЛЕЙ

Файл bd.php

```
<?php
$host='localhost'; $db='dusk_DBA'; $user='root'; $password_bd=""; $dbn = new
PDO("mysql:host=$host;dbname=$db", $user, $password_bd);?>
```

Файл index.php

```
<?php session_start(); include('bd.php'); ?>
<!DOCTYPE html>
<html>
<head>
<title>Рассвет - Главная</title>
<meta charset="utf-8">
<link href="/img/favicon.ico" rel="icon" type="image/x-icon" />
<link rel="stylesheet" type="text/css" href="css/normalize.css">
<link rel="stylesheet" type="text/css" href="css/styles.css">
<script type="text/javascript" src="js/jquery-1.6.4.min.js"></script>
<script type="text/javascript" src="js/jquery-ui.min.js"></script>
<script type="text/javascript" src="js/search.js"></script>
<script type="text/javascript" src="js/functions.js"></script>
</head>
<body>
<div class="it_in_cart"></div>
<div class="registration_block">
<span class="close"></span>
<p class="registration_head">Регистрация</p>
<div id="registration_result">
<div class="input-style clearfix">
<label for="login">Введите E-mail:</label>
<input name="login" type="e-mail" placeholder="email@example.com" id="login"
onkeyup="check_login(this.id)"><span class="login_info"></span>
```

```

<span                class="bad_login"                onmouseover="show_login_info()"
onmouseout="hide_login_info()"></span>
<span                class="good_login"               onmouseover="show_login_info()"
onmouseout="hide_login_info()"></span>
</div>
<div class="input-style clearfix">
<label for="name">Введите Ваше имя:</label>
<input name="name" type="text" id="name" placeholder="Английские или русские буквы">
</div>
<div class="input-style clearfix">
<label for="surname">Введите Вашу Фамилию:</label>
<input name="surname" type="text" id="surname" placeholder="Английские или русские
буквы">
</div>
<div class="input-style clearfix">
<label for="password">Введите пароль:</label>
<input name="password" type="password" id="password" onkeyup="check_passwords()">
</div>
<div class="input-style clearfix">
<label for="password_again">Подтвердите пароль:</label>
<input      name="password_again"      type="password"      id="password_again"
onkeyup="check_passwords()">
<div class="error"></div>
</div><button                class="regbtn"                id="reg_submit"
onclick="registration();">Зарегистрироваться</button>
</div>
</div>
<div class="login_block">
<span class="close" onclick="hide_login();"></span>
<p class="registration_head">Вход</p>
<div id="login_result">
  <div class="input-style clearfix">
    <label for="log_login">Введите E-mail:</label>
    <input name="log_login" type="text" id="log_login" onkeyup="check_log_input(this.id)">
  </div>

```

```

    <div class="input-style clearfix">
    <label for="log_password">Введите пароль:</label>
    <input      name="log_password"      type="password"      id="log_password"
onkeyup="check_log_input(this.id)">
    <div class="log_error"></div></div>
    <button      class="regbtn"      id="log_submit"      onclick="login();">Войти      в
профиль</button></div></div>
<div class="container-center">
<header>
<nav class="clearfix">
<div class="location">
<div class="location-icon"></div>
<p class="city">Ваше местоположение: Белгород</p>
</div>
<ul class="header-list">
<li><a target="_blank" href="index.php">Главная</a></li>
<li><a target="_blank" href="services.php">Услуги</a></li>
<li><a target="_blank" href="howorder.php">Как заказать</a></li>
<li><a target="_blank" href="payment.php">Оплата</a></li>
<li><a target="_blank" href="delivery.php">Доставка</a></li>
<li><a target="_blank" href="goodback.php">Возврат товара</a></li>
<li><a target="_blank" href="writeus.php">Напишите нам</a></li></ul></nav>
<div class="search-block">
<div class="logo"></div>
<div class="search-field">
<p class="slogan">Всегда низкие цены!</p>
<form action="search_good.php" method="POST">
<input type="text" name="search-field" id="search" class="search-input" placeholder="Поиск
товаров" value="" autocomplete="off">
<input type="submit" name="" value=" ">
<ul class="search_result" id="search_result"></ul>
</form>
</div>
<div class="contact-phone">
<p class="slogan">Телефон:<span> (4722) 54-98-59</span></p>

```



```

<p class="slogan">Email:<a href="mailto:pkfsvet@yandex.ru"> pkfsvet@yandex.ru</a></p>
</div>
</div>
<div class="main-navigation">
<ul>
<li class="catalog">Каталог товаров</li>
<li><span class="icon sale"></span><a target="_blank" href="actions.php">Акции</a></li>
<li><span class="icon about"></span><a target="_blank" href="aboutus.php">О
компании</a></li>
<li><span class="icon heart"></span><a target="_blank"
href="favorite_list.php">Избранное</a></li>
<li><span class="icon comprare"></span><a target="_blank"
href="comprare_list.php">Сравнение товаров</a></li>
<li class="container_quantity_in_cart">
<span class="icon basket"></span>
<a target="_blank" href="cart_interface.php">Корзина</a>
<div class="cart_information" id="small_cart">
<?php
if (empty($_SESSION['products'])) {echo 'Ваша корзина пуста;'} else {
echo '<p>Товаров в корзине: ' .$_SESSION['products_incart'].<p>';
echo '<p>Итоговая сумма: ' .$_SESSION['cart_coast']. ' руб<p>';
echo '<a class="cart_interface_a" href="cart_interface.php">Перейти к корзине</a>';} ?>
</div></li><?php if (empty($_SESSION['id'])) { ?>
<li style="max-width: 130px;" onclick="show_registration(); hide_login();">
<span class="icon registration"></span>
<a style="cursor: pointer;">Регистрация</a>
</li><li onclick="show_login(); hide_registration();">
<span class="icon profile"></span>
<a style="cursor: pointer;">Вход</a></li>
<?php }else { ?><li>
<span class="icon profile"></span>
<a href='lk.php'><?php echo $_SESSION['name'] ?></a></li><li>
<span class="icon logout"></span>
<a href="registration/logout.php">Выход</a></li><?php ?></ul></div></header><div
class="content clearfix"><div class="catalog-list"><ul class="category-items">

```

```

<?php $sql="SELECT * FROM categories"; foreach ($dbn->query($sql) as $row) {
    echo '<li class="outside"><a style="cursor: pointer;">'.$row['category'].'</a><div
class="accomplishment"><ul>';
    $sql1="SELECT * FROM subcategories";
    foreach ($dbn->query($sql1) as $row1) {if ($row['id_category']==$row1['id_category']){
        echo '<li><a target="_blank"
href="show_subcategory_goods.php?subcategory='.$row1['id_subcategory'].'">'.$row1['subcate
gory'].'</a></li>'}}echo '</ul></div><span class="right-shevron"></span></li>'}?</ul>
<div class="map"><p class="special-offer">Ha kapre:</p>
<script type="text/javascript" charset="utf-8" async src="https://api-
maps.yandex.ru/services/constructor/1.0/js/?um=constructor%3A5d3835bdd332d0d8ee34b022cc
d5de0ced40b37b277269a8d3ce54c0049c5d6a&amp;float=left&amp;width=270&amp;height=35
6&amp;lang=ru_RU&amp;scroll=true"></script></div></div>
<div class="some-products"><div class="slider"><div class="box-left"><a
id="left_but"></a></div><div class="box-top">
<ul id="slider_list" >
<li><a href="#"></a></li>
<li><a href="#"></a></li>
<li><a href="#"></a></li>
<li><a href="#"></a></li>
<li><a href="#"></a></li>
<li><a href="#"></a></li>
<li><a href="#"></a></li></ul></div>
<div class="box-right"><a id="right_but"></a></div></div>
<p class="special-offer">Специальные предложения:</p>
<?php$sql = "SELECT * FROM `goods` WHERE on_offer =1";foreach ($dbn->query($sql) as
$row){
    echo '<div class="item"><p class="articul">Apr: '.$row['articul'].'</p>';
    echo '';
    echo '<a target="_blank" href="show_good.php?id_good='.$row['id_good'].'" class="item-
name">'.$row['good'].'</a>';
    echo '<p class="price-item">Цена: '.$row['price'].' '.$row['price_name'].'</p>';
    if(!empty($_SESSION['id'])){
        $sql1 = "SELECT COUNT(*) FROM `favorite_list` WHERE
`id_favorite_list`=".$_SESSION['id_favorite']."' AND `id_good`=".$_row['id_good'];

```

```

$res = $dbn->query($sql1);
$result = $res->fetch(PDO::FETCH_ASSOC);
if (empty($result['COUNT(*)'])) {
echo '<p id="favorite'.$row['id_good'].'" onclick="add_to_favor('.$row['id_good'].')">Добавить
в избранное</p>';}
else {echo '<a target="_blank" href="favorite_list.php"
id="favorite'.$row['id_good'].'">Добавлено в избранное </a>';}} else {
if (isset($_SESSION['id_good_to_favor'])){
if (in_array($row['id_good'], $_SESSION['id_good_to_favor'])){
echo '<a target="_blank" href="favorite_list.php" id="favorite'.$row['id_good'].'">Добавлено в
избранное </a>';} else{
echo '<p id="favorite'.$row['id_good'].'" onclick="add_to_favor('.$row['id_good'].')">Добавить
в избранное</p>';} }
else{echo '<p id="favorite'.$row['id_good'].'"
onclick="add_to_favor('.$row['id_good'].')">Добавить в избранное</p>';} }
if (isset($_SESSION['id_good_to_comprare'])){if (in_array($row['id_good'],
$_SESSION['id_good_to_comprare'])){
echo '<a target="_blank" href="comprare_list.php"
id="comprare'.$row['id_good'].'">Добавлено в сравнение </a>';} else{
echo '<p id="comprare'.$row['id_good'].'"
onclick="add_to_comprare('.$row['id_good'].')">Добавить в сравнение</p>';} }
else{echo '<p id="comprare'.$row['id_good'].'"
onclick="add_to_comprare('.$row['id_good'].')">Добавить в сравнение</p>';}
echo '<button onclick="add_to_cart('.$row['id_good'].')" class="btn buy">Добавить в
корзину</button></div>';} ?>
</div></div></div></body></html>

```

Файл cart.php

```

<?php
function add_to_cart($product_id, $count=1) {
if (!empty($_SESSION['products'][$product_id])) {
$_SESSION['products'][$product_id]['count']++;} else {
$_SESSION['products'][$product_id]=array();
$q="SELECT price FROM goods WHERE id_good=".$product_id;
$res = $dbn->query($q);
$add_product = $res->fetch(PDO::FETCH_ASSOC);

```

```

$_SESSION['products'][$product_id]['coast']=$add_product['price'];
$_SESSION['products'][$product_id]['count']=$count;}update_cart();}
function update_cart() {$_SESSION['products_incart']=count($_SESSION['products']);
$_SESSION['cart_coast']=0;
foreach ($_SESSION['products'] as $key=>$value) {
$_SESSION['cart_coast']+=$_SESSION['products'][$key]['coast']*
$_SESSION['products'][$key]['count'];} }
function update_product_count($product_id, $count) {
$_SESSION['products'][$product_id]['count']=$count;
update_cart();}
function remove_from_cart($product_id) {
unset($_SESSION['products'][$product_id]);
update_cart();}
Файл cart_interface.php
<div class="some-products" id="cart_interface">
<?php
if (empty($_SESSION['products']))){
echo "<p class='special-offer' style='padding-top: 20px;'>Ваша корзина пуста<p><br>";
echo '<a class="product_delete" href="index.php">Вернуться к покупкам</a>';} else {
echo '<form action="confirm_order.php" type="POST"><div class="cart_interface clearfix">';
echo "<p class='special-offer' style='padding-left: 10px;'>Корзина покупок:</p>";
$_SESSION['weight'] = 0;
foreach ($_SESSION['products'] as $key=>$value) {
echo "<div class='product_in_cart clearfix'>\t";
$q="SELECT * FROM `goods` WHERE id_good=".$key;
$res = $dbn->query($q);
$product = $res->fetch(PDO::FETCH_ASSOC);
$_SESSION['weight'] += $_SESSION['products'][$key]['count'] * $product['weight'];
echo '';
echo "\n"; echo '<a href="show_good.php?id_good='.$product['id_good'].'"
class="product_in_cart_name">'.$product['good'].'</a>';?>
Количество: <input type="number" name = "product_count_<?php echo $key;?>"
min="1" max="<?php echo $product['quantity'];?>"
id="product_count_<?php echo $key;?>"

```

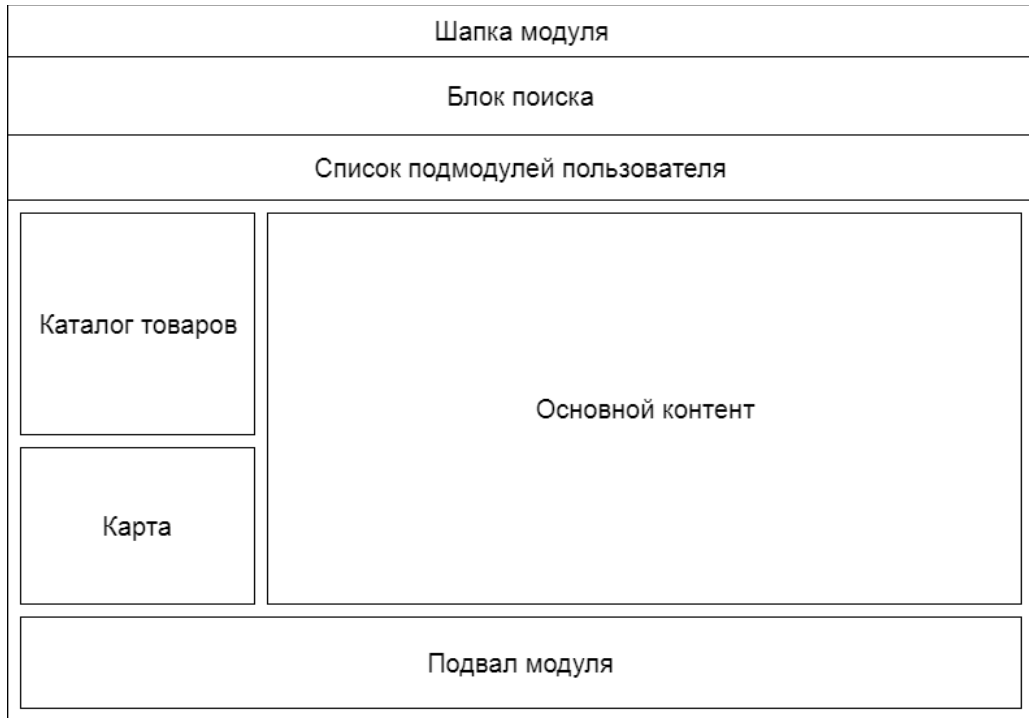
```

value="<?php echo $_SESSION['products'][$key]['count']?>"
onchange="update_product_count(<?php echo $key?>, $('#product_count_<?php echo
$key;?>').val())"
><br> <p style="padding-top: 5px;">Общая стоимость:
<?php echo ($_SESSION['products'][$key]['count']*$_SESSION['products'][$key]['coast']);echo
" руб";?></p><br><a class="product_delete" onclick="remove_from_cart(<?php echo
$key?>)">Удалить товар</a><?php echo "</div>\n\t\t\t\t";?></div><div class="cart_result
clearfix">
<p class='special-offer' style="padding-bottom: 10px;">Способ получения заказа:
</p><input class="radio" type="radio" id="delivery-method_1" name="delivery" value="1"
checked> <label for="delivery-method_1">Пункт выдачи в магазине</label><br>
<?php if($_SESSION['weight'] > 20)
{?><input class="radio" type="radio" id="delivery-method_2" name="delivery" value="2"
disabled="disabled">
<label for="delivery-method_2">Доставка курьером недоступна, Ваш заказ превышает 20
кг. !</label><?php}else{?>
<input class="radio" type="radio" id="delivery-method_2" name="delivery" value="2" >
<label for="delivery-method_2">Доставка курьером</label> <?php}?></div>
<div class="cart_result clearfix">
<p class='special-offer' style="padding-bottom: 10px;">
Способ оплаты заказа:
<input class="radio" type="radio" id="payment-method_1" name="payment" value="1"
checked> <label for="payment-method_1">Наличный расчет</label><br>
<input class="radio" type="radio" id="payment-method_2" name="payment" value="2">
<label for="payment-method_2">Банковская карта</label></div>
<div class="cart_result clearfix">
<p class='special-offer' style="padding-bottom: 10px;">Итоги заказа:</p>
<p style="padding: 5px">Количество товаров: <strong><?php echo
$_SESSION['products_incart'];?></strong></p>
<p style="padding: 5px">Вес заказа: <strong><?php echo $_SESSION['weight'];?>
кг.</strong></p>
<p style="padding: 5px">Итоговая стоимость: <strong><?php echo
$_SESSION['cart_coast']. " рублей";?></strong></p>
<input class="btn buy confirm_btn" type="submit" name="" value="Оформить заказ">
</div><?php}?></form></div>

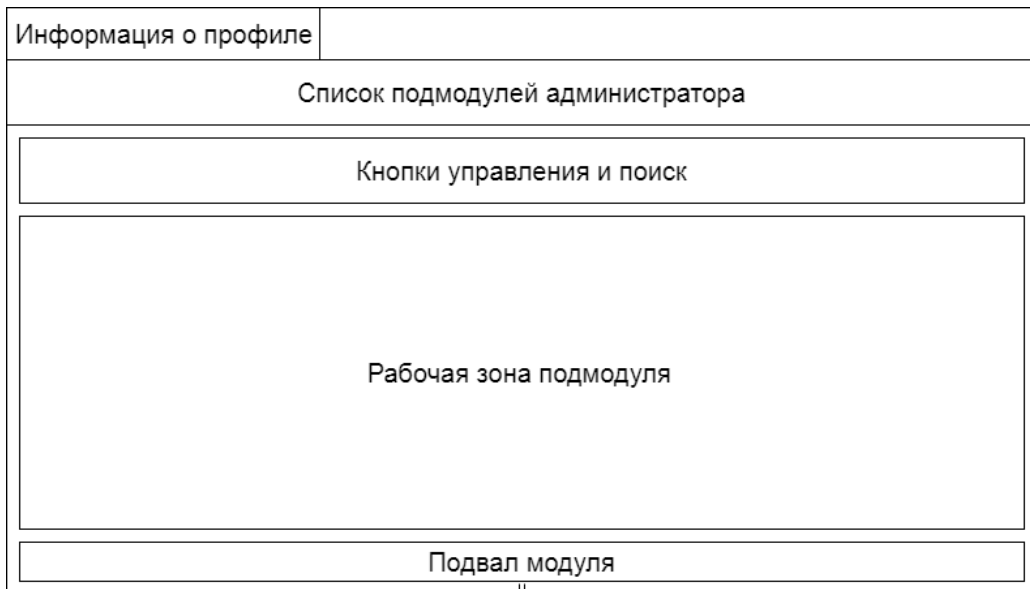
```

ПРИЛОЖЕНИЕ 3

РАЗРАБОТАННЫЕ МАКЕТЫ СТРАНИЦ ВЕБ-ПРИЛОЖЕНИЯ



Макет модуля пользователя



Макет модуля администратора

Список товаров	
Товар1	Товар2
Способ доставки	
Способ оплаты	
Итоги заказа	Оформление заказа (кнопка)

Макет подмодуля «Корзина»

Список избранных товаров	
Товар1	Товар2

Макет подмодуля «Избранное»

	Товар1	Товар2	Товар3
Свойство1	Значение1_1	Значение1_2	Значение1_3
Свойство2	Значение2_1	Значение2_2	Значение2_3
Свойство3	Значение3_1	Значение3_3	Значение3_3

Макет подмодуля «Сравнение товаров»

Личный кабинет пользователя	Регистрация
История заказов	Ввод e-mail
Личные данные	Ввод имени
Смена пароля	Ввод фамилии
	Ввод пароля
	Подтверждение пароля
	Кнопка регистрации

Макеты подмодулей «Личный кабинет» и «Регистрация»

ПРИЛОЖЕНИЕ 4

ТИПЫ ПОЛЕЙ ТАБЛИЦ БАЗЫ ДАННЫХ

Таблица 4.1

Таблица типов

Имя таблицы	Имя поля	Тип	Длина
categories	id_Category	INTEGER	
	Category_name	VARCHAR	30
subcategories	id_subcategory	INTEGER	
	id_Category	INTEGER	
	Subcategory_name	VARCHAR	30
goods	Id_good	INTEGER	
	Id_subcategory	INTEGER	
	Id_provider	INTEGER	
	Name_good	VARCHAR	30
	Articul	INTEGER	20
	Price	INTEGER	7
	Price_name	VARCHAR	8
	Description	TEXT	
	Producer	VARCHAR	30
	Country	VARCHAR	30
	Weight	INTEGER	11
	Length	INTEGER	11
	Width	INTEGER	11
	Height	INTEGER	11
Quantity	INTEGER	5	
provider	Id_provider	INTEGER	
	Name	VARCHAR	30
	Phone_number	DECIMAL	
	City	VARCHAR	30
	Street	VARCHAR	30
	House	VARCHAR	5
users	Id_user	INTEGER	
	E_mail	VARCHAR	50
	Password	TEXT	
	Name	TEXT	
	Surname	TEXT	

Продолжение таблицы 4.1

	City	VARCHAR	30
	Street	VARCHAR	30
	Phone_number	DECIMAL	
	House	VARCHAR	5
Favorite_list	id_user	INTEGER	
	id_good	INTEGER	
Goods_in_order	id_order	INTEGER	
	id_good	INTEGER	
Positions	Id_position	INTEGER	
	Name_position	VARCHAR	30
Position_Workers	id_position	INTEGER	
	id_worker	INTEGER	
Orders	Id_order	INTEGER	
	Id_user	INTEGER	
	Id_worker	INTEGER	
	Status	VARCHAR	15
	Cost	FLOAT	9
	Date_order	DATE	
	Date_perform	DATE	
	Method_delivery	VARCHAR	30
	Method_payment	VARCHAR	30
Subdivisions	Id_subdivision	INTEGER	
	Name_subdivision	VARCHAR	30
Workers	Id_worker	INTEGER	
	Id_subdivision	INTEGER	
	Name	VARCHAR	30
	Surname	VARCHAR	30
	Phone_number	DECIMAL	
	City	VARCHAR	30
	Street	VARCHAR	30
	House	VARCHAR	5
	Flat	VARCHAR	6
	Date_of_receipt	DATE	
	Salary	INTEGER	11
	Id_worker	INTEGER	

Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

__ . __ . ____ Г.

(дата)

(подпись)

(Ф.И.О.)