

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

К защите допустить:

Заведующий кафедрой ЭВМ

_____ Б.В. Никульшин

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
дипломного проекта
на тему

**ИНФОРМАЦИОННЫЙ ПОРТАЛ ПО УПРАВЛЕНИЮ И
РЕАЛИЗАЦИИ ПРОДУКЦИИ КОМПАНИИ**

БГУИР ДП 1 - 40 02 01 019200 ПЗ

Студент

А.С. Лапицкий

Руководитель

А.Г. Буймистров

Консультанты:

от кафедры ИСиТ

Д.В. Горбачев

по экономической части

В.В. Ермоленков

Нормоконтролер

Д.В. Горбачев

Рецензент

Минск 2020

РЕФЕРАТ

Дипломный проект предоставлен следующим образом: 4 чертежа форматом А1, 2 плаката форматом А1, пояснительная записка, включающая: 94 страниц, 36 рисунков, 42 таблицы, 14 литературных источников, 2 приложения.

Цель проектирования: разработка информационного портала по управлению и реализации продукции компании.

Методы исследования: системный, структурно-функциональный подход, изучение источников, сравнение, анализ, обобщение.

Результаты проектирования: выполнен обзор научно-технической литературы, описано функциональное назначение программного средства, описаны аналоги программного средства, разработаны алгоритмы работы, спроектирована логическая и физическая модель базы данных, описана структура программного средства с отображением диаграмм классов, уделено внимание вопросам технико-экономического обоснования, проведено тестирование работоспособности разработанного программного средства, разработана методическая часть проекта.

Анализ экономической эффективности показал целесообразность разработки программного продукта.

Область применения результатов проектирования: программное средство необходимо для облегчения работы компании с заказами в сети интернет, расширения сфер продаж онлайн и ознакомления с продукцией компании.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
1 ПОСТАНОВКА ЗАДАЧИ.....	8
1.1 Описание предметной области.....	8
1.2 Обзор существующих аналогов.....	10
1.3 Информационная база задачи.....	15
1.4 Этапы разработки.....	16
1.5 Функциональное назначение.....	17
2 ПРОЕКТИРОВАНИЕ ЗАДАЧИ.....	18
2.1 Алгоритм решения задачи.....	18
2.2 Логическое моделирование.....	22
2.3 Моделирование базы данных.....	23
2.4 Защита от несанкционированного доступа к данным.....	29
2.5 Моделирование алгоритма работы.....	32
3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ.....	34
3.1 Обоснование выбора инструментов разработки.....	34
3.2 Структура программного средства.....	37
3.3 Описание модулей.....	45
4 ТЕСТИРОВАНИЕ.....	48
5 ПРИМЕНЕНИЕ.....	53
5.1 Назначение и область применения программного средства.....	53
6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.....	54
7 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА.....	61
7.1 Функциональное назначение для пользователя.....	61
7.2 Расчет затрат на разработку программного средства.....	61
7.3 Оценка результата от использования ПС.....	77
ЗАКЛЮЧЕНИЕ.....	80
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	81
ПРИЛОЖЕНИЕ А Код программного средства.....	82
ПРИЛОЖЕНИЕ Б Ведомость документов дипломного проекта.....	94

ВВЕДЕНИЕ

Развитие высоких технологий сказывается положительно на развитии бизнес-индустрии. Важным этапом для компаний в развитии является быстрая реализация товаров – эффективность, а также рост компании. Результаты, как правило, зависят от того, насколько хорошо подана информация и насколько хорошо налажена коммуникация между компанией и клиентом.

Прогрессивное развитие высоких технологий сети интернет привело к видоизменению способов ведения бизнеса и появлению электронного рынка. Производство продукции и продажа через виртуальные магазины получило широкое распространение в виде электронного бизнеса или информационного в глобальной сети интернет. Использование глобальной сети для компаний приобретает стратегическое значение для ведения бизнеса. Проектирование, разработка и внедрение информационной системы является мощным инструментом для информирования целевой аудитории и позиционирования компании на рынке. Именно интернет является источником подробной и свежей информации для человека, на основе которой он может получить представление об интересующей компании, её предоставляемых услугах и продаваемых ею товарах.

Компания, работающая в сфере продаж, в большинстве случаев имеет личный информационный ресурс – сайт, который позволяет усилить положительный образ, а также говорит о надежности и основательности, о имидже компании. Сайт – это не просто площадка по продаже продукции и услуг компании, но это в первую очередь лицо компании. Каждый день все большее количество фирм старается представить свою продукцию в онлайн среде. При этом такое представление не ограничивается лишь созданием промо-сайтов и размещением рекламных баннеров, статей в электронных журналах и на информационных порталах. С развитием интернета развивается и само предложение. Теперь люди могут не только получать интересующую их информацию, но и совершать покупки онлайн с возможностью доставки на дом.

В настоящее время все большую популярность набирает торговля товарами с помощью интернет-магазина. Как правило продажа товаров может осуществляться на сайте, представленном в виде каталога – многостраничника либо в виде лендинга – одностраничного сайта, который выступает в роли витрины магазина. Как правило, одностраничные сайты направлены на привлечение покупателей, выступают визиткой компании. Многостраничные сайты позволяют осуществить реализацию каталога, где количество товара может быть неограниченным, включать разные категории.

Алгоритм интернет-магазинов заключается в последовательности пунктов:

- переход на сайт через внешние ссылки или напрямую;
- ознакомление с продукцией и логикой сайта;

- выбор интересующего товара;
- добавление в корзину;
- оформление заказа;
- выбор способов доставки и оплаты;
- оплата, доставка товара.

Удобство управления сайтом организуется благодаря наличию административной панели, что позволяет контролировать все заказы, управлять продукцией и в целом сайтом – его наполнением. Для гибкости коммуникации вводятся уведомления клиентов через социальные сети, через почту, подключаются внешние модули. Выбор компании в пользу открытия интернет-магазина – это переход на более высокий уровень развития бизнеса, а также предоставление ассортимента наиболее доступным для широкого круга пользователей.

Информационный ресурс имеет ряд преимуществ:

- внешние ссылки, ведущие на сайт;
- доступ к товарам магазина из любой точки мира;
- отсутствие дорогостоящей аренды помещения;
- отсутствие большого количества сотрудников;
- виртуальная площадь не ограничена;
- стоимость намного ниже реального магазина;
- возможность сдачи виртуального пространства и имени прибыли;
- неограниченность в описании товаров и услуг;
- более продуктивная реклама и снижения стоимости в счет уменьшения ненужных ресурсов продвижения;
- режим online магазина позволяет поддерживать обратную связь с клиентами, а также анализировать заинтересованность их в продукте компании;
- покупки не выходя из дома.

В ходе выполнения данного дипломного проекта, в соответствии с заданием, необходимо разработать информационный портал по реализации по управлению и реализации продукции компании. Данная система является информационным порталом, позволяющим просматривать актуальную информацию о товарах компании, а также осуществлять заказ товаров онлайн со стороны пользователя и со стороны администратора – управление сайтом. В данной системе должен быть необходимый перечень товара, а также интуитивно понятный пользователю интерфейс для облегчения работы с приложением, где главная задача – предоставить наиболее ясную информацию для пользователя. В рамках разработки дипломного проекта будет разработано программное средство с базой данных, написана пояснительная записка к дипломному проекту и разработана удобная графическая часть приложения.

1 ПОСТАНОВКА ЗАДАЧИ

1.1 Описания предметной области

Интернет-магазин – это эффективный инструмент торговли, который способен привлечь потенциальных покупателей. Его задача в первую очередь заключается в привлечении покупателей, а далее сподвиг к совершению покупки. Позволяет пользователю сформировать заказ, выбрать способ доставки и оплаты, но самое важное – обеспечить покупку не выходя из дома.

После выбора пользователем товара или услуги предоставляется возможность выбрать способ доставки или оплаты. Совокупность выбранных товаров, способов доставки и оплаты представляют законченный заказ, которые оформляется через форму, путем введения минимально необходимой информации о покупателе.

Информация о покупателе может храниться в базе данных магазина если же бизнес-модель интернет-магазина рассчитана на повторные покупки. В интернет-магазинах, рассчитанных на повторные покупки, также ведется отслеживание возвратов посетителя и автозаполнение данных в форму, ранее заполненную пользователем.

Интернет-магазины подразделяются на два вида, в зависимости от разновидности торговли: магазины, которые продают товар со склада и магазины-посредники, продающие товары других магазинов.

Первые магазины, продающие товары со склада, – хороший вариант дополнительного сбыта продукции, где обычно дают более низкую цену, чем даже реально-существующем магазине. Вторые, которые являются посредниками по продаже товаров других магазинов, занимаются как международной торговлей, так и торговлей внутри страны. В этом случае заработок заключается в комиссии за выставление товара на интернет-площадку. Здесь интернет-магазин выступает в роли гаранта сделки между покупателем и продавцом. Данные магазины используют систему «репутация» продавца. Также, в случае обмана, покупатель может пожаловаться на продавца администрации сайта и получить необходимую помощь по возврату затраченных средств на покупку.

По способу продажи магазины подразделяются: магазины с фиксированной ценой, магазины с системой аукциона. Фиксированная цена товара подразумевает цену с доставкой, которая включена в стоимость, либо с доставкой, которая будет считаться отдельно после оформления. Часто низкая цена товара может быть обусловлена тем, что продавец выставляет цену за доставку, тем самым поднимая общую цену товара. Система аукциона основана на том, когда на товар объявляется аукцион. Кроме начальной цены, продавец может объявить так называемую блиц-цену – это стоимость, за которую продавец готов отдать товар без торга. Нюансом в

данном случае выступает скрытая цена, где продавец ставит заниженную цену на товар, чтобы при поиске товара клиент заметил именно его лот, но включает опцию минимальной ставки на лот – это минимальная цена, которая скрыта от покупателя, он должен повышать ставки, пока не достигнет минимальной цены, иначе ставка не будет принята.

В создании интернет-магазинов применяются системы управления контентом, которые оснащены необходимыми модулями. Крупные же магазины функционируют на специально разработанных для них адаптивных типовых системах управления. Средние и малые магазины используют коммерческое или свободное программное обеспечение.

Представлением системы управления сайтом быть может как продукт из коробки, самостоятельно устанавливаемый на хостинг-площадку, а может быть частной разработкой веб-студии или может быть программным сервисом, предоставляемым с ежемесячной оплатой.

Для администрирования сайтов используется невидимая обычным пользователям часть – бэкенд или серверная часть, на которой расположена административная панель для управления всеми заказами, товарами и услугами, складами. С экономической стороны, для учета потока финансов, применяются и интегрируются специализированные системы учета.

Существуют другие виды веб-сайтов, имеющие иное назначение: коммерческие сайты, сайт-визитка, сайт-каталог.

Коммерческие сайты включают сайты, связанные непосредственно с ведением бизнеса. С этой точки зрения среди них можно выделить: стимулирующие продажи бизнеса в режиме offline, то есть в бизнесе, который сам по себе существует вне зависимости от существования интернета, ориентированные на онлайн-коммерцию, то есть те виды бизнеса, которые без интернета невозможны, например, интернет-магазины. Основу аудитории данных сайтов составляют действительные и потенциальные клиенты.

Коммерческий сайт может быть по своему типу и объему: сайт-визитка, сайт-каталог. Первый содержит сведения о владельце сайта: реквизиты, деятельность, контактные данные. То есть данный сайт представляет собой расширенную карточку предпринимателя или организации. Второй от первого отличается в первую очередь наличием подробного прайс-листа, каталога продаваемой продукции, в котором могут быть перечислены подробные технические характеристики товара.

Интернет-магазин включает каталог товаров или услуг, из которых пользователь может выбрать и осуществить заказ на необходимые ему позиции. Обязательным признаком интернет-магазина является возможность сделать онлайн заказ. Чаще всего эта возможность дополняется функцией «корзина», которая позволяет за один раз заказать несколько понравившихся товаров.

При начинаниях разработки интернет-магазина необходимо определиться с категориями товаров, которые будут реализовываться онлайн. В рамках данного дипломного проекта планируется разработать веб-приложение, реализующее продажи массажной продукции разного типа. Оно будет представлять собой многостраничник, то есть сайт-каталог. Главная страница будет выводить шапку сайта с категориями и контактными данными, обложку с кнопкой, ведущую на страницу определенного товара, а также товары, добавленные последними на сайт. Каждый товар имеет собственную страницу, которая отображает: заголовок, технические характеристики, подробное описание, цену, кнопку добавления в корзину с возможностью быстрого перехода к оформлению и галерею изображений товара. Для управления сайтом будет использована административная панель с разделением прав доступа.

1.2 Обзор существующих аналогов

Важным пунктом при разработке является обзор аналогов сайтов, направленных на торговлю. Целью является сравнение всех выбранных виртуальных магазинов, выявление достоинств и недостатков, заключительная постановка задачи, то есть описание технического задания для сайта, который планируется разработать в данном дипломном проекте.

Критериями для оценки сайта выступают: дизайн, ассортимент и информативность, простота, понятность и удобство для пользователя, наличие обратной связи, возможность заказать товар.

Аналогом разрабатываемого сайта был выбран сайт «Casada» [1]. Главная страница представлена на рисунке 1.1.

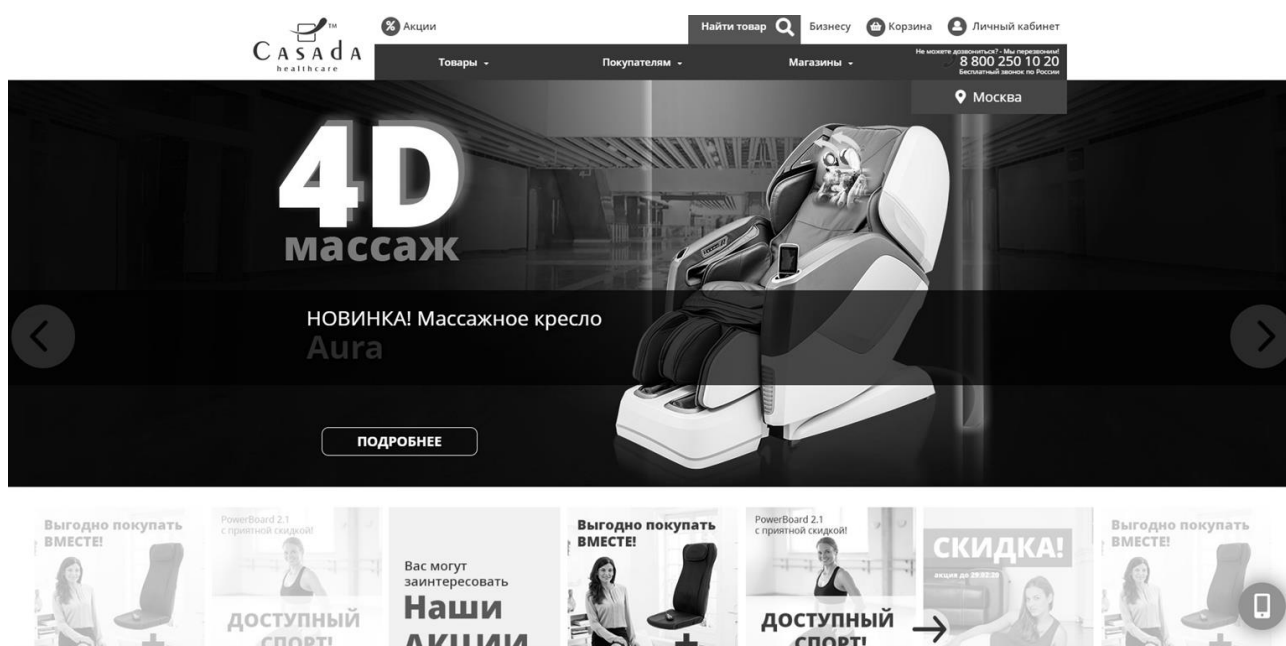


Рисунок 1.1 – Главная страница сайта «Casada»

Сайт представляет собой виртуальный магазин по продаже массажной продукции. На страницах магазина представлен большой ассортимент массажеров одного производителя, так же удобный поиск массажеров, обратная связь, акции, скидки, контактная информация и другие. Данная компания предлагает к покупке разные виды массажного оборудования: массажные кресла, ручные массажеры, массажные подушки, массажные маты и столы, фитнес-приборы, массажеры для ног. Для покупки понравившегося товара можно обратиться к консультанту онлайн, заказать товар через корзину, заполнив необходимую информацию о покупателе, либо позвонив по телефону, указанному на сайте компании.

На главной странице располагается вся основная информация. Главное меню содержит ссылки (см. рисунок 1.2):

- «Акции»;
- «Найти товар»;
- «Бизнесу»;
- «Корзина»;
- «Личный кабинет»;
- «Товары»;
- «Покупателям»;
- «Магазины».

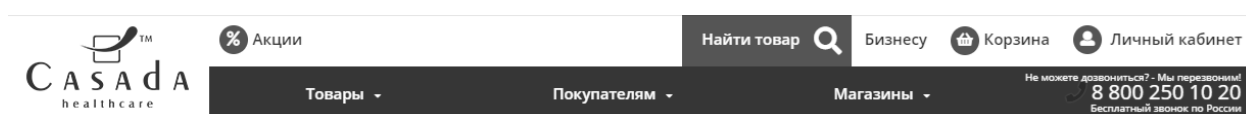


Рисунок 1.2 – Меню сайта casada

В разделе «Акции» содержится информация о всех текущих акциях на имеющийся ассортимент товаров.

По ссылке «Найти товар» можно найти любой товар, который располагается в каталоге товаров.

В разделе «Бизнесу» можно наблюдать информацию об ведении бизнеса в странах СНГ, а также возможность сотрудничества по франшизе с данной компанией.

По ссылке «Корзина» можно перейти в корзину и при необходимости оформить через нее заказ.

По ссылке «Личный кабинет» пользователю предоставляется возможность войти в свой кабинет, просмотреть заказанные товары, а также добавить товары в корзину и запомнить понравившиеся позиции.

В разделе «Товары» представлены все категории товаров с визуальным обозначением для быстрого ориентира пользователя.

В разделе «Покупателям» содержатся подразделы: доставка и оплата, гарантия и сервис, акции, скидки, тест-драйв, бонусы, отзывы, описание

компании, выкуп кресел и массажеров. В данных подразделах содержится вся необходимая информация: информация о времени и способе доставки, и вариантах оплаты заказа, сроках гарантийного обслуживания и условиях сервисного обслуживания, проводимых акциях, актуальных скидках, возможности тестирования продукции, отзывы реальных покупателей, описание деятельности компании, возможности покупке высшего в употреблении массажного оборудования и информация об условиях.

В разделе «Магазины» указана актуальная информация о адресах магазинов и точек продаж продукции компании.

При переходе на страницу одной из категорий, например, массажных кресел (см. рисунок 1.3), попадаем на страницу, где представлены товары из выбранной категории, однако возможности сортировки нет.



Рисунок 1.3 – Отображение категории массажных кресел

Данный сайт имеет много возможностей, для пользователя и располагает огромным размером информации, качественным контентом, так же имеет очень удобную структуру сайта и приятный для пользователей дизайн. Имеется так же дополнительная информация для обычных пользователей. Сайт сделан так, чтобы кроме информации о продукции компании, пользователь получил дополнительную информацию, связанную с данной тематикой и даже больше.

Однако, в счет большого количества информации, пользователь может легко запутаться и ему придется потратить немало времени, чтобы разобраться с контентом. Также медленная загрузка сайта за счет большого

количества объема графической информации может сыграть плохую роль в репутации сервиса. По итогам можно сказать, что данный сайт информативен, но оптимизация сайта нуждается в доработке.

Еще один аналог по продаже массажной продукции «Yamaguchi» [2]. Главная страниц представлена на рисунке 1.4.

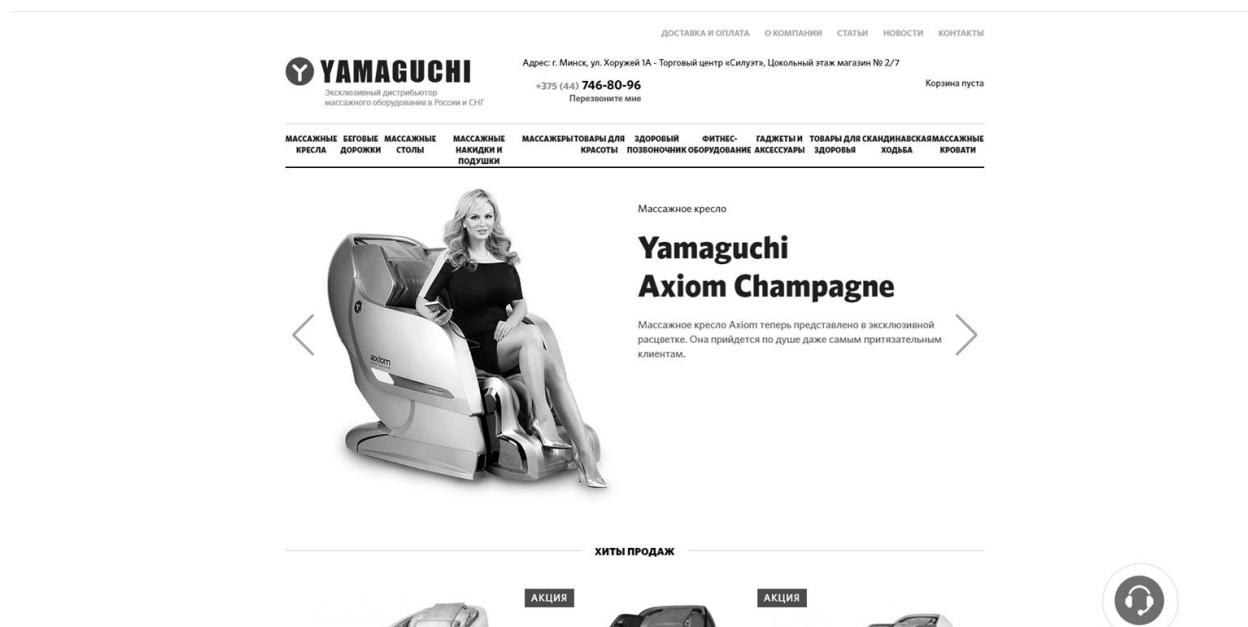


Рисунок 1.4 – Главная страница сайта «Yamaguchi»

Данный сайт также основан на продаже массажной продукции. Сайт располагает широкий ассортимент продукции разных производителей: массажные кресла, ручные массажеры, массажеры для глаз, спины, ног, массажные подушки, массажеры для лица.

Для покупки понравившегося товара можно заказать обратный звонок, либо заказать товар через корзину. На сайте представлена удобная навигация, что позволяет пользователю сразу перейти в нужную категорию. Также на главной странице расположены наиболее популярные товары из разных категорий.

Главное меню сайта (см. рисунок 1.5) содержит ссылки на разные категории массажной продукции.

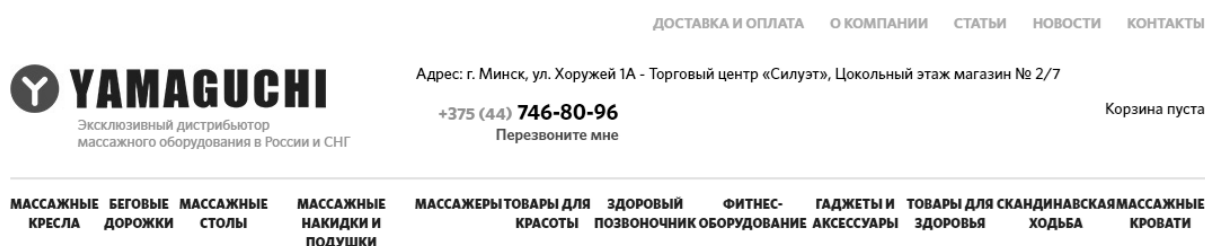


Рисунок 1.5 – Меню сайта yamaguchi

Также оно содержит ссылки на другие страницы:

- «Доставка и оплата»;
- «О компании»;
- «Статьи»;
- «Новости»;
- «Контакты»
- «Корзина».

В разделе «Доставка и оплата» содержится вся необходимая информация для покупателя: время работы курьерской службы, время доставки товаров, стоимость доставки с учетом стоимости заказанного товара, способы доставки и оплаты.

В разделе «О компании» расположена информация о компании, где описана деятельность компании и ее основные направления. Также представлена информация для будущих партнёров – условия, на которых возможно сотрудничество с компанией по франшизе.

В разделе «Статьи» находится информация о пользе массажа и приведены видеоролики о том, как правильный массаж может повлиять на здоровье человека в целом.

В разделе «Новости» отображены все актуальные новости от компании, а конкретнее, какие мероприятия будет проводить компания и какие изменения ее ждут в будущем.

В разделе «Контакты» предоставлена информация: адрес компании и основных действующих магазинов, телефоны и ссылки на имеющиеся социальные сети, а также карта, на которой отображены все магазины компании.

На главной странице также в меню отображены разные категории товаров, они не отображены выпадающим списком, потому как считается, что лучше всего отображать категории таким образом: человек, заходя на сайт, сразу видит все категории и выбирает то, что именно его интересует.

При переходе на одну из категорий массажного оборудования, например, товаров для красоты (см. рисунок 1.6), попадаем на страницу, где размещены все товары выбранной категорией с возможностью сортировки по цене, названию, производителю. Также имеется возможность сбросить фильтр настроек.

ЦЕНА

От 0,00 Р

До 318,78 Р

ПРОИЗВОДИТЕЛЬ

☐ YAMAGUCHI
☐ US MEDICA

ПОДОБРАТЬ

Сбросить все фильтры

Товары для красоты



Сортировать по **умолчанию** [цене](#) [названию](#)

Рисунок 1.6 – Отображение категории товаров для красоты

В целом сайт полностью удовлетворяет потребностям пользователей: он удобный для простых пользователей, имеет хороший дизайн. Имеется вся необходимая информация, которая позволяет быстро сориентировать покупателя, однако скорость работы сайт невелика.

1.3 Информационная база задачи

Информационная база задачи – это определенным способом организованная совокупность данных, которая хранится в памяти вычислительной системы в виде файлов, благодаря которым удовлетворяются информационные потребности управленческих процессов и решаемых задач.

Информационная база задачи подразделяется на: входную, постоянную и выходную информацию задачи.

Входной информацией для информационного наполнения сайта послужили лицензионные и нормативно-правовые документы компании, положение о деятельности, печатные альбомы продукции, прайс-листы оборудования и дополнительная информация по профильному оборудованию из информационных ресурсов интернет. Входные данные представляют собой текстовую и графическую информацию.

Входная информация представлена, как прайс-листы на массажное оборудование с полной характеристикой, фото, описанием и ценой.

Выходная информация сайта представляет собой главную страницу сайта, которая в свою очередь отображает деятельность сайта и содержание, а также отдельные страницы товаров с полным описанием.

1.4 Этапы разработки

Процесс разработки информационного портала делится на несколько этапов: предварительная подготовка к разработке сайта, разработка базы данных, разработка дизайна, создание системы администрирования, создание программных модулей, верстка сайта [3].

Предварительная подготовка подразумевает определение целей разработки и будущего функционального назначения ресурса, проработка логики сайта, а также проектирование базы данных.

Моделирование и разработка базы данных – важнейший этап при разработке сайта. Все данные хранятся именно в базе данных и в будущем, благодаря запросам в базу, осуществляется правильное функционирование веб-приложения.

Создание системы администрирования играет важную роль в наполнении контентом и управлении сайтом, ведь обновление сайта играет большое значение для продуктивности работы и предоставление актуальной информации, а управление позволяет обеспечить тонкости настроек. Как правило, обновление контента на сайте может осуществляться двумя вариантами: благодаря разработчиками либо собственными силами владельца сайта. Первый вариант заключается в передаче информации разработчику и последующего обновления контента на ресурсе. Второй же вариант предусматривает наличие административной панели управления сайтом, которая продумана заранее и позволяет без особых навыков поменять содержание страниц. Как правило, панель администрирования расположена на стороне бэкенд – невидимой для обычных пользователей части веб-приложения.

Разработка программных модулей – дополнительных возможностей, повышающих функциональность ресурса, удобность пользования и позволяют добиваться полной отдачи от сайта.

Разработка дизайна включает не только проработку красочности и сочетаемости цветовой гаммы ресурса и создание фирменного стиля компании, но и продуманное расположение необходимых блоков на страницах, что соответствует логике работы ресурса.

Верстка сайта или другими словами наполнение информацией страниц – это заключительный этап создания сайта. Все заготовленные продукты предыдущих этапов соединяются в один объект – сайт. На базе продуманного шаблона создаются все страницы, которые в последующем наполняются контентом. Верстка предполагает размещение блоков на странице так, как задумано при проектировании. Правильное расположение

блоков дает возможности добиться лучшего понимания и простоты использования пользователем ресурса.

1.5 Функциональное назначение

Для лучшего понимания задач, которые будут решать с помощью сайта необходимо определить цели создания, функциональное назначение. Правильно поставленные цели позволяют сделать сайт наиболее эффективным инструментом для продаж и минимизировать затраты на создание. От продуманности функционального назначения зависит дальнейшая эффективность работы сайта в дальнейшем.

При определении функционального назначения может быть много уровней ее достижения. Например, интернет-магазин вещей, который может предложить информацию о модных вещах для конкретного города, региона или страны, или общие тенденции моды, характерные для разных направлений. Чем больше тематики приведено на сайте, тем больше необходимо разместить информации на нем, что в свою очередь несет затраты. Поэтому функциональное назначение должно быть четко поставленным, конкретным. Для разработчиков сайтов правильно поставленная цель является гарантом правильно разработанного сайта, который будет выполнять все задачи в полном объеме, которые требует заказчик.

Экономия на создании сайта и вложение в раскрутку, рекламу может не привести плодов, ведь неудобность сайта и его дешевизна может отпугнуть клиентов и попросту сайт окажется ненужным. Но с другой стороны, если вложить в грамотный, построенный на базе новейших технологий сайт, функциональность которого только пригодится в будущем, то никому не известный сайт может не привести клиентов.

Преимущество интернет-магазина заключается в том, что здесь гораздо проще хранить товары, чем в реальном магазине. А поддержка обратной связи и высокий уровень сервиса показывает фундаментальное различие в реальных услугах и электронных магазинов. Наличие каталога позволяет реализовать множество категорий [4].

Целями создания данного сайта является организация правильно функционирующего, универсального сайта для товаров или услуг компании, обеспечивающего безотказную работу, отвечающего всем требованиям компании.

Функциональное назначение разрабатываемого сайта заключается в привлечении потенциального покупателя и продаже товаров компании онлайн, сбор заявок на покупку товара и облегчение обработки заказов для персонала.

По итогу реализации, приложение должно иметь следующие функции:

- автоматическая регистрация пользователя – пользователю автоматически присваивается идентификационный номер в сессии,

благодаря которому и происходит его регистрация в системе. В дальнейшем при повторной покупке товара ему нет необходимости повторно прописывать личные данные;

- управление товарами из административной панели – позволяет управлять товарами: удалять, изменять, обновлять, просматривать информацию, отключать видимость на сайте;
- управление заказами из административной панели – позволяет просматривает актуальные заказы на товары и информацию о них, назначать курьера и менеджера для обработки заказа;
- управление складами из административной панели – позволяет: добавлять, изменять, удалять склады, добавлять товары на склад;
- управление клиентской базой из административной панели;
- управление пользователями из административной панели с возможностью задания для каждого пароля;
- подключение к базе данных и выполнение запросов – обеспечение безотказной работы сайта в связке с базой данных при запросах пользователя;
- возможность отслеживания заказа клиентом – позволяет просматривать статус заказа после покупки;
- корзина товаров – отображение списка товаров с возможностью его изменения, формы для оформления заказа и формы оплаты;
- проверка всех полей ввода информации;
- вывод товаров в соответствии с категорией – сортировка товаров определенных категорий в соответствии с типом;
- обрезка фотографий под необходимое разрешение – позволяет выводить на сайте необходимые размеры изображений товаров при их загрузке из административной панели;
- отображение слайдера для товаров – позволяет выводить фотографии в виде слайда, что очень удобно для восприятия визуальной информации;
- каталог товаров с полным описанием, ценой, фотографиями;
- адаптивность, что позволяет выводит информацию ресурса строго в соответствии с шириной экрана устройств;
- разграничение по ролям: менеджер, администратор, курьер – каждый заходит на сайт под своим логином и паролем и каждому доступно только определенная часть интерфейса.

2 ПРОЕКТИРОВАНИЕ ЗАДАЧИ

2.1 Алгоритм решения задачи

Успешная реализация любого приложения связана с тем, насколько правильно написан алгоритм работы и поставлена логика в работе. Работоспособность приложения может оказаться некорректной из-за неправильно написанного алгоритма и неправильной логики, так как может появиться множество ошибок, которые будет трудно выявить в будущем. Правильно построенный алгоритм работы приложения позволит найти наиболее верный путь к реализации. Правильно поставленная логика в работе приложения позволит в дальнейшем избежать недопонимания со стороны пользователя и обеспечит простое управление, а также обслуживание сайта.

В рамках дипломного проекта необходимо разработать действующий и правильно функционирующий информационный портал по управлению и реализации продукции компании. Алгоритм работы приложения будет направлен именно на продажу товаров и постоянное взаимодействие с пользователем, базой данных.

Для покупки товаров пользователем при работе с программным средством, следует произвести следующие действия:

- открыть любой браузер;
- ввести в адресную строку доменное имя сайта;
- после перехода на сайт необходимо ознакомиться с интерфейсом сайта и логикой расположения блоков;
- перейти в интересующую категорию товаров;
- ознакомиться с информацией на сайте об интересующем товаре: подробное описание, фотографии, технические характеристики, цена;
- добавить понравившиеся товары в корзину;
- оформить товар через корзину, заполнив соответствующую форму;
- оплатить онлайн.

Сайт не требует регистрации пользователей и повторного заполнения пользователем данных в форме. При посещении сайта пользователем, для него открывается новая сессия, пользователю присваивается уникальный номер, таким образом пользователь автоматически регистрируется, его данные сохраняются в базу данных с уникальным идентификационным номером: ФИО, номер телефон, адрес электронной почты, домашний адрес. Это позволяет уменьшить затраты времени пользователя на регистрацию и лишнее запоминание паролей. В случае, если пользователь будет еще раз осуществлять покупку, то его данные будут автоматически подставлены в необходимые поля.

2.2 Логическое моделирование

На основании построенной UML-диаграммы прецедентов (см. рисунок 2.1) или, другими словами, вариантов использования, можно описать функциональность и поведение системы, в которой фигурируют: пользователь, администратор, интернет-магазин, менеджер, курьер, склад. На ее основании можно спроектировать наиболее правильную модель будущей базы данных и будущего алгоритма работы программного средства [5].

Логику взаимодействия в системе можно описать следующим образом. Пользователь, который еще не авторизован в системе, является гостем. На данном этапе гость может просматривать содержимое сайта (каталога), переходить по вкладкам и ссылкам, выбирать товар и добавлять в корзину. При добавлении товара в корзину, происходит авторизация или регистрация нового пользователя. Основные прецеденты для пользователя: оформление заказа (добавление заказа в корзину, ввод личных данных, оплата), просмотр статуса заказа, подтверждение заказа, получение заказа. В системе также есть менеджер, который: ведет учет товаров на складе, изменяет статус заказа, работает с заказами, ищет свободного курьера и назначает его. Менеджер является звеном, обеспечивающим обработку поступившего заказа. Курьер обеспечивает доставку товара до клиента (пользователя), получение товаров, входящих в заказ, со склада, а также изменение статуса заказа. Администратор обеспечивает заполнение сайта содержимым, он имеет право на добавление товаров, складов, управление пользователями, просмотр заказов – имеет высокие привилегии по отношению к остальным пользователям сайта. Важным звеном, без которого бы система не функционировала, является интернет-магазин – сайт, подключенный к базе данных, в которой хранятся все необходимые данные. Он непосредственно осуществляет обработку данных и запросов от пользователей, отображение страниц сайта и каталога товаров, осуществление регистрации или авторизации пользователей. Еще одним объектом системы является склад, осуществляющий хранение товаров. На каждом этапе происходит взаимодействие пользователя, гостя, администратора и курьера с интернет-магазином – важнейшим звеном в деятельности. При этом воздействие прецедентов одного действующего объекта может влиять на другого, так, например, изменение статуса заказа менеджером или курьером влияет на просмотр статуса заказа пользователем или, допустим, отмена заказа пользователем либо в случае недоставки курьером заказа, – также происходит смена статуса заказа для пользователя. Воздействия, которые могут повлиять на другие прецеденты отображены штриховыми линиями на диаграмме деятельности.

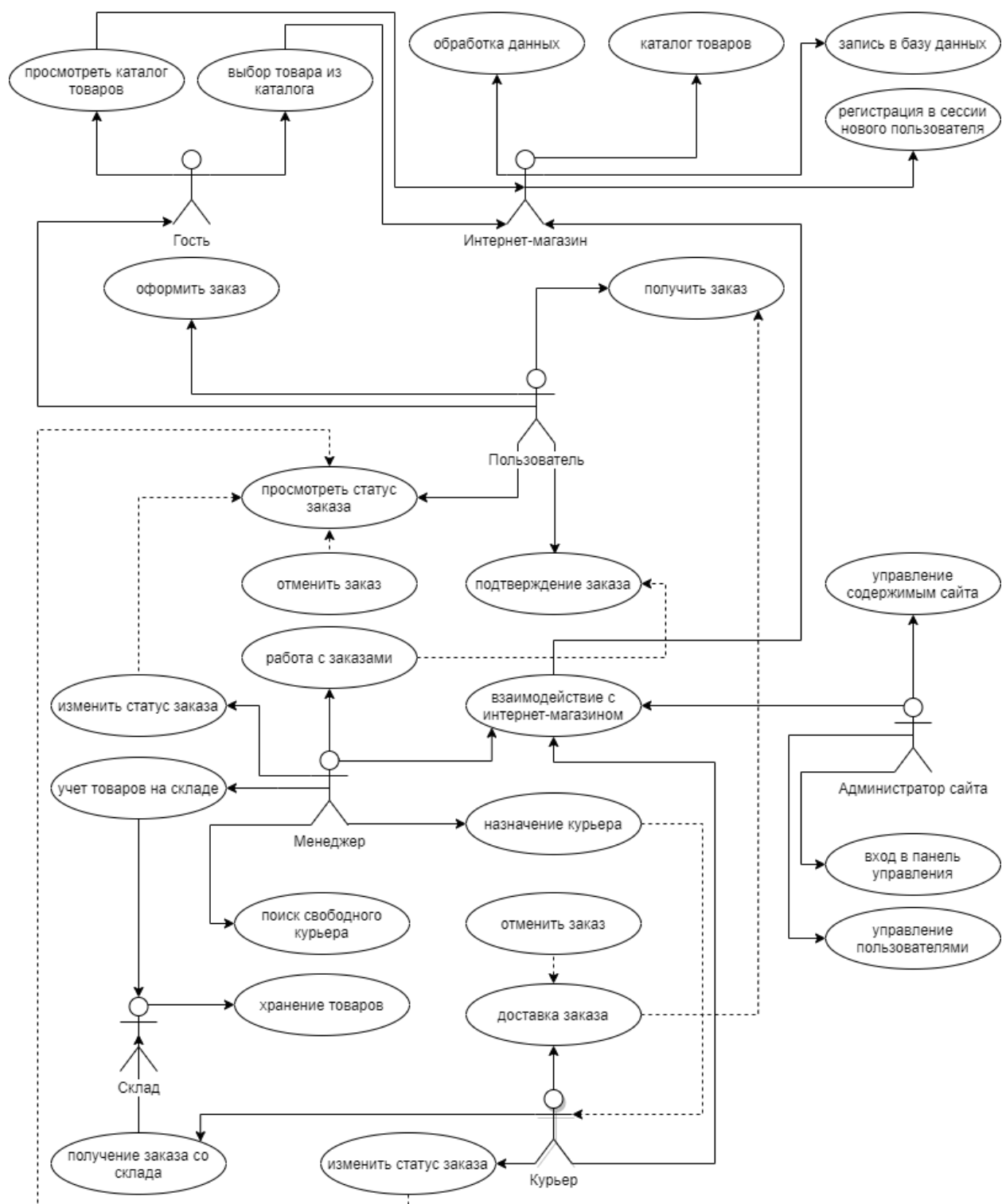


Рисунок 2.1 – UML-диаграмма прецедентов

Построенная диаграмма деятельности, представленная на рисунке 2.2, отображает последовательность действий объектов системы при работе с данным программным средством.

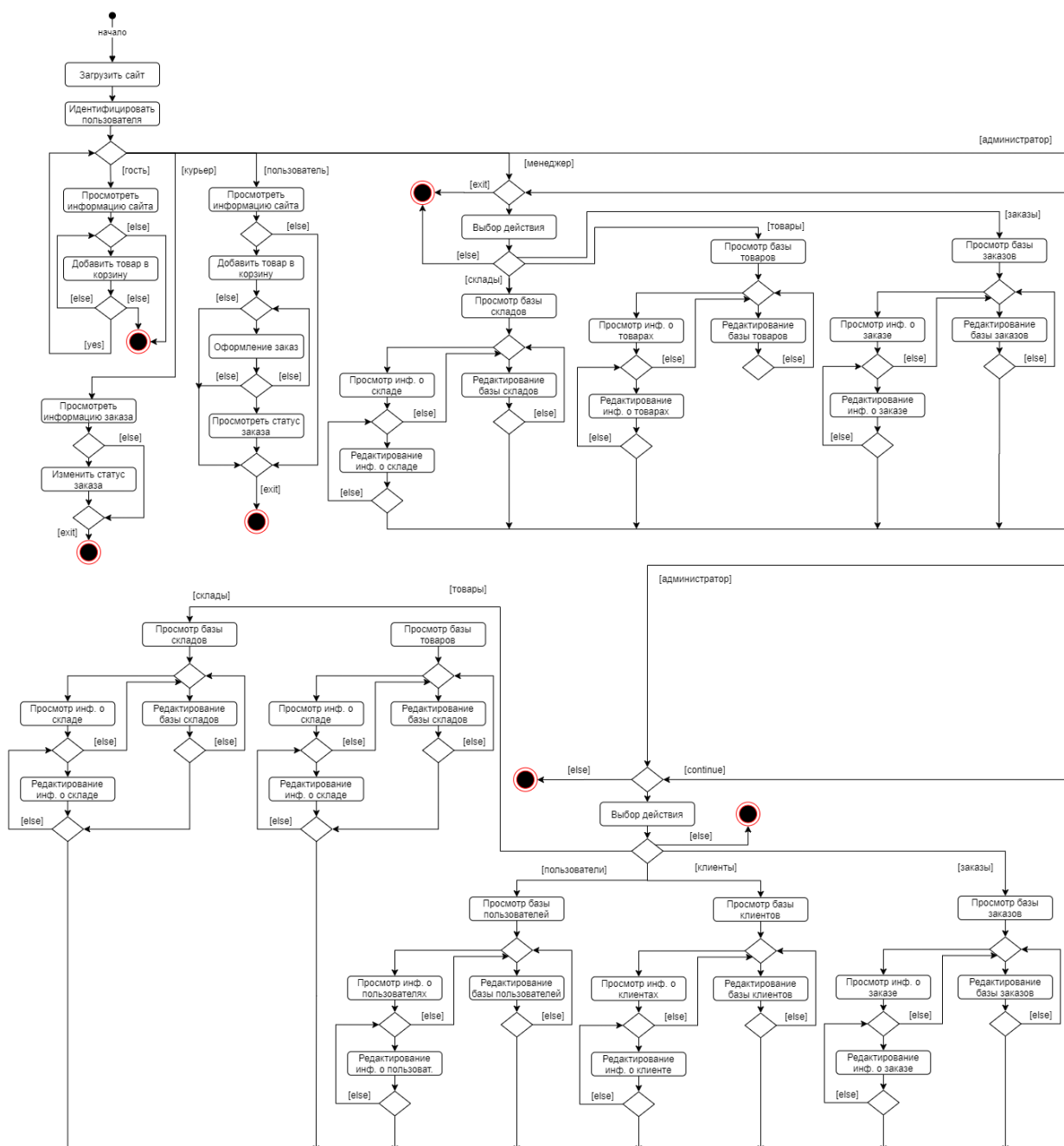


Рисунок 2.2 – UML-диаграмма деятельности

При работе с сайтом можно выделить четыре режима работы:

- работа в режиме гостя;
- работа в режиме администратора;
- работа в режиме пользователя;
- работа в режиме курьера;
- работа в режиме менеджера.

Работа в режиме гостя имеет следующий алгоритм действий:

- просмотр основной информации на сайте;
- добавление товара в корзину.

Работа в режиме пользователя предполагает действия:

- просмотр основной информации на сайте;
- добавление товара в корзину;
- оформление заказа;
- просмотр статуса заказа.

Работа в режиме менеджера предполагает действия:

- просмотр базы данных заказов;
- редактирование статуса заказа;
- просмотр информации о заказе;
- просмотр базы данных товаров;
- просмотр базы данных клиентов;
- редактирование базы данных товаров;
- назначение курьера;
- выбрать действие.

Работа в режиме администратора предполагает:

- просмотр базы данных пользователей;
- просмотр базы данных товаров;
- просмотр базы данных складов;
- просмотр базы данных заказов;
- просмотр базы данных клиентов;
- редактирование информации пользователей;
- редактирование информации о товаре;
- редактирование информации о складе;
- редактирование информации клиента;
- просмотр информации о заказах;
- выбрать действие.

Работа в режиме курьера:

- просмотр базы данных заказов;
- просмотр базы данных клиентов;
- редактирование статуса заказа.

На стадии логического моделирования можно определить сколько будет действующих объектов, как они будут взаимодействовать друг с другом и какие отношения будут иметь, а также какие функции будут выполнять, что в дальнейшем упростит разработку и правильное построение связей базы данных и алгоритма работы программного средства.

2.3 Моделирование базы данных

База данных является неотъемлемой и очень важной составляющей программного средства. Логическая модель, построена методом сущность-связь, разработанной базы данных отображена на рисунке 2.3.

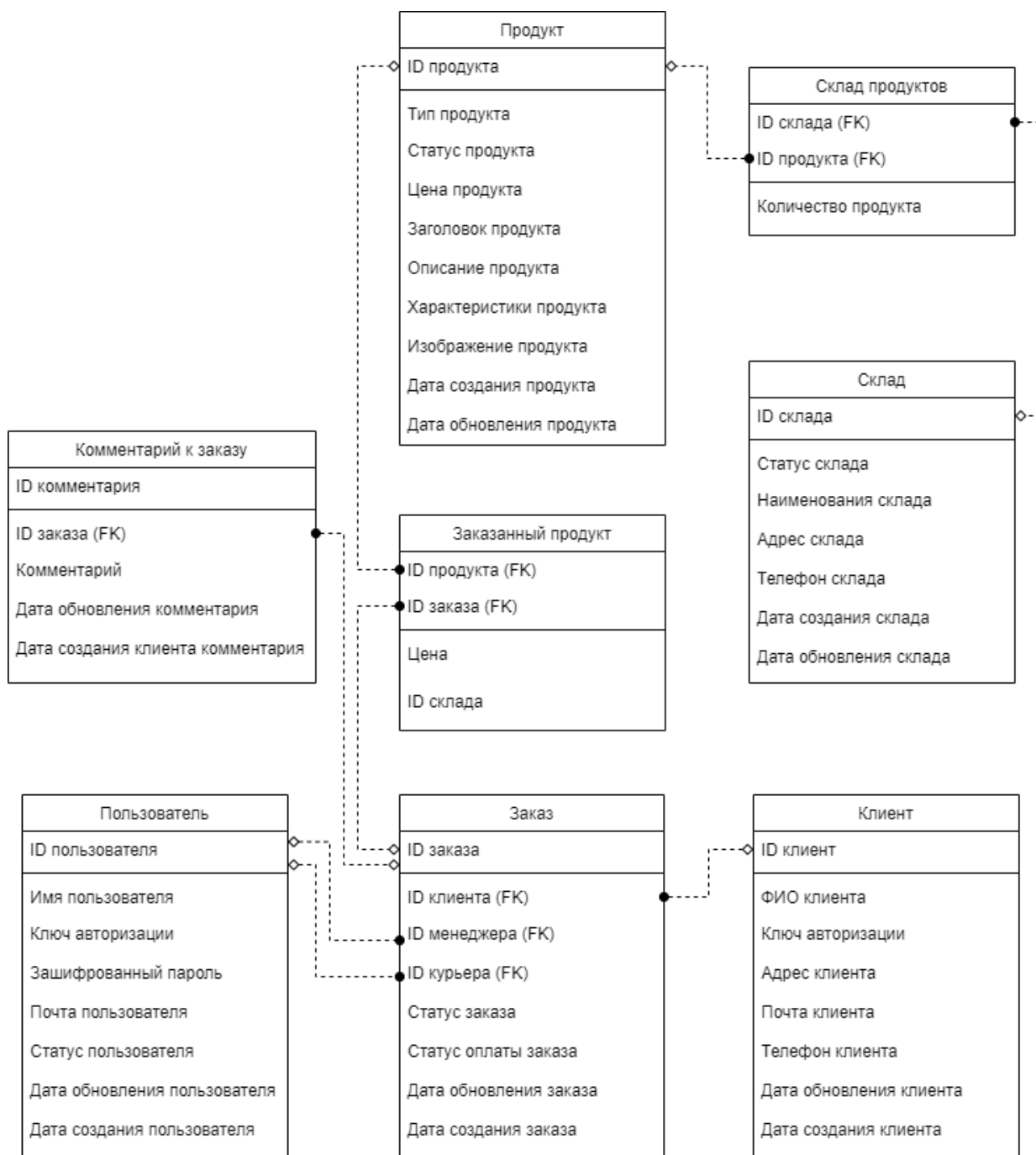


Рисунок 2.3 – Логическая модель базы данных

Логическая модель представляет собой модель базы данных, которая не привязана к конкретной системе управления базой данных [6]. В логической модели выделяют основные объекты базы данных и определяют связи между объектами. База данных содержит 8 сущностей: продукт, склад продуктов, комментарий к заказу, заказанный продукт, склад, пользователь, заказ, клиент.

Сущность «продукт» содержит 10 атрибутов: ID продукта (ключевой атрибут), тип продукта, статус продукта, цена продукта, заголовок продукта, описание продукта, характеристики продукта, изображение продукта, дата

создания и изменения продукта. Данная сущность предназначена для хранения информации о продукте: уникального номера продукта; типа продукта; статуса продукта; актуальной цены на продукт; заголовка; описания, характеристик; изображения и дат создания и обновления для продукта.

Сущность «пользователь» содержит 8 атрибутов: ID пользователя (ключевой атрибут), имя пользователя, ключ авторизации, зашифрованный пароль, почта пользователя, статус пользователя, дата обновления пользователя, дата изменения пользователя. Данная сущность предназначена для хранения информации о пользователе: уникального номера пользователя; имени пользователя и его пароля в зашифрованном виде; почты; статуса пользователя; ключа авторизации для автоматической регистрации пользователя; дат создания и обновления пользователя.

Сущность «заказ» содержит 8 атрибутов: ID заказа (ключевой атрибут), ID клиента, ID менеджера, ID курьера, статус заказа, статус оплаты заказа, дата обновления и изменения заказа. Данная сущность предназначена для хранения информации о заказе: уникальный номер заказа; уникальные номера для клиента, менеджера, курьера; статуса заказа; статуса оплаты заказа; дат обновления и обновления заказа.

Сущность «клиент» содержит 8 атрибутов: ID клиента (ключевой атрибут), ФИО (фамилия, имя и отчество) клиента, ключ авторизации, адрес клиента, почта клиента, телефон клиента, дата обновления и дата изменения клиента. Данная сущность предназначена для хранения информации о клиенте: уникального номера клиента; фамилии, имени и отчества пользователя; ключа авторизации для регистрации; адреса, телефона и почты клиента; даты создания и обновления клиента.

Сущность «склад» содержит 7 атрибутов: ID склада (ключевой атрибут), статус заказа, наименование склада, адрес склада, телефон склада, дата создания и обновления склада. Данная сущность предназначена для хранения информации о складе: уникального номера склада, статуса, адреса, телефона и дат обновления и создания склада.

Сущность «комментарий к заказу» содержит 5 атрибутов: ID комментария (ключевой атрибут), ID заказа, комментарий, дата создания комментария, дата обновления комментария. Данная сущность предназначена для хранения информации о комментарии: уникального номера комментария и заказа, комментарий (описание), дат создания и обновления комментария.

Сущность «заказанный продукт» является связующей сущностей «заказ» и «продукт», она содержит 3 атрибута: ID продукта и ID заказа, цена заказанного продукта.

Сущность «склад продуктов» является связующей для сущностей «продукт» и «склад», она содержит 3 атрибута: ID склада, ID продукта, количество продукта.

Все связи базы данных являются связями один ко многим. Для данного типа связи характерно, что несколько строк из дочерней таблицы зависят от одной строки в родительской таблице. База данных хранит реляционную модель данных.

На рисунке 2.4 представлена физическая модель базы данных, построенная на основе логической модели базы данных. Данная модель жестко связана с конкретной системой управления базами данных. В данной модели представлены реальные типы данных, ограничения, реальные названия таблиц базы данных.

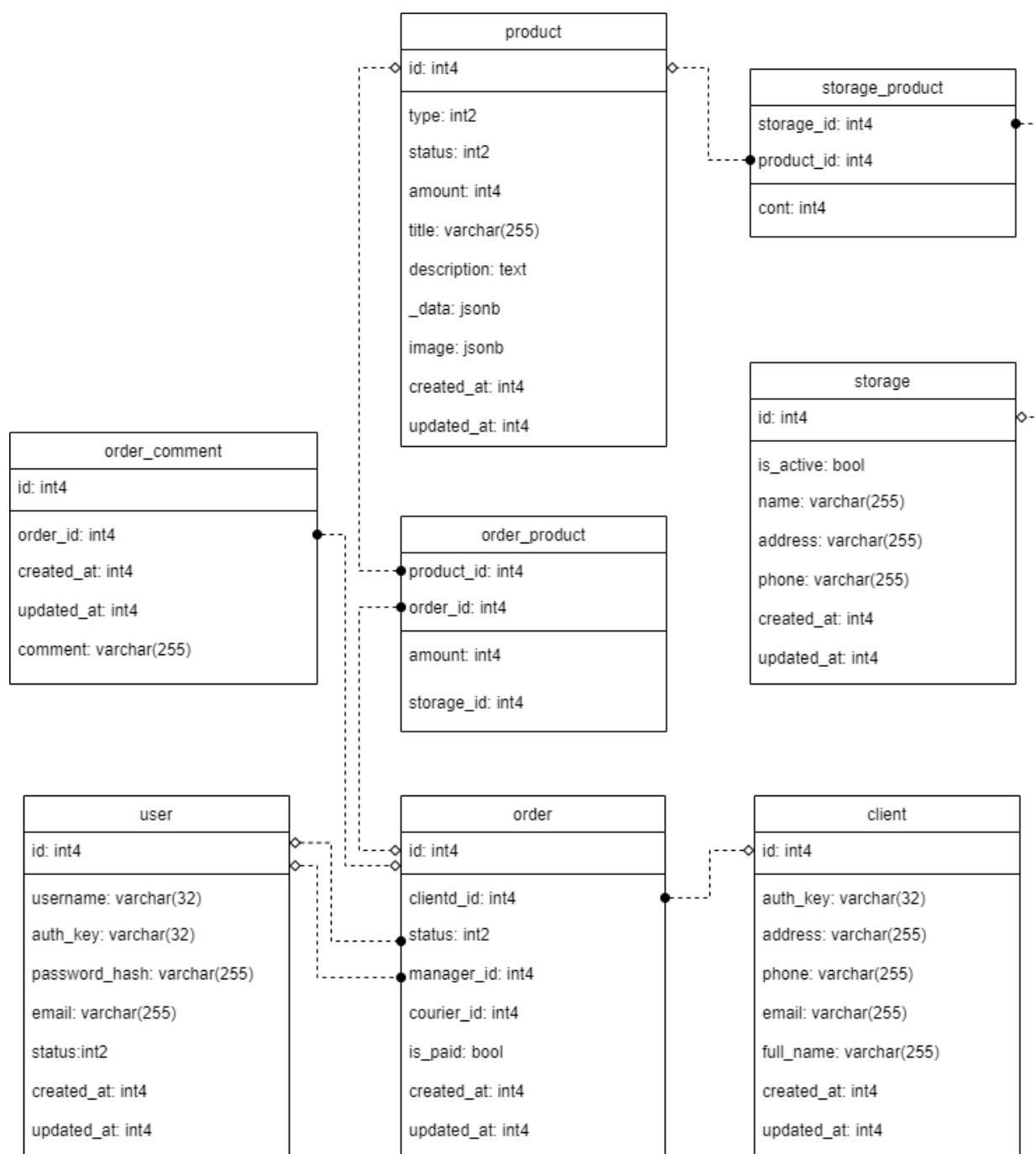


Рисунок 2.4 – Физическая модель базы данных

В таблице 2.1 отображено соответствие сущностей логической и физической моделей.

Таблица 2.1 – Соответствие сущностей логической и физической моделей

Тип продукта	type
Статус продукта	status
Цена продукта	amount
Заголовок продукта	title
Описание продукта	description
Характеристики продукта	_data
Изображение продукта	image
Дата обновления продукта	updated_at
Дата создания продукта	created_at
ID продукта	id
Цена	amount
ID комментария	id
Комментарий	comment
Дата создания комментария	created_at
Дата обновления комментария	updated_at
Количество продукта	cont
Статус склада	is_active
Наименование склада	name
Адрес склада	address
Телефон склада	phone
Дата создания склада	created_at
Дата обновления склада	updated_at
ID склада	id
ID пользователя	id
Имя пользователя	username
Ключ авторизации	auth_key
Зашифрованный пароль	password_hash
Почта пользователя	email
Статус пользователя	status
Дата создания пользователя	created_at
Дата обновления пользователя	updated_at
ID заказа	id
ID клиента	client_id
ID менеджера	id
ID курьера	courier_id
Статус заказа	status
Статус оплаты заказа	is_paid
Дата создания заказа	created_at
Дата обновления заказа	updated_at
ID клиента	id
ФИО клиента	full_name
Адрес клиента	address
Почта клиента	email
Телефон клиента	phone

Продолжение таблицы 2.1

Дата обновления клиента	created_at
Дата создания клиента	updated_at
ID продукта	product_id
ID заказа	order_id
ID назначенного склада	storage_id

Подробное описание сущностей представлено в таблицах 2.2 – 2.9.

Таблица 2.2 – Таблица user

id	int4
username	varchar (255)
auth_key	varchar (255)
password_hash	varchar (255)
email	varchar (255)
status	int2
created_at	int4
updated_at	int4

Таблица 2.3 – Таблица order

id	int4
client_id	int4
manager_id	int4
courier_id	int4
status	int2
is_paid	bool
created_at	int4
updated_at	int4

Таблица 2.4 – Таблица storage

id	int4
is_active	int4
name	varchar (255)
address	varchar (255)
phone	varchar (255)
created_at	int4
updated_at	int4

Таблица 2.5 – Таблица client

id	int4
auth_key	varchar (255)
full_name	varchar (255)
address	varchar (255)
phone	varchar (255)
email	varchar (255)
created_at	int4
updated_at	int4

Таблица 2.6 – Таблица product

id	int4
type	int2
status	varchar (255)
amount	varchar (255)
title	varchar (255)
description	varchar (255)
_data	jsonb
image	jsonb
created_at	int4
updated_at	int4

Таблица 2.7 – Таблица order_comment

id	int4
order_id	int4
comment	varchar (255)
created_at	int4
updated_at	int4

Таблица 2.8 – Таблица storage_product

storage_id	int4
product_id	int4
cont	int4

Таблица 2.9 – Таблица order_product

product_id	int4
order_id	int4
amount	int4
storage_id	int4

Данные таблицы, представленные выше (таблицы 2.2 – 2.9), описывают какие данные и какого типа включает каждая из сущностей. Данные, которые включают сущности имеют тип: int4, int2, varchar (255), bool, jsonb.

Тип данных bool – простейший тип данных, который выражает истинное значение. Он может быть true (истина), либо false (ложь).

Целочисленным типом данных являются int4 и int2, где цифра указывает размерность – количество байт, определяющих диапазон значений для данного типа данных.

Тип данных varchar (255) – определяет символьную строку с переменной длиной, где значение 255 указывает на размерность строки, где в данном случае размерность строки составляет 255 символов.

Тип данных jsonb – бинарное представление данных, являющимся разновидностью формата json – текстового формата, используемого для представления структурированного потока данных (чисел, строк и массивов объектов), используемый для передачи данных через глобальную сеть. Благодаря использованию jsonb данные сжимаются и более эффективны для

хранения и обработки. Формат jsonb позволяет хранить произвольную структуру данных и производить по ним поиск. Имена полей, данные в которых нельзя изменять напрямую именуются с нижнего подчеркивания, например, как «_data». При разработке базы данных было использовано программное обеспечение PostgreSQL – объектно-реляционная система управления базами данных.

2.4 Защита от несанкционированного доступа к данным

Для стабильной работы программного средства и сохранности важных данных необходимо позаботиться о безопасности данных [7].

Управление данными осуществляется из панели управления сайтом – административная панель. Доступ к панели управления предоставляется пользователям с привилегиями после успешной авторизации по логину и паролю. Для авторизации в панели управления необходимо находиться в корпоративной сети компании, где выход в интернет осуществляется через выделенный IP адрес. Таким образом сотрудники получают доступ к административной панели сайта только со своего рабочего места и в рабочее время. Для посетителей сайта информация о продукции компании предоставляется только для просмотра. У пользователя нет возможности прочитать произвольные данные. При заказе понравившегося товара не требуется авторизация, пользователь оставляет личные данные при только оформлении заказа. При оплате заказа данные банковской карты пользователя не сохраняются в системе, а только передаются для выполнения платежа через API (специальный интерфейс взаимодействия между программами) в процессинговый центр банковского учреждения.

SHH-ключи – в основе данной технологии заложено несколько криптографических ключей, которые используются для проверки подлинности в качестве альтернативы аутентификации с помощью логина и пароля. Система входа использует закрытый и открытый ключи, которые создают до аутентификации с помощью пароля. Закрытый ключ хранится в тайне надежным пользователем, в то время как открытый ключ может раздаваться с любого сервера SHH, к которому нужно подключиться.

Так как сайт располагается на сервере, то также необходимо предусмотреть меры защиты при его использовании: SHH-ключи, файрвол, VPN и private networking, PKI и SSL/TLS шифрование.

С помощью SHH любой вид аутентификации полностью зашифрован, однако, если разрешена аутентификация на основе пароля, злоумышленники могут добраться до данных сервера. Благодаря установке аутентификации по SHH-ключам можно забыть о паролях. Ключи имеют больше битов данных, чем пароли, что может означать большое количество комбинаций, которые сложно подобрать.

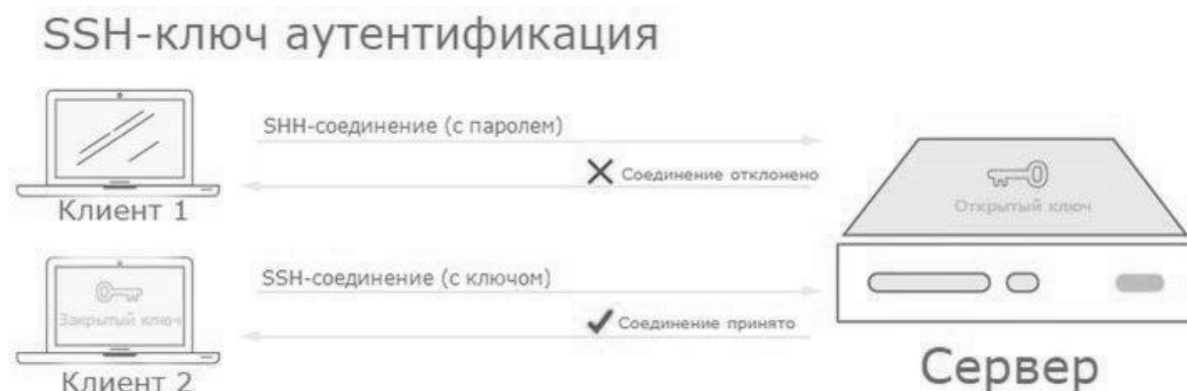


Рисунок 2.5 – Пример использования SSH-ключей

Файрвол – часть программного обеспечения или программно-аппаратного обеспечения, которая фильтрует трафик и контролирует доступ к сети – это означает ограничение доступа к каждому открытому порту, кроме исключений. Файрвол обеспечивает дополнительный уровень защиты. Уязвимые для атаки компоненты, покрытие файрволом, уменьшат поверхность атаки на сервер.

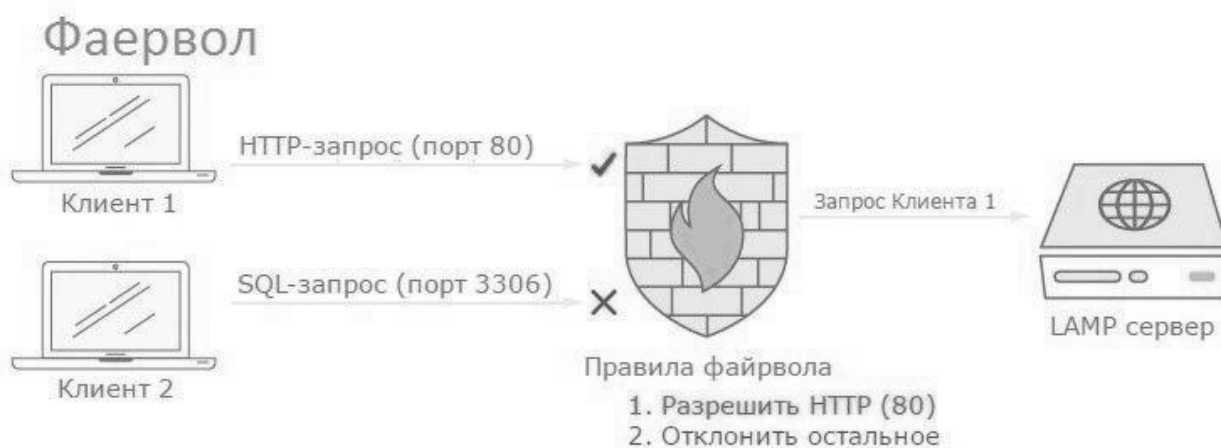


Рисунок 2.6 – Пример использования файрвола

VPN (виртуальная частная сеть) – способ создать защищенное соединение между удаленными компьютерами и текущим соединением. Использование VPN, по сути, способ создать частную сеть, которую могут видеть только частные серверы данной сети, где связь будет приватной и безопасной.

Виртуальная закрытая сеть (VPN)



Рисунок 2.7 – Пример использования VPN

Инфраструктура открытых ключей (PKI) – совокупность системы, которые предназначены для создания, управления и проверки сертификатов для идентификации лиц и шифрования передаваемых данных.

SSL/TLS шифрование



Рисунок 2.8 – Пример использования SSL/TLS шифрования

Создание центра сертификации и управления сертификатами для серверов позволяет каждому в пределах серверной инфраструктуры шифровать свой трафик и использовать проверки идентичности других пользователей. PKI поможет предотвратить атаки посредника, когда злоумышленник имитирует поведение сервера, чтобы перехватить трафик или подменить сообщение.

2.5 Моделирование алгоритма работы

Для правильной реализации программного средства необходимо смоделировать алгоритм работы. Алгоритм работы описывает логику работы программного средства [8].

Обобщенный алгоритм заказа товара представлен на чертеже ГУИР.400201.019200 ПД.4. Данный алгоритм отражает принцип работы всей системы. Изначально пользователь переходит на главную страницу сайта, после чего выбирает понравившийся ему товар на сайте и добавляет его в корзину. Выбор и добавление товара в корзину вынесено в предопределенный процесс (алгоритм добавления товара в корзину), который раскрывается на чертеже ГУИР.400201.019200 ПД.2. Данный алгоритм заключается в том, что пользователь изначально выбирает категорию товара, после чего добавляет товар в корзину. Те товары, которые выбрал пользователь, добавляются в сессию, в которой находится данный пользователь. Пользователь может добавлять товары снова-снова, поэтому в алгоритме отображена проверка продолжения покупок. Алгоритм заканчивается блоком перехода на страницу корзины.

После алгоритма выбора и добавления товара в корзину происходит проверка сессией на наличие информации о пользователе в базе данных. Происходит проверка условия: авторизован ли пользователь. В случае, если пользователь не авторизован, то происходит регистрация пользователя – создание пустой учетной записи с уникальным номером, последующая авторизация под данной учетной записью.

Если пользователь не совершит покупок на сайте и не оформит заказ, то его пустая учетная запись будет стерта. Далее осуществляется переход к оформлению заказа (предопределенный процесс), алгоритм оформления заказа представлен на чертеже ГУИР.400201.019200 ПД.3. На данном этапе пользователь заполняет все данные в форму заказа, при этом происходит проверка всех полей на заполненность. Если не все поля заполнены, то происходит вывод сообщения об ошибке. После завершения оформления заказа, данные о пользователе сохраняются в базе данных.

Далее следует предопределенный процесс – оплата, который заключается в заполнении информации банковской карточки и последующей проверки правильности введенных данных. После успешной оплаты информация об успешной оплате сохраняется в базе данных. Данные банковской карты не сохраняются в целях безопасности.

Управление данными осуществляется из панели администратора, в которую вход осуществляется с логином и паролем. Алгоритм авторизации в панели управления сайтом отображен на чертеже ГУИР.400201.019200 ПД.1. Алгоритм заключается в следующем: пользователь вводит данные форму авторизации, происходит проверка заполненности всех полей и вывод ошибки в случае, если не все поля заполнены. Если все поля заполнены, то происходит проверка наличия данных (логина и пароля) пользователя в базе данных и последующая проверка: совпадают ли введенные пароль и логин. Если данные совпадают, то id пользователя добавляется в сессию, пользователь авторизуется и попадает в панель управления.

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

3.1 Обоснование выбора инструментов разработки

Процесс создания веб-сайта или веб-приложения основывается на нескольких этапах: веб-дизайн, верстка страниц, программирование для веб на стороне клиента и сервера, а также конфигурирование веб-сервера.

Инструменты разработки, которые были использованы при проектировании данного дипломного проекта: PHP, JS, HTML, CSS, Linux Server Ubuntu версии 19.04, PostgreSQL [9].

Linux Server – серверная операционная система. Выбор данной операционной системы для разработки был основан на том, что большинство серверов используют данную операционную систему. Если бы разработка велась под другой операционной системой, например, Windows, то в будущем могли возникнуть непредвиденные ошибки из-за конфликта совместимости.

PostgreSQL – объектно-реляционная система управления базами данных, которая базируется на языке SQL. Основные преимущества данной системы:

- поддержка базы данных неограниченного размера;
- мощные и надёжные механизмы транзакций и репликаций;
- расширяемая система встроенных языков программирования и поддержка загрузки совместимых модулей;
- наследования;
- объекто-ориентированность;
- поддержка сторонними организациями;
- легкая расширяемость.

Одной из сильных сторон PostgreSQL является ее архитектура. Доступ из приложений к данным базы осуществляется посредством процесса базы данных. Клиентские программы не могут получить доступ к данным самостоятельно, даже если они работают на той же вычислительной машине, на которой выполняется серверный процесс. Такое разделение клиентов и сервера позволяет построить распределенную систему. Благодаря тому, что манипулирование данными сосредоточено на сервере, системе управления базами данных не приходится контролировать многочисленных клиентов, то есть PostgreSQL может поддерживать целостность данных даже при одновременном доступе большого количества пользователей.

В проекте были использованы фреймворки Yii2 (PHP-фреймворк), JQuery (JS-фреймворк), Bootstrap (HTML, CSS). Фреймворк – это платформа для создания сайтов и веб-приложений, облегчающая разработку и объединение разных компонентов большого программного проекта.

Yii2 – фреймворк для веб-программирования общего назначения, который может быть использован для разработки практически любых веб-

приложений. Основным преимуществом является функция кэширования и встроенная поддержка AJAX (технология обращения к серверу без перезагрузки страницы). А также другие преимущества:

- относительно простая базовая архитектура для организации кода, что обеспечивает универсальность использования и простоту работы;
- производительность, Yii2 является одним из самых быстрых структурных оболочек;
- способность взаимодействия практически со всеми типами баз данных;
- возможность подключения сторонних классов, библиотек и расширений;
- применение различных вариантов кэширования;
- генерация PHP-кода;
- поддержка, большое количество документации.

PHP (Hypertext Preprocessor) – это распространенный язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML. Задачей языка PHP является обеспечение эффективной связи веб-ресурса с сервером и базами данных. При этом данный язык необычайно прост в освоении и применении.

Код, написанный на PHP, направлен на выполнение двух задач:

- html-часть отвечает за внешний вид и отображение информации;
- php-часть, интегрированная в html, обеспечивает возможности интерактивности и динамику.

При этом подобные программные коды и, соответственно, составленные с их помощью проекты, являются легкими, эффективными, гибкими, многофункциональными, удобными в администрировании, редактировании, обслуживании.

На сегодняшний день подавляющее большинство сайтов, сервисов и приложений, а также такие популярные платформы как Joomla, Drupal, WordPress, 1С-Bitrix написаны именно на PHP-языке.

Язык PHP обладает рядом неоспоримых преимуществ:

- высокая скорость работы и, соответственно, общая производительность ресурсов;
- бюджетность, экономичность;
- простота освоения, простой синтаксис;
- отличная совместимость и переносимость – php-коды работают одинаково хорошо с разными платформами;
- набор текста кода и его редактирование можно осуществлять в любом текстовом или html-редакторе;
- высокая гибкость, емкость и функциональность. PHP-программу можно составлять отдельно от разработки веб-страницы, без

привязки, после чего совместить. Это существенно упрощает жизнь дизайнеров и программистов;

- многозадачность и широкие возможности – создание любых веб-приложений, блогов, гостевых книг, интернет-магазинов, сайтов, работа с редиректами, заголовками, pdf-документами, базами данных, электронной почтой.

Bootstrap – это инструмент с открытым исходным кодом для разработки с помощью HTML (HyperText Markup Language), CSS (Cascading Style Sheets), JS (JavaScript). Он включает в себя множество разных компонентов для веб-сайта: типографику, веб-формы, кнопки, блоки навигации и другие. Основные преимущества Bootstrap:

- адаптивность – дизайн сайта будет корректно отображаться на экранах устройств разных размеров вне зависимости от диагонали;
- кросс-браузерность – одинаковое отображение во всех браузерах;
- легкость в использовании;
- понятный код – позволяет писать качественный и понятный код, что облегчает чтение кода для других разработчиков;

HTML – язык разметки (маркировки) гипертекста. HTML дает возможность производить переход от одной части текста к другой, и, что замечательно, эти части могут храниться на совершенно разных компьютерах. Он создан специально для разметки веб-страниц. Именно язык разметки дает браузеру необходимые инструкции о том, как отображать тексты и другие элементы страницы на мониторе.

CSS — язык таблиц стилей, который позволяет прикреплять стиль (например, шрифты и цвет) к структурированным документам (например, документам HTML и приложениям XML). Обычно CSS-стили используются для создания и изменения стиля элементов веб-страниц и пользовательских интерфейсов, написанных на языках HTML и XHTML, но также могут быть применены к любому виду XML-документа, в том числе XML, SVG и XUL. Отделяя стиль представления документов от содержимого документов, CSS упрощает создание веб-страниц и обслуживание сайтов.

JavaScript – это интерпретируемый язык программирования, разработанный для взаимодействия с веб-страницами. Он используется для описания сценариев для активных веб-страниц. Программа на JavaScript встраивается непосредственно в исходный текст HTML-документа и интерпретируется браузером по мере загрузки этого документа. С помощью JavaScript можно динамически изменять текст загружаемого HTML-документа и реагировать на события, связанные с действиями посетителя или изменениями состояния документа или окна.

jQuery – javascript библиотека, написанная на языке JavaScript. Он обеспечивает гибкую выборку любого элемента на странице. Преимущества JQuery:

- простая работа с событиями;

- кроссбраузерность;
- удобная работы с AJAX (Asynchronous Javascript and XML) – асинхронные запросы к серверу;
- удобные методы для работы с эффектами;
- наличие готовых плагинов.

Использование фреймворков существенно облегчает разработку. Они позволяют сформировать типичную архитектуру, которую можно модифицировать на свое усмотрение, что обеспечивает разработку любого сайта под конкретные задачи.

3.2 Структура программного средства

Yii-приложения организованы по шаблону проектирования Model-View-Controller (MVC, «Модель-Представление-Контроллер») (см. рисунок 3.1). Модели представляют собой данные, бизнес-логику и бизнес-правила; представления отвечают за отображение информации, в том числе и на основе данных, полученных из моделей; контроллеры принимают входные данные от пользователя и преобразовывают их в понятный для моделей формат и команды, а также отвечают за отображение нужного представления.

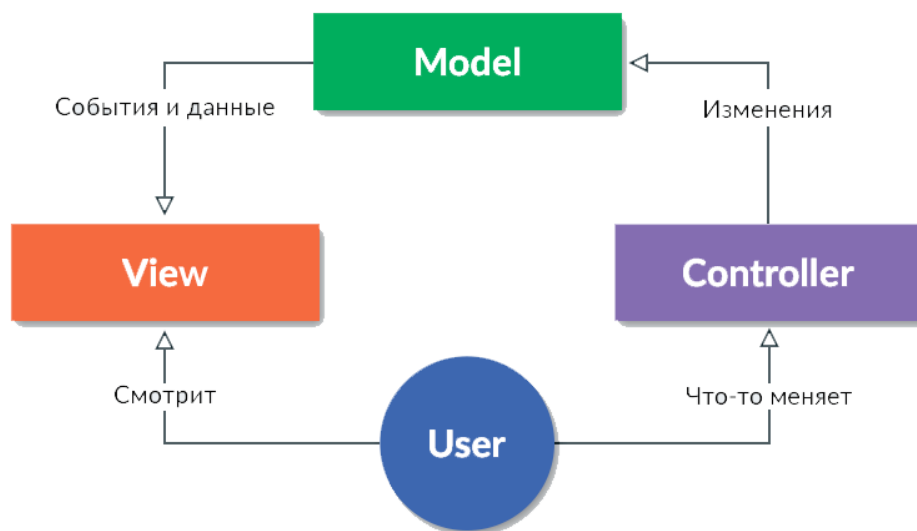


Рисунок 3.1 – Model-View-Controller

Модель не зависит от представления (не знает, как данные визуализировать) и контроллера (не имеет точек взаимодействия с пользователем), просто предоставляя доступ к данным и управлению ими. Модель строится таким образом, чтобы отвечать на запросы, изменяя свое состояние. Модель, за счет своей независимости от визуального представления, может иметь несколько различных представлений для одной «модели». Представления не обрабатывают данные введенные

пользователем, а только отвечают за получение необходимых данных из модели и отправляет их пользователю. Контроллер обеспечивает «связь» между пользователем и системой, использует модель и представление для реализации необходимого действия.

Кроме MVC, Yii приложения также имеют следующие сущности:

- входные скрипты: это PHP скрипты, которые доступны напрямую конечному пользователю приложения. Они ответственны за запуск и обработку входящего запроса;
- приложения: это глобально доступные объекты, которые осуществляют корректную работу различных компонентов приложения и их координацию для обработки запроса;
- компоненты приложения: это объекты, зарегистрированные в приложении и предоставляющие различные возможности для обработки текущего запроса;
- модули: это самодостаточные пакеты, которые включают в себя полностью все средства для MVC. Приложение может быть организовано с помощью нескольких модулей;
- фильтры: это код, который должен быть выполнен до и после обработки запроса контроллерами;
- виджеты: это объекты, которые могут включать в себя представления. Они могут содержать различную логику и быть использованы в различных представлениях.

Типичная структура Yii-приложения представлена на рисунке 3.2.

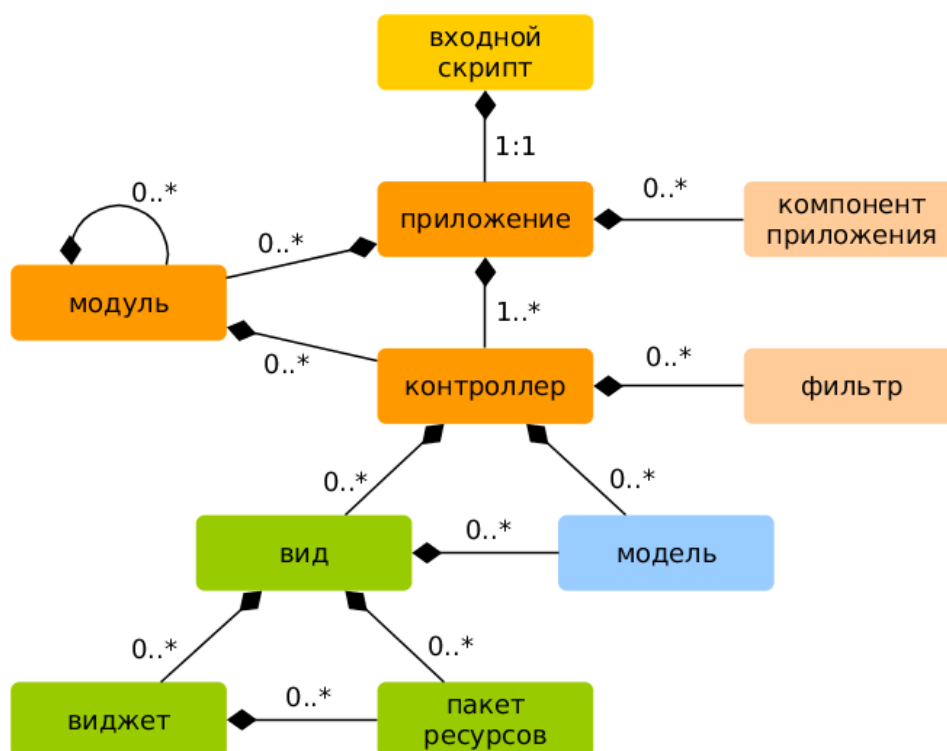


Рисунок 3.2 – Типичная структура Yii-приложения

Так как MVC не имеет строгой реализации, то реализован он может быть по-разному. Бизнес-логика может располагаться как в контроллере, так и в модели. В последнем случае, модель будет содержать все бизнес-объекты со всеми данными и функциями.

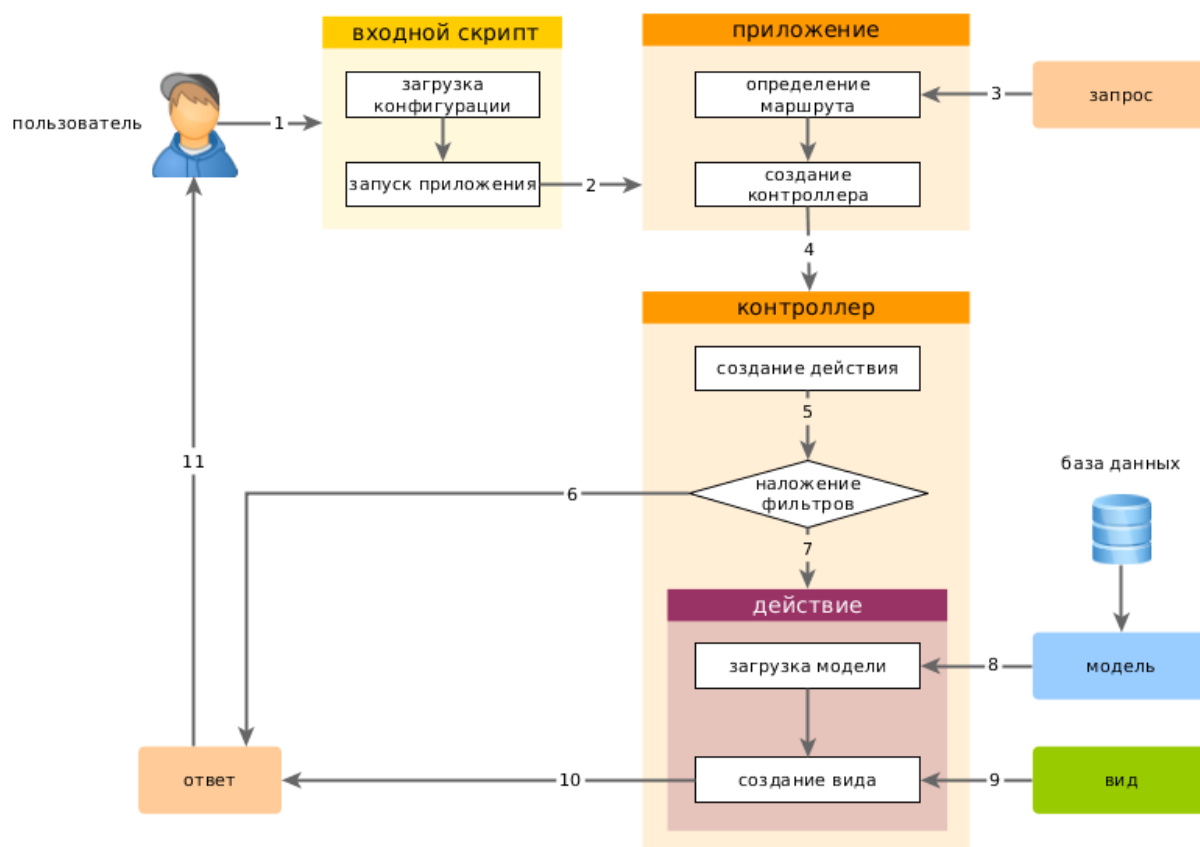


Рисунок 3.3 – Обработка запроса от пользователя приложением

Последовательность обработки запроса от пользователя приложением представлена на рисунке 3.3. Она включает в себя:

- пользователь обращается к точке входа `web/index.php`;
- скрипт загружает конфигурацию и создает экземпляр приложения для дальнейшей обработки запроса;
- приложение определяет маршрут запроса при помощи компонента приложения «запрос»;
- приложение создает экземпляр контроллера для выполнения запроса;
- контроллер, в свою очередь, создает действие и накладывает на него фильтры;
- если хотя бы один фильтр дает сбой, выполнение приложения останавливается;
- если все фильтры пройдены – приложение выполняется;
- действие загружает модель данных;

- действие генерирует вид, отображая в нем данные (в том числе полученные из модели);
- сгенерированный вид приложения передается как компонент «ответ»;
- компонент «ответ» отправляет готовый результат работы приложения браузеру пользователя.

Для разрабатываемого программного средства была создана фронтенд (англ. front-end) – клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса. Бэкенд (англ. back-end) – программно-аппаратная часть сервиса. Для бэкенд описаны классы и их функциональное назначение для программного средства. К классам относятся контроллеры и модели, компоненты. Описание классов контроллеров приведено в таблице 3.1. Описание классов моделей приведено в таблице 3.2. Описание классов компонентов приведено в таблице 3.3.

Таблица 3.1 – Описание классов контроллеров для бэкенд

Наименование контроллера	Назначение
ClientController	Работа с данными модели «Client»: просмотр, редактирование, удаление.
OrderController	Работа с данными модели «Order»: просмотр, редактирование, удаление.
OrderProdcutController	Работа с данными модели «OrderProduct»: редактирование, удаление.
StorageController	Работа с данными модели «Storage»: просмотр, редактирование, удаление, добавление.
StorageProductController	Работа с данными модели «StorageProduct»: просмотр, редактирование, удаление, добавление.
UserController	Работа с данными модели «User»: просмотр, редактирование, удаление, добавление, авторизация, деавторизация, поиск по имени.
ChairMassagerController	Работа с данными модели «ChairMassager»: просмотр, редактирование, удаление, добавление.
HandMassagerController	Работа с данными модели «HandMassager»: просмотр, редактирование, удаление, добавление.
SiteController	Отображает ошибки.

Таблица 3.2 – Описание классов моделей для бэкенд

Наименование модели	Назначение
Client	Работа с таблицей «client» в базе данных: отображение, добавление, удаление, валидация данных.

Продолжение таблицы 3.2

LoginForm	Проверка данных пользователя и авторизация.
Order	Работа с таблицей «order» в базе данных: отображение, добавление, удаление, валидация данных.
OrderDecorator	Предоставляет данные в удобном виде для отображения.
Product	Работа с таблицей «product» в базе данных: отображение, добавление, удаление, валидация данных.
Storage	Работа с таблицей «storage» в базе данных: отображение, добавление, удаление, валидация данных.
StorageProduct	Работа с таблицей «storage_product» в базе данных: отображение, добавление, удаление, валидация данных.
User	Работа с таблицей «user» в базе данных: отображение, добавление, удаление, валидация данных.
ChairMassager	Реализует работы с продуктами типа «TYPE_CHAIR_MASSAGER», которые хранятся в таблице «product».
HandMassager	Реализует работы с продуктами типа «TYPE_HAND_MASSAGER», которые хранятся в таблице «product».
Permission	Содержит константы прав доступа.

Таблица 3.3 – Описание классов компонентов для бэкенд

Наименование модели	Назначение
Nav	Расширение стандартного компонента для отображения меню с добавлением проверки прав доступа пользователей к элементам меню.

Для фронтэнд построена диаграмма классов контроллеров, отображенная на рисунке 3.4.

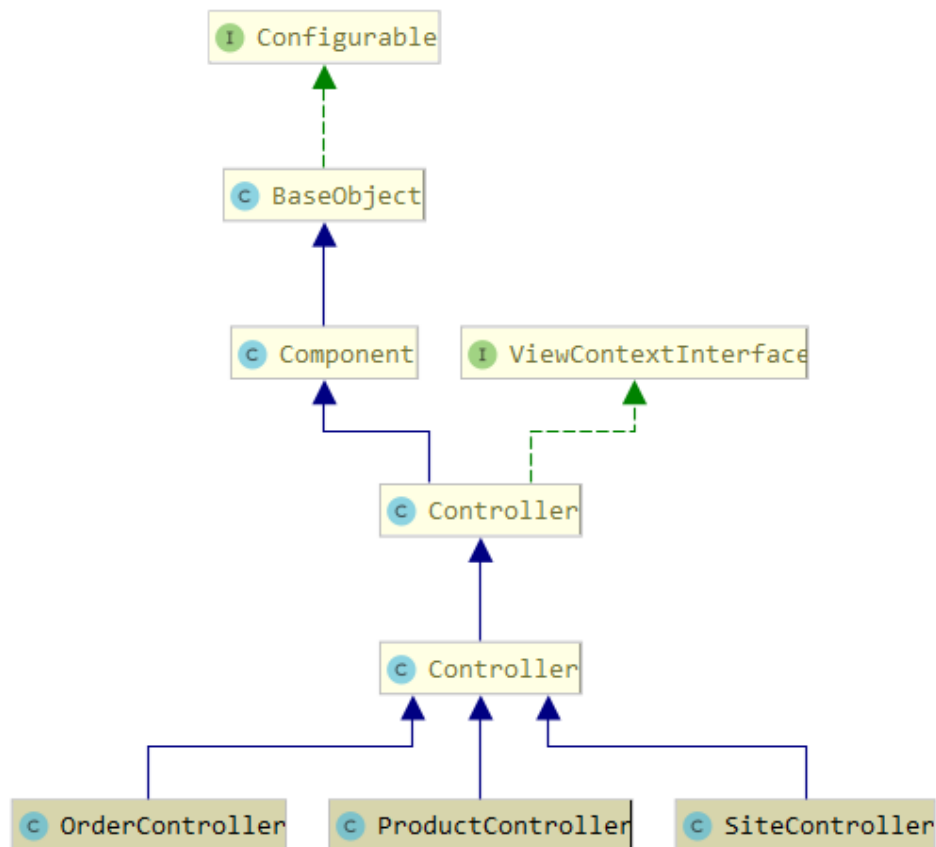


Рисунок 3.4. – Диаграмма классов контроллеров для фронтэнд

К классам, которые включает фронтэнд относятся контроллеры, модели и компоненты. К контроллерам относятся: «OrderController», «ProductController», «SiteController». К моделям относятся: «Card», «Client», «Order», «OrderModel», «Product», «ChairMassager», «HandMassager». К компонентам относится: «Order». Диаграмма классов моделей представлена на рисунке 3.5. Диаграмма классов для компонентов представлена на рисунке 3.6. Общая диаграмма классов для фронтенд с методами и переменными представлена на рисунке 3.7.

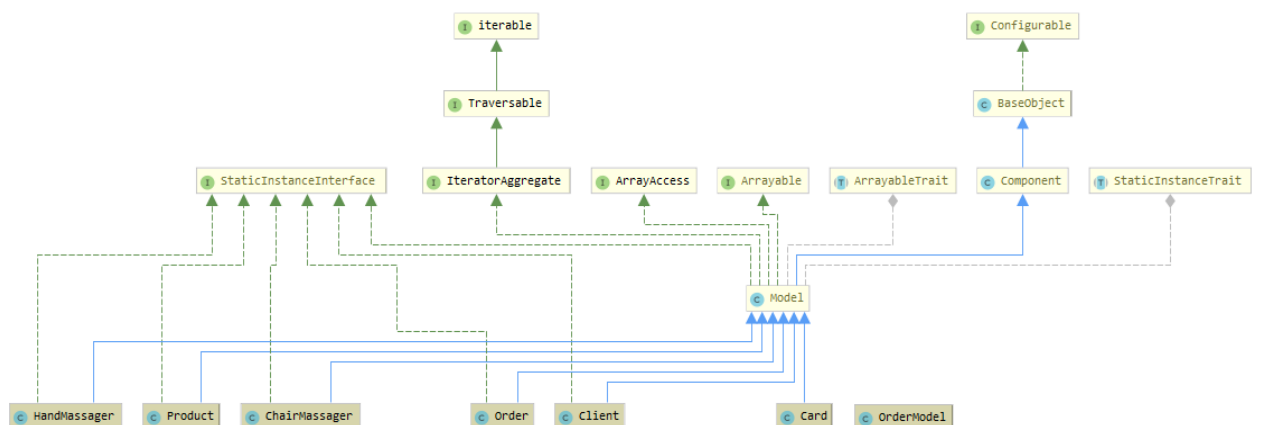


Рисунок 3.5 – Диаграмма классов моделей для фронтэнд

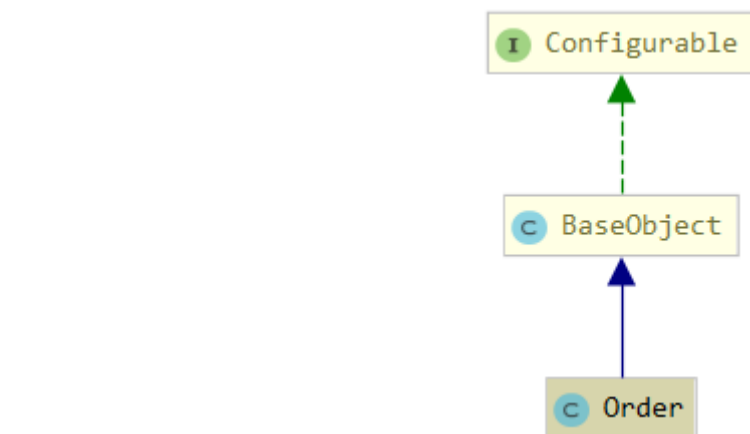


Рисунок 3.6 – Диаграмма классов компонентов для фронтэнда

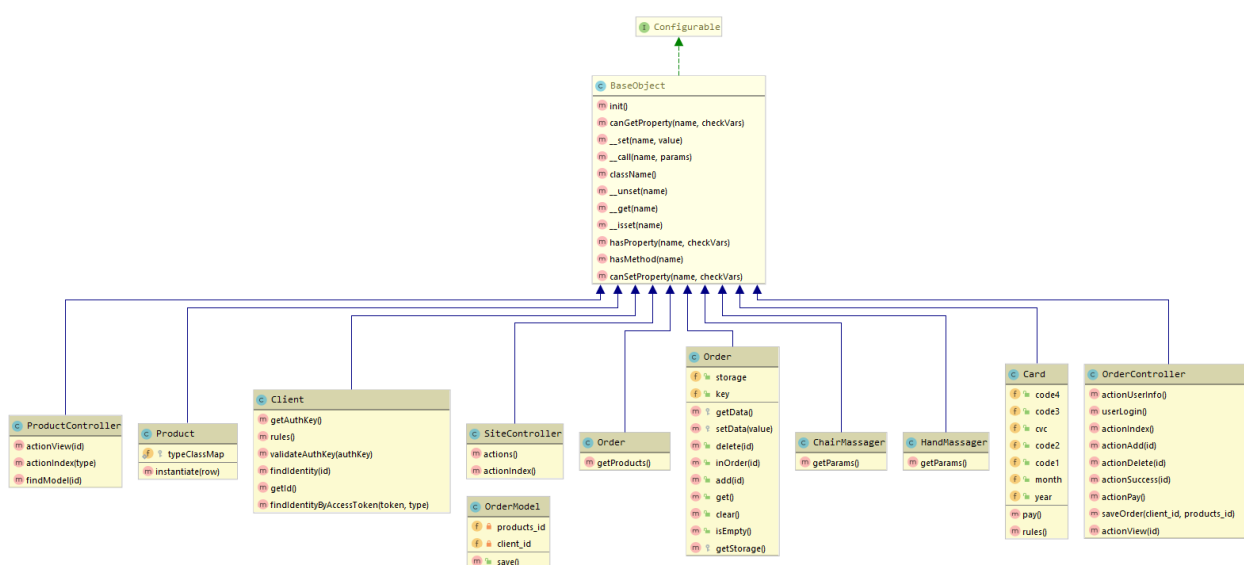


Рисунок 3.7 – Общая диаграмма классов фронтэнда

Методы, входящие в каждый класс описаны в таблицах 3.4 – 3.14.

Таблица 3.4 – Описание методов класса «OrderController»

Наименование метода	Назначение
actionIndex	Отображает список товар, которые пользователь добавил в корзину.
actionUserInfo	Отображает форму для заполнения персональной информации о пользователе.
actionPay	Отображает форму оплаты и выполняет оплату заказа. После успешной оплаты вызывает метод «saveOrder» и очищает корзину заказов.
saveOrder	Сохраняет заказ в базу данных.

Продолжение таблицы 3.4

actionAdd	Авторизация пользователя и добавление товара в корзину.
actionSuccess	Отображение уведомления об успешной оплате и выводит ссылку на просмотр статуса заказа.
userLogin	Проверяет авторизацию пользователя: если пользователь не авторизован, то создает для него новую учетную запись.
actionDelete	Удаляет выбранный товар из корзины.
actionView	Отображает сведения о заказе.

Таблица 3.5 – Описание методов класса «ProductController»

Наименование метода	Назначение
actionIndex	Отображает товары выбранного типа с постраничной навигацией и краткой информацией.
actionView	Вывод полной информации о товаре.
findModel	Получает модель «Product» и проверяет ее наличие в базе данных: если ее нет в базе, то выводит ошибку 404.

Таблица 3.6 – Описание методов класса «SiteController»

Наименование метода	Назначение
actionIndex	Отображает главную страницу сайта, на которой выводит 8 последних добавленных товаров.

Таблица 3.7 – Описание методов класса «Card»

Наименование метода	Назначение
rules	Задаёт правила валидации данных.
pay	Валидирует пользовательские данные и выполняет оплату.

Таблица 3.8 – Описание методов класса «Client»

Наименование метода	Назначение
findIdentity	Ищет модель пользователя по id.
rules	Задаёт правила валидации данных.
getId	Возвращает id клиента.
getAuthKey	Возвращает ключ авторизации пользователя.
validateAuthKey	Проверяет ключ авторизации пользователя.

Таблица 3.9 – Описание методов класса «Order»

Наименование метода	Назначение
getProducts	Переопределяет класс, на который будут осуществляться мапинг данных.

Таблица 3.10 – Описание методов класса «OrderModel»

Наименование метода	Назначение
__construct	Магический метод, который вызывается при создании нового объекта и принимает обязательные параметры для работы класса.
save	Создает новый заказ в базе данных и добавляет товары, которые выбрал пользователь с указанием их цены.

Таблица 3.11 – Описание методов класса «Product»

Наименование метода	Назначение
instantiate	Возвращает экземпляр класса с определенного типа продукта.

Таблица 3.12 – Описание методов класса «ChairMassager»

Наименование метода	Назначение
getParams	Получения значения динамических свойств класса.

Таблица 3.13 – Описание методов класса «HandMassager»

Наименование метода	Назначение
getParams	Получения значения динамических свойств класса.

Таблица 3.14 – Описание методов класса компонента «Order»

Наименование метода	Назначение
add	Добавление id товара в корзину.
delete	Удаление id товара из корзины.
clear	Очистка корзины.
get	Получение списка id товаров в корзине.
inOrder	Проверяет, есть товар в корзине с данным id.
isEmpty	Проверяет наличие товара в корзине.
getData	Возвращает данные из хранилища по ключу.
setData	Сохраняет данные в хранилище по ключу.
getStorage	Возвращает модель хранилища данных.

3.3 Описание модулей

Функциональное назначение и применение модулей для сайта велико, они обеспечивают наиболее упрощенную навигацию по сайту, увеличение конверсии, обеспечение коммуникации с клиентами и, в общем, расширение возможностей сайта [10]. Можно рассмотреть основные модули сайта: модуль вывода всех товаров, модуль галереи для товара, модуль корзины, модуль оформления заказа, модуль оплаты заказа, модуль специальных предложений – новинок, модуль последних поступлений товаров, модуль системы администрирования, модуль отслеживания заказа, модуль подгона

изображений под необходимый размер, модуль разделения каталога, модуль разделения прав доступа.

Модуль вывода всех товаров предназначен для вывода всех активных товаров интернет-магазина, где каждый товар представлен, как правило, с изображением, заголовком и ценой. Отсюда можно перейти на страницу отдельного товара с ценой, галерей изображений, подробным описанием, всеми характеристиками, кнопкой добавления в корзину, и рассмотреть его подробнее, кликнув по заголовку, который является ссылкой на отдельную страницу выбранного товара. Данный модуль удобен для первичного ознакомления с имеющимся товарами.

Модуль разделения каталога позволяет отображать определенное количество товара на странице. Количество товара, выводимое на странице составляет 24 позиции с постраничной навигацией. Такое количество выводимого товара было выбрано с учетом адаптивности сайта к дисплеям разных устройств.

Модуль галереи для товаров интегрирован на каждую страницу товара с подробным описанием. Он позволяет выводить и просматривать изображения товара с прокруткой влево или вправо. Он включает в себя изображения товара, навигационные точки и стрелки управления прокруткой.

Модуль корзины обеспечивает отображение списка добавленных товаров с последующим переходом на стадии оформления и оплаты, а также имеет навигационную полосу, где отображены этапы покупки. Данный модуль содержит в себе: уменьшенную миниатюру изображения товара, название товара, цену. Данный модуль предоставляет возможность удаления товара из списка корзины и добавления очередного товара, а также сохранения товара даже при выходе из корзины до момента его удаления пользователем.

Модуль оформления заказа обеспечивает вывод формы для заполнения данных пользователем. Данный модуль состоит из полей: адрес, телефон, email, ФИО. Данный модуль также имеет проверку на наличие данных в полях и выводит предупреждения в случае, если пользователь не заполнил нужное поле. Поля формы автоматически заполняются в случае, если пользователь ранее совершал покупку на сайте с конкретного устройства.

Модуль оплаты заказа обеспечивает проверку правильности введенных данных банковской карты. В перспективах данный модуль можно связать с платежной системой банковского учреждения и передавать данные пользователя через зашифрованное соединение для обеспечения безопасности.

Модуль специального предложения выводит новинку на главной странице с кнопкой «читать описание», которая ведет на страницу конкретного товара с подробным описанием. Его главная задача – привлечь покупателя приобрести новинку.

Модуль последних поступлений представляет блок, где отображены последние 8 добавленных товаров на сайт. Он содержит изображения товаров, заголовки и цену, где заголовок является ссылкой на товар с подробным описанием.

Модуль системы администрирования позволяет управлять всеми товарами, заказами, складами, клиентами, пользователями из панели администратора. Данный модуль представлен в виде таблиц с возможностью добавления, редактирования, удаления позиций из списка. Для товара предусмотрен предварительный просмотр в виде таблицы со всеми характеристиками и изображениями.

Модуль отслеживания заказа позволяет отслеживать заказ пользователем в реальном времени. Он содержит в себе актуальную информацию о заказе: информацию о заказанном товаре (цена, изображение, наименование), статус заказа, дату добавления и обновления, назначенные курьер и менеджер для заказа, наименование заказчика.

Модуль подгона изображений позволяет подгонять загружаемые изображения под превью товара. Он осуществляет обрезку изображения в соответствии с заданным разрешением.

Модуль разделения прав доступа обеспечивает разграничение прав доступа к управлению данными в панели администрирования.

4 ТЕСТИРОВАНИЕ

Тестирование веб-приложения является одним из важных этапов, который помогает выявить все ошибки в разработке и передать заказчику готовый исправный проект. Тестирование – это отклонение фактического результата от ожидаемого, то есть выявление и поиск ошибок. Для выявления ошибок разработанного веб-приложения можно воспользоваться тест-кейсами, которые позволят протестировать приложение [11].

Тест-кейсы для разработанного веб-приложения приведены в таблицах.

Тест-кейс №1 – запуск сайта на операционной системе Windows 10, браузер Yandex, Internet Explorer представлен в таблице 4.1.

Таблица 4.1 – Тест-кейс №1

Проверка корректности сайта при запуске		
Действия	Ожидаемый результат	Результат теста
Запустить виртуальную машину Virtual Box с установленной Linux Server. Открыть браузер. Перейти по ссылке в браузере http://diplom.my:8000/ .	Главная страница открывается. Синтаксических ошибок не имеется. CSS-стили подключены. База данных подключена. Все блоки на своих установленных местах. Адаптивная верстка работает исправно. Ссылки работают и переход осуществляется на указанный адрес.	Запуск сайта корректный, ошибок нет.

Тест-кейс №2 – проверка корректности карточек-товара, ОС Windows 10, браузер Yandex и Internet Explorer представлен в таблице 4.2.

Таблица 4.2 – Тест-кейс №2

Проверка корректности карточек-товара		
Действия	Ожидаемый результат	Результат теста
Запустить виртуальную машину Virtual Box с установленной Linux Server. Открыть браузер. Перейти по ссылке в браузере http://diplom.my:8000/ . Перейти по ссылке-заголовку товара.	Страница товара открыта. Информация о товаре корректно отображена (текст, фотогалерея). Все блоки страницы на месте. Все ссылки на странице товара работают. Адаптивность при смене разрешения работает исправно.	Отображение карточек-товара корректно.

Тест-кейс №3 – автоматическая регистрация пользователя под уникальным номером при добавлении товара в корзину, ОС Windows 10, браузер Yandex и Internet Explorer представлен в таблице 4.3.

Таблица 4.3 – Тест-кейс №3

Проверка автоматического регистрации пользователя		
Действия	Ожидаемый результат	Результат теста
Запустить виртуальную машину Virtual Box с установленной Linux Server. Открыть браузер. Перейти по ссылке в браузере http://diplom.my:8000/ . Перейти по ссылке-заголовку товара. Добавить товар корзину, нажав на кнопку «добавить в корзину».	При нажатии на кнопку «добавить в корзину» кнопка меняет цвет с зеленого на оранжевый. Значок корзины меняет цвет с белого на желтый. Гость становится пользователем сайта и ему присваивается уникальный номер – ID.	Пользователь успешно регистрируется в системе, ему присваивается уникальный номер.

Тест-кейс №4 – проверка корректности добавления товара в корзину, ОС Windows 10, браузер Yandex и Internet Explorer представлен в таблице 4.4.

Таблица 4.4 – Тест-кейс №4

Проверка корректности добавления товара в корзину		
Действия	Ожидаемый результат	Результат теста
Запустить виртуальную машину Virtual Box с установленной Linux Server. Открыть браузер. Перейти по ссылке в браузере http://diplom.my:8000/ . Перейти на страницу товара по ссылке-заголовку. Добавить товар в корзину, нажав на кнопку «добавить в корзину». Перейти в корзину по желтой иконке корзины в меню сайта.	При переходе в корзину добавленный товар отображается корректно в списке с ценой, миниатюрой, названием.	Товар добавляется в корзину и корректно отображается.

Тест-кейс №5 – проверка корректности удаления товара из корзины, ОС Windows 10, браузер Yandex и Internet Explorer представлен в таблице 4.5.

Таблица 4.5 – Тест-кейс №5

Проверка корректности удаления товара из корзины		
Действия	Ожидаемый результат	Результат теста
Запустить виртуальную машину Virtual Box с установленной Linux Server. Открыть браузер. Перейти по ссылке в браузере http://diplom.my:8000/ . Перейти на страницу товара по ссылке-заголовку.	Выбранный товар удаляется из корзины.	Удаление товара корректно.

Продолжение таблицы 4.5

Добавить товар в корзину, нажав на кнопку «добавить в корзину». Перейти в корзину по желтой иконке корзины в меню сайта. Нажать на иконку корзины для удаления товара.		
--	--	--

Тест-кейс №6 – проверка корректности отображения панели управления для администратора, ОС Windows 10, браузер Yandex и Internet Explorer представлен в таблице 4.6.

Таблица 4.6 – Тест-кейс №6

Проверка корректности отображения панели управления для администратора		
Действия	Ожидаемый результат	Результат теста
Запустить виртуальную машину Virtual Box с установленной Linux Server. Открыть браузер. Перейти по ссылке в браузере http://admin.diplom.my:8000/ . Ввести логин и пароль: admin. Нажать кнопку «login».	Вход в панель администратора успешен. Корректное отображение таблиц управления товарами, заказами, складами, клиентами, пользователями по ссылкам из панели управления.	Панель администратора корректно отображается.

Тест-кейс №7 – проверка корректности работы форм оформления заказа и оплаты, ОС Windows 10, браузер Yandex и Internet Explorer представлен в таблице 4.7.

Таблица 4.7 – Тест-кейс №7

Проверка корректности работы форм оформления заказа и оплаты		
Действия	Ожидаемый результат	Результат теста
Запустить виртуальную машину Virtual Box с установленной Linux Server. Открыть браузер. Перейти по ссылке в браузере http://diplom.my:8000/ . Перейти на страницу товара по ссылке-заголовку. Добавить товар в корзину, нажав на кнопку «добавить в корзину». Перейти в корзину по желтой иконке корзины в меню сайта. Нажать «далее». Заполнить поля и нажать «далее».	Форма отображается корректно. Работает проверка полей на правильность ввода и заполнения. Данные из формы оформления заказа сохраняются для пользователя в базе данных.	Формы работают исправно.

Тест-кейс №8 – проверка корректности добавления товаров из панели администратора, ОС Windows 10, браузер Yandex и Internet Explorer представлен в таблице 4.8.

Таблица 4.8 – Тест-кейс №8

Проверка корректности добавление товара из панели администратора		
Действия	Ожидаемый результат	Результат теста
Запустить виртуальную машину Virtual Box с установленной Linux Server. Открыть браузер. Перейти по ссылке в браузере http://admin.diplom.my:8000/ . Ввести логин и пароль: admin. Нажать кнопку «login». Перейти по ссылке в панели «product». Нажать на иконку «плюс». Ввести данные для нового товара, загрузить фото. Нажать на кнопку «save».	Переход на страницу добавления нового товара. Вывод всех полей для заполнения. Сохранение нового товара в базе данных со всеми данными.	Добавление товара корректно.

Тест-кейс №9 – проверка корректности удаления товара из панели администратора, ОС Windows 10, браузер Yandex и Internet Explorer представлен в таблице 4.9.

Таблица 4.9 – Тест-кейс №9

Проверка корректности удаления товара из панели администратора		
Действия	Ожидаемый результат	Результат теста
Запустить виртуальную машину Virtual Box с установленной Linux Server. Открыть браузер. Перейти по ссылке в браузере http://admin.diplom.my:8000/ . Ввести логин и пароль: admin. Нажать кнопку «login». Перейти по ссылке в панели «product». Нажать на ссылку в виде корзины, подтвердить удаление.	Товар удален из базы данных.	Удаление товара работает корректно.

Тест-кейс №10 – проверка корректности редактирования товара из панели администратора, ОС Windows 10, браузер Yandex и Internet Explorer представлен в таблице 4.10.

Таблица 4.10 – Тест-кейс №10

Проверка корректности редактирования товара из панели администратора		
Действия	Ожидаемый результат	Результат теста
Запустить виртуальную машину Virtual Box с установленной Linux Server. Открыть браузер. Перейти по ссылке в браузере http://admin.diplom.my:8000/ . Ввести логин и пароль: admin. Нажать кнопку «login». Перейти по ссылке в панели «product». Нажать на ссылку в виде карандаша. Внести нужные изменения. Нажать на кнопку «save».	Данные о товаре обновляются в базе данных на новые, внесенные пользователем.	Редактирование товара корректно.

Тест-кейс №11 – проверка корректности быстрого просмотра информации о товаре из панели администратора, ОС Windows 10, браузер Yandex и Internet Explorer представлен в таблице 4.11.

Таблица 4.11 – Тест-кейс №11

Проверка корректности быстрого просмотра информации о товаре		
Действия	Ожидаемый результат	Результат теста
Запустить виртуальную машину Virtual Box с установленной Linux Server. Открыть браузер. Перейти по ссылке в браузере http://admin.diplom.my:8000/ . Ввести логин и пароль: admin. Нажать кнопку «login». Перейти по ссылке в панели «product». Нажать на ссылку в виде глаза.	Отрывается страница быстрого просмотра товара. Выведена корректная информация о товаре.	Быстрый просмотр товара корректно отображен.

Как видно из результатов тестирования, приложение работает исправно.

5 ПРИМЕНЕНИЕ

5.1 Назначение и область применения программного средства

Спроектированное в данном дипломном проекте веб-приложение, а именно интернет-магазин, используется для продажи продукции компании через интернет, где областью применения выступает интернет-торговля.

Данное веб-приложение было спроектировано с учетом функционального назначения, где основная задача данного программного средства – продажа продукции и представление информации об имеющихся продуктах компании. Интернет-магазин является не только визитной карточкой компании, но и дает возможность пользователям заказывать продукцию в режиме онлайн.

Создание данного интернет-магазина позволит продвигать и продавать товары и услуги компании, уменьшить издержки на рекламу и приобрести дополнительный рынок сбыта своего товара, расширить клиентскую аудиторию и, как следствие выше перечисленного, увеличить прибыль. Реклама в интернете стоит дешевле, чем в других средствах информации. Основная концепция интернет-магазина заключается в расширении клиентской аудитории, увеличении объема продаж и рекламе магазина по средствам интернета.

Основные задачи, которые решает интернет-магазин:

- продажа продукции;
- предоставление информации о продукции;
- снижение расходов на хранение товаров;
- возможность работы под заказ;
- хранение данных о товарах, заказах и другой информации в базе;
- легкое управление данными.

Благодаря созданию универсального алгоритма работы веб-приложения можно использовать разработанный проект в качестве продажи любой продукции, предварительно изменив дизайн. А также благодаря использованию и внедрению новых модулей можно расширить функционал.

Таким образом, применение данного проекта велико и затрагивает почти все сферы, связанные со сбытом продукции.

6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Запускаем проект по ссылке: <http://admin.diplom.my:8000/> и переходим на главную страницу.

Руководство для пользователя приведено на примере покупки массажного кресла. Пользователь вводит в адресную строку доменное имя сайта и переходит на сайт (см. рисунок 6.1). Далее пользователь выбирает категорию товара (в нашем случае «массажные кресла») на примере рисунка 6.2. Переходит на страницу понравившегося товара и добавляет в корзину (см. рисунки 6.2 – 6.4). После проверки содержания корзины, пользователь нажимает на кнопку «далее». Переход на форму оформления заказа. Вводит свои личные данные и переходит на страницу оплаты, нажав на кнопку «далее». Оформление заказа товара приведено на рисунках 6.5. Заключительный этап покупки – отслеживание заказа (см. рисунок 6.8), осуществляется переходом по ссылке после оплаты заказа (см. рисунок 6.6).

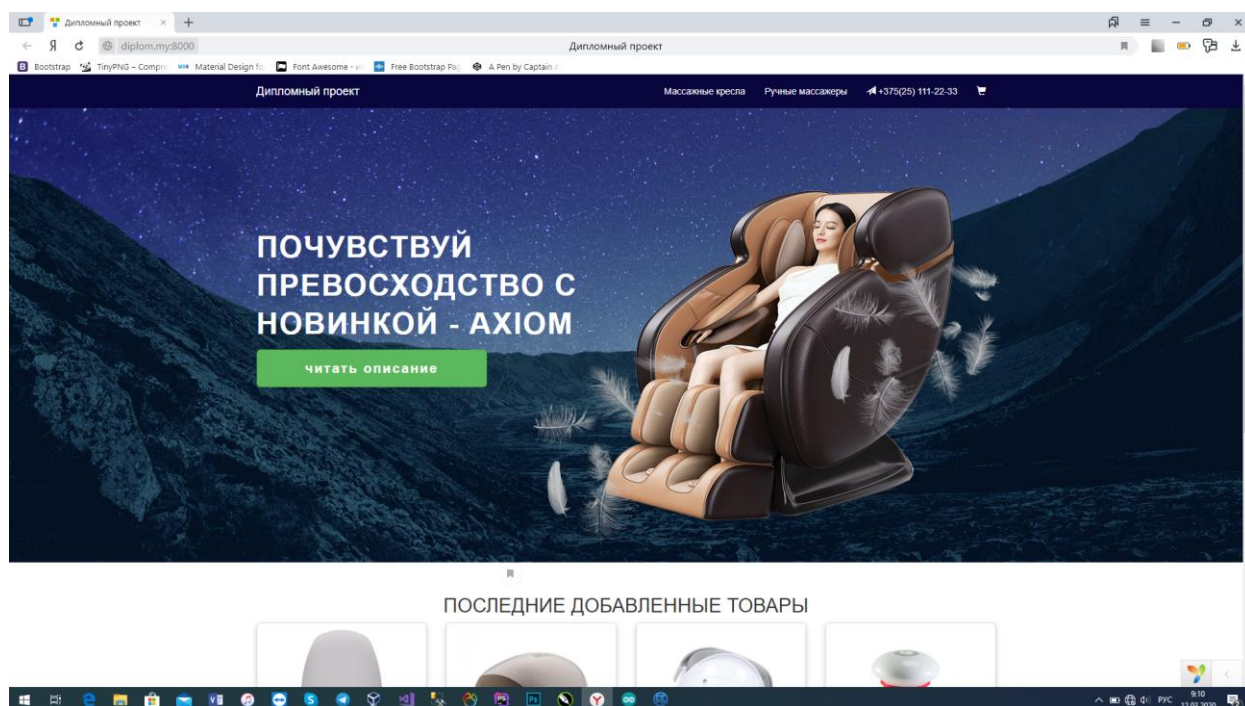


Рисунок 6.1 – Переход на главную страницу сайта

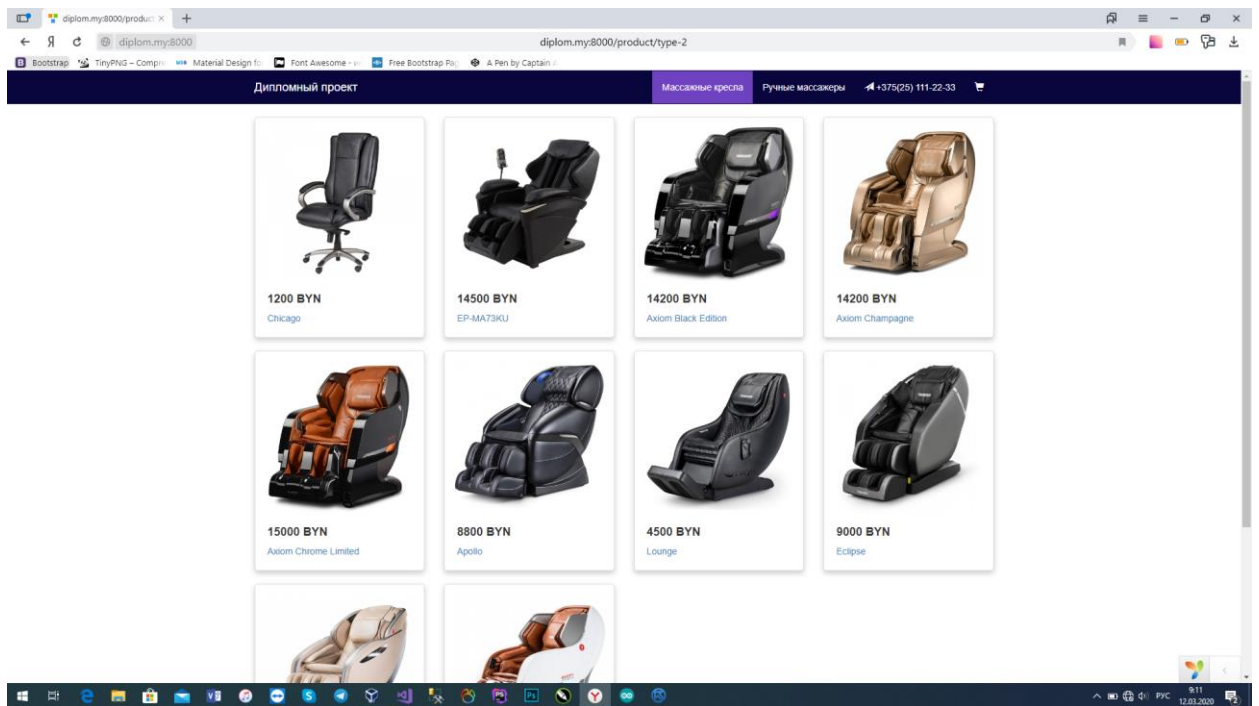


Рисунок 6.2 – Переход на категорию товара «массажные кресла»

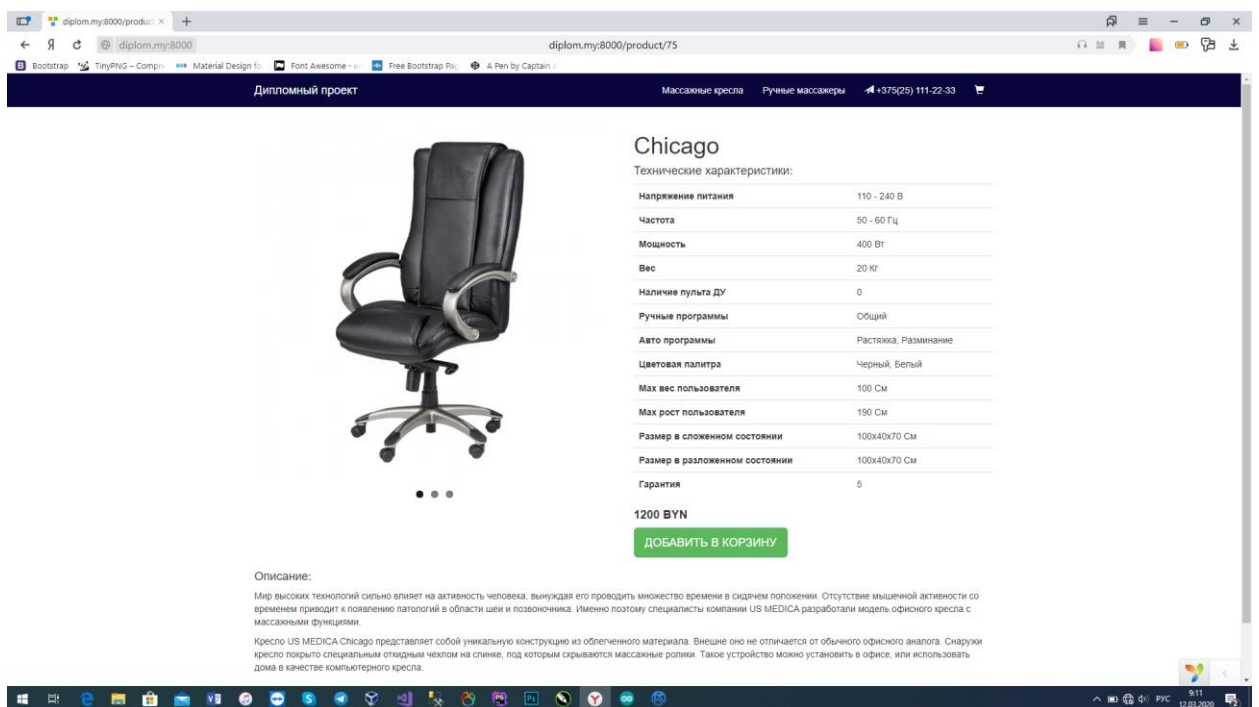


Рисунок 6.3 – Переход на страницу товара

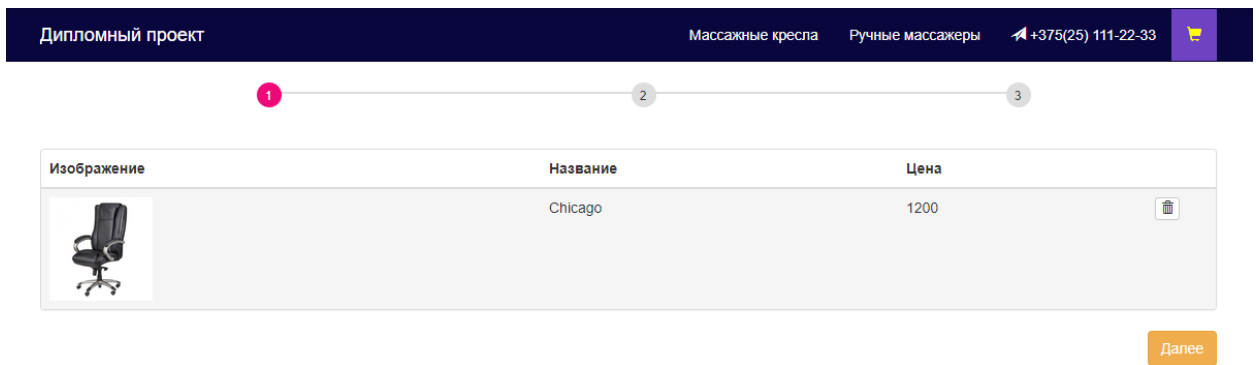


Рисунок 6.4 – Переход на страницу корзины

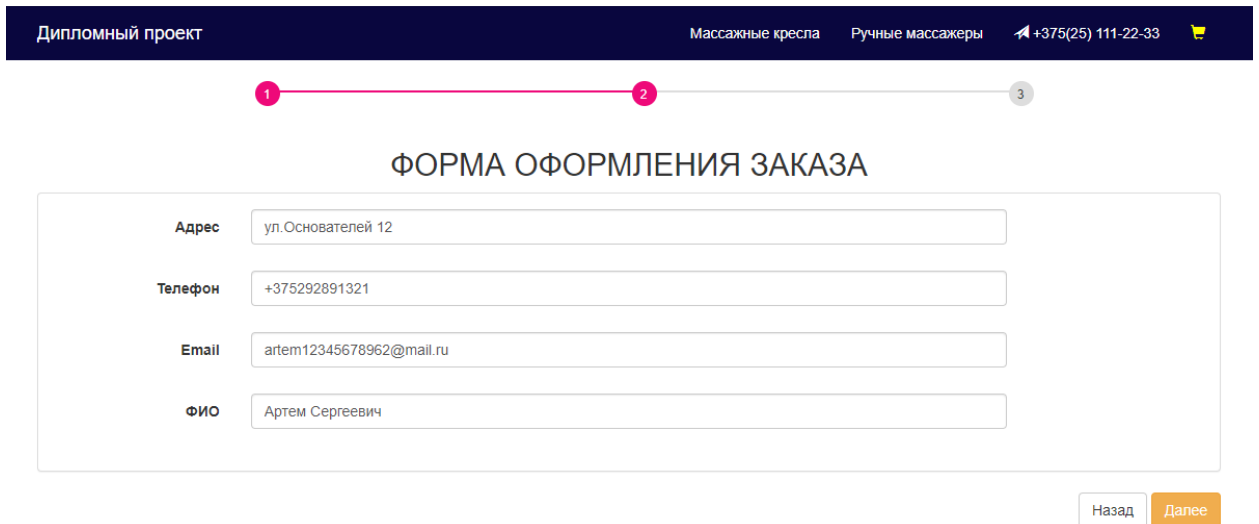


Рисунок 6.5 – Переход на страницу оформления заказа

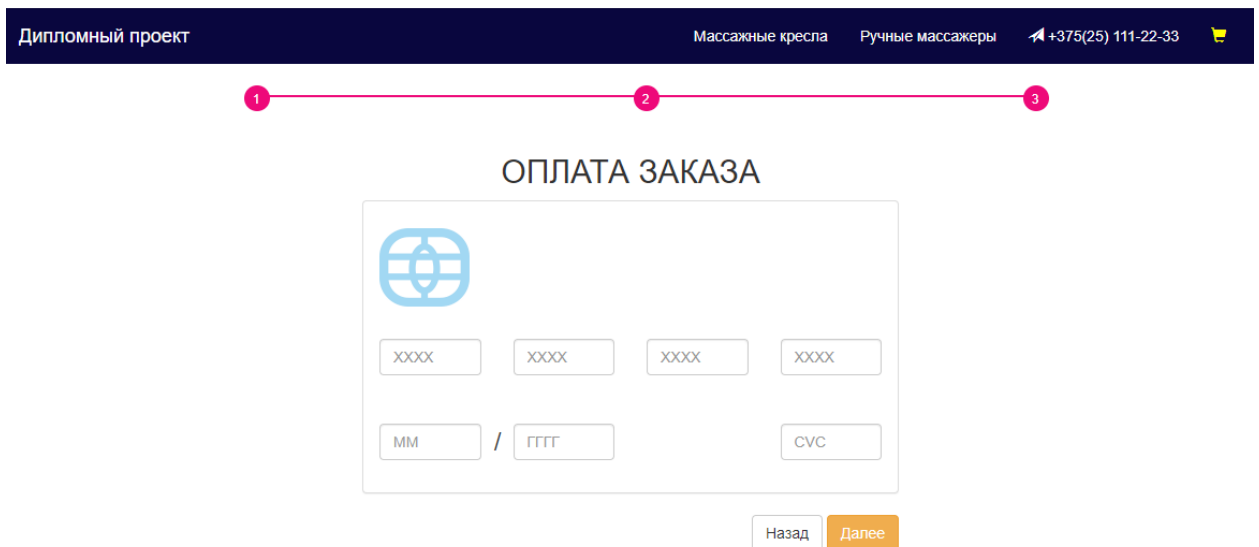


Рисунок 6.6 – Переход на страницу оплаты

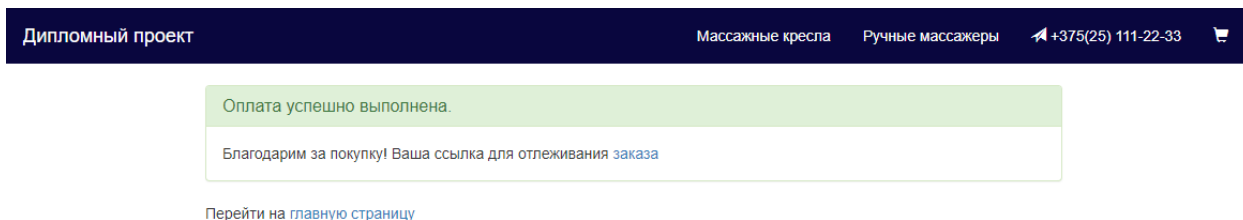


Рисунок 6.7 – Результат успешного заказа

Результата успешного заказа приведен на рисунке 6.7.

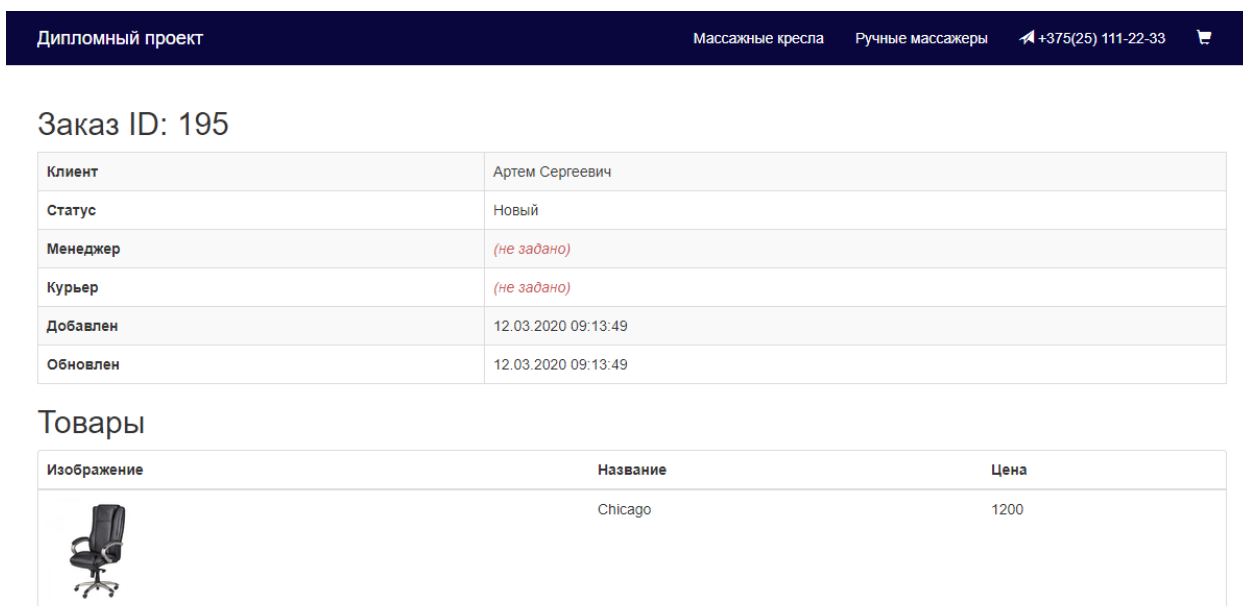


Рисунок 6.8 – Отслеживание заказа для пользователя

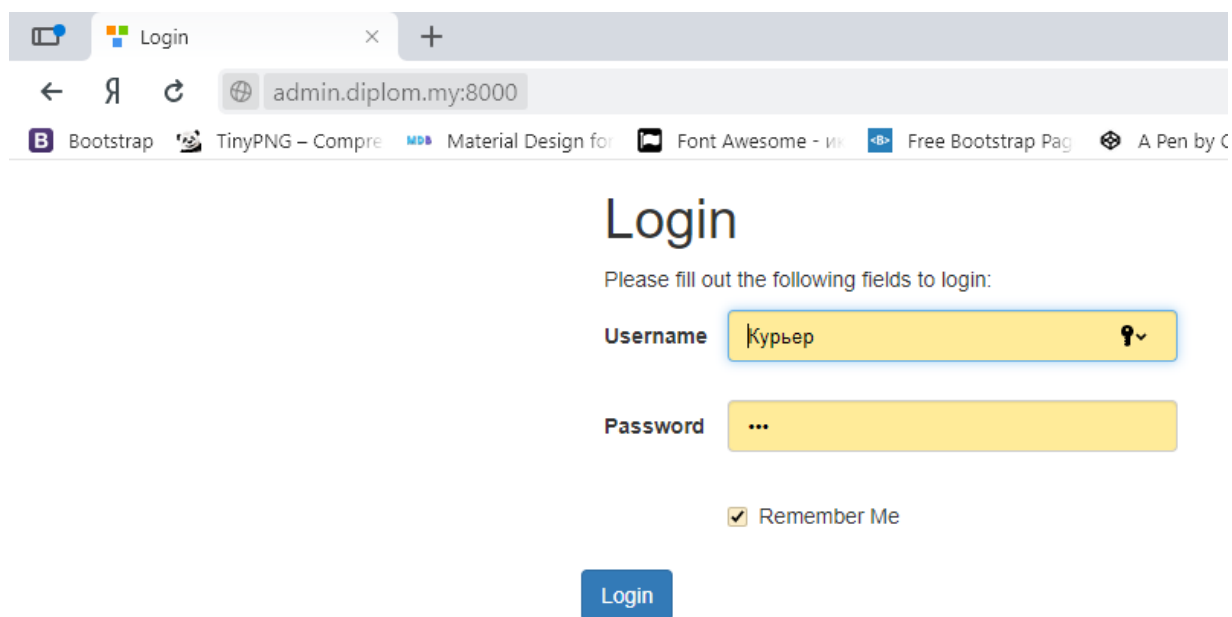


Рисунок 6.9 – Окно для ввода персональных данных для входа в панель администрирования

Руководство пользователя для администратора, менеджера, курьера (персонала компании). Для того, чтобы зайти в административную панель необходимо перейти по адресу: <http://admin.diplom.my:8000/>. Для входа в панель администратора используются данные учетных записей (администратора, менеджера, курьера). Окно для ввода персональных данных для входа в панель администрирования представлено на рисунке 6.9. После входа в панель управления под администратором открывается главное окно с ссылками в меню: «Order» – заказы, «Product» – товары, «Storage» – склады, «Client» – клиенты, «User» – пользователи (см. рисунок 6.10).

Дипломный проект - Лапицкий Артем

Order

Product

Storage

Client

User

Logout (Администратор)

Главная / Orders

Orders







ID	Статус	Client	Manager	Courier	Добавлен	
197	Новый	Виноградов Максим Олегович	(не задано)	(не задано)	12.03.2020 10:02:32	  
196	В обработке	Виноградов Максим Олегович	Менеджер	Курьер	12.03.2020 10:00:49	  

Рисунок 6.10 – Панель администрирования

Все действия обозначены иконками: глаз – быстрый просмотр, карандаш – редактирование, корзина – удаление. Например, перейдем на страницу товаров, нажав на ссылку «Product» (см. рисунок 6.11). Как видно, тут появляется зеленая кнопка для добавления. При нажатии на данную кнопку можно добавить новый товар (см. рисунок 6.12).

Дипломный проект - Лапицкий Артем

OrderProduct ▾StorageClientUserLogout (Администратор)

Главная / Product / Hand massager

Hand massager

ID	Название	Цена	Статус	Updated At	
78	Beurer MG510	90	Available	23.12.2019 15:58:16	<div></div>
77	Beurer MG40	100	Available	23.12.2019 15:53:42	<div></div>
79	Beurer MG16	40	Available	23.12.2019 16:01:36	<div></div>
80	Beurer CM50	90	Available	23.12.2019 16:06:20	<div></div>
81	Bork D603	340	Available	10.02.2020 16:09:04	<div></div>
82	Bork D604	580	Available	10.02.2020 16:15:03	<div></div>

Рисунок 6.11 – Страница товаров

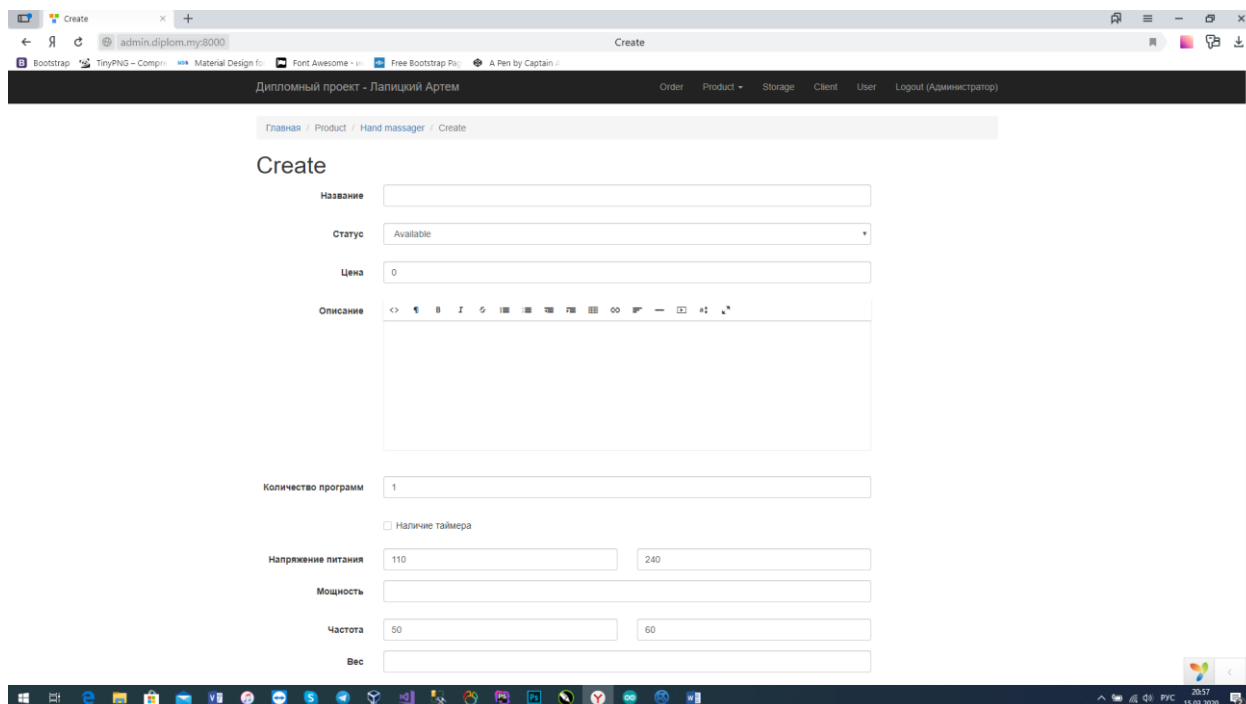


Рисунок 6.12 – Страница создания нового товара

При переходе по ссылке «Storage» получаем результат, представленный на рисунке 6.13.

Дипломный проект - Лапицкий Артем						Order	Product	Storage	Client	User	Logout (Администратор)
Главная / Storages											
Storages +											
ID	Active	Name	Address	Phone	Created At						
1	Да	Склад на Ждановичи	ул. Тимирязева, 125, корп. 2	+375172596280	17.11.2019 14:47:20						
3	Да	Склад массажных кресел на Сурганова	ул.Сурганова, 28А	+375292891321	12.03.2020 09:52:39						
2	Да	Склад на Уручье	ул. Уручская, 3А	+79586383499	01.03.2020 18:52:40						

Рисунок 6.13 – Страница складов

Для добавления нового склада необходимо нажать на зеленую кнопку, в результате откроется страница добавления нового склада (см. рисунок 6.14).

Дипломный проект - Лапицкий Артем
Order
Product
Storage
Client
User
Logout (Администратор)

Главная / Storages / Create Storage

Create Storage

☐ Active

Name

Address

Phone

Save

Рисунок 6.14 – Страница создания нового склада

Для перехода на страницу клиентов переходим по ссылке в меню «Client» (см. рисунок 6.15).

Дипломный проект - Лапицкий Артем
Order
Product
Storage
Client
User
Logout (Администратор)

Главная / Clients

Clients

ID	Адрес	Телефон	Email	ФИО	
35	ул.Первомайская, 2А	+375257896767	max123@mail.ru	Виноградов Максим Олегович	

Рисунок 6.15 – Страница клиентов

Для перехода и управления пользователями переходим по ссылке «User» (см. рисунок 6.16).

Дипломный проект - Лапицкий Артем
Order
Product
Storage
Client
User
Logout (Администратор)

Главная / Users

Users

+

ID	User name	Email	Role	Status	Created	
5	Менеджер	manager@diplom.my	manager	Active	03.11.2019 21:54:30	
6	Курьер	courier@diplom.my	courier	Active	08.03.2020 20:04:51	
1	Администратор	admin@diplom.my	admin	Active	28.10.2019 22:13:47	

Рисунок 6.16 – Страница пользователей

Интерфейс программного средства весьма прост в освоении и не требует дополнительного обучения сотрудников.

7 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА

7.1 Функциональное назначение для пользователя

Целью данного проекта является разработка веб-сайта по управлению и продвижению продукции компании.

Задача проекта – это разработать веб-сайт с удобной и мобильной структурой, которое обеспечит ряд возможностей как для пользователей веб-сайта, так и для его владельца.

Данный продукт представляет собой веб-сайт по продаже продукции компании и управлению ею, на котором размещено контактная информация, а также, карточки товара, с содержимым, которые описывают продукцию, представленную на веб-сайте. Описание продукции (массажной продукции в нашем случае) содержит полную и конкретную информацию (описание товара, цену, характеристики и другое), а также дополнительно, содержит фотографии для наглядного представления товара.

Данный продукт позволит рекламировать и продавать товары в сети интернет. Такая возможность очень подходит для начинающих предпринимателей и безусловно, для компании, которые занимаются продажей товаров и услуг. С помощью данного продукта возможна реклама и продажа на расстоянии (в сети интернет), так же доступ идет в свободном виде и круглосуточно, поэтому пользователь всегда может перейти на веб-сайт, чтобы выбрать и купить данный продукт, почти не выходя из дома.

7.2 Расчет затрат на разработку программного средства

Для расчёта плановой сметы затрат на разработку ПС необходимо определить объём ПС.

Общий объём программного средства определяется исходя из количества и объёма функций, реализуемых программой:

$$V_0 = \sum_{i=0}^n V_i , \quad (7.1)$$

где V_i – объём отдельной функции ПС, LOC;

n – общее число функций.

Тогда по формуле (7.1) получаем:

$$V_0 = 100 + 3220 + 325 + 3220 + 320 + 710 + 410 + 1008 + 2422 = 12060 \text{ LOC.}$$

Уточнённый объём ПС составит:

$$V_y = 150 + 450 + 300 + 2670 + 2780 + 720 + 7700 + 410 + 680 = 15860 \text{ LOC.}$$

На стадии технико-экономического обоснования проекта невозможно рассчитать точный объём функций, но могут быть получены ориентировочные оценки на основе имеющихся фактических данных по аналогичным проектам или путём применения нормативов. При этом оценки должны быть уточнённые. Объём функций приведён в таблице 7.1.

Таблица 7.1 – Функции программного продукта

№ функции	Наименование функции	Среда разработки	Объём функции, LOC	
			по каталогу	уточнённый
1	2	3	4	5
101	Организация ввода информации	PhpStorm	150	150
102	Контроль, предварительная обработка и ввод информации	PhpStorm	450	450
109	Организация ввода/вывода информации в интерактивном режиме	PhpStorm	320	300
204	Обработка наборов и записей баз данных	PhpStorm	2670	2670
210	Загрузка базы данных	PhpStorm	2780	2780
501	Монитор ПС (управление работой компонентов)	PhpStorm	740	720
502	Монитор системы (управление работой комплекса ПС)	PhpStorm	7740	7700
506	Обработка ошибочных и сбойных ситуаций	PhpStorm	410	410
604	Справка и обучение	PhpStorm	720	680

По уточнённому объёму ПС и нормативам затрат труда в расчёте на единицу объёма определяются нормативная и общая трудоёмкость разработки ПС.

Проект относится к категории крупных проектов, значит, трудоёмкость определяется по стадиям разработки:

- техническое задание;
- эскизный проект;
- технический проект;
- рабочий проект;
- внедрение.

Некоторые стадии могут быть объединены.

На основании принятого к расчёту уточнённого объёма V_y (15860 LOC) и категории сложности (2 категория) была определена нормативная

трудоемкость всего проекта ПО $T_n = 406$ чел.-дней, которая уточняется с учетом сложности и новизны проекта и степени использования стандартных модулей при разработке.

Для каждой стадии нормативная трудоемкость представлена в таблице 7.2.

Таблица 7.2 – Нормативные трудоемкости по стадиям

Название стадии проекта	Нормативная трудоемкость, чел.-дней
Техническое задание	36,54
Эскизный проект	28,42
Технический проект	28,42
Рабочий проект	247,66
Внедрение	64,96

Нормативная трудоемкость служит основой для определения общей трудоемкости.

Общая трудоемкость рассчитывается по формуле:

$$T_0 = \sum_{i=1}^n T_i, \quad (7.2)$$

где T_i – трудоемкость разработки ПС на i -й стадии, чел.-дней;

n – количество стадий разработки.

$$T_0 = 36,54 + 28,42 + 28,42 + 247,66 + 64,96 = 406 \text{ чел.-дней}$$

Трудоемкость стадий определяется на основе нормативной трудоемкости с учетом сложности, новизны, степени использования в разработке стандартных модулей ПС и удельного веса трудоемкости каждой стадии в общей трудоемкости ПС:

$$T_{yi} = T_n \cdot D_{cti} \cdot K_c \cdot K_t \cdot K_n, \quad (7.3)$$

где T_{yi} – уточнённая трудоемкость разработки ПС на i -й стадии, $T_{yi}=406$ чел.-дней;

D_{cti} – удельный вес трудоемкости i -й стадии разработки ПС в общей трудоемкости разработки ПС, $D_{cti}=1$;

K_c – коэффициент, учитывающий сложность ПС, $K_c=1,12$;

K_t – поправочный коэффициент, учитывающий степень использования при разработке стандартных модулей (учитывается только на стадии рабочего проекта), $K_t=0,7$;

K_n – коэффициент, учитывающий степень новизны ПС, $K_n=0,7$.

Тогда по формуле (7.3) получаем:

$$T_{y1} = 406 \cdot 1 \cdot 1,12 \cdot 0,7 \cdot 0,7 = 222,81 \text{ чел.-дней}$$

Общая трудоёмкость проекта определяется по формуле:

$$T_y = T_{yz} + T_{yз} + T_{yt} + T_{yp} + T_{yv} . \quad (7.4)$$

Таким образом, общая трудоёмкость проекта (с учётом дополнительных функций):

$$T_y = 34,65 + 22,28 + 22,28 + 141,92 + 56,93 = 278,06 \text{ чел.-дней}$$

Дополнительные работы при разработке проекта (отличные от написания кода) влияют на трудоёмкость разработки и, как следствие, на расчёт численности разработчиков (сроков разработки). Для правильного расчёта трудоёмкости и зависящих от неё параметров учтём эти работы.

Дополнительная трудоёмкость приведена в таблице 7.3.

Таблица 7.3 – Дополнительная трудоёмкость

Название работы	Дополнительная трудоёмкость, чел.-дней
Разработка базы данных	1
Дизайн интерфейса	5

- трудоёмкость стадии «Внедрение»:

$$T_{yv} = 406 \cdot 0,16 \cdot 1,12 \cdot 0,7 + 6 = 56,93 \text{ чел.-дней};$$

- трудоёмкость стадии «Рабочий проект»:

$$T_{yp} = 406 \cdot 0,61 \cdot 1,12 \cdot 0,7 \cdot 0,7 + 6 = 141,92 \text{ чел.-дней};$$

- трудоёмкость стадии «Технический проект»:

$$T_{yt} = 406 \cdot 0,07 \cdot 1,12 \cdot 0,7 + 6 = 22,28 \text{ чел.-дней};$$

- трудоёмкость стадии «Эскизный проект»:

$$T_{yз} = 406 \cdot 0,07 \cdot 1,12 \cdot 0,7 + 6 = 22,28 \text{ чел.-дней};$$

- трудоёмкость стадии «Техническое задание»:

$$T_{yz} = 406 \cdot 0,09 \cdot 1,12 \cdot 0,7 + 6 = 34,65 \text{ чел.-дней}.$$

Определим составляющие общей трудоёмкости для всего проекта в целом и по стадиям.

Сравнение характеристик разрабатываемого ПС с имеющимися аналогами позволяет определить экспертным путём степень его новизны.

Категория новизны проекта В, поскольку это ПС, являющееся развитием определённого параметрического ряда ПС, разработанных для ранее освоенных типов конфигурации ПК и ОС. Таким образом, коэффициент новизны проекта, $K_n = 0,7$.

Удельные веса трудоёмкости стадий определяются экспертным путём с учетом категории новизны ПС. Степень новизны ПС соответствует категории В. Таким образом, удельные веса трудоёмкости стадий в общей трудоёмкости представлены в таблице 7.4.

Таблица 7.4 – Удельные веса трудоёмкостей

Стадия разработки	Значение удельного веса, D_{cti}
Техническое задание	0,09
Эскизный проект	0,07
Технический проект	0,07
Рабочий проект	0,61
Внедрение	0,16

Все ПС принято подразделять на три категории сложности. Разрабатываемое ПС отнесено ко второй категории сложности, поскольку оно предполагает обеспечение настройки на изменения структур входных и выходных данных, обеспечение переносимости.

Посредством коэффициента сложности учитываются дополнительные затраты труда, связанные со сложностью разрабатываемого программного продукта:

$$K_c = 1 + \sum_{i=1}^n K_i, \quad (7.5)$$

где K_i – коэффициент, соответствующий степени повышения сложности ПО за счет конкретной характеристики;

n – количество учитываемых характеристик.

В данном случае ПС характеризуется наличием одновременно двух характеристик, поэтому коэффициент сложности равен $K_c = 1,12$.

Коэффициент, учитывающий степень использования при разработке стандартных модулей. Степень использования в разрабатываемом ПО стандартных модулей определяется их удельным весом в общем объёме проектируемого продукта.

Коэффициент, учитывающий степень использования при разработке стандартных модулей, $K_t = 0,7$, поскольку степень охвата реализуемых функций разрабатываемого ПС стандартными модулями, типовыми программами и ПС находится в промежутке от 40% до 60%.

На основе общей трудоёмкости определяется плановое число разработчиков $Ч_p$:

$$Ч_p = \frac{T_o}{T_p \cdot \Phi_{эф}}, \quad (7.6)$$

где $\Phi_{эф}$ – эффективный фонд времени работы одного работника в течение года, $\Phi_{эф} = 236$ дней;

T_o – общая трудоёмкость разработки проекта, $T_o = 278.06$ чел.-дней;

T_p – срок разработки проекта, в годах, $T_p = 0,5$ лет.

Для данного проекта вначале необходимо рассчитать количество исполнителей для всего проекта в целом по формуле (7.6):

$$Ч_p = \frac{278,06}{0,5 \cdot 236} = 2,36 \text{ чел.}$$

Далее, считаем, что на всех стадиях численность разработчиков составляет 2,36 чел. Исходя из этого, посчитаем сроки для каждой из стадий проекта при заданном количестве исполнителей:

$$T_p = \frac{T_o}{Ч_p \cdot \Phi_{эф}}, \quad (7.7)$$

где $\Phi_{эф}$ – эффективный фонд времени работы одного работника в течение года, $\Phi_{эф} = 236$ в днях;

T_o – общая трудоёмкость стадии проекта, $T_o = 278,06$ чел.-дней;

$Ч_p$ – численность исполнителей проекта, $Ч_p = 2,36$ чел.

По формуле (7.7) рассчитаем сроки каждой из стадий проекта:

- для стадии «Техническое задание»: $T_{pТЗ} = \frac{34,65}{2,36 \cdot 236} = 0,062$ лет;
- для стадии «Эскизный проект»: $T_{pЭП} = \frac{22,28}{2,36 \cdot 236} = 0,04$ лет;
- для стадии «Технический проект»: $T_{pТП} = \frac{22,28}{2,36 \cdot 236} = 0,04$ лет;
- для стадии «Рабочий проект»: $T_{pРП} = \frac{141,92}{2,36 \cdot 236} = 0,255$ лет;
- для стадии «Внедрение»: $T_{pВН} = \frac{56,93}{2,36 \cdot 236} = 0,102$ лет.

В таблице 7.5 для всех стадий разработки приведены следующие показатели:

- коэффициенты удельных весов трудоёмкости;
- распределения нормативной трудоёмкости ПС;
- коэффициент сложности ПС;
- коэффициент, учитывающий использование стандартных модулей;
- коэффициент, учитывающий новизну;
- численность исполнителей;
- сроки разработки.

Все эти показатели рассчитаны отдельно для каждой стадии разработки проекта.

Таблица 7.5 – Расчёт показателей по стадиям

Показатели	ТЗ	ЭП	ТП	РП	ВН	Итого
Коэффициенты удельных весов трудоемкости, D_{cti}	0,09	0,07	0,07	0,61	0,16	1
Распределение нормативной трудоемкости ПО, T_n , чел.-дней	36,54	28,42	28,42	247,66	64,96	406
Коэффициент сложности ПО, K_c	1,12	1,12	1,12	1,12	1,12	-
Коэффициент, учитывающий новизну ПС, K_n	0,7	0,7	0,7	0,7	0,7	-
Общая трудоемкость ПС, T_y , чел.-дней	34,56	22,28	22,28	141,92	56,93	278,06
Численность исполнителей $Ч_p$, чел.	2,36	2,36	2,36	2,36	2,36	2,36

Основной статьей расходов на создание ПС является заработная плата разработчиков проекта. Общая трудоемкость, плановая численность работников и плановые сроки разработки ПС являются базой для расчёта основной заработной платы разработчиков.

Месячная тарифная ставка T_m каждого исполнителя определяется по действующей месячной тарифной ставке первого разряда T_{m1} , и тарифного коэффициента T_k :

$$T_m = T_{m1} \cdot T_k, \quad (7.8)$$

где T_m – месячная тарифная ставка, $T_m=31$ руб.;

T_{m1} – месячная тарифная ставка первого разряда, $T_{m1}=31$ руб.;

T_k – месячный тарифный коэффициент, $T_k=1$.

Часовая тарифная ставка рассчитывается по формуле:

$$T_{ч} = \frac{T_m}{\Phi_p}, \quad (7.9)$$

где Φ_p – среднемесячная норма рабочего времени в часах, $\Phi_p = 180$ часов.

Действующая месячная тарифная ставка первого разряда $T_{m1} = 31$ руб., (01.01.2019 года).

Тарифная часовая ставка составит:

$$T_{ч1} = \frac{31}{180} = 0,172 \text{ руб.}$$

Основная заработная плата исполнителей на конкретное ПС рассчитывается по формуле:

$$З_o = \sum_{i=1}^n З_{oi} = \sum_{i=1}^n T_{чи} \cdot T_{ч} \cdot \Phi_{ni} \cdot K_{пр}, \quad (7.10)$$

где n – количество разработчиков;

$З_{oi}$ – основная заработная плата i -го исполнителя;

$T_{чи}$ – часовая тарифная ставка i -го исполнителя, $T_{ч1}=0,172$ руб.;

$T_{ч}$ – количество рабочих часов в день, $T_{ч} = 8$ часов;

Φ_{ni} – плановый фонд рабочего времени i -го исполнителя, $\Phi_{ni}=182$ дня;

$K_{пр}$ – коэффициент премирования, $K_{пр} = 1,2$.

Таким образом, основная заработная плата исполнителям данного проекта составит:

$$З_o = 0,172 \cdot 8 \cdot 182 \cdot 1,2 = 300,52 \text{ руб.}$$

Дополнительная заработная плата исполнителей $З_d$ включает выплаты, предусмотренные законодательством о труде и определяется по нормативу в процентах к основной заработной плате.

Дополнительная заработная плата в целом для исполнителей данного проекта:

$$З_d = З_o \cdot Н_d = 300,52 \cdot 0,2 = 60,10 \text{ руб.} \quad (7.11)$$

где $З_o$ – основная заработная плата, $З_o = 300,52$ руб.;

$Н_d$ – норматив дополнительной заработной платы, $Н_d = 20\%$.

Определяются в соответствии с действующими законодательными актами в процентном отношении к фонду основной и дополнительной зарплаты исполнителей, определённом по нормативу, установленному в целом по организации:

$$З_{соц} = (З_o + З_d) \cdot Н_{соц}, \quad (7.12)$$

где $Н_{соц}$ – норматив отчислений в фонд социальной защиты населения, $Н_{соц} = 35\%$;

$З_o$ – основная заработная плата, $З_o = 302,52$ руб.;

$З_d$ – дополнительная заработная плата, $З_d = 60,10$ руб.

$$З_{соц} = (300,52 + 60,10) \cdot 0,35 = 122,61 \text{ руб.}$$

Расходы по статье «Материалы» определяются на основании сметы затрат проекта с учетом действующих нормативов:

$$M = H_M \cdot V_o, \quad (7.13)$$

где H_M – норма расхода материалов в расчёте на 100 LOC ПО, $H_M = 3\%$;
 V_o – общий объём ПС, $V_o = 15860$ LOC.

$$M = 0,03 \cdot 15860 / 100 = 4,758 \text{ руб.}$$

В данном проекте специальное оборудование не применялось, затрат на приобретение спецоборудования не было.

Расходы по статье «Машинное время» включают оплату машинного времени, необходимого для разработки и отладки ПС, которое определяется по нормативам (в машино-часах) на 100 строк LOC в зависимости от характера решаемых задач и типа ПК, для данного проекта по формуле:

$$P_M = C_M \cdot V_o \cdot H_{M3}, \quad (7.14)$$

где C_M – цена одного машино-часа, $C_M = 1,5$ руб.;

V_o – общий объём ПС, $V_o = 15860$ LOC;

H_{M3} – норматив расхода машинного времени на отладку 100 LOC, $H_{M3} = 3\%$.

$$P_M = C_M \cdot V_o \cdot H_{M3} = 1,5 \cdot 15860 / 100 \cdot 0,03 = 7,93 \text{ руб.}$$

Сумма расходов по статье «Научные командировки» рассчитывается по формуле:

$$P_{рнк} = 3_o \cdot H_{рнк}, \quad (7.15)$$

где 3_o – основная заработная плата, $3_o = 300,52$ руб.;

$H_{рнк}$ – норматив расходов на командировки, $H_{рнк} = 30\%$.

$$P_{рнк} = 300,52 \cdot 0,3 = 90,16 \text{ руб.}$$

Включают расходы на приобретение и подготовку специальной научно-технической информации и специальной литературы.

Прочие расходы для данного проекта:

$$P_{пр} = 3_o \cdot H_{пр}, \quad (7.16)$$

где 3_o – основная заработная плата, $3_o = 300,52$ руб.;

$H_{пр}$ – норматив прочих расходов, $H_{пр} = 10\%$.

$$P_{пр} = 3_o \cdot H_{пр} = 300,52 \cdot 0,1 = 30,05 \text{ руб.}$$

Расходы по статье «Накладные расходы» для данного проекта:

$$P_{\text{нр}} = Z_o \cdot H_{\text{нр}}, \quad (7.17)$$

где Z_o – основная заработная плата, $Z_o = 300,52$ руб.;

$H_{\text{нр}}$ – норматив накладных расходов в целом по организации, $H_{\text{нр}} = 50\%$.

$$P_{\text{нр}} = 300,52 \cdot 0,5 = 150,26 \text{ руб.}$$

Общая сумма полной себестоимости рассчитывается по формуле:

$$C_{\Pi} = Z_o + Z_d + Z_{\text{соц}} + M + P_{\text{сп}} + P_m + P_{\text{рнк}} + P_{\text{пр}} + P_{\text{нр}} \quad (7.18)$$

Получаем:

$$C_{\Pi} = 300,52 + 60,10 + 122,61 + 4,758 + 0 + 7,93 + 90,16 + 30,05 + 150,26 = 766,39 \text{ руб.}$$

Рентабельность и прибыль единицы от реализации ПС заказчику по создаваемому ПС определяется исходя из результатов анализа рыночных условий, переговоров с заказчиком и согласованию с ним отпускной цены, включающей дополнительно налог на добавленную стоимость и отчисления в местный и республиканский бюджеты. Прибыль единицы от реализации ПС заказчику рассчитывается по формуле:

$$P_{\text{ед.}} = \frac{C_{\Pi} \cdot Y_{\text{рп}}}{100}, \quad (7.19)$$

где $P_{\text{ед.}}$ – прибыль единицы от реализации ПО заказчику;

$Y_{\text{рп}}$ – уровень рентабельности ПС, $Y_{\text{рп}} = 50\%$;

C_{Π} – полная себестоимость ПС, $C_{\Pi} = 766,39$ руб.

$$P_{\text{ед.}} = \frac{766,39 \cdot 50}{100} = 383,2 \text{ руб.}$$

Рассчитаем по формуле:

$$\Pi_{\Pi} = C_{\Pi} + P_{\text{ед.}}, \quad (7.20)$$

где Π_{Π} – прогнозируемая цена без налогов, руб.;

C_{Π} – полная себестоимость ПС, $C_{\Pi} = 766,39$ руб.;

$P_{\text{ед.}}$ – прибыль единицы от реализации ПС заказчику, $P_{\text{ед.}} = 383,2$ руб.

$$\Pi_{\Pi} = 766,39 + 383,2 = 1149,59 \text{ руб.}$$

Рассчитаем по формуле:

$$\text{НДС} = \text{Ц}_{\text{п}} \cdot \text{Н}_{\text{дс}}, \quad (7.21)$$

где $\text{Ц}_{\text{п}}$ – прогнозируемая цена без налогов, $\text{Ц}_{\text{п}} = 1149,59$ руб.;
 $\text{Н}_{\text{дс}}$ – норматив НДС, $\text{Н}_{\text{дс}} = 20\%$.

$$\text{НДС} = 1149,59 \cdot 0,2 = 229,92 \text{ руб.}$$

Рассчитаем по формуле:

$$\text{Ц}_0 = \text{Ц}_{\text{п}} + \text{НДС}, \quad (7.22)$$

где $\text{Ц}_{\text{п}}$ – прогнозируемая цена без налогов, $\text{Ц}_{\text{п}} = 1149,59$ руб.;
 НДС – налог на добавочную стоимость, $\text{НДС} = 229,92$ руб.

$$\text{Ц}_0 = 1149,59 + 229,92 = 1379,51 \text{ руб.}$$

Организация-разработчик участвует в освоении ПС и несет соответствующие затраты, на которые составляется смета, оплачиваемая заказчиком по договору. Сметой предусматривается не только затраты (основная и дополнительная заработные платы, начисления на зарплату и др.), но и налоги, предусмотренные законодательством, и прибыль организации-разработчика.

Для данного проекта расходы на освоение ПС:

$$\text{Р}_0 = \text{С}_{\text{п}} \cdot \text{Н}_0, \quad (7.23)$$

где Р_0 – затраты на освоение ПС, руб.;

$\text{С}_{\text{п}}$ – полная себестоимость ПС, $\text{С}_{\text{п}} = 766,39$ руб.;

Н_0 – норматив расходов на освоение для данной организации, $\text{Н}_0 = 30\%$.

$$\text{Р}_0 = 766,39 \cdot 0,3 = 229,92 \text{ руб.}$$

Организация-разработчик осуществляет сопровождение ПС и несет соответствующие расходы, которые оплачиваются заказчиком в соответствии с договором и сметой на сопровождение. Затраты на сопровождение вычисляются по формуле:

$$\text{Р}_c = \frac{\text{С}_{\text{п}} \cdot \text{Н}_c}{100\%}, \quad (7.24)$$

где $\text{С}_{\text{п}}$ – полная себестоимость ПС, $\text{С}_{\text{п}} = 766,39$ руб.;

Н_c – норматив расходов на сопровождение, $\text{Н}_c = 20\%$.

$$P_c = 766,39 \cdot 0,2 = 153,28 \text{ руб.}$$

Заказчик оплачивает организации-разработчику всю сумму расходов по проекту, включая прибыль. После уплаты налогов из прибыли в распоряжение заказчика остаётся чистая прибыль от проекта. Ввиду того, что ПО разрабатывалось для одного объекта, чистую прибыль можно считать в качестве экономического эффекта организации-разработчика от реализованного проекта.

Общие капитальные вложения заказчика K_o , связанные с приобретением, внедрением и использованием ПС, рассчитываются по формуле:

$$K_o = K_{\text{пр}} + K_{\text{ос}} + K_c + K_{\text{тс}} + K_{\text{об}}, \quad (7.25)$$

где $K_{\text{пр}}$ – затраты на приобретение ПС по отпускной цене у разработчика с учетом стоимости услуг по эксплуатации, $K_{\text{пр}} = 0$ руб.;

K_o – затраты пользователя на освоение ПС, $K_o = 229,92$ руб.;

K_c – затраты пользователя на оплату услуг по сопровождению ПО, $K_c = 153,28$ руб.;

$K_{\text{тс}}$ – затраты на доукомплектование ВТ техническими средствами в связи с внедрением нового ПС, $K_{\text{тс}} = 0$ руб.;

$K_{\text{об}}$ – затраты на пополнение оборотных средств, $K_{\text{об}} = 0$ руб.

$$K_o = 0 + 229,92 + 153,28 + 0 + 0 = 383,2 \text{ руб.}$$

Исходные данные представлены в таблице 7.6.

Показатели экономии рассчитываются на основе исходных данных.

Рассчитаем затраты $ЗП_{3\epsilon 2}$ на заработную плату в расчёте на одну задачу в новом варианте:

$$ЗП_{3\epsilon 2} = \frac{З_{\text{см}} \cdot T_{\text{с2}}}{D_{\text{рс}} \cdot T_{\text{ч}}}, \quad (7.26)$$

где $T_{\text{с2}}$ – средняя трудоёмкость работ в расчёте на одну задачу в новом варианте, $T_{\text{с2}} = 1$ чел.-часов;

$З_{\text{см}}$ – среднемесячная заработная плата одного программиста, $З_{\text{см}} = 200$ руб.;

$T_{\text{ч}}$ – количество часов работы в день, $T_{\text{ч}} = 8$ часов;

$D_{\text{рс}}$ – среднемесячное количество рабочих, $D_{\text{рс}} = 21,5$ дней.

$$ЗП_{3\epsilon 2} = \frac{200 \cdot 1}{21,5 \cdot 8} = 1,16 \text{ руб.}$$

Рассчитаем затраты ЗП₃₂ на заработную плату в расчёте на объём выполненных работ в новом варианте:

$$\text{ЗП}_{32} = \text{ЗП}_{3\epsilon 2} \cdot A_2, \quad (7.27)$$

где ЗП_{3ε2} – затраты на заработную плату в расчёте на одну задачу, ЗП_{3ε2} = 1,16 руб.;

A₂ – объём выполненных работ без использования нового ПС, A₂=300 задач.

$$\text{ЗП}_{32} = 1,16 \cdot 300 = 348 \text{ руб.}$$

Таблица 7.6 – Исходные данные для расчёта экономии ресурсов

Показатель	Источник информации	Обозначение	Ед. изм.	Сумма показателя в базовом варианте	Сумма показателя в новом варианте
1	2	3	4	5	6
Капитальные вложения, включая затраты пользователя на приобретение ПС	Договор заказчика с разработчиком	K _п р	руб. .	-	0
Затраты на освоение ПС	Договор заказчика с разработчиком	K _о	руб. .	-	227,78
Затраты на сопровождение ПС	Договор заказчика с разработчиком	K _с	руб. .	-	151,86
Затраты на укомплектование ВТ техническими средствами в связи с внедрением нового ПС	Сметы затрат на внедрение	K _{тс}	руб. .	-	0

Продолжение таблицы 7.6.

Затраты на пополнение оборотных средств в связи с эксплуатацией нового ПС	Сметы затрат на внедрение	$K_{об}$	руб.	-	0
Время простоя сервиса, обусловленное ПС, в день	Расчётные данные пользователя и паспорт ПО	Π_1, Π_2	мин.	20	5
Количество типовых задач, решаемых за год	План пользователя	$З_{т1}, З_{т2}$	задач	300	300
Объём выполняемых работ	План пользователя	A_1, A_2	задач	300	300
Средняя трудоёмкость работ в расчёте на 1 задачу	Рассчитывается по данным пользователя	$T_{с1}, T_{с2}$	чел.-часов на 1 задачу	2	1
Количество часов работы в день	Принято для расчёта	$T_ч$	ч	8	8

Рассчитаем заработную плату с учетом начисления на зарплату в новом варианте:

$$ЗП_{н2} = ЗП_{32} \cdot K_{нз}, \quad (7.28)$$

где $ЗП_{32}$ – затраты на заработную плату в новом варианте без учета начислений, $ЗП_{32} = 348$ руб.;

$K_{нз}$ – коэффициент начислений на зарплату, $K_{нз} = 1,5$.

$$ЗП_{н2} = 348 \cdot 1,5 = 522 \text{ руб.}$$

Рассчитаем затраты $З_{с2}$, связанные с простоем сервиса в новом варианте:

$$З_{с2} = \frac{П_2 \cdot Д_{рг} \cdot С_{п}}{60}, \quad (7.29)$$

где $П_2$ – время простоя сервиса, обусловленное ПС в минутах за день в новом варианте, $П_2=5$ минут за день;

$Д_{рг}$ – плановый фонд работы сервиса, $Д_{рг} = 225$ дней;

$С_{п}$ – стоимость одного часа простоя сервиса, $С_{п} = 10$ руб.

$$З_{с2} = \frac{5 \cdot 225 \cdot 10}{60} = 187,5 \text{ руб.}$$

Рассчитаем затраты $ЗП_{3e1}$ на заработную плату в расчёте на одну задачу в базовом варианте:

$$ЗП_{3e1} = \frac{З_{см} \cdot Т_{с1}}{Д_{р} \cdot Т_{ч}}, \quad (7.30)$$

где $Т_{с1}$ – средняя трудоёмкость работ в расчёте на одну задачу в базовом варианте, $Т_{с1} = 2$ чел.-часов;

$З_{см}$ – среднемесячная заработная плата одного программиста, $З_{см} = 200$ руб.;

$Т_{ч}$ – количество часов работы в день, $Т_{ч} = 8$ часов;

$Д_{р}$ – среднемесячное количество рабочих дней, $Д_{р} = 21,5$ дней.

$$ЗП_{3e1} = \frac{200 \cdot 2}{21,5 \cdot 8} = 2,33 \text{ руб.}$$

Рассчитаем затраты $ЗП_{31}$ на заработную плату в расчёте на объём выполненных работ в базовом варианте:

$$ЗП_{31} = ЗП_{3e1} \cdot A_1, \quad (7.31)$$

где $ЗП_{3e1}$ – затраты на заработную плату в расчёте на одну задачу, $ЗП_{3e1} = 2,33$ руб.;

A_1 – объём выполненных работ без использования нового ПО, $A_1 = 300$ задач.

$$ЗП_{31} = 2,33 \cdot 300 = 699 \text{ руб.}$$

Рассчитаем заработную плату с учетом начисления на зарплату в базовом варианте:

$$ЗП_{Н1} = ЗП_{31} \cdot K_{НЗ}, \quad (7.32)$$

где $ЗП_{31}$ – затраты на заработную плату в базовом варианте без учета начислений, $ЗП_{31} = 699$ руб.;

$K_{НЗ}$ – коэффициент начислений на зарплату, $K_{НЗ} = 1,5$.

$$ЗП_{Н1} = 699 \cdot 1,5 = 1048,5 \text{ руб.}$$

Рассчитаем затраты $З_{с1}$, связанные с простоем сервиса в базовом варианте:

$$З_{с1} = \frac{П_1 \cdot Д_{рг} \cdot C_{п}}{60}, \quad (7.33)$$

где $П_1$ – время простоя сервиса, обусловленное ПС в минутах за день в базовом варианте, $П_1 = 20$ минут за день.

$Д_{рг}$ – плановый фонд работы сервиса дней, $Д_{рг} = 225$ дней.

$C_{п}$ – стоимость одного часа простоя сервиса, $C_{п} = 10$ руб.

$$З_{с1} = \frac{20 \cdot 225 \cdot 10}{60} = 750 \text{ руб.}$$

Рассчитаем по формуле:

$$C_{н} = ЗП_{Н1} - ЗП_{Н2}. \quad (7.34)$$

Получаем:

$$C_{н} = 1048,5 - 522 = 526,5 \text{ руб.}$$

Рассчитаем экономию за счет сокращения простоев сервиса:

$$C_c = З_{с1} - З_{с2}. \quad (7.35)$$

Получаем:

$$C_c = 750 - 187,5 = 562,5 \text{ руб.}$$

Рассчитаем общую готовую экономию текущих затрат, связанных с использованием нового ПС:

$$Э_i = C_{н} + C_c \quad (7.36)$$

Получаем:

$$\mathcal{E}_i = 526,5 + 562,5 = 1089 \text{ руб.}$$

7.3 Оценка результата от использования ПС

Рассчитаем интегрированный экономический эффект $\Pi_{\text{ч}}$ для пользователя в итоге использования нового ПС по формуле:

$$\Pi_{\text{ч}}^{1,2,3,4} = \mathcal{E}_i \cdot \left(1 - \frac{H_{\Pi}}{100}\right), \quad (7.37)$$

где \mathcal{E}_i – общая готовая экономия текущих затрат, связанных с использованием нового ПС, $\mathcal{E}_i = 1089$ руб.;

H_{Π} – ставка налога на прибыль, $H_{\Pi} = 18\%$.

$$\Pi_{\text{ч}}^{1,2,3,4} = 1089 \cdot \left(1 - \frac{18}{100}\right) = 892,98 \text{ руб.}$$

Денежные средства получаемые и затрачиваемые в разные моменты времени, имеют разную стоимость. Возможность соизмерения разновременных денежных потоков достигается путём дисконтирования. Процесс дисконтирования предполагает определение нормы дисконта и коэффициента дисконтирования.

Расчёт требуемой нормы дисконта $E_{\text{н}}$ осуществляется с использованием кумулятивного метода по следующей формуле:

$$E_{\text{н}} = r + s + \sum_{i=1}^n g_i, \quad (7.38)$$

где r – реальная (без учета компенсации за инфляцию) безрисковая ставка ссудного процента;

s – инфляционное ожидание за период t , рассчитанное как среднее за расчётный период проекта;

i – факторы риска;

n – количество факторов риска;

g – премия за отдельный риск по конкретному фактору.

За расчётный период принят 2019 год.

Применительно к рассматриваемому проекту реальная ставка ссудного процента составит $r = 0,07\%$. Инфляционную премию к безрисковой ставке можно принять в размере $s = 0,05\%$. Также к расчёту приняты факторы риска, приведённые в таблице 7.7.

Таблица 7.7 – Факторы риска

№	Фактор риска	Премия
1	Возможное влияние непредвиденных обстоятельств на величину ставки процента	0,01
2	Риск падения спроса	0,01
3	Риск изменения дохода	0,01

$$E_n = r + s + \sum_{i=1}^n g_i = 0,07 + 0,05 + 0,01 + 0,01 + 0,01 = 0,15 \quad (7.39)$$

Рассчитаем коэффициенты дисконтирования по формуле (7.38):

- 2020 год, $t = 1$, $\alpha_1 = (1 + E_n)^{t-1-t} = (1 + 0,15)^{1-1} = 1$;
- 2021 год, $t = 2$, $\alpha_2 = (1 + E_n)^{t-1-t} = (1 + 0,15)^{1-2} = 0,8695$;
- 2022 год, $t = 3$, $\alpha_3 = (1 + E_n)^{t-1-t} = (1 + 0,15)^{1-3} = 0,7561$;
- 2023 год, $t = 4$, $\alpha_4 = (1 + E_n)^{t-1-t} = (1 + 0,15)^{1-4} = 0,6575$.

Интегрированный экономический эффект $\mathcal{E}_{\text{инт.}}$, руб. рассчитывается по формуле:

$$\mathcal{E}_{\text{инт.}} = \sum_{t=1}^n \text{ЧДД} = \sum_{t=1}^n (P_t \cdot \alpha_t - Z_t \cdot \alpha_t), \quad (7.40)$$

где n – расчётный период, лет;

P_t – чистый доход, полученный в году t , $P_t = 892,98$ руб.;

Z_t – затраты (инвестиции) в году t , $Z_t = 383,2$ руб.

α_t – коэффициент дисконтирования.

По формуле (7.40) экономический эффект равен:

$$\mathcal{E}_{\text{инт.}} = (892,98 \cdot 1 - 383,2 \cdot 1) + (892,98 \cdot 0,8695 - 383,2 \cdot 0,8695) + (892,98 \cdot 0,7561 - 383,2 \cdot 0,7561) + (892,98 \cdot 0,6575 - 383,2 \cdot 0,6575) = 220,38 \text{ руб.}$$

Коэффициент дисконтирования α_t рассчитывается по формуле:

$$\alpha_t = (1 + E)^{t-1}, \quad (7.41)$$

где E_n – требуемая норма дисконта, $E_n = 15\%$;

t_p – расчётный период;

t – период, потоки которого приводятся к расчётному.

Таблица 7.8 – Расчёт интегрированного экономического эффекта

Показатель	Обозн.	Ед. изм.	2020	2021	2022	2023
1	2	3	4	5	6	7
Затраты		руб.	1039	-	-	-

Продолжение таблицы 7.8.

Чистая прибыль	$P_{\text{ч}}$	руб.	892,98	892,98	892,98	892,98
То же с учетом фактора времени	$P_t \cdot \alpha_t$	руб.	892,98	776,45	675,18	587,13
Чистый дисконтированный доход	ЧДД	руб.	-383,2	443,26	675,18	587,13

Программное средство окупается на второй год, когда обеспечивается чистая прибыль 443,26 руб. В последующие года наблюдается рост чистой прибыли от использования программного средства (см. таблицу 7.8).

Таким образом, использование разработки и внедрения программного средства информационный портал по управлению и реализации продукции компании является перспективным для коммерческого успеха.

ЗАКЛЮЧЕНИЕ

В ходе дипломного проектирования была создана информационная система по управлению и реализации продукции компании. Данная система представляет собой веб-приложение, используемое для продажи продукции и управления.

Оно позволит продвигать и продавать товары, уменьшить издержки на рекламу и приобрести дополнительный рынок сбыта своего товара, расширить клиентскую аудиторию и, как следствие, увеличить прибыль, а удобное управление из административной панели позволит осуществить эффективную работу для персонала.

Результаты тестирования веб-приложения показали, что система работает без ошибок: стабильное соединение с сервером, совместимость со всеми популярными браузерами, правильная работа всех модулей, ссылок.

Были спроектированы: диаграмма деятельности, диаграмма вариантов использования, диаграмма классов, логическая и физическая модели базы данных, основные алгоритмы работы. Диаграмма деятельности отображает основную последовательность действий пользователя с программным средством. Диаграмма вариантов использования (сценариев поведения, прецедентов) является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Данная диаграмма состоит из актеров, вариантов использования и отношений между ними. Диаграмма классов отображает основные классы и входящие в нее методы с используемыми переменными. Логическая модель базы данных отображает логическую взаимосвязь между данными. Физическая модель жестко связана с базой данных и в ней отображены реальные типы данных, ограничения, реальные названия таблиц базы данных.

Приобретены практические умения использования PHP, HTML и CSS, JS, PostgreSQL, навыки настройки и управления Linux Server через виртуальную машину.

По итогам проектирования, было создано удобное веб-приложение, которое исправно выполняет поставленные задачи и имеет широкую область применения в счет ее универсального алгоритма работы. Оно адаптивно по отношению к экрану используемых устройств, универсально для любой продукции компаний, безопасно для использования, менее затрано реальных магазинов, удобно в управлении, эффективно в работе.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- [1] Casada [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.casada.ru/> – Дата доступа: 18.01.2020
- [2] Yamaguchi [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.yamaguchi.by/> – Дата доступа: 20.01.2020
- [3] Разработка веб-ресурса [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://m.habr.com/ru/post/273795/> – Дата доступа: 20.02.2020
- [4] Основа интернет-магазина [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://m.habr.com/ru/company/trinion/blog/286188/> – Дата доступа: 20.02.2020
- [5] Построение UML-диаграмм [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://prog-cpp.ru/uml-classes/> – Дата доступа: 20.02.2020
- [6] Основы для проектирования баз данных [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://habr.com/ru/post/193136/> – Дата доступа: 20.02.2020
- [7] Защита от несанкционированного доступа [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://habr.com/ru/company/pentestit/blog/282860/> – Дата доступа: 20.02.2020
- [8] Построение алгоритма работы [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://kvodo.ru/algorithm-programmy.html/> – Дата доступа: 20.02.2020
- [9] Основы для проектирования баз данных [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://habr.com/ru/post/193136/> – Дата доступа: 20.02.2020
- [10] Примеры модулей для сайта и описание [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://serpstat.com/ru/blog/kak-najti-harakternye-tematike-funkcionalnye-moduli/> – Дата доступа: 20.02.2020
- [11] Тестирование программного средства – основы [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://habr.com/ru/post/279535/> – Дата доступа: 20.02.2020
- [12] Яргер, Р. Дж. MySQL и mSQL: Базы данных для небольших предприятий и Интернета / Р. Дж. Яргер. – СПб: Символ-Плюс, 2013. – 560 с.
- [13] Горовой, В. Г. Экономическое обоснование проекта по разработке программного обеспечения : метод.пособие / В. Г. Горовой, А. В. Грицай, В.А. Пархименко. – Минск : БГУИР, 2014. – 12 с.
- [14] Горбушие, А. М. Экономический эффект программного продукта / А. М. Горбушие. – М : ВШ, 2007. – 275 с.