

Оглавление

Введение.....	2
1. Роль web-приложений в современном обществе.....	4
2. Обзор аналогов и требования к системе.....	14
2.1. Обзор имеющихся решений.....	14
2.2. Описание Java технологий.....	17
2.3. Требования к системе.....	21
3. Программное web-приложение.....	25
3.1. Архитектура и реализация.....	25
3.2. База данных.....	37
3.3. Функциональные возможности.....	40
3.4. Тестирование.....	42
Заключение.....	47
Список использованных источников.....	48
Приложение А.....	51

Введение

Для ведения успешного бизнеса основным моментом представляется продвижение и реклама услуг и товаров на соответствующем рынке. В успешных компаниях не задается вопрос: "что продавать", а ищется ответ на вопросы: "Как и кому продавать". Не следует заострять внимание на самих товарах или услугах, а следует постоянно проводить поиск новых возможностей для реализации.

Разработка и поддержка веб-приложения в сети Интернет - это достаточно эффективный способ рекламы товаров и услуг. Этот способ применяется как в малом предпринимательстве, так для крупных международных компаний. В настоящее время пользователями сети является большая часть населения Земли. Сегодня большое количество успешных компаний используют сеть Интернет как место рекламы тем или иным способом. Игнорирование такого инструмента увеличения продаж может сильно осложнять деятельность компаний из-за отсутствия стабильного числа клиентов, которые мог бы быть привлечены при помощи Интернет-ресурса [1].

Качественное продвижение сайта компании, которое проведено квалифицированными специалистами, позволит не только привлекать новых клиентов, но и сделать известным товарный знак компании, делая его более запоминающимся, а также заинтересовывать новых инвесторов и партнеров. Главными преимуществами работы с ресурсом в сети Интернет можно указать постоянную работу сайтов 24 часа в сутки и небольшую стоимость их создания и развития. С помощью сайта компании можно сообщать потенциальным клиентам необходимую информацию, прослеживать интерес к товарам и услугам, в оперативном режиме обрабатывать заказы.

Разработка веб-приложения для сети Интернет поможет компании достаточно быстро реагировать на любые изменения состояния рынка, создавать новые способы продажи товаров и услуг, а также увеличить объем продаж.

Тема работы: «Веб-приложение для поиска развлекательных заведений».

Актуальность работы проявляется в том, что в настоящее время наблюдается значительный сдвиг продаж в область электронной торговли. Это обуславливается расширением структуры электронных коммуникаций и ростом цен на ресурсы и аренду. В результате вышесказанного в условиях рыночной экономики актуальной становится тема создания Web-ресурсов.

Объектом исследования являются исследование средств разработки веб-приложений для коммерческих целей.

Предметом исследования является создание веб-приложения для поиска развлекательных заведений.

Для достижения цели работы были поставлены следующие задачи:

- Рассмотреть предметную область, определить комплексы задач автоматизации, выбрать место решаемых задач в этих комплексах;
- Осуществить формализацию задач, выбрать стратегию автоматизации;
- Обосновать проектное решение;
- Разработать информационную и датологическую модель;
- Разработать программу;
- Провести тестирование полученного программного продукта;
- Сделать выводы.

1. Роль web-приложений в современном обществе

Сеть Интернет является глобальной информационной сетью, все части которой связываются друг с другом в едином адресном пространстве, основанном на использовании протокола TCP/IP. Сеть Интернет построена из множества взаимосвязанных компьютерных сетей и предоставляет удаленный доступ к компьютерам, доскам объявлений, электронной почте, базе данных и дискуссионным группам [1].

Интернет – это перспектива коммуникаций. В настоящее время Сеть дает возможность получать видео из любого уголка мира, отправлять электронную почту с картинками, использовать Интернет телефонию и организовывать международные конференции.

Создание сайта на определенную тему для себя – это первый шаг к созданию компании. При создании информационного, мультимедийного сайта с заданной тематикой и продвигая его в ТОП поисковых систем, можно получить уже готовую, сформировавшуюся аудиторию, т.е. потенциальных будущих клиентов. А когда есть клиенты и выбрано направление деятельности, создавать собственное предприятие легче. В дальнейшем можно пойти по одному из трёх путей монетизации вашего проекта или выбрать сразу все [2]:

- первый – собственный сайт позволяет получать дополнительную прибыль путем размещения на своих страницах рекламы других компаний, соответствующих теме вашего сайта;
- второй – есть возможность открыть интернет магазин, соответствующий профилю вашего сайта и продавать продукт, ведь клиенты уже есть;
- третий – выбрав производственное направление деятельности для собственного предприятия, развивать сайт как рекламный рупор компании, способствующий увеличению продаж производимых продуктов.

Современные интернет-технологии позволяют создавать сайт не только по шаблону, но и делать его уникальным под определенную компанию, а это значительно повышает его рейтинг. Особое внимание в данном случае уделяется дизайну. Очень важно использовать при этом фирменные цвета, логотип и фирменный стиль компании, то есть те атрибуты, по которым будет узнаваема торговая марка или бренд.

Однако процесс создания сайта очень трудоемок, поэтому следует провести детальный анализ Интернет-ресурсов конкурирующих компаний, чтобы определиться со стилем и другими критериями привлечения потенциальных клиентов. Важно выбрать тип сайта, ведь не всегда сайт-визитка — это наиболее оптимальный вариант. Иногда лучше заказать разработку бизнес-сайта или сразу интернет-магазина. Многое зависит от сферы деятельности компании [2].

Одним из важнейших факторов определения успешности сайта является его наполнение. Если первое, что оценивает пользователь, открывая главную страницу - это оформление, то второе и самое важное, на что он обращает внимание – это содержание сайта, (его контент). Именно от того, насколько представленная информация интересна, полезна, актуальна и понятна пользователю, зависит, захочет ли он остаться на сайте, стать клиентом компании, продолжить или начать сотрудничество, или уйдет к конкуренту.

Веб-дизайном называют создание дизайна сайта и его отдельных частей, создание иллюстративных и рекламных материалов в Интернет. Часто услуги веб-дизайна предоставляются различными студиями веб-дизайна [6].

Веб-сервер – это сервер, который обеспечивает снабжение информацией в службе глобального соединения. Веб-сервер хранит и предоставляет во внешнюю сеть информацию, которая имеется на веб-страницах. Веб-сервер является ответственным за обработку запроса клиента к веб-сайту и исполнение CGI-, JSP-, ASP-, PYTHON- и других приложений.

Веб-страница – это самостоятельная составляющая веб-сайта; документ, имеющий уникальный адрес (URL). Веб-страница может иметь динамическое

или статическое построение. Как правило, веб-страницы организованы в виде гипертекста с включениями текста, графики, звука, видео или анимацию. В сети Интернет просмотр веб-страниц происходит посредством браузера.

В настоящий момент широко применяют концепцию веб-сайта, когда пользователю не нужно выходить из дома чтобы приобрести себе какой-либо товар или получить новую информацию [3].

На данный момент веб-сайт представляет собой новую, но уже устоявшуюся систему реализации товаров или услуг. По сути, эта система является более эффективной по сравнению со стандартным «физическим» представлением информации. Применение веб-сайта в качестве информационной системы для электронной коммерции дает возможность:

- улучшения качества обслуживания клиента, так как все действия по покупке товара производят в автоматическом режиме, а менеджеру следует только уточнять информацию и выдавать распоряжение о доставке товара;
- снижения трудоемкости и напряженности труда персонала компании, минимизации ошибок в их действиях.
- снижения издержек, так при этом отпадет необходимость в аренде здания для отдельного офиса.

Быстрое развитие информационных технологий, среди которых одно из главных мест занимает Интернет, появление и стремительный рост электронной коммерции стали основной причиной для развития новой ветви в нынешней концепции маркетинга взаимодействия - Интернет-маркетинга.

Однако авторы многих имеющихся сегодня в Интернет публикаций трактуют понятие Интернет-маркетинга как продвижение каких-либо товаров или услуг в Интернете, или объясняют его еще более односторонне: как рекламу в сети Интернет, либо еще ограниченнее – только как баннерную рекламу. Подобные трактовки неверны именно в силу узости такого понимания рассматриваемого вопроса.

Как правило, Интернет-маркетинг включает целый набор дочерних отраслей. Они содержат не только баннерную рекламу и public relations, а также и методику проведения маркетингового исследования в Интернете, а именно, изучение спроса и потребительской аудитории, изучение алгоритма формирования и обеспечения высокой эффективности рекламной кампании, способов рационального позиционирования торговой марки на рынках, и многое другое. Рассмотренное ранее узкое понимание данного предмета является, вероятно, своего рода «детской болезнью» Интернет-маркетинга. Это, в свою очередь, можно анализировать как беспристрастный положительный признак. Наличие подобных «детских болезней» свидетельствует о том, что Интернет-маркетинг у нас растет быстро и развивается активно [9].

Интернет-маркетинг - это теория и методология организации дела по маркетингу в среде Интернета [4].

Интернет имеет уникальные характеристики, которые серьезно отличаются от характеристик традиционных приемов маркетинга. Например, одним из главных качеств среды Интернет является ее гипермедийная природа. Это качество характеризуется высокой эффективностью в представлении и усвоении информации. В связи с этим значительно повышаются возможности маркетинга в усилении взаимосвязи предприятий и потребителей. Надо отметить, что многие специалисты в области PR и маркетинга оказались к этой революции не подготовлены. Они не смогли приспособиться, не воспринимают Интернет как уникальное киберпространство, где осознанно находятся миллионы людей. Поэтому многие компании были выброшены на обочину, стали одинокими и забытыми в этом киберпространстве.

Другая категория компаний смогла войти в Интернет, но не смогли добиться успехов, так как больше заботились о том, что они хотят получить от Интернета, не стремились при этом разобраться с требованиями новой целевой аудитории.

Анализируя определение маркетинга на электронном рынке, важно также ввести понятия, что является объектом и субъектом маркетинговой деятельности на данном рынке. Объектом маркетинговой деятельности на электронных рынках является информационно-аналитическое и экспертно-исследовательское функционирование фирм, с применением сетевых информационных технологий и систем [5]:

- выбор конкурентных позиций на данных рынках, на которых компания действует со своими товарами;

- определение стратегии продвижения продукции и её распределения, выбор рекламной и ценовой политик с учетом всех совокупностей факторов внешней и внутренней среды в условиях рисков и неопределенностей.

Субъектом маркетинговой деятельности на электронных рынках является, как и при использовании классического маркетинга, деятельность конкретных собственников или компаний по целенаправленному регулированию деятельности организаций. Регулирование производят по определенной технологии с применением систем методов анализа и обработки цифровых данных электронного рынка для реализации поставленной цели.

Процессы управления в маркетинге на электронных рынках отображают совокупность операций и процедур, которые выполняют работники маркетинговых служб компаний в нужной последовательности. Он включает [11]:

- сбор и анализирование данных о поведении рынка и конкурентов на нем с помощью применения возможностей сети Интернет, хранилища корпоративной базы данных;
- экспертизу стохастических и динамических процессов на данном сегменте рынка;
- построение модели психологических реакций поведения потребителей на электронном рынке, построение модели стратегии компании в условиях рисков и неопределенностей, формулировка новой стратегии

развития текущего рынка, внедрения на рынок, стратегических альянсов и консолидации, стратегии диверсификации и др.

Перед тем, как переходить к принципам продвижения сайта, рассматриваем возможные уже решенные аналогичные задачи, для решения которых система электронной коммерции создается и продвигается.

1. Модель Ad Supported (рекламная бизнес-модель). В рамках данной модели сайты создаются для образования с его использованием постоянной, четко сегментированной или, напротив, максимально широкой аудитории посетителей. Контакты с этой совокупностью пользователей продают рекламодателю или спонсору. Можно привести множество примеров. Например: www.citycaty.ru является крупнейшим рассылочным сервером, который получает доходы исключительно от использования рекламных сообщений в рассылках; www.autoca.ru является крупным автомобильным порталом Рунета, который является привлекательным для рекламодателя из автомобильного и сопутствующего бизнеса [7].

2. Модель Business Supported (или модель для поддержки существующих бизнесов). Если компания обладает бизнесом offline, то с применением сайтов и рекламных мероприятий в сети Интернет, фирма может расширить свою клиентскую базу, формировать благоприятный имидж. Имеется возможность открытия новых каналов продажи через сеть Интернет. Осуществляется через сеть поддержка клиентов и партнеров, оптимизируются с помощью интернет-технологии бизнес-процессы внутри организации.

3. Модель New Business (Модель создания новых бизнесов). Сеть предоставляет возможности создания нового бизнеса или бизнес-единицы внутри компаний. Пример такого нового бизнеса - это студия веб-дизайна, интернет-агентство, интернет-магазин, платный сервис и т.д. Можно привести в качестве примера сервер информационного агентства "Интерфакс" (www.interfax.ru), где доступ к значительной части содержащейся информации предоставляется за определенную абонентскую плату.

4. Модель Capitalization (капитализация проекта). В основе проектов лежат некоторые из приведенных выше моделей, но главная цель создателей модели состоит в росте капитализации (стоимости) и ликвидности (возможности продажи) проектов. В зависимости от бизнес-планов капитализация может определяться аудиторией электронного ресурса, денежными потоками, которые проходят через интернет-сервисы, технологического владения, know-how и т.д. Примеры магазин Ozon.ru, контент-проекты Referat.Ru, поисковые системы Yandex.Ru и т.д [18].

С точки зрения использования технологий маркетинга, электронные веб-приложения - это наборы информационных блоков и инструментов для реализации взаимодействия с одним или несколькими сегментами в целевой аудитории. Какие данные будут представлены на сайте, какие инструменты будут использоваться, и каким образом они будут взаимодействовать между собой, зависит от двух аспектов [8]:

- выбранная бизнес-модель, краткосрочная и долгосрочная задача;
- тип сегмента целевой аудитории и возможность взаимодействия с ней теми либо иными способами.

Функциями электронных веб-приложений являются [6]:

1. Представительская функция и имидж организации:

- При участии в переговорах или презентациях, представители компании иногда не имеют возможности и времени предоставлять все необходимые данные оппонентам. Адреса корпоративных серверов на визитках или в небольших буклетах могут помочь в решении этих проблем. Потенциальные партнеры, заинтересованные в услугах компании, обязательно зайдут на её сайт. Подробную информацию о виде деятельности, истории организации, о продукции или услугах поможет в установлении контакта с потенциальными партнерами по бизнесу, поставщиками и покупателями. Уже в настоящее время весь объем рекламной и

представительской продукции, рекламные ролики на телевидении и радио, снабжаются информацией о сайте компании.

- интернет-представительство (сайт) в сети Интернет окажет положительное влияние на имидж организации: это является показателем открытости компании, ее стабильной работы.
- При использовании собственного сайта можно активным образом вводить новую услугу, использующую сеть интернет и (или) интернет-технологии, запускать новую бизнес-единицу внутри организации.
- Умело сделанные сайты помогают формировать и укреплять бренд фирмы. Благодаря интернет-сайту, может быть быстро решена задача формирования лояльной аудитории посетителей различных проектов.
- Привлечение на собственные сайты дополнительных пользователей в сети Интернет служит увеличению узнаваемости торговой марки.

2. Рекламная функция и дополнительные каналы сбыта продукции:

- В случае, когда компания продвигает товары или услуги на другой рынок (региональный или международный), то трудно реализовать более дешевый и эффективный рекламный канал для этой цели. Подробное описание на сайте продвигаемых товаров сэкономит большое количество времени и средств на предоставление необходимых данных потенциальному партнеру или клиенту.
- Каталоги товаров или описание сервисов, предлагаемых организацией на местных рынках, может стать хорошей поддержкой организации сбыта в розничных сетях. Посетители сайта, получившие подробные данные о товарах (внешний вид и их стоимость, месторасположение торговой зоны), являются уже подготовленными покупателями.

3. Сервисное обслуживание клиентов.

- Наверное, одним из главных отличий сети Интернет от других средств массовой информации является возможность интерактивности, возможности обратной связи с потребителями. Любой клиент при приобретении товара или услуги для решения своих конкретных задач или потребностей, полагает важным получать профессиональные советы и грамотные консультации. Естественно, предоставив такой сервис потребителям, компания надолго оставит его своим потенциальным покупателем и в будущем.
- Раздел Frequently Asked Questions (FAQ), перевод - "Часто задаваемые вопросы" (ЧАВО). Представление таких данных является важным не только для существующего потребителя товаров, но также и для потенциального потребителя, для убеждения его в необходимости покупки товара именно в данной фирме.

4. Поддержка региональных политик:

- Сайты могут стать местами вербовки дилеров и расширения рынка сбыта по РФ или странам СНГ. Также возможна организация информационной или рекламной поддержки существующих дилерских сетей.
- Англоязычная (или на других языках) версия сайта может стать инициатором начала переговоров с партнерами по организации внешнеэкономической деятельности. Особенно актуальным это является для компании-экспортера.

5. Организация сбора маркетинговой информации:

- Имея собственное интернет-представительство легко организовать сбор откликов на новый товар.
- Интересные и полезные ресурсы в сети Интернет всегда привлекают много посетителей. А так как любая коммерческая организация требует сведения о своем потенциальном клиенте, то при проведении на сайте сбора статистики о посетителях даст возможность обработки данной информации о посетителях, их географии. Применение сайта

для поиска и мониторинга становится важной частью информации для бизнеса (анкетирование, проведение опроса).

6. Организация вербовки кадров:

Электронное веб-приложение может стать хорошим помощником для менеджеров по персоналу, которые могут разместить в соответствующих разделах информацию о свободной вакансии. Можно с большой уверенностью полагать, что сеть Интернет стала одним из эффективных средств для сбора информации о персонале [25].

7. Уменьшение традиционных расходов:

- Наличие собственного веб-сайта предполагает организацию дополнительного сервиса поддержки. Это приводит к уменьшению применения других видов обратной связи с пользователями и, соответственно, снижению издержек затрат на её обеспечение. Кроме того, если потребитель найдет представляемый объем поддержки более качественным и полным, это будет способствовать возрастанию положительного опыта взаимодействия покупателей и организации и увеличению их взаимосвязей.
- Применение собственных сайтов дает возможность существенного снижения затрат на PR мероприятия за счет перенесения акцентов с традиционных средств, например печатного материала, на информационные формы представления в сети Интернет.

2. Обзор аналогов и требования к системе

2.1. Обзор имеющихся решений

Foursquare

Это приложение найдет для вас лучшие развлекательные заведения по всему городу. Удобный интерфейс, благодаря которому можно за считанные минуты найти, где можно провести досуг или просто выпить чашечку кофе. У приложения есть история, в ней сохраняются ваши предпочтения и исходя из них подбираются максимально подходящие заведения. Также маленькие советы от приложения подскажут, что нельзя пропустить или, что лучше заказать в определенном месте. Правдивые отзывы клиентов помогут не прогадать вам с выбором заведения [28].

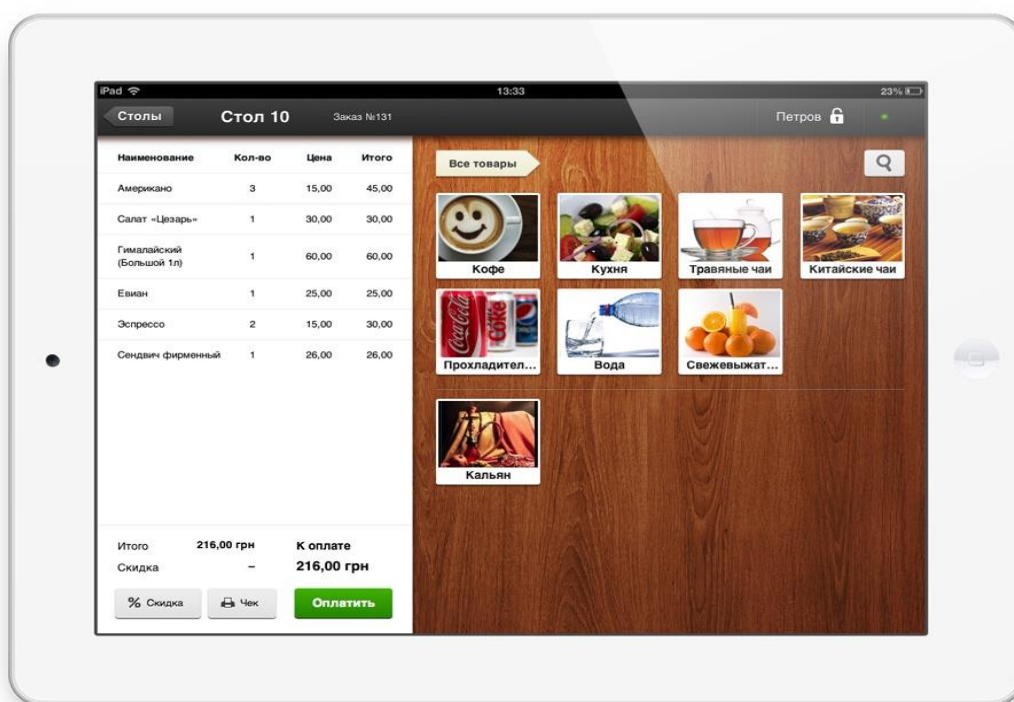


Рисунок 1. Приложение Foursquare

Resti

Сервис позволяет клиентам при помощи смартфона просматривать меню, информацию об акциях и скидках, бронировать столики, делать заказы, просить счет, оставлять отзывы. При поступлении заказа менеджер

развлекательного заведения получает оповещение на свой смартфон и может одним кликом выполнить заказ [29].



Рисунок 2. Приложение Resti

Eatsmart

Eatsmart - новый сервис онлайн-бронирования места в развлекательном заведении. В одной системе собираются все места, в которых пользователь может забронировать столик в несколько кликов и получить скидку до 50% с чека или комплимент от заведения. Сервис работает с заведениями по партнерской программе. Уже подключены около 500 мест во все крупных городах. Пользователи могут бронировать столик через сервис, приходить в развлекательное заведение и сразу же получать скидку. Есть приложение для iOS и Android. Т.е. пользователю не нужно копить баллы или распечатывать какие-то купоны, чтобы получить скидку с чека.



Рисунок 3. Приложение Eatsmart

iRECA

iRECA – это 6 мобильных приложений, упрощающих работу общепита и розницы: iRECA: Менеджер развлекательного заведения, iRECA: Меню, iRECA: Гость, iRECA: Курьер, iRECA: Solo и iRECA: Кассир. Одно из самых популярных мобильных приложений линейки - это «iRECA: Менеджер развлекательного заведения» – стремительно повышает скорость и качество работы официанта. Используя его, вы совершенно точно сможете удивить гостей вашего заведения скоростью приема и выполнения заказа. Мобильное приложение iRECA: Гость дает посетителям самостоятельно оформлять заказ. Основными преимуществами приложений являются: высокое качество разработки, простота подключения и настройки, решение всех необходимых вам бизнес-задач, повышение эффективности работы, а также усиление имиджа вашего заведения [21].



Рисунок 4. Приложение iRECA

2.2. Описание Java технологий

При решении поставленной задачи оптимальным представляется использование для представления информационных материалов язык JAVA [2]. Этот язык является языком высокого уровня и дает возможность быстрого и эффективного создания приложений.

JAVA – язык программирования, который исполняется на стороне веб-сервера. Спроектирован для создания динамических и интерактивных веб-сайтов [16].

Этот язык является достаточно гибким и мощным средством программирования, поэтому приобретает большую популярность и применяется для проектов любых масштабов: от простых блогов до крупнейших веб-приложений в сети Интернет:

Преимущества JAVA:

- представляет собой свободное программное обеспечение, распространяемое под особой лицензией (JAVA license);
- легко осваивается на всех этапах сложности;
- его поддерживают большое сообщество пользователей и разработчиков;
- обладает развитой поддержкой баз данных;
- имеет значительное количество встроенных библиотек и расширений;
- может применяться в изолированных средах;
- дает возможность использования нативных средств организации веб-сессии, программного интерфейса расширений;
- может быть полной заменой проприетарной среды ASP (Active Server Pages) от Microsoft;
- может разворачиваться практически на любых серверах;
- портируется под большое количество аппаратных платформ и операционных систем.

Для обоснования выбора языка программирования произведем сравнение JAVA с другими языками программирования для приложения. Его основными конкурентами являются – Perl, ASP .NET, ColdFusion и Java.

PHP – язык программирования, который исполняется на стороне веб-сервера. Спроектирован для создания динамических и интерактивных веб-сайтов [21].

Этот язык является достаточно гибким и мощным средством программирования, поэтому приобретает большую популярность и применяется для проектов любых масштабов: от простых блогов до крупнейших веб-приложений в сети Интернет:

Преимущества PHP:

представляет собой свободное программное обеспечение, распространяемое под особой лицензией (PHP license);

легко осваивается на всех этапах сложности;

его поддерживают большое сообщество пользователей и разработчиков;

обладает развитой поддержкой баз данных;

имеет значительное количество встроенных библиотек и расширений;

может применяться в изолированных средах;

дает возможность использования нативных средств организации веб-сессии, программного интерфейса расширений;

может быть полной заменой проприетарной среды ASP (Active Server iPages) от Microsoft;

может разворачиваться практически на любых серверах;

портируется под большое количество аппаратных платформ и операционных систем.

Для обоснования выбора языка программирования производим сравнение PHP с другими средами для разработки модуля приложения. В качестве его основных конкурентов выступают языки – Perl, ColdFusion, ASPi.NET и Java [27].

Язык Perl создавался при активном участии Ларри Уолла (Larry Wall) и его название может быть расшифровано как Practical Extraction and Report Language. Это выражение следует переводить как «язык для практического осуществления извлечения данных и составления отчетов». Perl - это язык-интерпретатор и он появляется намного позже, чем технологии Web. С созданием сетей типа Интернет, язык программирования Perl оказался достаточно удобным инструментарием для реализации динамических страниц в Web-пространстве. Большое применение языка Perl обуславливается тем, что он является доступным практически на любых Web-серверах, так как почти все они работают под управлением операционных систем типа UNIX, и в качестве альтернативы для него оказывается значительно более сложный язык программирования C [21].

Язык Perl уступает PHP в простоте реализации, учитывая, что PHP предлагается функциональность, которая является аналогичной Perl при наличии сравнительно меньшей избыточности и относительной простоте синтаксиса. Избыточность при применении языка Perl связана с тем фактом, что он разрабатывался для самых разнородных приложений, что не могло не наложить отпечаток на его синтаксис, тогда как PHP изначально разрабатывался с возможностью применения в сети Интернет.

Рассмотрим язык программирования Java.

Вообще говоря, язык Java и технология Java различаются. Язык Java является C-подобным языком, который разработан как «улучшенный C++». В технологию Java включается клиентская и серверная часть, а также доступ к базе данных, поэтому наиболее корректным будет проводить сравнение технологий Java со связкой Apache/PHP/MySQL. Технологию Java разрабатывали как межплатформенную технологию, которая дает возможность создания Web-приложений корпоративных масштабов. Основные достоинства этой технологии проявляются в межплатформенной переносимости и применении объектно-ориентированного языка, что дает возможность создания сложных и объемных приложений. К числу

недостатков относят медленное исполнение, занятие большого количества памяти (это плата за межплатформенную переносимость) и сложность разработки Web-приложения по сравнению с языком PHP. Между тем язык PHP практически не уступает языку Java в гибкости и масштабируемости созданного приложения. Кроме того, применение Java для разработки достаточно простого приложения вряд ли оправдывается, а при работе над серьезной задачей язык Java обходится дороже.

ASP (Active Server Pages) является языком сценариев от Microsoft. Если сам язык ASP значительно уступает PHP по многим параметрам, в первую очередь, по времени выполнения сценария, то с появлением технологий ASP.NET ситуация изменилась.

Основное достоинство ASP.NET заключено в возможности использования всех возможностей интерфейса ОС Windows для разработки Web-приложения. Это дает возможность разработки страниц со сложным интерфейсом, не заботясь о сохранении, шифровании и передаче переменных, которые используются на динамической странице.

ASP.NET тесно интегрируется в новую технологию NET компании Microsoft. Главная особенность разработки Web-приложения при использовании ASP.NET состоит в возможности использования любого языка программирования среды выполнения NET. Это могут быть как модифицированные варианты Visual Basic – Visual Basic.NET, так и новые языки C# или клоны Java – языки J#. Кроме того, в рамках одного Web-приложения могут быть применяться несколько разных языков программирования. Все это дает возможность объединения в рамках одного проекта программистов, которые владеют разными языками программирования. Код Web-приложений на ASP.NET компилируют в промежуточный язык MSIL (Microsoft Intermediate Language), который просто называют IL. Такой аппаратно-независимый код на языке IL помещают в кэш Web-серверов для повышения производительности, по аналогии с байт-кодом языка Java. Но в отличие от технологий Java, которая обеспечивает

платформенную независимость, iNET обеспечивает независимость от языков разработки. Выполнение Web-приложения ASP. iNET на сервере, работающем не под ОС Windows, является практически невозможным. Кроме того, если Web-приложения интенсивно использует компоненты iNET, просмотр динамических страниц будет возможен только под ОС Windows и только при помощи браузера Internet Explorer. Поэтому в смысле переносимости язык JAVA отличается от ASP.NET в лучшую сторону. Другой крайне важный аргумент в пользу выбора языка JAVA заключается в быстром исправлении обнаруженной ошибки. Теоретически исправлять их можно любым образом, если программист обладает достаточной квалификацией. Также можно написать письмо разработчикам программного обеспечения, и с большой долей вероятности, в следующей вышедшей версии ошибка будет исправлена. А версии JAVA выходят достаточно часто. С другой стороны, попробуйте написать об ошибке в компанию Microsoft и попросить ее исправить. Вам это вряд ли удастся. Еще одним плюсом JAVA, особенно важным для начинающего пользователя, является то, что его легче применять для построения Web-приложения.

2.3. Требования к системе

Область применения программного обеспечения (ПО) – персональные компьютеры.

Назначение разработки

Функциональное назначение разработки: веб-приложение для поиска развлекательных заведений.

Условия эксплуатации: приложение предназначено для эксплуатации на ПК, который подключён к локальной вычислительной сети.

Конечные пользователи: все категории пользователей.

Требования к программе

Перечень выполняемых функций

Перечень выполняемых функций:

- создание пользователей (администраторов);
- изменение логина, пароля, группы пользователь;
- удаление пользователя;
- аутентификация пользователя;
- распределение пользователей по группам;
- добавление нового развлекательного заведения;
- удаление развлекательного заведения;
- поиск развлекательного заведения в базе данных (обычный поиск, поиск по фильтрам);
- редактирование информации о развлекательных заведениях;
- формирование и печать отчётов.

Требования к надёжности

Программное средство должно обеспечивать контроль корректности входных данных. В случае обнаружения ошибок во входных данных пользователю должны выводиться соответствующие сообщения.

К надёжности программы предъявляются требования соблюдения целостности данных и корректности вносимых данных.

Условия эксплуатации

Данное программное средство не требует предварительной подготовки персонала.

Требования к составу и параметрам технических средств

Минимальными и достаточными требованиями по конфигурации оборудования для функционирования системы являются:

- процессор IntelPentium с частотой не менее 1000 МГц;

- объем оперативной памяти не менее 512 МБ;
- свободное дисковое пространство не менее 1 ГБ;
- Наличие сетевого адаптера, подключенного к сети.

Требования к информационной и программной совместимости

- Операционная система: Windows XP, Vista, Windows 7, Windows 10;
- Microsoft SQL Server 2012

Требования к маркировке и упаковке. Требования к транспортированию и хранению

Требования к маркировке и упаковке не предъявляются.

Требования к транспортированию и хранению не предъявляются.

Согласно заданию на дипломное проектирование программная документация должна включать следующие документы:

- техническое задание в соответствии с ГОСТ 19.201-78 ЕСПД;
- описание программы в соответствии с ГОСТ 19.402-2000 ЕСПД;
- руководство оператора в соответствии с ГОСТ 19.505-79 ЕСПД.
- программа и методика испытаний в соответствии с ГОСТ 19.301-2000 ЕСПД.

Экономическая эффективность от внедрения данной разработки должна быть обеспечена за счет:

- уменьшения затрат на приобретение и хранение бумажных копий книг;
- уменьшения физической работы с бумажными носителями;
- повышение удобства получения информации о книгах и авторах.

Этапы разработки:

- постановка задачи;
- анализ исходных данных;

- разработка программного средства;
- тестирование программного средства;
- разработка технической документации к данному программному средству.

Приемка готового программного средства осуществляется комиссией по защите преддипломной практики, составленной из преподавателей кафедры технологий программирования в соответствии с программой и методикой испытаний.

3. Программное web-приложение

3.1. Архитектура и реализация

Одним из основных понятий методологии проектирования ИС является жизненный цикл программного обеспечения (ЖЦ ПО). ЖЦ ПО – это постоянный процесс, начинающийся, когда принимается решение о необходимости его разработки, и заканчивающийся, когда его полностью изъяли из эксплуатации.

Для реализации данной ИС была выбрана технология RAD.

Rapid Application Development (RAD) – это ЖЦ процесса создания проекта для достижения наивысшей скорости проектирования и качества программы, что обуславливает возможность при традиционном подходе к разработке.



Рисунок 5. Концепция создания средств разработки программных продуктов.

Особенностью RAD является незначительное время перехода от определения требований до создания полноценной системы. Метод основывается на последовательности шагов эволюционной системы, критический анализ которых обговаривается с заказчиком. В процессе такого анализа появляются требования к продукту [21].

Создание каждого интегрированного продукта ограничивается определенным периодом времени, который, как правило, составляет 2 месяца и называется временным блоком.

Факторы, которые позволяют создать систему за 2 месяца, причем без урона качеству, включают в себя использование мощных инструментальных средств разработки, высокий уровень фактора повторного использования, а также осмысленные и выделенные ресурсы.

Фазы модели RAD

В модели RAD выделяются определенные фазы:

- Фаза планирования требований к программному обеспечению — определение требований выполняются при использовании рабочих методов, которые называются совместным планированием (Joint requirements planning, JRP). Эти методы представляют собой структурный анализ и обсуждение имеющейся коммерческой задачи;
- Описание пользователей. Проводится совместное проектирование программного обеспечения (Joint application design, JAD) с целью привлечения клиентов. На этом этапе разработки систем, которые не являются промышленными, команды проектировщиков часто применяют автоматизированные инструментальные средства, для обеспечения сбора пользовательской информации;
- Фаза конструирования ("до полной готовности"). На этом этапе объединяется детализированное проектирование, построение (создание кода и проведения тестирования), а также поставка программных продуктов заказчику в определенный срок. Период выполнения этого этапа в большой степени будет зависеть от применения генератора кода, экранного генератора и других инструментальных производственных средств;
- Этап перевода на новые системы эксплуатации. На этом этапе проводятся приемочные испытания, установка систем и обучение пользователей.

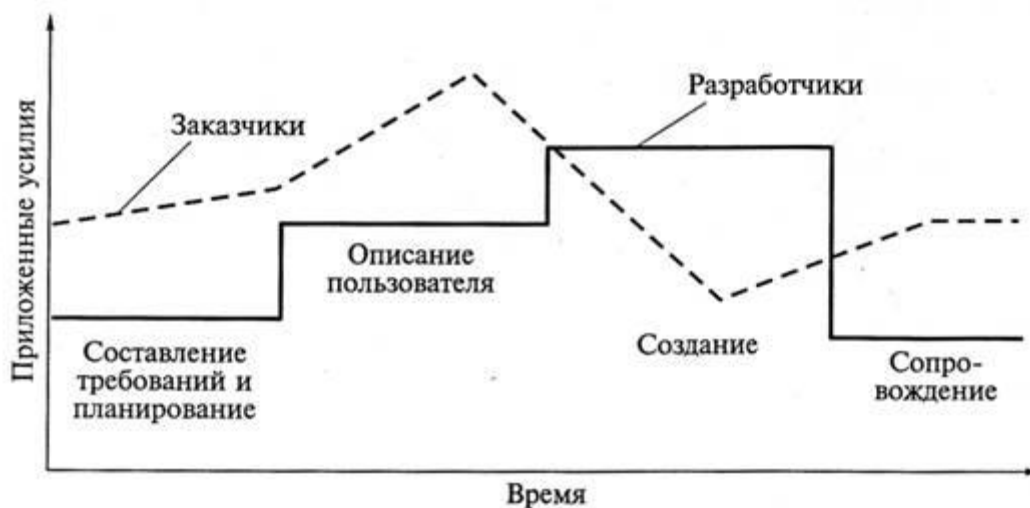


Рисунок 6. Модель быстрой разработки программного обеспечения

RAD предполагает, что разработка программного обеспечения осуществляется небольшой группой разработчиков за срок порядка 3-4 месяцев при применении методов инкрементного прототипирования и инструментальных средств для визуального моделирования и разработки. В технологии RAD предусматривается активное привлечение заказчиков уже на ранних стадиях разработки – обследование предприятий, выбор требований к разрабатываемой системе. Причины популярности технологии RAD состоят в следующих преимуществах, которые может обеспечить применение этой технологии.

Основные преимущества RAD:

- высокие скорости разработки;
- небольшая стоимость;
- высокое качество продуктов.

Последнее из указанных свойств говорит о том, что должны полностью выполняться требования заказчика как функциональные, так и нефункциональные, с учетом их возможных изменений во время разработки системы, а также создание качественной документации, которая обеспечивает удобство эксплуатации и сопровождения систем. Это значит, что дополнительные затраты на сопровождение программных продуктов сразу

после поставки клиенту будут значительно уменьшены. Вследствие этого, полное время от начала разработки программного продукта до получения приемлемого результата при выборе этого метода будет значительно сокращено.

В работе программы используются стандартные алгоритмы работы с базой данных. Все основные алгоритмы работы это ввод данных от пользователей, контроль вводимой информации на предмет нарушения целостности данных и добавление вводимой информации в базу данных, если вводимые данные не нарушают целостности базы данных.

Работа приложения начинается с авторизации пользователя и последующим переходом в главное меню. Пользователь имеет также возможность перейти на другие формы, которые необходимы в работе с данными.

Взаимодействие с пользователем осуществляется посредством экранных форм.

Приблизительный алгоритм работы с базой данных (в данном случае при вводе информации) представлен на рисунке ниже. Алгоритмы по редактированию данных и занесению их в базу, а также алгоритмы, осуществляющие удаление информации из базы данных также являются стандартными.

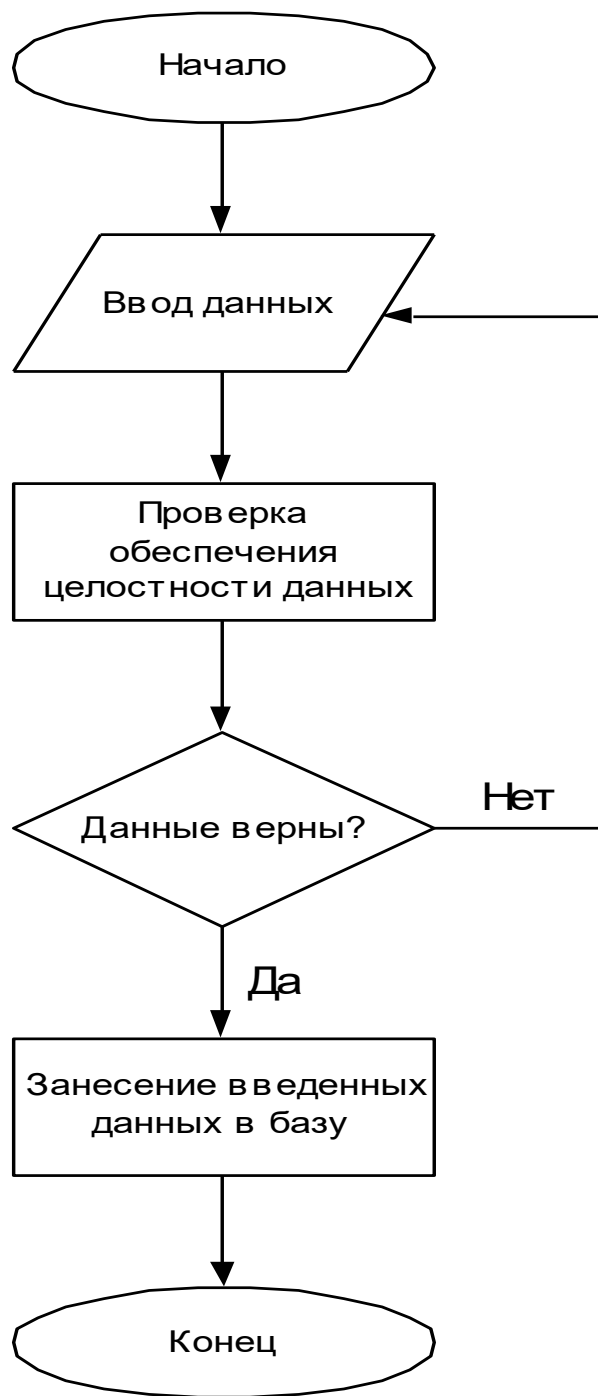


Рисунок 7. Алгоритм ввода данных в базу.

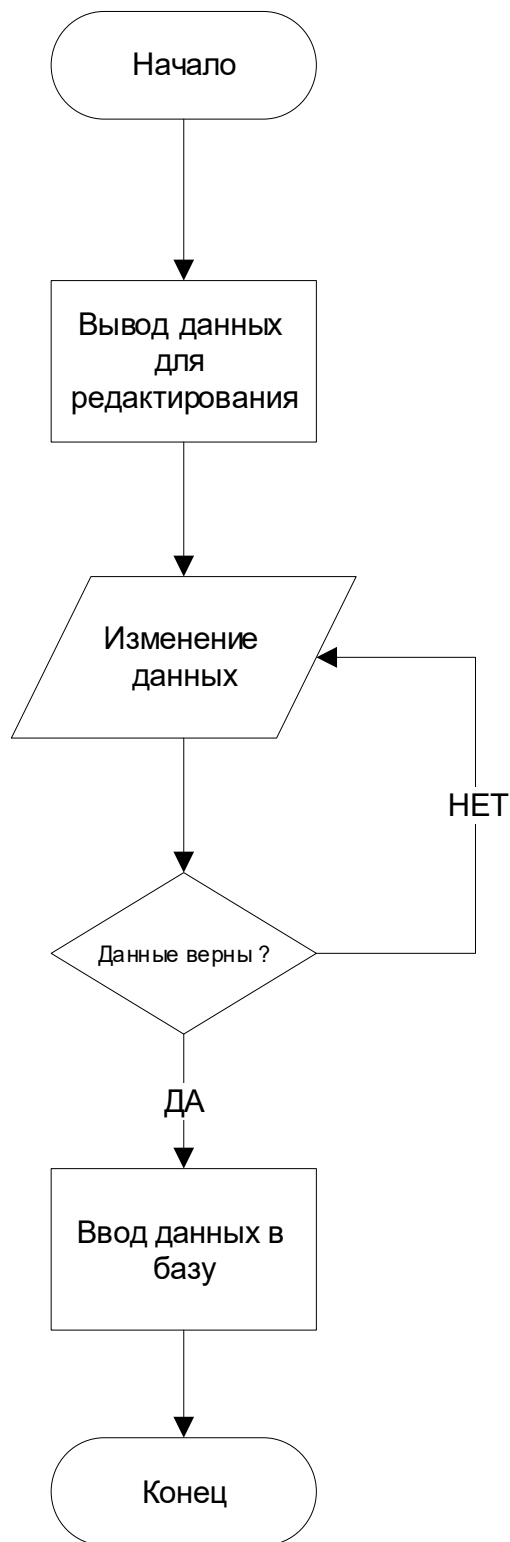


Рисунок 8. Алгоритм редактирования данных в базе.

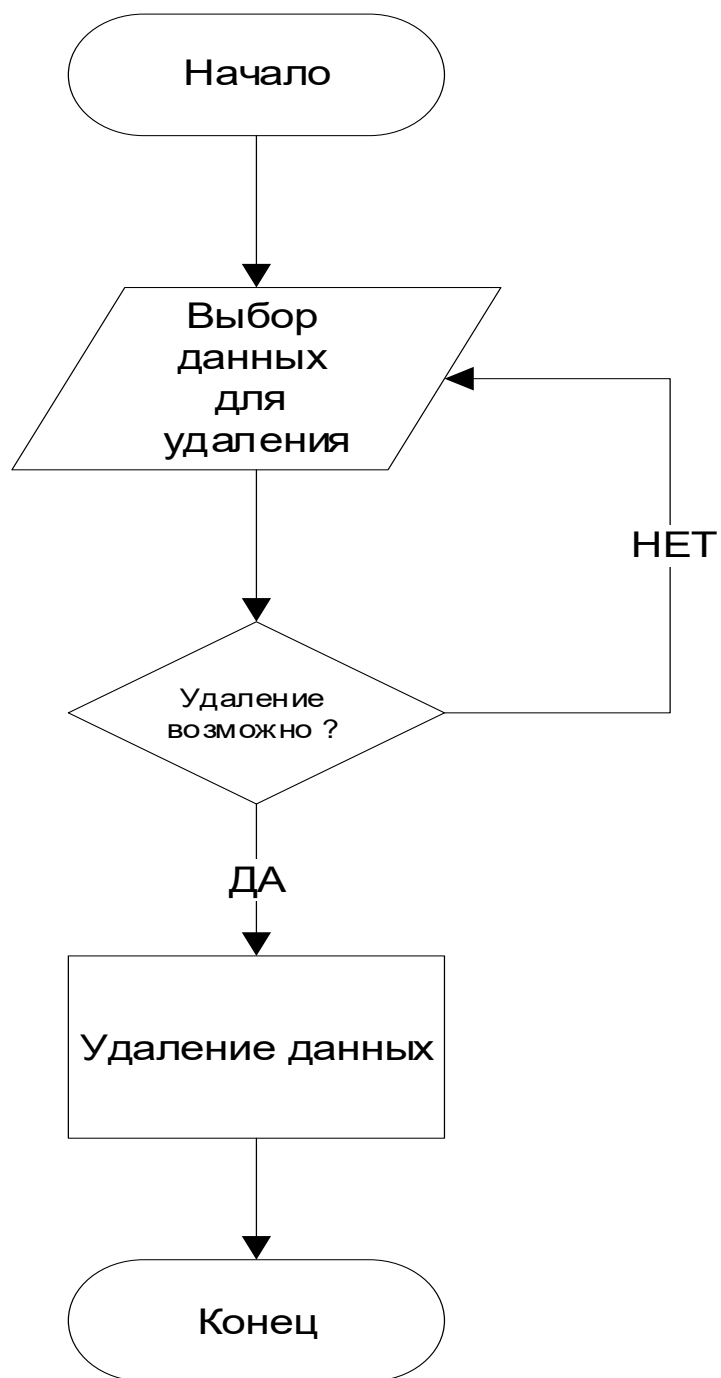


Рисунок 9. Алгоритм удаления данных из базы.

Рассмотрим алгоритм формирования выходных документов.

Все документы формируются на основе шаблонов и алгоритм их заполнения представлен на рисунке 9.

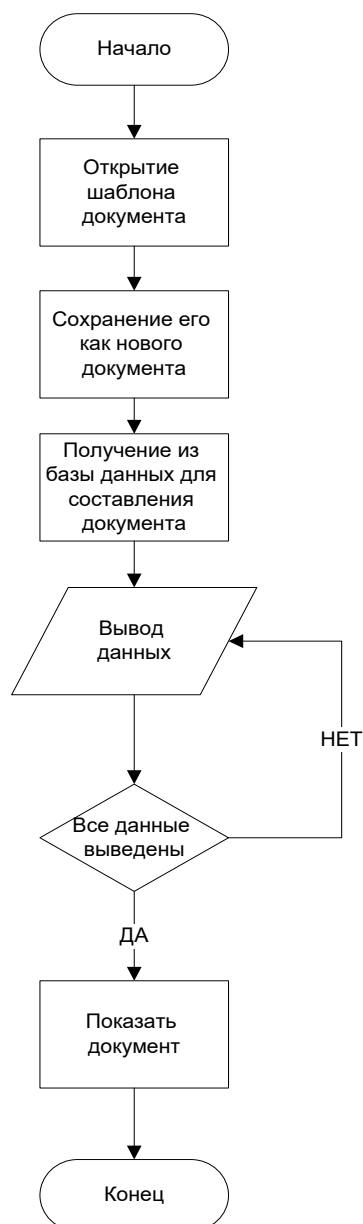


Рисунок 10. Алгоритм формирования выходных документов.

После разработки алгоритмов работы программы, выбора инструмента разработки и методологии началось написание программного кода.

ИС предназначена для работы на IBM – совместимых персональных компьютерах. Компьютер должен иметь:

- процессор Intel Pentium IV и выше;
- ОЗУ 2 Убайт и выше;
- жесткий диск (около 20 Гбайт);
- печатающее устройство;
- VGA–совместимый дисплей (рекомендуется SVGA–дисплей).

Наполнение рабочего места

Разработка дизайна приложения для поиска развлекательных заведений очень важная составляющая, от которой зависит многое. Если дизайн не будет привлекательным, интуитивно понятным, оригинальным и удобным, то посетители не будут задерживаться на сайте, не станут его постоянными клиентами, что в конечном итоге негативно отразится на рейтинге. Пользовательский поведенческий фактор очень важен для продвижения проекта, и он зависит не только от наполнения приложения для онлайн чтения книг из специализированной библиотеки, т.е. содержательной части, но и от оформления, компоновки, структуры.

Стоит обозначить главные моменты, которых стоит избегать при разработке дизайна:

1. слишком яркие цвета на странице, вызывающие рябь и дискомфорт в глазах;
2. слишком высокая, плохо подобранная контрастность, вызывающая неприятные ощущения и раздражение;
3. слишком низкая контрастность, затрудняющая чтение и восприятие материала;
4. неоправданно широкая цветовая палитра, мешающая концентрации внимания;
5. слишком сложная композиционная структура, не соответствующая замыслу, основной направленности и с низкой эргономикой;
6. интуитивно непонятный интерфейс, рассчитанный не на рядового посетителя;
7. плохо продуманная, вводящая пользователя в заблуждение навигация;
8. слишком большие или слишком маленькие размеры шрифтов, элементы форм, затрудняющие восприятие и взаимодействие пользователя;

9. неоправданное многообразие используемых шрифтов и их разновидностей, мешающее концентрации внимания и восприятию содержания.

Разработка дизайна требует творческих умений, а также чувство выдержки и стиля. Но в этом процессе нет ничего непостижимого. Обозначим основные моменты, от которых зависит разработка успешного дизайна сайта:

1. простота структуры и лаконичность композиции;
2. удобный эргономичный интерфейс и навигация;
3. разумный минимализм в цветовой палитре;
4. строгость и рациональность в использовании шрифтов;
5. обязательное чувство меры и вкуса при подборе графических элементов оформления и изображений;
6. неуклонное соблюдение фирменного стиля;
7. эксклюзивность разработанных атрибутов бренда – логотипа, цветовой схемы, шрифтового набора.

Разработка интерфейса, продумывание навигации, композиции, осуществляется на листе бумаги, а затем отрисовывается в программе Adobe Photoshop. Все элементы должны быть разделены по слоям и иметь четкую, понятную структуру. Лучшим образом их стоит разделить по каталогам, для облегчения дальнейшей верстки. Именовывать основные каталоги предлагается в соответствии с композиционными элементами шаблона: «head», «body», «footer» – как основные теги страницы.

В каталоге необходимо создавать подкаталоги, для того, чтобы при верстке можно было быстрее и проще разобраться и найти нужный элемент. Каждому слою необходимо давать осмысленное, понятное название. Правильную структуру слоев при отрисовке дизайна в Adobe Photoshop можно увидеть на рисунке 10.

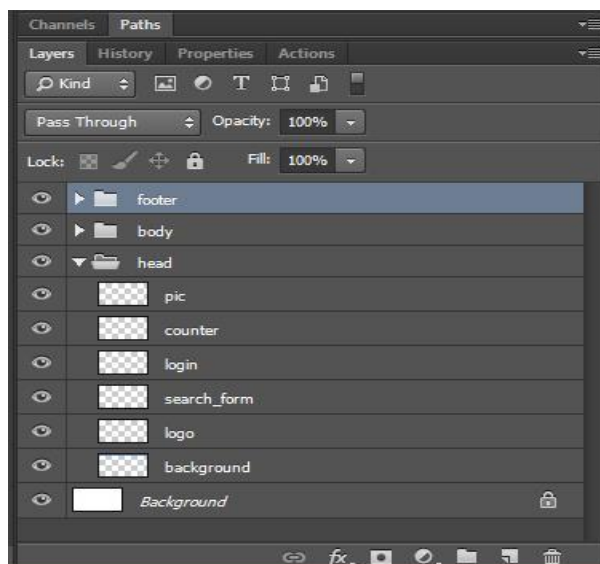


Рисунок 11. Правильная структура слоев в Adobe Photoshop при отрисовке макета

Именно такая структура была использована при разработке дизайна проекта.

Так же стоит отметить, что необходимо использовать направляющие в программе Adobe Photoshop, которые помогают точно выставить и выровнять элементы на странице и не сместить в сторону какой-либо из объектов.

Сетку из основных направляющих рекомендуется сделать в самом начале, перед отрисовкой дизайна, а затем, по мере необходимости добавлять вспомогательные направляющие. Сетку направляющих, а также макет главной страницы приложения для онлайн чтения книг из специализированной библиотеки, нарисованный в программе Adobe Photoshop можно увидеть на рисунке 11.

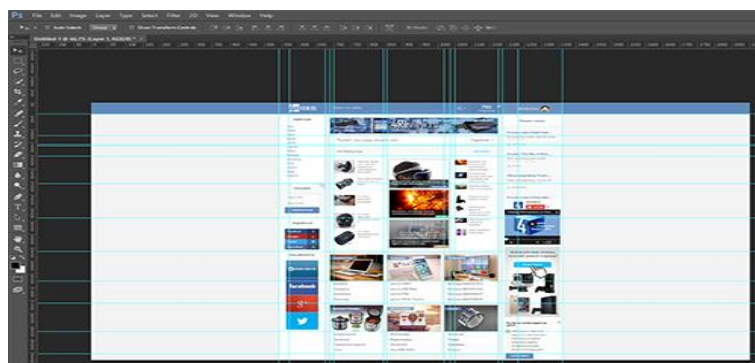


Рисунок 12. Сетка направляющих на макете главной страницы

Таким образом, необходимо отрисовать каждую страницу, на которой возможны изменения. Обычно для CMS рисуются следующие страницы:

1. главная страница;
2. страница каталога;
3. страница с кратким содержанием;
4. страница с полным содержанием;
5. страница поиска.

Все макеты страниц необходимо сохранять в формате .psd для того, чтобы при верстке можно было легко найти нужный элемент и использовать его.

После этого, основная работа в программе Adobe Photoshop закончена, и можно переходить к верстке макета под CMS DLE. Для этого понадобится удобный редактор программного кода. Для этого был выбран Sublime Text. Этот редактор удобен тем, что он может производить подсветку тегов, выдавать интерактивные подсказки, предлагающие теги на выбор, находить и исправлять теги, которые написаны с ошибками. Функционал этого редактора богат и разнообразен. Это очень помогает в работе.

Процесс верстки довольно рутинный и длительный, но, тем не менее, довольно важный, ведь если допустить ошибку, то весь макет сайта может отобразиться не верно. Так же необходимо знать HTML и CSS, чтобы правильно сверстать страницу.

HTML (англ. HyperTextMarkupLanguage — «язык гипертекстовой разметки») — стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержит описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.

CSS (англ. Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью

языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL.

Таким образом, при помощи HTML создается каркас страницы и разделение областей, а при помощи CSS производится оформление страницы, задаются цвета, стили шрифтов, задний фон, картинки и т.д.

Основной работой при верстке является именно проектирование каркаса, ранее отрисованного в Adobe Photoshop, а затем формирование стилей, которые делают страницу идентичной ранее нарисованной в графической программе. Наличие CMS значительно упрощает верстку, т.к. в движке уже есть некоторые заранее прописанные стили. Большая часть файлов берется из стандартного шаблона CMS, а затем верстаются основные страницы, написанные выше, и прописываются специальные теги CMS DLE, которые указывают где использовать поиск, авторизацию, комментарии и многое другое.

Расписать детально весь процесс верстки и отрисовки дизайна, не представляется возможным, поскольку это составит большое количество страниц и объём, значительно превышающий установленные рамки. Поэтому приведены разъяснения ключевых моментов и описаны в общих чертах основные этапы создания дизайна сайта. Детально процесс верстки и специфику работы в программе Adobe Photoshop следует изучать по тематическим книгам или на специализированных сайтах в интернете.

3.2. База данных

Такой продукт, как СУБД для Хранилища, требует целого набора свойств для управления значительными объемами информационных данных и сложными моделями данных, независимыми от конкретных приложений. Подобные особенности получаются только при наличии большого опыта в области внедрения продукта в эксплуатацию, а также четкого понимания требований пользователей.

Результаты сравнения СУБД сведены в таблицах 1 и 2.

Таблица 1 - Сравнение существующих СУБД

Критерии	Microsoft SQL Server 2008 [11]	Microsoft Access [11]	Microsoft Visual FoxPro [11]	MySQL [12]
Скорость в многопользовательском окружении.	5	1	1	4
Наличие транзакций.	5	1	0	5
Наличие триггеров.	5	1	0	5
Наличие хранимых процедур.	5	0	0	5
Наличие онлайн-овых «бекапов»	5	0	0	5
Поддержка производителя.	5	5	5	0
Удобство разработки ПО на основе СУБД.	5	2	4	3
Надежность.	5	3	4	5
Работа с большими объемами данных.	5	0	5	4
Масштабируемость.	5	0	3	5

Таблица 2 - Характеристика существующих СУБД

Название продукта	Microsoft SQL Server	Microsoft Access	Microsoft Visual FoxPro	MySQL
Основные преимущества	Отличная степень защиты данных. Мощный инструмент для работы с данными, а также высокая производительность.	Простота освоения. Возможность использования новичком-программистом. Имеет мощные инструменты подготовки отчетов из БД в различные форматы.	Высокий уровень объектной модели. Высокая скорость обработки данных. Интеграция объектно-ориентированного языка программирования с Xbase и SQL. Многоплатформенность.	Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.
Основное назначение	Хранение больших массивов данных. Хранение данных, требующих соблюдения режима секретности или при не допустимости их потери.	Создание отчетов произвольной формы на основании различных данных. Разработка не коммерческих приложений.	Создание приложений масштаба предприятия. Создание приложений для работы на различных платформах (Windows, Linux и т. д.)	Является решением для малых и средних приложений. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты.

В качестве СУБД была выбрана СУБД My SQL. Доступ к СУБД осуществляется посредством объектов доступа к данным Microsoft ADO (ActiveX Data Object), что дает возможность унифицированного доступа к СУБД с использованием различных типов поставщиков данных.

3.3. Функциональные возможности

Работу приложения моделируем при помощи диаграмм IDEF0 и ER.

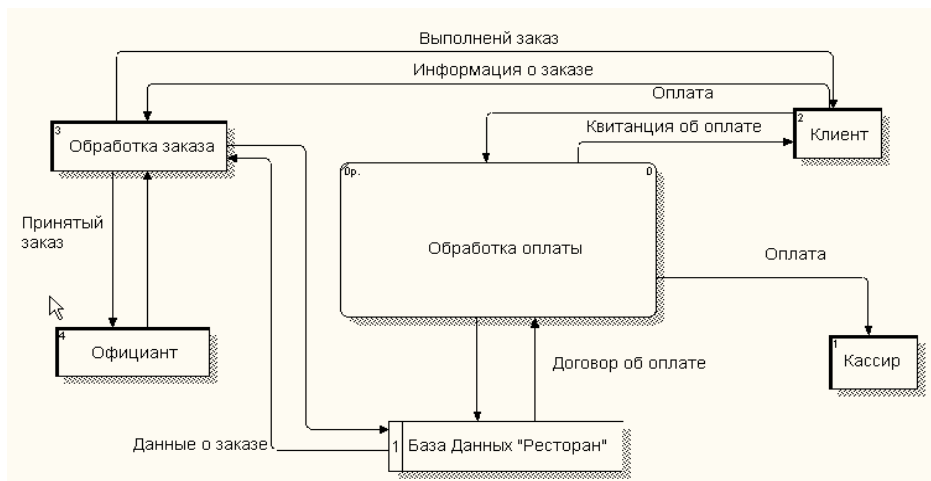


Рисунок 13. Диаграмма IDEF0 функциональной модели

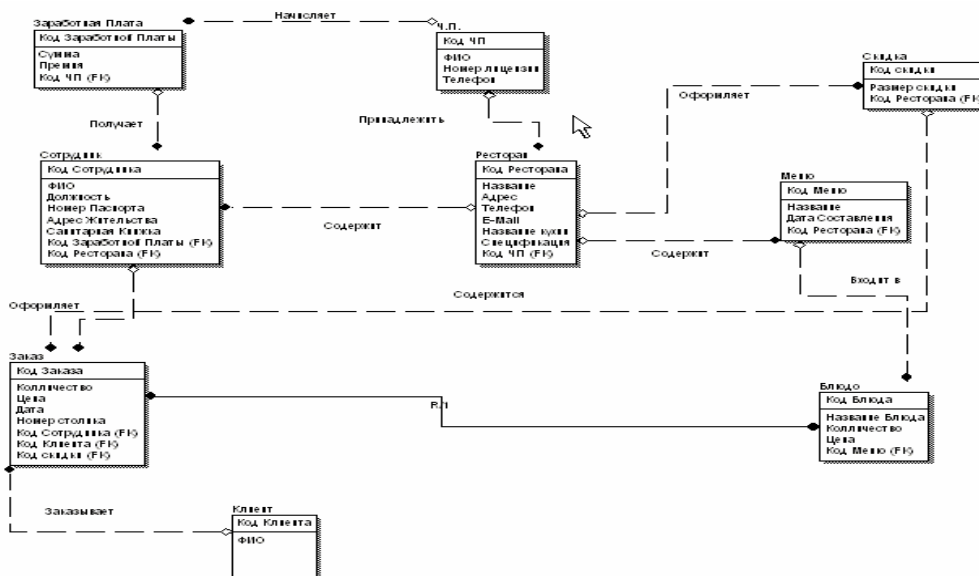


Рисунок 14. ER-диаграмма

Функциональные возможности модуля.

- Управление файлами, размещаемыми во внутреннем хранилище и относящимися к выбранной каталожной записи. Загрузка файла на сервер,

изменение его имени, удаление файла из хранилища, скачивание на клиентский компьютер.

- Управление ссылками на внешние ресурсы, относящиеся к выбранной каталожной записи. Создание, изменение и удаление ссылки.
- Поддержка любых форматов контента. Ресурсы в форматах, интерпретируемых браузером, воспроизводятся его средствами, прочие ресурсы загружаются на клиентский компьютер.
- Операции с логическим ресурсом: создание, редактирование, удаление.
- Выбор файлов или ссылок, включаемых в логический ресурс, определение их порядка.
- Установка атрибутов логического ресурса: способ доставки (загрузка на клиентский компьютер, отображение в браузере), способа представления (защищенный режим) и видов информации (текст, изображение, видео, аудио, программа, массив данных).
- Операции с пакетом: создание, редактирование, удаление. Установка порядка пакетов в контексте каталожной записи.
- Описание пакета: выбор категории пользователей, логических ресурсов и параметров доступа к ним.
- Поддержка выборочного доступа к контенту: для многокомпонентных ресурсов на основе количества или доли доступных компонентов, для многостраничных документов на основе количества или доли доступных страниц.
- Реализация защищенного режима отображения контента (pdf, txt, jpg, html), исключающего его сохранение на жестком диске компьютера пользователя.
- Управление правами доступа читателя к контенту электронной библиотеки в соответствии с категорией пользователя и сформированными описаниями пакетов и ресурсов.
- Фильтрация каталога по атрибутам электронной библиотеки.

- Полнотекстовый поиск по контенту электронной библиотеки с учетом морфологии русского и английского языков, совместного вхождения в текст словоформ из запроса, расстояний между словами и предложениями и других условий.

- Индексирование документов, размещенных во внутреннем хранилище электронной библиотеки, и используемых в ней Интернет-ресурсов, идентифицируемых по ссылкам на них.

Ведение журнала пользования электронными ресурсами, в котором фиксируются записи о доступе пользователей к пакетам контента.

3.4. Тестирование

После разработки приложения проведено его тестирование. Процесс продемонстрируем на скриншотах.

Афиша

Всегда свежая и актуальная информация о кинотеатрах и фильмах, выставках и музеях, театрах и спектаклях, клубах и вечеринках.

Места, где можно вкусно поесть

Все кафе и рестораны, бары и кофейни, фаст-фуд и новые заведения, завтраки и бизнес-ланчи.

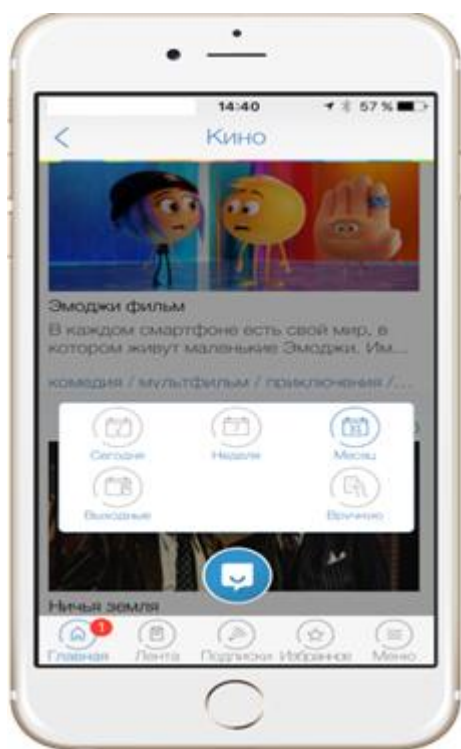
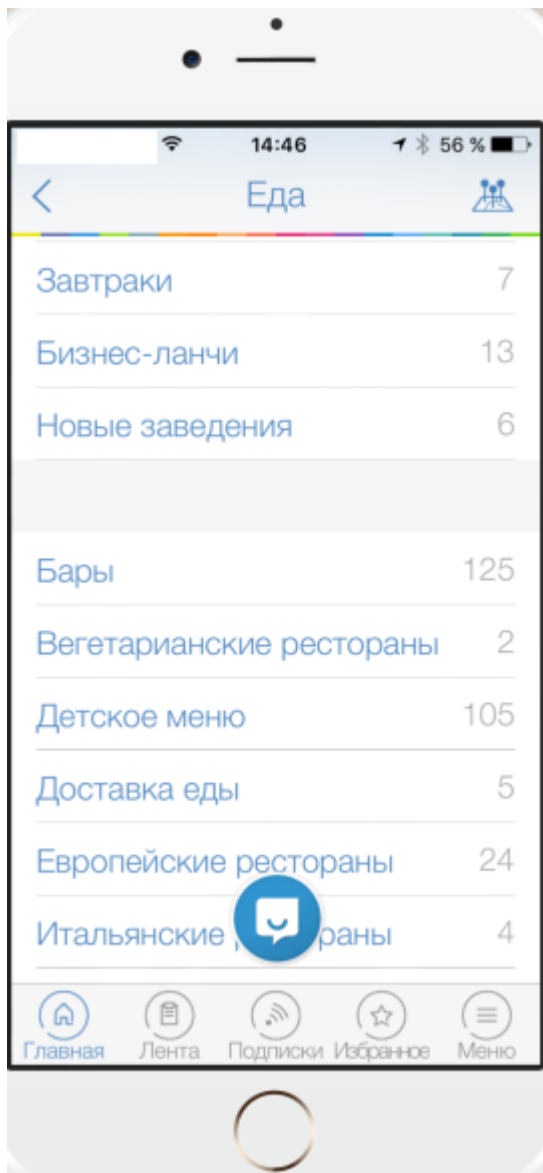
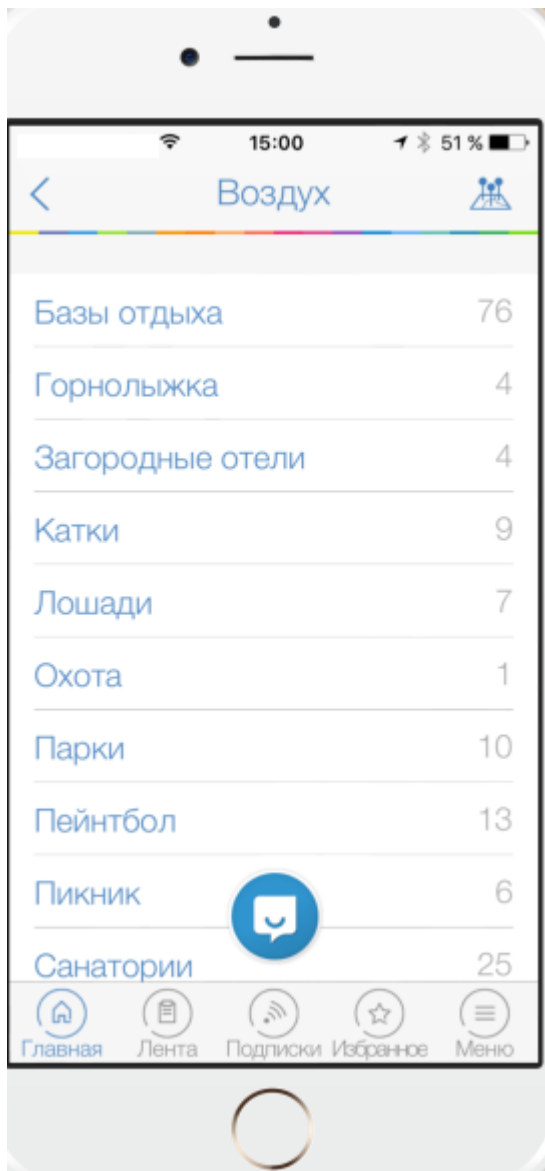


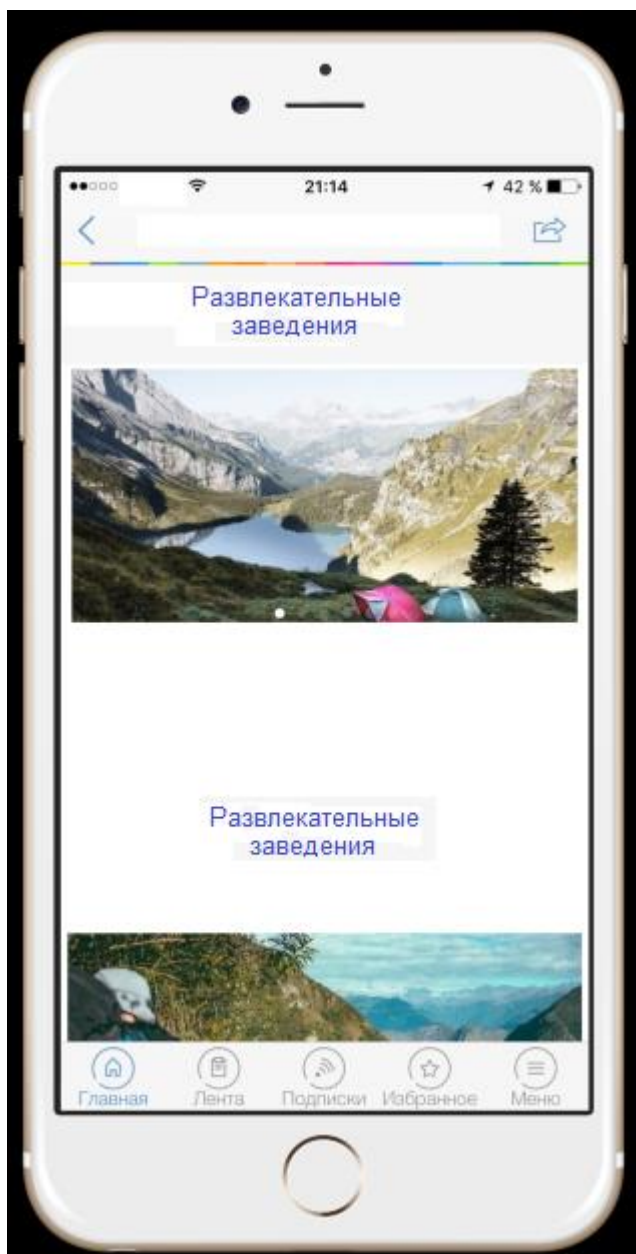
Рисунок 15. Экран приложения



Рисунок 16. Главное меню







Заключение

В ходе работы над проектом была достигнута цель работы – разработано и реализовано клиент-серверное приложение для поиска развлекательных заведений с поддержкой произвольных веб-ресурсов. Разработано решение для создания приложения на базе открытого программного обеспечения. Оно имеет весь необходимый функционал при использовании традиционного ПК, тонкого и нулевого клиента. Преимуществом предложенного решения является простота и прозрачность интеграции с открытыми платформами. Это создает реальные условия для дальнейшей удобной работы. Подтверждена актуальность темы исследования. Автоматизация поиска развлекательных заведений является очень востребованной задачей.

Роль web-приложений в современном обществе рассмотрена в первой главе работы. Во второй главе проведен обзор аналогов разрабатываемой системы и определены требования к ней. Также описаны Java-технологии для разработки таких приложений. Глава 3 посвящена разработке программного web-приложения для поиска развлекательных заведений. Рассмотрены архитектура и реализация системы, создана база данных, определены функциональные возможности приложения и проведено тестирование его работоспособности. В заключении обобщаются результаты работы и делаются выводы.

При написании работы использованы источники отечественных и зарубежных авторов. Из недостатков следует отметить недостаточный объем предоставления данных в разработанной системе. Этот недостаток может быть исправлен при дальнейшем усовершенствовании системы. Кроме того, в процессе работы подсчитаны экономические характеристики разработки и рассмотрены вопросы безопасности жизнедеятельности. Таким образом, можно сделать вывод, что поставленные в проекте задачи были решены.

Список использованных источников

1. Паронжанов С.А. Объектно-ориентированные средства анализа, проектирования и реинжиниринга информационных систем/ С.А. Паронжанов– М.: Учебные материалы конференции «Индустрия программирования 96». 2009 г. с.117-123.
2. (Strange K.). "Магический квадрат" СУБД для Хранилищ данных/ К.Стрэндж - (Magic Quadrant for Data Warehouse DBMSs/). Gartner Research Note M-22-2154. 2004.
3. Новости о мобильных устройствах и технологиях [Электронный ресурс]. – Режим доступа: URL: http://naviny.by/rubrics/computer/2005/11/13/art_12. – Название с экрана.
4. Блог о гаджетах: новости, статьи, заметки [Электронный ресурс]. – Режим доступа: URL: http://telnews.ru/Nadezhda_Balovsyak/c101972. – Название с экрана.
5. Системы видеонаблюдения, контроля доступа, охранные сигнализации и домофоны (видеодомофоны), видеорегистраторы [Электронный ресурс]. – Режим доступа: URL: <http://forward.lg.ua/pages/control.html>. – Название с экрана.
6. Журнал “КомпьютерПресс”: (Многообразие сенсорных дисплеев) / С.А. Асмаков // Б.М. Молчанов — 2015. — №8. — Режим доступа к журналу: <http://www.compress.ru/archive.aspx>.
7. Беленков В.Д. Электронные системы идентификации подписей. Защита информации. Конфидент. 200, №6, с. 39-42.
8. Иванов А.И., Сорокин И.А. Автоматическая система идентификации личности по динамике подписи. // Новые промышленные технологии. №6, 2013, с. 56-63.
9. Справочник по охране труда на промышленном предприятии / К.Н. Ткачук, Д.Ф. Иванчук, Р.В. Сабарно, А.Г. Степанов – К.: Тэхника, 2011.
10. Лебедев О.Н. Микросхемы памяти и их применение. – М.: Радио и связь, 2016. - 160 с.

11. Дантемманн Д. Программирование в среде Delphi – К.: НИПФ “ДиаСофт Лтд.”, 2015. – 608 с.
12. Джеймс. Р. Грофф, Пол Н. Вайнберг. SQL: полное руководство, пер с англ. Е.П. Курина. – К.: Издательская группа BHV, 2010. – 608с.
13. Epsilon Technologies. Введение в Delphi., урок 32. - <http://www.citforum.ru/programming/32less/les32.shtml>
14. Проектировка, разработка и продажи систем контроля доступа. - <http://www.perco.ru>
15. Олифер Н.А. Использование сетевых технологий. - http://www.citforum.ru/operating_systems/sos/glava_4.shtml#_1_4_3.
16. Справочная система помощи по Windows .
17. Болски М. И. Delphi: компоненты для работы с базами данных./ Пер. с нем. Денисенко С. В. - К.: Издательская группа BHV, 2014. - 200 с
18. Гильманов А.А., Клименко А.Я., Странгуль О.Н., Тарасенко В.П. Карточные технологии в автоматизации маркетинга. – Томск: Издательство НТЛ, 2014. –380 с.
19. Гамильтон Б. ADO Сборник рецептов для профессионалов./ Б. Гамильтон -. М.: Вильямс, 2011 г. с.127-183.
20. Руссинович М.Д. Внутреннее устройство Microsoft Windows/ М. Руссинович, Д. Соломон- М.: Экзамен, 6-е издание. 2013.
21. Дейт К. Дж. Введение в системы баз данных/ К. Дж. Дейт– М.: Вильямс, 1999.
22. Джеффри Д. Ульман. Введение в системы баз данных/ Д. Джеффри, Дженнифер Уидом. Ульман,– М.: Лори, 2000.
23. Бойко В.В., Савинков В.М. Проектирование баз данных информационных систем. – М.: Финансы и статистика, 2009.
24. Наумов А.Н. Системы управления базами данных и знаний: Справ.// А.Н. Наумов, А.М. Вендров, В.К. Иванов. Изд. / Под ред. А.Н. Наумова. – М.: Финансы и статистика, 1991. –352 с.: ил.

25. Волгина Н.А Экономика труда//Н.А. Волгина, Ю.Г. Одеговага – М.: Экзамен, 2006.
26. Сеппа Д. Microsoft ADO.NET / Д. Сеппа – М.: Финансы и статистика, 2011.
27. Интернет сайт СУБД MySQL. Режим доступа: [http:// www.mysql.com /](http://www.mysql.com/) . - Дата доступа : 25.01.2018.
28. Международные стандарты, поддерживающие жизненный цикл программных средств. М., МП «Экономика», 1999
29. Елманова Н.А. Системы управления базами данных ведущих производителей/ Н.А. Елманова КомпьютерПресс, июль 2005.
30. Кодд Е.Ф. Реляционное представление данных в хранилищах данных/ Е.Ф. Кодд - (A Relational Model of Data for Large Shared Data Banks)
31. СУБД Microsoft FoxPro . Режим доступа: [http:// Microsoft.com/FoxPro/](http://Microsoft.com/FoxPro/) /. - Дата доступа: 25.01.2018.
32. Интернет сайт технологии доступа к данным ODBC. -. Режим доступа: [http:// http://www.iodbc.org/](http://www.iodbc.org/) /. - Дата доступа: 25.01.2018.
33. Интернет сайт технологии доступа к данным Hibernate. [http:// http:// http://www.hibernate.org](http://www.hibernate.org/) // . - Дата доступа: 25.01.2018.
34. В Уилтон П. SQL для начинающих/ Пол Уилтон; Джон Колби - М.: Вильямс, 2010

Приложение А

Листинг программы

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
ComCtrls, Menus, StdCtrls, IniFile, ToolWin, Buttons, BaseFile, Grids, NewNaklad,
MyTypes, GetText, CheckLst, ExtCtrls, DataShow, ReData, Options, DBClient,
MConnect, MidasCon, QRPntr, SaleShow, Clients, SaleOrder, Kurs, Print,
About, SaleChange, Progress, Schet, DateFolder, Reserv, Count;

type

TMainForm = class(TForm)
 CategoryList: TListBox;
 ToolBar1: TToolBar;
 btExit: TSpeedButton;
 btNew: TSpeedButton;
 btPrice: TSpeedButton;
 btUchet: TSpeedButton;
 btOptions: TSpeedButton;
 btSale: TSpeedButton;
 btIncom: TSpeedButton;
 btDel: TSpeedButton;
 btAdd: TSpeedButton;
 btOpen: TSpeedButton;
 SaveNewBase: TSaveDialog;
 OpenBaseDialog: TOpenDialog;
 Bevel2: TBevel;
 ItemBox: TListBox;
 ChangePopup: TPopupMenu;
 N1: TMenuItem;

N2: TMenuItem;
N3: TMenuItem;
N4: TMenuItem;
OutButton: TBitBtn;
PlusButton: TBitBtn;
MinusButton: TBitBtn;
ClearButton: TBitBtn;
CategoryPopup: TPopupMenu;
N5: TMenuItem;
N777: TMenuItem;
Panel2: TPanel;
Panel1: TPanel;
RoznicaGrid: TStringGrid;
GroupBox3: TGroupBox;
Label5: TLabel;
BaksEdit: TEdit;
RubEdit: TEdit;
NaklLabel: TEdit;
ClientEdit: TEdit;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
btUp: TSpeedButton;
btDown: TSpeedButton;
CatUp: TSpeedButton;
CatDown: TSpeedButton;
Bevel4: TBevel;
Bevel1: TBevel;
Bevel3: TBevel;
Bevel5: TBevel;
SchetBtn: TBitBtn;

ToRes: TMenuItem;
ReservBtn: TBitBtn;
procedures mbExitClick(Senders: TObject);
procedures btExitClick(Senders: TObject);
procedures btNewClick(Senders: TObject);
procedures btAddClick(Senders: TObject);
procedures FormActivate(Senders: TObject);
procedures btOpenClick(Senders: TObject);
procedures CategoryListClick(Senders: TObject);
procedures btDelClick(Senders: TObject);
procedures btIncomClick(Senders: TObject);
procedures N1Click(Senders: TObject);
procedures ItemBoxKeyPress(Senders: TObject; var Key: Char);
procedures CategoryListKeyUp(Senders: TObject; var Key: Word;
Shift: TShiftState);
procedures ItemBoxKeyUp(Senders: TObject; var Key: Word;
Shift: TShiftState);
procedures FormClose(Senders: TObject; var Action: TCloseAction);
procedures N3Click(Senders: TObject);
procedures N4Click(Senders: TObject);
procedures N2Click(Senders: TObject);
procedures btOptionsClick(Senders: TObject);
procedures ReCalcClick(Senders: TObject);
procedures OutButtonClick(Senders: TObject);
procedures ItemBoxDbClick(Senders: TObject);
procedures btSaleClick(Senders: TObject);
procedures N6Click(Senders: TObject);
procedures btUchetClick(Senders: TObject);
procedures PlusButtonClick(Senders: TObject);
procedures ClearButtonClick(Senders: TObject);
procedures MinusButtonClick(Senders: TObject);
procedures FormCreate(Senders: TObject);

```

procedures FormShow(Senders: TObject);
procedures btPriceClick(Senders: TObject);
procedures RoznicaGridDbClick(Senders: TObject);
procedures RoznicaGridKeyPress(Senders: TObject; var Key: Char);
procedures ItemBoxClick(Senders: TObject);
procedures btUpClick(Senders: TObject);
procedures btDownClick(Senders: TObject);
procedures N7Click(Senders: TObject);
procedures CatUpClick(Senders: TObject);
procedures CatDownClick(Senders: TObject);
procedures N777Click(Senders: TObject);
procedures SchetBtnClick(Senders: TObject);
procedures ReservBtnClick(Senders: TObject);
procedures ToResClick(Senders: TObject);

private
  { Private declarations }

public
  { Public declarations }

end;

var
  MainForm: TMainForm;
  Start:boolean = true;
  TmplItemFile:file;
  bbFile:file;
  ItemInRoznicaGrid:word = 0;
  CurKurs:string;
  BaksSum:extended = 0;
  RubSum:extended = 0;
  WhatChosed:word = 0;
  iData:IniType;
  FirstStart:boolean = true;

```

_Client:client;

ChangeItem:ItemType;

function ReturnSelectedItem:string;

function ReturnSelectedCategory:string;

function ReturnCategoryNumber:Integer;

function ReturnSelectedItemNumber:word;

function CategoryNumberByName(name:string):Integer;

procedures DeleteItem(pos:integer);

procedures RefreshSales;

procedures Recalculate;

procedures OpenBase(name:string);

function IsSelectedItem:boolean;

function ReturnSelectedSale:integer;

procedures EnableMainButtons;

procedures DisableMainButtons;

procedures SwapItems(index1,index2:integer);

function GetZiroPos:integer;

procedures ClearZiroRecords;

procedures RefreshCategoryList;

implementation

{SR *.DFM}

{.....}

procedures TMainForm.mbExitClick(Senders: TObject);

begin

```

Close;
end;
{.....}
procedures TMainForm.btExitClick(Senders: TObject);
begin
    Close;
end;
{.....}
procedures TMainForm.btNewClick(Senders: TObject);
var IniData:IniType;
begin
    SaveNewBase.FileName:='untitled1.cbf';
    if SaveNewBase.Execute then
    begin
        If FileExists(SaveNewBase.FileName) then
        begin
            MessageBeep(MB_ICONQUESTION);
            if MessageDlg('File already exists! Owerwrite?',mtConfirmation, [mbYes, mbNo], 0) = mrNo then exit;
        end;
        If SaveNewBase.FileName <> '' then
        CreateBaseFile(SaveNewBase.FileName);
        //....
        IniData:=GetIniData;
        IniData.LastOpenedBase:='';
        IniData.OnStartOpenLastBase:=false;
        UpdateIniData(IniData);

        //....
        CategoryList.Clear;

        btAdd.Enabled:=true;
        btDel.Enabled:=true;

```



```

btIncom.Enabled:=true;
btOptions.Enabled:=true;
btSale.Enabled:=true;

ChangePopup.AutoPopup:=true;
CategoryPopup.AutoPopup:=true;

exit;
end;
MessageDlg(#13+'Ошибка при создании базы!',mtConfirmation, [mbOk], 0);
end;
{.....}
procedures TMainForm.btAddClick(Senders: TObject);
begin
AddCategory;
RefreshCategoryList;
end;
{.....}
procedures TMainForm.FormActivate(Senders: TObject);
begin
if Start then
begin
btAdd.Enabled:=False;
btDel.Enabled:=False;
btIncom.Enabled:=False;
btOptions.Enabled:=False;
btSale.Enabled:=False;
btUchet.Enabled:=False;
btPrice.Enabled:=False;
OpenedFile:="";
ChangePopup.AutoPopup:=false;
CategoryPopup.AutoPopup:=false;

```

```

Start:=false;
AssignFile(TmplItemFile,'TmplItem.lst');
Rewrite(TmplItemFile,1);

end;
end;
{.....}
procedures OpenBase(name:string);
var Kurs:extended;
    c:integer;
    {ConfirmPass:string[16];}
begin

    iData:=GetIniData;
    //...
    if iData.OnStartOpenLastBase = true then
        begin
            iData.LastOpenedBase:=Name;
            UpdateIniData(iData);
        end;

    OpenBaseFile(Name);
    If OpenedFile = '' then exit;
    {ConfirmPass:=ReturnText('Enter password!');
    if mResult = mrCancel then MainForm.Close;
    if ConfirmPass = 'OnlyCreator' then Proverka:=false;
    if ConfirmPass = BaseHeader.Password then Proverka:=false;
    if ConfirmPass <> BaseHeader.Password then Proverka:=true;}

    {Fill list of category}

```

```

MainForm.CategoryList.Clear;
MainForm.ItemBox.Items.Clear;
//.....
if Not TodayIsSet then
begin
  if iData.NoKurs = 0 then
  begin
    Kurs:=GetKurs;
    CurKurs:=FloatToStrF(Kurs,ffNumber,18,2);
  end
  else Kurs:=1;
    AddKursRecord(Kurs);
  end;
MainForm.label5.Caption:=FloatToStrF(CurrentKurs,ffNumber,18,2);
//.....

for c:=1 to TotalCategoryes do
begin
  If BaseHeader.TableCategory[c].Number <> -1 then
  MainForm.CategoryList.Items.Add(BaseHeader.TableCategory[c].Name);
end;
{End fill list of category}

MainForm.btAdd.Enabled:=true;
MainForm.btDel.Enabled:=true;
MainForm.btIncom.Enabled:=true;
MainForm.btOptions.Enabled:=true;
MainForm.btSale.Enabled:=true;
MainForm.btUchet.Enabled:=true;
MainForm.btPrice.Enabled:=true;

```

```

MainForm.ChangePopup.AutoPopup:=true;
MainForm.CategoryPopup.AutoPopup:=true;

MainForm.CategoryList.SetFocus;
if MainForm.CategoryList.Items.Count > 1 then
begin
    MainForm.CategoryList.ItemIndex:=0;
    MainForm.CategoryList.Selected[0];
end;

if MainForm.CategoryList.Items.Count = 0 then
begin
    MainForm.CategoryList.ItemIndex:=-1;
    {MainForm.CategoryList.Selected[0];}
end;

end;
{.....}
procedures TMainForm.btOpenClick(Senders: TObject);
{var c:integer;
    kurs:extended;}
begin
    CreateKursFile;
    If not OpenBaseDialog.Execute then
    begin
        Exit;
    end;
    if Not FileExists(OpenBaseDialog.FileName) then
    begin
        MessageBeep(MB_ICONASTERISK);
        MessageDlg(#13+'Такого файла не существует!',mtError, [mbOk], 0);
    end;
end;

```

```

end;

OpenBase(OpenBaseDialog.FileName);

HideAboutForm;

end;

{.....}

procedures TMainForm.CategoryListClick(Senders: TObject);

var Base:file;

    tHeader:bHeader;

    Counter:integer;

    cNumber:word;

    item:ItemType;

    ItemCount:integer;

    CurPos:integer;

    Total:integer;

begin

Total:=0;

If OpenedFile = '' then exit;

Screen.Cursor:=crHourGlass;

cNumber:=ReturnCategoryNumber;

if cNumber > 50 then cNumber:=1;

Caption:=BaseHeader.TableCategory[cNumber].name+' : Категория №'

                +IntToStr(cNumber);

ItemBox.Clear;

ItemCount:=0;

//.....

AssignFile(TmplItemFile,'TmplItem.lst');

Rewrite(TmplItemFile,1);

//.....

AssignFile(Base,OpenedFile);

Reset(Base,1);

BlockRead(Base,tHeader,BaseHeaderTypeSize);

```

```

Seek(Base,BaseHeaderTypeSize);
for counter:=0 to tHeader.TotalItem-1 do
begin
  CurPos:=FilePos(Base);
  BlockRead(Base,item,ItemTypeSize);
  if item.CategoryNumber = cNumber then
  begin
    ItemBox.Items.Add(item.Name);
    BlockWrite(TmplItemFile,CurPos,4);
    BlockWrite(TmplItemFile,item,ItemTypeSize);
    Inc(ItemCount);
    Total:=Total+item.Count;
  end;
end;
tHeader.TableCategory[cNumber].ItemsCount:=total;
seek(Base,0);
BlockWrite(Base,tHeader,BaseHeaderTypeSize);
CloseFile(Base);
ItemsCount:=ItemCount;
SelectedItem:=ReturnSelectedItemNumber;

If ItemBox.Items.Count <> 0 then btDel.Enabled:=false else btDel.Enabled:=true;

CategoryList.SetFocus;
Screen.Cursor:=crDefault;
end;
{.....}
procedures TMainForm.btDelClick(Senders: TObject);
var c:word;
    Name:string;
    nn:word;
begin

```

```

If CategoryList.Items.Count = 0 then exit;
for c:=0 to CategoryList.Items.Count-1 do
begin
  if CategoryList.Selected[c] then
    begin
      Name:=CategoryList.Items.Strings[c];
      nn:=GetCategory(Name);
      OpenBaseFile(OpenedFile);
      if BaseHeader.TableCategory[nn].ItemsCount <> 0 then
        begin
          MessageBeep(MB_iconasterisk);
          MessageDlg(#13+'Вы не можете удалить эту категорию,'+#13
            +'т.к. на неё в базе имеются ссылки!', mtInformation, [mbOk], 0);
          exit;
        end;
          MessageBeep(MB_ICONQUESTION);
          If MessageDlg(#13+'Вы действительно хотите удалить'
            +' категорию: '+Name+'?', mtConfirmation, [mbOk,mbCancel], 0) = mrOk then
            DeleteCategory(Name) else exit;
          break;
        end;
      end;
    end;

  _RefreshCategories;
  CategoryList.Clear;
  for c:=1 to TotalCategories do
    begin
      if BaseHeader.TableCategory[c].number <> -1 then
        CategoryList.Items.Add(BaseHeader.TableCategory[c].Name);
      end;
    end;
  end;
end;

```

```

{.....}
procedures TMainForm.btIncomClick(Senders: TObject);
begin
  OpenBaseFile(OpenedFile);
  If BaseHeader.CategoryCount <=0 then
  begin
    MessageBeep(MB_ICONHAND);
    MessageDlg(#13'Не создано ни одной категории!', mtWarning, [mbOk], 0);
    exit;
  end;
  ShowNaklad;
  //Update Category and Item boxes
  CategoryListClick(Senders);
end;
//.....
procedures TMainForm.N1Click(Senders: TObject);
var c:integer;
    rName:string;
    NewCategory:string;
    OldNumber:word;
    chItem:ItemType;
    chPos:integer;
    fFile:file;
    NewNumber:word;
    OldCategoryNumber:word;
    NewCat:string;
begin
  rName:='';
  OpenBaseFile(OpenedFile);
  rName:=ReturnSelectedItem;
  if rName = '' then exit;
  //.....

```



```

//Change request
Screen.Cursor:=crHourGlass;
Form2.ComboBox1.Items.Clear;
{Fill list of category}
for c:=1 to TotalCategories do
begin
NewCat:=BaseHeader.TableCategory[c].Name;
if NewCat = '<Category empty>' then continue;
Form2.ComboBox1.Items.Add(NewCat);
end;
{End fill list of category}
Screen.Cursor:=crDefault;
Form2.ComboBox1.Style:=csOwnerDrawVariable;
NewCategory:=ReturnText('В какую категорию перенести: '+rName);
Form2.ComboBox1.Style:=csDropDown;
If mResult = mrCancel then exit;

//.....
OldNumber:=ReturnSelectedItemNumber;
chItem:=ReturnItem(OldNumber,chPos);
OldCategoryNumber:=chItem.CategoryNumber;
NewNumber:=CategoryNumberByName(NewCategory);
chItem.CategoryNumber:=NewNumber;
//.....
Dec(BaseHeader.TableCategory[OldCategoryNumber].ItemsCount,chItem.Count);
Inc(BaseHeader.TableCategory[NewNumber].ItemsCount,chItem.Count);
AssignFile(bbFile,OpenedFile);
Reset(bbFile,1);
BlockWrite(bbFile,BaseHeader,BaseHeaderTypeSize);
CloseFile(bbFile);
//.....
AssignFile(fFile,OpenedFile);

```

```

Reset(fFile,1);
//.....
Seek(fFile,chPos);
BlockWrite(fFile,chItem,ItemTypeSize);
CloseFile(fFile);
//.....
ItemBox.Clear;
CategoryList.SetFocus;
CategoryListClick(Senders);
end;
//.....
//This function returns name of selected line in the ItemBox
function ReturnSelectedItem:string;
var c:integer;
begin
Result:="";
if MainForm.ItemBox.Items.Count = 0 then exit;
for c:=0 to MainForm.ItemBox.Items.Count-1 do
begin
if MainForm.ItemBox.Selected[c] then
begin
Result:=MainForm.ItemBox.Items.Strings[c];
exit;
end;
end;
end;
//.....
//.....
//This function returns number of selected line in the ItemBox
function ReturnSelectedItemNumber:word;
var c:integer;
begin

```

```

Result:=0;
if MainForm.ItemBox.Items.Count = 0 then exit;
for c:=0 to MainForm.ItemBox.Items.Count-1 do
begin
  if MainForm.ItemBox.Selected[c] then
    begin
      Result:=c;
      exit;
    end;
end;
end;
//.....

//This function returns name of selected line in the CategoryList
function ReturnSelectedCategory:string;
var c:integer;
begin
  Result:="";
  if MainForm.CategoryList.Items.Count = 0 then exit;
  for c:=0 to MainForm.CategoryList.Items.Count-1 do
  begin
    if MainForm.CategoryList.Selected[c] then
      begin
        Result:=MainForm.CategoryList.Items.Strings[c];
        exit;
      end;
    end;
  end;
end;
//.....

//.....
procedures TMainForm.ItemBoxKeyPress(Senders: TObject; var Key: Char);

```

```

begin
  if Key = #13 then ItemBoxDbClick(Senders);
end;
//.....
function ReturnCategoryNumber:Integer;
var c,d:word;
begin
  ReturnCategoryNumber:=1;
  If OpenedFile = '' then Exit;
  OpenBaseFile(OpenedFile);
  for c:=1 to BaseHeader.CategoryCount do
  begin
    if MainForm.CategoryList.Selected[c-1] then
      begin
        for d:=1 to TotalCategories do
          begin
            if MainForm.CategoryList.Items.Strings[c-1] = BaseHeader.TableCategory[d].name then
              begin
                CurrentCategoryNumber:=BaseHeader.TableCategory[d].number;
                Result:=BaseHeader.TableCategory[d].number;
                exit;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;
//.....
//.....
function CategoryNumberByName(name:string):Integer;
var d:word;
begin
  CategoryNumberByName:=-1;

```

```

If OpenedFile = '' then Exit;
OpenBaseFile(OpenedFile);
  for d:=1 to TotalCategories do
  begin
    if name = BaseHeader.TableCategory[d].name then
      begin
        Result:=BaseHeader.TableCategory[d].number;
        exit;
      end;
    end;
  end;
end;
//.....
//.....
procedures TMainForm.CategoryListKeyUp(Senders: TObject; var Key: Word;
Shift: TShiftState);
begin
Case Key of
38:CategoryListClick(Senders);
40:CategoryListClick(Senders);
45:btIncomClick(Senders);
{39:begin
  ItemBox.SetFocus;
  ItemBox.ItemIndex:=0;
  end;}
end;
{Caption:=IntToStr(Key);}
end;
//.....
procedures TMainForm.ItemBoxKeyUp(Senders: TObject; var Key: Word;
Shift: TShiftState);
begin
Case Key of

```

```

45:btIncomClick(Senders);
end;
{if Key = 9 then CategoryList.SetFocus;}
{Caption:=IntToStr(Key);}
end;
//.....
procedures TMainForm.FormClose(Senders: TObject; var Action: TCloseAction);
begin
  {if FileExists('Tmpltem.lst') then CloseFile(TmpltemFile);}
  //Erase(TmpltemFile);
end;
//.....
procedures TMainForm.N3Click(Senders: TObject);
var NewName:string[200];
    dNumber:integer;
    hItem:ItemType;
    NeedPos:integer;
    fFile:file;
begin
  if ItemBox.Items.Count<=0 then
    begin
      MessageBeep(MB_ICONHAND);
      MessageDlg(#13+'В этой категории нечего переименовывать!', mtInformation, [mbOk], 0);
      exit;
    end;
  dNumber:=ReturnSelectedItemNumber;
  hItem:=ReturnItem(dNumber,NeedPos);

  Form2.ComboBox1.Items.Clear;
  NewName:=RenameText('Новое название для: '+hItem.Name,hItem.Name);
  if mResult = mrCancel then exit;

```

```

AssignFile(fFile,OpenedFile);
Reset(fFile,1);
//check item
Seek(fFile,NeedPos);
BlockRead(fFile,hItem,ItemTypeInfo);
//.....
hItem.Name:=NewName;
Seek(fFile,NeedPos);
BlockWrite(fFile,hItem,ItemTypeInfo);
CloseFile(fFile);
//.....
ItemBox.Clear;
CategoryList.SetFocus;
CategoryListClick(Senders);
end;
//.....
procedures TMainForm.N4Click(Senders: TObject);
var nomer,ItemPos:integer;
    Reltem:ItemTypeInfo;
    ReFile:file;
    Count:integer;
begin
OpenBaseFile(OpenedFile);
If ItemBox.Items.Count <=0 then
begin
    MessageBeep(MB_ICONHAND);
    MessageDlg(#13+'В этой категории изменять нечего!', mtWarning, [mbOk], 0);
    exit;
end;
Nomer:=ReturnSelectedItemNumber;
Reltem:=ReturnItem(Nomer,ItemPos);
Count:=Reltem.Count;

```

```

ReDataShow(Reltem,Reltem.Name);
if ReDataResult = mrCancel then exit;
if Count <> Reltem.Count then
  begin
    if Count < Reltem.Count then Count:=Reltem.Count-Count;
    if Count > Reltem.Count then Count:=-Count-Reltem.Count;
  end;
//Update item in the base
AssignFile(ReFile,OpenedFile);
Reset(ReFile,1);
BlockRead(ReFile,BaseHeader,BaseHeaderTypeSize);
BaseHeader.TableCategory[Reltem.CategoryNumber].ItemsCount:=
BaseHeader.TableCategory[Reltem.CategoryNumber].ItemsCount+Count;
Seek(ReFile,0);
BlockWrite(ReFile,BaseHeader,BaseHeaderTypeSize);
//check item
//Seek(ReFile,ItemPos);
//BlockRead(ReFile,Reltem,ItemTypeSize);
//.....
Seek(ReFile,ItemPos);
BlockWrite(ReFile,Reltem,ItemTypeSize);
CloseFile(ReFile);

//Update Category and Item boxes
CategoryListClick(Senders);
if NewData.Visible then Show_ShowForm(Reltem);
end;
//.....
procedures TMainForm.N2Click(Senders: TObject);
var dNumber:word;
    dPos:integer;
    dltem:ItemType;

```



```

begin
  If ItemBox.Items.Count = 0 then exit;
  dNumber:=ReturnSelectedItemNumber;
  dltem:=ReturnItem(dNumber,dPos);
  if dltem.Count >= 0 then
    begin
      MessageBeep(MB_ICONQUESTION);
      if MessageDlg('Вы действительно хотите удалить '+
        dltem.Name+'?', mtConfirmation, [mbOk,mbCancel], 0)=
        mrCancel then exit else Deleteltem(dPos);
    end;
  //Update Category and Item boxes
  CategoryListClick(Senders);
end;
//.....
procedures Deleteltem(pos:integer);
var ddFile:file;
  NewFile:file;
  tltem:ItemType;
  PosOfEnd:integer;
  Saveltem:ItemType;
  Counter:integer;
  CurPos:integer;
  bbHeader:bHeader;
begin
  //.....
  AssignFile(ddFile,OpenedFile);
  Reset(ddFile,1);
  //...
  AssignFile(NewFile,'newname.tmp');
  Rewrite(NewFile,1);
  //...

```

```

PosOfEnd:=FileSize(ddFile)-ItemTypeSize;
//...
Seek(ddFile,Pos);
BlockRead(ddFile,tItem,ItemTypeSize);

//Rewrite header
Seek(ddFile,0);
BlockRead(ddFile,bbHeader,BaseHeaderTypeSize);
Dec(bbHeader.TableCategory[tItem.CategoryNumber].ItemsCount,tItem.Count);
//...
Seek(NewFile,0);
BlockWrite(NewFile,bbHeader,BaseHeaderTypeSize);
//End of rewrite header
if Pos = PosOfEnd then
begin
Dec(bbHeader.TotalItem);
Seek(ddFile,0);
BlockWrite(ddFile,bbHeader,BaseHeaderTypeSize);
Seek(ddFile,Pos);
Truncate(ddFile);
CloseFile(ddFile);
CloseFile(NewFile);
Erase(NewFile);
exit;
end;
//...
Seek(ddFile,BaseHeaderTypeSize);
Seek(NewFile,BaseHeaderTypeSize);
//...
for Counter:=0 to bbHeader.TotalItem-1 do
begin
CurPos:=FilePos(ddFile);

```

```

BlockRead(ddFile,Saveltem,ItemTypeSize);
if CurPos <> Pos then BlockWrite(NewFile,Saveltem,ItemTypeSize);
end;
//...
Dec(bbHeader.TotalItem);
Rewrite(ddFile,1);
Seek(ddFile,0);
BlockWrite(ddFile,bbHeader,BaseHeaderTypeSize);
//...
Seek(ddFile,BaseHeaderTypeSize);
Seek(NewFile,BaseHeaderTypeSize);
for Counter:=0 to bbHeader.TotalItem-1 do
begin
BlockRead(NewFile,Saveltem,ItemTypeSize);
BlockWrite(ddFile,Saveltem,ItemTypeSize);
end;
CloseFile(ddFile);
CloseFile(NewFile);
Erase(NewFile);
OpenBaseFile(OpenedFile);
end;
//.....

procedures TMainForm.btOptionsClick(Senders: TObject);
begin
If CategoryList.Items.Count = 0 then
begin
MessageBeep(MB_ICONASTERISK);
MessageDlg('Не создано ни одной категории!', mtInformation, [mbOk], 0);
exit;
end;
ShowOptions;

```

```

label5.Caption:=FloatToStrF(CurrentKurs,ffNumber,18,2);
end;
//.....
procedures TMainForm.ReCalcClick(Senders: TObject);
Var Num:word;
    f:file;
    c:word;
    ItemsCount:integer;
    Counter:integer;
    It:ItemType;
begin
Screen.Cursor:=crHourGlass;
if CategoryList.Items.Count = 0 then exit;
ItemsCount:=0;
Num:=ReturnCategoryNumber;
AssignFile(f,OpenedFile);
Reset(f,1);
Blockread(f,BaseHeader,BaseHeaderTypeSize);
//...
for Counter:=0 to BaseHeader.TotalItem-1 do
begin
    BlockRead(f,it,ItemTypeSize);
    if It.CategoryNumber = Num then
        Inc(ItemsCount,it.Count);
end;
BaseHeader.TableCategory[num].ItemsCount:=ItemsCount;
Seek(f,0);
BlockWrite(f,BaseHeader,BaseHeaderTypeSize);
CloseFile(f);
_RefreshCategoryes;
CategoryList.Clear;
for c:=1 to TotalCategoryes do

```

```
begin
  if BaseHeader.TableCategory[c].number <> -1 then
    CategoryList.Items.Add(BaseHeader.TableCategory[c].Name);
  end;
  Screen.Cursor:=crDefault;
end;
//.....
```