**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ 8

1. Исследование основ разработки web-сайта 10
   1. Понятие web-сайта 10
   2. Обоснование потребности в web-сайте 11
   3. Определение целей создания web-сайта 13
2. Разработка технического задания 15
   1. Анализ предметной области и требований к системе 15
      1. Позиционирование 15
      2. Описание пользователей 16
      3. Краткий обзор изделия 18
      4. Возможности продукта 19
   2. Построение диаграммы вариантов использования 20
      1. Выявление акторов 20
      2. Выявление вариантов использования 21
      3. Разработка диаграмм вариантов использования 22
      4. Структуризация вариантов использования 23
      5. Реестрвариантовиспользования 24
      6. Конкретизация вариантов использования 26
      7. Поиск ключевых вариантов использования 28
   3. Разработка технического задания 30
      1. Общее описание системы 30
      2. Список действующих лиц 31
      3. Глоссарий 32
      4. Структура проектируемой системы 34
      5. Требования для развертывания системы 35
      6. Прочие требования 36
   4. Построение дополнительных диаграмм 39
      1. Диаграмма последовательности 39
      2. Диаграмма состояний 41
   5. Построение диаграммы классов 42
   6. Проектирование базы данных 43
   7. Планирование развертывания системы 45
3. Выбор средств разработки для разрабатываемой системы 47
   1. Выбор средств и инструментов для разработки базы данных 47
   2. Выбор средств и инструментов для разработки программы 49
      1. Язык разметки гипертекстовых страниц HTML 49
      2. Таблицы стилей CSS 51
      3. Язык написания сценариев PHP 52
      4. Сценарный язык JavaScript 53
4. Реализация разрабатываемой программы 56
5. Интерфейсы информационной системы 56

ЗАКЛЮЧЕНИЕ 64

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 65

Приложение А (обязательное) Листинг

Приложение Б (обязательное) Графический материал

**ВВЕДЕНИЕ**

Современный период развития цивилизованного общества характеризует процесс информатизации.

Информатизация общества — это глобальный социальный процесс, особенность которого состоит в том, что доминирующим видом деятельности в сфере общественного производства является сбор, накопление, продуцирование, обработка, хранение, передача и использование информации, осуществляемые на основе современных средств микропроцессорной и вычислительной техники, а также на базе разнообразных средств информационного обмена. Информатизация общества обеспечивает:

* активное использование интеллектуального потенциала общества, сконцентрированного в печатном фонде, и научной, производственной и других видах деятельности его членов;
* интеграцию информационных технологий в научные и производственные виды деятельности, инициирующую развитие всех сфер общественного производства, интеллектуализацию трудовой деятельности;
* высокий уровень информационного обслуживания, доступность любого члена общества к источникам информации, визуализацию представляемой информации, существенность используемых данных.

Применение открытых информационных систем, рассчитанных на использование всего массива информации, доступной в данный момент обществу в определенной его сфере, позволяет усовершенствовать механизмы управления общественным устройством, способствует гуманизации и демократизации общества, повышает уровень благосостояния его членов. Процессы, происходящие в связи с информатизацией общества, способствуют не только ускорению научно-технического прогресса, интеллектуализации всех видов человеческой деятельности, но и созданию качественно новой информационной среды социума, обеспечивающей развитие творческого потенциала индивида.

В данной работе основной задачей является разработка информационной системы для автоматизации деятельности оператора Центра обработки вызовов службы исходящего телемаркетинга.

Автоматизированная система обработки информации должна:

* обеспечить получение общих или детализированных данных;
* позволять легко определять тенденции изменения важнейших показателей;
* обеспечивать получение информации, критичной во времени, без существенной задержки;
* выполнять точный и полный анализ данных;
* группировать информацию, как по логическим разделам, так и по степени важности.
* гарантировать сохранность данных.
* гарантировать защищенность данных от постороннего вмешетельства.

Разработка подобного приложения позволит намного сократить время на сбор и опубликование информации, как для сомой организации так и для целевой аудитории.

1. **Исследование основ разработки web-сайта**
   1. **Понятие web-сайта**

Информация, доступная пользователям Internet, располагается на компьютерах (Web-серверах), на которых установлено специальное программное обеспечение. Значительная часть этой информации организована в виде Web-сайтов.

Web-сайт – это информация, представленная в определенном виде, которая располагается на Web-сервере и имеет свое имя (адрес). Для просмотра сайтов на компьютере пользователя используются специальные программы, которые называются браузерами. В зависимости от того, какое имя сайта задается в адресной строке, браузер будет загружать соответствующую информацию.

Web-сайт состоит из связанных между собой страниц. Web-страница представляет собой текстовый файл, который содержит текстовую информацию и специальные команды – HTML-коды, определяющие в каком виде эта информация будет отображаться в окне браузера. Вся графическая, аудио- и видео-информация непосредственно в Web-страницу не входит и представляет собой отдельные файлы. В HTML-коде страницы содержатся только указания на такие файлы.

Каждая страница сайта также имеет свой адрес, который состоит из адреса сайта и имени файла, соответствующего данной странице. Таким образом, Web-сайт – это информационный ресурс, состоящий из связанных между собой гипертекстовых документов, размещенный на сервере и имеющий индивидуальный адрес. Посмотреть сайт может любой человек, имеющий компьютер, подключенный к Internet.

В настоящее время во всемирной паутине размещено несколько миллионов Web-сайтов и их число постоянно растет. Это личные страницы, содержащие информацию об авторе, его интересах. Широкую группу образуют информационные сайты. К ним относятся сайты учебных заведений, сообществ по интересам, фирм и др. Все большее развивается электронная коммерция. В сети встречаются виртуальные магазины, которые позволяют делать покупки, сидя за мониторами компьютеров. Сейчас в сети размещено также большое количество информационно-развлекательных сайтов.

* 1. **Обоснование потребности в web-сайте**

На сегодняшний день практически каждая организация имеет собственный web-сайт. В условиях использования современных информационных технологий – это необходимый фактор существования, позволяющий расширить поле рекламной деятельности и привлечь тем самым дополнительных клиентов.

Создание и разработка сайтов включает:

- утверждение первоначального технического задания на разработку сайта;

- определение структурной схемы сайта – расположение разделов, контента и навигации;

- web-дизайн – создание графических элементов макета сайта, стилей и элементов навигации;

- разработка программного кода, модулей, базы данных и других элементов сайта необходимых в проекте;

- тестирование и размещение сайта в сети.

Интернет предоставляет пользователям широкий набор инструментов коммуникации. Любой пользователь, подключившись к сети, сразу же приобретает возможность навигации по сайтам, возможность отправлять и получать электронные письма, участвовать в конференциях и чатах и многое, многое другое. Другими словами пользователь получает возможность интерактивного взаимодействия со средой Интернета.

Пользователем Интернета может стать и любая компания. Сразу же после подключения к Сети она может начать применять новые методы в ведении своей коммерческой деятельности.

Освоившись с базовыми службами Интернета, любой пользователь может пойти дальше и принять самое активное участие в реализации среды Интернета, создав собственный web-сайт, интерактивный магазин или торговую систему.

Web-сайт предоставляет компании широкий ряд дополнительных возможностей, в добавление к доступным ранее коммуникационным службам. Главная их особенность состоит в том, что теперь компания может предоставлять пользователям дополнительный сервис: давать информацию, оказывать пред- и послепродажное обслуживание, продавать товары и услуги. При наличии собственного сайта возрастает эффективность проводимых в Сети рекламных мероприятий, поскольку любая реклама может содержать ссылку на сайт компании, где пользователю станет доступно значительно большее количество информации, и он составит более полное мнение о фирме и ее предложении. Кроме того, информация о поведении посетителей на сайте может сразу же попадать в информационную систему компании и служить для более качественного их обслуживания.

Таким образом, web-сайт является одним из главных элементов системы маркетинга в Интернете и требует наибольшего внимания со стороны маркетинговой службы предприятия.

* 1. **Определение целей создания web-сайта**

Создание Web-сайтов многими компаниями воспринимается лишь как модная тенденция. Создать свой сайт стремятся практически все отечественные компании, но далеко не все понимают, насколько важна разработка сайтов для развития бизнеса.

Первым шагом на пути к созданию сайта является определение целей. Это необходимо для лучшего понимания задач, которые должны быть решены с помощью сайта. Правильно и четко поставленная цель позволяет превратить сайт в эффективный инструмент, направленный на достижение целей его создателей.

У всех компаний независимо от их особенностей и характеристик имеется основная общая цель. Этой целью является получение прибыли. Исходя из этого сайт компании обычно рассматривается, как инструмент повышения эффективности компании именно с этой точки зрения. Но это не означает, что сайт должен приносить прибыль непосредственно за счет прямых продаж. Получение прибыли от созданного сайта может происходить и опосредованно, за счет достижения целей второго уровня, к которым можно отнести такие цели, как:

- улучшение имиджа компании;

- информирование потенциальных потребителей о достоинствах и преимуществах предлагаемых компанией услуг или товаров;

- обеспечение связи потребителей с компанией;

- привлечение внимания потребителей к новинкам и наиболее интересным предложениям компании;

- облегчение оформления заказов;

- удержание и обеспечение лояльности клиентов компании;

- информационная и техническая поддержка постоянных клиентов;

- организация прямых продаж на сайте;

- формирование и продвижение партнерской сети.

После того как цели будут сформулированы нужно определить необходимые ресурсы и сроки достижения целей. Под ресурсами нужно понимать определенные умения и знания по созданию сайта, а также программные коды, базы данных, серверные технологии и многие другие вещи, без которых можно и обойтись, но которые необходимы для воплощения оригинальных идей. Всему этому можно научиться самостоятельно, затратив при этом собственное время, или обратиться в компанию, занимающуюся разработкой и продвижением сайтов.

Конкретно определенная цель предполагает и сроки ее достижения. Обычно процесс создания сайта разбивается на определенные этапы, для каждого этапа устанавливается срок его выполнения, ответственный и форма отчетности.

Также необходимо правильно оценить и обосновать цель. Необоснованные цели отпадут сами собой. Постановка целей – это не просто полезное занятие, это совершенно необходимый элемент любой деятельности.

1. **Разработка технического задания**
   1. **Анализ предметной области и требований к системе**
      1. **Позиционирование**

В настоящее время работа оператора центра обработки вызовов службы исходящего телемаркетинга осуществляется на основе использования электронных таблиц. В сравнении с имеющимся новое решение позволит обеспечить более удобный режим доступа заинтересованных лиц к информа-ции, повысить быстродействие, обеспечить надёжное хранение данных и более полный охват функций, подлежащих автоматизации.

В ходе анализа предметной области был определен ряд проблем (таблица 1).

Таблица 1 – Определение проблем

|  |  |  |  |
| --- | --- | --- | --- |
| **Проблема** | **Затрагивает** | **Следствием является** | **Успешное решение** |
| Низкая скорость обмена информацией между оператором и опрашиваемым абонентом. | Оператора, абонента. | Недостаточно оперативный опрос абонентов. | Оптимальная организация работы службы, экономия времени участников и затрат на обеспечение телефонной связи. |
| Высокая сложность процесса обработки полученных данных. | Оператора, менеджера. | Длительная обработка оперативных данных. | Автоматизированная выгрузка данных по опросам абонентов. |

Основные сведение о позициях изделия определены в таблице 2.

Таблица 2 – Определение позиций изделия

|  |  |
| --- | --- |
| **Для** | Провайдер «ЛинкЛайф» |
| **которой** | Требуется оптимизировать процесс работы оператора центра обработки вызовов службы исходящего телемаркетинга |
| **(Название продукта)** | АИС «Телемаркетинг» |
| **который** | Основан на промышленной СУБД и высоконадёжен |
| **В отличие от** | Существующего механизма на основе электронных таблиц |
| **наш продукт** | Исключает ошибки планирования |

* + 1. **Описание пользователей**

У системы существуют три основных пользователя: оператор, менеджер, контент-менеджер.

Оператор – проводит опрос абонентов, информирует по имеющимся в АИС данным, вводит данные о поступивших ответах абонентов.

Менеджер – имеет доступ к оперативным данным по опрошенным абонентам, производит необходимые расчеты по полученным статистическим данным.

Контент-менеджер – осуществляет добавление новых сведений в базу данных и редактирование старых: добавляет новые акции и предложения, обновляет список абонентов.

В настоящее время в службе телемаркетинга имеется 3 менеджера, 6 операторов, 1 контент-менеджер.

Разумное время планирования одного опроса – 5 минут, для длительных опросов – не более 10 минут.

Доступ к данным на сервере будет осуществляться через локальную сеть компании, либо через Интернет. Система будет работать на платформе IBM PC. Операционная система для серверной части – Ubuntu Linux, для терминала – Microsoft Windows Server 2008, для клиентской – Microsoft Windows 7.

Профили пользователей представлены в таблице 3.

Таблица 3 – Определение позиций изделия

|  |  |
| --- | --- |
| **Типичный представитель** | Менеджер |
| **Описание** | Пользователь, наделенный правами на чтение информации и выведение статистических данных |
| **Тип** | Пользователь |
| **Ответственности** | Выводит статистические данные |
| **Критерий успеха** | Возможность определения отношения поддержания клиентской базы |

|  |  |
| --- | --- |
| **Типичный представитель** | Оператор |
| **Описание** | Пользователь, наделенный правами на чтение информации и внесение данных по опросам абонентов |
| **Тип** | Пользователь |
| **Ответственности** | Ввод информации по опросам абонентов, информирование абонентов по имеющимся тарифам и предложениям |
| **Критерий успеха** | Быстрое и качественное проведение опросов абонентов |

|  |  |
| --- | --- |
| **Типичный представитель** | Контент-менеджер |
| **Описание** | Пользователь, наделенный правами на чтение информации, добавление и удаление данных |
| **Тип** | Пользователь |
| **Ответственности** | Ввод информации по акциям и тарифным планам, спискам операторов |
| **Критерий успеха** | Быстрое и качественное наполнение базы данных актуальными сведениями |

Оператор затрачивает большое количество времени на получение информации и внесение необходимых дополнений. Менеджеры затрачивают значительное количество времени на получение статистической информации от операторов. Контент-менеджеры тратят колоссальное количество времени на наполнение базы данных актуальными сведениями. Предприятие нуждается в системе, которая бы ускорила и оптимизировала вышеуказанные процессы.

# Краткий обзор изделия

Система является законченной независимой разработкой. В перспективе возможно использование системы в комплексе с системами автоматизации других отделов. Коммуникации – на уровне доступа к общей базе данных.

Система будет использоваться на территориально сосредоточенном (без внешних филиалов) предприятии. В случае изменений в формах документов АИС должна претерпеть малосущественные изменения (нужно будет модифицировать отчетные формы). В случае приобретения или разработки информационных систем, автоматизирующих смежные участки (маркетинг, обработка полученной информации другими подразделениями), будет необходимо разработать соответствующие средства импорта-экспорта информации.

* + 1. **Возможности продукта**
* Упрощение работы оператора – ускоренная обработка информации, необходимой для опроса абонентов и корректной работы с абонентами провайдера.
* Ускорение обращения информации – система позволит ускорить процесс получения необходимой информации о проведенных опросах абонентов.
* Формирование единой базы для планирования и последующего анализа – все заинтересованные пользователи со своих рабочих мест имеют доступ к оперативной информации о проведенных опросах абонентов; накопленные в базе данные позволят осуществить анализ статистики.
* Возможность индивидуального подхода к каждому заказу – Система позволяет классифицировать тарифы и предложения, осуществлять подбор корректных форм вопросов на определенный ответ абонента.
* Отказ от излишних коммуникаций – система позволяет пользователям получать нужную им информацию самостоятельно, не отвлекая от работы других участников процесса.

# Построение диаграммы вариантов использования

**2.2.1 Выявление акторов**

Проведем анализ акторов, работающих с системой (рисунок 1).

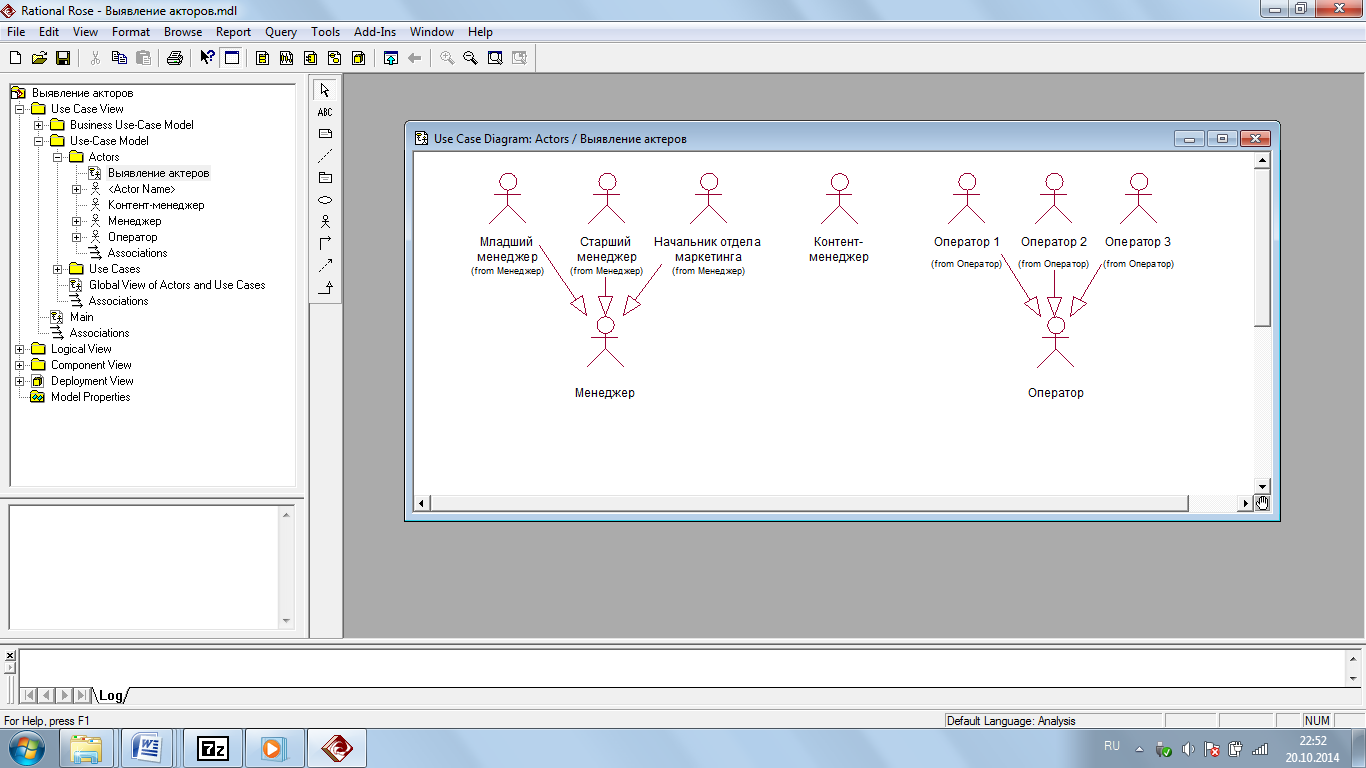


Рисунок 1 – Анализ акторов, работающих с системой

Интервью, проведённое с указанными выше кандидатами показало, что младший менеджер, старший менеджер и начальник отдела телемаркетинга предполагают использовать разрабатываемую АИС однотипно. Это позволило обобщить эти 3 роли в одну. Аналогичная ситуация со всеми операторами.

Краткое описание акторов представлено в таблице 4.

Таблица 4 – Выявление акторов

|  |  |
| --- | --- |
| **Актор** | **Краткое описание** |
| Контент-менеджер | Осуществляет добавление новых сведений в базу данных и редактирование старых: добавляет новые акции и предложения, обновляет список операторов. |
| Оператор | Указывает параметры для проведения опроса (выбирает акцию, список абонентов и конкретного абонента для опроса). После установления телефонного соединения с абонентом проводит опрос по выбранной ранее акции и фиксирует полученные результаты. |
| Менеджер | Просматривает и обрабатывает статистические результаты, полученные из ответов абонентов. На их основе составляют отчеты и принимают решении о разработке новых акций и тарифов. |

* + 1. **Выявление вариантов использования**

Выявленные варианты использования сведены в таблицу 5.

Таблица 5 – Выявление вариантов использования

|  |  |  |
| --- | --- | --- |
| **Основной актор** | **Наименование** | **Формулировка** |
| Контент-менеджер | Добавление акций и тарифных планов | По требованию менеджера осуществляет добавление акций и тарифных планов. |
| Контент-менеджер | Актуализация списка операторов | Осуществляет добавление в базу данных новых операторов и удаление отстраненных сотрудников. |
| Оператор | Указание параметров опроса | Указывает параметры для проведения опроса, а именно выбирает акцию, список абонентов и конкретного абонента для опроса. |
| Оператор | Формирование результатов опроса | Записывает в базу данных полученные от абонента ответы. |
| Менеджер | Просмотр результатов опроса | Просматривает и обрабатывает статистические данные, полученные на основе ответов абонентов. |
| Менеджер | Разработка и передача готовых акций и тарифных планов | Разработка условий новых акций и тарифных планов, а также перечня вопросов и ответов для проведения опроса. |

* + 1. **Разработка диаграмм вариантов использования**

Рассмотрим диаграмму прецедентов системы (рисунок 2).

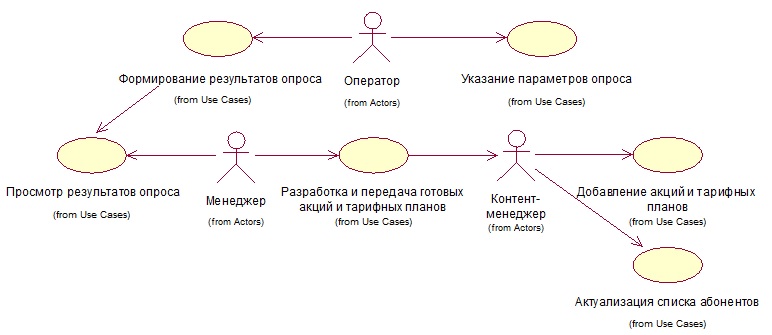


Рисунок 2 – Диаграмма прецедентов системы

* + 1. **Структуризация вариантов использования**

Анализ вариантов использования выявил следующие взаимосвязи:

1. Вариант использования «Указание параметров опроса» является обобщающим, поэтому было принято решение ввести новые варианты использования «Выбор акции», «Выбор списка абонентов» и «Указание неопрошенного абонента», оставить прецедент «Указание параметров опроса», как основной для вновь введённых прецедентов (рисунок 3).

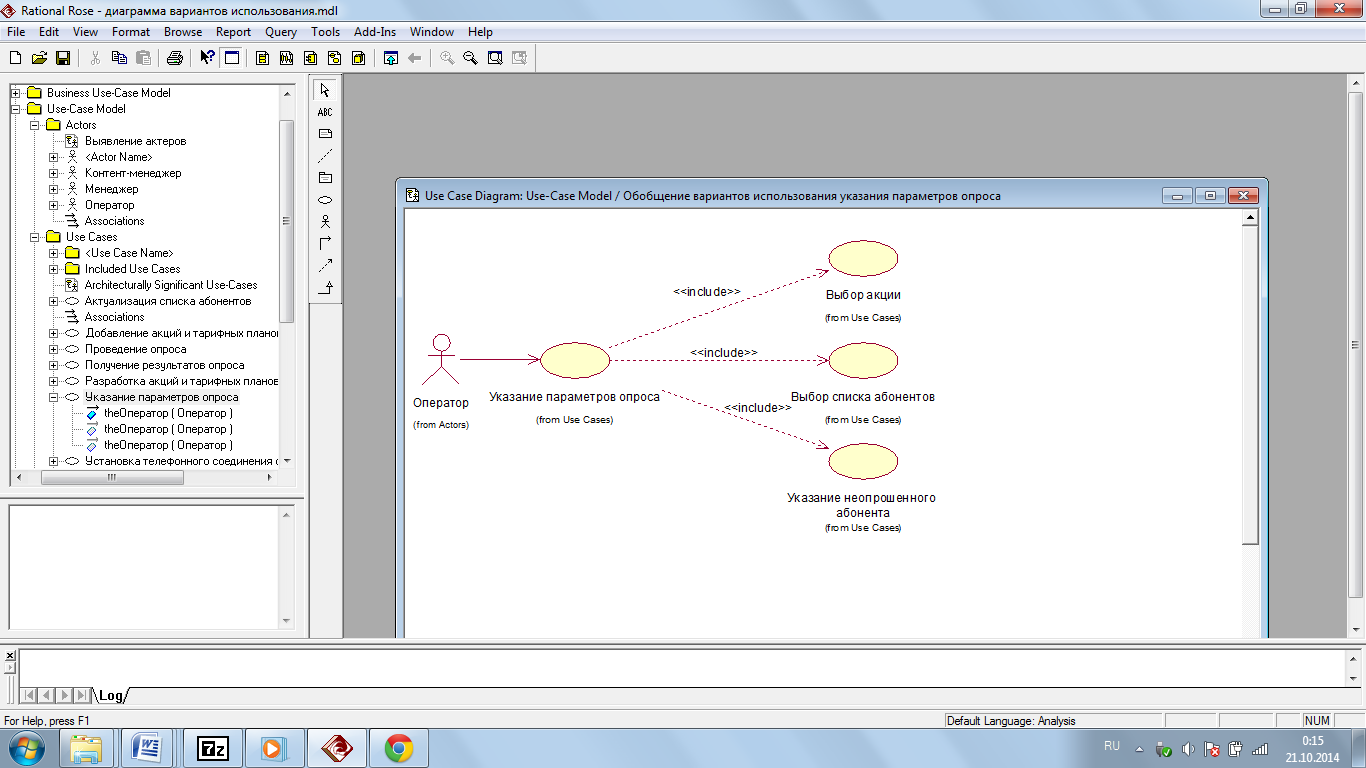


Рисунок 3 — Анализ связей расширения и включения для вариантов использования указания параметров опроса

1. Контент-менеджер получает доступ к базе данных при условии успешной проверки прав, после чего ему предоставляется возможность просматривать, редактировать, создавать и удалять записи в БД.
2. Оператор получает доступ к базе данных при условии успешной проверки прав, после чего ему предоставляется возможность просматривать, редактировать и создавать записи в БД.
3. Менеджер получает доступ к базе данных при условии успешной проверки прав, после чего ему предоставляется возможность только просматривать записи в БД.

Результирующая диаграмма вариантов использования показана на рисунке 4.

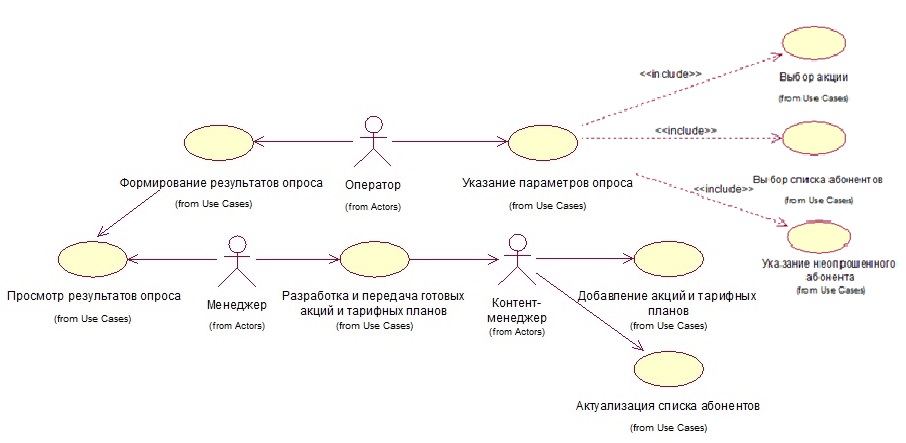


Рисунок 4 – Модифицированная диаграмма прецедентов системы

* + 1. **Реестр вариантов использования**

По результатам анализа, проведенного в параграфе «2.2.4 Структуризация вариантов использования», не выявлено исключенных вариантов использования, были выявлены некоторые дополнительные прецеденты и взаимосвязи между прецедентами. Результирующий список вариантов использования показан в таблице 6.

Таблица 6 – Реестр вариантов использования

|  |  |  |  |
| --- | --- | --- | --- |
| **Код** | **Основной актор** | **Наименование** | **Формулировка** |
| K1 | Контент-менеджер | Добавление акций и тарифных планов | По требованию менеджера осуществляет добавление акций и тарифных планов. |
| K2 | Контент-менеджер | Актуализация списка операторов | Осуществляет добавление в базу данных новых операторов и удаление отстраненных сотрудников. |
| O1 | Оператор | Указание параметров опроса | Указывает параметры для проведения опроса, а именно выбирает акцию, список абонентов и конкретного абонента для опроса. |
| O2 | Оператор | Формирование результатов опроса | Записывает в базу данных полученные от абонента ответы. |
| M1 | Менеджер | Просмотр результатов опроса | Просматривает и обрабатывает статистические данные, полученные на основе ответов абонентов. |
| M2 | Менеджер | Разработка и передача готовых акций и тарифных планов | Разработка условий новых акций и тарифных планов, а также перечная вопросов и ответов для проведения опроса. |

* + 1. **Конкретизация вариантов использования**

Рассмотрим подробнее варианты использования, указанные в таблице 6.

**К1. Добавление акций и тарифных планов**

## Основное действующее лицо: Контент-менеджер.

## Другие участники прецедента: отсутствуют.

## Связи с другими вариантами использования: отсутствуют.

## Краткое описание: Данный вариант использования позволяет Контент-менеджеру добавлять в базу данных необходимые акции и тарифные планы, разработанные и поступившие от Менеджера. Контент-менеджер заносит в БД текст условий акции, текст вопросов и соответствующих им возможных вариантов ответа.

## **К2. Актуализация списка операторов**

Основное действующее лицо: Контент-менеджер.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: отсутствуют.

Краткое описание: Данный вариант использования позволяет Контент-менеджеру внести в базу данных информацию о новых операторах и удалить сведения об отстраненных сотрудниках.

**О1. Указание параметров опроса**

Основное действующее лицо: Оператор.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: отсутствуют.

Краткое описание: Перед проведением опроса абонента оператор указывает параметры для проведения. В обязательном порядке оператор должен выбрать акцию, по которой будет производиться опрос, список абонентов для опроса, среди которых отбираются абоненты, которым подобный звонок еще не поступал.

## **О2. Формирование результатов опроса**

Основное действующее лицо: Оператор.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: отсутствуют.

Краткое описание: Во время проведения опроса после каждого заданного вопроса оператор отмечает тот ответ, который был выбран абонентом. Все полученная информация автоматически заносятся в базу данных.

**М1. Просмотр результатов опроса**

## Основное действующее лицо: Менеджер.

## Другие участники прецедента: отсутствуют.

## Связи с другими вариантами использования: отсутствуют.

## Краткое описание: После проведения всех опросов в базе данных формируется определенный набор сведений, который отражает мнения опрошенных абонентов о действующих акциях и тарифных планах. Менеджер может просматривать и обрабатывать полученные статистические результаты.

**М2. Разработка и передача готовых акций и тарифных планов**

Основное действующее лицо: Менеджер.

Другие участники прецедента: Контент-менеджер.

Связи с другими вариантами использования: отсутствуют.

Краткое описание: После принятия решения о необходимости введения новых акций или тарифных планов, менеджер подробно прорабатывает их условия, а для дальнейшего проведения опросов разрабатывает список требуемых вопросов и соответствующих им вариантов ответа. Разработанные сведения передаются Контент-менеджеру для добавления их в базу данных.

* + 1. **Конкретизация вариантов использования**

Анализ сформулированных вариантов использования показал, что с точки зрения потенциальных рисков и архитектурной значимости наиболее существенными являются прецеденты, связанные с работой оператора.

Для дальнейшей детализации выбран прецедент «O1. Указание параметров опроса» (таблица 7).

Таблица 7 – Реестр вариантов использования

|  |  |
| --- | --- |
| **Название:** | Указание параметров опроса |
| **Цель:** | Выбор параметров для проведения опроса, а именно выбор акции, списка абонентов и конкретного абонента для опроса. |
| **Специальные требования:** | Время указания параметров не должно превышать 2 минут. |
| **Предусловия:** | Перед тем как начинается этот прецедент, оператор должен авторизоваться в системе. |
| **Постусловия:** | При успешном окончании прецедента оператор проводит опрос абонента. При неуспешном – выбирается другой ранее неопрошенный абонент из выбранного списка. |
| **Дополнительные замечания:** | Дополнительных замечаний нет. |
| **Основной поток:** | 1. Оператор выбирает из списка конкретную акцию, по которой будет производиться опрос.  2. Система отображает информацию по выбранной акции.  3. Система отображает списки абонентов, доступных для опроса.  4. Оператор выбирает из предложенного списка конкретный список абонентов для проведения опроса.  5. Система отображает элемент управления (кнопку) для выбора конкретного абонента.  6. Оператор использует элемент управления для выбора конкретного абонента.  7. Система определяет всех абонентов из выбранного списка и автоматически выбирает одного из ранее неопрошенных абонентов.  8. Система отображает информацию по абоненту и элементы управления для указания статуса звонка абоненту. |
| **Альтернативный поток:** | А: Если при выполнении п. 8 основного потока событий абонент согласился на проведение опроса, то:  1. Оператор выбирает статус «Готов к разговору».  2. Система заносит статус в соответствую запись в БД.  3. Система загружает список вопрос по выбранной акции.  4. Система отображает первый вопрос и соответствующие ему варианты ответа.  5. Оператор переходит к опросу абонента.  Б: Если при выполнении п. 8 основного потока событий абонент не ответил или не согласился на проведение опроса, то:  1. Оператор выбирает статус «Не готов к разговору», «Перезвонить» или «Не беспокоить».  2. Система заносит статус в базу данных.  3. Переход к п. 5 основного потока событий. |

* 1. **Разработка технического задания**

**2.3.1 Общее описание системы**

**Наименование системы**

Полное наименование системы: Система автоматизации работы оператора Центра обработки вызовов службы исходящего телемаркетинга.

Условное обозначение системы: АИС «Телемаркетинг».

**Номер договора**

Договор №152637 от 1 октября 2014 года на разработку, внедрение и сопровождение веб-приложения, служащего для автоматизации работы оператора Центра обработки вызовов службы исходящего телемаркетинга.

**Наименования Разработчика**

Разработчик: Закрытое акционерное общество «Мастер IT»

Адрес: 430008, Саранск, ул. Косарева, д.46

Тел.: (8342) 25-10-12, факс: (8342) 43-56-58

Банковские реквизиты: ЗАО «Мастер IT», ИНН 7501004386, р/сч № 40603410800020007830 в АКБ Сбербанк России, БИК 044573589, корр. счет № 30101820400000004268

**Наименования Заказчика**

Заказчик: Открытое акционерное общество «ЛинкЛайф»

Адрес: 103255, Москва, ул. Авиамоторная, д.25

Тел.: (095)945-94-45 (095) 945-44-24, факс: (095)945-45-59

Банковские реквизиты: ОАО «ЛинкЛайф», ИНН 7501004335, р/сч № 40603410800020004053 в АКБ Сбербанк России, БИК 044579737, корр. счет № 30101820400000000418

**Основание для проведения работ**

Основанием для проведения работ по созданию АИС "Телемаркетинг" являются следующие документы:

* Договор № 152637 от 01.10.2014
* Приказ №57 от 15.09.2014
* Распоряжение №89 от 25.09.2014

**Сроки начала и окончания работ**

Дата начала работ: 01.11.2014

Дата окончания работ: 15.01.2015

**Источники и порядок финансирования работ**

Финансирование работ осуществляется из средств ОАО «ЛинкЛайф». Порядок финансирования работ определяется условиями Договора № 152637 от 01.10.2014 г.

**Порядок оформления и предъявления Заказчику результатов работ**

Работы по созданию Системы производятся и принимаются поэтапно.

По окончании каждого из этапов работ Разработчик представляет Заказчику соответствующую документацию и подписанный со стороны Разработчика Акт сдачи-приемки работ.

* + 1. **Список действующих лиц**
* Контент-менеджер – осуществляет добавление сведений в базу данных: добавляет новые акции и предложения, обновляет список абонентов.
* Оператор – указывает параметры для проведения опроса (выбирает акцию, список абонентов и конкретного абонента для опроса). Проводит опрос по выбранной ранее акции и фиксирует полученные результаты.
* Менеджер – просматривает и обрабатывает статистические результаты, полученные в результате ответов абонентов. На их основе составляют отчеты и принимают решении о разработке новых акций и тарифных планов.
  + 1. **Глоссарий**

**Понятия, используемые при описании исходной информации**

**Компания – провайдер услуг «ЛинкЛайф».**

**Опрос – проводимое оператором посредством телефонии анкетирование абонента.**

**Акция – это некое мероприятие (комплексное сочетание услуг), проводимое в компании с целью увеличения объемов продаж и повышения внимания имеющихся абонентов.**

**Абонент – физическое лицо, в отношении которого производятся телемаркетинговые мероприятия.**

**Список абонентов – максимально необходимый для проведения телемаркетинговых мероприятий набор данных о абонентах, представленный в структурированном виде.**

**Оператор – сотрудник ЕРЦОВ, непосредственно осуществляющий мероприятия телемаркетинга.**

**Менеджер – специалист, профессионально занимающийся торговой деятельностью.**

**Контент-менеджер – редактор, в обязанности которого входит полноценное наполнение базы данных текстовой и другими видами информации.**

**Телемаркетинг – вид прямого маркетинга, при котором общение оператора и клиента происходит с помощью средств телефонии.**

**Исходящий телемаркетинг – это обзвон интересующих компанию клиентов, в ходе которого подготовленный оператор собирает или предоставляет аудитории необходимую информацию.**

**АИС «Телемаркетинг» – программное обеспечение, выполненное в виде веб-приложения, предназначенное для автоматизации процесса телемаркетинга.**

**Веб-приложение – клиент-серверное приложение, в котором клиентом выступает браузер, а сервером – веб-сервер.**

**Понятия, используемые при планировании**

**Статус звонка – состояние звонка абоненту с точки зрения согласия. В первом приближении будем различать следующие статусы: «Готов к разговору», «Не готов к разговору», «Перезвонить» и «Не беспокоить».**

**Допустимый интервал – сроки, в которые может быть осуществлен опрос абонента, не нарушая технологической последовательности и сроков.**

**Критический срок исполнения работ – срок начала работы, перенос работы позже которого приводит к срыву сроков наполнения базы данных, исходя из технологических ограничений.**

**Коллизия – противоречивая информация, приводящая либо к невозможности выполнения работы, либо к нарушению объективности получаемых результатов. Коллизии возможны в случаях:**

* + **некорректного введения в базу данных информации об условиях акций и тарифных планов, списков абонентов, вопросов и ответов;**
  + **некорректной организации связи вопроса с возможными вариантами ответа.**

**Понятия, используемые при описании документов**

**Список статусов и комментариев – таблица, содержащая информацию обо всех статусах звонка.**

**Список кодов филиалов – таблица, содержащая информацию о номерах филиалов, обозначающих списки абонентов.**

**Форма отчета по абонентской базе – отчет, содержащий информацию об ответах всех абонентов базы.**

**Форма статистики по обработке списков абонентов – отчет, содержащий информацию о результатах проведения опросов всех списков абонентов.**

* + 1. **Структура проектируемой системы**

**Назначение системы**

АИС «Телемаркетинг» - программное обеспечение, выполненное в виде веб-приложения, предназначенное для:

* автоматизации работ оператора Центра обработки вызовов в процессе проведения исходящего телемаркетинга;
* получения оперативной информации о действующих акциях и тарифных планах;
* получения сведений об актуальных списках абонентов;
* записи ответов абонентов, полученных в ходе опроса;
* формирования и передачи статистических данных менеджерам для последующего анализа.

**Цели создания системы**

* создание единого механизма осуществления работ в процессе телемаркетинга;
* создание функционально полного механизма подготовки, согласования и хранения различных;
* обеспечение полноты, достоверности и оперативности информационной поддержки обработки ответов абонентов, включающих информирование по действующим акциям и тарифным планам компании, запись ответов абонентов, обработку статистическим данных и прочее.

**Характеристика объекта автоматизации**

Объектом автоматизации является набор процессов, указанных в «Разработка технического проекта», которые имеют место в рамках осуществления работы с абонентами провайдера услуг, а также ряда дополнительных участников, выполняющих функции информационной поддержки, контроля, а также нормативного регулирования объекта автоматизации.

В приложении АИС «Телемаркетинг» предусмотрена возможность построения различных отчетов.

Основные типы отчетов:

* Сводный отчет об абонентах, согласившихся на подключение тарифного плана в рамках акции;
* Сводный отчет об успешности проведения определенной акции.
  + 1. **Требования для развертывания системы**

**Применяемые стандарты**

Система должна соответствовать всем стандартам интерфейса пользователя Microsoft Windows.

**Требования к среде выполнения**

Минимальные системные требования для сервера:

* 2 Гб памяти;
* 200 Mb свободного дискового пространства;
* процессор с тактовой частотой 2500 MHz;
* Операционная система Microsoft Windows Server 2008.

Минимальные системные требования для клиентской части:

* 500 Mb памяти;
* 50 Mb свободного дискового пространства;
* процессор с тактовой частотой 1500 MHz;
* Операционная система Windows 7.

**Требования к СУБД и доступу к данным.**

В ядре системы должна быть представлена промышленная СУБД реляционного доступа.

* + 1. **Прочие требования**

**Требования к структуре системы**

АИС «Телемаркетинг» предназначена для автоматизации работы оператора Центра обработки вызовов службы исходящего телемаркетинга. Автоматизации подлежат операции указания параметров перед проведением опроса, получения перечня вопрос и соответствующих им ответов, консультация абонентов по действующим акция и тарифным планам, проведение опроса абонента, регистрация полученных ответов, формирование отчетов об успешности проведения акции и об абонентах, согласившихся на подключение.

Функциональная структура Системы должна включать основные прикладные подсистемы, выполняющие задачи автоматизации обмена информацией и обработки запросов операторов и менеджеров, для корректной консультации абонентов и проведения опроса, а также для предоставления информационных услуг.

**Требования к функциональности**

В АИС должны быть представлены таблица ролей пользователей (Оператор, Менеджер, Контент-менеджер) и таблица пользователей. Должна быть возможность назначения пользователю роли.

**Требования к применимости**

Интерфейс АРМ «Телемаркетинг» для оператора и менеджера должен обладать свойствами удобства и интуитивной ясности и не требовать дополнительной подготовки пользователей. Интерфейс для контент-менеджера должен быть рассчитан на предварительно обученного специалиста, хорошо ориентирующегося в ПО и компьютерных интерфейсах; время обучения не должно превышать 1 рабочий день.

**Требования к надежности**

АРМ «Телемаркетинг» должен быть доступен круглосуточно.

Время, затрачиваемое на обслуживание системы не должно превышать 3% от общего времени работы.

Среднее время безотказной работы – 10 рабочих дней. Максимальная норма ошибок или дефектов – 1 ошибка на десять тысяч строк кода.

**Требования к производительности**

Система должна быть способна поддерживать минимум 30 одновременно работающих пользователей, связанных с общей базой данных.

Время отклика для типичных задач – не более 5 секунд, для сложных задач – не более 20 секунд.

**Требования к пригодности к эксплуатации**

Система должна быть способна поддерживать минимум 30 одновременно работающих пользователей, связанных с общей базой данных и иметь возможность увеличить их количество на случай увеличения штата сотрудников предприятия. В настоящее время в службе телемаркетинга имеется 3 менеджера, 6 операторов, 1 контент-менеджер.

Пользователи АСУ должны иметь базовые навыки работы с операционными системами Microsoft (Microsoft Windows 7, 8, 8.1).

Техническое обслуживание и администрирование оборудования АСУ должно выполняться специалистами, имеющими соответствующую квалификацию и навыки выполнения работ.

Все администраторы АСУ должны иметь квалификацию «инженер» и обязательные навыки администрирования сети на основе операционной системы Microsoft Windows 7.

Обновление версий должно осуществляться в автоматизированном режиме на основе системы контроля версий и системы (сервера) обновления версий на рабочих местах пользователей.

**Требования к способам и средствам связи для информационного обмена между компонентами системы**

Информационный обмен между подсистемами должен осуществляться через единое информационное пространство и посредством использования стандартизированных протоколов и форматов обмена данными.

Все компоненты подсистем АСУ должны функционировать в пределах единого логического пространства, обеспеченного интегрированными средствами серверов данных и серверов приложений.

**Требования к совместимости со смежными системами**

Программное обеспечение системы должно обеспечивать интеграцию и совместимость на информационном уровне с другими системами. Информационная совместимость должна обеспечивается, на уровне экспорта-импорта XML-документов.

Требования к составу данных и режимам информационного обмена между подсистемами АСУ и системами, эксплуатирующимися на объекте автоматизации, определяются в общем регламенте взаимодействия.

**Требования к перспективам развития системы**

АСУ должна:

* иметь длительный жизненный цикл,
* быть построена с использованием стандартизованных и эффективно сопровождаемых решений,
* быть реализована как открытая система, и должна допускать наращивание функциональных возможностей,
* обеспечивать возможность модернизации как путем замены технического и общего программного обеспечения (ПО), так и путем совершенствования информационного обеспечения.

**Требования к показателям назначения**

Целевое назначение системы должно сохраняться на протяжении всего срока эксплуатации АСУ ОАО «ЛинкЛайф». Срок эксплуатации АСУ ОАО «ЛинкЛайф» определяется сроком устойчивой работы аппаратных средств вы-числительных комплексов, своевременным проведением работ по замене (обновлению) аппаратных средств, по сопровождению программного обеспечения системы и его модернизации. Время выполнения запросов информации в АСУ определяется на стадии проектирования системы.

Специальные требования к вероятностно-временным характеристикам, при которых сохраняется целевое назначение АСУ ОАО «ЛинкЛайф», определяются соответствующими требованиями к прикладным системам.

Прочие показатели назначения АСУ разрабатываются после проведения предпроектного обследования.

* 1. **Построение дополнительных диаграмм**
     1. **Диаграмма последовательности**

На диаграммах последовательности изображаются исключительно те объекты, которые непосредственно участвуют во взаимодействии и не показываются возможные статические ассоциации с другими объектами. Для диаграммы последовательности ключевым моментом является именно динамика взаимодействия объектов во времени.

Рассмотрим диаграмму последовательности для прецедента «O1. Указание параметров опроса» (рисунок 5).

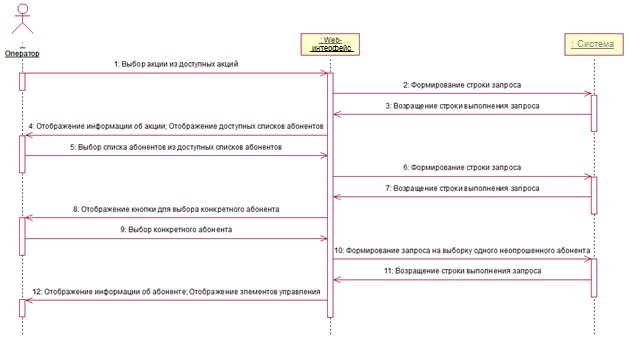


Рисунок 5 – Диаграмма последовательности для прецедента «O1. Указание параметров опроса»

В данном случае в качестве актера выбран Оператор. Данная диаграмма наглядно иллюстрирует выполнение операций взаимодействия Оператора, Web-приложения и СУБД во времени при указании параметров опроса.

После указания значения параметра система отображает информацию для указания значения следующего параметра.

**2.4.2 Построение диаграммы состояний**

Данная диаграмма позволяет отобразить алгоритм реализации выполняемых системой операций.

Составные части диаграммы:

* Действия. К ним относятся: выбор акции, списка абонентов, абонента, отображение информации по акции, доступных списков абонентов, информации по абоненту, сообщений и элементов управления.
* Точки принятия решения – есть ли неопрошенные абоненты.
* Полоса синхронизации – позволяет указать, какие действия допускают единое выполнение или логическое объединение. Разделение – после того, как последующий вопрос не найден, система отображает как соответствующее сообщение, так и доступные списки абонентов.
* Начальное и конечное действие.

Рассмотрим диаграмму состояний для прецедента «O1. Указание параметров опроса» (рисунок 6).

Идея диаграммы – указание оператором значений параметров для проведения опроса абонента. После выбора акции система предоставляет возможность выбора списка абонентов. После выбора списка система анализирует наличие в этом списке неопрошенных абонентов. Если абоненты не найдены, отображает соответствующее сообщение и предлагает указать другой список абонентов, иначе отображает элемент управления (кнопку) для выбора конкретного абонента. После выбора абонента система отображает информацию об абоненте и прочие элементы для осуществления дальнейших действий.

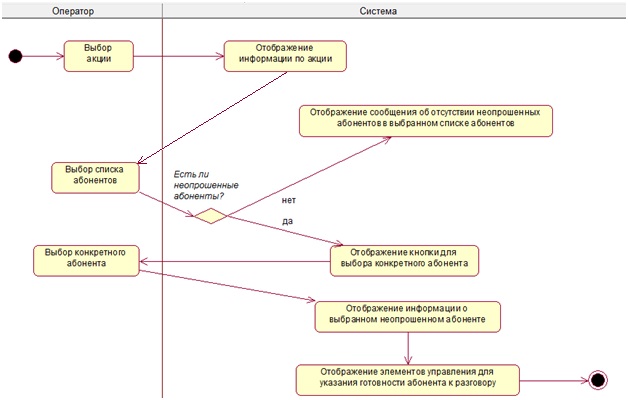


Рисунок 6 — Диаграмма состояний для описания возможных состояний прецедента «O1. Указание параметров опроса»

* 1. **Построение диаграммы классов**

Диаграммы классов является основным логическим представлением модели и служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений.

Данная диаграмма состоит из трех видов классов. Граничный или интерфейсный используется для связи актера и системы.

Управление применяется при реализации характеристик поведения системы, присущих одному или нескольким прецедентам.

Класс сущности моделирует структуру данных.

Рассмотрим диаграмму классов (рисунок 7).

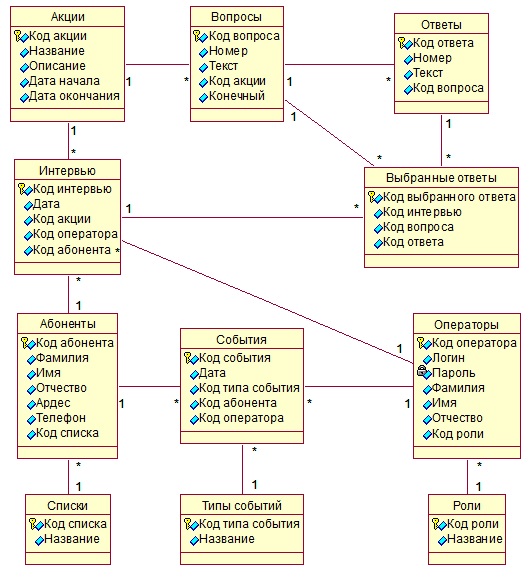


Рисунок 7 – Диаграмма классов

* 1. **Проектирование базы данных**

ER диаграмма — это модель данных, позволяющая описывать концептуальные схемы. Она предоставляет графическую нотацию, основанную на блоках и соединяющих их линиях, с помощью которых можно описывать объекты и отношения между ними какой-либо другой модели данных. В этом смысле ER-модель является мета-моделью данных, то есть средством описания моделей данных

В итоге, структура получившейся базы данных достаточно проста. Таблица «Операторы» содержит в себе всю информацию об операторах системы, «Списки» — о существующих группах абонентов, «Абоненты» — об абонентах компании, «Акции» — о существующих в организации акциях и тарифных планах. Таблицы «Вопросы» и «Ответы» используются для хранения структуры вопросов и соответствующих им возможных вариантов ответов.

Вспомогательная таблица «События» применяется для отражения факта указания готовности абонента перед проведением опроса, «Интервью» — факта проведения опроса абонента, «Выбранные ответы» — для отражения выбранных абонентом вариантов ответов. Кроме того, разработаны две дополнительные таблицы: «Типы событий» и «Роли». Они предназначены для хранения перечня существующих типов событий и ролей соответственно.

После установления связей между таблицами можно наблюдать получившуюся схему данных (рисунок 8).

При связывании таблиц учитывается целостность данных, возможность каскадного обновления и удаления записей.

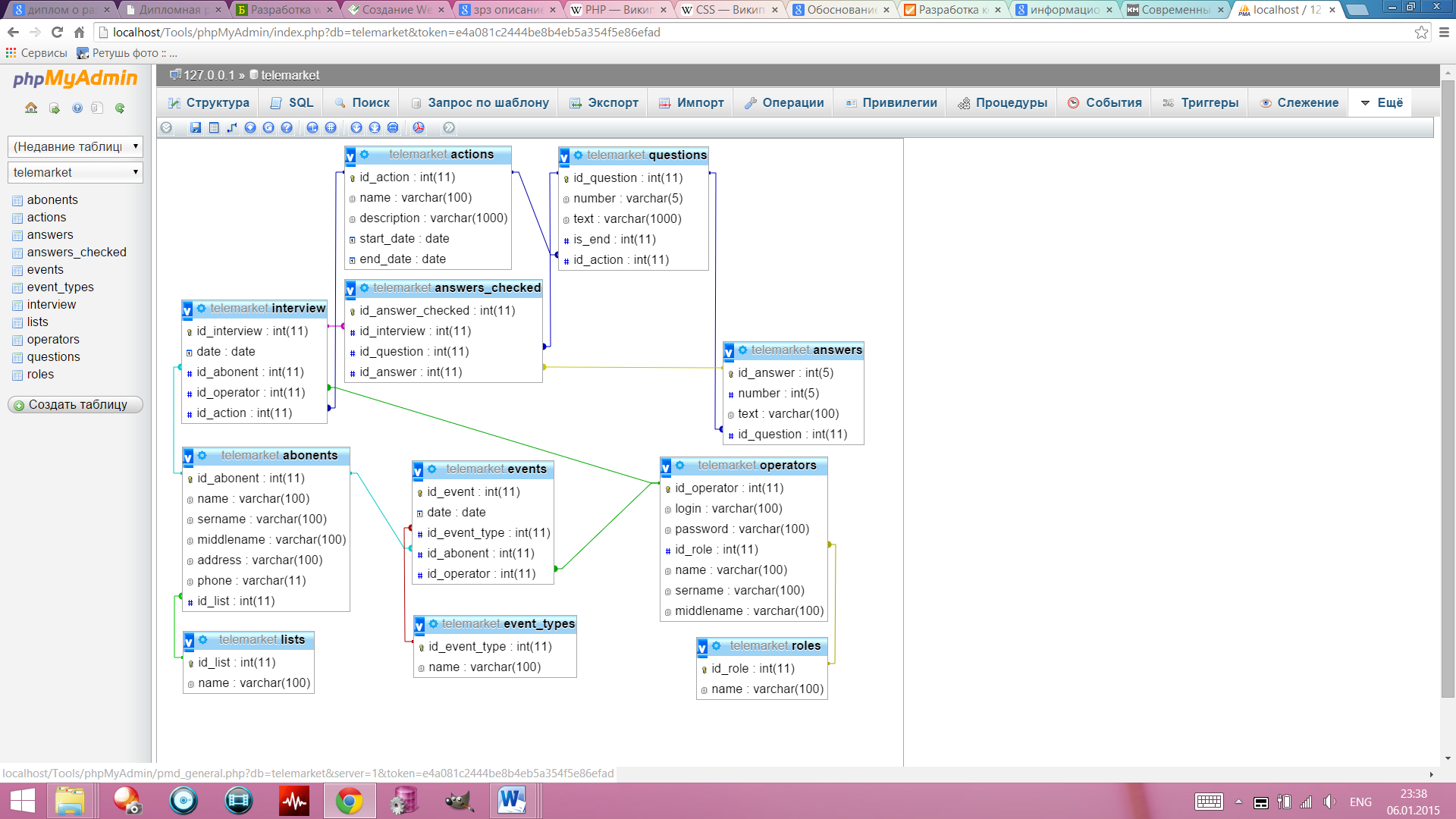


Рисунок 8 –ER-диаграмма

* 1. **Планирование развертывания системы**

При проектировании программного обеспечения АИС «Телемаркетинг» основным принципом было максимальное использование ранее разработанных баз данных. Поэтому было принято решение о реализации данного ПО в виде Web-приложения и размещении его на сервере компании.

Диаграмма развертывания системы представлена на рисунке 9.

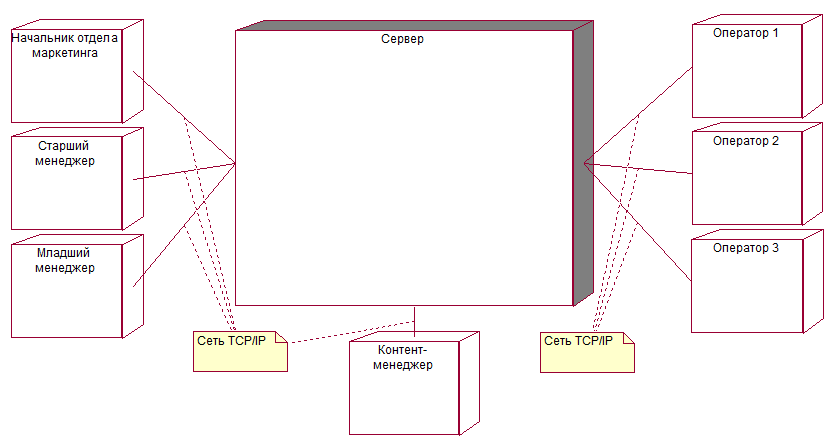
****

Рисунок 9 — Диаграмма развертывания системы

**3 Выбор средств разработки для разрабатываемой системы**

* 1. **Выбор средств и инструментов для разработки базы данных**

MySQL – это система управления базами данных.

База данных представляет собой структурированную совокупность данных. Эти данные могут быть любыми – от простого списка предстоящих покупок до перечня экспонатов картинной галереи или огромного количества информации в корпоративной сети. Для записи, выборки и обработки данных, хранящихся в компьютерной базе данных, необходима система управления базой данных, каковой и является ПО MySQL. Поскольку компьютеры замечательно справляются с обработкой больших объемов данных, управление базами данных играет центральную роль в вычислениях. Реализовано такое управление может быть по-разному – как в виде отдельных утилит, так и в виде кода, входящего в состав других приложений.

MySQL – это система управления реляционными базами данных.

В реляционной базе данные хранятся не все скопом, а в отдельных таблицах, благодаря чему достигается выигрыш в скорости и гибкости. Таблицы связываются между собой при помощи отношений, благодаря чему обеспечивается возможность объединять при выполнении запроса данные из нескольких таблиц. SQL как часть системы MySQL можно охарактеризовать как язык структурированных запросов плюс наиболее распространенный стандартный язык, используемый для доступа к базам данных.

Программное обеспечение MySQL – это ПО с открытым кодом.

ПО с открытым кодом означает, что применять и модифицировать его мо-жет любой желающий. Такое ПО можно получать по Internet и использовать бесплатно. При этом каждый пользователь может изучить исходный код и изменить его в соответствии со своими потребностями.

MySQL является наиболее приспособленной для применения в среде web СУБД. Не секрет, что для исполнения приложений клиента на большинстве хостинг-площадок провайдеры предоставляют небольшое количество ресурсов (как вычислительных, так и дисковых). Поэтому для данного применения необходима высокоэффективная СУБД, обладающая при этом высокой надежностью (большинство web-приложений и сайтов должны работать в режиме 24/7).

По всем этим причинам MySQL стала стандартом в области СУБД для web, а теперь в ней развиваются возможности для использования ее в любых критичных бизнес-приложениях, то есть конкурирует на равных с такими СУБД таких производителей, как Oracle, IBM, Microsoft и Sybase.

Основные преимущества MySQL:

* Сохранение времени, необходимого для ведения записей. В случае использования СУБД много времени на просмотр все картотеки на предмет необходимости добавления новой записи не требуется. Вы просто вводите ее в систему, не заботясь о месте размещения.
* Сокращения времени, необходимого для поиска записей. При поиске данных в СУБД нет необходимости последовательно просматривать все записи, чтобы найти интересующую.
* Гибкость поиска. Нет необходимости искать записи строго в соответствии с порядком, в котором они были записаны (по фамилии, например). Информационной системе можно приказать расположить записи отсортированные в любом порядке: по фамилии, названию страховой компании. Дате последнего визита и т. д.
* Гибкость формата вывода. После того как необходимые записи были найдены, копировать записи вручную не нужно. Можно сделать запрос информационной системе на вывод нужного списка. Иногда достаточно просто распечатать информацию. В других случаях вам может понадобиться воспользоваться этими данными в другой программе.
* Одновременно многопользовательский доступ к записям. Предположим, что сразу два человека хотят одновременно просмотреть одну запись. При бумажном способе ведения дел второй кандидат всегда вынужден ждать, пока первый закончит просмотр бумаг. СУБД позволяет получить доступ к одной и той же записи одновременно.
* Удаленный доступ и передача записей в электронном виде. Бумажная технология ведения дел требует от вас быть там, где находятся сами бумаги, в противном случае кто-то должен скопировать и переслать их вам. Электронный способ ведения записей позволяет производить удаленный доступ к записям и передачу их в электронном виде.
  1. **Выбор средств и инструментов для разработки программы**

**3.2.1 Язык разметки гипертекстовых страниц HTML**

Язык разметки гипертекстовых страниц (HTML – Hypertext Markup Language) представляет собой язык, разработанный специально для создания Web-документов. Он определяет синтаксис и размещение специальных инструкций (тегов), которые не выводятся на экран, но указывают браузеру, как отображать содержимое документа. Он также используется для создания ссылок на другие документы, локальные или сетевые, например, находящиеся в сети Интернет.

Стандарт HTML и другие стандарты для Web разработаны под руководством консорциума W3C (World Wide Web Consortium). На практике на стандарт HTML большое влияние оказывает наличие тегов, предложенных и поддерживаемых наиболее известными браузерами, такими как Microsoft Internet Explorer и Netscape Navigator.

Информации о тегах HTML Compendium созданно Ron Woodall. Компендиум содержит список тегов и их атрибутов в алфавитном порядке, а также обновленную информацию о поддержке каждого из них со стороны браузеров.

Документы HTML являются обычными текстовыми ASCII-файлами. Это означает, что для их создания можно использовать любой текстовый редактор, даже с минимальными возможностями. Существуют средства редактирования, разработанные специально для написания HTML. Они позволяют экономить время, так как содержат клавиши быстрого доступа для выполнения повторяющихся операций.

Документ HTML содержит текст и встроенные теги – инструкции о структуре, внешнем виде и функции содержимого. Документ HTML разделяется на две основные части: заголовок – head и тело – body. Заголовок содержит сведения о документе: его название и методическая информация, описывающая содержимое. В теле находится само содержим документа.

Каждый тег состоит из имени, за которым может следовать список необязательных атрибутов, все они находятся внутри угловых скобок < >. Атрибуты являются свойствами, которые расширяют или уточняют функцию тега. Как правило, имя и атрибуты внутри тега не чувствительны к регистру.

Большинство тегов являются контейнерами. Это означает, что у них имеется начальный и конечный теги. Текст, находящийся между тегами, будет выполнять содержащиеся в них инструкции. Конечный тег имеет то же имя, что и начальный, но перед ним стоит слеш (/).Конечный тег никогда не содержит атрибутов. В некоторых случаях конечный тег не обязателен, и браузер определяет конец тега из контекста.

Некоторые теги не имеют завершающих тегов, потому что они используются для размещения отдельных (автономных) элементов на странице. Одним из них является тег изображения <img>, он просто помещает графику в поток страницы. Другие автономные теги - это разрыв строки (<br>), горизонтальная линия (<hr>) и теги, содержащие информацию о документе и не влияющие на содержимое, выводимое на экран, такие как <meta> и <base>.

Атрибуты добавляются в тег для расширения или модификации его действий. К одному тегу можно добавить несколько атрибутов. Если атрибуты тега следуют после имени тега, они разделяются одним или несколькими пробелами. Порядок следования не важен. Большинство атрибутов имеют значения, которые следуют за знаком равенства (=), находящимся после имени атрибута.

**3.2.2 Таблицы стилей CSS**

При разработке Web-страница должна быть доступна для максимально возможного числа пользователей. Необходимо определить наиболее часто используемое разрешение дисплея и разработать страницу таким образом, чтобы она гарантированно заполняла все рабочее пространство. Чтобы при просмотре пользователям не пришлось применять горизонтальную прокрутку. Горизонтальная прокрутка всегда затрудняет восприятие, поэтому дизайнеры традиционно ее отвергают.

Для добавления информации о любом элементе, можно использовать атрибут title.

Каскадные таблицы стилей или CSS (от английского Cascading Style Sheets) являются следствием дальнейшего развития HTML и дают возможность перейти на следующий уровень представления информации. Таблицы стилей позволяют разделить смысловое содержимое странички и его оформление.

При показе странички конкретному устройству должна быть задействована соответствующая случаю таблица стилей. Для сотового телефона и дисплея компьютера они, разумеется, должны быть разными. В первом случае используется минимальное оформление, которое позволит представить информацию наиболее оптимально и компактно. Во втором же случае в распоряжении имеется все богатство шрифтового и цветового оформления.

Таблицу стилей нужно написать всего один раз при создании сайта для каждого из устройств, на котором планируется вывод информации. К тому же таблица стилей может быть единой для целого сайта. И, следовательно, не нужно будет повторять одни и те же описания стилей на каждой из страниц.

Размещение всей стилевой информации в одном внешнем файле открывает и другие полезные возможности – ведь изменив содержимое только одного стилевого файла, возможно в считанные секунды сменить весь дизайн сайта. Причем никаких других переделок не понадобится.

CSS4 (Cascading Style Sheets, Level 4) – самая последняя рекомендация по каскадным таблицам стилей, предоставляет механизмы для улучшенной интерпретации страниц неграфическими и не визуальными устройствами.

**3.2.3 Язык написания сценариев PHP**

PHP (от английского Hypertext Preprocessor) – скриптовый язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов.

Язык и его интерпретатор разрабатываются группой энтузиастов в рамках проекта с открытым кодом.

В области веб-программирования, в частности серверная часть, PHP — один из популярных сценарных языков благодаря своей простоте, скорости выполнения, богатой функциональности, кроссплатформенности и распространению исходных кодов на основе лицензии PHP.

Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств для разработки веб-приложений. Основные из них:

- автоматическое извлечение POST и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы;

- взаимодействие с большим количеством различных систем управления базами данных (MySQL, Oracle, Microsoft SQL Server и др.);

- автоматизированная отправка HTTP-заголовков;

- работа с HTTP-авторизацией;

- работа с cookies и сессиями;

- работа с локальными и удалёнными файлами, сокетами;

- обработка файлов, загружаемых на сервер.

В настоящее время PHP используется сотнями тысяч разработчиков. Согласно рейтингу, базирующемся на данных поисковых систем, PHP находился на 5 месте среди языков программирования.

**3.2.4 Сценарный язык JavaScript**

JavaScript — прототипно-ориентированный сценарный язык программирования. Является диалектом языка ECMAScript.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

Название «JavaScript» является зарегистрированным товарным знаком компании Oracle Corporation.

JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам — функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания — что придает языку дополнительную гибкость.

Несмотря на схожий с Си синтаксис, JavaScript имеет коренные отличия:

* объекты, с возможностью интроспекции;
* функции как объекты первого класса;
* автоматическое приведение типов;
* автоматическая сборка мусора;
* анонимные функции.

В языке отсутствуют такие полезные вещи, как:

* модульная система: JavaScript не предоставляет возможности управлять зависимостями и изоляцией областей видимости;
* стандартная библиотека: в частности, отсутствует интерфейс программирования приложений по работе с файловой системой, управлению потоками ввода-вывода, базовых типов для бинарных данных;
* стандартные интерфейсы к веб-серверам и базам данных;
* система управления пакетами, которая бы отслеживала зависимости и автоматически устанавливала их.

**4 Реализация разрабатываемой программы**

**4.1 Интерфейсы информационной системы**

Вход в систему осуществляется через интернет браузер персонального компьютера, подключенного к сети, по адресу сайта: /arm\_telemarketing.

После открытия Web-приложения пользователю будет предложен интерфейс стартовой страницы с формой для авторизации (рисунок 10). Здесь представлен информационный минимум и возможность авторизации для зарегистрированного пользователя.

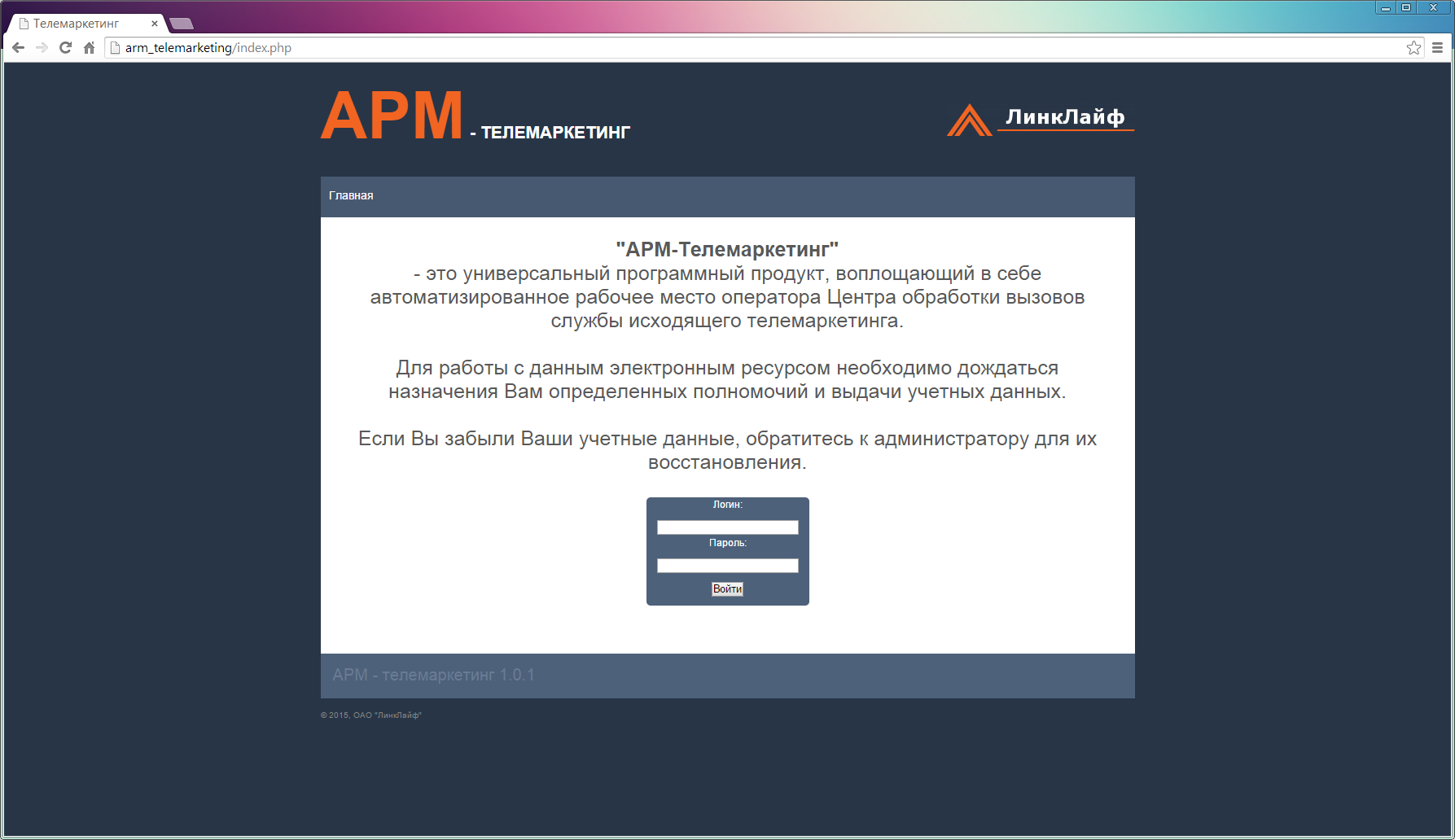


Рисунок 10 – Интерфейс стартовой страницы с формой для авторизации

При авторизации происходит процесс проверки прав, которыми наделен пользователь. Пользователю, прошедшему авторизацию, будет доступен один из трех возможных интерфейсов: интерфейс оператора, менеджера или контент-менеджера.

Интерфейс оператора (рисунок 11) предоставляет максимально возможный перечень информационных ресурсов зарегистрированному пользователю. После выбора акции, по которой оператор будет проводить опрос, списка абонентов, система определяет наличие в базе ранее неопрошенных абонентов. Если подобных абонентов в базе не оказалось, система выводит уведомляющее сообщение (рисунок 12), иначе – отображает информацию о первом найденном и элемент ыдля указания готовности к опросу. После начала опроса на форме отображается вопрос и соответствующие ему варианты ответа. После ответа на последний вопрос система уведомляет о завершении опроса, и вновь предоставляется возможность выбора акции и списка абонтов.

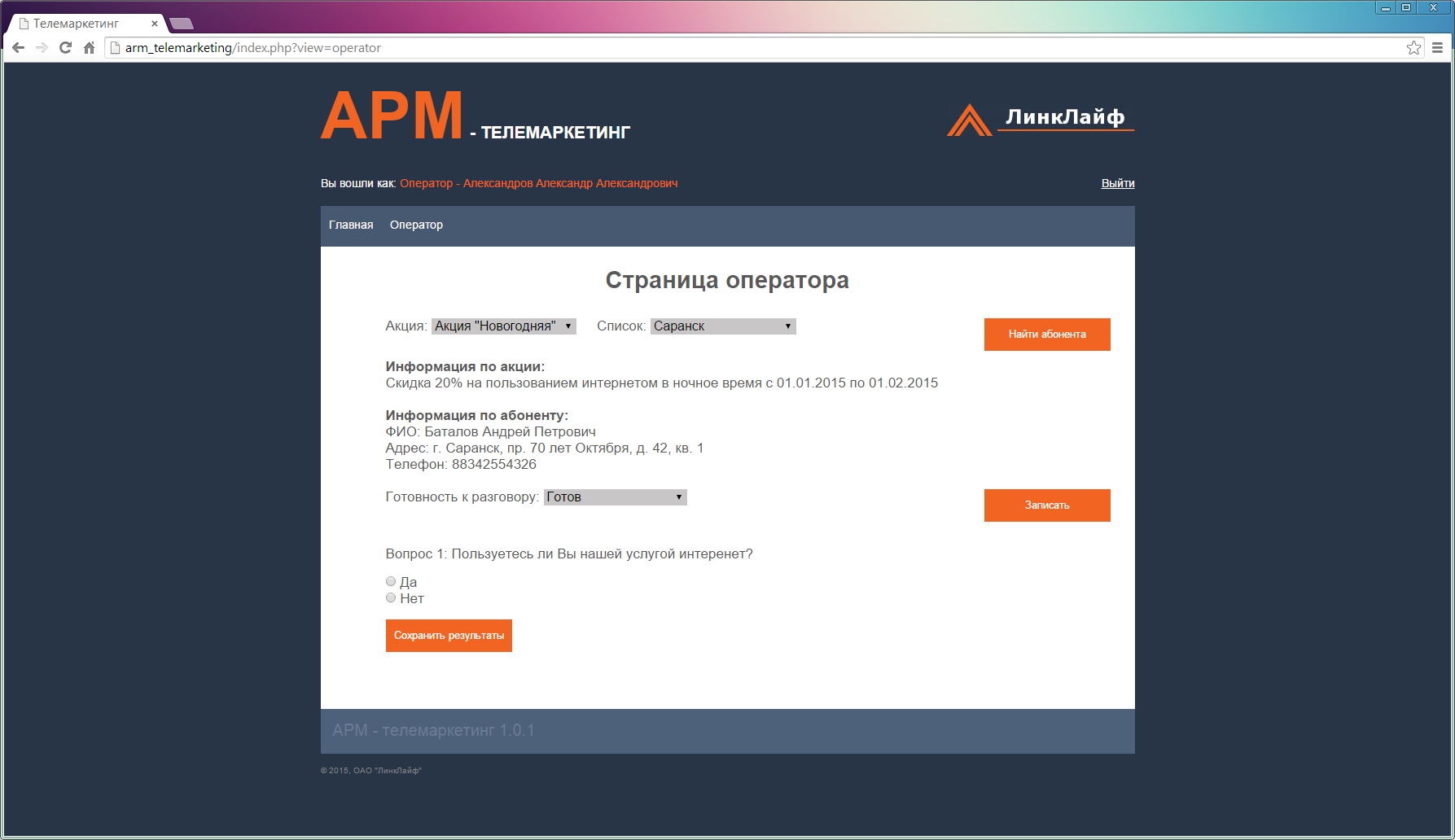


Рисунок 11 – Интерфейс оператора

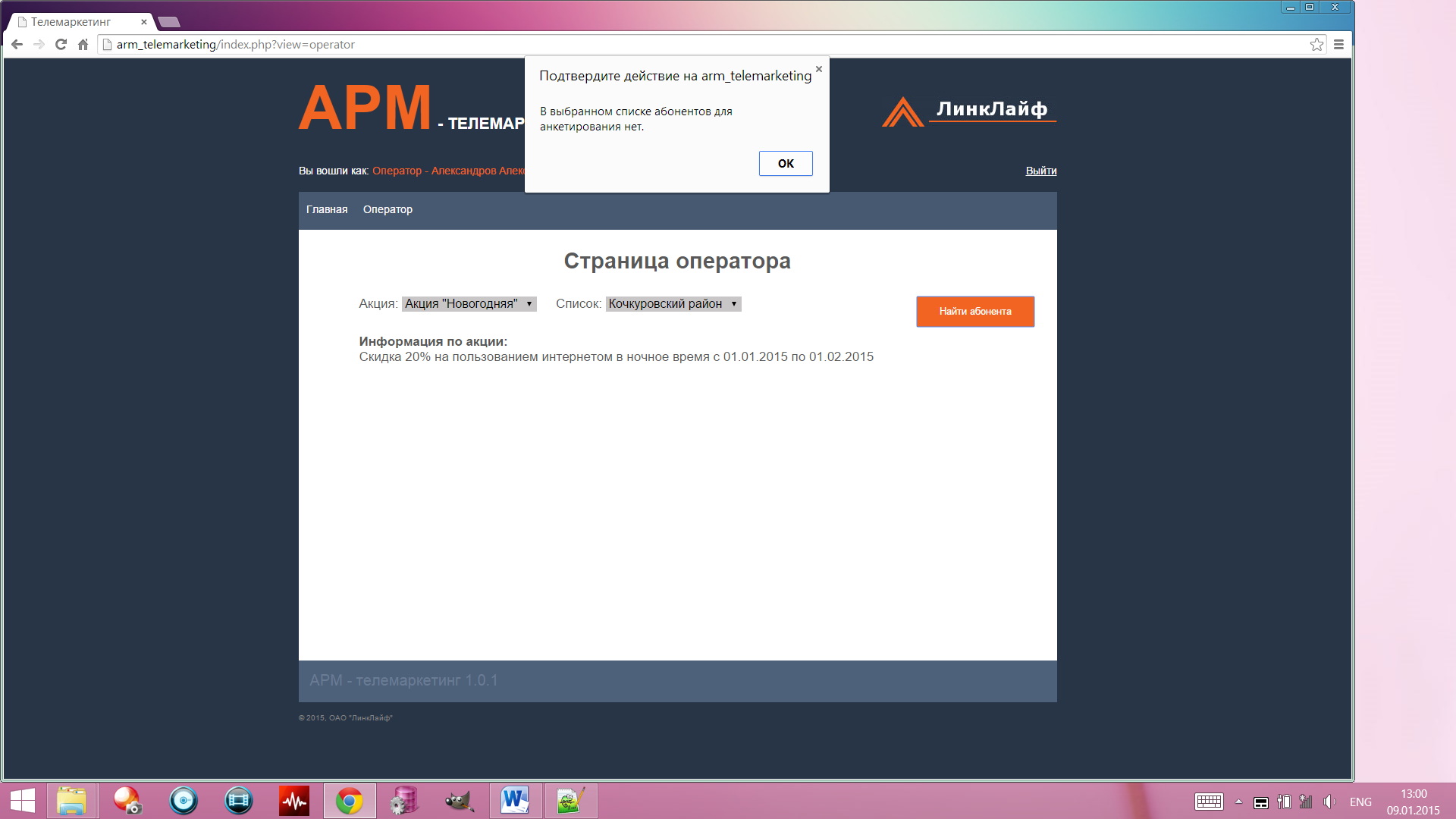


Рисунок 12 – Уведомление об отсутствии доступных для опроса абонентов

Интерфейс менеджера (рисунок 13) предоставляет возможность формирования 2 типов отчетов: по оператору и по рейтингу акций.

Принцип формирования отчетов аналогичен. На страницах представлены элементы управления для указания параметров, по котрым будет сформирован тот или иной отчет.

После нажатия кнопки «Сформировать» выполняется запрос к базе данных по указанным параметрам. В результате отображается общее количество полученных строк и сами данные в виде таблицы (рисунок 14, рисунок 15).

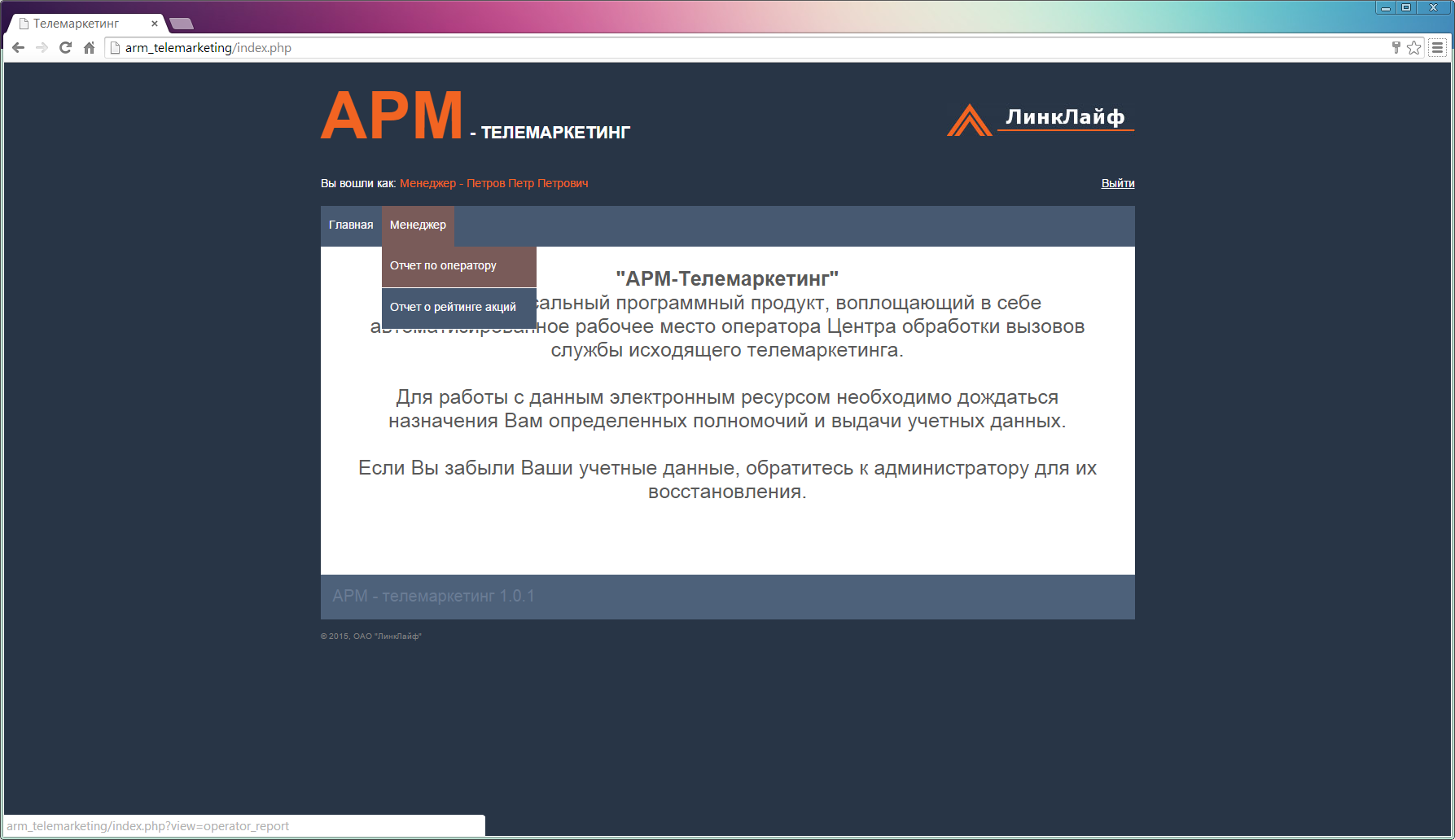


Рисунок 13 – Интерфейс менеджера

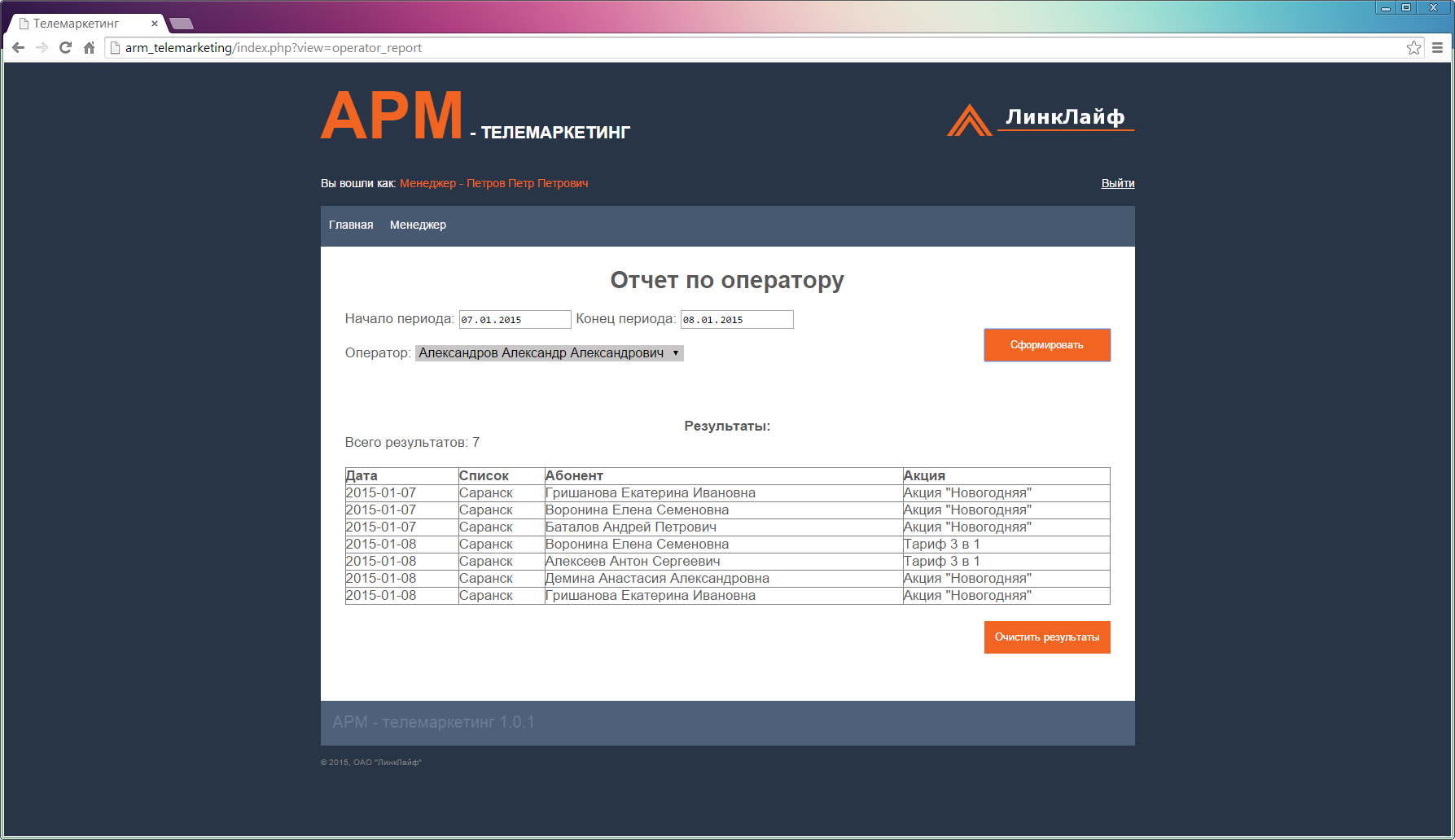


Рисунок 14 – Отчет по оператору

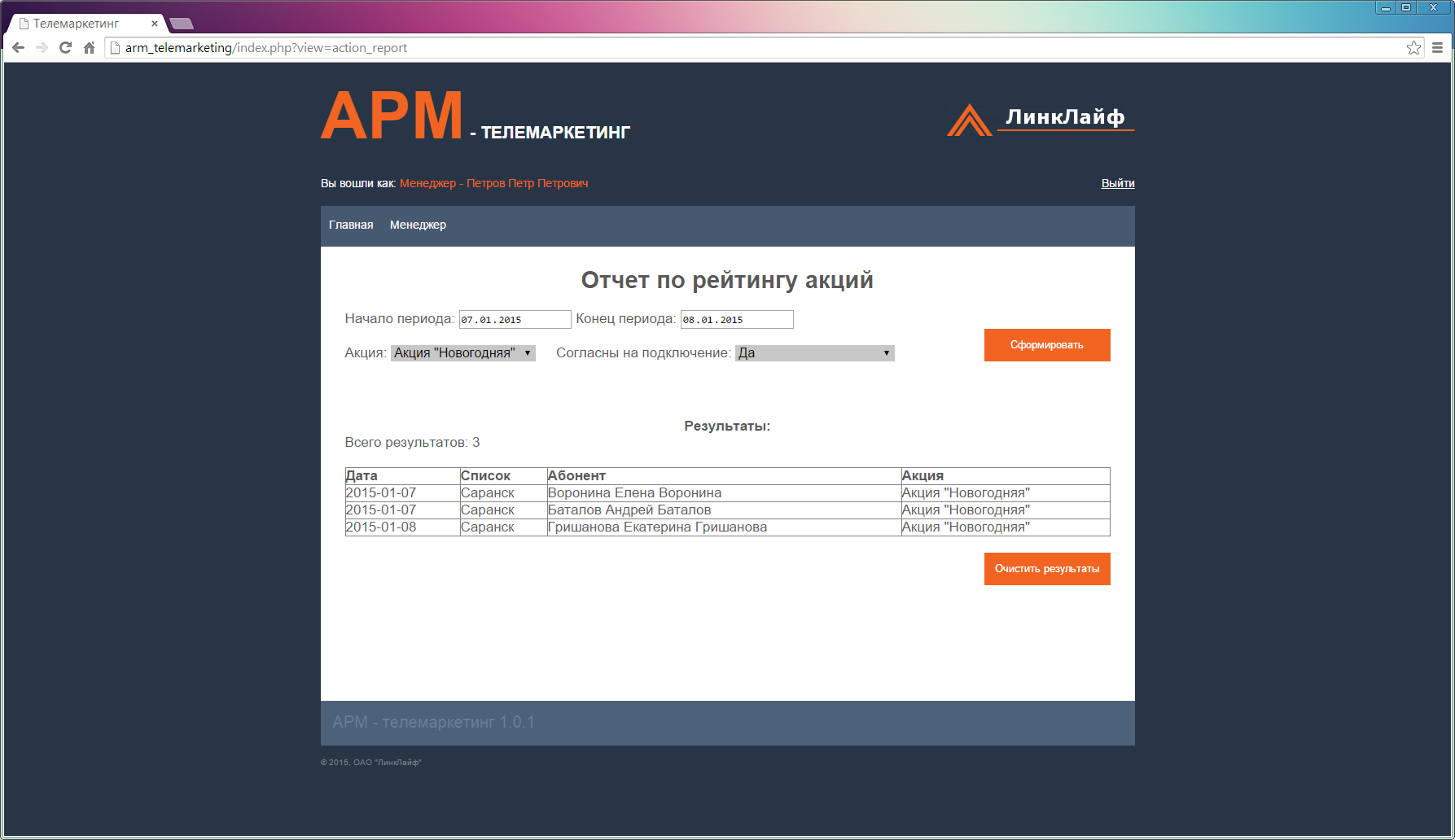


Рисунок 15 – Отчет по рейтингу акции

Интерфейс контент-менеджера (рисунок 16) используется для создания и удаления записей из таблиц базы данных. Работа со всеми таблицами происходит аналогичным образом (рисунок 17, рисунок 18, рисунок 19, рисунок 20).

В пункте «Создание» необходимо заполнить все поля и после нажатия кнопки «Создать» запись появится в соответствующей таблице базы данных.

В пункте «Удаление» требуется в раскрывающемся списке указать ту запись, которую необходимо удалить из базы. После нажатия кнопки «Удалить», строка из базы данных удаляется.

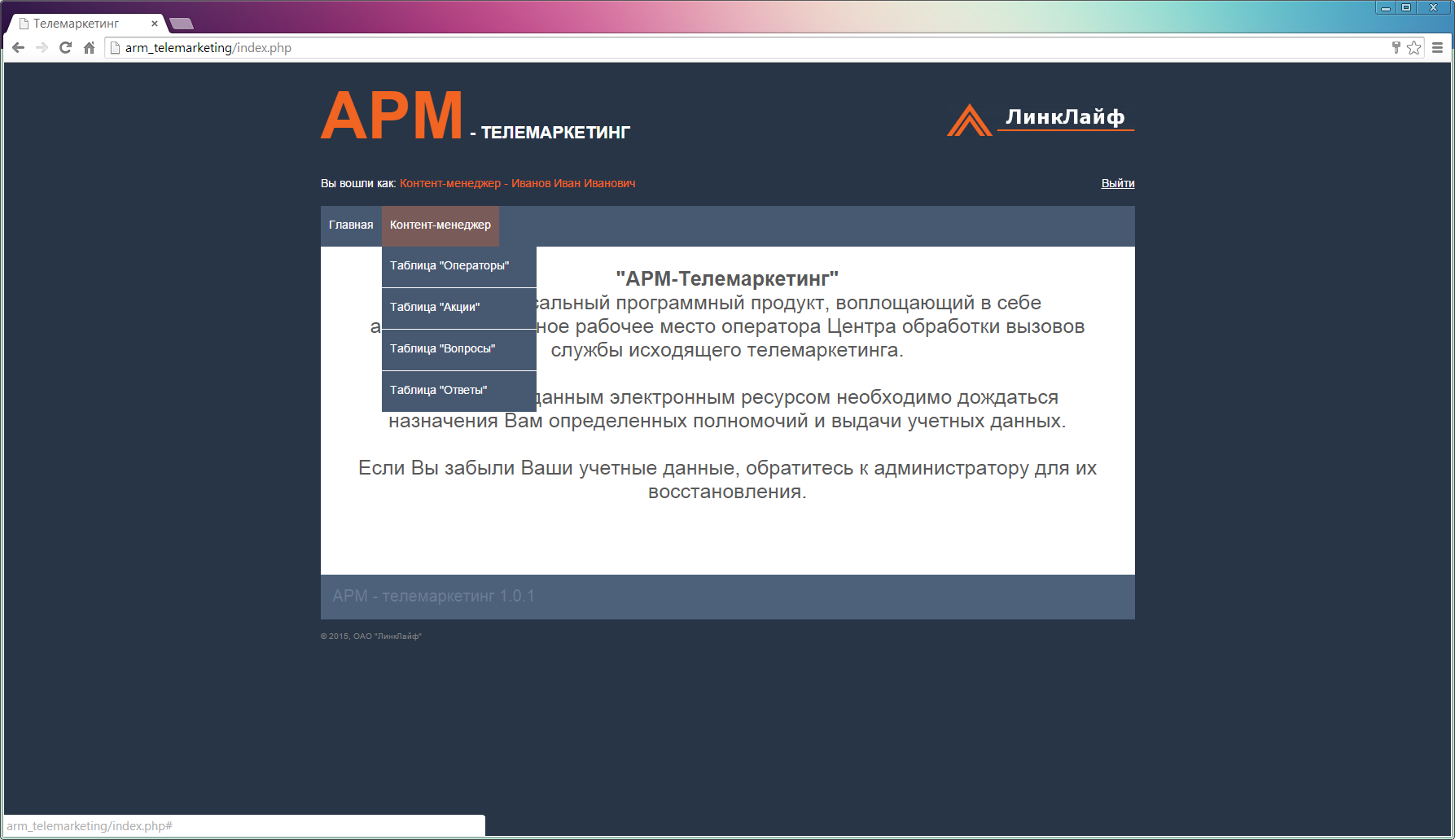


Рисунок 16 – Интерфейс контент-менеджера

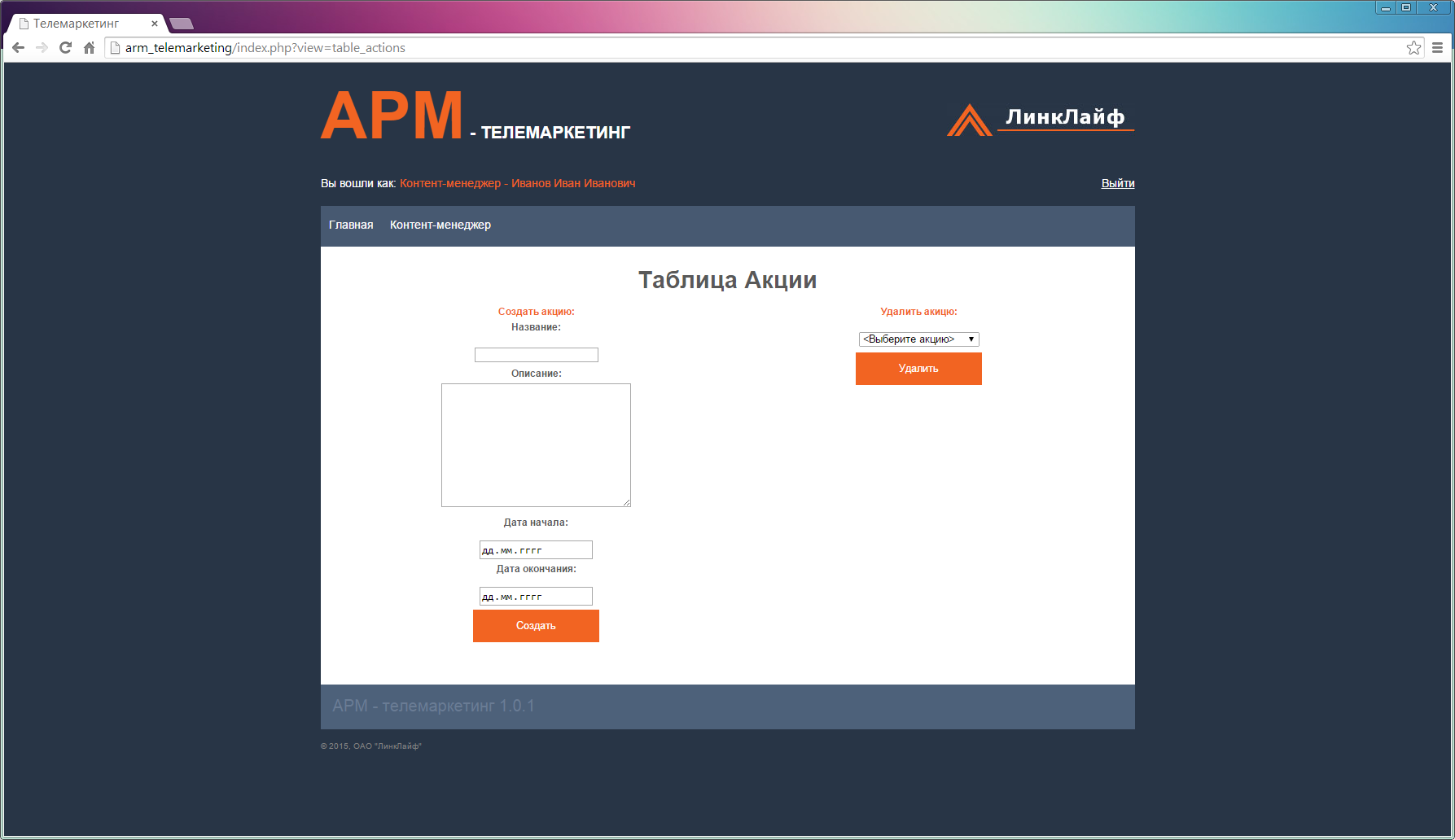


Рисунок 17 – Таблица «Акции»

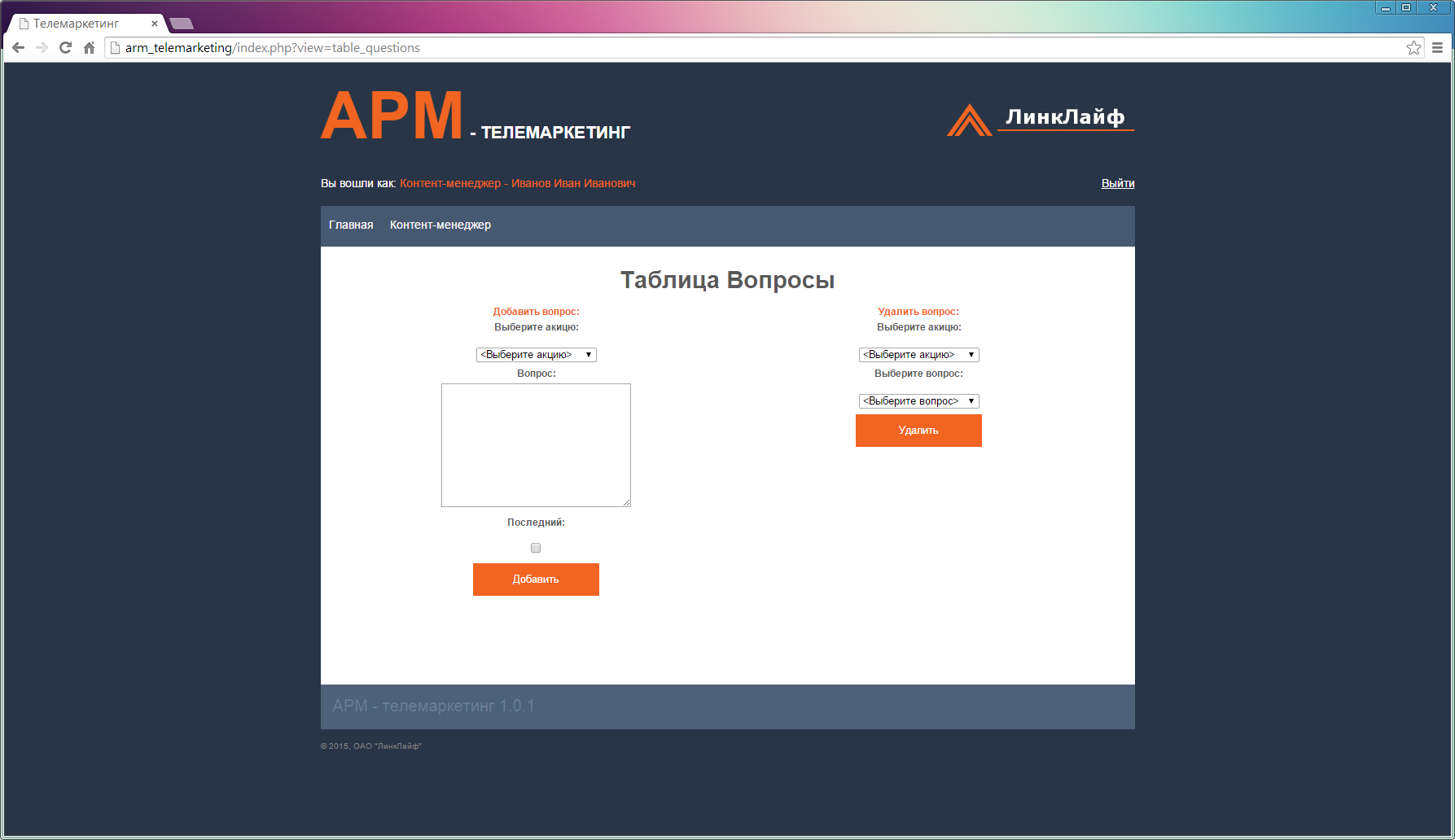


Рисунок 18 – Таблица «Вопросы»

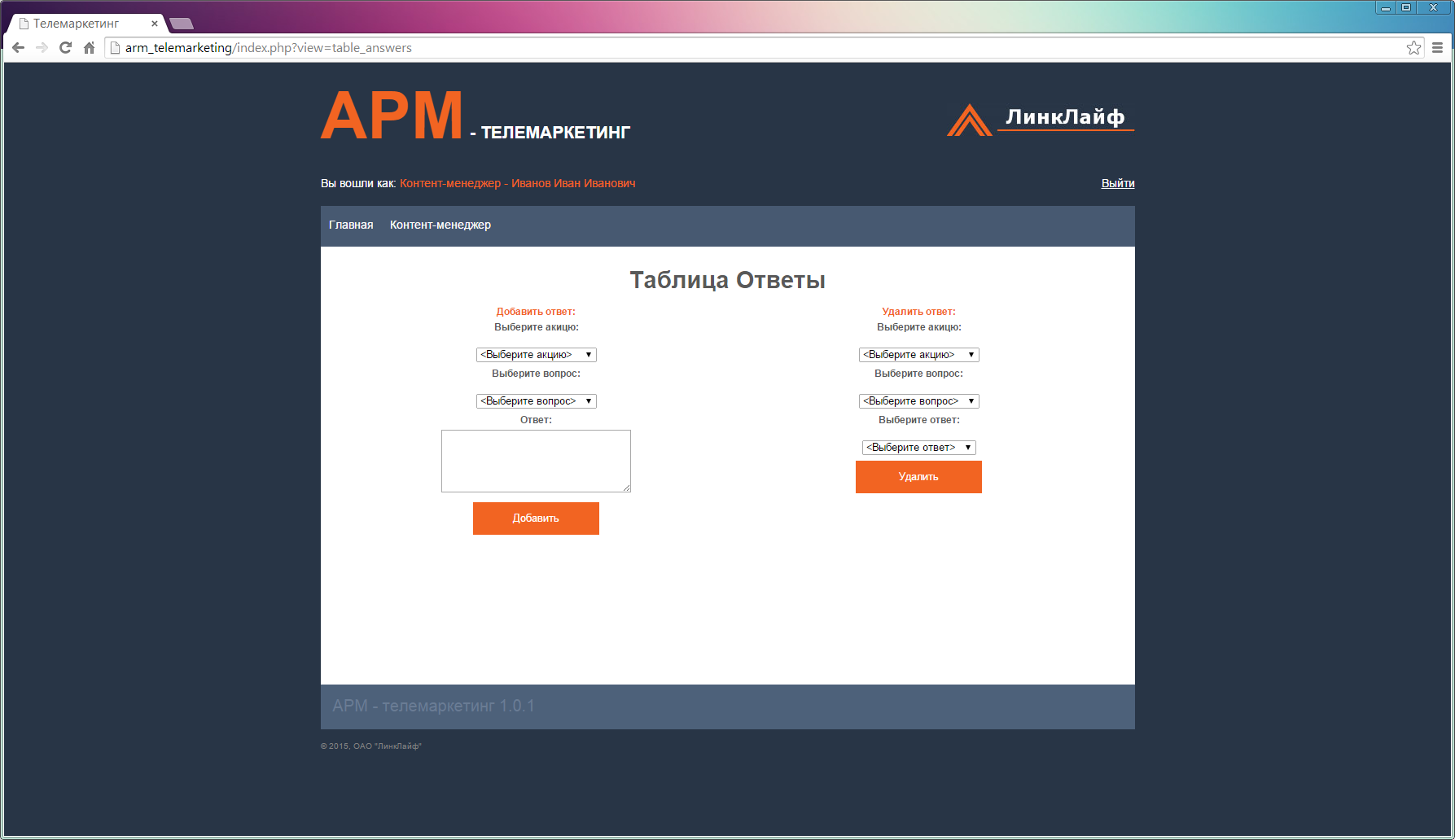


Рисунок 19 – Таблицы «Ответы»

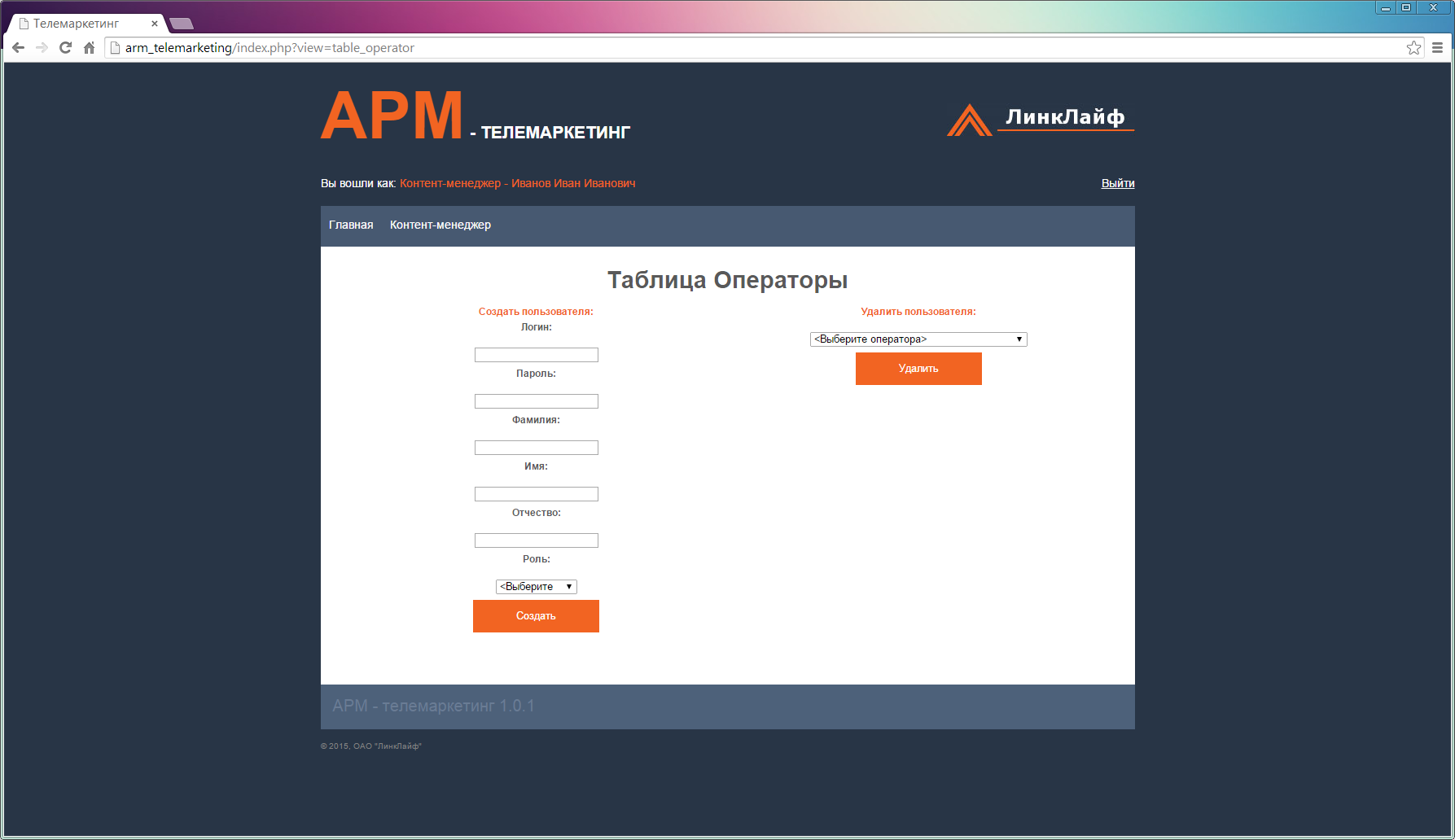


Рисунок 20 – Таблица «Операторы»

**ЗАКЛЮЧЕНИЕ**

В данной работе была создана модель «Системы автоматизации работы оператора центра обработки вызовов службы исходящего телемаркетинга».

Работа над проектом началась с анализа деятельности оператора центра обработки вызовов и подбора технической литературы необходимой для разработки базы данных и для последующего проектирования Web-приложения. В результате для создания Web-приложения были выбраны наиболее распространенные языки программирования HTML, CSS, PHP, JavaScript, а для базы данных наилучшим образом подошла СУБД MySQL. Было сделано описание работы Web-приложения и основных принципов ее функционирования. Согласно заданию и выдвигаемым требованием была создана база данных, обращение к которой происходит по средствам запросов.

Программа рассчитана на пользователей, имеющих необходимый минимум знаний в области работы с персональным компьютером. Для использования приложения не требуется специальное обучение персонала.

Важно отметить, что разработанное Web-приложение способно значительно облегчить труд сотрудникам центра обработки вызовов. Она позволит сократить затраты времени на трудоемкие операции, а также предоставлять информацию целевой аудитории более оперативно и в качесвенно новом виде. Применение данного программного обеспечения приведет к повышению эффективности работы организации за счет:

* перехода на новый уровень взаимодействия с целевой аудиторией;
* ускорения процессов обработки информации;
* высвобождению людских ресурсов.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Таланов В.М. Проектирование информационных систем и баз данных: учебное пособие / Таланов В.М., Федосин С.А. – Саранск: Издательство Мордовского университета, 2001 г. – 72 с.;
2. Буч Гр., Рамбо Дж., Джекобсон А. UML. Руководство пользователя. Издательство: ДМК, 2006 г. – 175 с.;
3. Маглинец Ю.А. Анализ требований к автоматизированным информационным системам: курс лекций. www.intuit.ru
4. Маклаков С.В. BPwin и ERwin. CASE - средства разработки информационных систем / Маклаков С.В. – М.: ДИЛОГ-МИФИ, 2007 г. – 256 с.;
5. Зандстра М. PHP, третье издание / СПб.: Вильямс, 2008 г. – 566 с.;
6. Флэнаган Д. JavaScript. Подробное руководство, пятое издание / СПб.: Символ, 2008 г. – 992 с.;
7. Фленов М.Е. PHP глазами хакера / Фленов М.Е. – СПб: БХВ-Петербург, 2009 г. – 304 с.;
8. Интернет университет информационных технологий: официальный сайт. [Электронный ресурс] // Режим доступа: http://www.intuit.ru

**Приложение А**

(обязательное)

**Листинг программного кода**

Листинг А.1 – Файл index.php

<?

ob\_start();

session\_start();

require\_once('func.php');

$view = empty($\_GET['view']) ? 'start' : $\_GET['view'];

//авторизация

if(isset($\_POST['submit']) && !isset($\_SESSION['loggedin'])){

$login = $\_POST['login'];

$password = $\_POST['password'];

$user = select\_name\_sername($login, $password);

if($user){

$\_SESSION['login'] = $login;

$\_SESSION['loggedin'] = true;

$\_SESSION['FIO'] = $user['sername']." ".$user['name']." ".$user['middlename'];

$\_SESSION['role'] = $user['id\_role'];

$\_SESSION['id\_operator'] = $user['id\_operator'];

$\_SESSION['role\_name'] = select\_role\_name($user['id\_role']);

}

else{

$error\_user = "Неверный логин/пароль!";

}

}

//конец

switch($view){

case 'operator':

$actions = select\_all\_actions();

$lists = select\_all\_lists();

$event\_types = select\_all\_event\_types();

break;

case 'operator\_report':

$operators = select\_all\_operators();

break;

case 'table\_operator':

$operators = select\_all\_users();

$roles = select\_all\_fields('roles');

break;

case 'table\_actions':

$actions = select\_all\_actions();

break;

case 'action\_report':

$actions = select\_all\_actions();

break;

case 'table\_questions':

$actions = select\_all\_actions();

break;

case 'table\_answers':

$actions = select\_all\_actions();

break;

}

require\_once($\_SERVER['DOCUMENT\_ROOT'].'/views/templates/template.php');

?>

Листинг А.2 – Файл func.php

<? define("HOST", "localhost");

define("USER", "root");

define("PASSWORD", "");

define("DB", "telemarket");

function connect\_db(){

$connect = mysql\_connect(HOST, USER, PASSWORD);

if(!$connect){

echo 'Не удалось соединиться с mysql '.mysql\_error();

return false;

}

$select\_db = mysql\_select\_db(DB, $connect);

if(!$select\_db){

echo 'Не удалось выбрать базу данных '.mysql\_error();

return false;

}

mysql\_query( 'SET NAMES "utf8" COLLATE "utf8\_general\_ci"' );

return true;

}

function db\_result\_to\_array($result){

$res\_array = array();

$i = 0;

while($row = mysql\_fetch\_assoc($result)){

$res\_array[$i] = $row;

$i++;

}

return $res\_array;

}

function select\_name\_sername($login, $password){

connect\_db();

$query = "SELECT \* FROM operators WHERE login='$login' AND password='$password'";

$result = mysql\_query($query);

if(!$result){

echo "запрос на выборку не выполнен".mysql\_error();

}

$result = db\_result\_to\_array($result);

if($result)

return $result[0];

else return false;

}

function select\_role\_name($id\_role){

connect\_db();

$query = "SELECT name FROM roles WHERE id\_role=$id\_role";

$result = mysql\_query($query);

if(!$result){

echo "запрос на выборку не выполнен".mysql\_error();

}

$result = db\_result\_to\_array($result);

if($result)

return $result[0]['name'];

else return false;

}

function select\_last\_interview(){

connect\_db();

$query = "SELECT MAX(id\_interview) FROM interview";

$result = mysql\_query($query);

if(!$result){

echo "запрос на выборку не выполнен".mysql\_error();

}

$result = mysql\_fetch\_row($result);

return $result;

}

function select\_all\_actions(){

connect\_db();

$query = "SELECT \* FROM actions WHERE `actions`.`end\_date` > CURDATE()";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

$result = db\_result\_to\_array($result);

return $result;

}

function select\_all\_lists(){

connect\_db();

$query = "SELECT \* FROM lists";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

$result = db\_result\_to\_array($result);

return $result;

}

function select\_action\_info($id\_action){

connect\_db();

$query = "SELECT \* FROM actions WHERE `actions`.`end\_date` > CURDATE() AND `actions`.`id\_action` = $id\_action";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

$result = db\_result\_to\_array($result);

return $result;

}

function select\_abonent\_by\_list($list){

connect\_db();

$query = "SELECT A.\*

FROM abonents AS A

LEFT JOIN events AS E ON ( A.id\_abonent = E.id\_abonent )

WHERE (A.id\_list = $list AND (E.id\_event\_type is NULL OR E.id\_event\_type = 2 OR E.id\_event\_type = 3) AND E.date = CURDATE())

ORDER BY RAND()";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

$result = db\_result\_to\_array($result);

return $result;

}

function select\_all\_event\_types(){

connect\_db();

$query = "SELECT \* FROM event\_types";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

$result = db\_result\_to\_array($result);

return $result;

}

function update\_event($event, $abonent, $id\_operator){

connect\_db();

$query = "UPDATE events

SET events.id\_event\_type = $event,

events.id\_operator = $id\_operator

WHERE events.id\_abonent = $abonent AND events.date = CURDATE()";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

return $result;

}

function add\_operator($login, $password, $id\_role, $name, $sername, $middlename){

connect\_db();

$query = "INSERT INTO operators(login, password, id\_role, name, sername, middlename) VALUES('$login', '$password', $id\_role, '$name', '$sername', '$middlename')";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

return $result;

}

function add\_question($number, $text, $is\_end, $id\_action){

connect\_db();

$query = "INSERT INTO questions(number, text, is\_end, id\_action) VALUES($number, '$text', $is\_end, '$id\_action')";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

return $result;

}

function add\_answer($number, $text, $id\_question){

connect\_db();

$query = "INSERT INTO answers(number, text, id\_question) VALUES($number, '$text', '$id\_question')";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

return $result;

}

function del\_operator($id\_operator){

connect\_db();

$query = "DELETE FROM operators WHERE id\_operator=$id\_operator";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

return $result;

}

function del\_action($id\_action){

connect\_db();

$query = "DELETE FROM actions WHERE id\_action=$id\_action";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

return $result;

}

function del\_question($id\_question){

connect\_db();

$query = "DELETE FROM questions WHERE id\_question=$id\_question";

echo $query;

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

return $result;

}

function del\_answer($id\_answer){

connect\_db();

$query = "DELETE FROM answers WHERE id\_answer=$id\_answer";

echo $query;

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

return $result;

}

function add\_interview($id\_abonent, $id\_operator, $id\_action){

connect\_db();

$query = "INSERT INTO interview(`date`, id\_abonent, id\_operator, id\_action) VALUES(CURDATE(), $id\_abonent, $id\_operator, $id\_action)";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

return $result;

}

function add\_action($name, $description, $start\_date, $end\_date){

connect\_db();

$query = "INSERT INTO actions(name, description, start\_date, end\_date) VALUES('$name', '$description', '$start\_date', '$end\_date')";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

return $result;

}

function add\_answer\_checked($id\_interview, $id\_question, $id\_answer){

connect\_db();

$query = "INSERT INTO answers\_checked(id\_interview, id\_question, id\_answer) VALUES($id\_interview, $id\_question, $id\_answer)";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

return $result;

}

function select\_questions($id\_action){

connect\_db();

$query = "SELECT \* FROM questions

WHERE questions.id\_action = $action";

$result = mysql\_query($query);

$result = db\_result\_to\_array($result);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

return $result;

}

function select\_questions\_by\_action($id\_action){

connect\_db();

$query = "SELECT \* FROM questions

WHERE questions.id\_action = $id\_action";

$result = mysql\_query($query);

$i = 0;

while($row = mysql\_fetch\_row($result)){

$data[$i] = $row;

$i++;

}

return $data;

}

function select\_answers\_by\_question($id\_question){

connect\_db();

$query = "SELECT \* FROM answers

WHERE answers.id\_question = $id\_question";

$result = mysql\_query($query);

$i = 0;

while($row = mysql\_fetch\_row($result)){

$data[$i] = $row;

$i++;

}

return $data;

}

function select\_question\_number($action){

connect\_db();

$query = "SELECT COUNT( questions.id\_question )

FROM questions

WHERE questions.id\_action = $action";

$result = mysql\_query($query);

$row = mysql\_fetch\_row($result);

if(!$row){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

return $row[0];

}

function select\_answer\_number($id\_question){

connect\_db();

$query = "SELECT COUNT( answers.id\_answer )

FROM answers

WHERE answers.id\_question = $id\_question";

$result = mysql\_query($query);

$row = mysql\_fetch\_row($result);

if(!$row){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

return $row[0];

}

function select\_question($action, $number){

connect\_db();

$query = "SELECT \* FROM questions

WHERE questions.id\_action = $action AND questions.number = $number";

$result = mysql\_query($query);

$result = db\_result\_to\_array($result);

if($result)

return $result;

else

return false;

}

function select\_answers($id\_question){

connect\_db();

$query = "SELECT \* FROM answers

WHERE answers.id\_question = $id\_question";

$result = mysql\_query($query);

$i = 0;

while($row = mysql\_fetch\_row($result)){

$data[$i] = $row;

$i++;

}

return $data;

}

function select\_all\_fields($table){

connect\_db();

$query = "SELECT \* FROM $table";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

$result = db\_result\_to\_array($result);

return $result;

}

function select\_all\_operators(){

connect\_db();

$query = "SELECT \* FROM operators WHERE id\_role='3'";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

$result = db\_result\_to\_array($result);

return $result;

}

function select\_all\_users(){

connect\_db();

$query = "SELECT \* FROM operators";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

$result = db\_result\_to\_array($result);

return $result;

}

function select\_interview\_by\_operator($id\_operator, $start\_date, $end\_date){

connect\_db();

$query = "SELECT `interview`.`date`, `lists`.`name` AS list,`abonents`.`name`,`abonents`.`middlename`,`abonents`.`sername`, `actions`.`name` AS action

FROM `interview`

INNER JOIN `abonents` ON `interview`.`id\_abonent`=`abonents`.`id\_abonent`

INNER JOIN `lists` ON `abonents`.`id\_list`=`lists`.`id\_list`

INNER JOIN `actions` ON `interview`.`id\_action`=`actions`.`id\_action`

WHERE `interview`.`id\_operator`='$id\_operator'

AND '$start\_date' <= `interview`.`date`

AND '$end\_date' >= `interview`.`date`

ORDER BY `interview`.`date` ASC";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

$i = 0;

while($row = mysql\_fetch\_row($result)){

$data[$i] = $row;

$i++;

}

return $data;

}

function select\_interview\_by\_action($id\_action, $selection\_action, $start\_date, $end\_date){

connect\_db();

$query = "SELECT `interview`.`date`, `lists`.`name` AS list,`abonents`.`name`,`abonents`.`middlename`,`abonents`.`sername`, `actions`.`name` AS action

FROM `interview`

INNER JOIN `abonents` ON `interview`.`id\_abonent`=`abonents`.`id\_abonent`

INNER JOIN `lists` ON `abonents`.`id\_list`=`lists`.`id\_list`

INNER JOIN `actions` ON `interview`.`id\_action`=`actions`.`id\_action`

INNER JOIN `answers\_checked` ON `answers\_checked`.`id\_interview` = `interview`.`id\_interview`

WHERE `interview`.`id\_action`='$id\_action'

and `answers\_checked`.`id\_answer` = (select `id\_answer` from `answers`

where `text`='$selection\_action'

and `id\_question` = (select `id\_question` from `questions` where `is\_end`='1' and `id\_action`='$id\_action'))

AND '$start\_date' <= `interview`.`date`

AND '$end\_date' >= `interview`.`date`

ORDER BY `interview`.`date` ASC";

$result = mysql\_query($query);

if(!$result){

echo "Запрос не выполнен ".mysql\_error();

return false;

}

$i = 0;

while($row = mysql\_fetch\_row($result)){

$data[$i] = $row;

$i++;

}

return $data;

}

?>

Листинг А.3 – Файл template.php

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8"/>

<link rel="stylesheet" type="text/css" href="styles/style.css">

<link rel="stylesheet" type="text/css" href="styles/menu.css">

<script src="scripts/jquery.js" type="text/javascript"></script>

<script src="scripts/functions.js" type="text/javascript"></script>

<title>Телемаркетинг</title>

</head>

<body>

<div id="main">

<header>

<div id="logo\_title">

<h2><span>АРМ</span> - ТЕЛЕМАРКЕТИНГ</h2>

</div>

<div id="logo\_image">

<img src="images/logo.jpg" alt="logo"/>

</div>

</header>

<div id="content">

<?php if(isset($\_SESSION['loggedin'])){?>

<span id="user">Вы вошли как: <span id="role" class="orange"><?=$\_SESSION['role\_name']?></span><span id="id\_user" style="display: none;"><?=$\_SESSION['id\_operator']?></span><span id="user\_fio" class="orange"> - <?=$\_SESSION['FIO']?></span></span>

<a href="index.php?view=exit\_user" id="exit\_user">Выйти</a>

<?php } ?>

<!--menu begin-->

<div id="main-menu">

<ul id="menu">

<!--для теста-->

<li><a href="index.php">Главная</a></li>

<? if ($\_SESSION['role'] == 3){?>

<li><a href="index.php?view=operator">Оператор</a></li>

<? } ?>

<? if ($\_SESSION['role'] == 2){?>

<li>

<a href="#">Менеджер</a>

<ul>

<li><a href="index.php?view=operator\_report">Отчет по оператору</a></li>

<li><a href="index.php?view=action\_report">Отчет о рейтинге акций</a></li>

</ul>

</li>

<? } ?>

<? if ($\_SESSION['role'] == 1){?>

<li>

<a href="#">Контент-менеджер</a>

<ul>

<li><a href="index.php?view=table\_operator">Таблица "Операторы"</a></li>

<li><a href="index.php?view=table\_actions">Таблица "Акции"</a></li>

<li><a href="index.php?view=table\_questions">Таблица "Вопросы"</a></li>

<li><a href="index.php?view=table\_answers">Таблица "Ответы"</a></li>

</ul>

</li>

<? } ?>

<!-- конец -->

</ul>

</div>

<!--content-block-->

<div id="content-block">

<? require\_once ($\_SERVER['DOCUMENT\_ROOT'].'/views/pages/'.$view.'.php'); ?>

</div>

</div>

<footer>

<div id="footer-top">

<p>АРМ - телемаркетинг 1.0.1</p>

</div>

<div id="footer-bottom">

<p>&copy; 2015, ОАО "ЛинкЛайф"</p>

</div>

</footer>

</div>

</body>

</html>

Листинг А.4 – Файл action\_report.php

<h3>Отчет по рейтингу акций</h3>

<div id="action\_report" class="all\_page">

<form id="action\_report\_sel" action="#" method="GET" class="form\_sel">

<span class="action\_report\_capt">Начало периода: </span>

<input type="date" name="start\_date" id="start\_date">

<span class="action\_report\_capt">Конец периода: </span>

<input type="date" name="end\_date" id="end\_date">

<br>

<span class="action\_report\_capt">Акция: </span>

<select name="list\_action" id="list\_action">

<option value="-1" selected disabled>&lt;Выберите акцию&gt;</option>

<?php foreach ($actions as $field){?>

<option value="<?=$field['id\_action']?>"><?=$field['name']?></option>

<?php } ?>

</select>

<span class="action\_report\_capt">Согласны на подключение: </span>

<select name="selection\_action" id="selection\_action">

<option value="-1" selected disabled>&lt;Выберите значение&gt;</option>

<option value="Да">Да</option>

<option value="Нет">Нет</option>

</select>

<input type="button" name="button" id="sform\_action" value="Сформировать">

</form>

<center><span class="action\_report\_capt"><b>Результаты: </b></span></center>

<span class="action\_report\_capt">Всего результатов: <span id="res\_num">0</span></span>

<table id="result\_action\_report">

</table>

<br>

<input type="button" name="button" id="sform\_clear" value="Очистить результаты">

</div>

Листинг А.5 – Файл exit\_user.php

<?

if(isset($\_SESSION['loggedin'])){

$\_SESSION = array();

}

header("Location: index.php");

?>

Листинг А.6 – Файл operator.php

<h3>Страница оператора</h3>

<div id="info" class="all\_page">

<form action="#" method="get" id="call\_form">

<div id="select\_abon">

<span class="action\_capt">Акция: </span>

<select name="actions" id="actions">

<option value="-1" selected disabled>&lt;Выберите акцию&gt;</option>

<?php foreach ($actions as $field){?>

<option value="<?=$field['id\_action']?>"><?=$field['name']?></option>

<?php } ?>

</select>

<span class="list\_capt">Список: </span>

<select name="lists" id="lists">

<option value="-1" selected disabled>&lt;Выберите список&gt;</option>

<?php foreach ($lists as $field){?>

<option value="<?=$field['id\_list']?>"><?=$field['name']?></option>

<?php } ?>

</select>

<input type="button" id="search\_abon" value="Найти абонента">

<br>

</div>

<br>

<div id="action\_info">

<p><b>Информация по акции:</b></p>

<p> </p>

</div>

<br>

<div id="search\_result">

<p><b>Информация по абоненту:</b></p>

<p>Абонент: </p>

<p>Адрес: </p>

<p>Телефон: </p>

</div>

<br>

<div id="ready\_select">

<span class="action\_capt">Готовность к разговору: </span>

<select name="list\_call" id="list\_call">

<option value="-1" selected disabled>&lt;Выберите список&gt;</option>

<?php foreach ($event\_types as $field){?>

<option value="<?=$field['id\_event\_type']?>"><?=$field['name']?></option>

<?php } ?>

</select>

<input type="button" id="write" value="Записать">

</div>

<br>

<div id="action\_questions">

<span class="action\_capt question\_text">Вопрос 1: Текст вопроса 1</span>

<br>

<div id="action\_question">

<input type="radio" name="question\_1" value="1"> Ответ 1<Br>

</div>

<input type="button" id="save" name="save" value="Сохранить результаты">

</div>

</form>

</div>

Листинг А.7 – Файл operator\_report.php

<h3>Отчет по оператору</h3>

<div id="operator\_report" class="all\_page">

<form id="operator\_report\_sel" action="#" method="GET" class="form\_sel">

<span class="operator\_report\_capt">Начало периода: </span>

<input type="date" name="start\_date" id="start\_date">

<span class="operator\_report\_capt">Конец периода: </span>

<input type="date" name="end\_date" id="end\_date">

<br>

<span class="operator\_report\_capt">Оператор: </span>

<select name="list\_operator" id="list\_operator">

<option value="-1" selected disabled>&lt;Выберите оператора&gt;</option>

<?php foreach ($operators as $field){?>

<option value="<?=$field['id\_operator']?>"><?=$field['sername']." ".$field['name']." ".$field['middlename']?></option>

<?php } ?>

</select>

<input type="button" name="button" id="sform" value="Сформировать">

</form>

<center><span class="operator\_report\_capt"><b>Результаты: </b></span></center>

<span class="operator\_report\_capt">Всего результатов: <span id="res\_num">0</span></span>

<table id="result\_operator\_report">

</table>

<br>

<input type="button" name="button" id="sform\_clear" value="Очистить результаты">

</div>

Листинг А.8 – Файл start.php

<p><b>"АРМ-Телемаркетинг"</b><br>- это универсальный программный продукт, воплощающий в себе автоматизированное рабочее место оператора Центра обработки вызовов службы исходящего телемаркетинга.</p>

<p>Для работы с данным электронным ресурсом необходимо дождаться назначения Вам определенных полномочий и выдачи учетных данных.</p>

<p>Если Вы забыли Ваши учетные данные, обратитесь к администратору для их восстановления.</p>

<? if(!isset($\_SESSION['loggedin'])){ ?>

<form id="aut\_form" action="index.php" method="POST">

<p class="aut\_form\_caption">Логин:</p>

<input type="text" class="aut\_input" name="login">

<p class="aut\_form\_caption">Пароль:</p>

<input type="password" class="aut\_input" name="password">

<input type="submit" name="submit" value="Войти">

</form>

<?=$error\_user?>

<? }?>

Листинг А.9 – Файл table\_actions.php

<h3>Таблица Акции<h3>

<div id="table\_aactions" class="all\_page">

<div class="left\_column">

<p class="ac\_caption main\_capt">Создать акцию:</p>

<p class="ac\_caption">Название: </p><input type="text" name="ac\_name" id="ac\_name">

<p class="ac\_caption">Описание: </p><textarea rows="10" cols="30" name="ac\_description" id="ac\_description"></textarea>

<p class="ac\_caption">Дата начала: </p><input type="date" name="ac\_start\_date" id="ac\_start\_date">

<p class="ac\_caption">Дата окончания: </p><input type="date" name="ac\_end\_date" id="ac\_end\_date">

<p class="ac\_caption"><input type="button" name="ac\_create" id="ac\_create" value="Создать"></p>

</div>

<div class="right\_column">

<p class="ac\_caption main\_capt">Удалить акицю:</p>

<select name="ac\_list\_delete" id="ac\_list\_delete">

<option value="-1" selected disabled>&lt;Выберите акцию&gt;</option>

<?php foreach ($actions as $field){?>

<option value="<?=$field['id\_action']?>"><?=$field['name']?></option>

<?php } ?>

</select>

<p class="ac\_caption"><input type="button" name="ac\_delete" id="ac\_delete" value="Удалить"></p>

</div>

</div>

Листинг А.10 – Файл table\_answers.php

<h3>Таблица Ответы<h3>

<div id="table\_answers" class="all\_page">

<div class="left\_column">

<p class="ans\_caption main\_capt">Добавить ответ:</p>

<p class="ans\_caption">Выберите акицю:</p>

<select name="ans\_list\_action\_add" id="ans\_list\_action\_add">

<option value="-1" selected disabled>&lt;Выберите акцию&gt;</option>

<?php foreach ($actions as $field){?>

<option value="<?=$field['id\_action']?>"><?=$field['name']?></option>

<?php } ?>

</select>

<p class="ans\_caption">Выберите вопрос:</p>

<select name="ans\_list\_ques\_add" id="ans\_list\_ques\_add">

<option value="-1" selected disabled>&lt;Выберите вопрос&gt;</option>

</select>

<p class="ans\_caption">Ответ: </p><textarea rows="5" cols="30" name="ans\_text" id="ans\_text"></textarea>

<p class="ans\_caption"><input type="button" name="ans\_create" id="ans\_create" value="Добавить"></p>

</div>

<div class="right\_column">

<!--del-->

<p class="ac\_caption main\_capt">Удалить ответ:</p>

<p class="ans\_caption">Выберите акицю:</p>

<select name="ans\_list\_action\_del" id="ans\_list\_action\_del">

<option value="-1" selected disabled>&lt;Выберите акцию&gt;</option>

<?php foreach ($actions as $field){?>

<option value="<?=$field['id\_action']?>"><?=$field['name']?></option>

<?php } ?>

</select>

<p class="ans\_caption">Выберите вопрос:</p>

<select name="ans\_list\_ques\_del" id="ans\_list\_ques\_del">

<option value="-1" selected disabled>&lt;Выберите вопрос&gt;</option>

</select>

<p class="ans\_caption">Выберите ответ:</p>

<select name="ans\_list\_del" id="ans\_list\_del">

<option value="-1" selected disabled>&lt;Выберите ответ&gt;</option>

</select>

<p class="ans\_caption"><input type="button" name="ans\_delete" id="ans\_delete" value="Удалить"></p>

</div>

</div>

Листинг А.11 – Файл table\_operator.php

<h3>Таблица Операторы<h3>

<div id="table\_operators" class="all\_page">

<div class="left\_column">

<p class="op\_caption main\_capt">Создать пользователя:</p>

<p class="op\_caption">Логин: </p><input type="text" name="op\_login" id="op\_login">

<p class="op\_caption">Пароль: </p><input type="text" name="op\_password" id="op\_password">

<p class="op\_caption">Фамилия: </p><input type="text" name="op\_sername" id="op\_sername">

<p class="op\_caption">Имя: </p><input type="text" name="op\_name" id="op\_name">

<p class="op\_caption">Отчество: </p><input type="text" name="op\_middlename" id="op\_middlename">

<p class="op\_caption">Роль:</p><select name="op\_role" id="op\_role">

<option value="-1" selected disabled>&lt;Выберите роль&gt;</option>

<?php foreach ($roles as $field){?>

<option value="<?=$field['id\_role']?>"><?=$field['name']?></option>

<?php } ?>

</select>

<p class="op\_caption"><input type="button" name="op\_create" id="op\_create" value="Создать"></p>

</div>

<div class="right\_column">

<p class="op\_caption main\_capt">Удалить пользователя:</p>

<select name="op\_list\_delete" id="op\_list\_delete">

<option value="-1" selected disabled>&lt;Выберите оператора&gt;</option>

<?php foreach ($operators as $field){?>

<option value="<?=$field['id\_operator']?>"><?=$field['name']." ".$field['middlename']." ".$field['sername']?></option>

<?php } ?>

</select>

<p class="op\_caption"><input type="button" name="op\_delete" id="op\_delete" value="Удалить"></p>

</div>

</div>

Листинг А.12 – Файл table\_questions.php

<h3>Таблица Вопросы<h3>

<div id="table\_questions" class="all\_page">

<div class="left\_column">

<p class="ques\_caption main\_capt">Добавить вопрос:</p>

<p class="ques\_caption">Выберите акицю:</p>

<select name="ques\_list\_action\_add" id="ques\_list\_action\_add">

<option value="-1" selected disabled>&lt;Выберите акцию&gt;</option>

<?php foreach ($actions as $field){?>

<option value="<?=$field['id\_action']?>"><?=$field['name']?></option>

<?php } ?>

</select>

<p class="ques\_caption">Вопрос: </p><textarea rows="10" cols="30" name="ques\_text" id="ques\_text"></textarea>

<p class="ques\_caption">Последний: </p> <input type="checkbox" name="ques\_last" id="ques\_last" value="1">

<p class="ques\_caption"><input type="button" name="ques\_create" id="ques\_create" value="Добавить"></p>

</div>

<div class="right\_column">

<p class="ac\_caption main\_capt">Удалить вопрос:</p>

<p class="ques\_caption">Выберите акицю:</p>

<select name="ques\_list\_action\_del" id="ques\_list\_action\_del">

<option value="-1" selected disabled>&lt;Выберите акцию&gt;</option>

<?php foreach ($actions as $field){?>

<option value="<?=$field['id\_action']?>"><?=$field['name']?></option>

<?php } ?>

</select>

<p class="ques\_caption">Выберите вопрос:</p>

<select name="ques\_list\_del" id="ques\_list\_del">

<option value="-1" selected disabled>&lt;Выберите вопрос&gt;</option>

</select>

<p class="ques\_caption"><input type="button" name="ques\_delete" id="ques\_delete" value="Удалить"></p>

</div>

</div>

Листинг А.13 – Файл ajaxAddAnswer.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

$number = $\_GET['number'];

$text = $\_GET['text'];

$id\_question = $\_GET['id\_question'];

add\_answer($number, $text, $id\_question);

?>

Листинг А.14 – Файл ajaxAddAnswerChecked.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

$id\_interview = $\_GET['id\_interview'];

$id\_question = $\_GET['id\_question'];

$id\_answer = $\_GET['id\_answer'];

$result = add\_answer\_checked($id\_interview, $id\_question, $id\_answer);

?>

Листинг А.15 – Файл ajaxAddInterview.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

$id\_action = $\_GET['id\_action'];

$id\_abonent = $\_GET['id\_abonent'];

$id\_operator = $\_GET['id\_operator'];

$result = add\_interview($id\_abonent, $id\_operator, $id\_action);

$id\_interview = select\_last\_interview();

echo "$id\_interview[0]";

?>

Листинг А.16 – Файл ajaxAddQuestion.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

$number = $\_GET['number'];

$text = $\_GET['text'];

$is\_end = $\_GET['is\_end'];

$id\_action = $\_GET['id\_action'];

add\_question($number, $text, $is\_end, $id\_action);

?>

Листинг А.17 – Файл ajaxCreateAction.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

$name = $\_GET['name'];

$description = $\_GET['description'];

$start\_date = $\_GET['start\_date'];

$end\_date = $\_GET['end\_date'];

add\_action($name, $description, $start\_date, $end\_date);

?>

Листинг А.18 – Файл ajaxCreateOperator.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

$login = $\_GET['login'];

$password = $\_GET['password'];

$name = $\_GET['name'];

$sername = $\_GET['sername'];

$middlename = $\_GET['middlename'];

$role = $\_GET['role'];

add\_operator($login, $password, $role, $name, $sername, $middlename);

?>

Листинг А.19 – Файл ajaxDeleteAction.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

if(isset($\_GET['id\_action'])){

$id\_action = $\_GET['id\_action'];

del\_action($id\_action);

}

?>

Листинг А.20 – Файл ajaxDeleteAnswer.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

if(isset($\_GET['id\_answer'])){

$id\_answer = $\_GET['id\_answer'];

del\_answer($id\_answer);

}

?>

Листинг А.21 – Файл ajaxDeleteOperator.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

if($\_GET['id\_operator']){

$id\_operator = $\_GET['id\_operator'];

del\_operator($id\_operator);

}

?>

Листинг А.22 – Файл ajaxDeleteQuestion.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

if(isset($\_GET['id\_question'])){

$id\_question = $\_GET['id\_question'];

del\_question($id\_question);

}

?>

Листинг А.23 – Файл ajaxFindAbonents.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

$lists = $\_GET['lists'];

$abonents = select\_abonent\_by\_list($lists);

if($abonents)

echo json\_encode($abonents[0]);

else

echo 'no';

?>

Листинг А.24 – Файл ajaxFindActionInfo.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

$id\_action = $\_GET['id\_action'];

$action = select\_action\_info($id\_action);

if($action)

echo json\_encode($action[0]);

else

echo 'no';

?>

Листинг А.25 – Файл ajaxSelectAnswers.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

$id\_question = $\_GET['id\_question'];

$answers = select\_answers($id\_question);

echo json\_encode($answers);

?>

Листинг А.26 – Файл ajaxSelectAnswersByQuestion.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

$id\_question = $\_GET['id\_question'];

$answers = select\_answers\_by\_question($id\_question);

echo json\_encode($answers);

?>

Листинг А.27 – Файл ajaxSelectNumberLastAnswer.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

if(isset($\_GET['id\_question'])){

$id\_question = $\_GET['id\_question'];

$result = select\_answer\_number($id\_question);

if($result){

echo $result;

}

else{

echo 0;

}

}

?>

Листинг А.28 – Файл ajaxSelectNumberLastQuestion.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

if(isset($\_GET['id\_action'])){

$id\_action = $\_GET['id\_action'];

$result = select\_question\_number($id\_action);

if($result){

echo $result;

}

else{

echo 0;

}

}

?>

Листинг А.29 – Файл ajaxSelectQuestions.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

$id\_action = $\_GET['action'];

$question\_number = $\_GET['question\_number'];

$question = select\_question($id\_action, $question\_number);

echo json\_encode($question[0]);

?>

Листинг А.30 – Файл ajaxSelectQuestionsByAction.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

$id\_action = $\_GET['id\_action'];

$questions = select\_questions\_by\_action($id\_action);

echo json\_encode($questions);

?>

Листинг А.31 – Файл ajaxShowActionReport.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

if($\_GET['list\_action']){

$id\_action = $\_GET['list\_action'];

$start\_date = $\_GET['start\_date'];

$end\_date = $\_GET['end\_date'];

$selection\_action = $\_GET['selection\_action'];

$interview = select\_interview\_by\_action($id\_action, $selection\_action, $start\_date, $end\_date);

if($interview)

echo json\_encode($interview);

else echo 'no';

}

?>

Листинг А.32 – Файл ajaxShowOperatorReport.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

if($\_GET['list\_operator']){

$id\_operator = $\_GET['list\_operator'];

$start\_date = $\_GET['start\_date'];

$end\_date = $\_GET['end\_date'];

$interview = select\_interview\_by\_operator($id\_operator, $start\_date, $end\_date);

if($interview)

echo json\_encode($interview);

else echo 'no';

}

?>

Листинг А.33 – Файл ajaxUpdateEvents.php

<? require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/func.php';

$id\_event\_type = $\_GET['list\_call'];

$id\_abonent = $\_GET['id\_abonent'];

$id\_operator = $\_GET['id\_operator'];

$result = update\_event($id\_event\_type, $id\_abonent, $id\_operator);

if($result){

echo "Данные обновлены!";

}else{

echo "Данные не обновлены!";

}

?>

Листинг А.34 – Файл function.js

$(document).ready(function(){

var id\_abonent;

//загрузка текста акции

$('#actions').change(function(){

var action = $('#actions').val();

$.get('files/ajaxFindActionInfo.php', {'id\_action':action}, function(data){

var object = JSON.parse(data);

id\_abonent = object.id\_abonent;

$('#action\_info').html("<p><b>Информация по акции:</b></p><p>"+object.description+"</p>");

$('#action\_info').css("display","block");

});

});

//поиск абонента

$('#search\_abon').click(function(){

var lists = $('#lists').val();

var action = $('#actions').val();

if (lists == null || action == null){

alert('Не все поля заполнены!');

}else{

$.get('files/ajaxFindAbonents.php', {'lists':lists}, function(data){

//проверка на наличие абонентов для опроса

if (data == 'no'){

alert('В выбранном списке абонентов для анкетирования нет.');

}

var object = JSON.parse(data);

id\_abonent = object.id\_abonent;

$('#search\_result').html("<p><b>Информация по абоненту:</b></p><p>ФИО: "+object.sername+" "+object.name+" "+object.middlename+"</p><p>Адрес: "+object.address+"</p><p>Телефон: "+object.phone+"</p>");

$('#search\_result').css("display","block");

$('#ready\_select').css("display","block");

$('#action\_questions').css("display","none");

});

}

});

//вывод вопросов

var question\_number = 1;

var id\_question;

var id\_interview = 0;

$('#write').click(function(){

var action = $('#actions').val();

var list\_call = $('#list\_call').val();

var id\_user = $('#id\_user').html();

//обновить значение в табоице events

$.get('files/ajaxUpdateEvents.php',{'list\_call':list\_call, 'id\_abonent':id\_abonent, 'id\_operator':id\_user},function(data){

alert(data);

if((list\_call == 2) || (list\_call == 3) || (list\_call == 4)){ location.reload(); }

})

////end-обновить значение в таблице events

if(list\_call == 1){

//добавляем новое интервью

$.get('files/ajaxAddInterview.php',{'id\_abonent':id\_abonent, 'id\_operator':id\_user, 'id\_action':action},function(data){

//alert(data);

//получаем id созданного интервью

id\_interview = data;

});

//end-добавляем новое интервью

$.get('files/ajaxSelectQuestions.php',{'action':action, 'question\_number':question\_number},function(data){

var object = JSON.parse(data);

$('.question\_text').html("Вопрос " + question\_number + ": " + object.text);

id\_question = object.id\_question;

$.get('files/ajaxSelectAnswers.php',{'id\_question':id\_question},function(data){

var object = JSON.parse(data);

//var questions = $('#action\_question').html();

var questions = "";

for(i = 0; i < object.length; i++){

questions += "<input type='radio' name='question\_"+question\_number+"' value='"+object[i][0]+"'> "+object[i][2]+"<Br>";

}

$('#action\_question').html(questions);

$('#action\_questions').css("display","block");

});

});

}

});

//следующий вопрос

var end = 0;

$('#save').click(function(){

//записываем ответы

id\_answer = $('input:radio:checked').val();

$.get('files/ajaxAddAnswerChecked.php',{'id\_interview':id\_interview, 'id\_answer':id\_answer, 'id\_question':id\_question} ,function(data){

});

//end-записываем ответы

if(end == 0){//если вопрос не последний

var questions = "";

$('#action\_question').html("<br>");

question\_number++;

var list\_call = $('#list\_call').val();

var action = $('#actions').val();

if(list\_call == 1){

$.get('files/ajaxSelectQuestions.php',{'action':action, 'question\_number':question\_number},function(data){

var object = JSON.parse(data);

$('.question\_text').html("Вопрос " + question\_number + ": " + object.text);

id\_question = object.id\_question; $.get('files/ajaxSelectAnswers.php',{'id\_question':id\_question},function(data){

var object = JSON.parse(data);

for(i = 0; i < object.length; i++){

questions += "<input type='radio' name='question\_"+question\_number+"' value='"+object[i][0]+"'> "+object[i][2]+"<Br>";

}

$('#action\_question').html(questions);

$('#action\_questions').css("display","block");

});

if(object.is\_end == 1){ //если вопрос последний

end = 1;

}

});

}

}else{//если вопрос последний

alert('Анкетирование завершено.');

$('#action\_questions').css("display","none");

question\_number = 1;

end = 0;

location.reload();

}

});//end save click

//вывод отчета по оператору

$('#sform').click(function(){

list\_operator = $('#list\_operator').val();

start\_date = $('#start\_date').val();

end\_date = $('#end\_date').val();

if (list\_operator == null || start\_date == '' || end\_date == ''){

alert('Не все поля, требуемые для формирования отчета, заполнены!');

}else{

$.get('files/ajaxShowOperatorReport.php',{'list\_operator':list\_operator, 'start\_date':start\_date, 'end\_date':end\_date }, function(data){

if(data == 'no'){

var msg = "Результаты не найдены";

$('#result\_operator\_report').html(msg);

$('#res\_num').html(object.length);

}else{

var object = JSON.parse(data);

var table = "<tr><th>Дата</th><th>Список</th><th>Абонент</th><th>Акция</th><tr>";

for(i = 0; i < object.length; i++){

table += "<tr><td>"+object[i][0]+"</td><td>"+object[i][1]+"</td><td>"+object[i][4]+" "+object[i][2]+" "+object[i][3]+"</td><td>"+object[i][5]+"</td></tr>";

}

$('#result\_operator\_report').html(table);

$('#res\_num').html(object.length);

}

});

}

});

//очищение результатов отчетов

$('#sform\_clear').click(function(){

location.reload();

});

//вывод отчета по акции

$('#sform\_action').click(function(){

list\_action = $('#list\_action').val();

start\_date = $('#start\_date').val();

end\_date = $('#end\_date').val();

selection\_action = $('#selection\_action').val();

if (list\_action == null || start\_date == '' || end\_date == '' || selection\_action == null){

alert('Не все поля, требуемые для формирования отчета, заполнены!');

}else{

$.get('files/ajaxShowActionReport.php',{'list\_action':list\_action, 'selection\_action':selection\_action, 'start\_date':start\_date, 'end\_date':end\_date }, function(data){

if(data == 'no'){

var msg = "Результаты не найдены";

$('#result\_action\_report').html(msg);

$('#res\_num').html(object.length);

}else{

var object = JSON.parse(data);

var table = "<tr><th>Дата</th><th>Список</th><th>Абонент</th><th>Акция</th><tr>";

for(i = 0; i < object.length; i++){

table += "<tr><td>"+object[i][0]+"</td><td>"+object[i][1]+"</td><td>"+object[i][4]+" "+object[i][2]+" "+object[i][4]+"</td><td>"+object[i][5]+"</td></tr>";

}

$('#result\_action\_report').html(table);

$('#res\_num').html(object.length);

}

});

}

});

//создать пользователя

$('#op\_create').click(function(){

login = $('#op\_login').val();

password = $('#op\_password').val();

sername = $('#op\_sername').val();

name = $('#op\_name').val();

middlename = $('#op\_middlename').val();

role = $('#op\_role').val();

$.get('files/ajaxCreateOperator.php', {'login':login, 'password':password, 'sername':sername, 'name':name, 'middlename':middlename, 'role':role}, function(data){

alert('Пользоваель добавлен!');

location.reload();

});

});

//удалить пользователя

$('#op\_delete').click(function(){

id\_operator = $('#op\_list\_delete').val();

$.get('files/ajaxDeleteOperator.php', {'id\_operator':id\_operator}, function(data){

alert('Пользоваель удален!');

location.reload();

});

});

//добавить акцию

$('#ac\_create').click(function(){

name = $('#ac\_name').val();

description = $('#ac\_description').val();

start\_date = $('#ac\_start\_date').val();

end\_date = $('#ac\_end\_date').val();

$.get('files/ajaxCreateAction.php', {'name':name, 'description':description, 'start\_date':start\_date, 'end\_date':end\_date}, function(data){

alert('Акция добавлена!');

location.reload();

});

});

//удалить акцию

$('#ac\_delete').click(function(){

id\_action = $('#ac\_list\_delete').val();

$.get('files/ajaxDeleteAction.php', {'id\_action':id\_action}, function(data){

alert('Акция удалена!');

location.reload();

});

});

//добавить вопрос

var number;

$('#ques\_create').click(function(){

text = $('#ques\_text').val();

is\_end = $('input:checked').val();

if(is\_end != 1)

is\_end = 0;

id\_action = $('#ques\_list\_action\_add').val();

//number = 0;

//получаем номер для нового вопроса

$.get('files/ajaxSelectNumberLastQuestion.php', {'id\_action':id\_action}, function(data){

number = data;

number++;

//добавляем новый вопрос

$.get('files/ajaxAddQuestion.php', {'number':number, 'text':text, 'is\_end':is\_end, 'id\_action':id\_action}, function(data){

alert("Вопрос добавлен!");

location.reload();

});

});

});

//удалить вопрос

//заполняем select вопросами из выбранной акции

$('#ques\_list\_action\_del').change(function(){

id\_action = $('#ques\_list\_action\_del').val();

$.get('files/ajaxSelectQuestionsByAction.php', {'id\_action':id\_action}, function(data){

content = "<option value='-1' selected disabled>&lt;Выберите вопрос&gt;</option>";

var object = JSON.parse(data);

for(i = 0; i < object.length; i++){

content+= "<option value="+object[i][0]+">"+object[i][2]+"</option>";

}

$('#ques\_list\_del').html(content);

});

});

//удалить выбранный вопрос

$('#ques\_delete').click(function(){

id\_question = $('#ques\_list\_del').val();

$.get('files/ajaxDeleteQuestion.php', {'id\_question':id\_question}, function(data){

alert('Вопрос удален!');

location.reload();

});

});

//---\* добавить/удалить вопрос \*---//

//---\*добавить вопрос\*---//

//заполняем select вопросами по выбранной акции

$('#ans\_list\_action\_add').change(function(){

id\_action = $('#ans\_list\_action\_add').val();

$.get('files/ajaxSelectQuestionsByAction.php', {'id\_action':id\_action}, function(data){

content = "<option value='-1' selected disabled>&lt;Выберите вопрос&gt;</option>";

var object = JSON.parse(data);

for(i = 0; i < object.length; i++){

content+= "<option value="+object[i][0]+">"+object[i][2]+"</option>";

}

$('#ans\_list\_ques\_add').html(content);

});

});

//добавляем вопрос по нажатию кнопки

var number\_answer;

$('#ans\_create').click(function(){

text = $('#ans\_text').val();

id\_question = $('#ans\_list\_ques\_add').val();

//получаем номер для нового ответа

$.get('files/ajaxSelectNumberLastAnswer.php', {'id\_question':id\_question}, function(data){

number\_answer = data;

number\_answer++;

//добавляем новый вопрос

$.get('files/ajaxAddAnswer.php', {'number':number\_answer, 'text':text, 'id\_question':id\_question}, function(data){

alert("Ответ добавлен!");

location.reload();

});

});

});

//---\*удалить вопрос\*---//

//заполняем селекты данными в зависимости от выбранной акции

$('#ans\_list\_action\_del').change(function(){

id\_action = $('#ans\_list\_action\_del').val();

$.get('files/ajaxSelectQuestionsByAction.php', {'id\_action':id\_action}, function(data){

content = "<option value='-1' selected disabled>&lt;Выберите вопрос&gt;</option>";

var object = JSON.parse(data);

for(i = 0; i < object.length; i++){

content+= "<option value="+object[i][0]+">"+object[i][2]+"</option>";

}

$('#ans\_list\_ques\_del').html(content);

});

});

//заполняем селекты данными в зависимости от выбранного вопроса

$('#ans\_list\_ques\_del').change(function(){

id\_question = $('#ans\_list\_ques\_del').val();

$.get('files/ajaxSelectAnswersByQuestion.php', {'id\_question':id\_question}, function(data){

content = "<option value='-1' selected disabled>&lt;Выберите ответ&gt;</option>";

var object = JSON.parse(data);

for(i = 0; i < object.length; i++){

content+= "<option value="+object[i][0]+">"+object[i][2]+"</option>";

}

$('#ans\_list\_del').html(content);

});

});

//удаляем вопрос по нажатию на кнопку

$('#ans\_delete').click(function(){

id\_answer = $('#ans\_list\_del').val();

$.get('files/ajaxDeleteAnswer.php', {'id\_answer':id\_answer}, function(data){

alert('Ответ удален!');

location.reload();

});

});

});

Листинг А.35 – Файл style.css

\*{

margin: 0px;

padding: 0px;

}

body{

font: 14px Arial,sans-serif, Tahoma;

background-color: #283546;

}

#main{

width: 1000px;

margin: 0 auto;

}

header{

margin: 20px 0 20px 0;

height: 100px;

}

#logo\_title{

color: #fff;

width: 400px;

float: left;

}

#logo\_title h2 span{

color: #f26422;

font-size: 80px;

}

#logo\_image{

float: right;

margin-top: 30px;

}

.orange{

color: #f26422;

}

#content-block{

background-color: #fff;

color: #595959;

font-size: 25px;

text-align: center;

/\* height: 450px;\*/

padding: 25px 30px 30px 30px;

}

#footer-top{

color: #6a7c94;

background-color: #4d617a;

font-size: 20px;

height: 40px;

padding: 15px 0 0 15px;

}

#footer-bottom{

color: #828282;

font-size: 10px;

height: 40px;

padding: 15px 0 0 0;

}

#aut\_form{

border-radius: 5px;

background-color: #4d617a;

width: 200px;

margin: 0 auto;

padding-bottom: 10px;

}

.aut\_input{

width: 170px;

}

.aut\_form\_caption{

color: #fff;

font-size: 12px;

line-height: 1.5;

}

#info{

text-align: left;

font-size: 17px;

margin-left: 50px;

margin-top: 30px;

}

#info select, #operator\_report select, #action\_report select{

margin-right: 20px;

background-color: #c8c6c6;

border: none;

font-size: 16px;

}

#select\_abon{

margin: 10px 0 10px 0;

}

#search\_abon, #write, #save, #sform, #sform\_action, #sform\_clear, #op\_create, #op\_delete, #ac\_create, #ac\_delete, #ques\_create, #ques\_delete, #ans\_create, #ans\_delete{

background-color: #f26422;

color: #fff;

height: 40px;

width: 155px;

border: none;

margin: 0 0 20px 0;

cursor: pointer;

}

#search\_abon, #write{

float: right;

}

#sform, #sform\_action, #sform\_clear{

float: right;

}

.action\_question{

margin: 10px 0 10px 0;

}

.action\_question > input{

margin-left: 20px;

}

#operator\_report, #action\_report{

text-align: left;

font-size: 17px;

margin-top: 20px;

}

#result\_operator\_report tr, th, td{

border: 1px solid #828282;

}

#result\_action\_report tr, th, td{

border: 1px solid #828282;

}

#result\_action\_report, #result\_operator\_report{

border-collapse: collapse;

width: 940px;

margin-top: 20px

}

#list\_operator, #list\_action{

margin: 20px 0 20px 0;

}

#action\_info, #search\_result, #ready\_select, #action\_questions{

display: none;

}

#user{

color: #fff;

margin-bottom: 100px;

}

.op\_caption, .ac\_caption, .ques\_caption, .ans\_caption{

font-size: 12px;

margin: 5px 0 5px 0;

}

#op\_create{

margin-bottom: 20px;

}

#op\_role{

width: 100px;

}

.main\_capt{

color: #f26422;

}

#exit\_user{

color: #fff;

float: right;

}

.all\_page{

min-height: 450px;

}

.left\_column{

float: left;

width: 50%;

margin-top: 10px;

/\*margin: 5px 1px 5px 0px;\*/

}

.right\_column{

width: 50%;

margin-top: 10px;

/\* margin: 5px 0px 5px 1px; \*/

display: inline-block;

}

.form\_sel{

margin-bottom: 50px;

}

Листинг А.36 – Файл menu.css

/\*-------menu begin-------------\*/

#main-menu{

height: 50px;

background-color: #475971;

margin-top: 20px;

}

#menu,

#menu ul {

list-style: none;

}

#menu {

float: left;

height: 50px;

}

#menu > li {

float: left;

}

#menu li a {

display: block;

height: 35px;

padding: 15px 10px 0 10px;

text-decoration: none;

}

#menu ul {

position: absolute;

display: none;

z-index: 999;

}

#menu ul li a {

width: 170px;

border-bottom: 1px solid #fff;

}

#menu li:hover ul {

display: block;

}

/\* Main menu ------------------------------------------\*/

#menu {

font-size: 14px;

}

#menu > li > a {

color: #fff;

}

#menu > li:hover > a {

background: #795b59;

color: #fff;

}

/\* Submenu ------------------------------------------\*/

#menu ul {

background: #475971;

}

#menu ul li a {

color: #fff;

}

#menu ul li:hover a {

background: #795b59;

}

/\*-------menu end-------------\*/

**Приложение Б**

(обязательное)

**Графический материал**

1. Диаграмма прецедентов системы
2. Диаграмма последовательности
3. Диаграмма состояний
4. Диаграмма классов
5. ER-диаграмма
6. Дианрамма развертывания системы

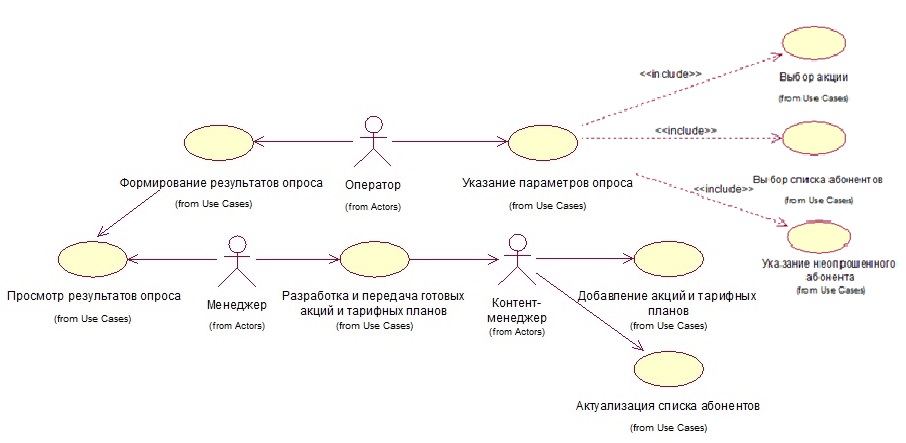


Рисунок Б.1 –Диаграмма прецедентов системы

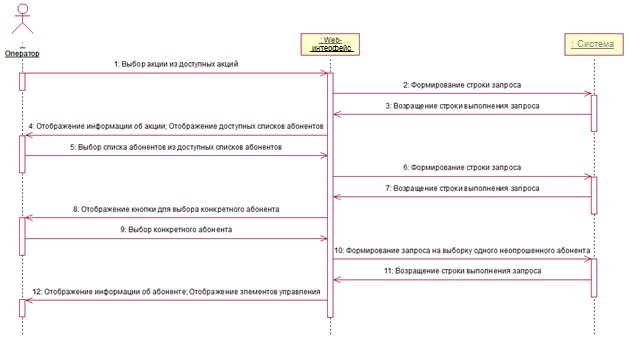


Рисунок Б.2 – Диаграмма последовательности для прецедента «O1. Указание параметров опроса»

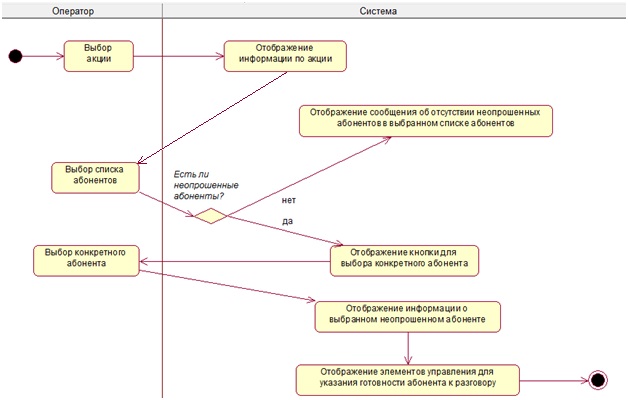


Рисунок Б.3 — Диаграмма состояний для описания возможных состояний прецедента «O1. Указание параметров опроса»

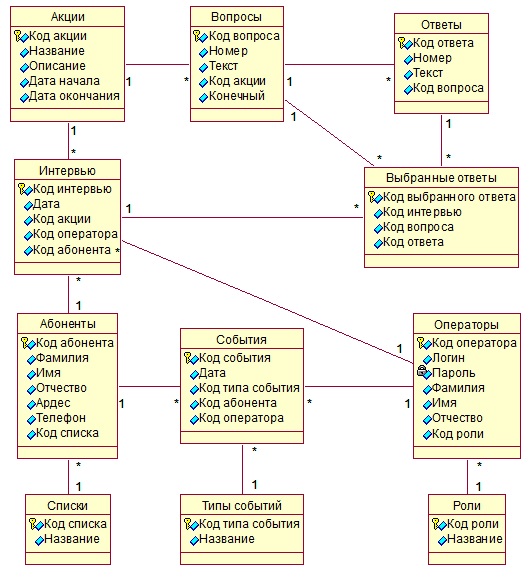


Рисунок Б.4 – Диаграмма классов

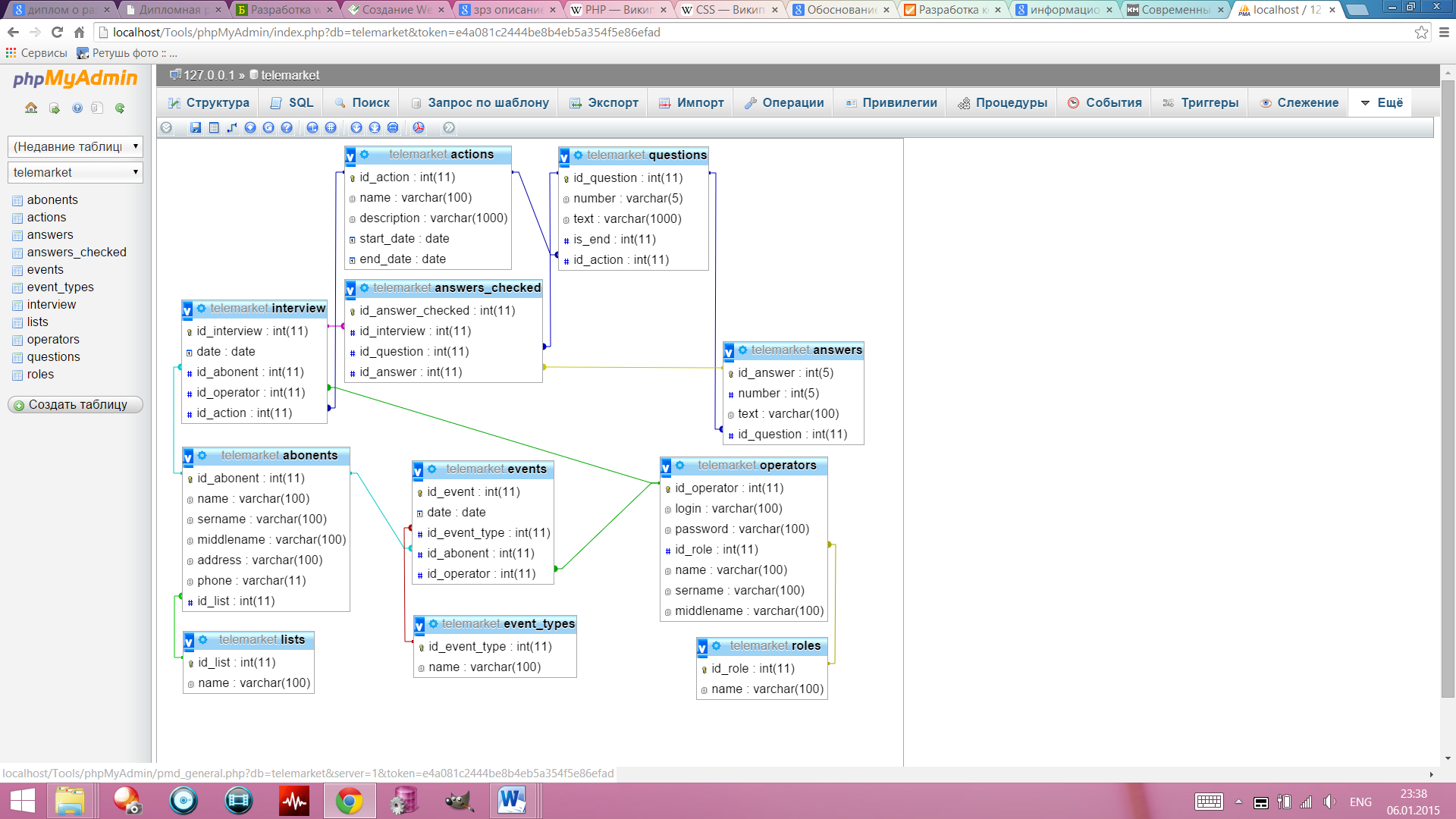


Рисунок Б.5 – ER-диаграмма

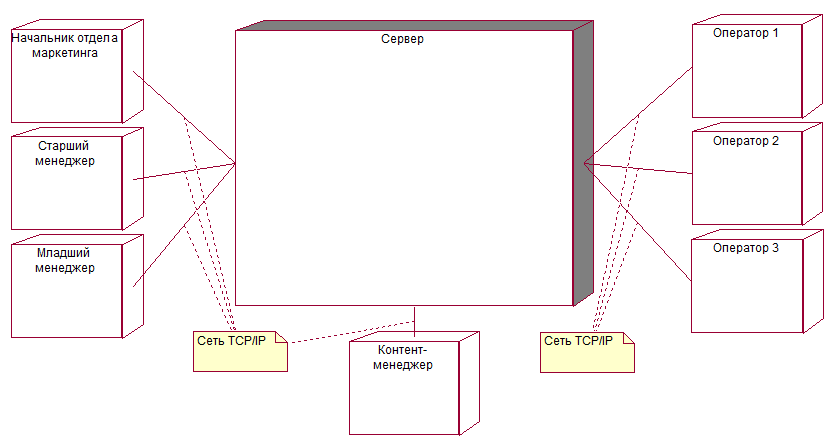
****

Рисунок Б.6 — Диаграмма развертывания системы