

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

ПРИКЛАДНАЯ ИНФОРМАТИКА В СОЦИАЛЬНОЙ СФЕРЕ

БАКАЛАВРСКАЯ РАБОТА

на тему: **Разработка Web-сайта для CRM системы ООО «ФАС»**

Студент К.А. Филоненко _____

Руководитель А.И. Туищев _____

Консультант
по аннотации _____

Допустить к защите

Заведующий кафедрой к.тех.н, доцент, А.В. Очеповский _____

« _____ » _____ 2017г.

Тольятти 2017

АННОТАЦИЯ

Тема работы: Разработка Web-сайта для CRM системы ООО «ФАС».

Бакалаврская работа посвящена развитию создаваемой CRM системы заказчика посредством реализации элементов Web-сайта.

Целью бакалаврской работы является проектирование и разработка Web-сайта – фронт офиса CRM системы ООО «ФАС».

Для достижения поставленной цели были поставлены следующие задачи:

1. Проанализировать бизнес-область и изучить требования заказчика.
2. Провести проектирование архитектуры сайта и моделирование его элементов.
3. Реализовать элементы Web-сайта.
4. Провести тестирование разработанного программного обеспечения и разработать план внедрения.

В первой главе бакалаврской работы производится анализ ситуации «Как-есть», анализируются недостатки, формируются требования к разработке, производится обзор аналогов и принимается решение о разработке.

Во второй главе приведено описание проектирования архитектуры сайта и проектирование его модулей.

Третья глава посвящена вопросам тестирования и развертывания разработанного программного обеспечения.

Бакалаврская работа содержит 78 страниц, 37 рисунков, 10 таблицы, 35 источников.

ABSTRACT

The title of the graduation work is Development of a Web-site for the CRM system of FAS Ltd.

Bachelor's work is dedicated to the development of the customer's CRM system for the implementation of elements of the Web site.

The aim of the bachelor's work is the design and development of the Web site - the front of the CRM office of the FAS LLC.

To achieve this goal, the following tasks were set:

1. Analyze the business area and study the requirements of the customer.
2. Design the site architecture and model its elements.
3. Implement the elements of the Web site.
4. Test the developed software and develop an implementation plan.

In the first chapter of the bachelor's work, the situation is analyzed "As-is", shortcomings are analyzed, requirements for the development, analysis and analysis of solutions are formed.

The second chapter describes the design of the architecture and the design of its modules.

The third chapter is devoted to testing and deployment of the developed software.

Bachelor's work contains 78 pages, 37 drawings, 10 tables, 35 sources.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
Глава 1 АНАЛИЗ И МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ АВТОМАТИЗАЦИИ.....	6
1.1 Описание структуры компании ООО “ФАС”	6
1.2 Анализ существующей технологии продажи	8
1.3 Сравнительный анализ существующих CRM систем	12
1.4 Выбор подхода и методологии проектирования	22
1.5 Выбор CASE-средств.....	25
Глава 2 ИССЛЕДОВАНИЕ И РАЗРАБОТКА ПРОЕКТНЫХ РЕШЕНИЙ	28
2.1 Разработка модели будущей технологии продаж.....	28
2.2 Проектирование логики (алгоритма) работы приложений	33
2.3 Разработка архитектуры CRM-системы	38
2.4 Разработка подсистемы хранения данных	41
2.4.1 Этапы проектирования подсистемы хранения данных.....	41
2.4.2 Концептуальное проектирование CRM-системы	43
2.4.3 Разработка логической модели данных	46
Глава 3 РЕАЛИЗАЦИЯ, ТЕСТИРОВАНИЕ И ВНЕДРЕНИЕ WEB САЙТА	49
3.1 Выбор СУБД и средств разработки CRM-системы.....	49
3.2. Выбор технологической платформы разработки	52
3.2.1. Технология WebForms	52
3.2.2. Технология MVC Framework	53
3.3 Реализация доступа к данным	55
3.3.1 Выбор библиотеки ORM	55
3.3.2 Разработка физической модели БД.....	56
3.3.3 Реализация классов репозитория.....	57
3.4. Разработка контроллеров	61
3.4.1. Разработка фабрики контроллеров.....	61
3.4.2. Классы контроллеров приложения	62
3.5 Разработка представлений и интерфейса приложения	64

3.5.1 Разработка шаблонов страниц приложения	64
3.5.2 Разметка представлений	66
3.6. Наполнение базы данных и тестирование приложения.....	71
ЗАКЛЮЧЕНИЕ	73
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	75
ПРИЛОЖЕНИЕ А	Ошибка! Закладка не определена.
ПРИЛОЖЕНИЕ Б.....	84
ПРИЛОЖЕНИЕ В	86
ПРИЛОЖЕНИЕ Г	91

ВВЕДЕНИЕ

Система управления взаимодействием с клиентами (Customer Relationship Management System – CRM) – стратегия, основанная на применении новых управленческих и информационных технологий, с помощью которых компании аккумулируют знания о клиентах и выстраивают взаимовыгодных отношений с ними. Информационные подсистемы данного класса способствуют увеличению прибыли в силу того, что привлекают новых клиентов, а также помогают удерживать старых. Концепция CRM реализуется с помощью набора специализированного программного обеспечения и технологий, позволяющих автоматизировать и совершенствовать бизнес-процессы в сфере продаж, маркетинга и обслуживания клиентов. “Все это дает возможность компании обращаться к заказчикам услуг с интересными предложениями в наиболее удобный момент времени и по наиболее удобным каналам связи” [21].

“CRM расширяет концепцию продажи от дискретного действия, выполненного продавцом, к непрерывному процессу, вовлекающего каждого сотрудника компании. Это - искусство и наука сбора и использования информации о клиентах, позволяющая повышать лояльность клиента и увеличивать его ценность. При текущем уровне развития информационных технологий, и высоких ожиданиях клиентов в качестве обслуживания, практически невозможно подходить к этим проблемам без использования соответствующих технологий. Необходимо отметить, что отношения с клиентом - это, прежде всего, человеческие отношения, которые и являются основной движущей силой” [21].

Актуальность данной работы заключается в потребности ООО “ФАС” в организации клиенто-ориентированного бизнеса для повышения уровня удержания и удовлетворенности своих наиболее прибыльных клиентов, при одновременном снижении издержек и увеличении эффективности взаимодействия с клиентом.

Новизна работы заключается в разработке CRM-системы, наиболее полно удовлетворяющей потребностям ООО «ФАС», оптимально соответствующей специфике данной фирмы и, тем самым, по своим совокупным характеристикам, в большей степени соответствующей требованиям ООО «ФАС», чем применение существующих готовых CRM-решений.

Объектом исследования является процесс взаимоотношения с клиентами ООО «ФАС».

Предметом исследования является CRM система ООО «ФАС».

Цель дипломной работы: разработать Web сайт для системы управления взаимоотношениями с клиентами ООО «ФАС».

Задачи дипломной работы:

1. Проанализировать предметную область автоматизации, построить модель и выдвинуть требования к Web сайту.
2. Выполнить этап проектирования Web сайта.
3. Реализовать и протестировать Web сайт.
4. Разработать план внедрения Web сайта.

Бакалаврская работа состоит из введения, трех глав, заключения, списка используемой литературы, приложений.

В первой главе выпускной квалификационной работы производится анализ ситуации «Как-есть», анализируются недостатки, формируются требования к разработке, производится обзор аналогов и принимается решение о разработке.

Во второй главе приведено описание проектирования архитектуры сайта и проектирование его модулей.

Третья глава посвящена вопросам тестирования и развертывания разработанного программного обеспечения.

Глава 1 АНАЛИЗ И МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ АВТОМАТИЗАЦИИ

1.1 Описание структуры компании ООО “ФАС”

Компания ООО “ФАС” была создана в 2006 году. Основной сферой деятельности компании является оптовая продажа автозапчастей для автомобилей ВАЗ и других марок на рынке автозапчастей. Скомплектованный заказ может быть отгружен со склада на условиях самовывоза, доставлен по железной дороге в стандартных 3-х тонных, 5-ти тонных или 20-ти тонных контейнерах, а так же отправлен любой транспортной компанией, расположенной в Тольятти. Кроме того, по желанию клиента возможно совмещение заказа с партией деталей другого поставщика.

Организационно - штатная структура компании ООО “ФАС” включает генерального директора, финансового отдела во главе с главным бухгалтером, отдела продаж и подразделений снабжения. Начальник отдела продаж является заместителем генерального директора. В штат отдела продаж входят два менеджера по продажам. Подразделения снабжения - склад и отдел доставки - подчиняются заместителю генерального директора по снабжению.

В ходе осуществления бизнес-процесса продаж принимают участие: генеральный директор, сотрудники бухгалтерии, начальник отдела продаж и его менеджеры, сотрудники склада и отдела доставки. Структурная схема компании представлена на рис.1.1. На схеме цветом выделены подразделения которые задействованы в процессе продаж. Деятельность этих подразделений полностью или частично будет автоматизироваться с помощью разрабатываемой системы. Непосредственно с клиентами в компании работают 3 человека: зам директора по продажам и два менеджера по продажам.

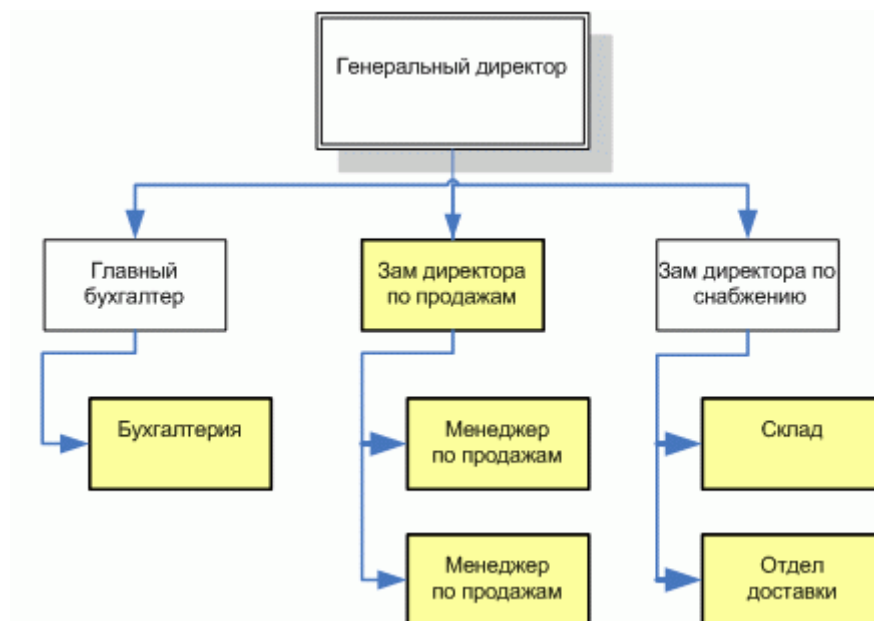


Рисунок 1.1 - Структурная схема компании

В отделе продаж создана локальная сеть из 3-х ПЭВМ.

Менеджеры отдела продаж поддерживают в актуальном состоянии сайт-визитку компании, который размещен в интернете с использованием внешнего хостинга. IT структура отдела продаж представлена на рис. 1.2.

Основным ПО, используемым в процессе работы, являются Microsoft Office Word, Microsoft Office Excel и веб-браузер Internet Explorer.

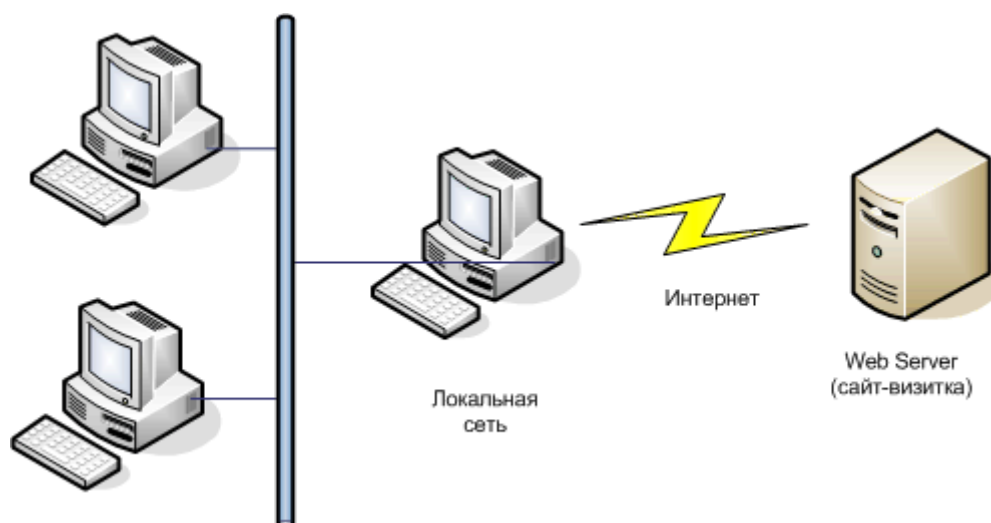


Рисунок 1.2 – IT структура отдела продаж компании

В силу того, что при работе с клиентами осуществляется достаточно большое количество операций и формируется ряд документов, рассмотрим

существующую технологию подробнее, выполнив моделирование по методологии IDEF0

1.2 Анализ существующей технологии продажи

В настоящее время клиент для совершения покупки должен прийти в офис компании и сообщить перечень нужных ему автозапчастей. Менеджер по продажам оформляет договор купли - продажи и выставляет счет клиенту. После оплаты счета, формируется партия заказанных товаров и отправляется клиенту в соответствии с договором. Данные по продажам фиксируются на бумажных носителях и в отдельных файлах.

Построим диаграммы IDEF0 бизнес процесса продажи. Выберем точку зрения менеджера по продажам. Диаграмма уровня А-0 изображена на рис.1.3.

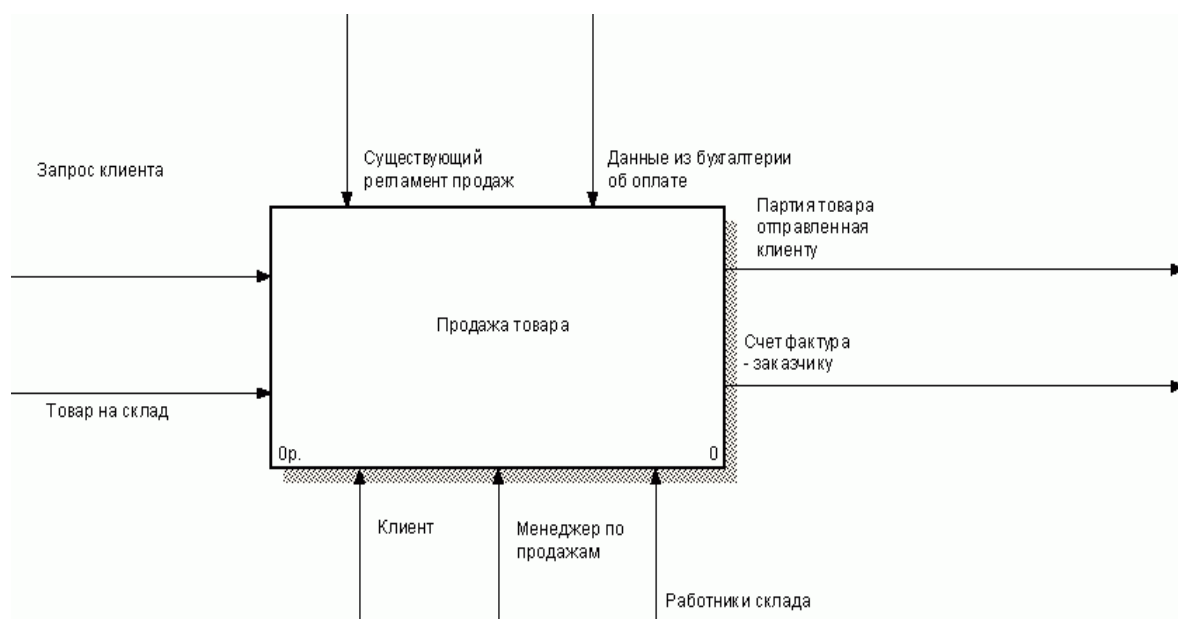


Рисунок 1.3 – Диаграмма А-0 «Продажа товара»

Бизнес-процесс продажи состоит из следующих работ:

- прием заказа от клиента;
- оформление договора купли продажи;
- оформление счета и выставление его клиенту;
- отправка товара клиенту.

Данные работы изображены на рис.1.4.

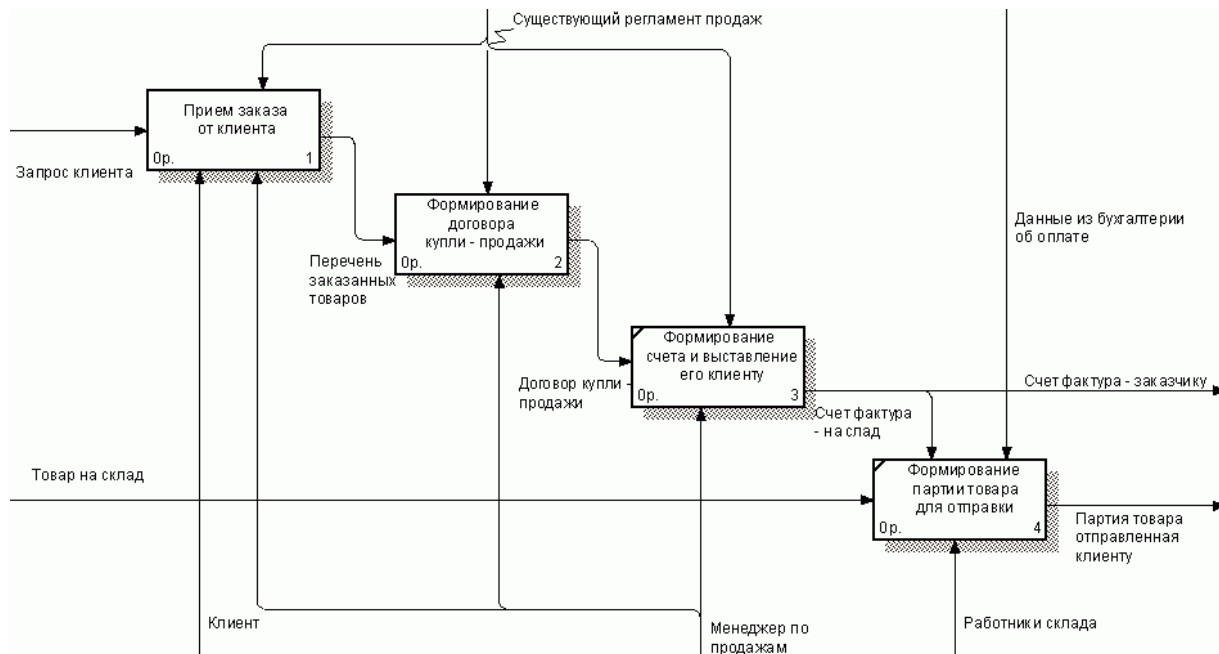


Рисунок 1.4 – Диаграмма А0 «Продажа товаров»

Прием заказа от клиента в свою очередь состоит из следующих действий:

- ознакомить клиента с действующим прайс-листом;
- проверить наличие на складе выбранных клиентом позиций, и уточнить, при необходимости, сроки доставки товаров под заказ;
- согласовать с клиентом окончательный список товаров.

Данные работы изображены на рис.1.5.

Оформление договора купли продажи в свою очередь состоит из следующих действий:

- проверить существование и сроки действия долгосрочного договора купли - продажи с данным клиентом;
- в случае отсутствия такого договора, заключить новый договор купли - продажи с данным клиентом;
- в случае истечения сроков действия договора купли -продажи, заключить новый договор купли - продажи с данным клиентом.

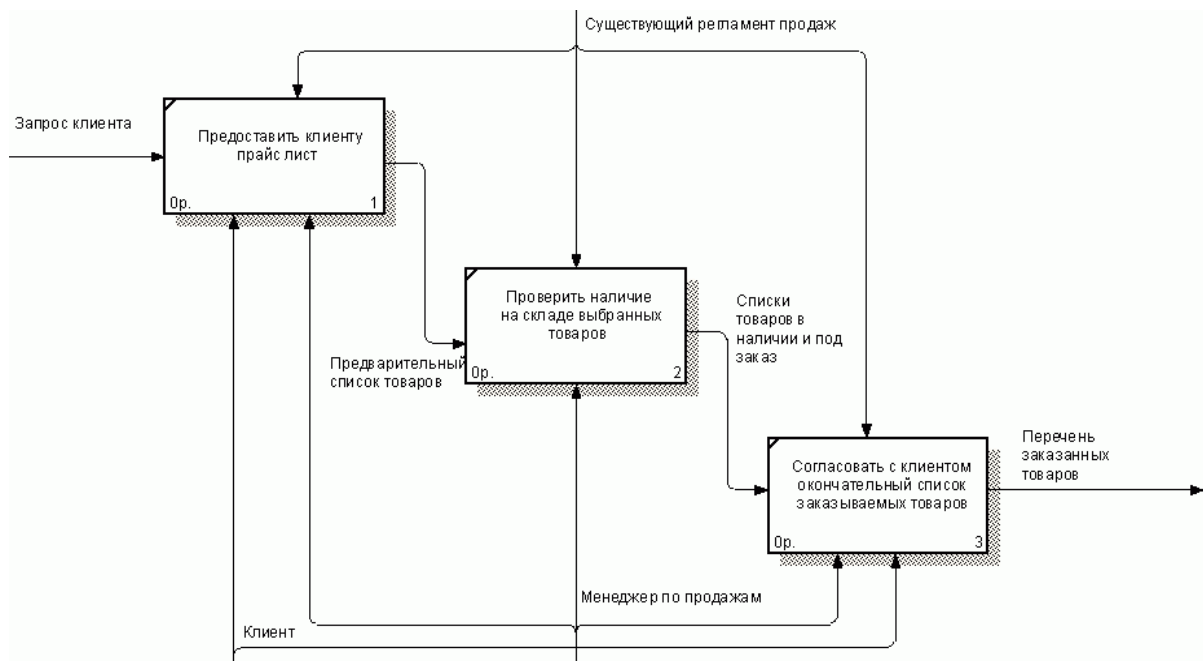


Рисунок 1.5 – Диаграмма А1 «Прием заказа от клиента»

Оформление договора купли продажи в свою очередь состоит из следующих действий:

- проверить существование и сроки действия долгосрочного договора купли - продажи с данным клиентом;
- в случае отсутствия такого договора, заключить новый договор купли - продажи с данным клиентом;
- в случае истечения сроков действия договора купли -продажи, заключить новый договор купли - продажи с данным клиентом.

Данные работы изображены на рис.1.6.

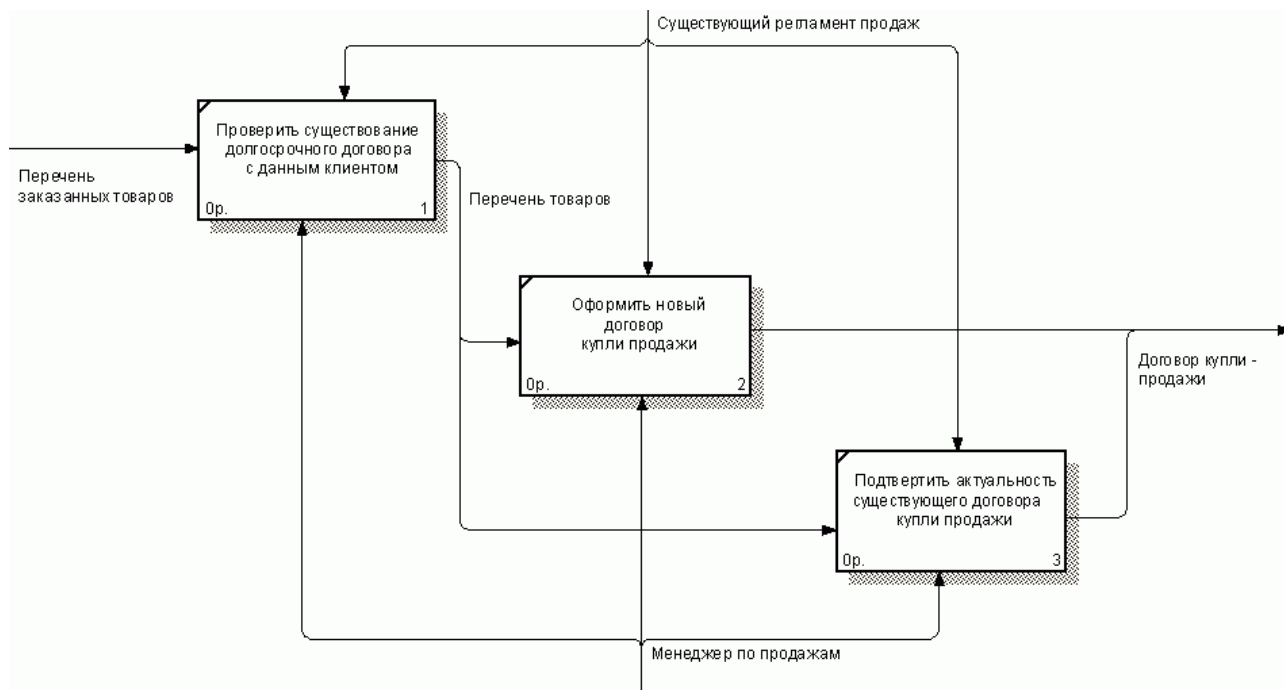


Рисунок 1.6 – Диаграмма А2 «Формирование договора купли–продажи»

Недостатками существующего процесса являются:

- обязательное присутствие клиентов для оформления заказа;
- временные затраты на процесс оформления, как клиента, так и менеджера;
- невозможность осуществления взаимодействия с клиентом через интернет;
- невозможно вести базу данных клиентов;
- невозможно оперативно определить степень участия сотрудников в продажах;
- фиксация данных на бумажных носителях не позволяет производить их автоматизированную обработку (история клиента, отслеживание состояния заказа и т.п.).

Для устранения указанных недостатков необходимо автоматизировать процесс продажи товара.

Автоматизация процесса позволит:

- повысить качество обслуживания клиентов;

- стандартизировать форматы данных по работе с клиентами и аккумулировать их в единой базе данных;
- обеспечить контроль деятельности менеджеров;
- предоставить доступ к отчетам для топ-менеджеров в режиме он-лайн;
- осуществить предварительную регистрацию потенциальных клиентов зашедших на сайт для ознакомления;
- предоставить возможность клиентам, заключившим длительные договора, на осуществление покупок товаров фирмы дистанционно через интернет;
- предоставить клиентам возможность контролировать стадии реализации их заказов.

Исходя из того, что CRM-система – это система управления взаимодействием с клиентами, служащая для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов, все вышеперечисленные пункты относятся к CRM-системе.

Разработать новую технологию можно двумя путями:

1. Закупить готовые информационные решения и адаптировать их под требования заказчика.
2. Провести разработку системы «с нуля».

Проведем обзор аналогов CRM-решений

1.3 Сравнительный анализ существующих CRM систем

На данный момент на рынке CRM-систем наблюдается стабильный рост, а CRM-методология доказала свое право на существование и стала рабочим инструментом бизнеса. Ассортимент CRM-решений различного назначения, представленных на российском рынке, значительно скромнее по сравнению с западным. Компаниям предоставляется на выбор два десятка разноплановых пакетов, продвижением которых занимаются отечественные системные интеграторы и представительства зарубежных фирм-разработчиков, а также более десяти отечественных систем. В число специализированных западных

CRM-пакетов, предлагаемых в России, входят системы Siebel (IBS, «Ланит»), Remedy (TopS, «Открытые технологии»), CRM-решения SAS Institute, Invensys («БААН Евразия»), Pivotal (Columbus IT Partner Russia), Clientele («ЭпикРус»), Microsoft CRM Professional Edition, а также приложения для call-центров (в частности, от Avaya или Cisco). CRM-компоненты есть и в составе зарубежных учетно-управленческих систем, например от Frontstep, J. D. Edwards (Robertson & Blums), Oracle, Navision («ИмпактСофт», Columbus IT Partner Russia и другие партнеры) и SAP. Ассортимент решений отечественной разработки также включает клиентские компоненты российских учетно-управленческих систем («Галактика», «Парус») и автономные продукты для автоматизации продаж и маркетинга [21].

Наиболее востребованные решения на российском рынке:

- 1С-Парус: 1С:Предприятие 8. CRM. Базовая версия.
- Про-Инвест- ИТ : Quick Sales;
- Про-Инвест- ИТ : Sales Expert;
- Terrasoft : TerraSoft CRM;
- Компас : КОМПАС Маркетинг и Менеджмент;
- Парус : ПАРУС Менеджмент и Маркетинг.

«1С:Предприятие 8. CRM. Базовая версия» представляет собой однопользовательскую программу, которая позволяет автоматизировать процессы управления взаимоотношениями с клиентами и поставщиками в компаниях малого бизнеса и у индивидуальных предпринимателей [17]. В средних и крупных компаниях продукт «1С:CRM. Базовая версия» может использоваться для задач локальной автоматизации рабочих мест отдельных сотрудников.

Продукт «1С:CRM. Базовая версия» обеспечивает:

- сбор, хранение и обработку информации о клиентах и истории взаимоотношений с ними;
- оперативное получение полной информации о клиентах и сделках;

– автоматизацию процесса продаж и закупок, телемаркетинга и e-mail рассылок;

– сбор, хранение и быстрый доступ к «базе знаний» компании;

– сокращение временных затрат на рутинные операции продаж и закупок;

– календарное планирование работы сотрудника;

– аналитическую отчетность.

CRM-системы Sales Expert, Quick Sales предназначены для ведения клиентской базы и фиксирования истории работы с каждым клиентом, а также для анализа накопленной информации [22].

В рамках организации сбыта и управления процессом продаж системы решают следующие основные задачи:

– сбор и обработка информации о клиентах и рынке в единой базе данных;

– определение процедур и правил работы менеджеров по продажам;

– предоставление сотрудникам отделов продаж автоматизированного рабочего места;

– контроль работы с клиентами и анализ сбытовой деятельности компании;

– оценка результативности маркетинга;

– защита клиентской базы;

– согласованная работа сотрудников различных подразделений с клиентами.

Под маркой Sales Expert представлено два решения: Sales Expert I (новое название – Quick Sales) и Sales Expert II. Обе системы решают задачу организации и управления продажами, но с различными возможностями по адаптации под бизнес клиента.

Система Terrasoft CRM предоставляет комплекс инструментов, позволяющих систематизировать данные о клиентской базе, процессы привлечения клиентов и развития отношений с ними, совершенствования

системы продаж, маркетинга и обслуживания, повышения удовлетворенности клиентов [23].

Программный продукт Terrasoft CRM позволяет решать следующие задачи:

- управление бизнес-процессами;
- управление контактами. история взаимодействия с клиентами;
- планирование и управление продажами;
- управление маркетинговыми кампаниями;
- автоматизация документооборота;
- управление рабочим временем;
- отслеживание результатов работы и аналитика.

CRM система "Маркетинг и менеджмент" позиционирует следующие свои преимущества [20]:

- высокая гибкость системы;
- отпуск товаров с учетом складских остатков;
- развитая система прайс-листов, наценок, скидок;
- наличие функционала контроля взаиморасчетов с предельными сроками платежей;
- процедура автоматического ранжирования клиентов - на основе анализа уровня их доходности;
- встроенный функционал управления материальной мотивацией;
- поддержка оптимальных стратегий общения с клиентом;
- планирование затрат и контроль бюджета на маркетинговые кампании;
- планирование маркетинговых кампаний по образцу;
- автоматизация внутреннего управления предприятием;
- сегментация клиентов по произвольным наборам признаков;
- наличие резидентной программы напоминания в режиме online.

Базовая конфигурация CRM-системы "ПАРУС - Менеджмент и Маркетинг" позволяет автоматизировать задачи [19]:

- ведения единой структурированной базы клиентов, потенциальных клиентов, партнеров и других контрагентов;
- ведения истории взаимоотношений с контрагентами;
- учета договорных отношений;
- обслуживания и поддержки клиентов;
- учета клубных и бонусных систем;
- планирования и учета продаж (воронка продаж);
- учета заявок от клиентов и их поэтапное выполнение;
- учета маркетинговых мероприятий и акций;
- учета загрузки персонала;
- ведения проектов;
- регистрации и адресного распределения входящих звонков и обращений;
- маркетинговых исследований и опросов;
- формирования отчетности и анализа данных;
- взаимодействия с бухгалтерскими системами, почтовыми клиентами, контакт-центрами, интернет-сайтами, контрольно-пропускными системами офиса (связь с журналом учета рабочего времени сотрудников), офисными АТС и Pocket PC и др.

Основные параметры рассматриваемых аналогов сведены в табл. 1.1 - 1.4.

Информация о разработчиках CRM-систем приведена в табл. 1.1.

Таблица 1.1 – CRM системы и их разработчики

Наименование программы	Разработчик	Страна	Главный офис	Официальный сайт
1	2	3	4	5
1С:Предприятие 8. CRM. Базовая версия	1С-Рарус	Россия	Москва	http://www.rarus.ru/
Quick Sales	Про-Инвест-ИТ	Россия	Москва	http://www.pro-invest.com/

1	2	3	4	5
Sales Expert	Про-Инвест-ИТ	Россия	Москва	http://www.pro-invest.com/
TerraSoft CRM	Terrasoft	Украина	Киев, Москва	www.tscrm.ru , www.tscrm.com
КОМПАС: Маркетинг и Менеджмент	Компас	Россия	СПб	http://www.compas.ru/
Парус- менеджмент и маркетинг	Парус	Россия	Москва	http://www.parus.ru/

Основные возможности CRM-систем приведены в табл. 1.2.

Таблица 1.2 – Основные возможности анализируемых CRM систем

Наименование программы	Самостоятельность решения	Возможность настройки в соответствии с бизнес-процессом	Совместимость	Импорт-экспорт
1	2	3	4	5
1С-Парус: CRM Контакты 1.0	Самостоятельно или модуль в 1С:Предприятие 7.7	+	1С:Предприятие	DBF, Excel и т.д.
Quick Sales	Самостоятельное	-	-	
1	2	3	4	5
Sales Expert	Самостоятельное	+	-	
TerraSoft CRM	Самостоятельное	+	1С, Парус, Navision, Бэст и другие	DBF, Excel и т.д.
КОМПАС: Маркетинг и Менеджмент	Самостоятельное или как модуль в ERP системе Компас	+	С системами с открытым кодом	DBF, Excel и т.д.
Парус-менеджмент и маркетинг	Самостоятельное	-	-	-

Требования рассматриваемых CRM-систем к СУБД приведены в табл.1.3.

Таблица 1.3 - Требования к СУБД

Наименование программы	СУБД платформа.	Количество записей в БД.	Используемое серверное ПО.	Потребность в дополнительном ПО.	Потребность в выделенном сервере
1	2	3	4	5	6
1С-Парус: CRM Контакты 1.0	1С	Ограничения СУБД	MS Windows NT 4.0/2000/XP/Server 2003	1С:Предприятие 7.7	Да, рекомендуется
Quick Sales	Firebird	Ограничения СУБД	MS Windows	Interbase 6	Нет
Sales Expert	Firebird	Ограничения СУБД	MS Windows	Interbase 6	Нет
TerraSoft CRM	MS SQL, CACHE	Ограничения СУБД	MS SQL Server, MSDE, CACHE	СУБД	До 30 пользователей не обязательно
КОМПАС: Маркетинг и Менеджмент	MS SQL, Oracle	Ограничения СУБД	MS SQL Server, Oracle SQL Server	СУБД	Да, рекомендуется
Парус-менеджмент и маркетинг		Ограничения СУБД			

Системные требования, предназначение и стоимость сведены в табл. 1.4.

Таблица 1.4 – Требования к оборудованию и стоимость

Наименование программы	Требования к серверу минимальные	Требования к рабочим местам минимальные	Предназначение	Стоимость (цены снижаются при увеличении количества лицензий)
1	2	3	4	5
1С-Парус: CRM Контакты 1.0	Pentium 3 (866), RAM от 256 Мб, от 20 Мб для ПО	Pentium 133 МHz, RAM от 64 Мб, от 20 Мб для ПО	Малый, средний, крупный	400\$ (5 пользователей) и т.д.
Quick Sales	Pentium от 133 МHz, RAM от 32 Мб, от 50 Мб для ПО	Pentium от 133 МHz, RAM от 16 Мб, от 10 Мб для ПО	Малый, средний, крупный	295\$ (3 рабочих места) и т.д.

1	2	3	4	5
Sales Expert	Pentium от 133 МHz, RAM от 32 Мб, от 50 Мб для ПО	Pentium от 133 МHz, RAM от 16 Мб, от 10 Мб для ПО	Малый, средний, крупный	860\$ (3 рабочих места) и т.д.
TerraSoft CRM	В соответствии с требованиями СУБД	Pentium от 133 МHz, RAM от 64 Мб, от 10 Мб для ПО	Малый и средний	от 299\$ за рабочее место
КОМПАС: Маркетинг и Менеджмент	Pentium 3 от 450, RAM от 512 Мб, от 10 Gb	Pentium от 133 МHz, RAM от 64 Мб, от 1 Gb для ПО	Малый и средний	От 480\$ за рабочее место (версия Элит)
Парус-менеджмент и маркетинг			Малый и средний	

В данной таблице приведены минимальные требования рассмотренных CRM-систем к серверному оборудованию и рабочим станциям, а также ориентировочная стоимость данных продуктов.

Основные функции, предоставляемые существующими CRM-системами следующие:

- ведение базы данных по клиентам. От типового «Справочника контрагентов», который есть в любой учетной системе, эта база отличается большим числом реквизитов (часто настраиваемых) и информацией о сотрудниках клиента;

- ведение контактов с клиентами. Фактически использование такой системы позволяет отказаться от бумажных ежедневников и планировщиков и устранить риск потери контактной информации при уходе отдельных сотрудников;

- ведение сделок. Под сделкой чаще всего понимается бизнес-процесс: поиск клиента – коммерческое предложение – этап уговаривания 1 – этап уговаривания N – этап согласования 1 – этап согласования N – заключение договора – оплата – поставка – установка... Число этапов зависит от конкретного продукта и настраивается под конкретный бизнес. По числу

потенциальных клиентов, находящихся на каждом этапе, создается «воронка продаж» (чем дальше этап, тем меньше клиентов), исходя из которой можно прогнозировать объемы будущих продаж;

– организация рассылок. То, что в CRM-программу встраиваются функции e-mail рассылки и подготовки писем по шаблону, сейчас фактически стало стандартом;

– построение выборок по клиентской базе. Клиентов, как правило, можно отфильтровать по всем параметрам клиентской базы. Далее выборка используется либо для подготовки отчетов, либо для организации рассылки.

Из перечисленных функций видно, что чаще всего CRM-системы используются предприятиями, предлагающими достаточно дорогое оборудование или услуги. На данном сегменте каждая продажа – событие, и путь к заключению контракта достаточно долг. Бизнес-процесс продажи в этом случае действительно делится на несколько этапов, и каждый этап тщательно отслеживается. Потеря каждого клиента может обернуться значительными убытками для компании. Естественно, CRM-технологии оказались востребованы прежде всего на рынках такого типа. Поэтому большинство классических CRM-программ, как зарубежных, так и отечественных, ориентированы именно на этот сегмент. Профессиональные версии этих программ платные и имеют лишнюю функциональность.

Для проведения сравнительного анализа исследуемых аналогов их характеристики сведены в табл.1.5.

Таблица 1.5 – Сравнительные характеристики исследуемых аналогов

Параметр / Аналог	1С - Парус	Quick Sales	Sales Expert	TerraSoft CRM	КОМПАС: Маркетинг и Менеджмент	Парус-менеджмент и маркетинг
1	2	3	4	5	6	7
Возможность	+	-	+	+	+	-

1	2	3	4	5	6	7
программирования						
Совместимость с 1С	+	-	-	+	-	-
Ведение базы клиентов	+	+	+	+	+	+
Система анализа и сегментирования клиентов	+	+	+	+	+	+
Возможность доступа через интернет	-	-	-	-	-	-
Использование серверных СУБД	-	-	-	+	+	-
Сумма баллов	4	2	3	5	4	2

Анализ известных решений показал, что все они обладают необходимым набором функций для достижения поставленной в работе цели.

В то же время эти системы дороги, обладают избыточной функциональностью и не позволяют в полной мере учесть специфику бизнес-процессов объекта исследования работы.

На основании вышеизложенного можно сделать вывод о целесообразности разработки собственной системы CRM.

Необходимо создать автоматизированную информационную систему для поддержки бизнеса ООО «ФАС». Данная АИС должна послужить инструментом, облегчающим работу менеджера по обработке заказов, а также для клиентов, избавляя их от обязательного присутствия при осуществлении заказа, предоставляя им гибкий механизм для оформления и конструирования заказов по своему усмотрению. К такому же этапу подошло и руководство ООО «ФАС», вследствие чего было принято решение об автоматизации системы взаимодействия с клиентами и разработке подсистемы хранения информации.

При этом разрабатываемая система должна способствовать переходу компании ООО «ФАС» к реализации CRM стратегии ведения бизнеса.

Для реализации необходимых функций и выполнения указанных требований необходимо:

- разработать концептуальную модель данных предметной области;
- разработать логическую модель базы данных;
- выбрать архитектуру системы;
- выбрать программные средства реализации системы;
- разработать физическую модель базы данных;
- разработать web-интерфейс пользователя;
- разработать модель обеспечения безопасности данных;
- обеспечить информационную поддержку пользователей;
- интегрировать все элементы в законченное приложение.

1.4 Выбор подхода и методологии проектирования

Существует два основных подхода к проектированию информационных систем [6]:

- структурный;
- объектно-ориентированный.

Сущность структурного подхода к разработке ИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны. При разработке системы "снизу вверх" от отдельных задач ко всей системе целостность теряется, возникают проблемы при информационной стыковке отдельных компонентов.

В структурном анализе используются в основном две группы средств, иллюстрирующих функции, выполняемые системой, и отношения между

данными. Каждой группе средств соответствуют определенные виды моделей (диаграмм), наиболее распространенными, среди которых являются следующие:

- SADT (Structured Analysis and Design Technique) модели и соответствующие функциональные диаграммы;
- DFD (Data Flow Diagrams) диаграммы потоков данных;
- ERD (Entity Relationship Diagrams) диаграммы "сущность связь".

На стадии проектирования ИС модели расширяются, уточняются и дополняются диаграммами, отражающими структуру программного обеспечения: архитектуру ПО, структурные схемы программ и диаграммы экранных форм.

Перечисленные модели в совокупности дают полное описание ИС независимо от того, является ли она существующей или вновь разрабатываемой. Состав диаграмм в каждом конкретном случае зависит от необходимой полноты описания системы.

Классический структурный подход к созданию информационных систем предполагает последовательную реализацию этапов анализа, проектирования, создания модулей, объединения модулей в единую систему, тестирования и внедрения. Применение CASE технологий и CASE средств, позволяет в несколько раз сократить время разработки информационных систем и значительно снизить вероятность появления ошибок за счет автоматизации начальных этапов разработки (а, как следствие более качественного планирования и проектирования) и автоматической генерации структуры сервера БД и кода клиентского приложения. Однако, эта технология не лишена недостатков. Код клиентского приложения генерируется на основе информации о структуре БД. К структуре БД предъявляются определенные требования (нормализации), в результате чего данные хранятся в таблицах БД не всегда в той же форме, в которой они должны представляться на экранных формах. Другими словами, если код приложения генерируется не на основе описания предметной области, невозможно построить эффективное приложение со

сложной бизнес-логикой. Кроме того, при структурном подходе к разработке информационных систем, риск остается большим на всех этапах создания системы вплоть до этапа тестирования, когда мы можем обнаружить допущенные ошибки и оценить состоятельность системы. В случае обнаружения ошибки, необходимо вернуться на тот этап разработки, на котором допущена ошибка, и заново пройти последующие этапы.

Альтернативой структурному подходу стали лишённые перечисленных недостатков объектно-ориентированные методы разработки информационных систем. В начале девяностых годов был предложен разработанный на основе наиболее популярных объектных методов универсальный язык объектного проектирования Unified Modeling Language, UML. Одним из CASE средств, поддерживающих язык UML, является выпущенный фирмой Rational Software программный пакет Rational Rose, который позволяет генерировать код приложения, в полной мере отвечающий бизнес-правилам и с наименьшим риском.

Снижение риска в объектной технологии достигается за счет реализации технологии итерационной разработки (так называемая спиральная модель жизненного цикла разработки). Разработка состоит из ряда итераций, которые в дальнейшем приводят к созданию информационной системы. Каждая итерация может приводить к созданию фрагмента или новой версии и включает этапы выработки требований, анализа, проектирования, реализации и тестирования. Поскольку тестирование проводится на каждой итерации, риск снижается уже на начальных этапах жизненного цикла разработки.

Модель в данном подходе представляет собой совокупность диаграмм, описывающих различные аспекты структуры и поведения информационной системы.

В рамках данной дипломной работы будем использовать структурный подход в нотациях IDEF0, DFD для описания функциональной модели системы.

1.5 Выбор CASE-средств

Стандарт ISO/IEC 14102.1995 [16] под case-средством понимает программное средство, поддерживающее процессы жизненного цикла ПО, включающие анализ требований, проектирование ПО и БД, генерацию кода, тестирование, документирование, обеспечение качества, управление конфигурацией ПО, управление проектом и т.д.

Для проектирования ИС сравним такие case-средства, как: BPWin, ARIS, Oracle Designer.

Результаты сравнительного анализа case-средств по этим критериям сведем в табл. 1.6.

Таблица 1.6 – Сравнение Case-средств проектирования

Показатель	BPWin	Oracle Designer	ARIS
1	2	3	4
Визуализация модели	3	3	3
Динамическое моделирование	0	0	3
Оптимизация модели	0	2	2
1	2	3	4
Функционально-стоимостной анализ	1	2	2
Генерация программного кода	3	2	0
Распространенность	3	2	2
Доступность	3	2	2
Разовый проект по описанию бизнес-процессов	3	2	1
Длительный (непрерывный) проект по описанию деятельности компании	3	3	3
Библиотека элементов ОБП	3	2	2
Групповая работа	2	2	2
Итого	24	22	22

Таким образом, по результатам проведения сравнительного анализа современных средств проектирования ИС, для решения поставленной задачи выбрано Case-средство BPWin.

Теперь необходимо выбрать case-средство для проектирования структуры БД, обеспечивающее моделирование и генерацию БД (SQL). Наиболее популярным на сегодняшний день средством является AllFusion ERwin Data Modeler, предназначенное для разработки информационных моделей по методике IDEF1X и позволяющее проектировать, документировать и сопровождать базы данных, хранилища данных и витрины данных. Так же у компании Rational Software, есть средство проектирования баз данных – Rational Rose Data Modeler. Это case-средство для моделирования и анализа баз данных, основанное на языке UML.

Сравним Erwin Data Modeler и Rational Rose Data Modeler по следующим критериям:

- поддержка большинства возможностей структурных case-средств в плане физического моделирования данных;
- обеспечение ссылочной целостности – генерация эффективной физической структуры БД, поддерживающей механизмы обеспечения ссылочной целостности;
- возможность прямого и обратного проектирования базы данных;
- возможность автоматической генерации SQL-запросов из логической структуры базы данных.

Результаты сравнительного анализа CASE-средств по этим критериям сведем в табл.1.7, причем наличие той или иной функции будем обозначать числом-баллом в интервале от 0 (не поддерживается) до 1 (полная поддержка).

Таким образом, по результатам проведенного сравнительного анализа критериев выбора case-средства для проектирования базы данных видно, что наиболее подходящим средством является AllFusion ERwin Data Modeler.

Подводя итоги, скажем, что, для проектирования нашей ИС будем использовать BPWin, а для проектирования базы данных – AllFusion ERwin Data Modeler.

Таблица 1.7 – Выбор CASE-средства проектирования базы данных

CASE-средство / Критерий оценки	Erwin Data Modeler	Rational Rose Data Modeler
1	2	3
Поддержка возможностей структурных case-средств	0.5	1
Обеспечение ссылочной целостности	1	1
Возможность прямого и обратного проектирования базы данных	1	1
Возможность генерации SQL-запросов	1	1
Возможность проектирования физической структуры базы данных	1	0
ИТОГО	4.5	4.0

Обобщая материалы главы можно сделать следующие выводы:

- анализ бизнес-процессов в компании ООО «ФАС» «As-Is», проведенный с помощью методологий IDEF0, подтвердил целесообразность внедрения CRM системы и позволил сформулировать требования к данной системе;

- в результате проделанной работы были сформированы и уточнены требования к проектируемой информационной системе и цель моделирования;

- анализ известных CRM - решений показал, что они не удовлетворяют в полной мере требованиям, предъявляемым к новой CRM. В этой связи принято решение о разработке CRM с учетом новых требований;

- был выбран объектно-ориентированный подход проектирования и методология RUP (Rational Unified Process);

- так же выбраны case-средства моделирования системы: BPwin и AllFusion ERwin Data Modeler.

Глава 2 ИССЛЕДОВАНИЕ И РАЗРАБОТКА ПРОЕКТНЫХ РЕШЕНИЙ

2.1 Разработка модели будущей технологии продаж

Проведенный анализ существующих в настоящий момент в ООО «ФАС» бизнес-процессов показывает, что текущая реализация этих процессов не является оптимальной. Для повышения уровня исполнения бизнес-процессов необходимо создать CRM-систему, способствующую улучшению качества обслуживания клиентов.

Предлагается следующий регламент продаж:

1. Клиент заходит на сайт компании и авторизуется.
2. Смотрит каталог товаров, выбирает нужные ему позиции и заносит их в корзину.
3. Сохраняет сделанный выбор.
4. Система формирует заказ и отправляет извещение на Email адрес клиента и менеджера по продажам.
5. Менеджер по продажам уточняет порядок доставки товара и высылает клиенту договор купли продажи и инвойс (счёт-фактуру).
6. Клиент, получив договор купли продажи и инвойс (счёт-фактуру), оплачивает счет.
7. Получив подтверждение прохождения оплаты, менеджер по продажам формирует заказ и высылает его на склад для исполнения.
8. Склад, формирует партию товара и отправляет ее клиенту. Сообщение об отправке передается менеджеру по продажам.
9. Менеджер по продажам фиксирует данные в системе и высылает на Email адрес клиента извещение об отправке товара.

Данные по п.п. 3, 4, 5, 6, 8, 9 фиксируются в системе для отслеживания статуса заказа. Клиент, менеджер по продажам и руководитель отдела продаж могут в любой момент проверить статус заказа.

Построим диаграммы IDEF0 бизнес процесса продаж. Выберем точку зрения менеджера по продажам. Диаграмма уровня А-0 изображена на рис.2.1.

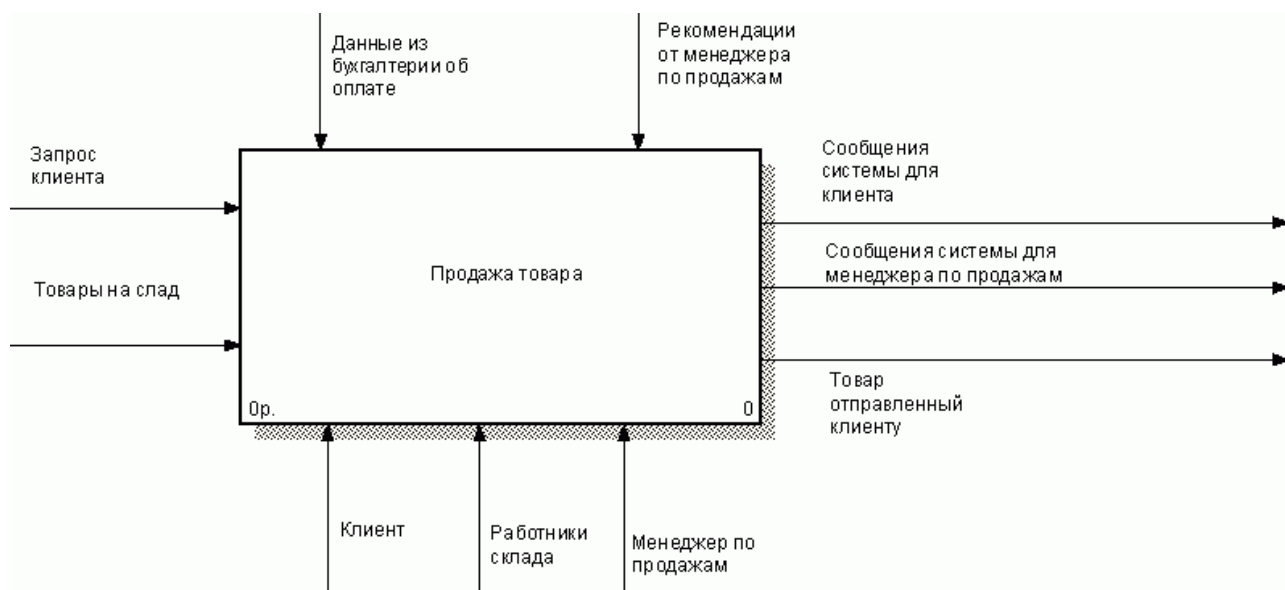


Рисунок 2.1 – Диаграмма А - 0 «Продажа товаров»

Бизнес процесс продажи состоит из следующих работ:

- вход на сайт и авторизация;
- заказ товара клиентом;
- обработка заказа менеджером;
- отправка заказа клиенту.

Данные работы изображены на рис.2.2.

Произведем декомпозицию элементов первого уровня.

Вход на сайт и авторизацию выделим в отдельный процесс.

Диаграмма заказа товара клиентом изображена на рис.2.3.

Заказ товара клиентом в свою очередь состоит из следующих действий:

- формирование заказа;
- подтверждение заказа;
- мониторинг состояния заказа;
- получение заказанных товаров.

Данные работы изображены на рис.2.3.

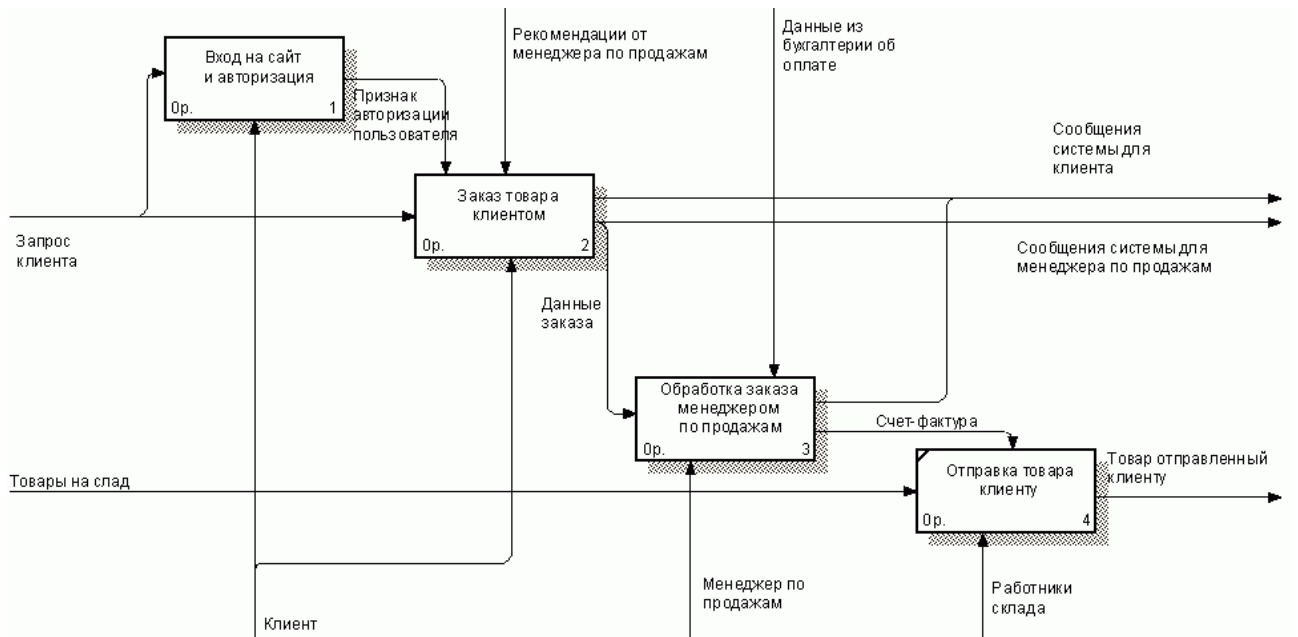


Рисунок 2.2 – Диаграмма А0 «Продажа товаров»

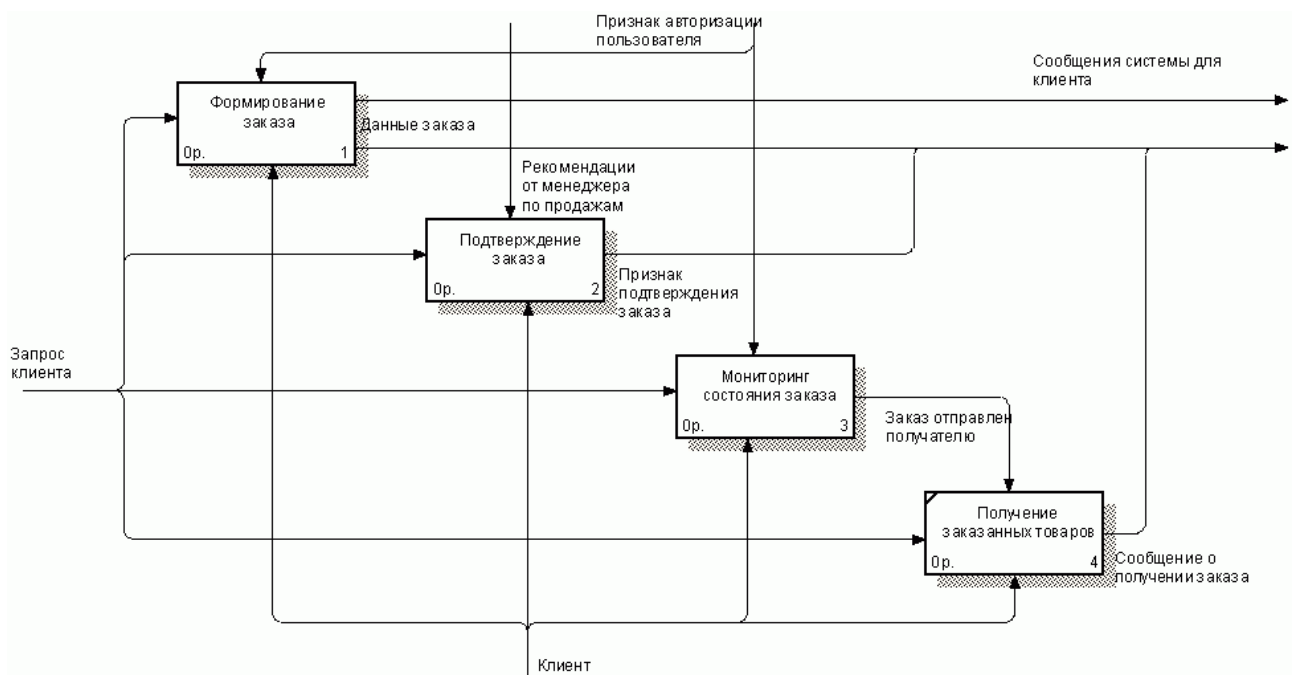


Рисунок 2.3 – А2 Диаграмма «Заказ товара клиентом»

Формирование заказа в свою очередь состоит из следующих действий:

- выбор товара;
- занесение товара в корзину;
- сохранение сформированной корзины заказа.

Данные работы изображены на рис.2.4.

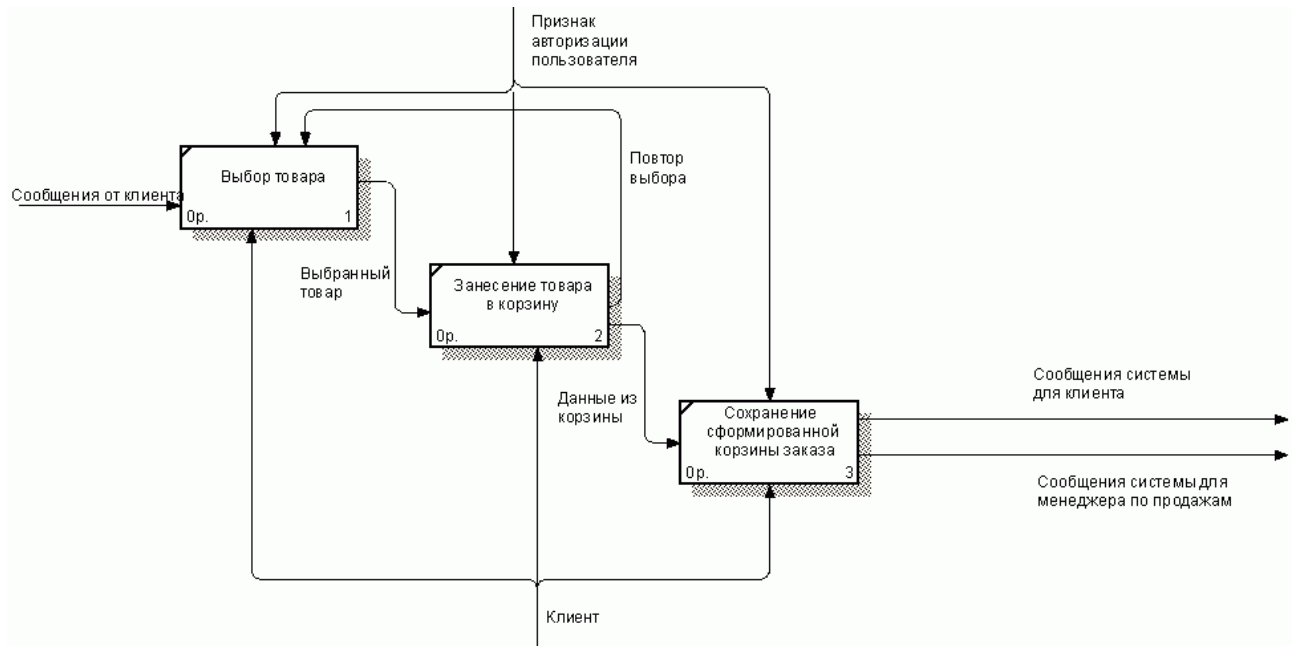


Рисунок 2.4 – Диаграмма A21 «Формирование заказа»

Подтверждение заказа состоит из следующих действий:

- проверка клиентом счет-фактуры;
- оплата клиентом счет-фактуры.

Данные работы изображены на рис.2.5.

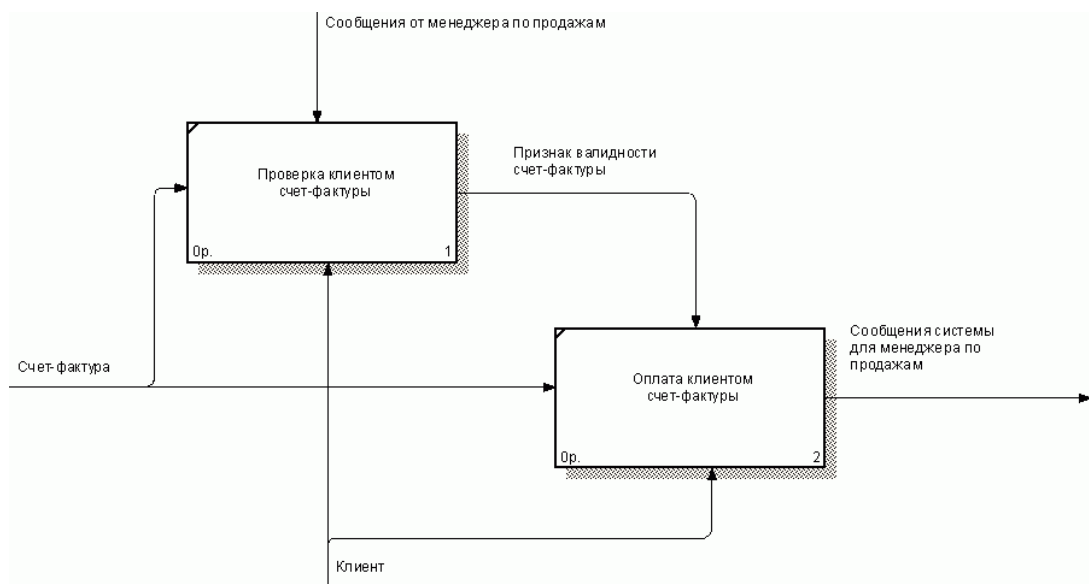


Рисунок 2.5 – Диаграмма A22 «Подтверждение заказа»

Обработка заказа менеджером состоит из следующих действий:

- проверка наличия товара и стоимости его транспортировки;
- формирование договора купли-продажи и счет-фактуры;
- проверка полноты оплаты;
- формирование документов на поставку для склада.

Данные работы изображены на рис.2.6.

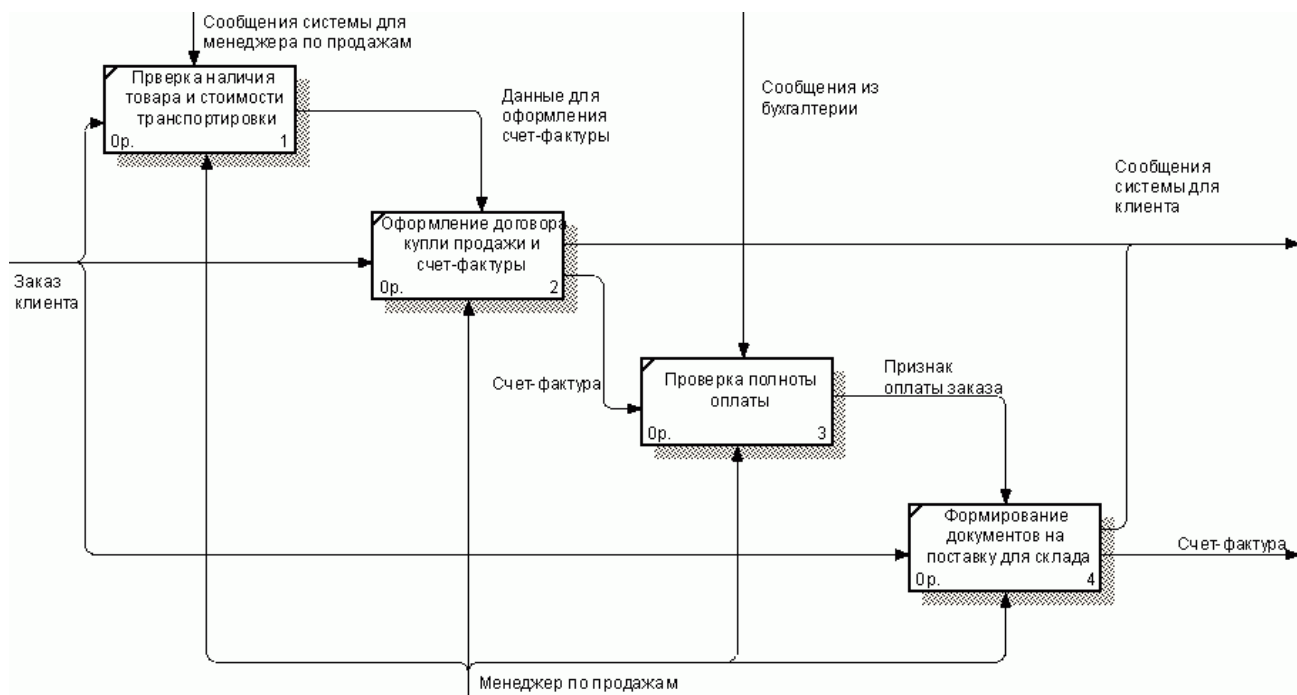


Рисунок 2.6 – АЗ Диаграмма «Обработка заказа менеджером»

Построенные диаграммы отображают предлагаемый порядок исполнения бизнес-процессов. Особенностью этих бизнес-процессов являются:

- возможность клиента осуществлять заказы через интернет, без личного присутствия в офисе фирмы;
- возможность клиента получать необходимые документы для осуществления покупок посредством интернет;
- возможность клиента осуществлять мониторинг исполнения заказа;
- возможность менеджера по продажам поддерживать в актуальном состоянии базу клиентов и их заказов;
- возможность менеджера по продажам осуществлять анализ заказов.

2.2 Проектирование логики (алгоритма) работы приложений

Исходя из требований к системе, система должна реализовать следующие действия:

- идентификацию пользователей при входе в систему;
- авторизацию пользователей в соответствии с их ролями в системе;
- регистрацию новых пользователей;
- предоставить Покупателю возможность выбрать товар и сформировать заказ;
- предоставить возможность Менеджеру и Продавцу контролировать наличие договоров купли - продажи;
- предоставить возможность Менеджеру и Продавцу фиксировать прохождение оплаты товара;
- предоставить возможность Менеджеру и Продавцу формировать заказ к отгрузке и фиксировать факт отгрузки;
- предоставить возможность Покупателю, Менеджеру и Продавцу контролировать состояние заказов;
- предоставить возможность Администратору контролировать состояние базы данных.

Произведем моделирование потоков данных в соответствии с методологией DFD, используя нотацию Гейна-Сарсона.

Идентификация и авторизация пользователей приведена на рис 2.7.

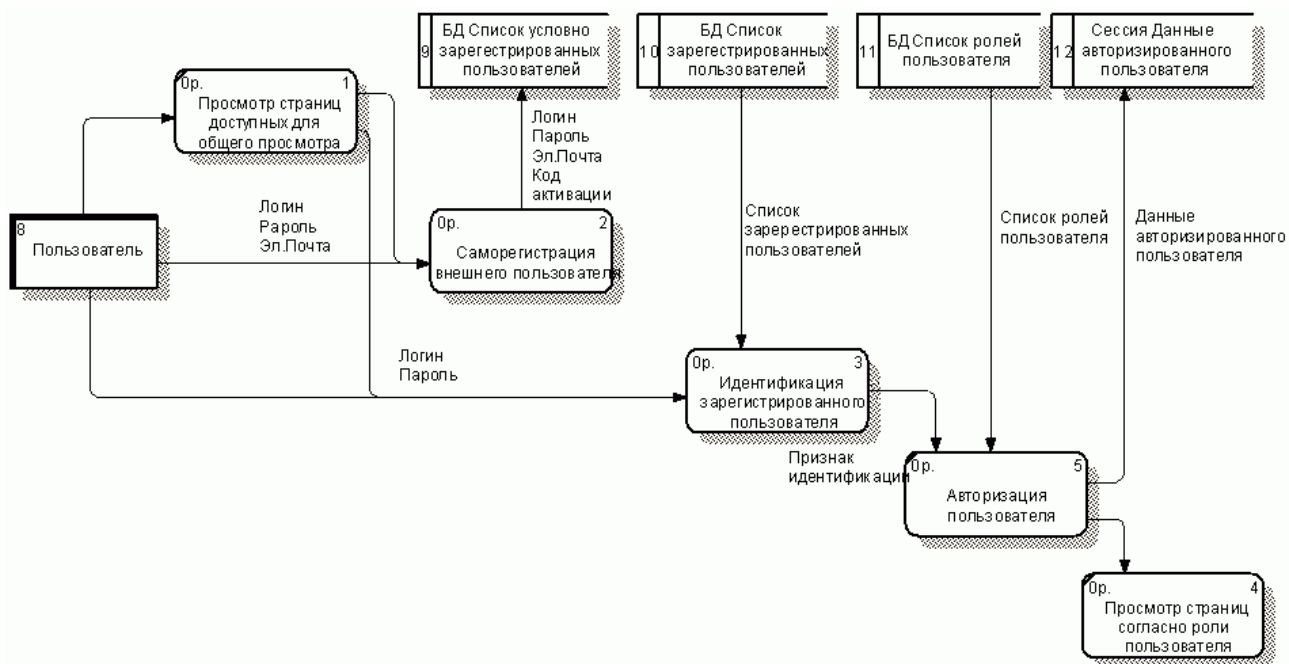


Рисунок 2.7 – Диаграмма А1 Вход на сайт и авторизация

В соответствии с требованиями заказчика, регистрация новых пользователей осуществляется по двум вариантам:

– регистрация Сотрудников, Поставщиков и Перевозчиков осуществляется Администратором системы;

– регистрация Покупателей с целью фиксации всех возможных потенциальных клиентов осуществляется лично Покупателем в два этапа: первичная регистрация потенциального клиента и полная регистрация Покупателя через Интернет.

Процесс саморегистрации Покупателя показан на рис. 2.8.

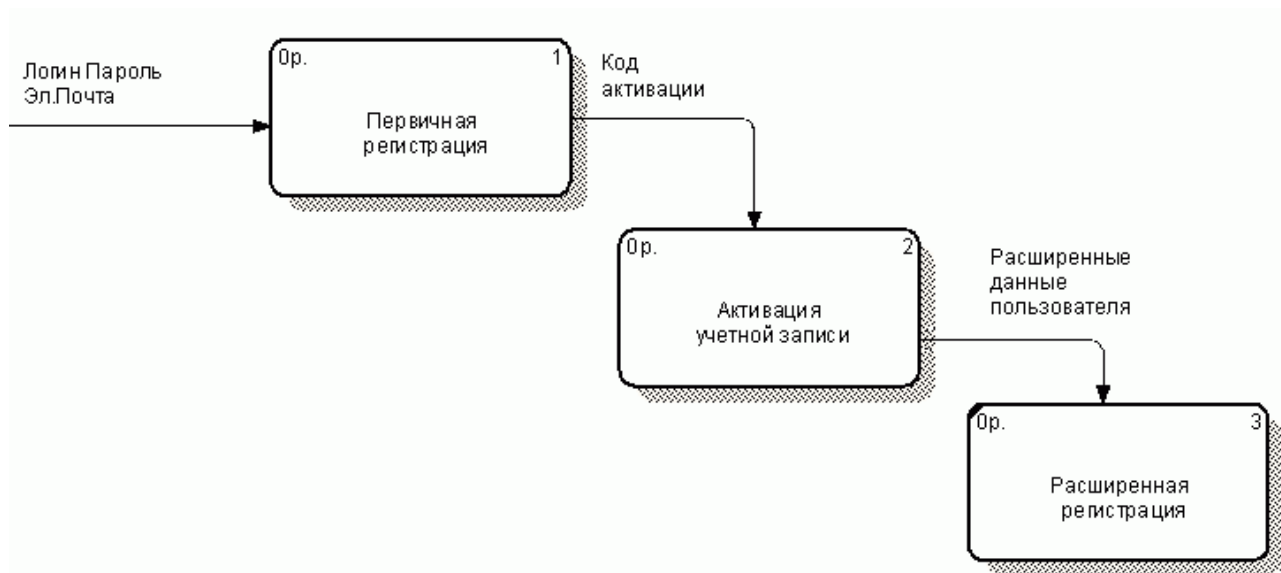


Рисунок 2.8 – Диаграмма A12 Регистрация внешнего пользователя

Первичная регистрация пользователя осуществляется путем ввода пользователем таких данных, как логин, пароль, и e-mail. При этом система проверяет отсутствие в базе пользователя с указанным логином, в случае успешного прохождения проверки генерирует индивидуальный ключ авторизации, записывает эти данные в таблицу условно зарегистрированных пользователей и задает временную метку, ограничивающую время жизни записи (время, в течении которого пользователь должен подтвердить регистрацию) до трех суток. Затем система отправляет на указанный пользователем e-mail письмо с указанием ссылки, по которой надо перейти для активизации учетной записи. Данный процесс показан на рис. 2.9.

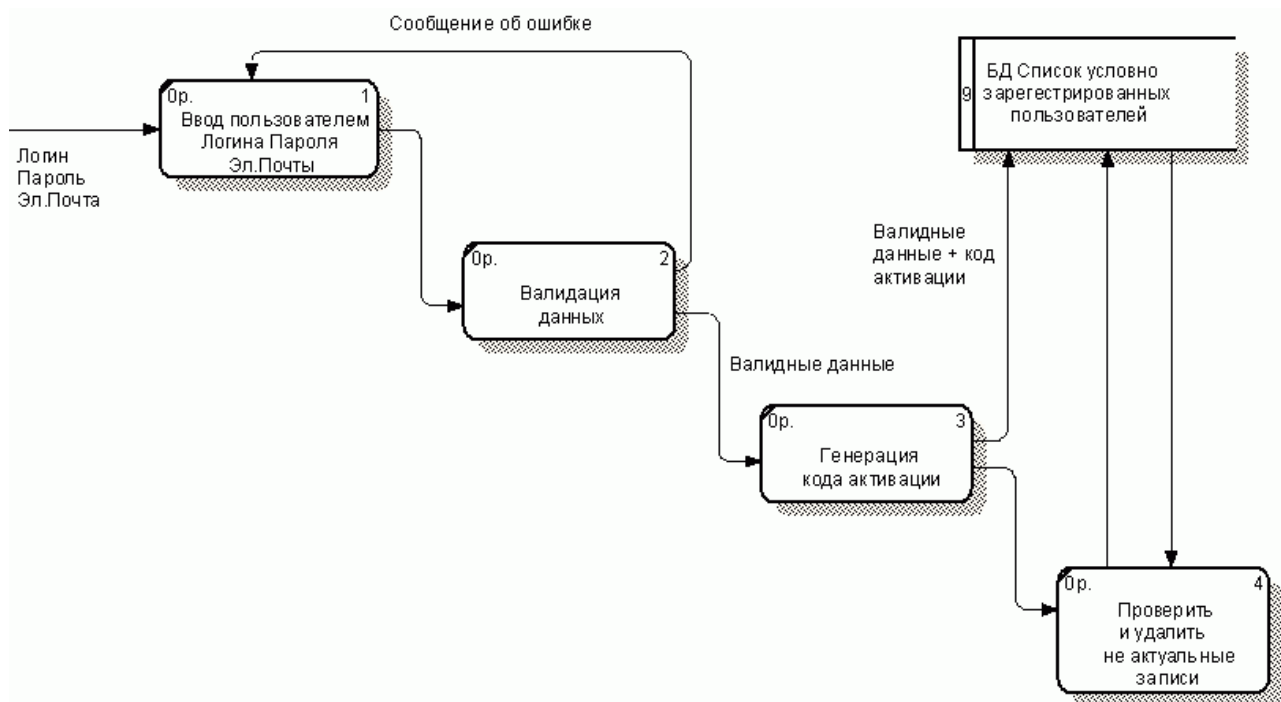


Рисунок 2.9 – Диаграмма A121 Первичная регистрация

Клиент, получив по электронной почте сообщение со ссылкой подтверждения, должен перейти по этой ссылке не позднее, чем через трое суток с момента первичной регистрации для подтверждения факта регистрации. Данный процесс показан на рис. 2.10.

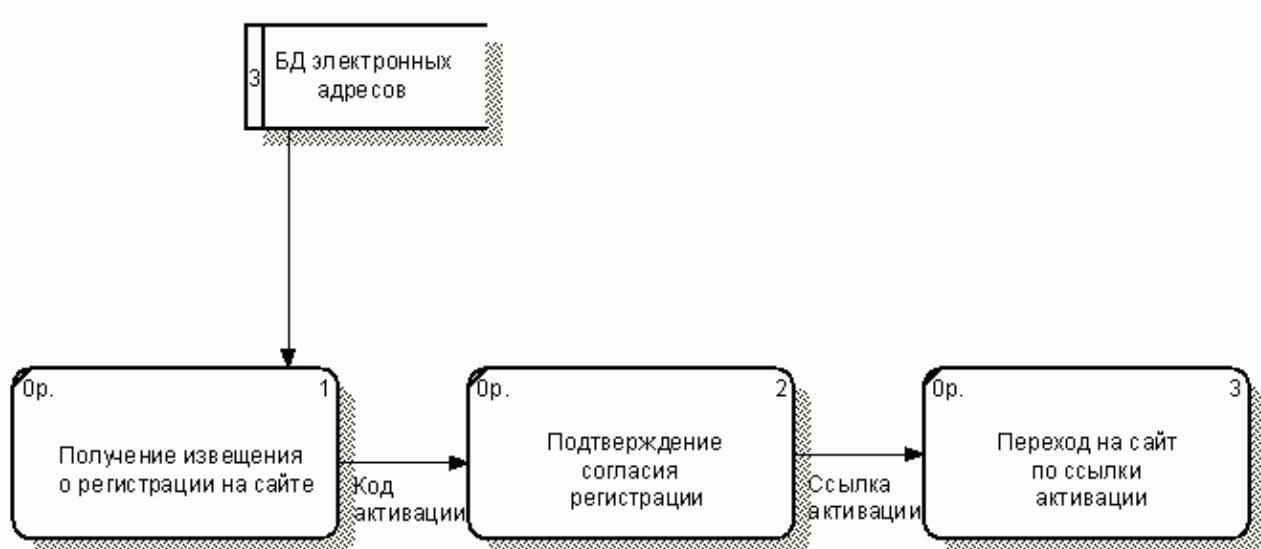


Рисунок 2.10 – Диаграмма A122 Активация учетной записи

Зарегистрированный клиент имеет возможность открыть прайс-лист компании, выбрать товары по категориям и занести их в корзину в

необходимом количестве. После формирования корзины клиент сохраняет заказ и система отправляет на электронный адрес клиента подтверждение заказа, а также формирует сообщение для менеджера о проведенном клиентом заказе. Менеджер по продажам проверяет наличие выбранного товара на складе, оценивает стоимость доставки товара в соответствии с прайс-листами компаний-перевозчиков, проверяет наличие договора купли-продажи с клиентом. Если договор с клиентом оформлен, то менеджер по продажам формирует счет-фактуру и высылает ее клиенту. Если клиент первый раз обратился в компанию и договора купли-продажи с ним еще нет, то менеджер по продажам организует оформление и подписание такого договора.

Процесс формирования заказа зарегистрированным клиентом приведен на рис.2.11.

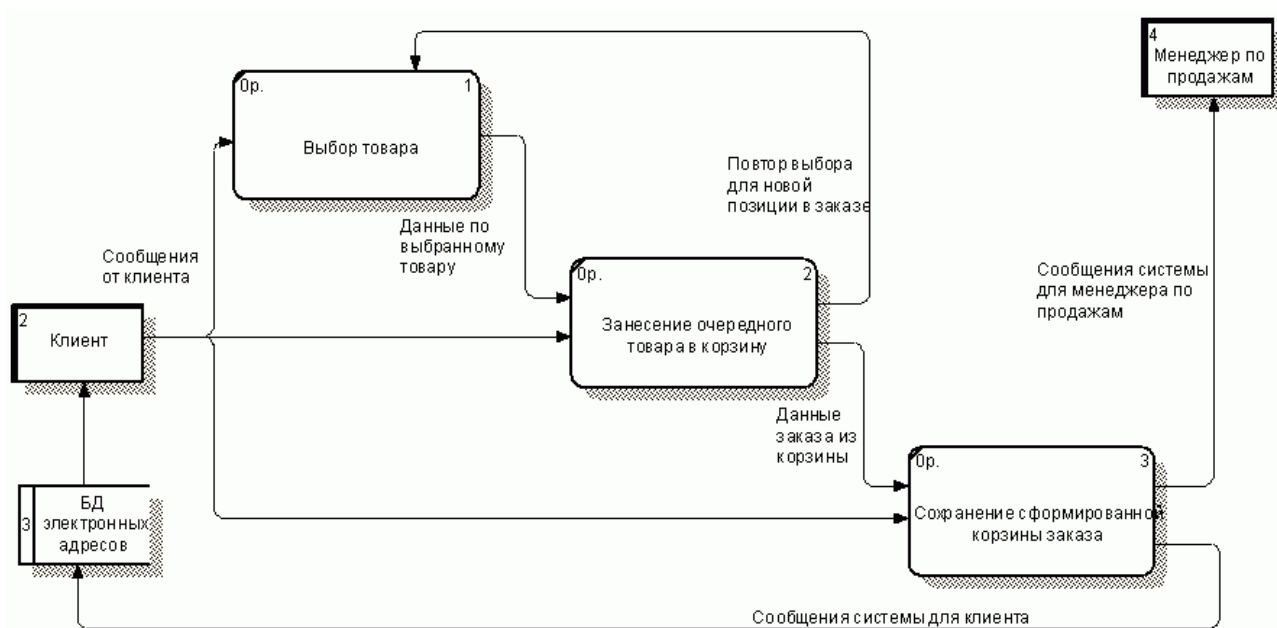


Рисунок 2.11 – Диаграмма A21 Формирование заказа

При получении счет-фактуры клиент проверяет ее полноту и правильность и осуществляет оплату. Этот процесс показан на рис.2.12.

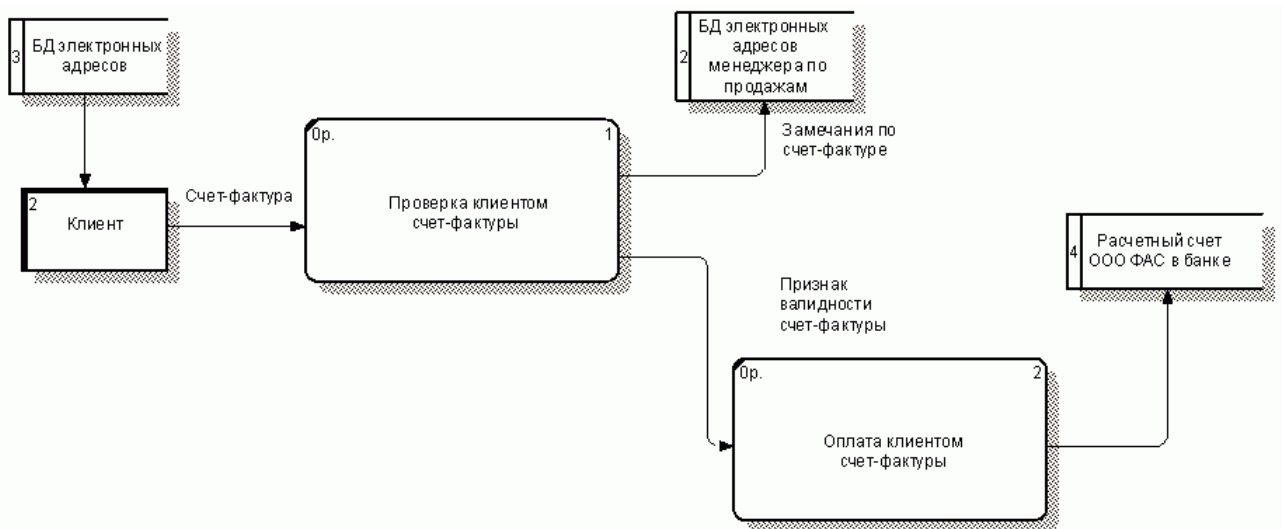


Рисунок 2.12 – Диаграмма A22 Подтверждение заказа

Действия менеджера по обработке заказа приведены на рис.2.13.

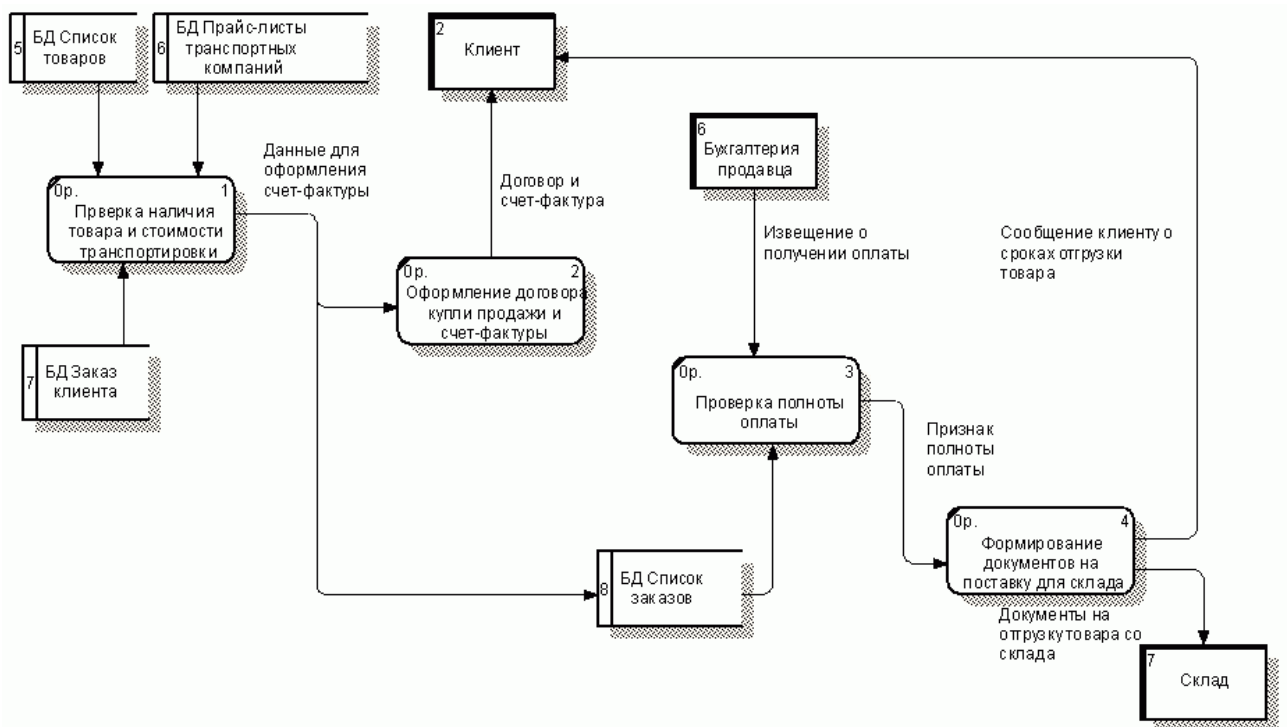


Рисунок 2.13 – Диаграмма A3 Обработка заказа менеджером по продажам.

2.3 Разработка архитектуры CRM-системы

При проектировании информационной системы необходимо выбрать ее архитектуру, определяющую модель работы приложения с базой данных.

Различают следующие основные архитектуры информационных систем [1]:

- системы с локальной базой данных;
- архитектура “файл-сервер”;
- архитектура “клиент-сервер”.

В системах с локальной базой данных, как понятно из названия, база данных располагается на том же компьютере, что и приложение. Работа с базой данных происходит в однопользовательском режиме. Данное ограничение не позволяет нам использовать эту архитектуру при разработке нашей системы.

Архитектура “файл-сервер” определяет, что база данных находится на файловом сервере сети. Системы с этой архитектурой могут быть многопользовательскими. При каждом запросе от приложения к базе данных, приложение получает всю таблицу и выполняет необходимые операции с ее данными. Фактически, работа происходит с локальной копией базы данных (или ее частью), которая обновляется при необходимости. При этом изменения, которые каждый пользователь вносит в БД, могут быть до определенного момента неизвестны другим пользователям, что делает актуальной задачу систематического обновления данных на компьютере пользователя из реальной БД.

В системах с архитектурой “клиент-сервер” функции разделяются между клиентом (пользователем) и сервером. Работа основывается на формировании запроса к серверу БД (на котором установлена СУБД), обработке запроса системой управления базой данных и отправке результата выполнения запроса обратно клиенту.

К основным недостаткам архитектуры “файл-сервер” относят:

- проблемы с синхронизацией БД;
- низкая скорость работы и высокая нагрузка на сеть;
- незащищенность от получения или изменения данных злоумышленником, имеющим доступ к файл-серверу;

– подверженность ошибкам из-за отсутствия единой системы контроля и работы с БД.

В свою очередь архитектура “файл-сервер” имеет иногда значимое преимущество – низкие требования к серверу.

В настоящее время архитектура “файл-сервер” практически не используется по приведенным выше причинам, и вследствие появления, развития и распространения архитектуры “клиент-сервер”, лишенной данных недостатков.

Таким образом, при разработке нашей автоматизированной системы будет использоваться архитектура “клиент-сервер”.

В свою очередь, архитектура “клиент-сервер” разделяется на [1]:

- двухзвенная архитектура “клиент-сервер”;
- трехзвенная архитектура “клиент-сервер”;
- многозвенная архитектура “клиент-сервер”.

Классической моделью является двухзвенный вариант архитектуры, описанный выше, когда приложение передает запросы СУБД и получает от нее результаты запросов. В трехзвенной архитектуре “клиент-сервер” добавляется промежуточное звено между слоем, с которым непосредственно работает пользователь, и слоем базы данных. Примером трехзвенной архитектуры “клиент-сервер” может быть www (браузер (клиент) – web-узел база данных) или приложение, работающее с базой, данной на сервере в Интернет через web-сервис. В многозвенной архитектуре “клиент-сервер” клиент (приложение) может обращаться к нескольким промежуточным слоям (службам).

В нашей системе мы будем использовать трехзвенную архитектуру клиент-сервер, представленную на рис.2.14.

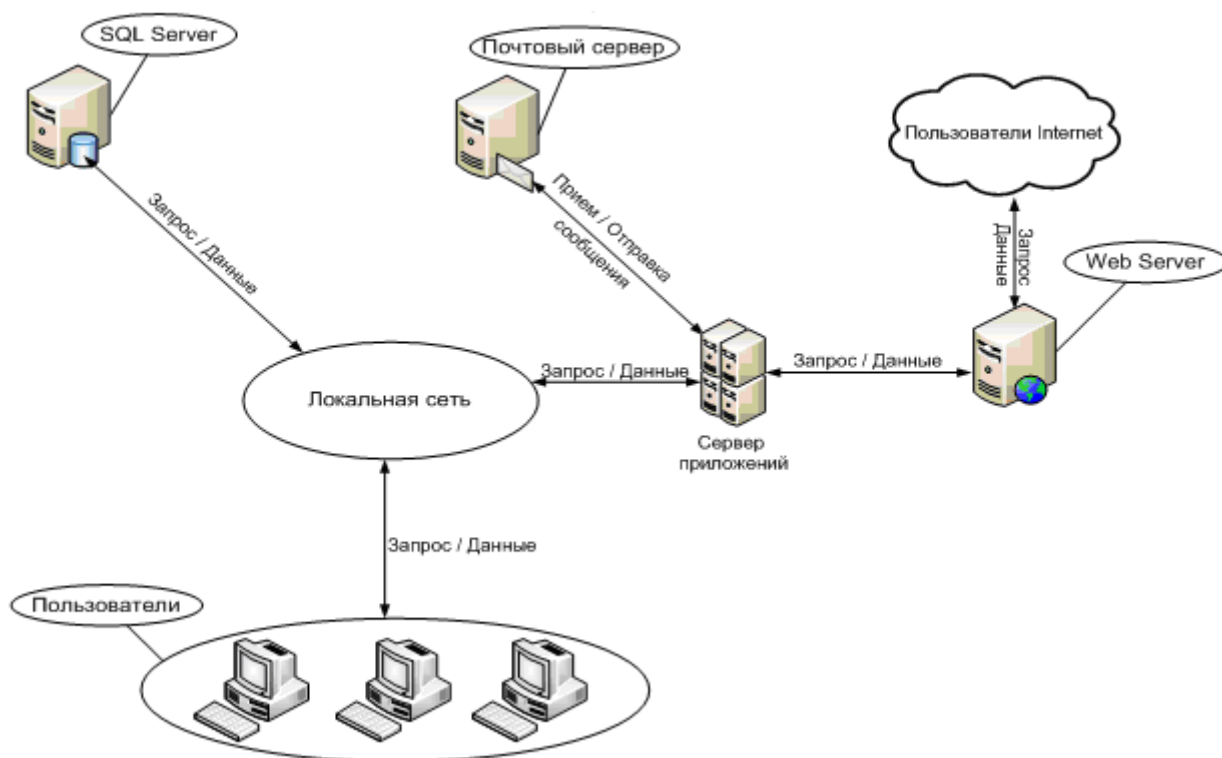


Рисунок 2.14 Архитектура CRM-системы

2.4 Разработка подсистемы хранения данных

2.4.1 Этапы проектирования подсистемы хранения данных

При проектировании и разработке БД выделяется несколько этапов моделирования, при помощи которых происходит переход от предметной области к конкретной реализации БД средствами конкретной СУБД.

Можно выделить следующие этапы:

- выбор языка представления т.н. "семантической" модели для фиксации сведений о предметной области, их последующего анализа и синтеза модели БД;

- анализ собранных сведений о предметной области, классификация, формализация и интеграция элементов описания предметной области;

- синтез концептуальной модели БД: проектирование целостной концептуальной схемы БД на выбранном языке семантического моделирования. Модель предметной области - это знания о предметной области. Текстовый способ представления модели предметной области крайне

неэффективен. Гораздо более информативными и полезными при разработке БД являются описания предметной области, выполненные при помощи специализированных графических нотаций. Модель предметной области описывает скорее процессы, происходящие в предметной области и данные, используемые этими процессами;

- выбор конкретной модели данных и СУБД для реализации БД;
- проектирование логической модели БД для выбранной СУБД.

Логическая модель описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью. Логическая модель данных является начальным прототипом будущей базы данных. Логическая модель строится в терминах информационных единиц, но без привязки к конкретной СУБД. Более того, логическая модель данных необязательно должна быть выражена средствами именно реляционной модели данных. Основным средством разработки логической модели данных в настоящий момент являются различные варианты ER-диаграмм (диаграммы сущность-связь). Решения, принятые на этапе разработки модели предметной области, определяют некоторые границы, в пределах которых можно развивать логическую модель данных, в пределах этих же границ можно принимать различные решения;

- разработка физической модели БД. Физическая модель данных описывает данные средствами конкретной СУБД. Отношения, разработанные на стадии формирования логической модели данных, преобразуются в таблицы, атрибуты становятся столбцами таблиц, для ключевых атрибутов создаются уникальные индексы, домены преобразуются в типы данных, принятые в конкретной СУБД [4].

Ограничения, имеющиеся в логической модели данных, реализуются различными средствами СУБД. При этом решения, принятые на этапе логического моделирования определяют некоторые границы, в пределах которых можно развивать физическую модель данных. Точно также, в пределах этих границ можно принимать различные решения. Например, отношения,

содержащиеся в логической модели данных, должны быть преобразованы в таблицы, но для каждой таблицы можно дополнительно объявить различные индексы, повышающие скорость обращения к данным. Здесь очень многое зависит от конкретной СУБД.

2.4.2 Концептуальное проектирование CRM-системы

Определим основные понятия предметной области:

Поставщик - организации или физические лица, с которыми заключаются договора на поставку товара.

Покупатель - организации или физические лица, подающие заказы на приобретение того или иного товара.

Договор поставки - соглашение, по которому одна сторона (поставщик) обязуется передать в обусловленный срок другой стороне (покупателю) товары для использования в хозяйственных целях.

Заказ - перечень товаров, которые хочет приобрести заказчик. Атрибуты заказчика. Способ доставки. Отображается на две сущности: Заказ и Перечень товаров в заказе.

Поставка - перечень товаров, которые закуплены у поставщика. Атрибуты поставщика. Отображается на две сущности: Поставка и Перечень товаров в поставке.

Товар - любая вещь, не ограниченная в обороте, свободно отчуждаемая и переходящая от одного лица к другому по договору купли-продажи [17]. В нашем случае - авто комплектующие.

Категория товара - группа товаров, объединенная по какому либо признаку.

Сотрудник - служащий организации, принимающий участие в бизнес процессах организации.

Пользователь - физическое лицо, исполняющее в системе обязанности в соответствии с заданной ролью.

Гость - физическое лицо, начавшее процесс саморегистрации, но еще не завершившее его.

Перевозчик - способ доставки заказа покупателю, либо компания грузоперевозчик.

Роль в системе - группа действий, которые пользователь может исполнять в системе. Каждому пользователю в системе поставлено в соответствие одну или несколько ролей. Как правило, пользователю присваивается одна роль. Пользователи соотносятся с одной из сущностей: поставщик, покупатель, сотрудник и перевозчик. Сущностям поставщик, покупатель и перевозчик соответствуют роли: Поставщик (supplier), Покупатель (buyer, client, customer), Перевозчик (shipper). Пользователи, относящиеся к сущности сотрудник, имеют разные роли: Менеджер по закупкам или закупщик (buyer), Менеджер по продажам или продавец (seller), Начальник отдела продаж или управляющий (manager), администратор системы (admin). Эти пользователи могут иметь несколько ролей в системе. Кроме этого, в системе существует роль Гость (guest). Это пользователь, введенный в систему, но регистрация которого еще не завершена и ему еще не присвоена какая либо иная роль. Этот пользователь считается зарегистрированным в системе, но его права ни как не отличаются от прав незарегистрированных пользователей. Эта роль нужна для реализации двухступенчатой системы саморегистрации пользователей.

Моделирование предметной области базируется на использовании графических диаграмм.

Используем наиболее популярную семантическую моделей данных - модель "Сущность-Связи" (ER-модель).

В этом случае наша модель данных в нотации Баркера примет следующий вид, показанный на рис.2.15, где в таблицах отображены только поля, по которым построены первичные внешние ключи. Таблица Гость предназначена для временного хранения информации об условно-зарегистрированных пользователях, и поэтому не имеет связей с другими таблицами.

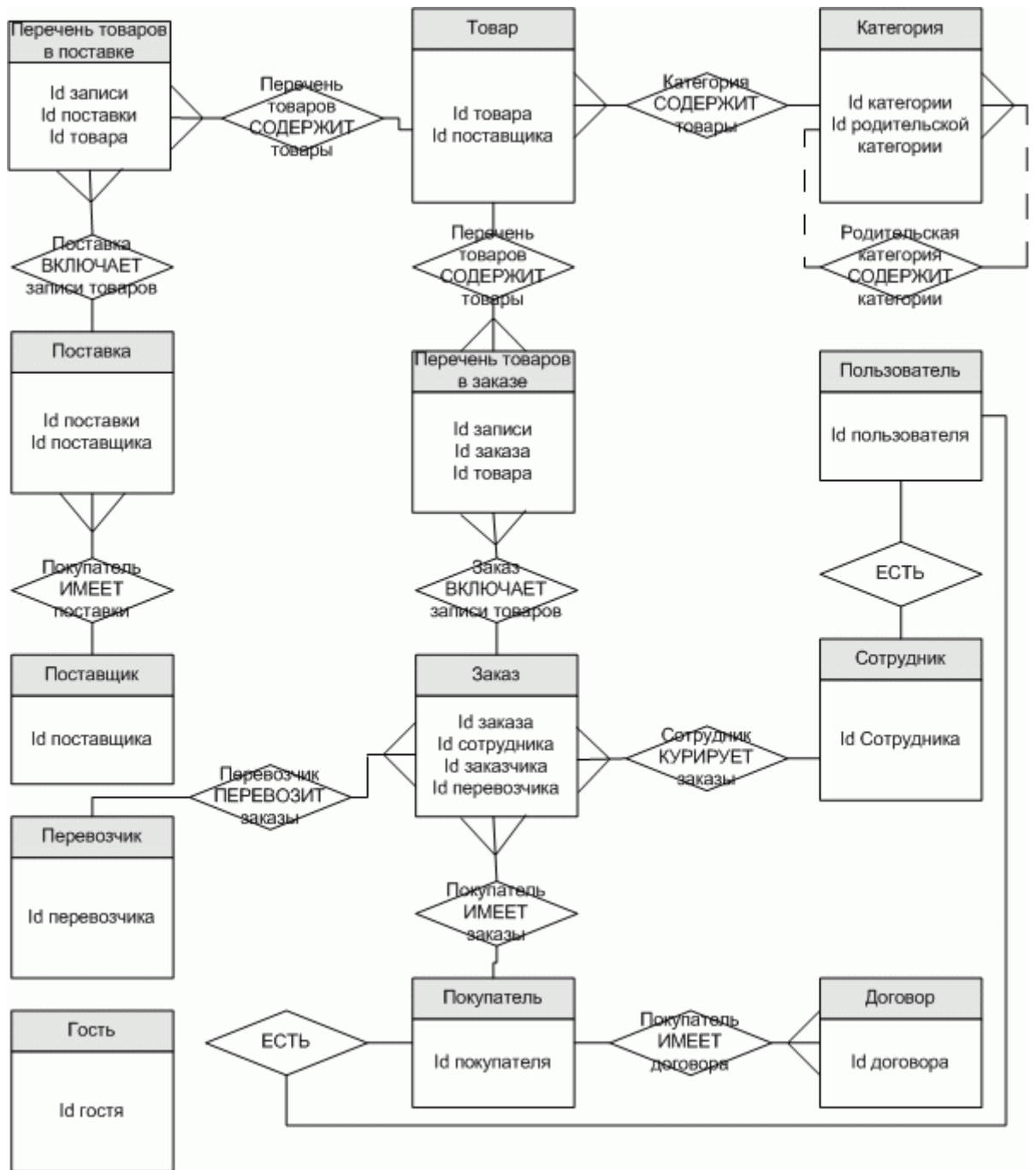


Рисунок 2.15 – Концептуальная модель данных предметной области

Существуют различные модели данных: иерархическая, сетевая, объектная, реляционная и т.д.

Иерархическая модель данных строится по принципу иерархии типов объектов, то есть один тип объекта является главным, а остальные, находящиеся на низших уровнях иерархии, – подчиненными.

В сетевой модели данных каждый узел (набор) базы данных взаимодействует с другими узлами посредством сложной системы связей. В

сетевой модели главный объект обозначается термином “владелец набора”, а подчиненный – термином “член набора”. Один и тот же объект может одновременно выступать и в роли владельца, и в роли члена набора.

В объектной модели данных в базе хранятся не только данные, но и методы их обработки в виде программного кода.

В реляционной модели база данных состоит из ряда неупорядоченных таблиц; ими можно оперировать посредством непроцедурных операций, результатом выполнения которых также являются таблицы.

База данных, спроектированная по принципам реляционной модели, обладает многими достоинствами, среди которых можно выделить следующие:

- эффективность процедур добавления, обновления и удаления данных;
- эффективность процедур выборки данных, их суммирования и генерации отчётов;
- предсказуемость поведения базы данных благодаря хорошо продуманной модели;
- так как большая часть информации хранится в базе данных, а не в приложении, БД является самодокументированной;
- изменения в структуре БД и в структуре таблиц легко выполнимы.

В данной работе для проектирования БД выберем реляционную модель данных, как более гибкую и простую в понимании и разработке.

2.4.3 Разработка логической модели данных

Логическая модель данных описывает понятия предметной области и их взаимосвязи и является прототипом будущей базы данных. Логическая модель разрабатывается в терминах информационных понятий, но без какой-либо ориентации на конкретную СУБД.

Произведена нормализация базы данных до третьей нормальной формы, определены ключевые поля, связи между таблицами, типы и мощности связей. Во избежание создания сложных первичных ключей, состоящих из нескольких

полей, для ряда сущностей созданы псевдо-первичные ключи на основе инкрементных полей.

В графическом виде логическая модель показана на рис.2.16.

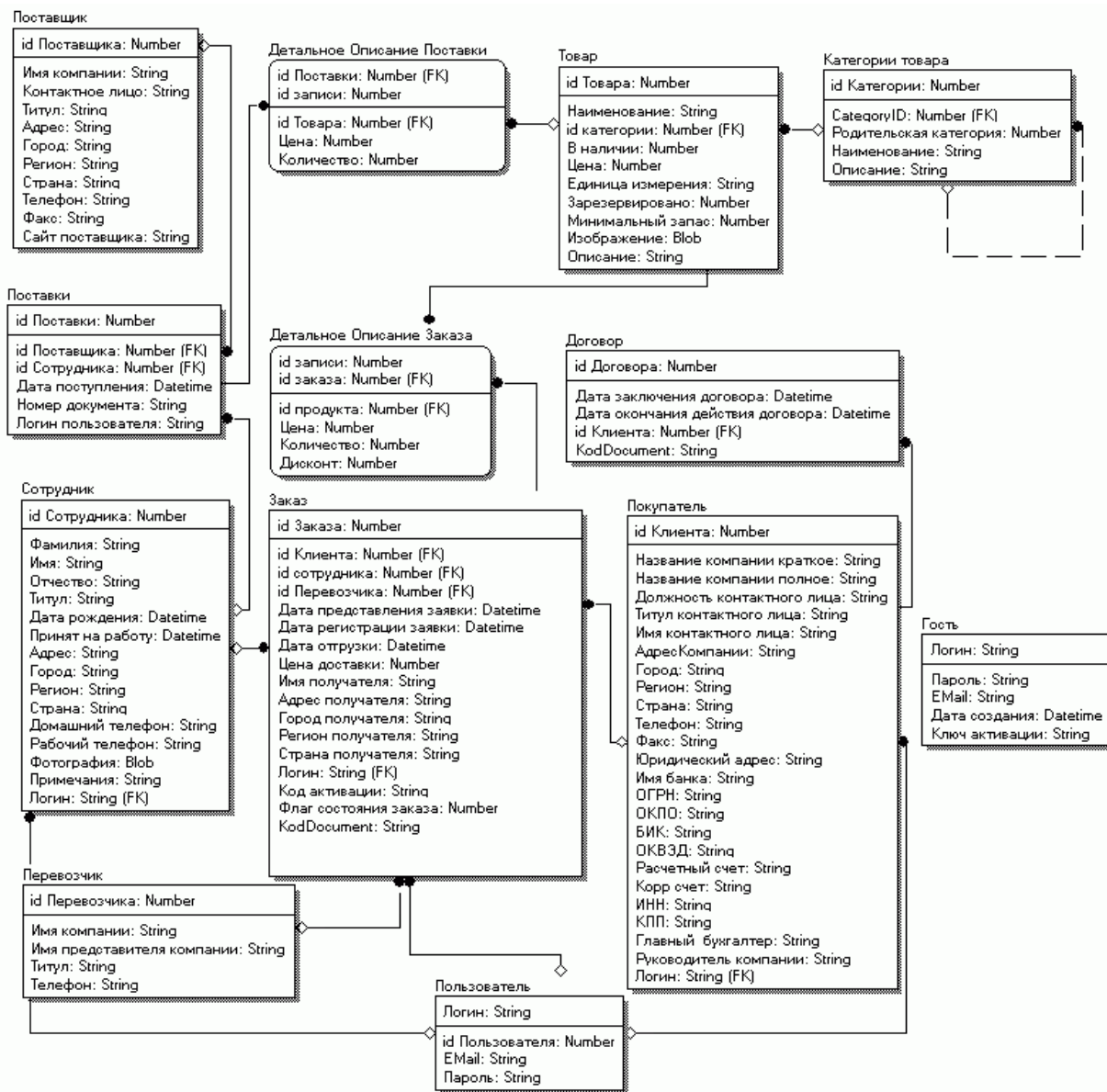


Рисунок 2.16 – Логическая модель данных

Для рассматриваемой предметной области имеют значения сущности, атрибуты и связи сущностей.

Обобщая материалы главы можно сформулировать следующие итоги:

– на основании анализа существующих бизнес - процессов компании разработан новый регламент осуществления бизнес - процесса купли - продажи для компании ООО «ФАС»;

– разработаны диаграммы вариантов использования, которые на более низком уровне абстракции описывают порядок действий пользователей системы с различными ролями в системе. Определен порядок регистрации новых пользователей в системе;

– определен перечень возможных действий пользователей в соответствии с их ролями в системе;

– для конкретизации порядка взаимодействия различных пользователей системы разработаны диаграммы деятельности процесса саморегистрации и бизнес - процесса купли-продажи;

– выбран вариант архитектуры разрабатываемой CRM системы. В системе будем использовать трехзвенную архитектуру клиент-сервер;

– осуществлены начальные этапы проектирования БД. Разработана концептуальная модель данных предметной области в нотации Баркера. Обосновано и принято решение при проектировании БД использовать реляционную модель. На основании концептуальной модели разработана логическая модель базы данных.

Глава 3 РЕАЛИЗАЦИЯ, ТЕСТИРОВАНИЕ И ВНЕДРЕНИЕ WEB САЙТА

3.1 Выбор СУБД и средств разработки CRM-системы

При рассмотрении платформ для создания WEB-приложений можно выделить два основных существующих подхода [18]:

- непосредственная обработка запросов и формирование ответов;
- встраивание программного кода в шаблоны HTML страниц.

Первый подход предоставляет наибольшие возможности по управлению обработкой и повышению производительности. Он предусматривает передачу всех данных о запросе непосредственно исполняемому коду, который может как сформировать ответ со страницей для пользователя, так и открыть на передачу поток двоичных данных, например для передачи изображения. Однако при таком подходе все данные для передачи формируются программным путем, что замедляет разработку простых страниц и усложняет взаимодействие между верстальщиком и программистом. Примерами этого подхода служат технологии CGI, Java Servlets.

Второй подход использует шаблоны страниц пользователя, оформленные особым образом, что позволяет вставлять в них участки программного кода. Этот подход особенно эффективен при создании простых приложений, основная информация в которых статична, а динамическая информация может быть сгенерирована простыми программными конструкциями. При разработке сложных программных систем этот вариант усложняет взаимодействие между компонентами и затрудняет реализацию сложной архитектуры. Также он менее эффективен по производительности и ограничивает возможности по реализации сложных страниц. Примерами этого подхода служат очень популярные на данный момент технологии PHP, ASP, JSP.

Помимо различного подхода к генерации страниц платформы разработки в разной степени удовлетворяют современным требованиям, выдвигаемым при

создании сложных Web систем. Наиболее важные из этих требований, наличие которых делает систему привлекательной для использования, приведены ниже:

- язык реализации;
- производительность, масштабируемость;
- возможности расширения и интеграции;
- простота использования, наличие средств разработки;
- наличие необходимых программных библиотек.

Рассмотрим платформы для создания WEB-приложений. CGI не входит в обзор, поскольку является неудобной в использовании, и имеющей низкую эффективность, а расширения серверов слишком сильно привязаны к конкретным программным продуктам. По схеме обработки запросов платформы распределяются следующим образом:

- PHP - шаблоны. При выполнении на Web - сервере Apache интерпретатор может являться расширением сервера (в экспериментальном режиме IS);

- Java Servlets - накапливание процессов для каждого сервлета;

- JSP - шаблоны. При обработке выполняется их предкомпиляция в Java Servlets, позволяя использовать схему накапливания процессов;

- ASP.NET - шаблоны. Используется схема предварительной компиляции, а не интерпретации кода. В результате используется расширение Web - сервера IIS. Могут использоваться и низкоуровневые обработчики.

Основные оценочные характеристики платформ сравним в сводной табл.3.1. Где: "0" - полное отсутствие поддержки, "1" - недостаточная поддержка, "2" - поддержка не в полном объеме, и "3" - полная поддержка. Для сравнительных характеристик, таких как язык реализации или производительность, оценки соответствуют степени превосходства технологии.

Таблица 3.1 – Оценочные характеристики платформ

Параметры	PHP	Java Servlets	JSP	ASP .NET
Многоплатформенность	2	3	3	1
Производительность	1	2	2	3
Масштабируемость	0	3	3	3
Язык реализации	2	3	3	3
Возможности расширения и интеграции	0	3	2	3
Простота использования, наличие средств разработки	2	2	3	3
Наличие необходимых программных библиотек	3	3	3	3
Разделение дизайна и логики	2	1	2	3
Средства визуальной разработки	1	2	3	3
Возможность построения компонентной архитектуры	0	3	2	3
Итого	13	25	26	28

Основываясь на полученных результатах сравнения, для дипломной работы выбираем платформу ASP.NET.

Реализовывать технологию ASP.NET будем с помощью свободной среды разработки Visual C# Express Edition, СУБД Microsoft SQL Server Express Edition и веб-сервер IIS. Язык C# самый новый и мощный язык программирования, в том числе и Web приложений. Среда разработки Visual C# очень функциональная и самая мощная среда разработки. В процессе разработки среда предоставляет услуги встроенного сервера СУБД разработчика - SQL Express Server. Таким образом, мы выбираем мощную и лицензионно чистую среду разработки, что позволяет быстро и качественно реализовывать и внедрять программные комплексы в организациях

3.2. Выбор технологической платформы разработки

Для обоснования выбора технологической платформы разработки веб-приложений WebForms или MVC Framework, рассмотрим сильные и слабые стороны каждой из этих технологий, а также основные принципы, на которых основаны эти технологии.

3.2.1. Технология WebForms

Технология WebForms в ASP.NET была создана для того, чтобы сделать веб-разработку максимально простой для разработчика, знакомого с созданием клиентских приложений на Windows-платформе.

Кроме того, декларативный подход технологии WebForms позволяет очень просто преобразовывать классические веб-приложения для использования новейших технологий, таких как Ajax и Silverlight.

Преимущества WebForms:

- высокая скорость разработки;
- большое количество готовых компонентов;
- богатая поддержка средствами разработки;
- автоматическое управление состоянием;
- событийная модель элементов управления;
- высокая степень абстракции над html-разметкой;
- простота изучения технологии;
- возраст технологии.

Недостатки WebForms:

- связка разметки и логики страницы. Разметка страницы, определенная в ASPX-файле, жестко привязана к коду логики, определенному в code-behind-файле;
- отсутствие полного контроля над конечной разметкой страницы;
- сложность тестирования логики приложения;
- неестественная для веб-среды модель сохранения состояния. HTTP-протокол не поддерживает состояния и, при попытке это состояние добавить в

жизненный цикл приложения, приходится создавать нагромождения дополнительного слоя логики, отвечающего за работу с состоянием.

3.2.2. Технология MVC Framework

Аббревиатура MVC, давшая название MVC Framework, скрывает в себе всю суть архитектурного подхода построения приложений по принципу MVC: модель, представление и контроллер — это те компоненты, из которых состоит каждое приложение, созданное в этой парадигме.

Модель — компонент приложения, отвечающий за взаимодействие с источником данных, а также содержащий описание объектов, описывающих данные, с которыми работает приложение.

Представление — компонент, отвечающий за отображение пользовательского интерфейса.

Контроллер — компонент, содержащий логику приложения.

Важно отметить слабую связанность компонентов между собой.

Преимущества MVC Framework:

- четкое разделение логических слоев;
- полный контроль над разметкой;
- логическое разделение функциональности;
- "красивые" URL-адреса. Поскольку MVC Framework предполагает использование гибкой системы маршрутизации, пользователь получает удобные и понятные URL-адреса страниц в виде /Products/List, /Products/Edit/1 и т. п.

- прозрачный процесс обработки запроса. Жизненный цикл страницы в WebForms описывается механизмом событий, этот механизм легко расширяется и позволяет добавлять расширенную функциональность в страницу без модификации кода самой страницы;

- поддержка различных движков генерации разметки;

– полноценная поддержка всех возможностей платформы ASP.NET. Применяя MVC Framework, разработчик может использовать все возможности платформы ASP.NET.

Недостатки MVC Framework:

- более высокий "порог входа" в технологию;
- отсутствие механизма хранения состояния. MVC Framework не предлагает механизма хранения состояния, и разработчику необходимо реализовывать его самостоятельно, используя скрытые поля, cookie-файлы или хранение данных в URL;
- сложности создания библиотек компонентов;
- молодость технологии.

Основные оценочные характеристики вышеприведенных технологий сравним в сводной табл.3.2, где "0" - полное отсутствие поддержки, "1" - недостаточная поддержка и "2" - полная поддержка.

Таблица 3.2 – Набор требований для выбора технологии разработки веб-приложения

Требование	WebForms	MVC
1	2	3
Поддержка визуального редактора страниц и декларативных элементов управления	2	1
Возможность автоматического тестирования логики приложения	1	2
Возможность замены движка генерации разметки	0	2
Высокая скорость разработки	2	1
1	2	3
Простота поддержки при необходимости частого внесения изменений	1	2
Небольшая команда разработчиков, отсутствие	1	2

1	2	3
выделенных разработчиков для разработки и поддержки разных частей приложения		
Простота поисковой оптимизации готового решения	1	2
Участие разработчиков, знакомых только с созданием настольных приложений	2	0
Богатый скриптовый функционал на стороне клиента	1	2
Веб-приложение является интерфейсом для базы данных	2	0
Миграция фрагментов существующего приложения на PHP	0	2
Итого	13	16

Сравнивая достоинства и недостатки рассмотренных технологий остановим свой выбор на более молодой и перспективной технологии MVC Framework.

3.3 Реализация доступа к данным

3.3.1 Выбор библиотеки ORM

Согласно паттерну проектирования MVC основное назначение модели — это определение объекта приложения, представлением которого является вид (View) [7]. Модель и вид отделены друг от друга и взаимодействуют только в виде оповещений.

Рассмотрим механизмы ORM присутствующие в .NET.

Компания Microsoft представила свои решения в области ORM, это LINQ для SQL и Entity Framework. Работа с обеими библиотеками производится с помощью интегрированного языка запросов LINQ (Language Integrated Query - интегрированные в язык запросы).

LINQ для SQL является встроенным в .NET Framework механизмом, что отличает его от появившегося немного позднее Entity Framework. Благодаря этой встроенной поддержке и удобным инструментам мэппинга базы данных, LINQ для SQL получил очень широкое распространение. Способствовало этому также простота работы с ORM и низкий порог вхождения для любого разработчика.

Microsoft спустя некоторое время анонсировала новую разработку ORM, Entity Framework.

Entity Framework — это ORM, в которой присутствуют все те вещи, которых не было в LINQ для SQL. Но, вместе со значительно улучшенным функционалом, Entity Framework стала более требовательной к разработчику. Так, в ней пропали некоторые механизмы LINQ для SQL, например, механизм ленивой загрузки (Lazy Loading) данных.

Учитывая относительную простоту работы с LINQ и включение ее в состав .NET, а также исходя из проектируемой сложности БД остановим свой выбор на LINQ для SQL и, соответственно, в качестве СУБД будем использовать MS SQL.

3.3.2 Разработка физической модели БД

Преобразуем логическую модель базы данных в физическую модель базы данных для СУБД MS SQL. Переименуем таблицы и поля таблиц латиницей. Зададимся типами полей в соответствии с существующими типами полей СУБД MS SQL и создадим индексы. Преобразование логической модели в физическую модель базы данных для СУБД MS SQL покажем на примере преобразования таблицы Категории товаров. Так, в данной таблице поле «номер категории» целое, автоинкрементное и является первичным ключом, поэтому данным параметрам соответствуют следующие характеристики MS SQL, соответственно: имя поля - CategoryID, тип - INT, свойство - IDENTITY, ключ - PRIMARY KEY. Поле «родительская категория» целое, не должно быть пустым и используется в качестве внешнего ключа, что задается следующими

параметрами MS SQL, соответственно: имя – ParentCategoryID, тип - INT, свойство - NOT NULL, ключ - FOREIGN KEY. Поле «наименование» символьное, переменной длины, не может быть пустым, неиндексированное, что дает, соответственно: имя – Name, тип – NVARCHAR(64), свойство NOT NULL. Поле “описание» символьное, переменной длины, неиндексированное, что соответствует: имя – Description, тип NVARCHAR(255) Преобразование данной таблицы приведено в табл. 3.3.

Таблица 3.3 – Таблица Категории

Имя столбца	Тип данных	Свойство	Ключ/Индекс
CategoryID	int	IDENTITY	PRIMARY KEY
ParentCategoryID	int	NOT NULL	FOREIGN KEY
Name	nvarchar(64)	NOT NULL	
Description	nvarchar(255)	NULL	

В графическом виде физическая модель базы данных будет выглядеть как показано на рис.3.1.

3.3.3 Реализация классов репозитория

Для реализации принципов ORM необходимо создать классы, которые в программе представляют объекты из реального мира, и реализовать программное обеспечение, которое может как обрабатывать данные в объектно-ориентированном виде, так и сохранять эти данные в реляционной форме. Для этого воспользуемся возможностями Visual Studio 2008 и создадим классы типа LINQ to SQL Classes.

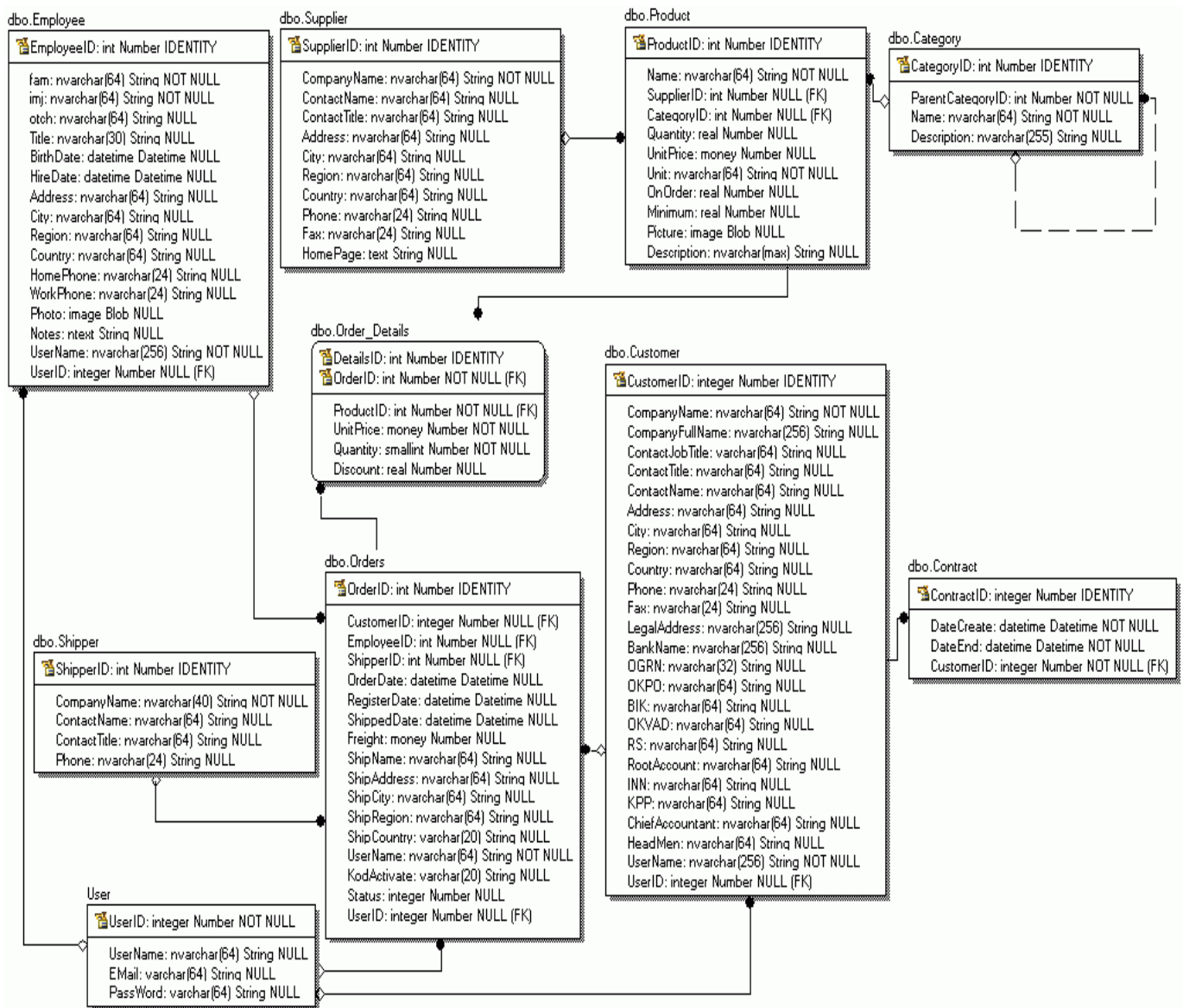


Рисунок 3.1 – Физическая модель базы данных

Создадим контекст базы данных, наследующий от класса `System.Data.Linq.DataContext` [7]. В LINQ для SQL его создают с помощью мастера, который автоматически сгенерирует LINQ для SQL-классов. В результате получим публичный, частичный класс, наследуемый от `System.Data.Linq.DataContext` и имеющий ссылки на все таблицы нашей базы данных.

В этом классе будут созданы методы :

- `OnCreated()`;
- `Insert()`;

- UpDate();
- Delete().

Данные методы предназначены для создания, добавления, обновления и удаления объектов в проекции на каждую таблицу реляционной базы данных.

Схема наследования класса `ais00_dbDataContext` приведена на рис.3.2.,

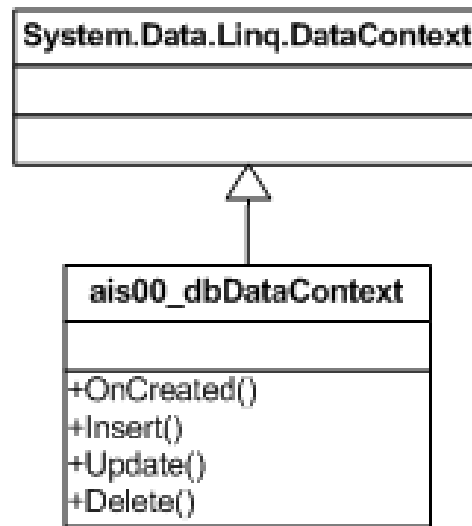


Рисунок 3.2 – Схема наследования класса `ais00_dbDataContext`

Для каждой таблицы нашей базы данных, в соответствии с технологией LINQ, получим публичные, частичные классы наследующие интерфейсы, которые предназначены для уведомления клиентов об изменении значения свойств. Пример создания класса `Category` приведен на рис.3.3.

Коды классов для LINQ to SQL приведен в приложении А.

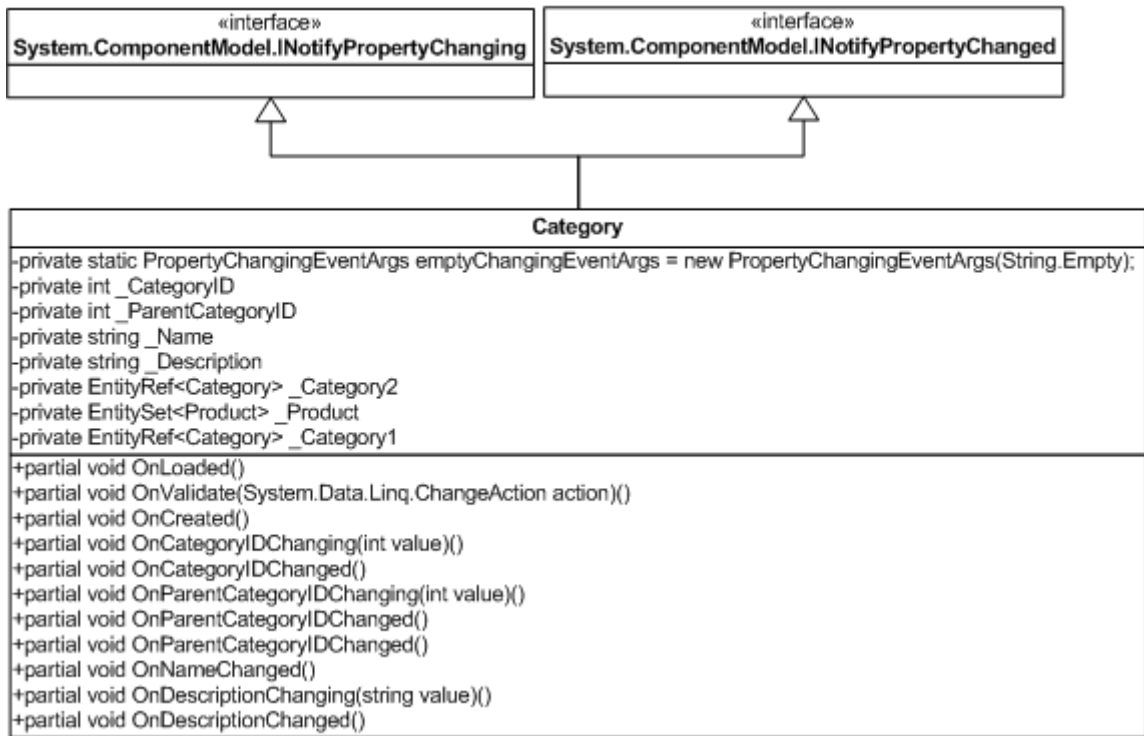


Рисунок 3.3 – Схема наследования класса Category

Для каждой таблицы из базы данных создадим класс обертку, который в своей переменной экземпляра класса будет инициализировать ссылку на контекст базы данных. Используем паттерн Repository. Пример класса-обертки для таблицы Category показан на рис.3.4.

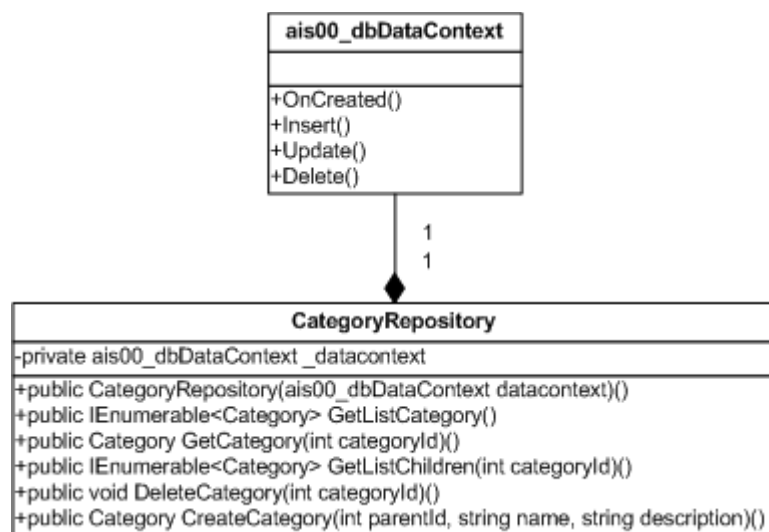


Рисунок 3.4 – Схема отношения агрегации класса CategoryRepository

Пример кода класса репозитория приведен в приложении Б.

Для упорядочивания работы с репозиториями всех объектов служит один класс, который будет работать со всеми репозиториями. Воспользуемся паттерном “Отложенная (ленивая) инициализация” (англ. Lazy initialization). Схема класса приведена на рис.3.5.

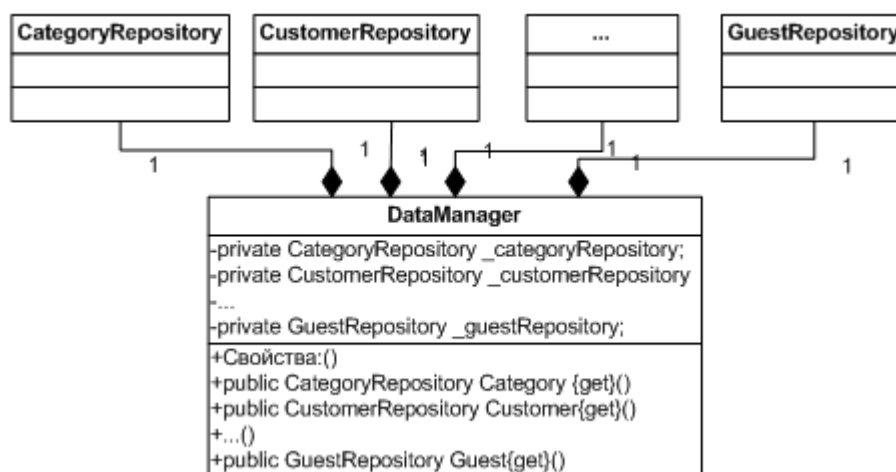


Рисунок 3.5 – Схема отношения агрегации класса DataManager

Код класса DataManager приведен в приложении В.

3.4. Разработка контроллеров

3.4.1. Разработка фабрики контроллеров

Контроллер — компонент, содержащий логику приложения. В контроллере описана логика взаимодействия с пользователем — в случае веб-приложения логика обработки HTTP-запросов к веб-приложению. Контроллер взаимодействует с объектами модели, которые, в свою очередь, влияют на представление.

В ASP.NET MVC фабрика контроллеров - это один из механизмов, который позволяет настроить поведение механизма MVC под себя, расширить функционал, предоставить разработчику возможность гибкой настройки действий MVC [7].

Сгруппируем действия по обработке данных в группы, каждая из которых будет соответствовать какой-либо роли пользователя в системе либо какому-то функциональному блоку системы. И для каждой группы создадим свой контроллер. При этом каждый контроллер должен иметь доступ к данным. Единой точкой входа к доступу к данным является класс `DataManager`. В каждом контроллере создадим ссылку на объект класса `DataManager`. Что бы избежать повторения кода инициализации данного объекта при написании кода каждого контроллера, вынесем этот код в одно место - фабрику контроллеров. Для этого нам нужно создать собственную фабрику контроллеров - `mMarketControllerFactory`, в которой переопределим метод `GetControllerInstance` фабрики контроллеров по умолчанию.

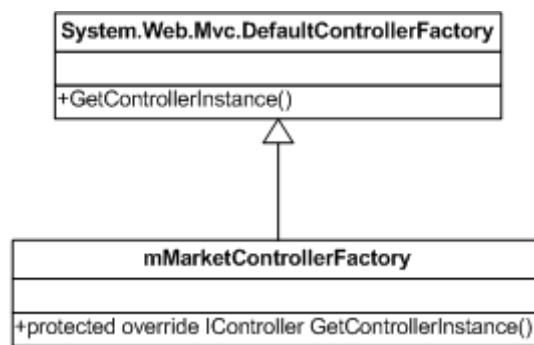


Рисунок 3.6 – Схема наследования класса `mMarketControllerFactory`

Зарегистрируем новую фабрику контроллеров `mMarketControllerFactory` в файле `Global.asax`.

3.4.2. Классы контроллеров приложения

Для реализации функционала продажи и редактирования списка товаров создадим контроллер `ProductController`. Для реализации функционала обработки заказов создадим контроллер `OrderManagerController`. Для реализации функционала представления пользователю прайс-листа продукции и начального управления действиями пользователей создадим контроллер `HomeController`. Для реализации функционала самостоятельной регистрации

клиента на сайте создадим контроллер `GuestController`. Для реализации функционала регистрации сотрудников на сайте создадим контроллер `EmployeeController`. Для реализации функционала в соответствии с ролью “покупатель” создадим контроллер `CustomerController`. Для реализации функционала управления категориями товара создадим контроллер `CategoryController`. Для реализации функционала допуска пользователей на сайт создадим контроллер `AccountController`.

Схема наследования классов контроллеров, переменные экземпляров контроллеров и методы контроллеров представлены на рис.3.7.

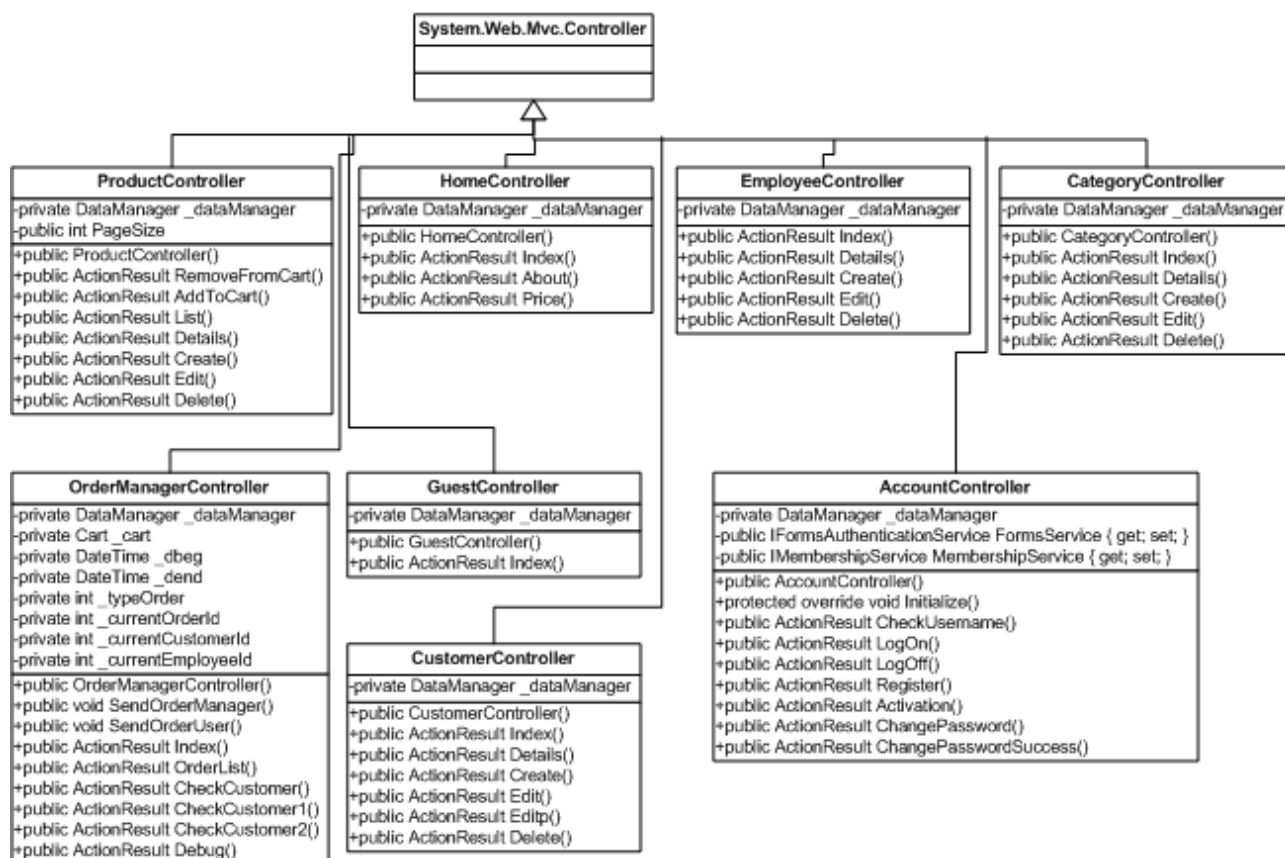


Рисунок 3.7 – Схема наследования классов контроллеров

Пример кода класса контроллеров приведен в приложении Г.

3.5 Разработка представлений и интерфейса приложения

3.5.1 Разработка шаблонов страниц приложения

Шаблоны страниц приложения или мастерские страницы (master pages) представляют собой инструмент для определения общего шаблона набора страниц, определяют общую разметку для набора контекстных страниц и расположения блоков, содержащихся в контекстных страницах.

Принцип построения мастерской страницы следующий:

- задается общий дизайн страницы и блоки, общие для набора страниц;
- в разметки мастерской страницы определяются блоки, которые будут заполнены содержимым контекстных страниц с помощью разметки;
- при создании разметки контекстных страниц используются блоки;
- мастерские страницы могут быть вложенными.

Создадим главную master pages - Site.Master:

- раздел заголовка окна;
- раздел заголовка страницы сайта - с логотипом, заголовком, окном регистрации на сайте и полосой меню;
- основной раздел страницы - с блоком сменной информации;
- раздел подвала страницы - с постоянной информацией для всех страниц.

Макет главной master pages страницы приведен на рис.3.8.

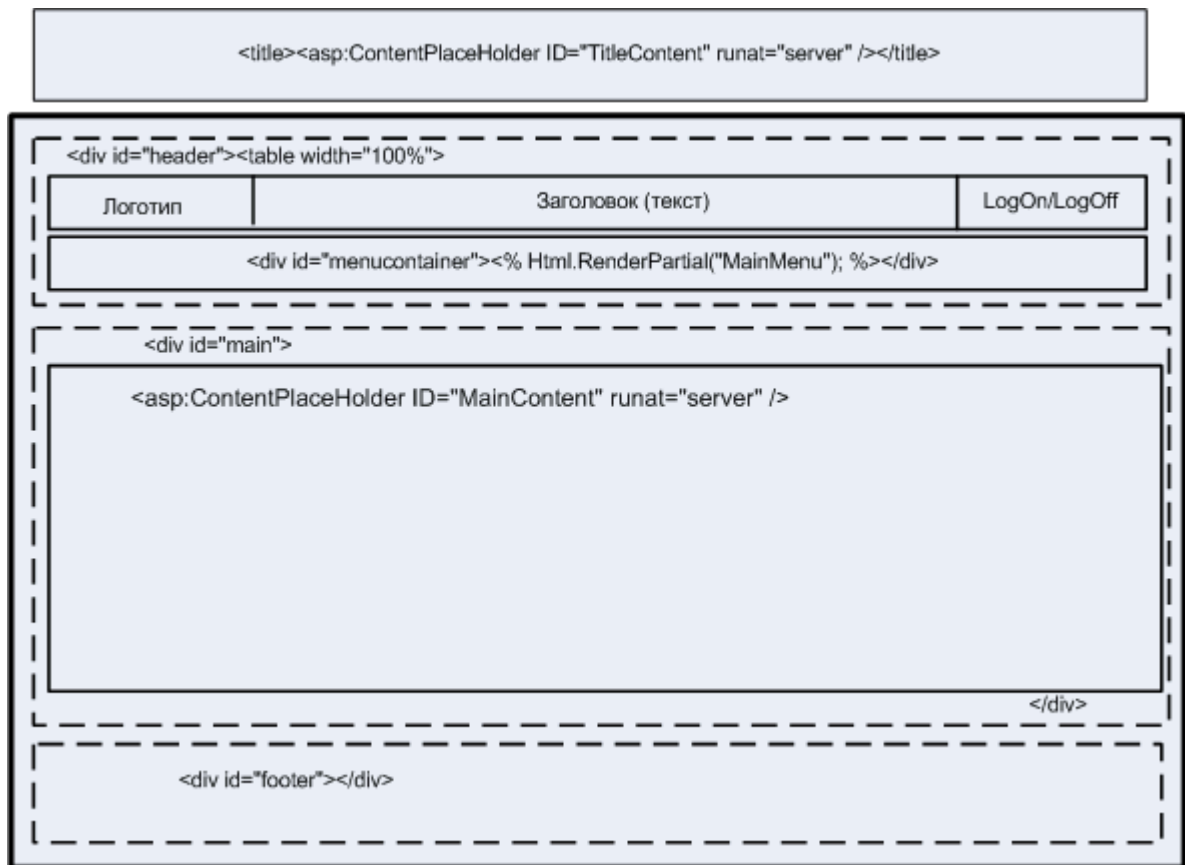


Рисунок 3.8 – Макет главной master pages страницы

Создадим вложенную страницу master pages с задачей выделить левый столбец страницы для размещения либо вспомогательного меню, либо дерево категорий. Правый столбец будет использоваться для основной сменной информации. Макет вложенной master pages страницы приведен на рис.3.9.

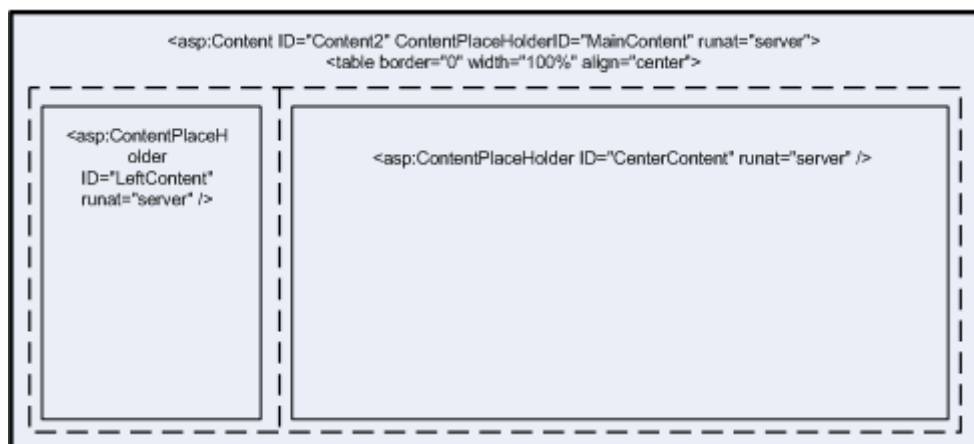


Рисунок 3.9 – Макет вложенной master pages страницы

Код разметки master pages приведен в приложении Д.

Цветовое решение внешнего вида приложения осуществлено с использованием CSS, что позволяет при необходимости легко его переделать с учетом пожелания клиента.

3.5.2 Разметка представлений

В проекте реализовано 33 элемента MVC2 View Content Page и 11 элементов MVC2 View User Control. View User Control позволили избежать повторного кодирования повторяющихся вариантов разметки отдельных элементов. Большое количество View Content Page представляют собой строго типизированные View типа Create, Details, Edit и List.

Примеры таких экранных форм приведены на рис.3.10-3.12.



Скриншот веб-приложения с заголовком «ООО-ФАС КОМПАНИЯ ФАС Автозапчасти оптом». В меню навигации: Главная, О компании, Контакты, Прайс-лист, Менеджер. Таблица «Покупатели»:

	Код	Компания	Контакт	Титул	Адрес	Город	Регион	Телефон	Логин
	6	customer	Покупатель	customer	customer	Самара	Самарская обл.	(8482)765434	customer
	7	sidorov	Сидоров С.Г.	Господин	Революционная 4Б	Тольятти	Самарская обл.	8(84862)21503	sidorov
	1	ООО "ФАС"	Федорова С.Г.	Госпожа	ул.Автостроителей, 34-52	Тольятти	Самарская обл.	(8482)765434	customer

Рисунок 3.10 – Экранная форма типа List

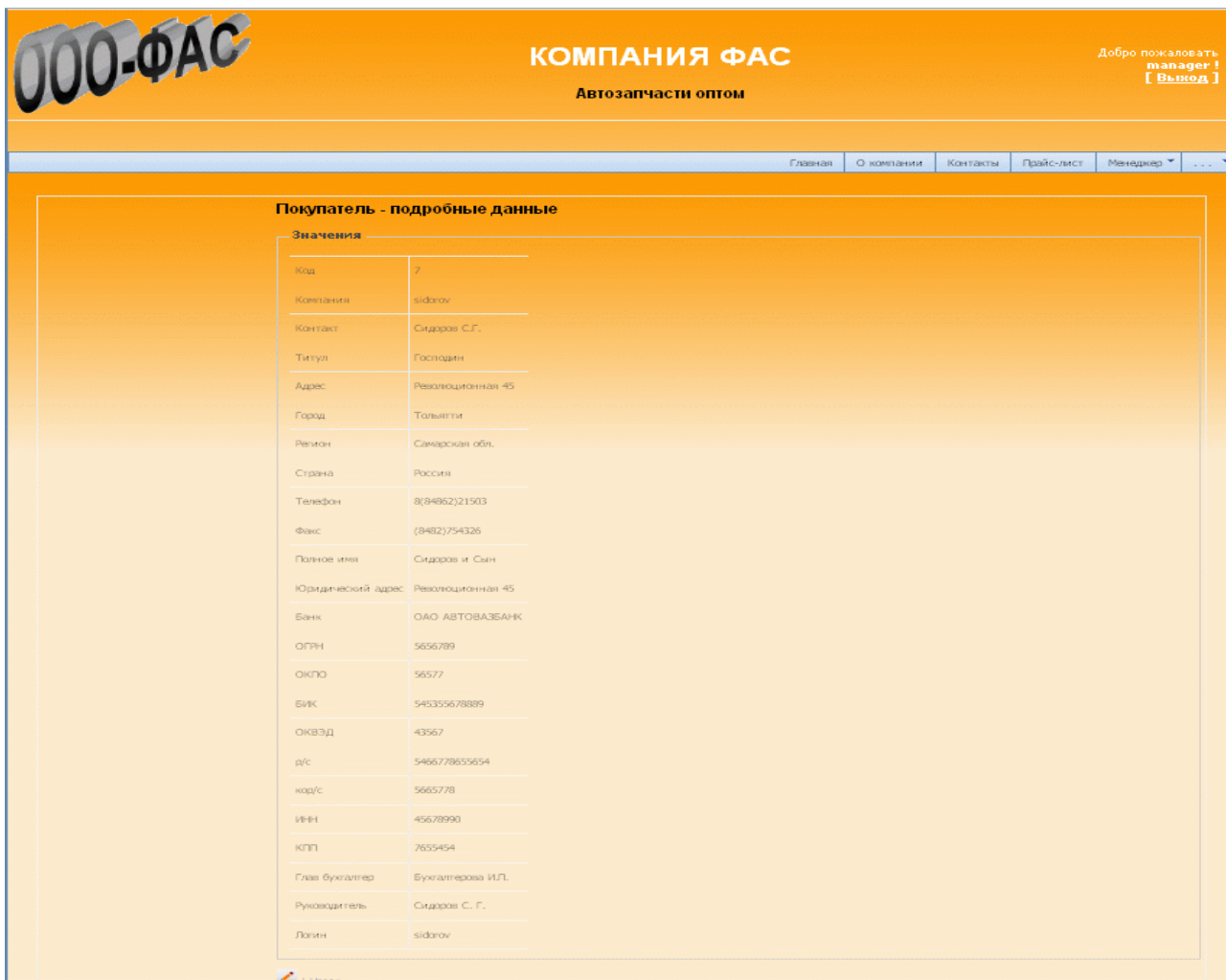


Рисунок 3.11 – Экранная форма типа Details

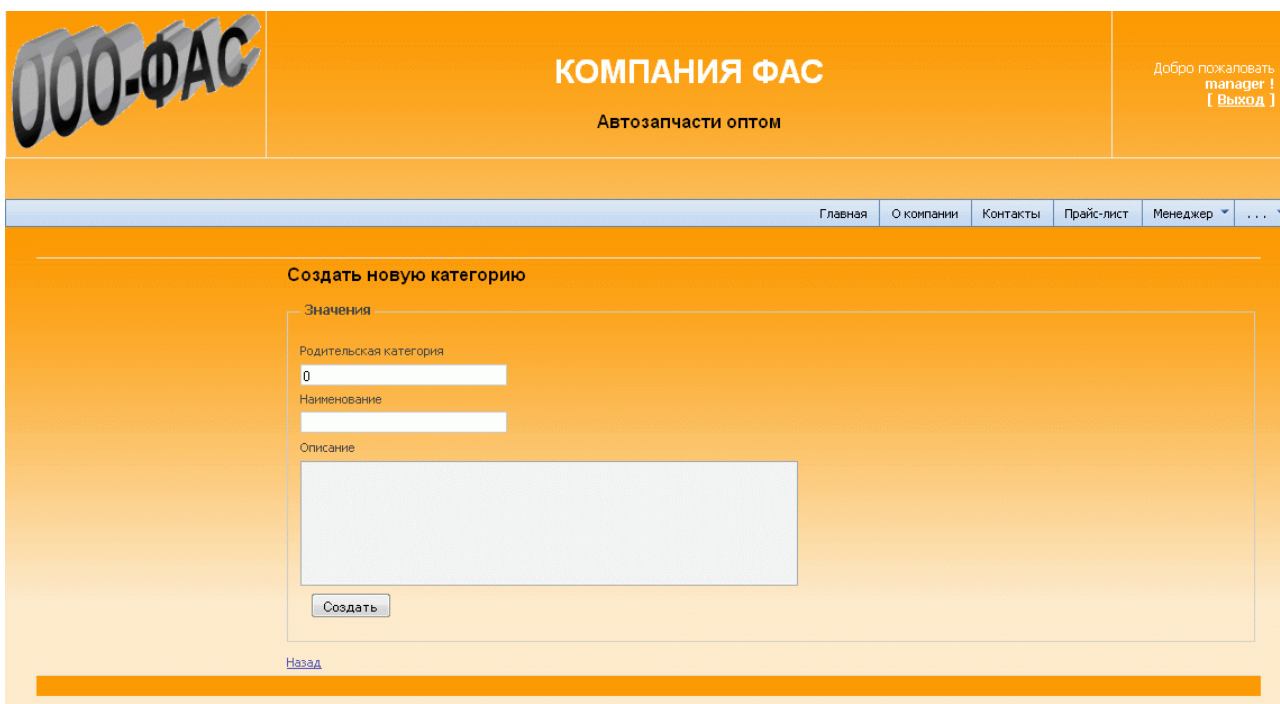


Рисунок 3.12 – Экранная форма типа Create

Без использования строго типизированных форм реализованы экранные формы типа Edit и Create для объектов имеющих двухуровневую систему такие как “Категории - Подкатегории”, “Заказы-Детальные данные по заказам”. Обработка данных из этих форм требует согласованного внесения нескольких изменений в базу данных, что привело к необходимости использования транзакций. Пример экранной формы типа Edit приведен на рис.3.13.



Рисунок 3.13 – Экранная форма типа Edit

Более сложно организована экранная форма отбора заказов. Она позволяет отбирать заказы, поступившие за период от: и до:, отобрать заказы по их статусу (не активированные, в работе, завершённые), отобрать заказы конкретного клиента, отобрать заказы курируемые конкретным сотрудником. Эта форма позволяет менеджеру отслеживать состояние заказов за всю фирму, продавцу - отслеживать состояние заказов которые поручено курировать ему и

вносить изменения в статус заказов, покупателю - отслеживать состояние его заказов.

Доступность некоторых элементов управления меняется в зависимости от роли пользователя открывшего эту экранную форму.

Пример экранной форма для отбора заказов из базы данных приведена на рис.3.14.

System.Web.Mvc.Html.MvcForm

Панель: Главная | О компании | Контакты | Прайс-лист | Менеджер | ...

Параметры отбора:

За период - от: 10.12.2010 до: 21.03.2011
 Выбрать заказы: Все | Выбрать Клиента: Все | Выбрать сотрудника: Все | Поиск

Список заказов

Код заказа	Код сотрудника	Дата	Регистрация	Отгрузка	Стоимость доставки	Имя для доставки	Адрес доставки	Город доставки
20		10.12.2010 18:16				Госпожа Федорова С.Г.	ул.Автостроителей, 34-52	Тольятти
21		10.12.2010 18:17				Госпожа Федорова С.Г.	ул.Автостроителей, 34-52	Тольятти
22		10.12.2010 18:23				Госпожа Федорова С.Г.	ул.Автостроителей, 34-52	Тольятти
23		10.12.2010 19:04				Госпожа Федорова С.Г.	ул.Автостроителей, 34-52	Тольятти
24		10.12.2010 19:05				Госпожа Федорова С.Г.	ул.Автостроителей, 34-52	Тольятти
25		10.12.2010 19:12				Госпожа Федорова С.Г.	ул.Автостроителей, 34-52	Тольятти
26		12.12.2010 10:28				Госпожа Федорова С.Г.	ул.Автостроителей, 34-52	Тольятти
27		07.03.2011 0:00	07.03.2011 0:00		860,00	Господин sidorov	Революционная 45	Тольятти

Выбранный заказ - детально

Заказ

Код ордера	Код документа	Код покупателя	Код сотрудника	Дата	Дата регистрации	Дата отгрузки	Стоимость доставки	Активация
27		7		07.03.2011 0:00	07.03.2011 0:00		860,00	

Имя доставки	Адрес доставки	Город доставки	Регион доставки	Страна доставки	Статус	Код поставщика	Логин
Господин sidorov	Революционная 45	Тольятти	Самарская обл.	Россия	2		seller

Позиции

id	Код заказа	Код товара	Наименование товара	Цена	Количество	Ед Изм	Скидка
44	27	10932	Вороток большой	49,50	1	шт	
45	27	10933	Вороток малый	39,60	1	шт	
46	27	10934	Индикатор качества смеси	352,00	1	шт	
47	27	10935	Ключ 8x9x10	38,50	1	шт	
48	27	10936	Ключ карданчик 12	49,50	2	шт	
49	27	10937	Ключ карданчик 13	55,00	1	шт	

Итого: 633,6000

Рисунок 3.14 – Экранная форма для отбора заказов из базы данных

Экранная форма редактирования данных пользователя системы подготавливает данные для изменения сразу в нескольких таблицах базы данных. Для правильного внесения изменений в базу данных в процессе обработки данных поступающих от этой формы открывается транзакция для

согласованного изменения данных в таблицах aspnet_Users, aspnet_UsersInRoles, aspnet_Membership, aspnet_PersonalizationPerUse, Employee.

Пример экранной формы редактирования данных пользователя системы показан на рис.3.15.

The screenshot shows a web application interface for editing user data. The header includes the company logo 'ООО-ФАС' and the name 'КОМПАНИЯ ФАС' with the tagline 'Автозапчасти оптом'. A navigation menu at the top right includes 'Главная', 'О компании', 'Контакты', 'Прайс-лист', and 'Администратор'. The main content area is titled 'Edit' and contains three panels. The 'Сотрудник' panel has input fields for 'Логин' (Pupkin) and 'Email' (pupkin@mail.ru). The 'Роли' panel has checkboxes for 'Администратор', 'Снабженец', 'Руководитель', and 'Продавец', with 'Руководитель' and 'Продавец' selected. The 'Данные по сотруднику' panel has a form with fields for 'Код сотрудника' (2), 'Фамилия' (Пупкин), 'Имя' (Василий), 'Отчество' (Иванович), 'Титул' (Господин), 'Дата рождения' (12.09.196), 'Дата приема на работу' (17.12.201), 'Адрес' (Революционная 45), 'Город' (Самара), 'Регион' (Самарская обл.), 'Страна' (Россия), 'Домашний телефон' (8482)768792, and 'Рабочий телефон' (8482)768445. A 'Сохранить' button is at the bottom of the form.

Рисунок 3.15 – Экранная форма редактирования данных пользователя системы

ASP.NET 3.5 предлагает несколько серверных элементов управления связанных с защитой приложения: Login, LoginName, LoginView, CreateUserWizard. Эти комбинированные элементы управления инкапсулируют

большую часть шаблонного кода и разметки, и используются для регистрации и авторизации пользователя.

3.6. Наполнение базы данных и тестирование приложения

При разработке приложения использованы материалы с сайта - визитки фирмы ООО “ФАС”. На сайт перенесена вся статическая информация со страниц: “Главная”, “О компании”, “Контакты”. В базу данных перенесена информация по более чем 2000 позиций товаров, которыми торгует фирма. Выполнено наполнение всех таблиц базы данных тестовыми данными.

Произведено первичное тестирование базы данных с помощью SQL Manager Lite for MySQL путем выполнения запросов на вставку записей с передачей данных, нарушающих правила, определенных для полей таблиц с последующим контролем невозможности выполнения вставок таких данных .

Произведена проверка работоспособности классов репозиториев путем вставки, удаления и добавления записей с последующим контролем их исполнения с помощью SQL Manager Lite for MySQL.

Произведена проверка работоспособности классов репозиториев и контроллеров путем выборки из базы данных с применением фильтров с последующим контролем их выполнения посредством отладчика Visual C# 2008 Express Edition.

Произведена проверка работоспособности всех экранных форм и контроллеров путем задействования элементов управления на экранных формах и последующим наблюдением за реакцией системы на эти действия.

Протестирована работоспособность элементов, отвечающих за отправку сообщений по электронной почте с помощью иммитатора почтового сервера из состава Visual C# 2008 Express Edition.

Осуществлен перенос приложения с машины разработчика на машину, иммитирующую сервер, на котором были развернуты WEB-сервер IIS, сервер базы данных и осуществлен тестовый доступ к серверу из сети Интернет.

Отработаны вопросы развертывания приложения на реальной машине-сервере.

В настоящее время осуществляется тестирование работоспособности приложения на рабочих местах.

Обобщая материалы главы можно сформулировать следующие итоги:

– обоснован выбор использования технологии ASP.NET, среды разработки Visual C# 2008 Express Edition, СУБД Microsoft SQL Server 2005 и веб-сервер версии IIS 6.0. Выбран самый новый и мощный язык программирования C#, в том числе и Web приложений.

– произведено сравнение технологической платформы разработки веб-приложений webforms и MVC Framework, и обоснован выбор MVC Framework.

– осуществлен выбор библиотеки ORM - LINQ для SQL.

– разработана физическая модели БД.

– разработаны классы репозитория и модель данных в терминологии MVC Framework.

– разработаны классы контроллеров приложения, что позволило реализовать бизнес-логику приложения.

– разработан интерфейс приложения в виде представлений. Всего создано 33 строго типизированных и не типизированных представления.

– осуществлено наполнение базы данных реальными данными по товарам фирмы и тестовыми данными по остальным позициям.

– выполнено тестирование всех представлений и контроллеров приложения с тестовыми данными;

– осуществлено развертывание приложения на рабочих местах и начато тестирование приложения на реальных данных.

ЗАКЛЮЧЕНИЕ

В современных условиях эффективное управление представляет собой ценный ресурс организации, наряду с финансовыми, материальными, человеческими и другими ресурсами. Следовательно, повышение эффективности управленческой деятельности становится одним из направлений совершенствования деятельности предприятия в целом. Наиболее очевидным способом повышения эффективности протекания трудового процесса является его автоматизация.

Развивая информационные системы (ИС) необходимо стремиться в производственную часть бизнеса, создавая возможность не только примитивного набора информации, оптимизации бизнес процессов и других атрибутов внедрения, но обеспечивать возможность аналитической обработки информации на уровне свойств продукта, технологий, ресурсов.

В данной работе были проанализированы бизнес-процессы существующие в компании ООО «ФАС» в настоящее время, и сделан вывод о целесообразности внедрения CRM системы в деятельность компании. На основании этого были сформулированы требования к CRM системе компании. Анализ известных CRM - решений показал, что они не удовлетворяют в полной мере требованиям, предъявляемым к новой CRM. В этой связи принято решение о разработке CRM с учетом особенностей компании.

На основе анализа существующих бизнес процессов, разработан новый регламент осуществления бизнес процесса купли продажи.

Разработаны диаграммы движения информации в системе, которые на более низком уровне абстракции описывают порядок действий пользователей системы с различными ролями в системе.

В соответствии с этапами моделирования БД разработана модель базы данных системы.

Обоснован выбор использования технологии ASP.NET, определен инструментарий разработки. Произведено сравнение технологической

платформы разработки веб-приложений WebForms и MVC Framework, и обоснован выбор MVC Framework.

Обоснован выбор библиотеки ORM - LINQ для SQL, разработаны классы репозитория и модель данных в терминологии MVC Framework.

Разработаны классы контроллеров и представлений приложения.

Осуществлено наполнение базы данных реальными данными по товарам фирмы и тестовыми данными по остальным позициям.

Выполнено тестирование всех представлений и контроллеров приложения с тестовыми данными, осуществлено развертывание приложения на рабочих местах и начато тестирование приложения на реальных данных.

Достоинствами разработанной системы являются возможности повысить качество обслуживания клиентов, стандартизировать форматы данных по работе с клиентами и аккумулировать их в единой базе данных, обеспечить контроль деятельности менеджеров, предоставить доступ к отчетам для топ-менеджеров в режиме он-лайн, осуществить предварительную регистрацию потенциальных клиентов зашедших на сайт для ознакомления, предоставить возможность клиентам, заключившим длительные договора, дистанционно через интернет осуществлять закупку товаров фирмы, предоставить клиентам возможность контролировать стадии реализации их заказов.

Направлением дальнейшего перспективного развития системы предполагается разработка модуля, обеспечивающего возможность поставщикам отслеживать наличие продукции на складе с целью представления своих предложений по ее пополнению. Следующим направлением развития системы предполагается создания модуля, отслеживающего состояние заказов с целью своевременного напоминания менеджерам о необходимых очередных действиях. Также направлением дальнейшего развития системы предполагается построение блока аналитической обработки заказов для построения воронки продаж.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

Нормативно-правовые акты

1. ГОСТ 7.82-2001. Библиографическая запись. Библиографическое описание электронных ресурсов.
2. ГОСТ 19.701 – 90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения (ИСО 5807–85) [Текст]. Введен 1992–01–01. – М.: Изд-во стандартов, 1992. – 14 с. – (Единая система программной документации).
3. ГОСТ Р 34.13-2015. Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров [Электронный ресурс] // Технический комитет по стандартизации «Криптографическая защита информации» (ТК 26) [Электронный ресурс]. URL: https://www.tc26.ru/standard/gost/GOST_R_3413-2015.pdf
4. ГОСТ 2.105 – 95. Общие требования к текстовым документам [Текст]. М.: Изд-во стандартов, 1996. – 29 с. – (Единая система конструкторской документации).
5. ISO/IEC 14102:1995. Информационные технологии. Руководство по оцениванию и выбору инструментальных CASE-средств.
6. ГОСТ Р 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры [Электронный ресурс] // Технический комитет по стандартизации «Криптографическая защита информации» (ТК 26) [Электронный ресурс]. URL: https://tc26.ru/standard/gost/GOST_R_3412-2015.pdf
7. ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание документа.
8. ГОСТ 7.32-2001. Отчет о научно-исследовательской работе. Структура и правила оформления.

Научная и методическая литература

9. Агальцов В. П. Базы данных [Электронный ресурс] : учебник. В 2-х кн. Кн. 2. Распределенные и удаленные базы данных / В. П. Агальцов. - Москва : Форум : ИНФРА-М, 2017. - 271 с. : ил. - (Высшее образование). - ISBN 978-5-8199-0394-0..

10. Белов В. В. Проектирование информационных систем : учеб. для студентов вузов, обуч. по направлению "Прикладная информатика" и др. эконом. специальностям / В. В. Белов, В. И. Чистякова ; под ред. В. В. Белова. - 2-е изд., стер. ; гриф УМО. - Москва : Академия, 2015. - 351, [1] с. : ил. - (Высшее образование. Бакалавриат). - Библиогр.: с. 345-347. - ISBN 978-5-4468-2440-3 : 893-55.

11. Гущина О. М. Структурный анализ и проектирование систем управления образовательным процессом : учеб. пособие / О. М. Гущина, С. В. Лаптева ; ТГУ ; Ин-т математики, физики и информ. технологий ; каф. "Информатика и вычислит. техника". - ТГУ. - Тольятти : ТГУ, 2013. - 187 с. : ил. - Библиогр.: с. 179-181. - Глоссарий: с. 182-185. - ISBN 978-5-8259-0750-5 : 63-04.

12. Калянов Г.Н. Моделирование, анализ, реорганизация и автоматизация бизнес-процессов: учеб. - М.: Финансы и статистика, 2006. - 319 с.

13. Магдануров Г.И. ASP.NET MVC Framework / Г.И. Магдануров, В.А. Юнев. - СПб.: БХВ-Петербург, 2010. - 320 с.

14. Пауэрс Л. Microsoft Visual Studio 2008 / Л. Пауэрс, М. Снелл. - СПб.: БХВ-Петербург, 2009. - 1171 с.

15. Пейн Э. Руководство по CRM. Путь к совершенствованию менеджмента клиентов. - Минск: Гревцов Паблишер 2007. - 384 с.

16. Раттц-мл Д. LINQ язык интегрированных запросов в C# 2008 для профессионалов . - М.: Вильямс 2008. - 591 с.

17. Реинжиниринг бизнес-процессов : учеб. пособие для вузов / А. О. Блинов [и др.] ; под ред. А. О. Блинова. - Гриф УМО. - Москва : ЮНИТИ-

ДАНА, 2010. - 341 с. : ил. - Библиогр.: с. 334-338. - ISBN 978-5-238-01823-2 : 300-00.

18. Сандерсон С. ASP.NET MVC Framework с примерами на С# для профессионалов. - СПб.: Вильямс, 2010. - 556 с.

19. Стиллмен Э. Изучаем С# / Э. Стиллмен, Грин Дж. - Санкт-Петербург : Питер, 2013. - 694, [1] с. : ил. - (Head First). - ISBN 978-5-4461-0105-4 : 780-00.

20. Столбовский Д. Н. Основы разработки Web-приложений на ASP.NET [Электронный ресурс] : учеб. пособие / Д. Н. Столбовский. - 2-е изд., испр. и доп. - Москва : ИНТУИТ, 2016. - 375 с. : ил. - (Основы информационных технологий). - ISBN 978-5-94774-991-5.

21. Трофимов С.А. CRM для практиков. Второе дыхание отдела продаж. - М.: АвтоКод, 2006. - 304 с.

22. Трофимов С.А. Rational XDE для Visual Studio.NET. -М.: ООО «Бином-Пресс», 2004 - 304 с.

23. Шилдт Г. Полный справочник по С#. : Пер. с англ. - М.: Издательский дом «Вильямс», 2004. - 752 с.

Электронные ресурсы

24. 1С:Предприятие 8. CRM. Базовая версия [Электронный ресурс]. - <http://www.crmonline.ru/software/native/1c/>

25. CRM – технология взаимодействия с клиентами. [Электронный ресурс]. - http://www.crmonline.ru/phparticles/show_news_one.php?n_id=280

26. CRM-система "КОМПАС: Маркетинг и менеджмент" [Электронный ресурс]. - <http://www.compas.ru/solutions/mandm.php>

27. Sales Expert [Электронный ресурс]. - <http://worldseo.co.cc/tag/salesexpert/>

28. Terrasoft CRM – эффективная CRM-система для Вашего бизнеса [Электронный ресурс]. - <http://www.terrasoft.ua/products/CRM/crm>

29. Обзор современных web технологий построения приложений серверной стороны [Электронный ресурс]. -

<http://www.sciteclibrary.ru/rus/catalog/pages/6643.html>

30. Парус - Менеджмент и маркетинг [Электронный ресурс]. - <http://www.crmonline.ru/software/native/parus/>.

Литература на иностранном языке

31. Chadwick, Jess. Programming ASP.NET MVC 4 : Developing Real-World Web Applications with ASP.NET MVC / Jess Chadwick, Todd Snyder, Hrusikesh Panda. – O'Reilly Media, 2012.

32. Ciliberti, John. ASP.NET MVC 4 Recipes : A Problem-Solution Approach / John Ciliberti. – Apress, 2013.

33. Griffiths, Ian. Programming C# 5.0 : Building Windows 8 Metro, Web, and Desktop Applications for the .NET 4.5 Framework / Ian Griffiths. – O'Reilly Media, 2012.

34. Ruivo P. et al. Defining the ERP and CRM integrative value //Procedia Technology. – 2014. – Т. 16. – С. 704-709.

35. Šebjan U., Bobek S., Tominc P. Organizational factors influencing effective use of CRM solutions //Procedia Technology. – 2014. – Т. 16. – С. 459-470.

