

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

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ

Автоматизированная система конфигурирования ПК для компьютерного магазина через web-интерфейс

Выпускная квалификационная работа студента

Очной формы обучения
направления подготовки_02.03.03 Математическое обеспечение и
администрирование информационных систем

4 курса группы 07001302
Очиенг Питер

Научный руководитель
к.т.н., доцент Ю.Г. Чашин

БЕЛГОРОД 2017

СОДЕРЖАНИЕ

Введение	4
1. ОБЗОР И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ, И СРЕДСТВ КОМПЬЮТЕРОВ ПРОЕКТИРОВАНИЯ САЙТА КОНФИГУРАТОРА	8
1.1 Актуальность данной работы	8
1.2 Анализ принципов построения информационных систем для торговли через интернет.....	11
1.3 Анализ принципов построения электронных магазинов.....	12
1.4 Интернет-технологии в электронной коммерции	13
1.5 Анализ существующих веб-сервисов для составления конфигураций компьютеров.....	17
1.6 Требования к разрабатываемая программная обеспечения	23
2 РАЗРАБОТКА СТРУКТУРЫ ПОСТРОЕНИЯ ЭЛЕКТРОННОГО КОНФИГУРАЦИИ КОМПЬЮТЕРОВ МАГАЗИНА	24
2.1 Архитектура электронного магазина конфигурации компьютеров.....	24
2.2 Разработка архитектуры интернет-магазина	24
2.2.1 Описания структура Интернет магазина конфигурации компьютеров.....	25
2.3 Выбор среды разработки программного обеспечения	27
2.3.1 Системы управления контентом (CMS).....	27
2.3.2 Выбор среды разработки.....	29
2.3.3 Технологии Веб-сервер.....	31
2.4 Выбор языка программирования	34
2.5 Виртуальные серверы (наборы разработчиков)	36

3 ПРОЕКТИРОВАНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ КОНФИГУРАТОРА.....	39
3.1 Разработка базы данных конфигулятора	39
3.2 Выбор Базы данных PostgreSQL	42
3.3 Проектирование иерархии диаграмм потоков данных.....	45
3.4. Разработка Веб -интерфейса конфигулятора.....	49
3.5. Реализация алгоритмы работы конфигулятора.....	51
3.6. Реализация структуры программы конфигулятора.....	56
3.7 Безопасность.....	62
4 ТЕСТИРОВАНИЕ СОЗДАННОГО ПО.....	63
4.1 Интерфейс Web-приложения.....	63
4.1.1 Тестовые данные.....	63
ЗАКЛЮЧЕНИЕ.....	69
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	71
ПРИЛОЖЕНИЕ	73

ВВЕДЕНИЕ

За последнее десятилетие компьютер стал неотъемлемой частью жизни любого современного человека. Компьютер – это универсальный инструмент, способный выполнять как рабочие, так и развлекательные функции. Уже давно перестав быть дорогостоящей роскошью, современный компьютер стал вполне доступной необходимостью.

Сегодня на рынке представлено огромное количество компьютеров и комплектующих для них. Вследствие такого разнообразия все чаще встает вопрос о том, какой купить компьютер и сколько это может стоить. В зависимости от планируемых задач, можно купить хороший компьютер с дорогими комплектующими и высокой производительностью – для мультимедийного использования; а можно купить компьютер дешево – для выполнения несложных офисных задач. Любителям современных интерактивных развлечений стоит купить игровой компьютер – это мощные машины, для которых практически нет ограничений в производительности. В любом случае, разнообразие существующих продуктов позволит вам купить компьютер под любые нужды.

Успех любого бизнеса во многом зависит от того, насколько эффективно компания доводит свою информацию до клиентов и партнеров.

Создание интернет-сайта и размещение его в Интернете – один из комплексных методов позиционирования компании и информирования целевой аудитории.

Именно в Интернете многие ищут подробную, свежую информацию, на основе которой можно получить представление об интересующей компании, товарах и услугах.

В настоящее время, в связи с ростом пользователей Интернет, увеличением пропускной способности каналов связи и удешевлением доступа в Интернет, увеличивается и количество используемых интернет-сервисов.

Сейчас трудно представить крупную компанию без интернет-сайта. Наличие у компании хорошего сетевого представительства не только усиливает положительный образ фирмы, но и говорит в пользу надежности и стабильности компании, создает положительный имидж.

Преимущества конфигураций компьютера и его отношения к интернет-маркетингу:

- 1) не требует затрат на аренду площади, найма продавцов и другого персонала;
- 2) доступ к виртуальным витринам магазина может получить любой покупатель, не зависимо от места проживания;
- 3) интернет-магазин не имеет ограничений на виртуальную площадь;
- 4) владелец интернет-магазина может сдавать в аренду свои виртуальные торговые площади так же, как и владелец обычного магазина;
- 5) срок и стоимость создания интернет-магазина несоизмеримо ниже, чем обычного магазина;
- 6) потенциальный покупатель получает исчерпывающую информацию об интересующем его товаре или услуге без помощи посредников (продавца, менеджера);
- 7) интернет-магазин, работающий в режиме онлайн, предполагает обратную связь с покупателем (форумы, чаты, гостевые книги), что позволяет оперативно ответить на вопросы пользователей, провести маркетинговые исследования, получить отзывы и пожелания;
- 8) можно разместить сколь угодно много товаров или описать любое количество услуг;

9) возможность делать покупки, не выходя из дома, делают интернет-магазины еще более привлекательными для покупателей. Делая свой выбор в пользу открытия интернет-магазина, компания переходит на более высокий уровень развития своего бизнеса и делает свой ассортимент более доступным для широкого круга покупателей.

Темой исследования выпускной квалификационной работы является автоматизированная система конфигурирования ПК для компьютерного магазина через web-интерфейс.

Актуальность задачи автоматизации учета компьютерного конфигурации и другого оборудования на предприятии возрастает при наличии большого парка компьютеров, офисной техники, торгового и другого оборудования. Наибольшая потребность знать, где и какая единица находится, оперативно отслеживать изменения, связанные с оборудованием, возникает у ИТ-подразделений.

Особую значимость задача автоматизации учета оборудования приобретает на крупных предприятиях. Обработка все время растущих массивов информации стала возможна только с использованием современных компьютерных технологий.

Организовать систему учёта техники на предприятии, вести учёт компьютеров и комплектующих, сейчас невозможно без дополнительно установленного на компьютер программного обеспечения.

Целью выпускной квалификационной работы является разработка Автоматизированная информационная система конфигурирования персонального компьютера через Web-интерфейс.

Для достижения этой цели необходимо решить следующие задачи:

- 1) Анализ современного состояния и тенденций развития интернет-технологий;
- 2) Анализ существующих сайтов;

- 3) Формирование требований к системе и постановка задачи на разработку;
- 4) Выбор инструментальных средств технологий разработки;
- 5) Проектирование и реализация базы данных;
- 6) Проектирование и разработка веб-конфигуратора для комплектующий конфигуратора персональная компьютера;
- 7) Функциональное моделирование контекстной диаграммы и диаграмм декомпозиций бизнес – процесса (IDEF0) «Учет компьютерной техники предприятия»;
- 8) Апробация и внедрение результатов исследования.

В качестве методов будут использоваться анализ методов разработки сайта и синтез проектных решений.

В ходе выполнения данной выпускной квалификационной работы, необходимо разработать приложения по продаже сборки компьютера. В нем должен быть необходимый перечень компьютерных аксессуаров, а также интуитивно понятный пользователю веб-интерфейс.

1.ОБЗОР И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ, ИСРЕДСТВ ПРОЕКТИРОВАНИЯ САЙТА КОНФИГУРАЦИИ КОМПЬЮТЕРОВ

1.1.Актуальность данной работы

Одной из основных тенденцией современного развития общества является распространение Интернета и постоянное увеличение количества его пользователей.

Доминирующим и наиболее динамично развивающимся элементом строящегося информационного общества является всемирная сеть Интернет, которая, как следует ожидать, сохранит статус глобальной информационной среды в ближайшее время. По прогнозам аналитиков и экспертов, число пользователей этой глобальной сети будет неуклонно расти [5, с.56].

В 2016 году аудитория и проникновение интернета в России продолжили рост, причём темпы роста сохранялись на уровне 2014 года.

По данным ФОМ, осенью 2016 года 74 миллиона россиян (65% совершеннолетнего населения страны) пользовались интернетом хотя бы раз в месяц, а почти 63 миллионов человек выходили в сеть ежедневно.

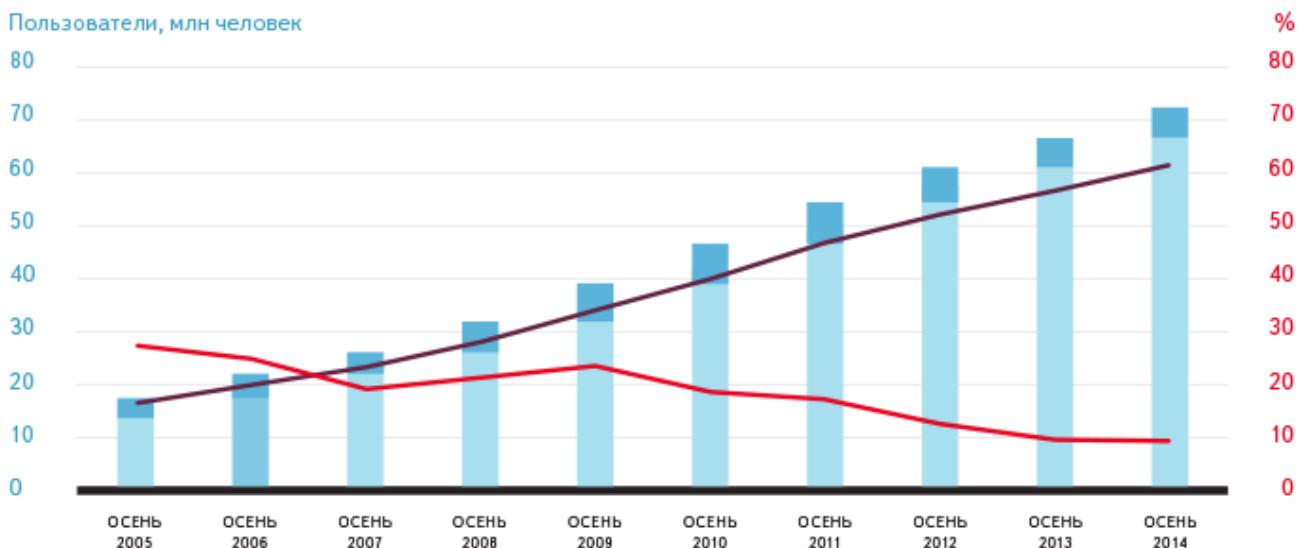
При росте самого сетевого сообщества также активно будет развиваться и рынок интернет-рекламы, который может предоставить большие возможности для продвижения любого товара или услуги. Особенности представления информации в интернете, а также появление и развитие новых форм рекламы в интернете позволяют достигать максимально тесного взаимоотношения между разными компаниями (как коммерческими, так и некоммерческими) и основной целевой аудиторией при минимальных издержках [11].

При этом важно понимать, что сайт должен соответствовать всем текущим новым решениям в плане дизайна, содержания и возможностям интерфейса [14].

На рис. 1.1 представлено ...

АУДИТОРИЯ И ПРОНИКНОВЕНИЕ ИНТЕРНЕТА В РОССИИ

■ Прирост за год, млн человек — Проникновение, %
 ■ Пользователей, млн человек — Годовой прирост числа пользователей, %



Здесь и далее, если не оговорено иное, аудитория интернета — это все пользователи, которые выходят в сеть не реже одного раза в месяц с любого устройства, включая мобильные.
 Проникновение интернета — доля людей, пользующихся интернетом хотя бы раз в месяц, среди населения страны.

Рис. 1.1. Изменение численности интернет-аудитории

За год в стране появилось почти 6 миллионов новых пользователей, это как четыре Новосибирска. Ещё сильнее выросла дневная аудитория — на 6,7 миллиона человек. Уровень проникновения интернета в России самый высокий в СНГ и БРИКС, но ниже, чем в других странах «Большой восьмёрки» и странах Балтии.

С функционированием огромного количества приложений, созданных на базе различных технологий в современных бизнес-системах, возникает задача объединения данных приложений и предоставления пользователю всех типов информации и сервисов. Задача веб-интеграции заключается в том, чтобы

объединить разнородные веб-приложения и системы в единую среду на базе сети Интернет, а также обеспечить защиту предоставляемой информации.

На рис. 1.2 изображено ...

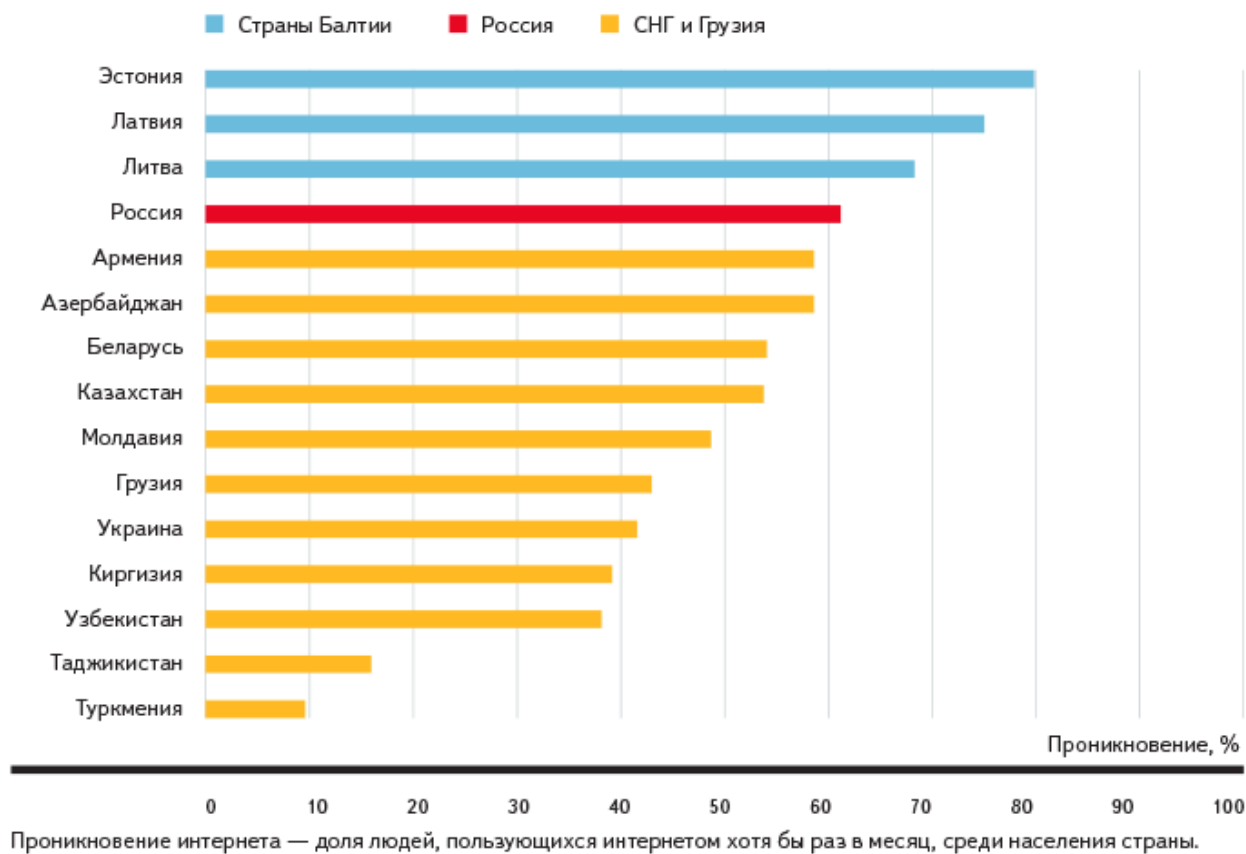


Рис 1.2. Аудитория и проникновение интернета в разных регионах в миру

По состоянию на февраль 2015 года 59% организаций, зарегистрированных в Справочнике Яндекс. Карт, указали, что у них есть свой сайт. Год назад таких организаций было 58%. Выше всего доля организаций, имеющих сайт, в Санкт-Петербурге, Москве и Центральном федеральном округе. При этом рост в регионах (кроме Крыма) продолжается, а в столицах замедлился.

При росте самого сетевого сообщества также активно будет развиваться и рынок интернет-рекламы, который может предоставить большие возможности для продвижения любого товара или услуги. Особенности представления информации в интернете, а также появление и развитие новых форм рекламы в

интернете позволяют достигать максимально тесного взаимоотношения между разными компаниями (как коммерческими, так и некоммерческими) и основной целевой аудиторией при минимальных издержках [11].

В любом случае современная интернет-аудитория более образована и финансово обеспечена, нежели любая другая аудитория другого СМИ, практически все руководители и топ-менеджеры активно пользуются услугами интернет для наладки коммуникационных каналов к источникам нужных данных. Таким образом, эффективность интернета в качестве средства для передачи рекламной информации очень высока, поскольку в любом случае есть большая вероятность получить нужный контакт с целевым потребителем, недели при использовании обычных традиционных средств коммуникации [13].

Помимо этого, в условиях непредсказуемой рыночной среды и переходе информационному обществу даже самой небольшой компании очень важно иметь свое веб-представительство, обычно в виде стандартного сайта, чтобы поддерживать конкурентоспособные отношения и иметь доступ к новым клиентам. При этом важно понимать, что сайт должен соответствовать всем текущим новым решениям в плане дизайна, содержания и возможностям интерфейса [14].

1.2. Анализ принципов построения информационных систем для торговли через интернет

На сегодняшний день, путешествуя по сети Интернет, можно найти много очень хорошо выполненных сайтов, которые быстро загружаются, привлекательны на вид, имеют четко разграниченную по разделам информацию и удобные функции навигации. Такие сайты просто приглашают вас посетить их, расслабиться, задержаться на некоторое время, развлечься и узнать что-то новое.

Скажем звучит определение веб-сайта. «Совокупность электронных документов требует очень скрупулезной над собой работы. Целью создания веб-сайтов является предоставление информации населению. В современных рыночных условиях информация занимает огромное место. С каждым днем разрабатываются все новые и новые веб-сайты, которые увеличивают передача свежей информации и предоставляют широко распространённый возможности. Но для того, чтобы веб-сайт имел высокую степень посещения необходимо позаботиться о его привлекательности, грамотно организованной структуре, а также об актуальности, размещенной на нем информации [17].

1.3. Анализ принципов построения электронных магазинов.

Успешный веб-сайт - это очень эффективный инструмент торговли - он способен привлечь внимание потенциальных клиентов, как и любой другой инструмент маркетинга, основанный на принципе немедленного ответа, он должен, прежде всего, заинтриговать потенциального посетителя, а затем побудить его принять меры. Тем не менее, многие клиенты выходят эту функцию домашней страницы, что часто приводит к тому, что часто клиенты не остаются на месте в течение длительного времени и оставить его, как только они прибывают. Такие сайты, даже если они иногда содержат много ценных советов и статей, не всегда удается достичь ожидаемого уровня посещаемости, не говоря уже о результатах продаж.

Если вы делаете только несколько изменений, простой веб-сайт может оказаться более последовательным и эффективным инструментом. Следует помнить, что каждый день для потенциальных клиентов залит информацией из различных рекламных сообщений, и что с точки зрения завоевания их интересов настолько, насколько это возможно, существует жесткая

конкуренция. Веб-сайт, который может привлечь внимание и вызвать некоторое любопытство, поощрять клиентов не только увидеть оставшиеся страницы и делать покупки, но и неоднократно в течение посетить эти страницы и снова через некоторое время, и рекомендовать их друзьям или перенести эти страницы в социальных медиа, следовательно, это приводит к увеличению продаж.

Поэтому что видит пользователь, который посетил магазин?

Наполнение корзины, клиент пружина команды «Оформить заказ» и выбрать удобную форму оплаты. Если он делает покупку в магазине в первый раз, они, как правило, просят предоставить некоторые данные о себе -. Имя, номер телефона, адрес и другие корпоративные покупатели сообщили название компании, номер счета, имя и номер телефона контактного лица. На этом этапе заказчик присваивается определенный идентификационный код. Это делается для того, что, когда он будет идти в магазин в следующий раз, вся информация может не войти в выше - введите код достаточно. Дальнейшие расчеты осуществляются и прямая передача товара покупателю.

Существуют многочисленные формы оплаты: наличными курьеру при доставке, банковским переводом (для организаций), банковский перевод, электронные платежи, оплата наложенным платежом или почтовым и телеграфным переводом.

1.4.Интернет-технологии в электронной коммерции

Информация в современном обществе - являются наиболее важными факторами производства, которая напрямую зависит от конкурентоспособности и эффективности любой организации или бизнеса. Основной целью информации является развитие качества жизни за счет увеличения производительности и повышения условий их занятости.

Современные информационные и коммуникационные технологии проникают во все сферы жизни, значительно меняется способов производства и услуг, а также форм досуга, реализации гражданских прав человека, методов и форм обучения и подготовки кадров. Лучшие и улучшенные технологии оказывают положительное решающее влияние на мировую экономику, политику, развитие общественных институтов во всем мире расширение и совершенствование развития роста.

Основу экономики, современные информационные и коммуникационные технологии проникают во все сферы жизни, значительно меняется способов производства и услуг, а также форм досуга, реализации гражданских прав человека, методов и форм обучения и подготовки кадров. Высокие технологии оказывают решающее влияние на экономику, политику и развитие общественных институтов.

Большинство из которых принесет переход к глобальной информационной экономике следующие достижения в области информационных технологий: расширение прав и влияния на глобальную сеть Интернет; Развитие и рост электронной коммерции и цифровых маркетинговых инструментов; преимущественное использование баз данных в различных областях производства в обществе; широкая доступность современных цифровых технологий (Wi-Fi, LP, Интернет и т.д.).

Ведущим и наиболее динамичным элементом в построении информационного общества является всемирная сеть Интернет, который можно было бы ожидать, чтобы сохранить свой статус в качестве глобальной информационной среды в ближайшем будущем. По мнению аналитиков и экспертов, число пользователей этой глобальной сети будет продолжать расти во всем мире в ближайшем будущем с улучшением современной технологии.

В результате, по данным исследовательской компании TNS в феврале 2013 года; по крайней мере один раз в месяц в Интернете приходит 76500000.

Россияне (или 53% от общей численности населения). [7] По данным ФОМ, в российских ежедневных интернет-пользователей около 50 миллионов человек. Часть активных пользователей Интернета, что приходит к сети, по крайней мере один раз в день в настоящее время составляет 45%, или 55,1 млн людей. В то же время ежегодный прирост интернет-пользователей выходит в сеть, по крайней мере один раз в месяц, до 13%, а для ежедневной аудитории эта цифра составляет 16%. [8]

Во всех федеральных округах России, распределение всего интернет-пользователей, а также более чем 50% от всего населения России. Число интернет-пользователей в России в 2013 году заняла первое место в Европе, которая ранее была оккупирована Германией, и на шестом месте в мире. Первое место в мире является Китай, где интернет-пользователи составляют 600 миллионов людей, а затем в США, Япония, Индия и Бразилия. В то же время, около 40 миллионов людей в России не являются пользователями сети, поэтому потенциал роста интернет сохраняются. Существенные особенности связаны с более старшей возрастной группой; число пользователей выросло за прошлый год на 54% [6].

Принимая во внимание фактор роста сетевого сообщества, а также будет расти и рынок интернет-рекламы, который предлагает возможности для продвижения любых видов товаров и услуг. Информационные особенности Интернета, а также развитие новых форм интернет-рекламы, позволяют достичь наиболее тесный контакт между различными организациями, что является как коммерческие, так и некоммерческие организации, и целевой аудитории на взаимной минимизации затрат.

Зрители формируются через Интернет и материально гарантированы по сравнению с аудиторией любых других средств распространения информации; большинство руководителей и топ-менеджеры используют Интернет в качестве канала связи и источника информации. Таким образом, Интернет в качестве

канала для эффективности передачи рекламной информации высок, поскольку вероятность получения контакта с целевым потребителем несравненно больше, чем при использовании традиционных методов коммуникации.

Колебания, происходящие во внешней торговой среде, способствовали тому, что предприятия для повышения эффективности его деятельности были вынуждены изменить философию управления. В то же время, как показывает бизнес-практика для устойчивой прибыли компании разрабатывать и производить свою деятельность в направлении маркетинговой концепции в соответствии с законом общества, который в настоящее время является основой для эффективного управления, и подавляющее большинство современные компании хотят работать на этой основе. [10]

Поэтому работающие предприятия и организации должны убедить своих потенциальных клиентов, что при рассмотрении компания может удовлетворить их потребности лучше, чем их конкуренты. Фокус на решение проблем потребителей и очевидно, опыт компании-клиента - два ключевых ингредиента для успеха на рынке. Однако, этого недостаточно. Крайне важно, чтобы информация о содействии фирмы или организации и производит продукцию и услуги достигают потребитель в короткий промежуток времени и с ограниченными ресурсами. Это эффективно решает проблему интернета.

В настоящее время многие компании пытаются делать бизнес в Интернете, внедрили методы электронной коммерции или электронного бизнеса, но не все понимают, в то же время, в какой форме она будет идеально подходит для них. Главной особенностью бизнес-стратегии - ориентация на клиента. Используя Интернет, компания может быть еще ближе к потребителю; более тонко и точно организовать индивидуальное обслуживание для клиентов, партнеров и клиентов. Именно это новое качество взаимодействия с клиентами приводят к возникновению интереса к интернет-среде, в большинстве компаний и происхождения, что это работает в Интернете.

Конечно, Интернет является рекордсменом по скорости распространения среди когда-либо существовавших СМИ. На протяжении последних лет сек Интернет захватил 50% населения США (по сравнению с телефонным потребовалось 35 лет, телевидение - 26 лет, радио - 22 года, мобильной связи - 13 лет). К началу 2001 года число пользователей Интернета было 360 миллионов человек во всем мире за последние 11 лет этот показатель вырос в 5 раз. Основная область, где 44% всех интернет-пользователей, является Азия (922 млн человек), а самым быстрорастущим регионом в плане интернет-издания является Африка, где число пользователей за 11 лет увеличилось в 25 раз (с 4,5 до 118 миллионов людей) [7].

Эта статистика показывает, что Интернет является самым быстрорастущим среды в истории человечества и в последние годы все чаще становится социально-экономической системы, а не просто информация. Благодаря огромной аудитории, что можно изучать, организовывать и сегмент, все больше и больше компаний и организаций осознают необходимость использования сетевых средств для продвижения своих товаров и услуг. Все это привело к появлению новых научных концепций в области экономики и управлении и эффективное средство коммуникации - интернет-маркетинге.

1.5. Анализ существующих веб-сервисов для составления конфигурации компьютеров

В ходе выполнения выпускной квалификационной работы было проведено подробное исследование существующих интернет-магазинов, предлагающих компьютерные конфигураторы для своих потенциальных клиентов. В результате были обнаружены две компании, существующие уже более 10 лет на цифровом рынке - это DNS «www.dns.ru» и «www.edelws.ru» Эти две компании предлагают продукцию самого высокого качества, широко участвуют в

рекламной деятельности, проводят различные акции и другие общественные мероприятия, которые способствуют продвижению компании и её продукции на рынке. Сайты этих компании имеют веб-сервис для подбора конфигурации компьютерных комплектующих, после анализа работы веб-сервисов подбора комплектующих был определен ряд недостатков. Поэтому было решено создать дополнительный сервис подбора компьютерных комплектующих: веб-сайт, который является полностью функциональным, лишенным вышеописанных недостатков, привлекательным, обладающим удобным пользовательским интерфейсом.

Удобство для постоянных клиентов заключается в том, что они могут узнать информацию о товарах, услугах, ценах, количестве товаров, складе, специальных акциях и новых позициях, просто посетив веб-сайт компании в Интернете. Там они смогут узнать больше о компании, оставить заявку на временно отсутствующие аксессуары.

Web-сайт сможет так же приносить дополнительный доход при продаже продукции через сеть Интернет покупателям, находящимся в других городах. Заинтересовав клиента, относительно невысокими ценами, широким ассортиментом скоростью обработки заказов и быстрой доставкой

22 февраля 2017 года аналитическая компания IDC опубликовала некоторые итоги исследования российского рынка персональных компьютеров за 2016 год и отметила улучшение ситуации на нем. При этом годовые продажи сократились.

В 2016 году на российский рынок ПК было поставлено 4,47 млн десктопов и ноутбуков, что почти на 8% меньше показателя годичной давности. Во втором полугодии выросли закупки оборудования в корпоративном секторе, а в потребительском сегменте ситуация улучшилась благодаря уменьшению конкурентного давления со стороны планшетов. В результате в октябре-декабре 2016 года поставки ПК в Россию увеличились на 4,2% относительно

аналогичного периода предыдущего года и достигли 1,34 млн штук. В деньгах рост оказался еще больше — на 14,3%.

Принципы построения интернет –магазина DNS «DNS»

Интернет – магазин «DNS» располагается в сети интернет по адресу: www.DNS.ru. Он выполнен по технологии базирующей свои услуги на Windows NT-системах, с доступом к MS SQL. Web-приложение написано, как набор файлов (ASP), представляющих собой расширение HTML, в который добавлены инструкции на языках JScript. Эти инструкции выполняются на сервере при запросе данной страницы и управляют формированием ответа пользователю.

В ходе проведенного исследования пользовательского интерфейса сайта интернет-магазина DNS, можно сделать вывод, что пользовательский интерфейс очень сложен в использовании и не слишком удобен для его потенциальных клиентов.

Существующий сайт был разработан на бесплатной системе управления содержимым WordPress. Просмотр Главной страницы сайта приведен на рис. (1.3).

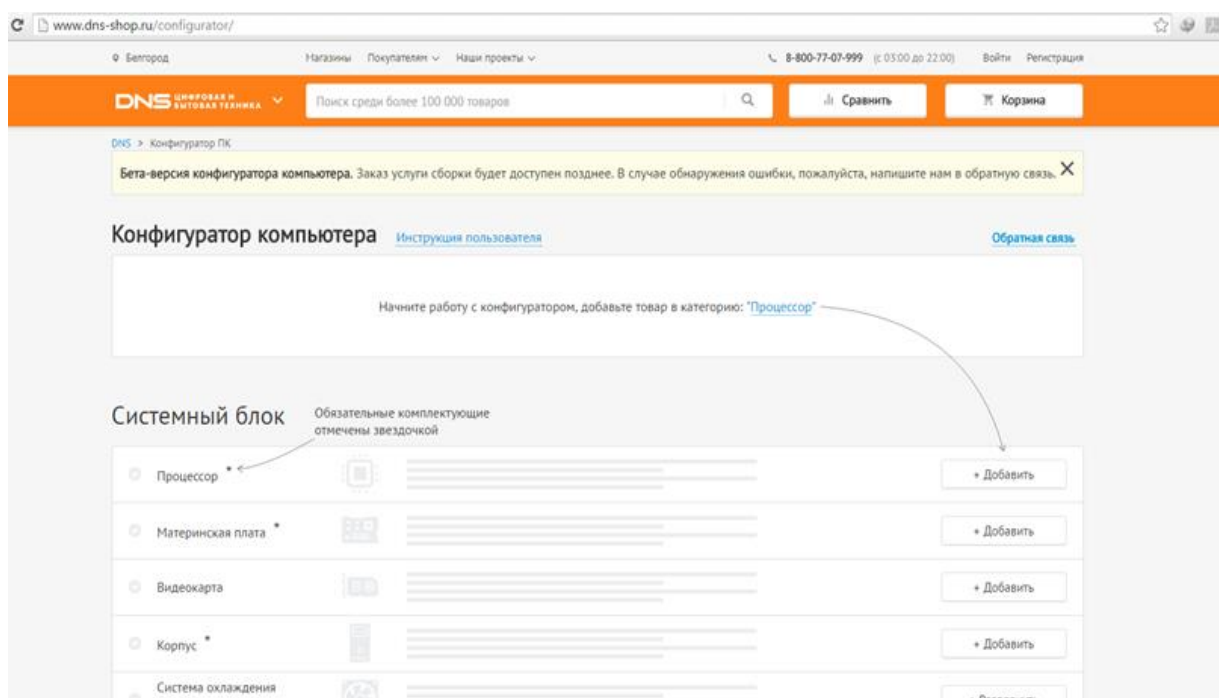


Рис 1.3 Главная страница сайта интернет-магазина «DNS»

Интернет – магазин «edelws» располагается в сети интернет по адресу: www.edelws.ru. Он выполнен по технологии, базирующей свои услуги на 1С-Битрикс, с доступом к nginx 1.8. Web-приложение написано, как набор файлов(ASP) с поддержкой Apache HTTP сервера, представляющих собой как бы расширение HTML, Js, jQuery, в который добавлены инструкции на языках JScript и другие. Эти инструкции выполняются на сервере при запросе данной страницы и управляют формированием ответа пользователю.

Система управления содержимым (CMS) 1С-Битрикс. Для хранения данных сайта используется файловая система сервера и реляционная СУБД. Поддерживаются следующие СУБД: MySQL, Oracle, MS SQL. Продукт работает на Microsoft Windows и UNIX-подобных платформах, включая Linux.

В ходе проведенного исследования пользовательского интерфейса сайта интернет-магазина «edelws», можно сделать вывод, что пользовательский интерфейс так же, как и интерфейс DNS очень сложен в использовании и не слишком удобен для его потенциальных клиентов. Главная страница сайта приведена на рис. (1.5).

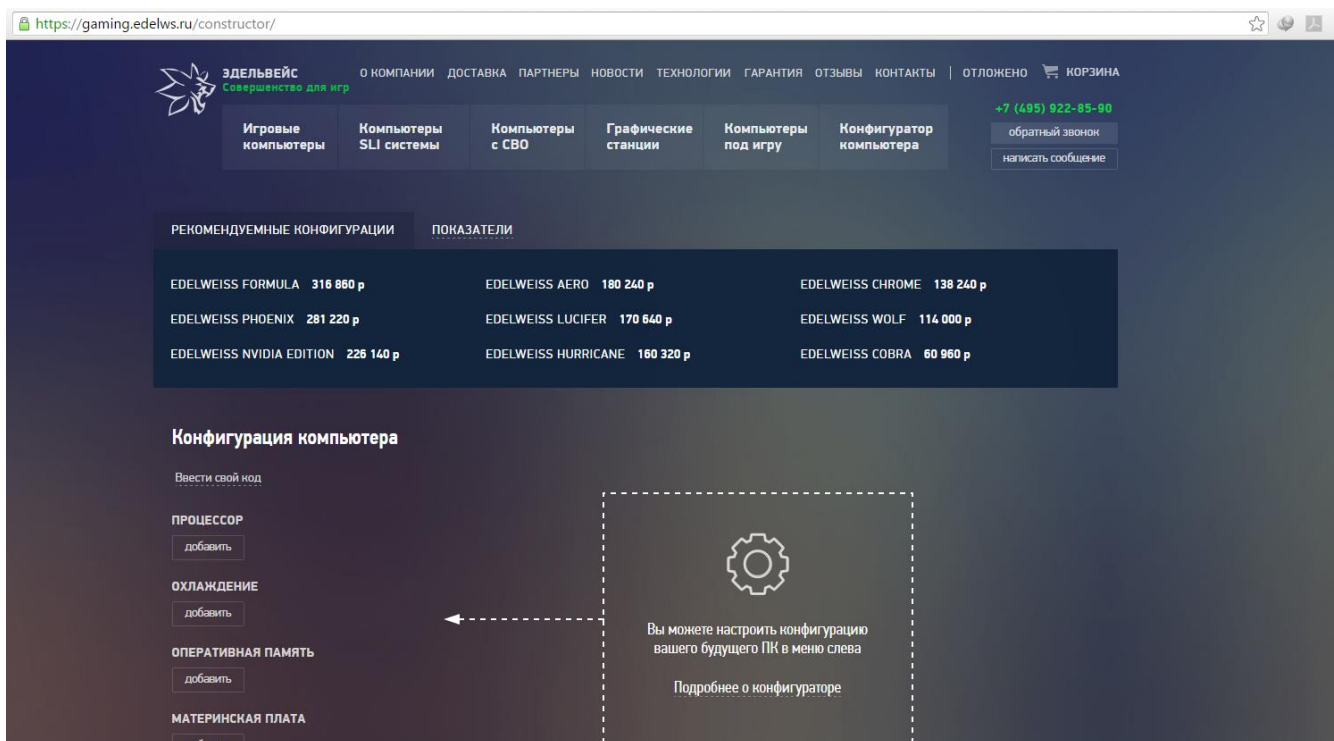


Рис 1.5. интернет магазина «www.edelws.ru»

На существующем сайте отсутствует удобная система администрирования. Существующая система администрирования представлена панелью администратора системы Ucoz. Данная система администрирования имеет только одну роль – администратор, который добавляет, редактирует, удаляет контент сайта. Отсутствует разграничение по ролям. Ограничение по ролям позволяет определенной роли только добавлять контент, другой роли только редактировать и т.д. Разделение по ролям очень удобно, так как каждый выполняет свои функции и не имеет доступа к другим функциям. Также на существующем сайте введены различные ограничения, так как система Ucoz имеет несколько пакетов функциональности и "фишек", и все пакеты кроме базового, уже платные. Существует трудность в изменении дизайна сайта и можно выбрать тему только из существующего набора шаблонов, свою тему установить невозможно (только в платных пакетах). Наличие различной рекламы является также проблемой при использовании базового пакета.

Выполненный Анализ текущих тенденций строительства веб-сайт означает, что для того, чтобы посетитель или потребитель чувствовать себя более комфортно с соответствующей предоставленной информацией, сайт не может быть просто полезным и качественной информацией. Поэтому необходимо создать и обеспечить гармоничное и красивый дизайн сайта и содержание одного и того же, чтобы поместить информацию о нем, а также создать вокруг информационного сообщества, увеличивая, таким образом, привлечение посетителей на сайт в процессе формирования контента. Идеальная система, основанная на принципе «содержания + сообщество» будет иметь положительный результат воздействия. Чем больше людей, участвующих в сообществе, тем больше содержания они создают и сайт привлекает все больше посетителей. Для тех, кто владеет веб-сайт, актуальным вопросом управления ресурсами: наполнение его контентом, общение с пользователями в Интернете и т.д. С этой целью, - даты возможность практически неподготовленных людей, чтобы управлять их веб-ресурсов - были созданы различными CMS (Content Management System). На сегодняшний день эти системы являются огромным числом.

На основе анализа, сравнение преимуществ и недостатков, а также функциональности, а также по просьбе Генерального директора компании, было выбрано

Это приложение является большим осуществлением управления и работы с модулями, шаблонами, компонентами и расширениями. Таким образом, позволяет быстро и с минимальными усилиями со стороны разработчика, чтобы создать рабочий интернет-магазин, большой корпоративный сайт для крупной компании или информационного портала.

1.6. Требования к разрабатываемому программному обеспечению

Перед тем, как выполнять операции по улучшению программной Поддержки конфигулятора, рассмотрим требования, предъявляемые к разрабатываемой информационной системе. Требования к программному обеспечению:

Программа должна отвечать требованиям надежности функционирования. Программа должна нормально функционировать на персональных компьютерах, минимальные системные требования которых превышают рекомендованные системные требования для поддержки любого графического карта. Программа не должна иметь привязки к аппаратной части для возможности переноса ее на новую платформу из – за неизбежные старения компьютерной техники.

Программная оболочка должна иметь интуитивно ясный дружественный интерфейс и не должна требовать от пользователей специальной подготовки, не связанной с их профессиональными обязанностями. Программа, как и база знаний должна иметь возможность наращивания информационной мощности.

2.РАЗРАБОТКА СТРУКТУРЫ ПОСТРОЕНИЯ ЭЛЕКТРОННОГО МАГАЗИНА КОНФИГУРАЦИИ КОМПЬЮТЕРОВ

2.1. Архитектура электронного магазина конфигурации компьютеров.

Архитектура сайта - систематизация информации и навигации по ней для того, чтобы более эффективно помочь посетителям найти нужные им данные более легко и успешно. Хорошо продуманная грамотная архитектура сайта гарантирует, что пользователи будут тратить меньше времени на поиск нужной информации.

Разработка архитектуры сайта должна осуществляться с учетом наиболее важной информации с точки зрения продвижения товаров и услуг на интернет-рынке. В процессе создания структуры нового сайта или оптимизации существующей структуры, необходимо сосредоточить внимание потребителей на одной и той же информации, а также для управления трафиком сайта потенциальных клиентов в наиболее важных разделах сайта в соответствии с позиционированием на рынке, продвижение товаров и услуг.

Правильное распределение приоритетов между разделами и страницами сайта, что делает их главной точкой входа на сайт, который позволит потенциальному пользователю быстро найти необходимую информацию о необходимых продуктах и услугах и повысить успешность бизнеса в Интернете, следовательно, больше продаж на продукты.

2.2. Разработка архитектуры интернет-магазина

Архитектура интернет-магазина должна быть проста и интуитивно удобна. И состоит из Клиентской части, Программной части и

Администрирования. Схема архитектуры интернет-магазина показана на рис. 2.1.

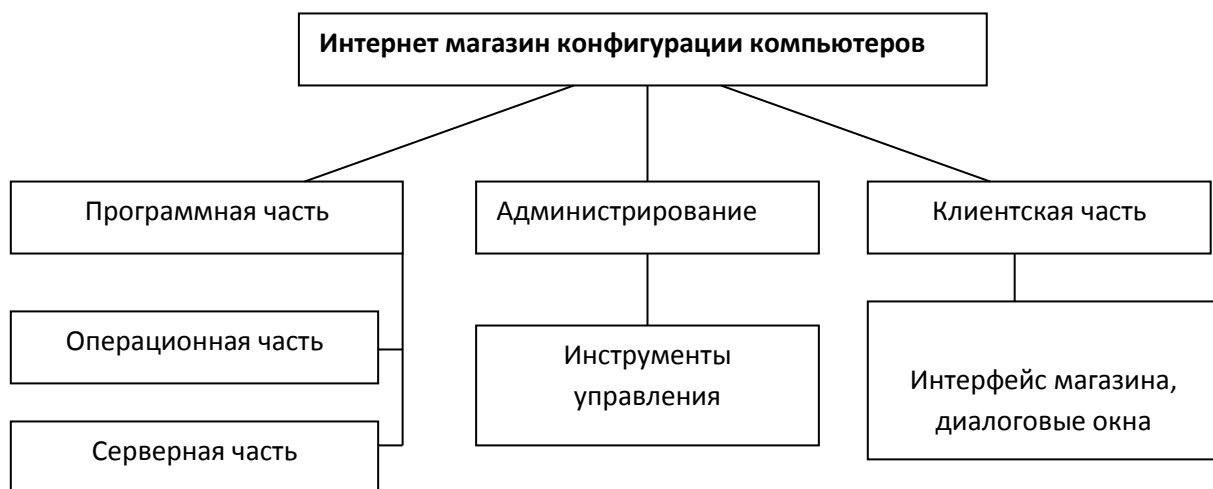


Рис 2.1. Архитектура интернет–магазина конфигурации компьютеров

2.2.1. Описание структуры Интернет-магазина конфигурации компьютеров

Программная часть архитектуры интернет–магазина конфигурации компьютеров рассматривается как взаимосвязь операционной и серверной части.

В операционной части рассматривается среда разработки интернет-магазина.

Серверная часть содержит в себе размещение интернет магазина на сайте провайдера, поддерживающие технологии, используемые при создании интернет магазина конфигурации компьютеров.

В связи с функционированием огромного количества приложений, созданных на базе различных технологий в современных бизнес-системах, возникает задача объединения данных приложений и предоставления пользователю всех типов информации и сервисов. Задача веб-интеграции

заключается в том, чтобы объединить разнородные веб-приложения и системы в единую среду на базе сети Интернет, а также обеспечить защиту предоставляемой информации.

Сравнительная характеристика использования основных технологий в различных сферах электронной коммерции показала, что наиболее используемыми технологиями в электронной коммерции являются JAVA, PHP, PYTHON и RUBY (таблица 1.1). Таблица 1.1

Применение web-технологий в электронной коммерции

Технология сферы электронной коммерции	ASP.NET	PHP	Python	Ruby	JAVA	HTML	JAVA SCRIPT
сайт	+ -	+	+ -	+ -	+	+	+
портал	+ -	+	+ -	+ -	+	+	+
аукцион	+ -	+	+ -	+ -	+	+	+
экспертная система	+	+	+	+	+	-	-
система прогнозирования	+	+	+	+	+	-	+
обучающая система					+	-	+
информационная система	+	+	+	+	+	+	+
аналитическая система			+	+	+	-	+

Практическое использование информационных технологий в области электронной коммерции позволяет более эффективно вести бизнес, т.е. снизить издержки производства, внедрить эффективные маркетинговые исследования, автоматизировать процессы информации о клиентах и продаж, анализ рынка, а также моделирования и прогнозирования бизнес-процессов, управление

персоналом и выбрать более оптимальную стратегию бизнеса для роста бизнеса, кроме того, это увеличивает скорость производства, следовательно, больше продаж.

Используя правый канал и подход к использованию Интернета в системе распределения, компания обязана заранее, чтобы попытаться решить все возможные проблемы для своих клиентов, и главный результат не должен быть никаких конфликтов, а также рост объема продаж в и на новой целевой группы клиентов.

2.3. Выбор среды разработки программного обеспечения

2.3.1. Системы управления контентом (CMS)

Рассмотрим существующие системы управления контентом, которые, как правило, используются для разработки веб-представительства. Учитывая все затраты, очень важно сделать правильный выбор системы управления контентом.

Основным критерием выбора должно быть соответствие информационной модели системы требованиям, предъявляемым к сайту. Информационная модель структура того, из чего с точки зрения системы состоит сайт. Ошибка в выборе информационной модели может привести к многократному увеличению сложности внесения изменений на сайт и практической невозможности дальнейшего его развития.

Проанализируем существующий рынок системы управления контентом для того, чтобы понять, возможно ли использование какой-либо из них для разработки веб-конфигуратора.

Существует несколько их типов: двигатель жизни Paid данных, 1-C Bitrix, Free WordPress, Joomla, каждый из них имеет свои преимущества и недостатки

[1]. В настоящее время наиболее часто используемый и популярный CMS - это WordPress, Joomla, Drupal. Многие системы управления специально разработаны для решения конкретных задач, так что перед установкой CMS, необходимо правильно оценить, какие из них лучше всего подходит для создания веб-сайта организации. Основные требования, предъявляемые к организации или бизнес-сайт, включает в себя следующее:

- 1) Дизайн-читаема информация и создание позитивного отношения;
- 2) Доступность для USER- обратной связи через сайт;
- 3) Возможность дальнейшего развития веб-сайта, использование различных плагинов, и как следствие, легкость управления, наличие дополнительных модулей и простоты подключения;

Сильная безопасность для взлома и вирусов, которые могут повлиять на систему;

- 4) Возможности поиска поисковой оптимизации;
- 5) Простота мониторинга трафика с помощью системного администрирования.

Анализ имеющейся информации о системах управления показывает, что большинство веб-разработчиков в России создавать организации среди сайтов бесплатных движателей предпочитают использовать WordPress и Joomla, в том числе оплаченного несомненным лидером является 1С-Битрикс.

Рейтинг CMS на ноябрь 2015 года по данным сайта <http://track.ruward.ru/> приведен на рис 2.2.

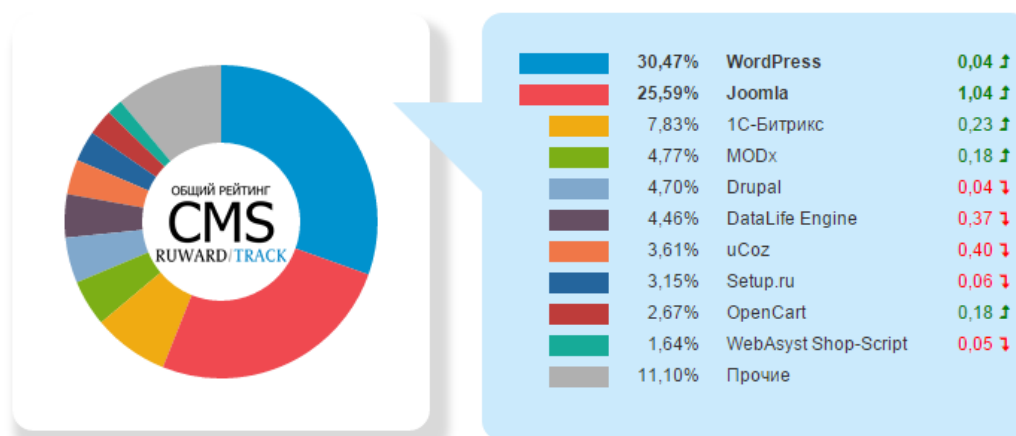


Рис 2.2. Рейтинг CMS на ноябрь 2015 года

Дальше рассмотрим

Рейтинг бесплатных CMS на ноябрь 2015 года приведен на рис 2.3.

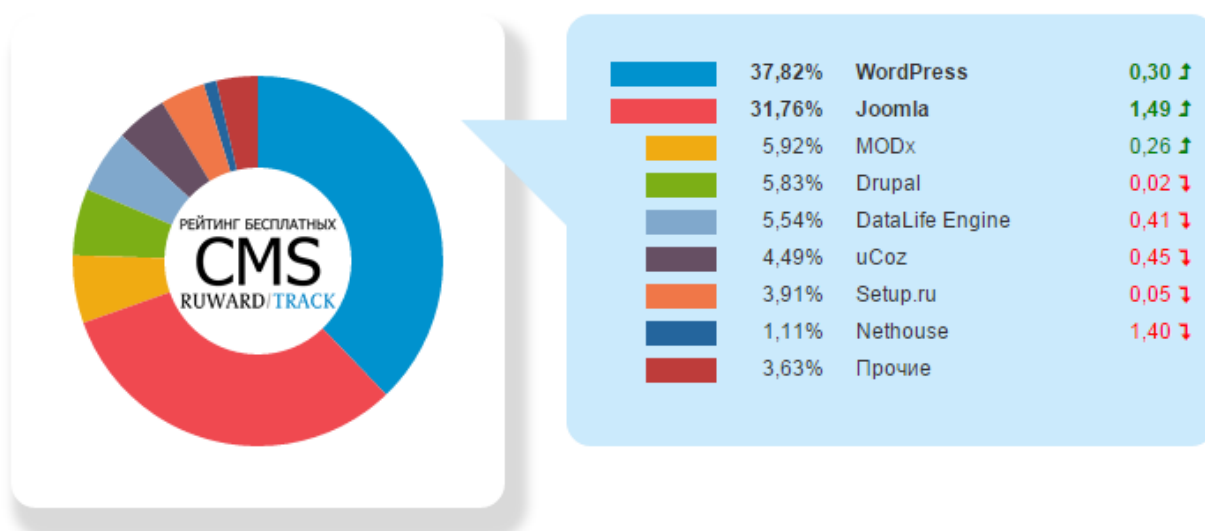


Рис 2.3. Рейтинг бесплатных CMS на ноябрь 2015 года

2.3.2 Выбор среды разработки

В этом приложении планируется использование программного обеспечения текстовый редактор с именем intelij IDEA. Редактор является

мощным инструментом для создания и редактирования исходного кода. Как и любой другой редактор IDE, он поддерживает базовые функции, такие как закладки, контрольные точки, подсветка синтаксиса, завершение кода, изменение масштаба, складывание блоков кода, и т.д. Там, однако, множество дополнительных функций, таких как макросы, выделенный TODO-элементы, анализ кода, намерение действия, умный и быстрая навигация, и многое другое.

Чтобы настроить среду редактирования, используйте страницу параметров редактора и его дочерних страниц. Существует также команда быстрого переключателя Схема, которая позволяет изменять цветовые схемы, темы, ключевые карты и т.д. с парой нажатий клавиш.

Редактор вкладках. Все операции с закладками редактора доступны из контекстного меню вкладки, или из окна | Редактор вкладок узла главного меню ниже на рис 2.4.

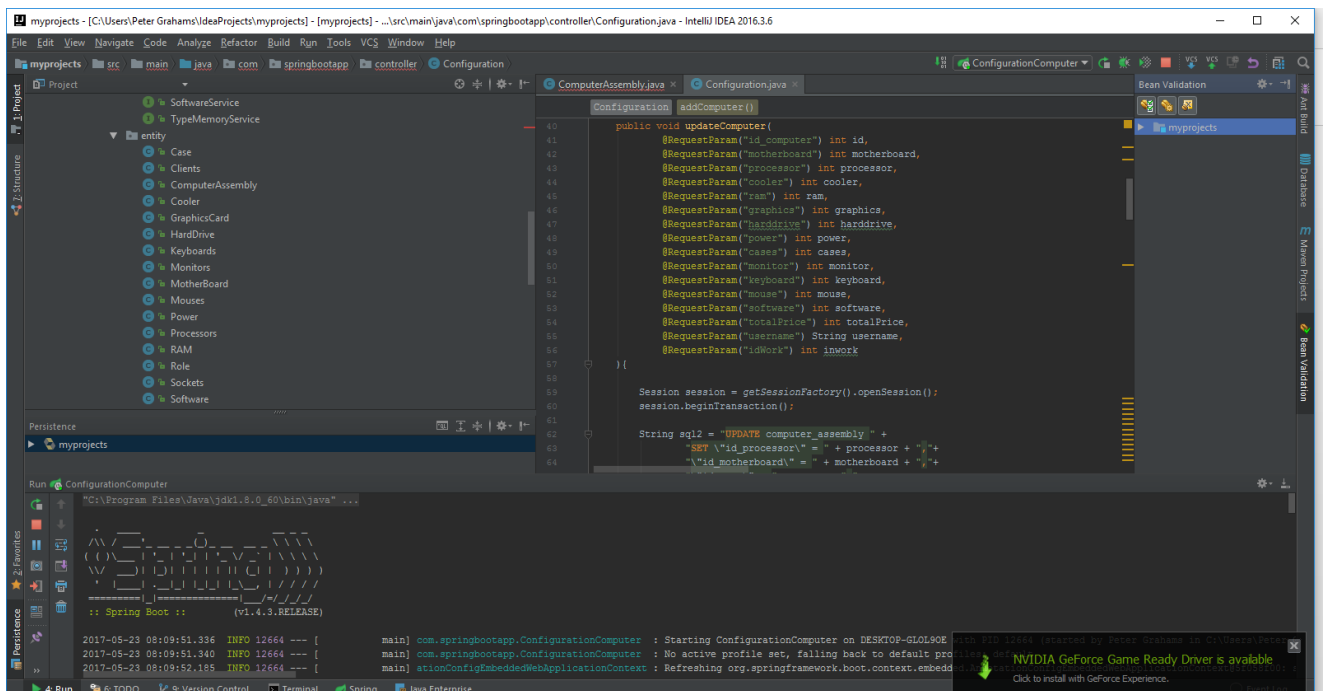


Рис 2.4. Интерфейс Редактор (IDE)

2.3.3. Технологии Веб-сервер

Веб-сервер – это программное обеспечение, отвечающее за прием запросов браузеров, поиск указанных файлов и возвращение их содержимого.

Основная функция веб-сервера для хранения, обработки и доставки веб-страниц для клиентов. Связь между клиентом и сервером происходит с помощью протокола передачи гипертекста (HTTP). Страницы, доставленные наиболее часто HTML документы, которые могут включать в себя изображения, таблицы стилей и скрипты в дополнение к текстовому наполнению.

В то время как основная функция заключается в обслуживании контента, полная реализация HTTP также включает в себя способы получения контента от клиентов. Эта функция используется для представления веб-форм, в том числе загрузки файлов. Есть два основных типа веб-серверов, обычно используемые веб-хосты. В ходе разработки выпускной квалификацию, были реализовали с использованием Apache Tomcat. Tomcat позволяет запускать веб-приложения, содержит ряд программ для самого конфигурирования. Tomcat используется в качестве самостоятельного веб-сервера, в качестве сервера контента в сочетании с веб-сервером Apache HTTP Server. Ниже показывает Доступ к базе данных с помощью Apache Tomcat Рис 2.5.

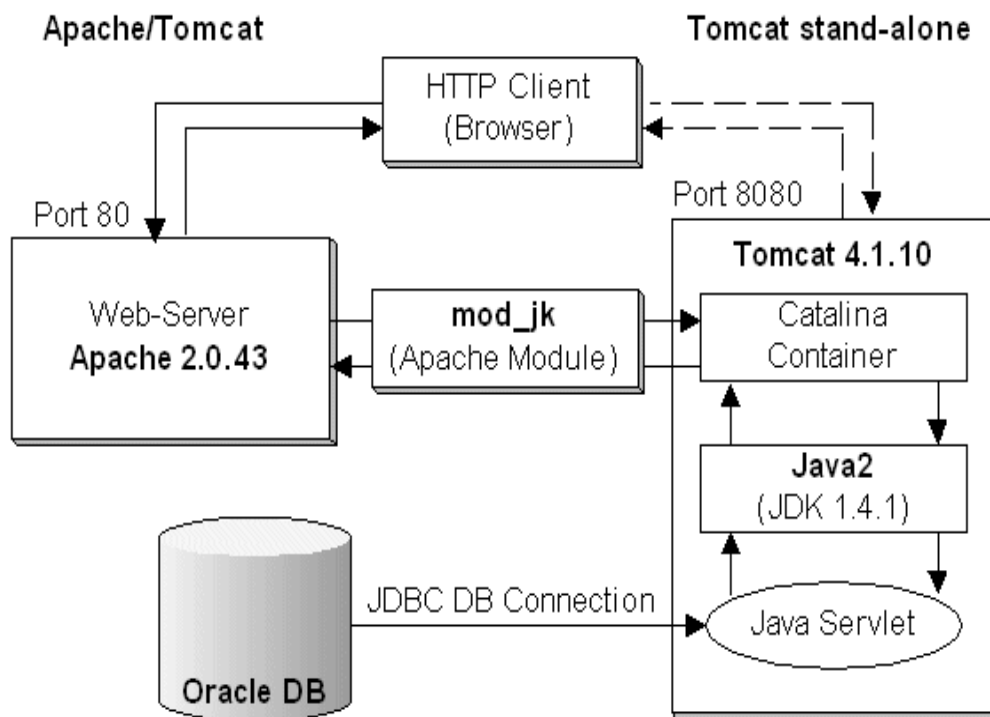


Рис 2.5. Доступ к базе данных с помощью Apache / Tomcat

Самый популярный веб-сервер в Интернете сегодня является Apache. Он был самым популярным с апреля 1996 и согласно результатам опроса, Netcraft [7], проведенной в ноябре 2005 года "более 70% сайтов в Интернете используют Apache, что делает его более широкое применение, чем все другие веб-серверы вместе взятые".

Эти веб-серверы Apache, работающие на веб-серверах UNIX и IIS, работающих на Windows. Каждый веб-сервер будет отличаться в скриптовые технологии, которые он поддерживает, так оба сервера должны быть сопоставлены, чтобы определить, какая платформа, предложенная CMS будет работать дальше.

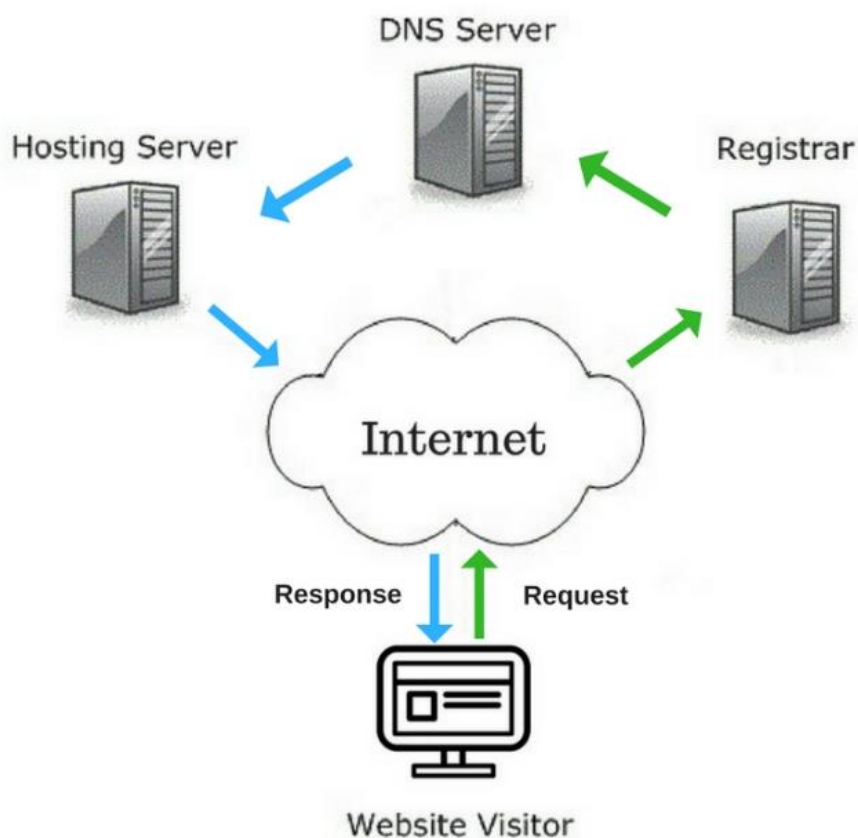


Рис 2.6. Технологии веб-сервера

Веб-сервер — сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-потокком или другими данными.

Клиент, которым обычно является веб-браузер, передаёт веб-серверу запросы на получение ресурсов, обозначенных URL-адресами. Ресурсы — это HTML-страницы, изображения, файлы, медиа-потокки или другие данные, которые необходимы клиенту. В ответ веб-сервер передаёт клиенту запрошенные данные. Этот обмен происходит по протоколу HTTP.

2.4. Выбор языка программирования

JAVA

В процессе выбора, какой язык программирования для использования, необходимо определить требования к среде программирования. В процессе требований, анализа и классификации наиболее прогнозируемым ИС были конкретно следующие требования к среде разработки:

- 1) Кроссплатформенный, необходимые для интеграции с другими подсистемами
- 2) Гибкость изменения, разработать информационную систему без каких-либо дополнительных затрат.
- 3) Безопасность как важный фактор среди прочих.
- 4) Язык программирования должен предоставлять различные возможности для работы с современными системами баз данных.
- 5) Это должно обеспечить всеобщие клиентские приложения.

Java - строго типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (позже приобретенных Oracle). Java приложения, как правило, транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуры с использованием Java-машины.

Основная цель спецификации - для обеспечения масштабируемости приложений и целостности данных во время работы системы. J2EE в основном сосредоточены на его использование через Интернет в сети Интернет и локальных сетей.

Платформа J2EE использует модель многоуровневого распределенного приложения. Логически приложение разделено на компоненты в зависимости от их функциональности. Различные компоненты, составляющие J2EE-приложение, установленное на разных компьютерах, в соответствии с их

уровнем в многоуровневой среде J2EE, в котором компонент принадлежит. J2EE-приложения состоят из компонентов. J2EE-законченный компонент представляет собой функциональный программный модуль интегрирован в приложение J2EE с соответствующими классами и файлами и взаимодействия с другими компонентами. J2EE-спецификация определяет следующие J2EE-компоненты:

- 1) Клиентские приложения и апплеты - его компоненты, работающие на клиентской машине.
- 2) Компоненты Java-сервлетов и технологии Java Server Pages (JSP) - Веб-компоненты, которые выполняются на сервере.
- 3) Предприятие бобы - это бизнес-компонент, который работает на сервере.

J2EE-компоненты написаны на языке программирования Java и скомпилированы таким же образом, как и любой другой Java-программы.

Спецификация Java EE предоставляет большие возможности для разработки клиент-серверных приложений, и особенно удобные способы реализации на стороне сервера Java-приложений. Клиент является частью наиболее часто основаны на стандарте HTML-страниц, дополненных Java-апплетов и клиентских приложений на основе.

JavaScript. JavaScript является самым популярным на стороне клиента язык сценариев, и как все языки сценариев может быть встроен в HTML. Это объектно-ориентированное язык программирования, который свободно доступен и может быть использован для добавления интерактивности на веб-страницах. Он является открытым исходным кодом и кросс системная платформа поддерживается обоими серверами Windows, и Linux, что делает его реальным выбором для моего проекта.

Вместо того, чтобы скрипт обрабатывается на сервере, вся обработка осуществляется в браузере клиента. Все основные браузеры, такие как IE,

Mozilla, Firefox, Netscape и Opera будет достаточно иметь возможность обрабатывать эти сценарии без необходимости применения JavaScript. JavaScript также может определить, какой браузер пользователь использует и загрузить страницу разработанный специально для этого браузера. Это выгодно, поскольку это обеспечит членов всех сторонников просматривать лучшую возможную версию сайта.

С точки зрения развития JavaScript является языком у меня нет опыта использования. Она тем не менее охватывает подобный синтаксис, что и Java, и C, для которых у меня есть некоторый опыт работы и, следовательно, должны ускорить процесс обучения.

2.5 Виртуальные серверы (наборы разработчиков)

Open Server. Open Server — это портативная серверная платформа и программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий.

Программный комплекс имеет богатый набор серверного программного обеспечения, удобный, многофункциональный продуманный интерфейс, обладает мощными возможностями по администрированию и настройке компонентов. Платформа широко используется с целью разработки, отладки и тестирования веб-проектов, а также для предоставления веб-сервисов в локальных сетях.

Хотя изначально программные продукты, входящие в состав комплекса, не разрабатывались специально для работы друг с другом, такая связка стала весьма популярной среди пользователей Windows, в первую очередь из-за того, что они получали бесплатный комплекс программ с надёжностью на уровне Linux серверов.

Удобство и простота управления безусловно не оставят вас равнодушными, за время своего существования Open Server зарекомендовал себя как первоклассный и надёжный инструмент необходимый каждому веб-мастеру.

Компоненты программного комплекса:

- Open Server 5.2.2;
- Apache 2.2.29;
- Apache 2.4.10;
- Bind 9.10.1;
- PostgreSQL 9.3.5.1;
- Nginx 1.7.8;
- MySQL 5.1.73;

Почти все рассмотренные технологии разработки веб-приложений функционально практически идентичны, т.е. все, что можно реализовать, используя одну из них, можно сделать и на любой другой. Основные различия состоят в скорости и общей стоимости разработки, масштабируемости полученного решения и удобства средств разработки. Однако, в каждом конкретном случае, можно выделить ряд существенных критериев, по которым одна из технологий выигрывает на фоне других, а значит более подходит для решения конкретно поставленной задачи и реализации разрабатываемого проекта. Результат такого сравнения приведен в таблице 1.2

Таблица 1.2

Сравнение технологий разработки веб-приложений

Критерий	Интерпретируемый серверный язык PHP	ASP.NET	Java EE
Кроссплатформенность	+	-	+

Бесплатность	+	-	+
Поддержка ООП	+	+	+

Таблица 1.2. Продолжение

Наличие большого числа библиотек	+	+	+
Простота освоения	+	-	+

Как видно из таблицы сравнения, технология ASP.NET не подходит для решения поставленной задачи, т.к. требует дополнительных расходов на приобретение лицензий. Технология Java EE в целом удовлетворяет выдвинутым требованиям, однако, имеет достаточно большой порог входа и требует больше времени на освоение. Интерпретируемый серверный язык PHP подходит для реализации поставленной задачи по всем параметрам.

В результате для разработки веб-конфигуратора выбран язык JAVA благодаря своей распространенности и широким функциональным возможностям, СУБД PostgreSQL в результате сравнения с другими СУБД по нескольким показателям, а также набор разработчика Open server как один из наиболее простых в использовании и установке локальных веб-серверов.

3.ПРОЕКТИРОВАНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ КОНФИГУРАТОРА

3.1. Разработка базы данных конфигулятора

Основное назначение подсистемы информационного обеспечения состоит в своевременном формировании и выдаче достоверной информации для принятия управленческих решений.

Информационное обеспечение — включает совокупность единой системы показателей, потоков информации, вариантов организации документооборота, систем классификации и кодирования экономической информации, унифицированную систему документации и различные информационные массивы (файлы), хранящиеся в машине и на машинных носителях и имеющие различную степень организации [3.2].

В состав информационного обеспечения включаются два комплекса: компоненты вне машинного информационного обеспечения (классификаторы технико-экономической информации и документы) и внутри машинного информационного обеспечения (макеты, экранные формы для ввода первичных данных в ЭВМ или вывода результатной информации, структура информационной базы: входных, выходных файлов, базы данных).

Центральным компонентом информационного обеспечения является база данных, через которую осуществляется обмен данными различных задач. База данных обеспечивает интегрированное использование разных информационных объектов в функционировании подсистем.

Схемы информационных потоков отражают маршруты движения информации, её объёмы, места возникновения первичной информации и

использования результатной информации. За счёт анализа структуры подобных схем можно выработать меры по совершенствованию всей системы управления.

Построение схем информационных потоков, позволяющих выявить объёмы информации и провести её детальный анализ, обеспечивает:

- исключение дублирующей и неиспользуемой информации;
- классификация и рациональное представление информации.

При этом подробно должны рассматриваться вопросы взаимосвязи движения информации по уровням управления. К каждому исполнителю должна поступать только та информация, которая используется.

Центральным компонентом информационного обеспечения является база данных, через которую осуществляется обмен данными различных задач. База данных обеспечивает интегрированное использование различных информационных объектов в функциональных подсистемах.

Информационная модель — модель объекта, представленная в виде информации, описывающей существенные для данного рассмотрения параметры и переменные величины объекта, связи между ними, входы и выходы объекта и позволяющая путём подачи на модель информации об изменениях входных величин моделировать возможные состояния объекта [3.3].

Для проектирования баз данных, информационных систем, архитектур компьютерных приложений, и других систем чаще всего используются ER-модели.

Логическая модель базы данных была спроектирована с использованием средства визуального проектирования баз данных Erwin. Информационная модель системы приведена на Рис.(3.1).

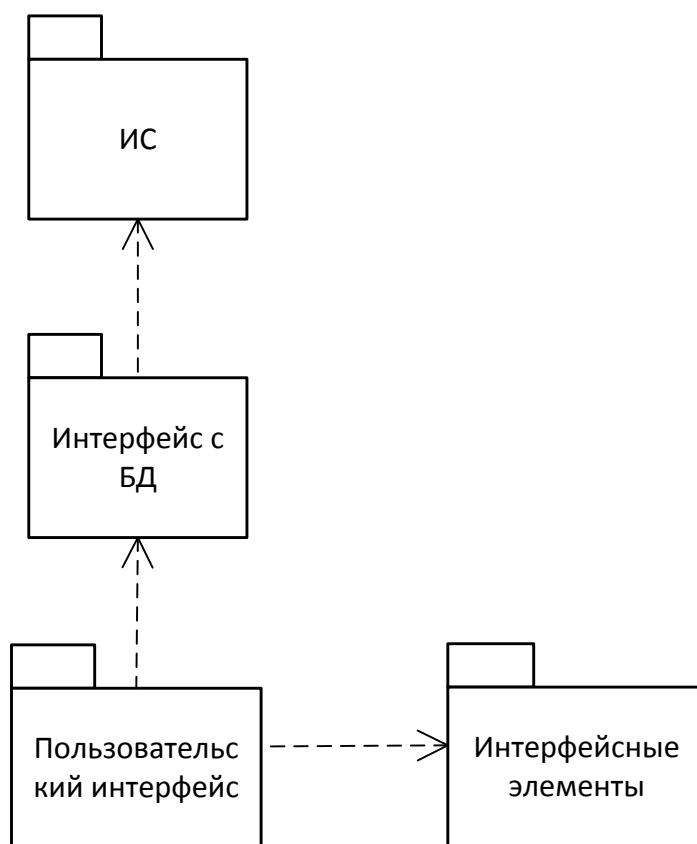


Рис 3.1. Информационная модель системы

Для своего функционирования конфигуратор должен иметь:

- 1) сервер БД системы. На сервере запущена БД, в которой хранится информация о сайте. Сайт работает с информацией, взятой с БД;
- 2) сервер с веб-сайтом. На сервере запущен Apache Tomcat веб-контейнер, он настроен на работу с spring java, и оно активизировано расширение для работы с PostgreSQL;

На основании информационной модели была построена логический модель базы данных с учетом особенностей выбранной СУБД (Рис 3.2.).

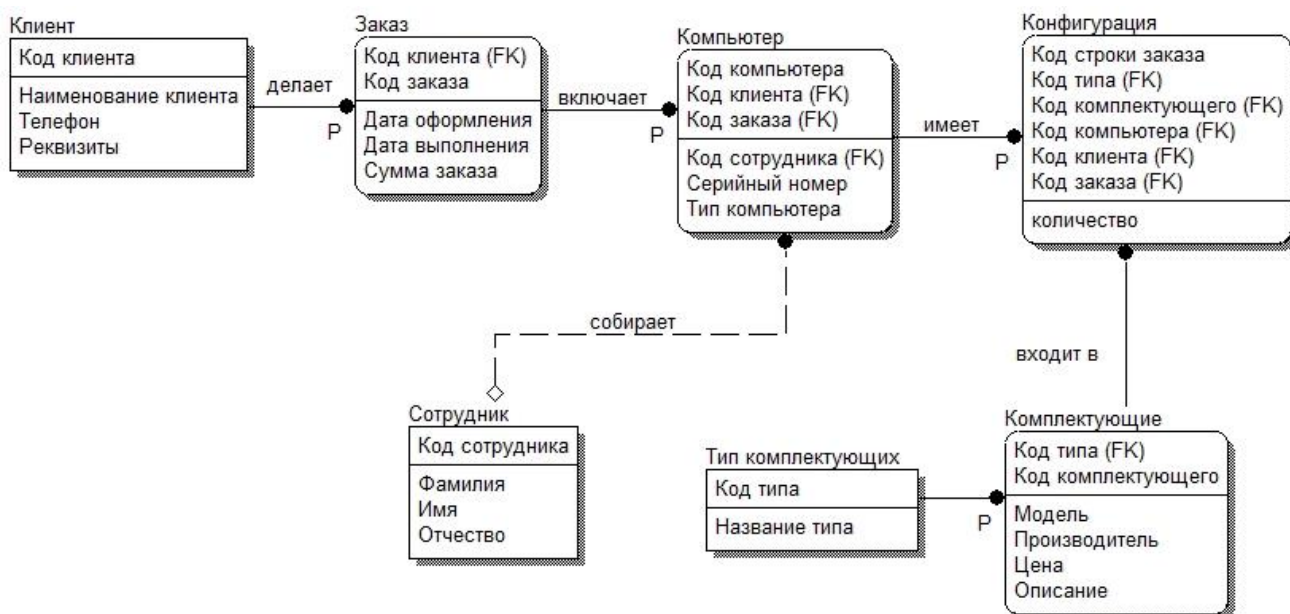


Рис 3.2.Схема базы данных

3.2. Выбор базы данных PostgreSQL

PostgreSQL - это объектно-реляционная система управления базами данных (ORDBMS) (по-русски ОРСУБД или просто СУБД), которая была разработана в Научном Компьютерном Департаменте Беркли Калифорнийского Университета. POSTGRES является пионером во многих аспектах, которые стали доступны в некоторых коммерческих СУБД много позже.

PostgreSQL является открытым исходным кодом, не зависит от платформы системы управления базами данных похожа на MySQL, но менее широко используется. Он предназначен для больших баз данных и включает в себя расширенные возможности по сравнению MySQL, такие как использование триггеров [13]. Недостатком POSTGRES является то, что это происходит медленнее, чем MySQL для выполнения запросов PostgreSQL базируется на языке SQL и поддерживает многие из возможностей стандарта, как показано в таблице 3.1.

Таблица 2.2

Ограничения СУБД PostgreSQL

Максимальный размер БД	Неограничен
Максимальный размер таблицы	32 ТВ
Максимальный размер записи (строки)	1.6 ТВ
Максимальное количество записей	не ограничено
Максимальное количество полей	250 - 1600 в зависима от типа данных
Максимальное индексов на таблицу	не ограничено

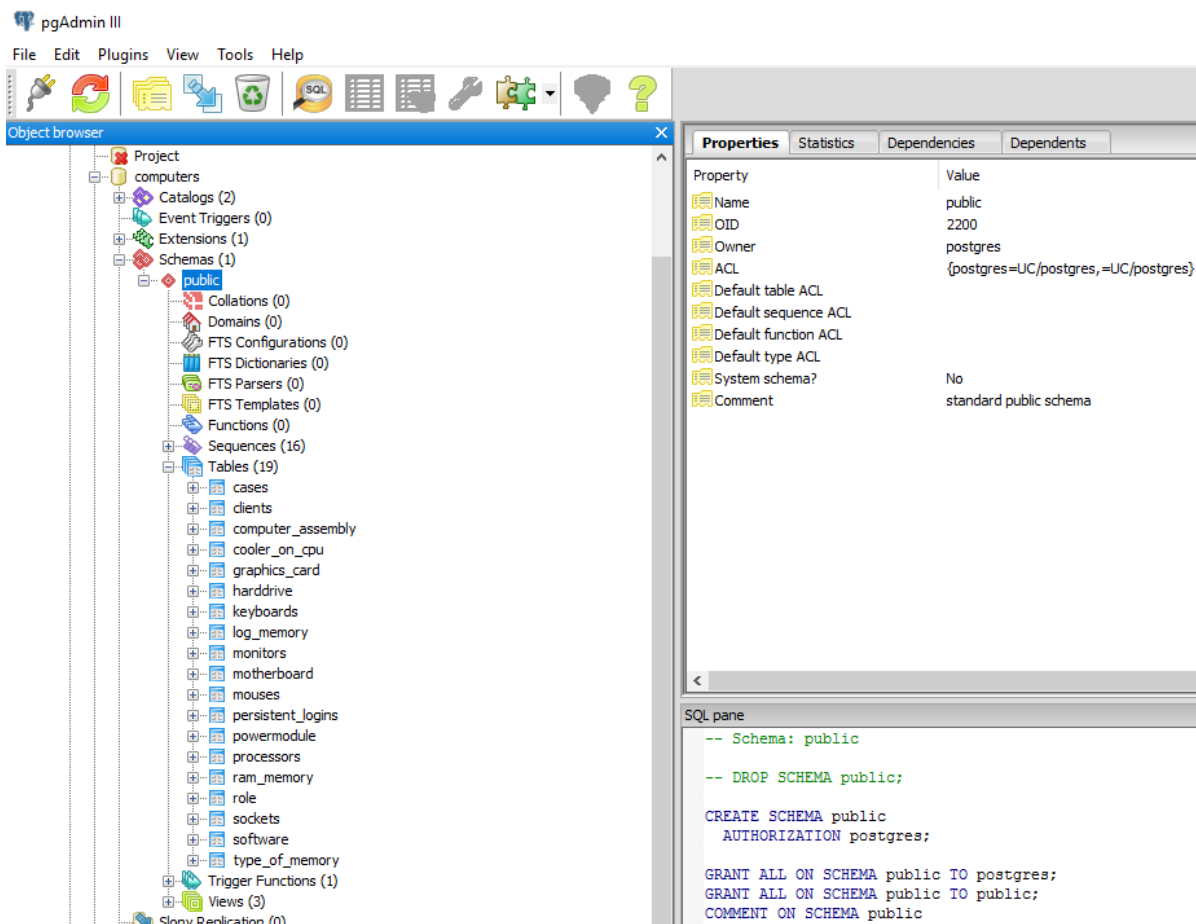


Рис 3.3. данных в PostgreSQL

Edit Data - localhost (localhost:5432) - computers - public.motherboard

File Edit View Tools Help

No limit

	id [PK] integer	name character varying(150)	id_socket integer	id_typedmemory integer	number_of_memory_slots character varying(100)	numbers_usb integer	power character varying(100)	formfactor character varying(100)
1	1	MSI H110M PRO-VD	14	1	2	3	20pin	microATX
2	2	ASUS Z170 PRO GAMING	7	3	4	5	24pin	ATX
3	3	MSI 970A-G43	7	1	4	2	24pin	ATX
4	4	MSI H110M PRO-D	1	2	2	5	20pin	microATX
5	5	ASUS Z170-A	13	1	4	4	24pin	ATX
6	6	ASUS Z170-P	8	1	2	5	20pin	microATX
7	7	ASUS B85M-G	3	2	4	4	24pin	ATX
8	8	ASUS M5A78L-M LX3	9	3	4	3	24pin	ATX
9	9	ASUS H110M-R/C/SI	2	1	4	5	24pin	ATX
10	10	ASRock H110M-DGS	8	3	4	5	24pin	ATX
11	11	MSI Z170A GAMING M3	14	2	4	3	24pin	ATX
12	12	ASUS Z170-K	13	2	2	2	20pin	microATX
13	13	ASUS H110M-K	10	2	4	4	24pin	ATX
14	14	ASUS MAXIMUS VIII HERO	9	1	2	5	20pin	microATX
15	15	ASUS MAXIMUS VIII RANGER	11	2	4	4	24pin	ATX
16	16	MSI Z170A GAMING M5	8	2	4	2	24pin	ATX
17	17	ASUS ROG STRIX X99 GAMING	10	2	4	5	24pin	ATX
18	18	MSI B150M BAZOOKA	14	2	4	3	24pin	ATX
19	19	ASUS M5A97 R2.0	11	1	4	5	24pin	ATX
20	20	MSI 970 GAMING	1	3	4	2	24pin	ATX
21	21	ASUS H81M-K	7	2	2	4	20pin	microATX
22	22	MSI H81M-P33	13	2	4	5	24pin	ATX

Рис 3.4. Пример таблица с данными в PostgreSQL

Хотя Access предоставляет полезные графические интерфейсы для нетехнических пользователей, есть надежда, веб-приложение может быть получено, что не требует от пользователя изменить базу данных непосредственно в любом случае. Конструкция должна обеспечить базу данных изменяется только через веб-интерфейс.

MS SQL Sever будет полезным СУБД для этого решения, потому что это хорошо при обработке нескольких запросов к базе данных от многократного использования, которые могут только повысить производительность приложения. Основным недостатком MS SQL является его стоимость и из-за небольшого финансового бюджета Владелец работе сети, это маловероятно решение.

PostgreSQL и MySQL являются наиболее жизнеспособные решения, поскольку они являются свободно доступными и не зависит от платформы

3.3. Проектирование иерархии диаграмм потоков данных

Диаграммы потоков данных (Data flow diagramming, DFD) используются для описания документооборота и обработки информации. DFD представляет модельную систему как сеть связанных между собой работ. DFD описывает:

- функции обработки информации (работы);
- документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке информации;
- внешние ссылки (external references), которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;
- таблицы для хранения документов (хранилище данных, data store).

Далее рассмотрим виды управления IDEF0 диаграмма предприятия на Рисунок 3.4



Рис 3.4 Работа в IDEF0

Описание работа в IDEF0 для предприятия деятельность для конфигуратора.

- Вход – это потребляемая или изменяемая работой информация или материал
- Выход – информация или материал, которые производятся работой
- Управление – процедуры, правила, стратегии или стандарты, которыми руководствуется работа
- Механизмы – ресурсы, которые выполняют работу (например, сотрудники, оборудование, устройства и т.д.)

Для рассматриваемого предприятия входными стрелками будут:

- 1) Заказы клиентов - список компьютеров и их конфигурация, которые клиент желает приобрести
- 2) Комплектующие от поставщиков - комплектующие, полученные от поставщиков, из которых собираются компьютеры и ноутбуки

Выходные стрелки:

- 1) Готовая продукция - собранные компьютеры и ноутбуки
- 2) Заказы поставщикам - список комплектующих, которые предприятие закупает у поставщиков
- 3) Оплата за комплектующие - деньги поставщикам за комплектующие
- 4) Маркетинговые материалы - прайс-листы и рекламой.

Стрелки управления:

- 1) Законодательство - различные законодательные документы, которыми руководствуется предприятие в процессе своей деятельности
- 2) Правила и процедуры - различные правила и процедуры, которыми руководствуется предприятие в процессе своей деятельности (например, правила сборки и тестирования компьютеров, процедура общения с клиентами и т.п.)
- 3) Стрелки механизмов:

4) Бухгалтерская система

5) Персонал

Итоговая контекстная диаграмма имеет вид (рис.3.5):

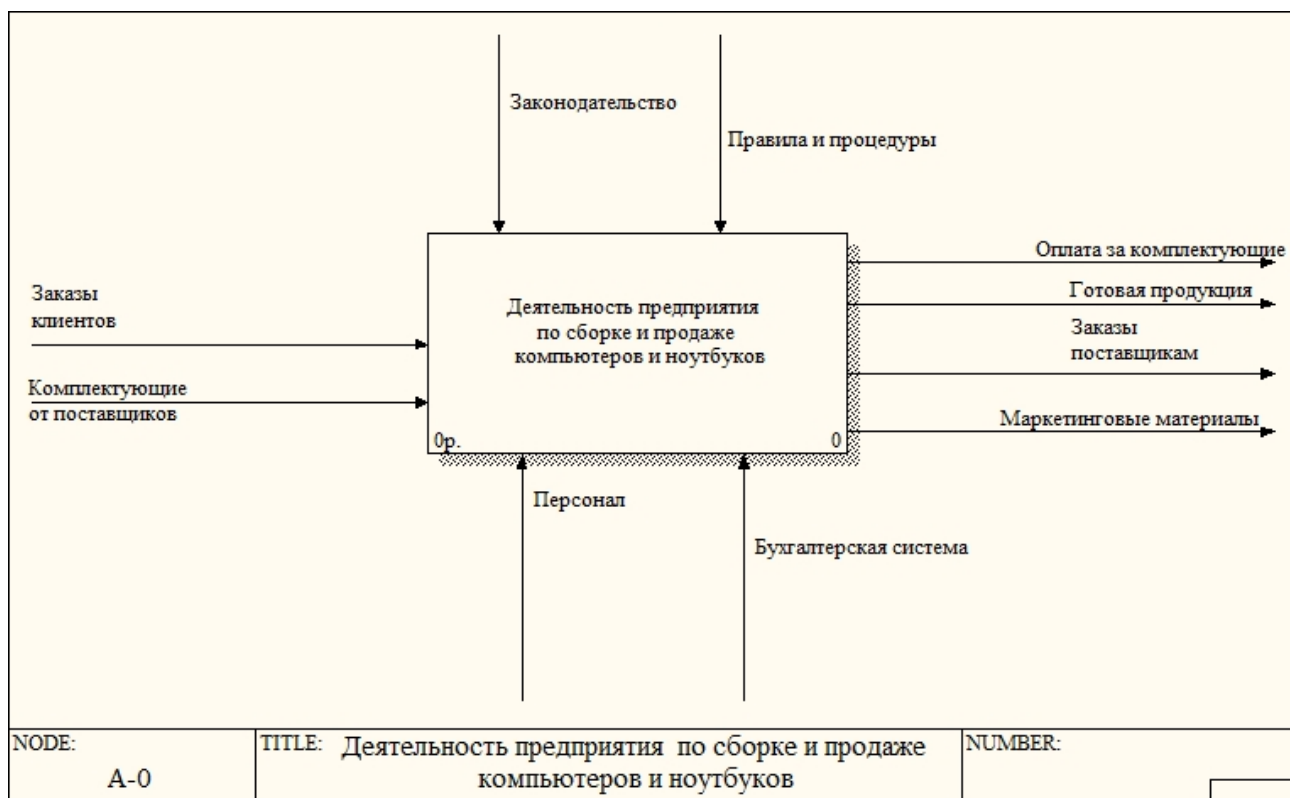


Рис 3.5 Итоговая Контекстная диаграмма

На данном этапе построения диаграммы выяснилось, что мы не учли такой важный фактор, как деньги, которые клиенты дают за готовую продукцию. Деньги клиентов - это вход работы " по сборке и продаже компьютеров.

После соединения граничных стрелок с работами следующим шагом соединим работы между собой. Поскольку работа "Управление" включает в себя общее управление предприятием, то одним из ее результатов будет являться "Управляющая информация", поступающая на вход управления всех остальных работ.

Работа "Продажи и маркетинг" получает на входе заказы клиентов (т.е. количество компьютеров и их конфигурация), информацию о которых она передает работе "Сборка и тестирование компьютеров" в качестве управляющей информации.

Работе "Сборка и тестирование" для своего функционирования необходимы комплектующие, которые она заказывает у работы "Отгрузка и снабжение" (выходная стрелка "Список необходимых комплектующих"). Собранные компьютеры она также передает работе "Отгрузка и снабжение" (выходная стрелка "Собранные компьютеры"). Информация о результатах сборки и тестирования необходима работе "Продажи и маркетинг" (выходная стрелка "Результаты сборки и тестирования").

Результат всех перечисленных действий показан на Рис 3.6.

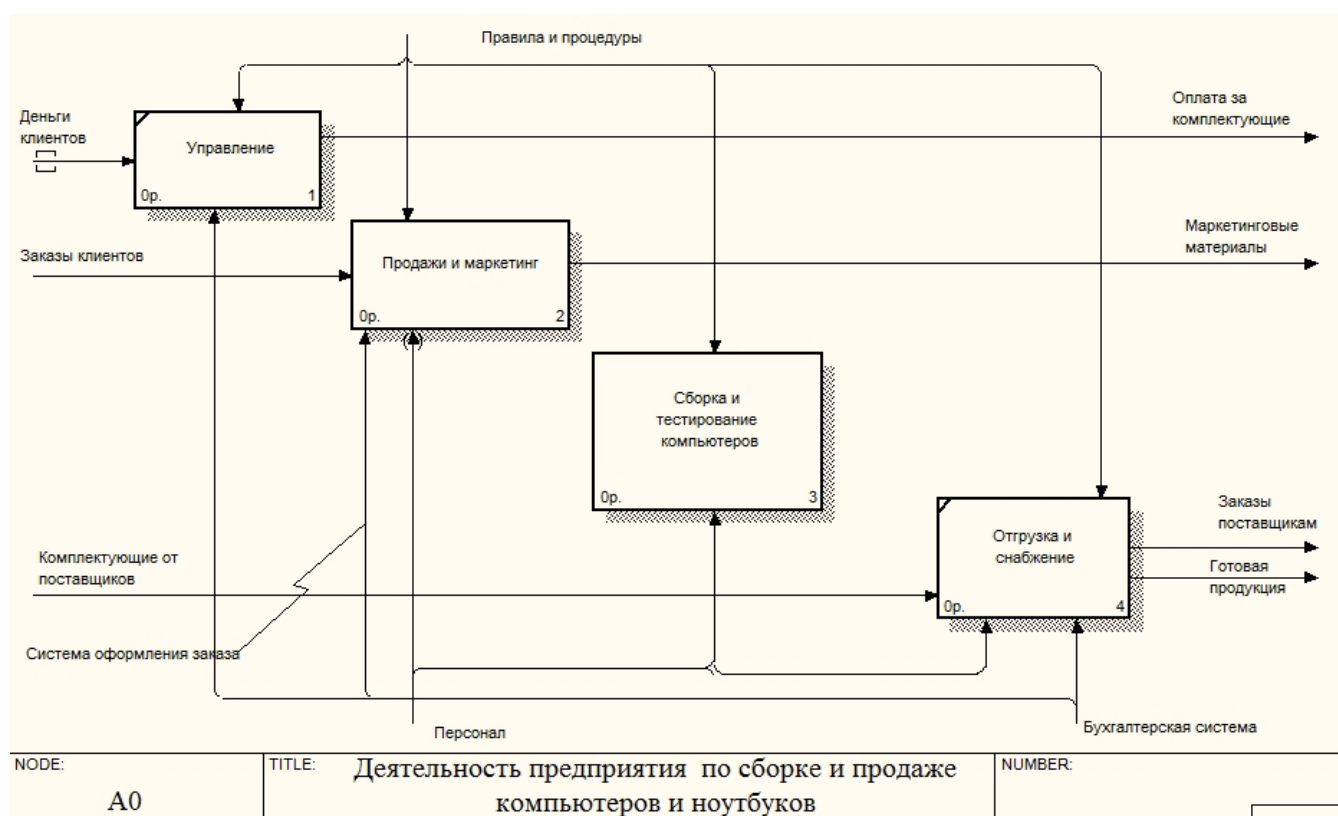


Рис 3.6. Результат всех действий контекстных диаграмм

3.4.Разработка Веб -интерфейса конфигуратора

С ростом интернета широкое распространение получили веб- интерфейсы, позволяющие взаимодействовать с различными программами через браузер (например, управление своим заказом в интернет-магазине). [3]

Веб-интерфейсы удобны тем, что дают возможность вести совместную работу сотрудникам, не находящимся в одном офисе (например, веб-интерфейсы часто используются для заполнения различных баз данных или публикации материалов в интернет – СМИ). [3]

Интерфейс - в широком смысле - определенная стандартами граница между взаимодействующими независимыми объектами. Интерфейс задает параметры, процедуры и характеристики взаимодействия объектов.

Интерфейс пользователя - элементы и компоненты программы, которые способны оказывать влияние на взаимодействие пользователя с программным обеспечением. В том числе:

- средства отображения информации, отображаемая информация, форматы и коды;
- командные режимы, язык пользователь-интерфейс;
- устройства и технологии ввода данных;
- диалоги, взаимодействие и транзакции между пользователем и компьютером;
- обратная связь с пользователем;
- поддержка принятия решений в конкретной предметной области;
- порядок использования программы и документация на нее.
- Есть несколько простых правил, позволяющих интерфейсу магазина быть понятным клиенту.

Окончательный интерфейс можно посмотреть на Рис 3.5. и 3.6

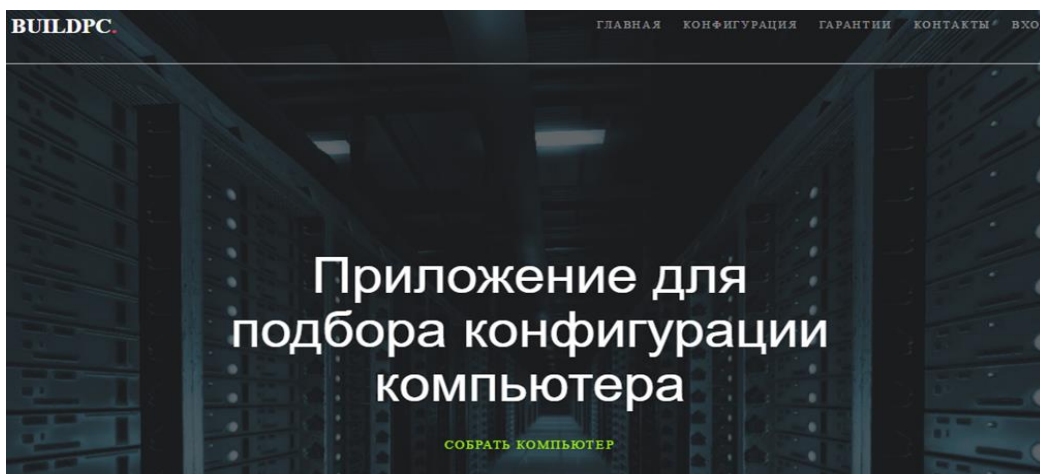


Рис 3.5. Главный интерфейс Web-приложения.

На главной странице приложения можно заказать сборки на нужный вам компьютер: в правом верхнем углу находится навигация по всему сайту.

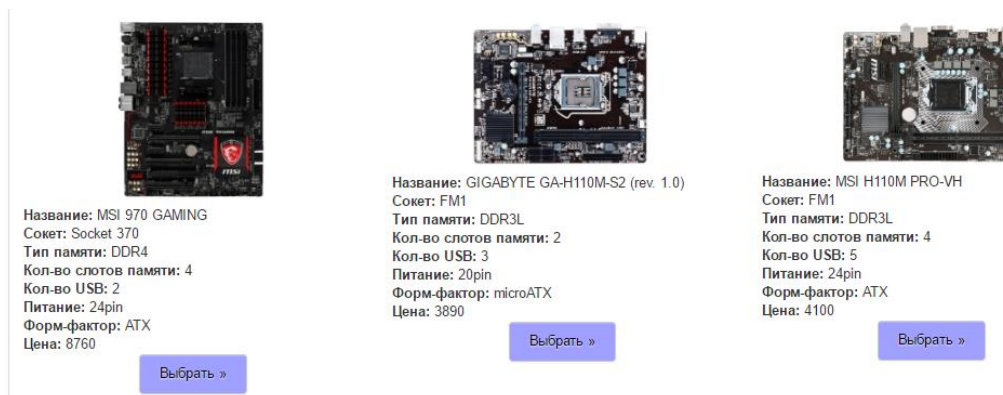


Рис 3.6. Выбор материнской платы



Рис 3.7. пример страница конфигуратора с выбора процессора

3.5. Реализация алгоритмы работы конфигуратора

В диаграмме прецедентов 1 актер. На рис. 3.8 представлена диаграмма прецедентов, которая представляет функции пользователя по работе с системой конфигуратора.

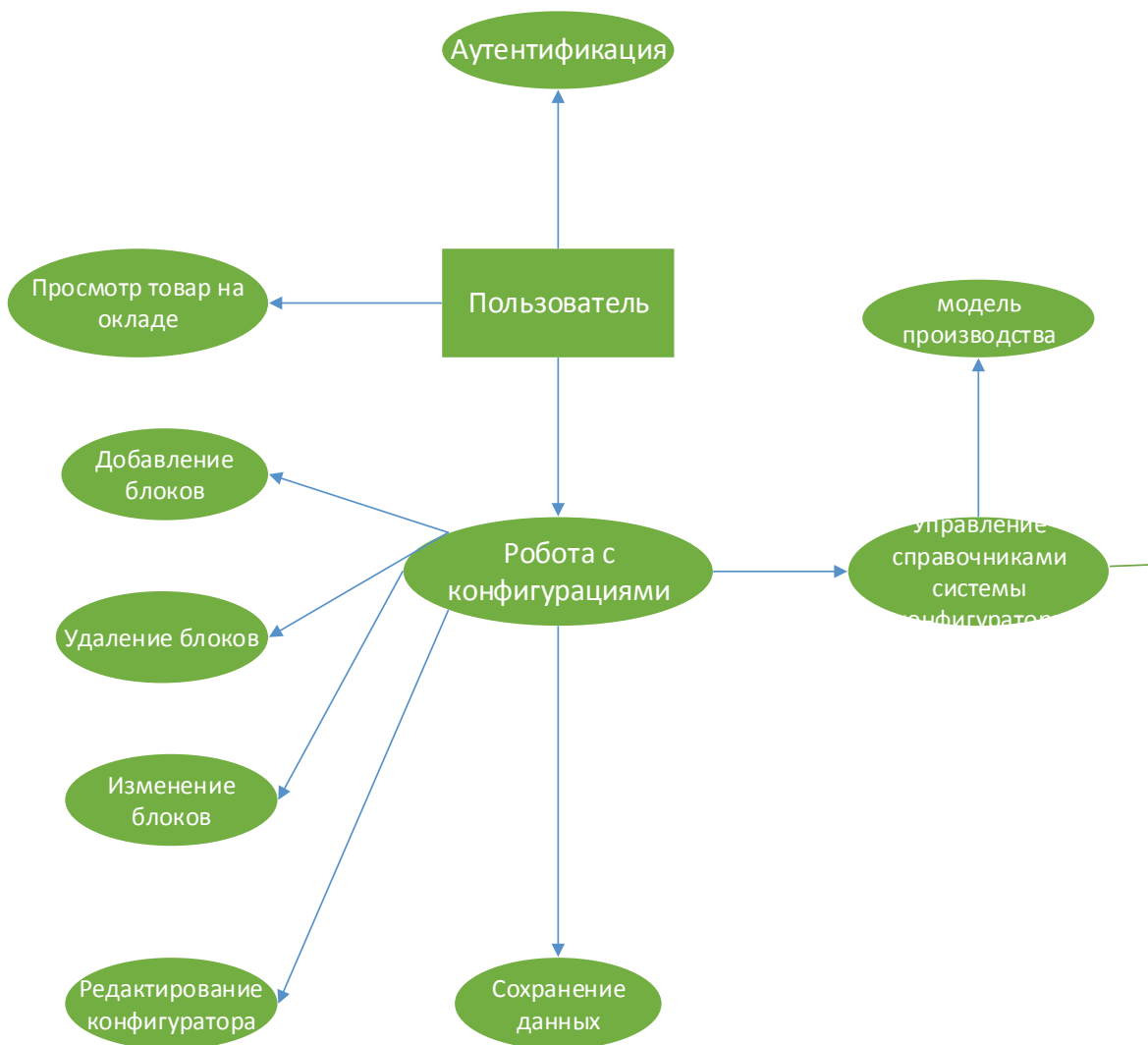


Рис 3.8 Диаграмма прецедентов пользователя системы

Пользователь имеет возможность формирования конфигураций компьютера, а также работы со справочниками. Под работой со справочниками

подразумевается добавление, удаление и редактирование входящей в их состав информации.

Прецедент «Аутентификация пользователя» активизируется всеми пользователем ИС. Прежде чем пользователю начать работу с БД, система запрашивает его пароль и логин. Если пользователь не зарегистрирован или в пароле и/или логине допустил ошибку, то он не получает доступа к работе в ИС. После успешной проверки логина и пароля пользователя открывается главное

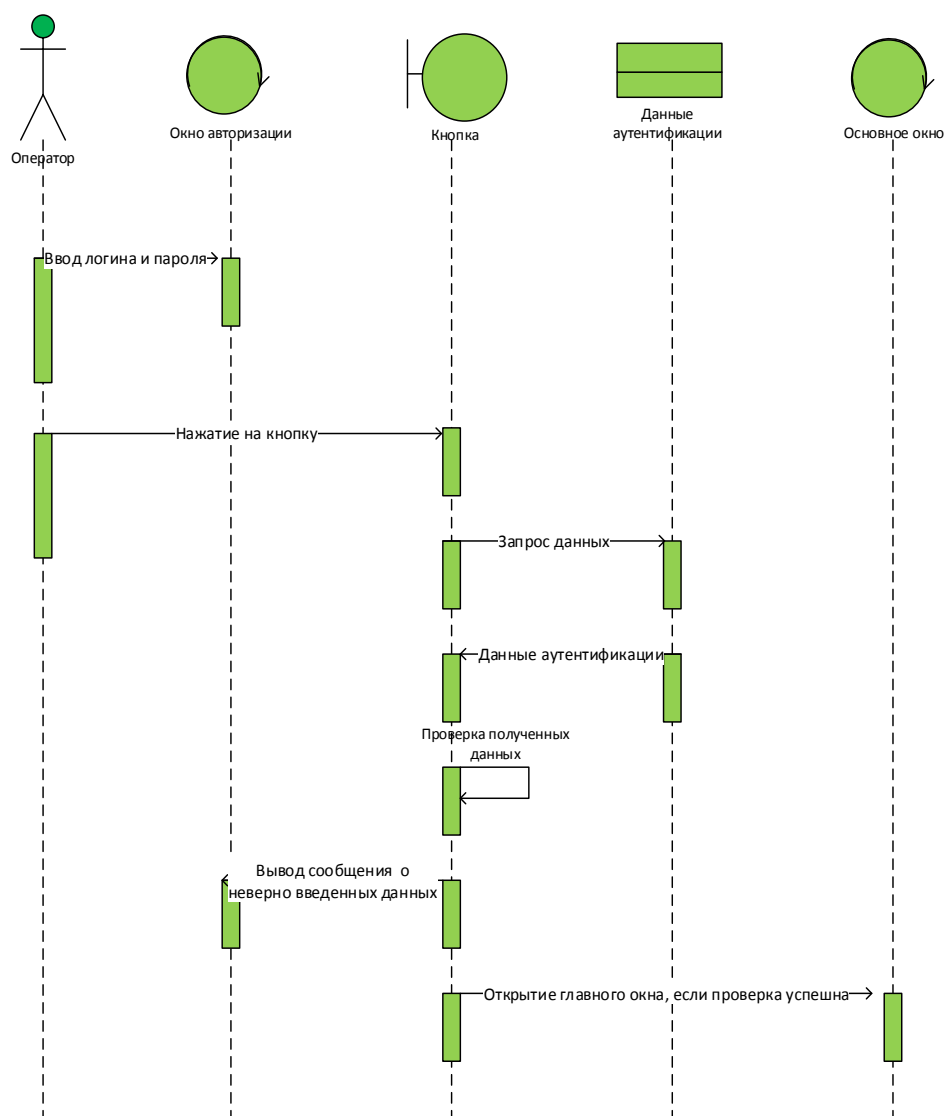


Рис 3.9. Диаграмма последовательности «Аутентификация пользователя»

Прецедент «Учет конфигураций» активизируется субъектом ИС пользователь. Данный прецедент описывает процесс ввода новых данных в БД. Данные вводятся в формы ввода на основании документов, а также всей справочная информация, которая нужна в процессе работы с системой. На рис (3.10) представлены диаграмма последовательности данного прецедента.

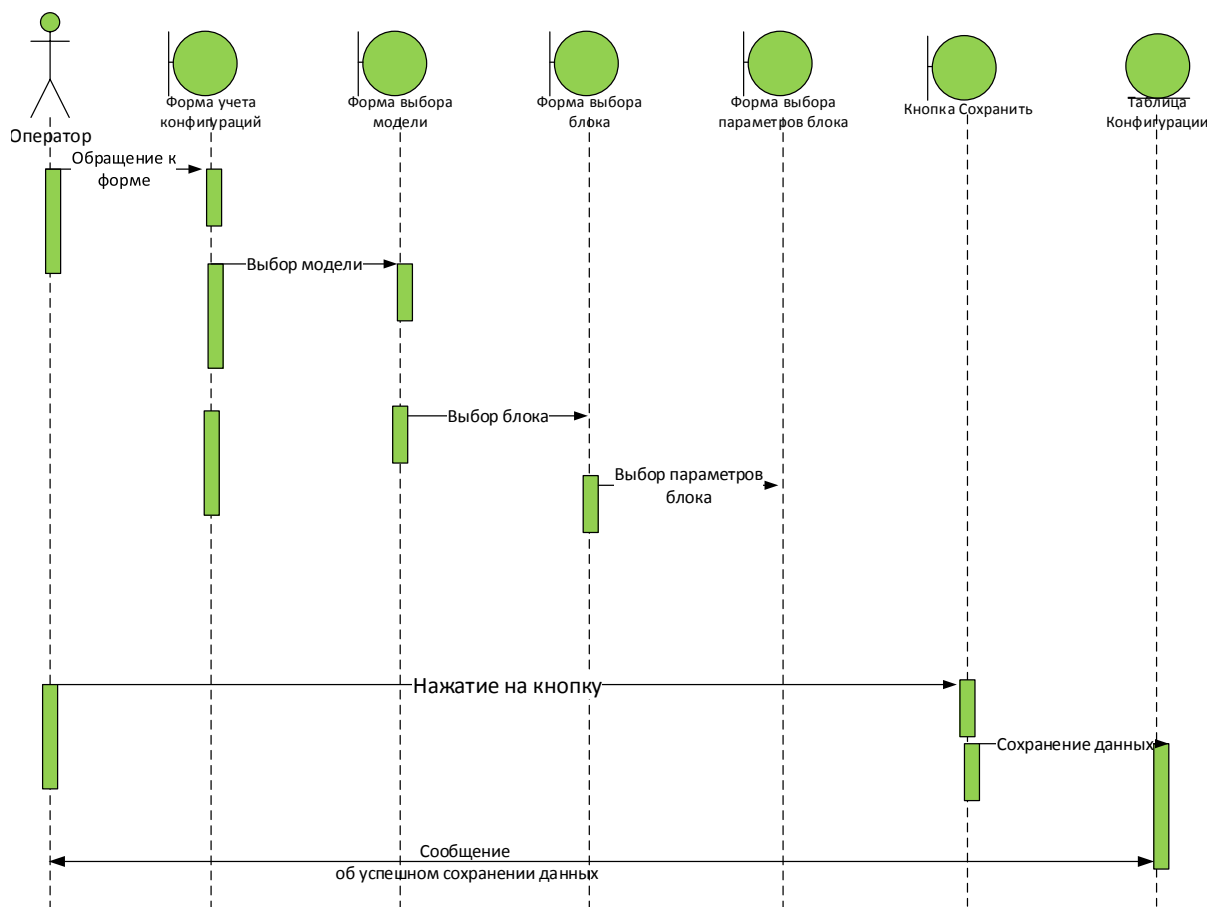


Рис 3.10 Диаграмма последовательности «Учет конфигураций»

Рассмотрим порядок работы со справочниками системы на примере учета информации по моделям конфигуратора. Диаграмма последовательности «Учет моделей» показывает порядок учета товаров пользователем ИС. Алгоритм работы конфигуратора на рис 3.10

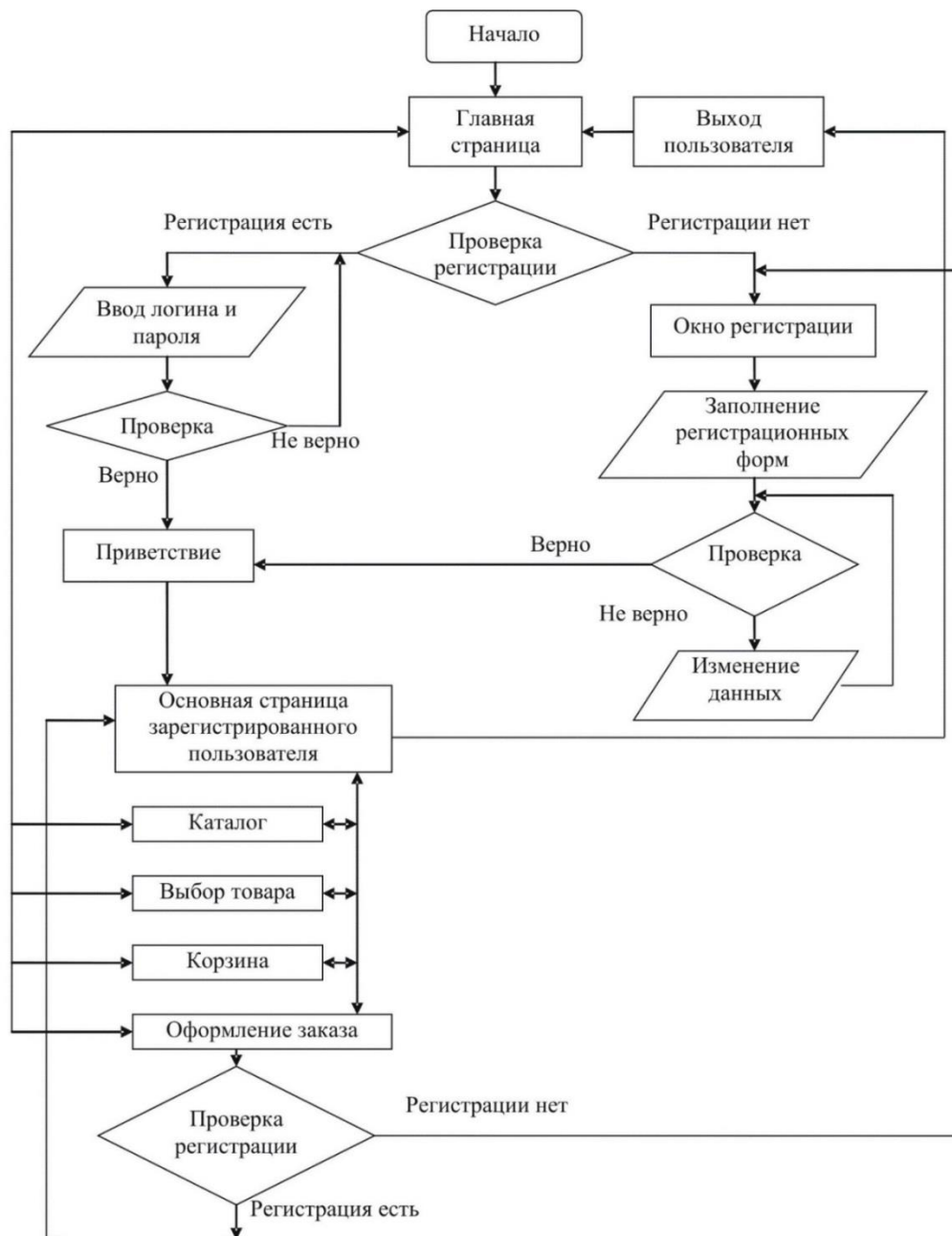


Рис. 3.11. Алгоритм работы конфигуратора

Диаграмма пакетов показывает, из каких частей состоит проектируемая система и как эти части связаны друг с другом. Пакет – совокупность описаний классов и других программных ресурсов, в том числе и самих пакетов.

Анализ концептуальной модели позволил выделить следующие пакеты:

- 1) интерфейсные элементы - классы, реализующие интерфейсные компоненты;
- 2) пользовательский интерфейс - классы, реализующие объекты интерфейса с пользователем;
- 3) интерфейс с БД - классы, реализующие интерфейс с базой данных;
- 4) База данных.

Диаграмма пакетов приведена на рис 3.12.

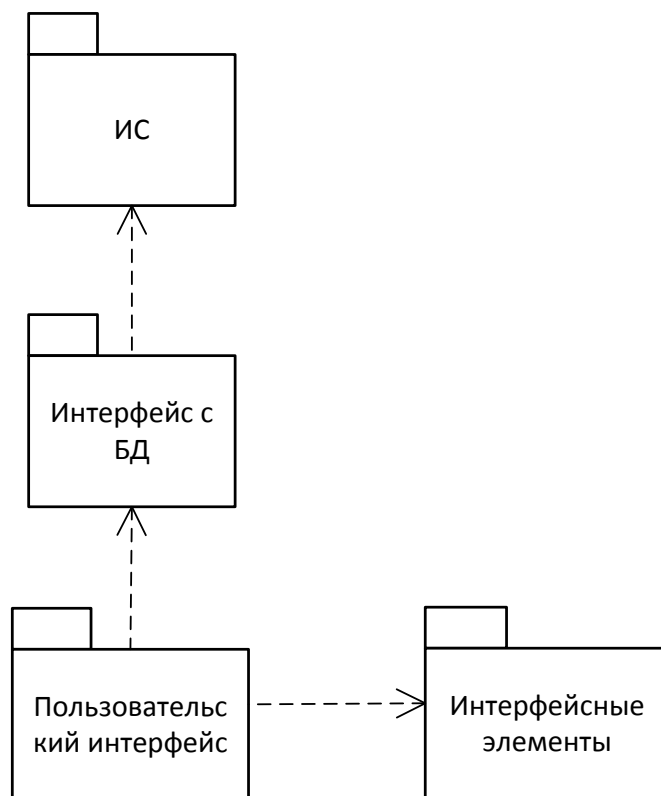


Рис 3.12. Диаграмма пакетов

Для своего функционирования конфигуратор должен иметь:

- сервер БД системы. На сервере запущена БД, в которой хранится информация о сайте. Сайт работает с информацией, взятой с БД;
- сервер с веб-сайтом. На сервере запущен Apache Tomcat или другой веб-контейнер, он настроен на работу с JAVA. У Java должно быть активизировано расширение для работы с PostgreSQL;

- ПК администратора. На этом компьютере запущен веб-браузер с панелью управления сайтом. Через нее происходит работа с сайтом. Опционально ПК может использоваться, как и рабочая станция пользователя;

Диаграмма развертывания представлена на рис 3.13.

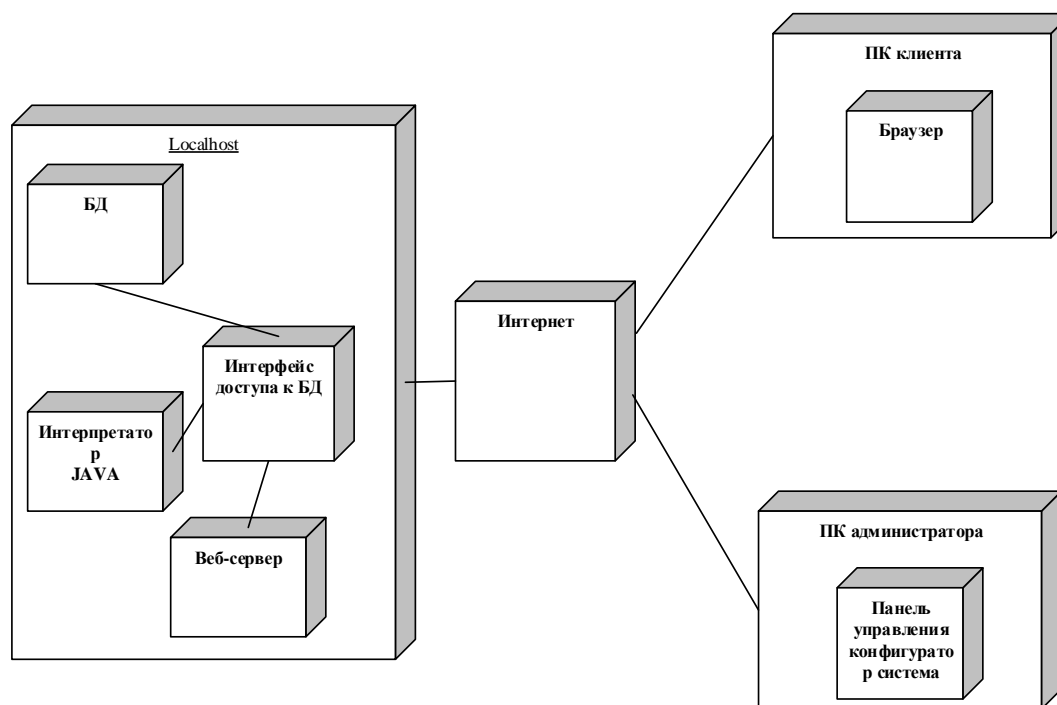


Рис 3.13. Диаграмма развертывания

3.6. Реализация структуры программы конфигулятора

Для разработки Web-приложения использовался язык java на и JavaScript. java – язык программирования общего назначения, активно использующийся для создания сайтов. Является одним из наиболее популярных скриптовых языков в мире, что обеспечено его простой, обширной функциональностью и скоростью исполнения [5] Как было установлено ранее, система должна состоять из базы данных и трех приложений, а именно: клиентского приложения, сервера приложений и панели администратора.

Клиентское приложение — это процесс- «демон», который запускается со старта операционной системы или ручным способом и постоянно исполняется в фоновом режиме. Механизм работы клиентского приложения заключается в постоянной отправке специальных пакетов серверу приложений через заданное время, а также в получении запроса от сервера приложений на отправку списка запущенных процессов на данном компьютере серверу приложений и в обработке приказа от сервера приложений о выключении приложения.

Эта функция программное реализована с помощью создания отдельного потока, в котором постоянно исполняется цикл. В цикле находится ожидание отведенного времени, по прошествии которого происходит установка TCP соединения с сервером и передача ему текущего IP-адреса и имени компьютера клиентского приложения. Ниже, на рис. (3.14) и рис (3.15) приведен участок кода, на клиентскую часть приложения.

```
<li><a href="/">Главная</a></li>
  <li><a href="/configuration">Конфигурация</a></li>
  <li><a href="/warranty">Гарантия</a></li>
  <li th:if="${#httpServletRequest.remoteUser == 'manager'}"><a
href="/panel">Панель</a></li>
  <li th:if="${#httpServletRequest.remoteUser == 'admin'}"><a
href="/admin">Панель</a></li>
  <li><a href="/contact">Контакты</a></li>
  <li th:if="${#httpServletRequest.remoteUser == null}"><a
```

Рис. 3.14 Исходный код реализации в клиентском приложении

```
href="/login">Вход</a></li>
|
  <li th:if="${#httpServletRequest.remoteUser != null}" class="dropdown">
    <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
<span th:if="${#httpServletRequest.remoteUser != null}">
  <span th:text="${#httpServletRequest.remoteUser}">
```

Рис 3.15. Исходный код реализации в клиентском приложении

Сервер приложений. Работу сервера приложений можно описать следующими словами: постоянное получение информации от клиентских

приложений, ее обработка и запись в базу данных, а также перенаправление операций, получаемых из панели администратора к клиентским приложениям и обратно. В число таких операций входят выключение компьютера клиентского приложения и получение его списка процессов.

Сервер приложений также является процессом- «демоном», который запускается либо при старте операционной системы, либо ручным способом. При запуске сервер приложений автоматически устанавливает статусы компьютеров клиентских приложений в базе данных в отключенное положение. Сервер приложений состоит из четырех потоков, каждый из которых работает независимо.

Первый поток постоянно получает данные от клиентских приложений. Реализован данный механизм следующим образом. Клиентское приложение

Устанавливает соединение и передает свой пакет серверу приложений. Сервер приложений получает пакет и обрабатывает, выделяя из него имя компьютера и IP- адрес компьютера с клиентским приложением, после чего записывает полученную информацию в базу данных, устанавливая статус компьютера и добавляя текущее время для дальнейшей проверки последней отправки пакета от клиентского приложения. После этого соединение разрывается и сервер приложений переходит в режим слушателя, в котором он ждет новое соединение на заданном порту. Участок кода, показывающий реализацию логики данного потока, расположен на листинг 3.1.

Листинг 3.1.Сервер приложений

```
Session session = sessionFactory().openSession();
    session.beginTransaction();
    List<ComputerAssembly> qwe = computerAssemblyService.getAll();
    int id = qwe.size() + 1;
```

```

String sql = "INSERT INTO computer_assembly (\`id\`, \`id_case\`,
\`id_cooler\`, \`id_graphicscard\`, \`id_harddrive\`, \`id_keyboard\`, \`id_monitors\`,
\`id_motherboard\`, \`id_mouse\`, \`id_powermodule\`, \`id_processor\`, \`id_ram\`,
\`id_software\`, \`totalprice\`, \`username\`, \`in_work\`)n" +
        "VALUES(" + id + ", " + cases + ", " + cooler + ", " + graphics + ", " +
harddrive + ", " + keyboard + ", " + monitor + ", " + motherboard + ", " + mouse+ ", "
+ power + ", " + processor + ", " + ram + ", " + software + ", " + totalPrice + ", " +
username + "', '0')";

Query query = session.createSQLQuery(sql);
query.executeUpdate();
session.getTransaction().commit();
session.close();
}

@RequestMapping(value = "/addWork", method = RequestMethod.POST)
public void addComputer(
        @RequestParam("id") int idmotherboard    ){

```

панель администратора

Панель администратора является единственным приложением в этой системе управления конфигурацией, которая имеет графический интерфейс. Интерфейс состоит из трех частей. Первая часть - это меню навигации, которое находится в верхней части окна. Он содержит кнопку выхода, а также кнопки просмотра журнала. Вторая часть интерфейса расположена в крайнем правом углу окна приложения и отображает текущие вкладки, что заставляет клиентов и администратора регистрироваться в приложении. На Рис (3.15) показано главное окно интерфейса админ-панели.

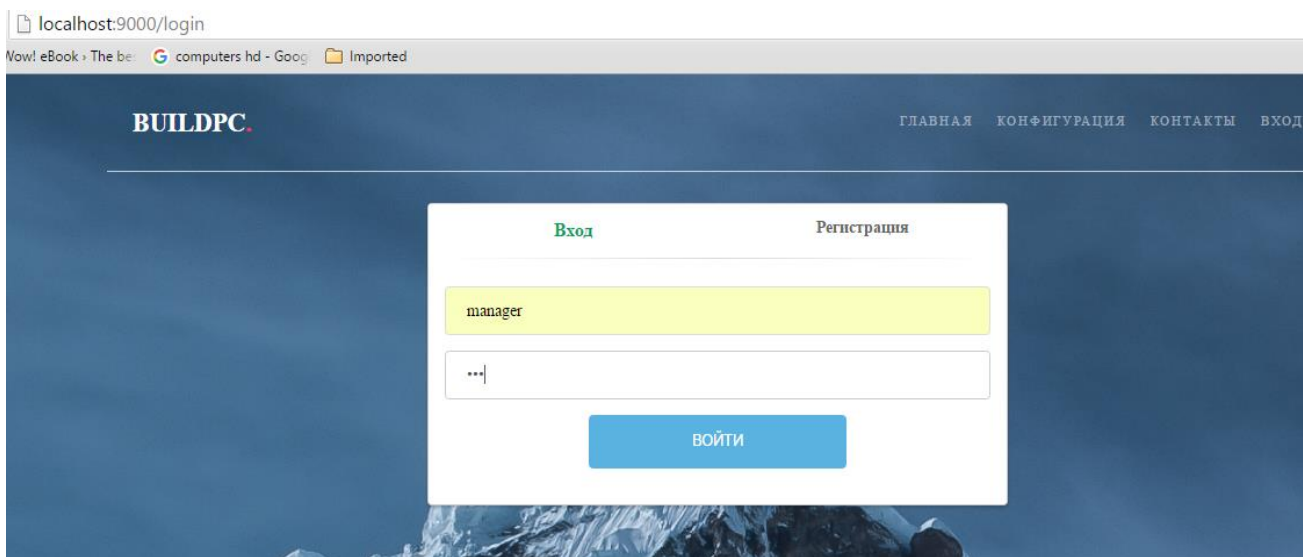


Рис 3.16. Главное окно интерфейса админ-панели.

Для того чтобы собрать нужны сборки необходимо чтобы клиент пройти регистрация для получения доступ к выходу к сборке. На Рис 3.16 показано главное окно регистрация на админ-панели.

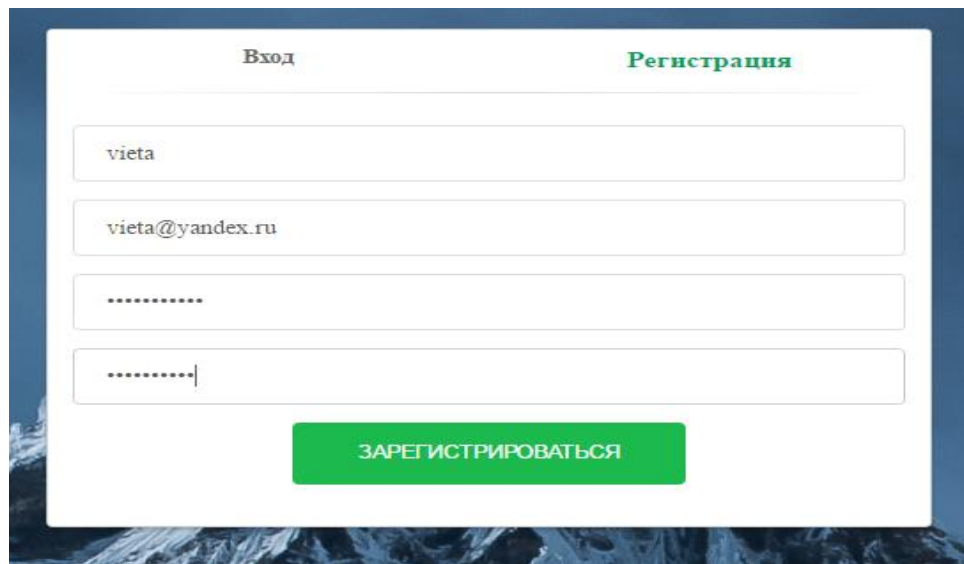


Рис 3.17. Главное окно регистрация на админ-панели.

После в приложения регистрация поток устанавливает соединение с и ждет получения данных. После получения он пересылает их панели администратора и дают успешно сообщения к регистрации. Такая же логика

работы и с получением журналов последних действий. На Листинг (3.2) изображена часть метода, показывающая логику работы третьего потока в исходного кода.

Листинг 3.2 системная регистрация

```
<script>
function register() {
    $.ajax({
        url: '/configuration/register',
        type: 'POST',
        data: {
            'username': $("#username").val(),
            'email': $("#email").val(),
            'password': $("#password").val()
        },
        beforeSend: function(xhr){
            xhr.setRequestHeader($("#meta[name='_csrf_header']").attr("content"),
                $("#meta[name='_csrf']").attr("content"));
        },
        success: function(data) {
            console.log(data);
            console.log('the best');
        },
        error: function (xhr, ajaxOptions, thrownError) {
            console.log(xhr.status + ": " + thrownError);
        }
    });
}
```

```

<div id="gtcc-logo"><a href="index.html">BUILDPC<span>.</span></a></div>
</div>
<div class="col-xs-10 text-right menu-1">
  <ul>
    <li><a href="/">Главная</a></li>
    <li><a href="/configuration">Конфигурация</a></li>
    <li><a href="/warranty">Гарантии</a></li>
    <li th:if="${httpServletRequest.remoteUser == 'manager'}"><a href="/panel">Панель</a></li>
    <li th:if="${httpServletRequest.remoteUser == 'admin'}"><a href="/admin">Панель</a></li>
    <li><a href="/contact">Контакты</a></li>
    <li th:if="${httpServletRequest.remoteUser == null}"><a href="/login">Вход</a></li>

    <li th:if="${httpServletRequest.remoteUser != null}" class="dropdown">
      <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
        <span th:if="${httpServletRequest.remoteUser != null}">
          <span th:text="${httpServletRequest.remoteUser}">
            sample_user
          </span>
        </span>
        <span class="caret"></span>
      </a>
      <ul class="dropdown-menu" style="...">
        <li>
          <form th:action="@{/logout}" method="post">
            <button style="..." type="submit" class="btn btn-default">Выход</button>
          </form>
        </li>
      </ul>
    </li>
  </ul>
</div>
</div>

```

Рис 3.18. Исходный код из Admin панель

3.7.Безопасность

Безопасность всегда является проблемой, где хранится секретная информация. В этом приложении были рассмотрены три аспекта безопасности: безопасность данных, хранящихся в базе данных; Доступ к средству администрирования онлайн; и доступ к ограниченным областям веб-сайта, предназначенного только для членов. Для обеспечения безопасности данных в базе данных база данных защищена логином и паролем к

Предотвращения несанкционированного доступа к информации, хранящейся в таблицах. Для предотвращения несанкционированного доступа пользователей к защищенным страницам пользователь должен ввести войти деталей, прежде чем страница может отображаться. Тот же метод применяется к объекту администрирования сайта.

4.ТЕСТИРОВАНИЕ СОЗДАННОГО ПО

Тестирование программного обеспечения является исследование, проведенное для предоставления заинтересованным сторонам информацию о качестве продукта или услуги в рамках теста. [1] Тестирование программного обеспечения может также дать объективную, независимое представление программного обеспечения, чтобы позволить бизнесу оценить и понять риски внедрения программного обеспечения. Методы испытаний включают в себя процесс выполнения программы или приложения с целью обнаружения ошибок в программном обеспечении (ошибки или другие дефекты), а также для проверки того, что программный продукт, пригодный для использования.

Тестирование программного обеспечения включает в себя выполнение программного компонента или компонента системы для оценки одного или нескольких свойств, представляющих интерес. В общем, эти свойства указывают на степень, в которой компонент или тестируемая система:

- отвечает требованиям, которыми руководствовались его проектирование и разработку,
- правильно отвечает на все виды входов выполняет свои функции в течение приемлемого времени
- может быть установлена и запущена в ее предполагаемую среду

4.1. Интерфейс Web-приложения

4.1.1. Тестовые данные

После разработки приложения его необходимо протестировать, чтобы у поставщика не было проблем. Проверку приложения будем делать на пару страниц. Главный Пользовательский интерфейс показан на Рис (3.19)

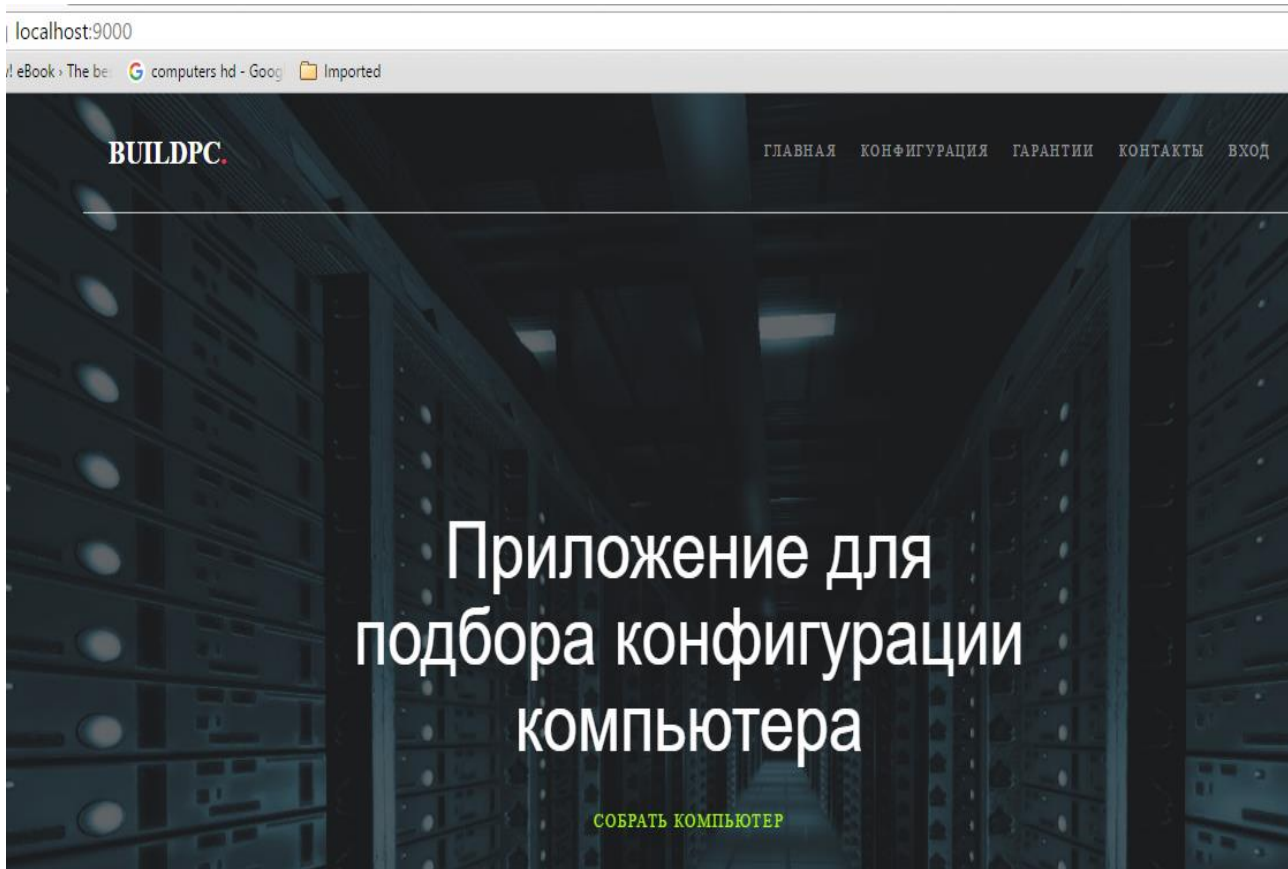


Рис 3.19. Главный Пользовательский интерфейс

На главной странице приложения можно заказать сборки на нужный вам нужный вам компьютер (рис.3.20). В правом верхнем углу находится навигация по всему сайту.



Рис 3.20. Главная страница для Выбора материнскую плату

На рис.(3.21) производится просмотр одной из главной страницы для
Выбора процессор



Рис 3.21. главной странице для выбора процессора

На рис.3.22 производится просмотр одной из главной страницы для
Выбора оперативную память



Рис 3.22. Главная страница для выбора оперативной памяти

На рис.3.23 производится просмотр пример готовы сборки компьютера под заказ для клиента на все характеристик с возможность изменения конфигурацию.



Видео-карта
NVIDIA GEFORCE GTX 1080 Ti 11Гб

Процессор
Intel Core i7-7700 3600 МГц

Оперативная память
DDR4 16Гб KINGSTON HyperX FURY

Материнская плата
MSI Z270 GAMING PRO CARBON

Охлаждение
Corsair Hydro H100i

Жёсткий диск
1000 Гб 7200 rpm Seagate

Диск SSD
240 Гб Kingston

Блок питания
AeroCool KCAS 1000W

Беспроводная связь
Не установлено

Система
Windows 10

[все характеристики](#) [изменить конфигурацию](#)

 **144 240 р.**

Рис.3.23. Пример готовой сборки компьютера под заказ



Видео-карта
NVIDIA GEFORCE GTX 1080 Ti 11Гб SLI

Процессор
Intel Core i7-7700K 4200 МГц

Оперативная память
DDR4 32 Гб Corsair Vengeance LED

Материнская плата
Gigabyte GA-Z270X-Gaming 7

Охлаждение
Corsair Hydro H110i

Жёсткий диск
4000 Гб 7200 rpm Seagate

Диск SSD
480 Гб Corsair Neutron XT

Блок питания
Corsair RM1000

Беспроводная связь
Wi-Fi Внутренний PCI-E TP-Link

Система
Windows 10 Pro

[все характеристики](#) [изменить конфигурацию](#)

 **255 600 р.**

Рис.3.24. Пример готовой сборки компьютера под заказ

На рис.3.25 производится просмотр структура проект базы данных в PostgreSQL

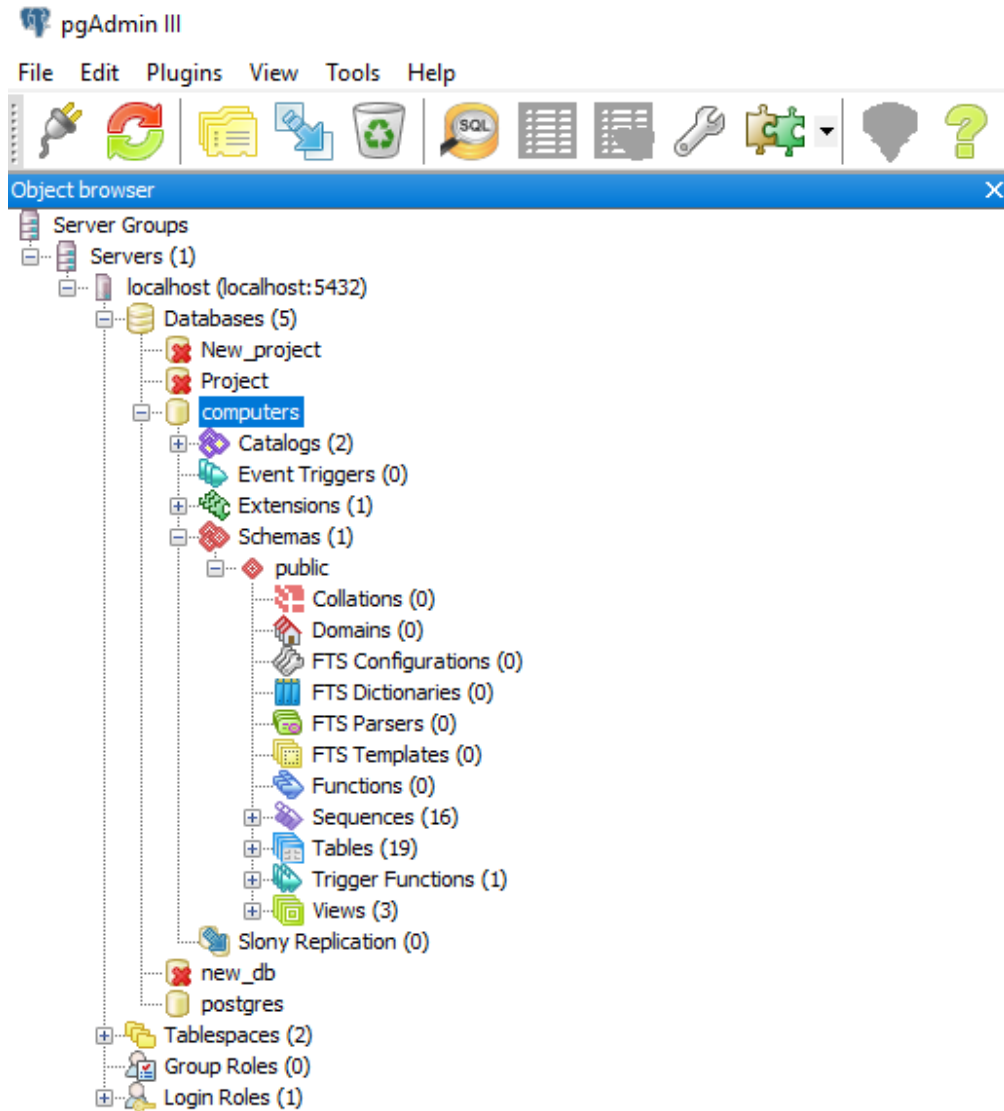


Рис 3.25 структура базы данных в PostgreSQL

ЗАКЛЮЧЕНИЕ

Разработанный конфигуратор позволит компании увеличить продажи, расширить географию покупателей, увеличить общую эффективность работы компании. Однако, внедрение конфигуратора должно быть связано с изменением некоторых бизнес-процессов компании, в частности, требует наличия должности администратора сайта (это, как правило, менеджер отдела продаж), то есть требует большей связи с информационными технологиями.

При решении задачи проведения обзора существующих систем и технологий для создания конфигуратора были выявлены преимущества и недостатки готовых решений, а также было установлено, что реализованные технологии в исследуемых решениях не подходят под установленные требования.

Решая задачи, направленные на разработку архитектуры и проектирование системы конфигуратор для сборки ПК было принято решение использовать существующую трехуровневую архитектуру клиент-сервер. Спроектированная система выполняет все выявленные общие требования.

В результате выполнения всех вышеописанных задач решается главная задача, направленная на реализацию системы конфигуратора, и которая выполняет основную цель работы, все поставленные требования, и которая обладает функционалом, необходимым для администрирования.

Приложение спроектировано с использованием современных методик автоматизации бизнес-процессов компании, поэтому коренной перестройки процессов не потребуется.

Необходимость разработки конфигуратора обусловлена необходимостью расширения продаж, выходом на всероссийский рынок.

Разработанный конфигуратор может использоваться на любом аналогичной рассмотренной компании с учетом доработок под конкретные виды компьютером и правила, и способы доставок, оплаты и работы магазина. Необходимость разработки конфигуратора обусловлена необходимостью расширения продаж, выходом на всероссийский рынок.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Маклаков С.В. Создание информационных систем с Infusion Modeling Suite. -М.: ДИАЛОГ_МИФИ, 2005-435с.
2. 2.Культин Н. Б. С/С++ в задачах и примерах. — СПб. БХВ-Петербург, 2005.
3. 3.SQL. Руководство по изучению языка. Справочное и методическое пособие: К. Фиайли - Санкт-Петербург, Бином-Пресс, 201 г.- 960 с.
4. Nakano, Russell, (2002), Web Content Management: a collaborative approach, Addison- Wesley
5. 4.Колисниченко Д.О. Профессиональное программирование на PHP.- СПб.: ВHV - Петербург, 2007.- 416 с.
6. Коггзолл Д. PHP 5 Полное руководство / Пер. с англ.-М.:Вильямс, 2006.- 752 с. (ч.з. ИнЕУ) (ч.з. ИнЕУ).
7. Хомоненко А.Д. и др. Базы данных: Учебник для средних и высших учебных заведений /А.Д. Хомоненко, В.М.Цыганков, М.Г. Мальцев.- СПб.: КОРОНА принт, 2006,-416 с. (ч.з. ИнЕУ).
8. Connolly, T, Begg, C. 2005, Database Systems. A Practical Approach to Design, Implementation, and Management, 4td ed. Addison-Wesley
9. Jepson, B, PostgreSQL vs. MySQL: building better databases, www.webtechniques.com/archives/2001/09/jepson/ [6th Decemeber 2005]
- 10.Маклаков С.В. BPWin и ERWin. CASE. - средства разработки информационных систем. М.: Диалог-МИФИ, 2000.
- 11.Яргер Р., Риз Дж., Кинг Т. MySQL и mSQL. Базы данных для небольших предприятий и Интернета. - СПб: Символ-Плюс, 2000 - 560 с.
- 12.Орлов С.Д. Технологии разработки программного обеспечения. Учеб. пособие. 2-е изд. СПб.: Питер, 2003. – 480 с.

13. Томсон Л., Веллинг Л. Разработка Web-приложений на PHP и MySQL. - К.: "ДиаСофт", 2001. - 672 с
14. Ратшиллер Т., Геркен Т. PHP4: разработка Web-приложений. - СПб: Питер, 2001. - 384 с.
15. Савицкая Г.В. Анализ хозяйственной деятельности предприятия. [Текст]/ Савицкая Г.В. Минск: ООО "Новое знание", 2001 год.
16. Макнейл П. Веб-дизайн. Книга идей веб – разработчика. – СПб.: Питер, 2014. – 288 с.: ил.
17. Лещев Д. Создание интерактивного web-сайта: учеб. курс / Д. Лещев. – СПб. и др.: Питер: Питер принт, 2005. - 543 с.

ПРИЛОЖЕНИЕ**Исходный код файла для конфигулятора**

```
package com.springbootapp.controller;
import com.springbootapp.DAO.impl.ComputerAssemblyServiceImpl;
import com.springbootapp.entity.ComputerAssembly;
import org.hibernate.Query;
import org.hibernate.Session;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import javax.annotation.Resource;
import java.util.List;
import java.util.Map;
import static com.springbootapp.config.HibernateUtil.getSessionFactory;
@Controller
@RequestMapping("/configuration")
public class Configuration {
    @Resource
    ComputerAssemblyServiceImpl computerAssemblyService;
    @RequestMapping(value = "/deleteComputer", method =
RequestMethod.POST)
    public void deleteComputer(@RequestParam("id_computer") int id){
        Session session = getSessionFactory().openSession();
        session.beginTransaction();
        String sql = "DELETE FROM computer_assembly WHERE id = " +
id;
        Query query = session.createSQLQuery(sql);
        query.executeUpdate();
    }
}
```

```

        session.getTransaction().commit();
        session.close();
    }
    @RequestMapping(value = "/updateComputer", method =
RequestMethod.POST)
    public void updateComputer(
        @RequestParam("id_computer") int id,
        @RequestParam("motherboard") int motherboard,
        @RequestParam("processor") int processor,
        @RequestParam("cooler") int cooler,
        @RequestParam("ram") int ram,
        @RequestParam("graphics") int graphics,
        @RequestParam("harddrive") int harddrive,
        @RequestParam("power") int power,
        @RequestParam("cases") int cases,
        @RequestParam("monitor") int monitor,
        @RequestParam("keyboard") int keyboard,
        @RequestParam("mouse") int mouse,
        @RequestParam("software") int software,
        @RequestParam("totalPrice") int totalPrice,
        @RequestParam("username") String username,
        @RequestParam("idWork") int inwork
    ){
        Session session = getSessionFactory().openSession();
        session.beginTransaction();
        String sql2 = "UPDATE computer_assembly " +
            "SET \"id_processor\" = " + processor + "," +
            "\"id_motherboard\" = " + motherboard + "," +

```

```

"\id_case\" = " + cases + ","+
"\id_ram\" = " + ram + ","+
"\id_harddrive\" = " + harddrive + ","+
"\id_graphicscard\" = " + graphics + ","+
"\id_cooler\" = " + cooler + ","+
"\id_powermodule\" = " + power + ","+
"\id_keyboard\" = " + keyboard + ","+
"\id_mouse\" = " + mouse + ","+
"\id_software\" = " + software + ","+
"\id_monitors\" = " + monitor + ","+
"\totalprice\" = " + totalPrice + ","+
"\username\" = " + username + ","+
"\in_work\" = " + inwork + " "+
"WHERE \id\"=" + id + "";

```

```

Query query = session.createSQLQuery(sql2);
query.executeUpdate();
session.getTransaction().commit();
session.close();
}
@RequestMapping(value = "/register", method = RequestMethod.POST)
public void addComputer(
    @RequestParam("username") String username,
    @RequestParam("email") String email,
    @RequestParam("password") String password) {
    Session session = getSessionFactory().openSession();
    session.beginTransaction();

```

```

String sql = "INSERT INTO clients(" +
    "\"username\", \"password\", \"enabled\"" +
    "VALUES ('" + username + "', '" + password + "', 'TRUE)";
String sql2 = "INSERT INTO role(" +
    "\"username\", \"authority\"" +
    "VALUES ('" + username + "', 'ROLE_USER)";
Query query = session.createSQLQuery(sql);
query.executeUpdate();
Query query2 = session.createSQLQuery(sql2);
query2.executeUpdate();
session.getTransaction().commit();
session.close();
}
@RequestMapping(value = "/addComputer", method =
RequestMethod.POST)
public void addComputer(
    @RequestParam("motherboard") int motherboard,
    @RequestParam("processor") int processor,
    @RequestParam("cooler") int cooler,
    @RequestParam("ram") int ram,
    @RequestParam("graphics") int graphics,
    @RequestParam("harddrive") int harddrive,
    @RequestParam("power") int power,
    @RequestParam("cases") int cases,
    @RequestParam("monitor") int monitor,
    @RequestParam("keyboard") int keyboard,
    @RequestParam("mouse") int mouse,
    @RequestParam("software") int software,

```

```

        @RequestParam("totalPrice") int totalPrice,
        @RequestParam("username") String username
    ){
        Session session = getSessionFactory().openSession();
        session.beginTransaction();
        List<ComputerAssembly> qwe = computerAssemblyService.getAll();
        int id = qwe.size() + 1;
        String sql = "INSERT INTO computer_assembly (\\"id\\", \\"id_case\\",
\\"id_cooler\\", \\"id_graphicscard\\", \\"id_harddrive\\", \\"id_keyboard\\",
\\"id_monitors\\", \\"id_motherboard\\", \\"id_mouse\\", \\"id_powermodule\\",
\\"id_processor\\", \\"id_ram\\", \\"id_software\\", \\"totalprice\\", \\"username\\",
\\"in_work\\")\n" +
            "VALUES(" + id + ", " + cases + ", " + cooler + ", " + graphics +
            ", " + harddrive + ", " + keyboard + ", " + monitor + ", " + motherboard + ",
            " + mouse+ ", " + power + ", " + processor + ", " + ram + ", " + software + ",
            " + totalPrice + ", " + username + ", '0)";
        Query query = session.createSQLQuery(sql);
        query.executeUpdate();
        session.getTransaction().commit();
        session.close();
    }
    @RequestMapping(value = "/addWork", method =
RequestMethod.POST)
    public void addComputer(
        @RequestParam("id") int idmotherboard
    ){
        Session session = getSessionFactory().openSession();
        session.beginTransaction();

```

```
String sql = "UPDATE computer_assembly\n" +
    "SET in_work = 1" +
    "WHERE id = " + idmotherboard + """;
Query query = session.createSQLQuery(sql);
query.executeUpdate();
session.getTransaction().commit();
session.close();
}
@RequestMapping
public String configuration(Map<String, Object> model) {
    return "pages/configuration";
}
@RequestMapping("/motherboard")
public String motherboard(Map<String, Object> model) {
    return "pages/configuration/motherboard";
}
@RequestMapping("/processor")
public String processor(Map<String, Object> model) {
    return "pages/configuration/processor";
}
@RequestMapping("/cooler")
public String cooler(Map<String, Object> model) {
    return "pages/configuration/cooler";
}
@RequestMapping("/ram")
public String ram(Map<String, Object> model) {
    return "pages/configuration/ram";
}
}
```

```
@RequestMapping("/graphics")
public String graphics(Map<String, Object> model) {
    return "pages/configuration/graphics";
}

@RequestMapping("/harddrive")
public String harddrive(Map<String, Object> model) {
return "pages/configuration/harddrive";
}

@RequestMapping("/power")
public String power(Map<String, Object> model) {
    return "pages/configuration/power";
}

@RequestMapping("/case")
public String cases(Map<String, Object> model) {
    return "pages/configuration/case";
}

@RequestMapping("/monitor")
public String monitor(Map<String, Object> model) {
    return "pages/configuration/monitor";
}

@RequestMapping("/keyboard")
public String keyboard(Map<String, Object> model) {
    return "pages/configuration/keyboard";
}

@RequestMapping("/mouse")
public String mouse(Map<String, Object> model) {
    return "pages/configuration/mouse";
}
```

```
@RequestMapping("/software")
public String software(Map<String, Object> model) {
    return "pages/configuration/software";
}
}
<script>
function register() {
    $.ajax({
        url: '/configuration/register',
        type: 'POST',
        data: {
            'username': $("#username").val(),
            'email': $("#email").val(),
            'password': $("#password").val()
        },
        beforeSend: function(xhr){
            xhr.setRequestHeader($("#meta[name='_csrf_header']").attr("content"),
                $("#meta[name='_csrf']").attr("content"));
        },
        success: function(data) {
            console.log(data);
            console.log('the best');
        },
        error: function (xhr, ajaxOptions, thrownError) {
            console.log(xhr.status + ": " + thrownError);
        }
    });
}
```



```

</script>
<div class="container">
  <div class="navbar">
    <div class="navbar-inner">
      <ul class="nav">
        <li><a th:href="@{/}" href="home.html">
Home </a></li>
      </ul>
    </div>
  </div>
  <div th:if="{#httpServletRequest.remoteUser != null}">
    <p th:text="{#httpServletRequest.remoteUser}">
      sample_user
    </p>
  </div>
  <h1 th:text="{title}">Title</h1>
  <div th:text="{message}">Fake content</div>
  <div id="created" th:text="{#dates.format(date)}">July 11,
  </div>
  <form th:action="@{/logout}" method="post">
    <input type="submit" value="Sign Out"/>
  </form>
</div>
</body>
</html>

```

```
@RequestMapping("/panel")
public String panel(Map<String, Object> model) {
    return "pages/panel";
}

@RequestMapping("/test")
public String test(Map<String, Object> model) {
    return "angular/index";
}

@RequestMapping("/index")
public String index(Map<String, Object> model) {
    return "index";
}

@RequestMapping("/contact")
public String contact(Map<String, Object> model) {
    return "pages/contact";
}

@RequestMapping("/admin")
public String admin(Map<String, Object> model) {
    return "pages/adminpanel";
}

@RequestMapping("/warranty")
public String warranty(Map<String, Object> model) {
    return "pages/warranty";
}

@Resource
ComputerAssemblyServiceImpl computerAssemblyService;
```

```
@RequestMapping("/admin/{id}")  
public String updateComputers(Model model, @PathVariable("id") int id) {  
    model.addAttribute("computer", id);  
    return "pages/updatecomputer";  
}
```

