

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

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
ПРИКЛАДНОЙ МАТЕМАТИКИ
КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ

**АВТОМАТИЗИРОВАННАЯ СИСТЕМА
АГЕНТСТВ НЕДВИЖИМОСТИ**

Выпускная квалификационная работа
обучающегося по направлению подготовки 02.03.02 Фундаментальная
информатика и информационные технологии
заочной формы обучения,
группы 07001250
Никонова Максима Александровича

Научный руководитель
к.т.н., доцент
Муромцев В.В.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1 ПРОБЛЕМЫ, ВОЗНИКАЮЩИЕ В АГЕНТСТВАХ НЕДВИЖИМОСТИ	7
1.1 Анализ деятельности агентств недвижимости.....	7
1.2 Анализ функционала существующих систем.....	14
ГЛАВА 2 ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ....	24
2.1 Разработка модели и архитектуры автоматизированной системы...	24
2.2 Обоснование выбора СУБД.....	25
2.3 Обоснование выбора программ и инструментов.....	27
2.4 Разработка модели БД.....	28
ГЛАВА 3 РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ.....	30
3.1 Разработка функционала для администрирования.....	30
3.2 Разработка основного функционала.....	32
3.3 Разработка дополнительного функционала.....	47
3.4 Тестирование автоматизированной системы.....	49
ЗАКЛЮЧЕНИЕ.....	51
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	53
ПРИЛОЖЕНИЕ	55

ВВЕДЕНИЕ

Не секрет, что автоматизация различных процессов в современном бизнесе, несомненно, стала привычным явлением. С каждым днем, все сложнее представить себе какой-либо учет без применения специального программного обеспечения, предприниматели пользуются специальными программами для манипуляций с заказами прямо с планшета, ноутбука или мобильного телефона, огромный поток заказов поступает с сайта уже в виде готовых к обработке документов. Но большинство владельцев бизнеса почему-то оставляют без внимания необходимость автоматизации учета взаимоотношений с клиентами.

Как отсутствие автоматизации в отделе продаж влияет на работу всего агентства? Конечно, каждый менеджер работает так, как ему удобнее, ведет учет клиентов, фиксацию звонков и других видов взаимодействия с клиентами по своему усмотрению и своим правилам: одни работают на бумаге, иные в Excel таблицах, а некоторые, к сожалению, совсем не считают необходимым фиксировать процесс своей работы. Данные о звонках или заявках с сайта от новых и постоянных заказчиков тоже не фиксируются, очень часто довольно сложно определить, кто из менеджеров занимается текущей заявкой. Таким образом реально отследить учет возможно только лишь на уровне уже оплаченных и отгружены заказах. Но отследить насколько эффективно работает отдел продаж, обрабатываются ли все поступающие заявки, проводится ли работа с уже имеющимися контактами, невозможно. В связи с этим в случае увольнения или болезни сотрудника, могут потеряться все его неоконченные переговоры и необработанные контакты, а это крайне нежелательно для отличной работы отдела продаж.

Напрашивается только один вариант - автоматизация учета отношений с клиентами. Автоматизированная система управления это - прикладное

программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности, для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путем сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов. Система применима в любом бизнесе, где клиент персонифицирован, где высока конкуренция и успех зависит от предоставления наиболее выгодных для клиента условий.

У автоматизированной системы может быть отличный дизайн и интерфейс, но если система не удовлетворяет даже минимальным требованиям по работе агентства недвижимости, то эта система будет не востребована на рынке. Поэтому стоит провести анализ специализированного функционала автоматизированных систем.

Основной целью внедрения автоматизированной системы, как правило, ставится увеличение степени удовлетворённости клиентов за счёт анализа накопленной информации о клиентском поведении, регулирования тарифной политики, настройки инструментов маркетинга. Благодаря применению автоматизированной централизованной обработки данных появляется возможность эффективно и с минимальным участием сотрудников учитывать индивидуальные потребности заказчиков, а за счёт оперативности обработки - осуществлять раннее выявление рисков и потенциальных возможностей.

Выдвигается гипотеза, что если рассматривать все необходимые для потребителя функции, то использование разработанной нами системы эффективнее, чем использование других ее аналогов.

Цель дипломной работы состоит в разработке автоматизированных систем для агентств недвижимости. Для этого требуется решение следующих задач:

- дать характеристику основных принципов автоматизированных систем.

- проанализировать требования к функционалу системы
- проанализировать недостатки автоматизированных систем
- изучить важные критерии СУБД
- разработать модели СУБД
- разработать интерфейс и схемы навигации

Предметом исследования является автоматизация поиска собственника квартир, анализ, сортировка и группировка входящего потока клиентов, системы интеллектуальной состыковки объект-клиент.

Объектом исследования является полноценная автоматизированная система, которая полностью удовлетворила бы требования компании.

Теоретической основой написания дипломной работы явились учебники, учебные пособия, статьи в периодических изданиях по изучаемому вопросу.

В дипломной работе широко используются такие методы исследования, как: анализ, наблюдение, моделирование и сравнение.

Выпускная квалификационная работа представлена пояснительной запиской и презентацией. Пояснительная записка представлена на 54 листах и одном приложении.

В первой главе даны основные понятия автоматизированных систем. Были рассмотрены существующие автоматизированные системы для агентств недвижимости. Рассмотрены основные принципы автоматизированных систем, а также причины реализации собственной автоматизированной системы. Был проведен анализ и выявлены недостатки сравниваемых систем, результаты анализа представлены в виде сравнительных таблиц, которые были разделены по основным категориям.

Вторая глава работы заключается в проектировании автоматизированной системы для агентств недвижимости. Была разработана архитектура системы и методология проектируемой системы. Был проведен

анализ существующих СУБД и обоснование выбора, а также веб-интерфейса для администрирования СУБД.

В третьей главе описывается процесс программной реализации автоматизированной системы для агентств недвижимости.

В заключении констатируются основные этапы проведенной работы и делаются выводы.

ГЛАВА 1 ПРОБЛЕМЫ, ВОЗНИКАЮЩИЕ В АГЕНТСТВАХ НЕДВИЖИМОСТИ

1.1 Анализ деятельности агентств недвижимости

На нашем Российском рынке работает огромное множество различных агентств недвижимости и частных риелторов, они зарабатывают на операциях, связанных с недвижимостью. Такие фирмы называют агентствами недвижимости. Важно определить, что же такое «риэлтерская деятельность».

Это название пришло из английского языка, недвижимость звучит как «realty». Основой любой, будь то самостоятельная риэлтерской деятельности или деятельность риэлтерских агентств, является любая операция, связанная с недвижимостью.

Риэлтерская деятельность - это производимая на постоянной основе деятельность по предоставлению для других лиц за плату или другое вознаграждение услуг по реализации общественно правовых договоров по недвижимости и правами на нее [1, С. 390].

А это значит, что агентства недвижимости - это юридические лица, которые занимаются риэлтерской деятельностью, непосредственными работниками которых являются риелтор.

Наши российские агентства недвижимости предоставляют не только посреднические услуги, а они сами являются активными субъектами рынка недвижимости: они покупают за счет компании недвижимое имущество и продают его, сдают собственное жилье в аренду, в том числе нежилые помещения, они также проводят самые разнообразные операции с недвижимостью, равно, как и любое физическое лицо.

Большинство агентств недвижимости предоставляют своим клиентам некоторые виды услуг:

- посреднические;
- юридические;
- нотариальные;
- кредитные;
- строительные;
- социальные;
- консультационные;
- банковские;
- оценочные;
- рекламно-информационные;
- страховые.

Каждой фирме необходимо чётко понимать, чем она занимается, зачем она это делает и как необходимо действовать. Решение этих вопросов способствует быстрому и качественному совершенствованию работы фирмы.

Фирма - это живой организм, наделенный своим ритмом, в котором происходит взаимосвязь, взаимодействие друг с другом, взаимовлияние, взаимопонимание, взаимозаменяемость, как с внутренним, так и с внешним миром [2, С. 282].

Риэлтерская деятельность связана со сбором, переработкой и использованием большого объема информации разной степени важности. В такой ситуации важное значение принимает упорядоченная система в работе с информацией, начиная с загрузки и выгрузки, внесения в базу данных, использование для работы с клиентами. В том числе большую роль играет грамотное построение рекламной политики. Учитывая вес расхода на рекламу в нынешних затратах, эффективность её существенно повлияет на работоспособность агентства недвижимости.

Работодателю необходимо тщательно подобрать основной актив собственной фирмы - кадровый состав. Важна квалификация работников,

личные и деловые качества, существующие основные проблемы - оплата труда и управление в фирме [3, С. 272].

Проанализируем деятельность агентства недвижимости ООО «София-недвижимость».

Агентство недвижимости ООО «София- недвижимость» ведет свою деятельность с 2010 года. Представленное агентство является одним из самых перспективных и динамично развивающихся на сегодняшнем рынке недвижимости.

Агентство недвижимости «София- недвижимость» при появлении на рынке предлагало ряд услуг:

- подготовка пакета документов для проведения сделок купли-продажи;
- подготовка пакета документов для передачи жилых помещений.

Главной целью агентства недвижимости «София - недвижимость» является качественное удовлетворение потребностей потребительского рынка в получении необходимой для использования информации.

За 7 лет работы значительно расширился список предоставляемых агентством услуг:

- оказание услуг при оформлении сделок купли-продажи;
- оказание услуг при оформлении сделок купли-продажи коммерческой недвижимости;
- оказание услуг при оформлении договоров приватизации;
- юридические консультации в области недвижимости, в том числе корпоративное обслуживание;
- оказание услуг при аренде коммерческой недвижимости;
- оформление документов на объекты недвижимости;
- оказание услуг по согласованию с государственными органами.

На данный момент агентство недвижимости «София- недвижимость» действует на арендуемой площади.

Верхнюю ступень управления занимает директор, учредитель фирмы. В обязанности директора, как главы агентства, входит контроль и главным образом стратегическое управление компанией: в том числе решение различного рода хозяйственных вопросов, а также координация и корректирование деятельности основных, вспомогательных и обслуживающих функциональных звеньев, управление финансовыми потоками и их распределение, планирование и контроль деятельности фирмы.

Главную и самую объемную часть деятельности компании выполняет штат риелторов. Агенты отвечают за поиск покупателей на выставленные на продажу объекты, а также поиском объектов недвижимости по различным запросам и требованиям клиентов фирмы, в том числе организуют показы объектов и встречи участников сделки. В случае положительного решения о сделке сторон участниц, риелторы подготавливают пакет документов, который предназначается для проведения сделки купли-продажи недвижимости. Агенты предоставляют свою помощь с решением финансовых вопросов сторон и проводят сделку.

Обязанности юриста заключается в правовом обеспечении основной деятельности компании, а также оказывают консультационные услуги клиентам агентства.

В обязанности бухгалтера любого агентства недвижимости входит ведение бухгалтерского учета и предоставление отчетов в контролирующие органы власти.

Функции различного рода различных процессов внутри агентства недвижимости входят в обязанности менеджера. Менеджер направляет работу агентства и его работников, отбирает и подготавливает материалы для размещения рекламы в различных средствах массовой информации, подготавливает документы и договоры, заключаемые с клиентами агентства, в его обязанности также входит обновление базы данных по недвижимости, осуществляет общение с потребителями по сети интернет.

Проанализировав деятельность агентства недвижимости «София-недвижимость», были выделены следующие типы задач:

- Учет клиентов
- Регистрация новых клиентов в базе
- Регистрация новых объектов в базе
- Ведение базы данных
- Получение оперативных выборок объектов и клиентов из базы
- Планирование событий
- Выгрузка рекламы
- Регистрация нового сотрудника в системе
- Ограничения доступов сотрудников к некоторому функционалу
- Удобная и настраиваемая фильтрация базы
- Настраиваемая карточка объекта и клиента
- Аналитика, статистика, графики и отчеты.

Рассмотрим подробнее некоторые из представленных задач:

Учет клиентов:

Ведение учета клиентов - это первый шаг к управлению взаимоотношениями с ними. Руководитель всегда знает, сколько существует, на данный момент, потенциальных клиентов, все контакты в быстром и удобном доступе. Всегда известно, кто общался с клиентом, а вместе с этим какие задачи при этом выполнялись. Становится понятно, как повысить эффективность работы с потребителями, повышая при этом их интерес. Появляется возможность понять, как повысить продажи, предлагая клиентам именно то, что они хотят.

Регистрация новых клиентов в базе:

Клиентская база - не просто собрание различных сведений о клиентах, но важнейший актив компании, неисчерпаемый источник дохода и увеличения продаж. В клиентской базе могут находиться как продавцы недвижимости, так и покупатели. Важно, чтобы риелторы при регистрации

нового клиента в базе могли отмечать важные данные решаемого вопроса, такие как спрос или предложение клиента [4, С. 688].

Современные методы работы с клиентской базой позволяют получить ответы на стратегически важные вопросы:

- кто является клиентом, какие факторы движут им при выборе.
- как снизить стоимость привлечения клиентов;
- что нужно делать, чтобы покупали больше и чаще;
- насколько эффективна была рекламная кампания, массовая рассылка или какая-либо другая маркетинговая инициатива;
- удовлетворены ли клиенты текущими сервисами компании.

Поддерживать клиентскую базу в полном и актуальном состоянии - трудоемкая задача, но и наиболее окупаемая в долгосрочной перспективе. Больше информации - больше способов заручиться доверием и интересом клиента. Если предпочтения, ожидания и поведение в схожей ситуации уже известны, риелтору понадобится вдвое меньше усилий, чтобы клиент совершил покупку.

Грамотное обращение с клиентской информацией дает возможность точно знать, кто будет покупать, что и в каком объеме.

Один из критериев успеха агентства - это объем заявок, обращений от потенциальных покупателей, которые оно получает ежедневно. Заявки и звонки - это важный элемент для любого бизнеса, связанного с продажами.

Эффективный способ получения звонков - это реклама агентства на профильный и многопрофильный интернет порталах, таких как Avito, Из рук в руки, Cian, Winner и других многочисленных сайтах. С помощью автоматической выгрузки объявления будут отправляться на портал ежедневно, без необходимости предпринимать какие-либо дополнительные действия для обеспечения выгрузки. Для каждого объекта в базе должно быть можно выбрать на какие интернет-площадки он будет выгружаться. Ведь разные порталы могут специализироваться на объектах разного типа и

географической привязки. При этом в режиме редактирования объекта все его характеристики должны проверяться на соответствия правилам тех площадок, куда он настроен выгружаться. В случае, если одна или несколько характеристик объекта заполнены неправильно, либо требования площадок поменялись, ответственный агент увидит красный индикатор ошибки, получит возможность ознакомиться с ошибками заполнения и исправить их на страницах редактирования объекта [5, С. 280].

Фильтрация - способ отображения данных выборки из базы данных агентства недвижимости. Фильтры позволяют просмотреть только отдельные записи в форме или таблице либо напечатать некоторые записи из отчета или таблицы. С помощью фильтра можно ограничить объем отображаемых данных, не изменяя макет базовых объектов. Удобная и настраиваемая фильтрация – важный элемент для быстрой, слаженной, а главное удобной работы. При правильно реализованной фильтрации риелторы компании смогут делать быструю и точную выборку данных из базы данных, что значительно повысит эффективность работы.

Каждому успешному агентству необходимо раз в определенный промежуток времени иметь возможность проанализировать эффективность работы. Директору, для улучшения качества работы, необходимо оценить эффективность работы менеджеров, прогнозировать доход, анализировать клиентов. Эффективные отчеты, графики дают возможность оценить и контролировать работу отдела продаж, увидеть объем работ, суммы, выявить критичные точки и еще много всего. Любой пользователь должен иметь возможность переключиться с обычного списка данных на страницу таких отчетов и оценить информацию по ним.

1.2 Анализ функционала существующих систем

Разработчики программного обеспечения часто пытаются навязать свое видение того, что можно назвать автоматизированной системой. Обычно перечень необходимых качеств полностью совпадает с тем вариантом, который они реализовали в своем программном продукте.

В некоторых случаях подобный маркетинг работает достаточно явно и навязчиво. Необходимо понимать, что не существует общих стандартов и четкого понимания того, что подразумевают под термином автоматизированная система. Это может быть любая удобная для бизнеса система управления взаимоотношениями с клиентами. Например, для одного клиентов автоматизированная система - это, прежде всего, учет контактов и взаимодействия с ними, другие возможности он считает необязательными, так как почти не использует. А многие разработчики мощных автоматизированных платформ будут считать, что это понятие намного шире, с их точки зрения даже системы многих банков окажутся недостаточно функциональными, чтобы называться настоящими автоматизированными системами. В среднем и малом бизнесе самое главное требование – это не потерять клиента, не потерять взаимодействия, чтобы все усилия, которые были потрачены на его привлечение, не пропали даром. А потому очень важно контролировать поток входящих звонков и поступление запросов с сайта и по email [6, С. 232].

Основными принципами всех автоматизированных систем является:

- Наличие единого хранилища информации, куда собираются сведения о взаимодействии с клиентами - клиентской базы;
- Использование многих каналов взаимодействия: обслуживание на точках продаж, телефонные звонки, электронная почта, мероприятия, встречи, регистрационные формы на веб-сайтах, рекламные ссылки;

Анализ собранной информации о клиентах и подготовка данных для принятия соответствующих организационных решений.

Автоматизированная система может включать:

- фронтальную часть, обеспечивающую обслуживание клиентов на точках продаж с автономной, распределенной или централизованной обработкой информации;
- операционную часть, обеспечивающую авторизацию операций и оперативную отчетность;
- хранилище данных;
- резервное хранилище данных;
- аналитическую подсистему;
- распределенную систему поддержки продаж: реплики данных на точках продаж или смарт-карты.

Итак, автоматизированная система нужна для того, чтобы:

- Не потерять потенциального клиента, не пропустить ни одного входящего звонка и запроса. В малом и среднем бизнесе в нашей стране конкуренция очень высокая. Компании прилагают значительные усилия для того, чтобы привлечь клиентов, чтобы на них обратили внимание. Автоматизированные системы позволяют получить уверенность, что именно так и будет работать отдел продаж.

- Контроль работы сотрудников и стандартизация работы с клиентами. Без общей стандартизированной системы каждый сотрудник работает так, как он привык. Кто-то ведет учет в электронных таблицах, кто-то - в записной книжке или ежедневнике, кто-то не ведет учет вообще, ориентируется исключительно на отчеты из 1С или на собственную память. Благодаря использованию автоматизированной системы вся рабочая информация собирается в одной общей базе в стандартизированном виде. В результате руководитель может анализировать статистику работы, составлять различные.

- Готовые решения, от которых можно отталкиваться в построении собственной системы работы. Каждая система - это воплощение видения разработчиков того, как нужно работать с клиентом. В ней заложено множество готовых инструментов, которые позволяют перевести работу на качественно новый уровень.

- Ведение учета входящего потока объектов на рынке недвижимости, их продвижение и реклама. Стандартизация заполняемости и полноты информации.

Необходимо проанализировать уровень автоматизации агентств недвижимости. Для более подробного анализа будет сравниваться функционал четырех автоматизированных систем разделенный на три сравнительных этапа.

В таблицах будут сравниваться следующие системы:

- Megaplan CRM
- bpm' online Sales
- Bitrix 24 CRM
- Amo CRM

Для российского бизнеса эти системы имеют ряд преимуществ:

- Данные системы учитывают специфику российского бизнеса, они в большинстве случаев интегрированы с другими решениями, ориентированными на российский рынок;

- они имеют стабильную цену, так как расчет производится в рублях;
- вся документация данных систем представлена на русском языке;
- поддержка пользователей также оказывается на русском языке;
- так как системы российские, специалиста поддержки проще найти, чем для иностранной системы.

Для упрощения анализа функционала автоматизированных систем агентств недвижимости данные будут представлены в виде сравнительных

таблиц. Функционал автоматизированных систем разделим на 3 сравнительных таблицы:

- Администрирования, управление и настройка системы.
- Функционал для профессиональной работы риелтора.
- Дополнительные функции

В первую очередь проанализируем функционал для администрирования у выбранных систем, результаты сравнения представлены в таблице 1.1.

В процессе работы любой компании возникают вопросы, для решения которых должен присутствовать пользователь, с расширенными правами который должен решать следующие вопросы:

- Доступ сотрудников в систему - создание и удаление сотрудников.
- Настройка и редактирование данных компании, загрузка логотипа компании.
- Управление дополнительным (платным) функционалом компании.
- Отчеты, статистики по базе данных.

Таблица 1.1

Сравнение функционала систем для администрирования

Функция системы	Megaplan CRM	bpm ⁷ online Sales	Bitrix 24 CRM	Amo CRM
1. Настройка компании	Да	Да	Да	Нет
2. Создание новых пользователей	Да	Нет	Да	Да
3. Отчеты	Да	Да	Нет	Нет
4. Ограничение доступов	Нет	Нет	Нет	Нет

На основании этой таблицы видно, что не все востребованные функции можно получить, используя уже готовые решения, а некоторый функционал отсутствует полностью. Одним из главных требований агентств недвижимости в их работе это ограничение доступов к разделам данных для некоторых агентов. Так же при разработке собственной системы следует отметить, что необходимо ограничивать доступ не только к разделам базы данных, но и доступ по времени к системе в целом. При разработке собственной системы необходимо реализовать такую возможность.

Главной отличительной особенностью для многих систем является ее функционал, многие агентства недвижимости уже работали во многих автоматизированных системах и уже имеют свои потребности в специализированном функционале [7, С. 736]. Чтобы понять насколько существующие системы удовлетворяют спросу среди агентств недвижимости необходимо сравнить требования компаний с тем, что предлагает рынок систем. Для удобства сравнения результаты будут представлены в виде сравнительной таблицы 1.2.

Создавая базу данных, агентства недвижимости стремятся упорядочить информацию по различным признакам и быстро извлекать выборку с произвольным сочетанием признаков. Сделать это возможно, только если в системе будет реализован качественный алгоритм поиска информации. Многие готовые решения, приведенные в сравнительной таблице, имеют возможность делать выборки из базы, но они не учитывают тонкости работы риелтора. Такая важная функция как закрепление заявки за агентом вообще не реализована во всех сравниваемых системах, а это крайне важно в работе риелтора, так как во многих компаниях часть объектов представляется эксклюзивно от знакомых риелтора и очень важно ограничить доступ к этой информации от других риелторов компании.

Таблица 1.2

Сравнение функционала систем для ведения баз данных

Функция системы	Megaplan CRM	bpm' online Sales	Bitrix CRM	24 Amo CRM
1. Создание и Изменение информации	Да	Да	Да	Да
2. Возможность закрепления карточки	Нет	Нет	Нет	Да
3. Планировщик	Нет	Нет	Нет	Да
4. Персональная реклама объектов	Нет	Нет	Нет	Нет
5. Архив данных	Нет	Да	Нет	Да
6. Комментарии или заметки к объектам	Нет	Нет	Да	Нет
7. Возможность поделиться данным объектом с другими агентствами	Нет	Нет	Нет	Нет
8. История изменения данных по объекту	Да	Нет	Да	Нет

Все приведенные в сравнительной таблице автоматизированные системы дают возможность создания и редактирование информации в базе данных, но в работе риелтора есть такое понятие - как закрепление объекта. Закрепляя за собой заявку, агент назначает себя ответственным за решение вопроса закрепленной заявки - такую возможность не дает ни одна из готовых систем.

Каждый день риелтор совершает сотни звонков по сотням различных объектов, он согласовывает показы квартир с покупателями и для автоматизации этого процесса необходим правильный планировщик, представленный в виде календаря, где риелтор может планировать свой день.

Исходя из сравнения существующих систем, лишь только у Amo CRM есть такая функция. Еще одним важным процессом работы любого риелтора

заключается реклама объектов, т.к. доски объявлений требуют платную регистрацию, риелторы используют множество учетных записей, зарегистрированных как частное лицо. Каждое поданное объявление необходимо продлевать каждые 20-30 дней. В связи с этим у риелторов встает задача - как хранить данные учетных записей и как помнить, когда необходимо продлить то или иное объявление. Поэтому необходима такая функция как "Персональная реклама объектов", которая отсутствует во всех приведенных автоматизированных системах.

В процессе работы любого агентства недвижимости появляются клиенты, которые отложили свой вопрос на длительный период, для этого необходим архив объектов, который будет хранить такие заявки. Комментарии или заметки к объектам служат для хранения личных данных по конкретному объекту недвижимости и предназначены для упрощения запоминания важных данных и упрощению работы риелтора в целом.

Так как на рынке существует множество агентств недвижимости, которые используют разные способы привлечения клиентов, многие из которых являются эксклюзивными клиентами того или иного агентства - встает вопрос о возможности информирования других агентств о взаимовыгодном предложении на рынке недвижимости. Такая возможность отсутствует во всех приведенных автоматизированных системах, в очередной раз показывая, что рынку требуется собственная специализированная система.

Последним этапом анализа функционала автоматизированных систем является сравнение дополнительных функций, представленных в таблице 1.3. Дополнительный функционал должен иметь множество специфических функций служащих для более тонкой настройки системы, пользователя, а также для возможных интеграций со сторонними программами.

Таблица 1.3

Сравнение дополнительного функционала систем

Функция системы	Megaplan CRM	bpm' online Sales	Bitrix CRM	24 Amo CRM
1. Изменение личной информации	Да	Да	Да	Да
2. Сообщения между сотрудниками	Нет	Да	Нет	Да
3. Отчеты	Да	Да	Нет	Нет
4. Массовая реклама объектов	Нет	Нет	Нет	Нет
5. Анализ данных базы	Нет	Нет	Да	Да
6. Выявление дублирующих записей	Нет	Нет	Нет	Нет

Все приведенные системы дают возможность поменять пользователю информацию о себе, сменить личную фотографию, но далеко не во всех системах реализована возможность отправки личных сообщений между сотрудниками, а сообщения — это большая часть работы, так как сотрудники могут находиться на больших расстояниях, работая на дому или находясь на показах объектов недвижимости.

Возможность XML выгрузки рекламы на бесплатные сайты не возможна не в одной из приведенных в таблице систем, а бесплатными сайтами пользуются тысячи людей. Отсутствие такой возможности лишает агентства недвижимости сотен звонков ежедневно. Каждый день на рынке появляются десятки новых квартир, которые возможно несколько лет назад уже были добавлены в базы, такая функция, как "Выявление дублирующих записей" должна предотвращать появление дублирующих записей в базе.

Сравнивая автоматизированные системы, стало известно, что на рынке нет системы удовлетворяющей запросы стандартного агентства недвижимости даже на 40%, а системы специализирующейся на работе агентств недвижимости не учитывают базовые потребности и особенности работы риелтора. Почти все представленные системы конкурентов имеют

небольшие расширенные настройки для риелторов, но они не удовлетворяют потребности рынка.

Основные задачи, которые не реализованы в сравниваемых автоматизированных системах:

- Возможность ограничения доступа к базам данных сотрудникам, директором компании.
- Возможность директору компании устанавливать ограничения на авторизацию во временном интервале для сотрудников.
- Возможность риелтору закреплять за собой объекты компании.
- Планировщик и менеджер рекламы учитывающий особенности работы риелтора.
- Возможности комментирования и оставление индивидуальных заметок для объектов.
- Сообщения между сотрудниками.
- Архив и история изменений данных.
- Возможность поделиться данным объектом с другими агентствами.
- Массовая выгрузка объектов на бесплатные доски объявлений.
- Выявление дублирующих записей в базе.

При разработке собственной системы основной наклон будет направлен на реализацию всех основных потребностей агентства недвижимости.

Разработанная система позволит директору компании не только контролировать весь рабочий процесс, но и в разы повысить эффективность работы сотрудников.

Стоит отметить, что многие риелторы большую часть своего рабочего времени работают вне офиса, а это означает, что разрабатываемая система должна иметь адаптивный под мобильный и планшетный дизайн. Кроме этого, для обеспечения быстрого доступа с мобильного или планшета страницы системы должны иметь малый загружаемый вес.

Используя профессиональные инструменты, риелторы, не прилагая усилий, могут отслеживать любые изменения на рынке, оперативно получать информацию по новым объектам и оповещать о них своих клиентов. Администратор автоматизированной системы в деталях видит работу, как каждого отдельного сотрудника, так и работу всех филиалов в целом.

ГЛАВА 2 ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ

2.1 Разработка модели и архитектуры автоматизированной системы

На первом этапе проектирования разрабатываемой среды решается вопрос, как реализовать решения, принятые на этапе анализа. Для этих целей будем использовать методологию IDEF3 (см. рис. 2.1).

Согласно данной методологии система представляется в виде функциональных блоков, так же могут использоваться различные соединители [8, С. 384].

Входом в систему являются "Логин" и "Пароль", после того как пользователь ввел "Логин" и "Пароль" осуществляется авторизация и проверка его прав. Далее происходит формирование стартовой страницы пользователя. Далее пользователь может выбрать из режимов работы только один раздел системы, которые разделяются на:

- Объекты компании;
- Объекты пользователя;
- Контакты;

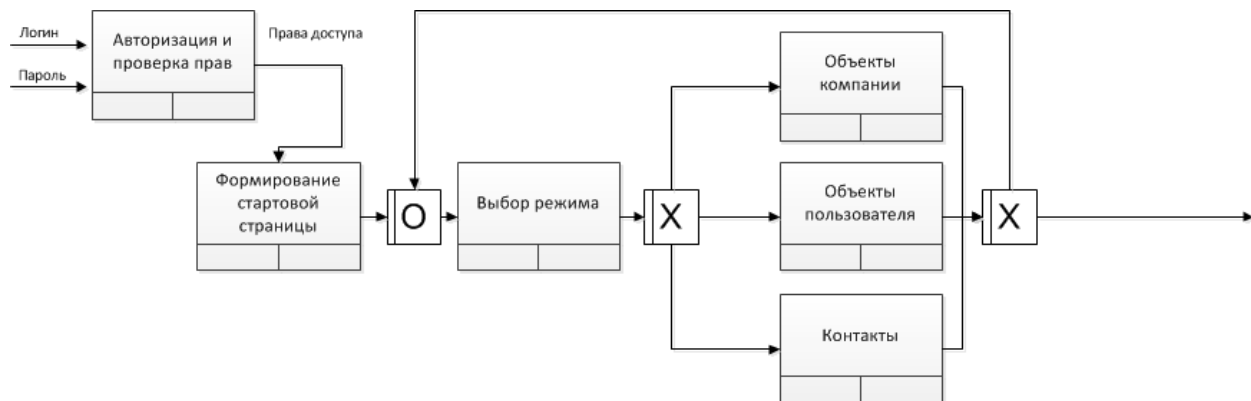


Рис. 2.1. Методология IDEF3

Понятие архитектуры в значительной мере субъективно и имеет множество противоречивых толкований. На этапе проектирования автоматизированной системы необходимо понимать, как будет происходить взаимодействие пользователя и программной части сервера. Представим процесс обращения пользователя к серверу и схематично покажем процессы обработки данных и запись их в базу MySQL (см. рис. 2.2).

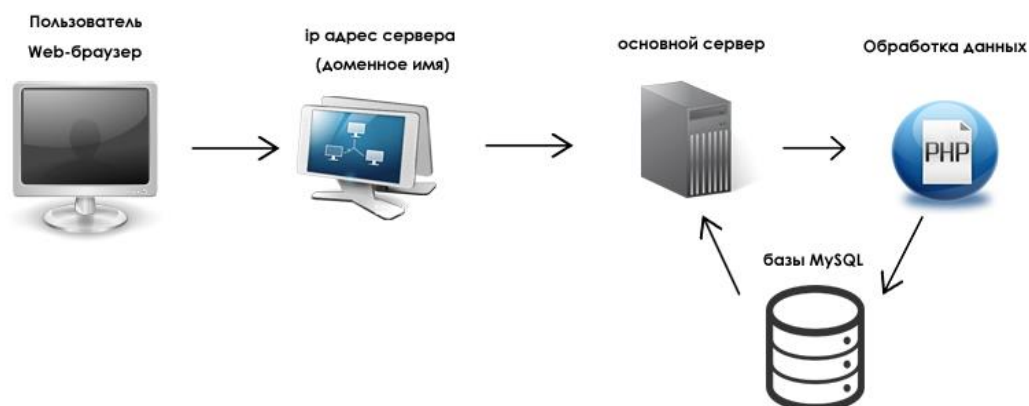


Рис. 2.2. Процесс взаимодействия пользователя и сервера

2.2 Обоснование выбора СУБД

Система управления базами данных (СУБД) - это программное средство, служащее для ведения логически взаимосвязанных данных на машинном носителе, а также обеспечивающее доступы к данным хранящихся в базах. Современные базы данных можно разделить на три категории:

- Программные продукты корпоративного направления - Oracle и MS SQL Server;
- СУБД для работы с информационными массивами в небольших компаниях, - MS Access и Borland Interbase;
- СУБД для Web, реализующих создание web-сайтов с небольшими базами данных, - Oracle, MS SQL Server, Borland Interbase и MySQL.

Так как разрабатываемая автоматизированная система расположена на Web сервере, для сравнения будем рассматривать только Oracle MS SQL Server, Borland Interbase и MySQL.

СУБД должны обеспечивать высокую надежность хранения данных, что обеспечивается резервным копированием; безопасными - иметь защиту от несанкционированного доступа; работать с огромными объемами данных и обладать широкими функциональными возможностями.

СУБД для Web отличаются высокой скоростью обработки данных, нетребовательностью к ресурсам и удобным удаленным администрированием.

Наиболее популярными СУБД для Web являются Oracle, MS SQL Server, Borland Interbase, MySQL и MS Access.

Одна из отличительных особенностей сервера Oracle это возможность хранения и обработки множества типов данных. Такая функциональность встроена в ядро СУБД и поддерживается модулем interMedia в составе Oracle Database. Таким образом обеспечивается работа с текстовыми документами, включая различные виды поиска. СУБД Oracle не только предоставляет обширный набор встроенных типов данных, но и позволяет конструировать новые типы данных со спецификацией методов доступа к ним.

СУБД InterBase привлекает крайне низкими системными требованиями и при этом достаточно высокой производительностью и легкостью в администрировании. Кроме этого, архитектура позволяет отойти от использования протокола транзакций, который в других СУБД служит для восстановления базы данных после сбоев, поэтому InterBase обладает очень высокой надежностью и устойчивостью.

Основным компонентом базы данных является таблица. В таблицах хранятся вводимые пользователем данные. Каждая таблица состоит из столбцов, которые называют полями, и строк, называемых записями.

MySQL это свободная система управления базами данных. Разработку и обновление MySQL осуществляет корпорация Oracle. СУБД MySQL

обеспечивает высокую гибкость благодаря поддержке большого количества различных типов таблиц [9, С. 320].

Основными преимуществами СУБД MySQL является:

- MySQL запускается на любой ОС, что обеспечивает ее мультиплатформенность.
- MySQL является бесплатным продуктом.
- MySQL имеет интеграцию с языками Delphi, C, C++, Эйфель, Java, Лисп, Perl, PHP, Python, Ruby
- Система, основана на привилегиях и доступам по паролям, за счет этого обеспечивается гибкость и безопасность при работе.
- Управляет очень большими базами данных.

Таким образом разработку базы данных для автоматизированной системы будет осуществляться с помощью СУБД MySQL.

2.3 Обоснование выбора программ и инструментов

Основываясь на разделе 2.1, где была выбрана СУБД MySQL появляется необходимость в удобной панели администрирования базы данных. Для этих целей будет рассмотрено веб-приложение phpMyAdmin, которой предоставляется бесплатно хостинг провайдером, на котором размещается автоматизированная система для агентств недвижимости.

PhpMyAdmin это веб-приложение с открытым кодом, написанное на языке PHP и представляет собой интерфейс для администрирования СУБД MySQL. Приложение позволяет осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных [11, С. 146].

На сегодняшний день PHPMyAdmin активно применяется на практике. Большинство российских провайдеров используют это приложение в

качестве панели управления для того, чтобы предоставить своим клиентам возможность администрирования выделенных им баз данных.

Программа является наиболее распространенным средством для администрирования, управления и работы с базами данных MySQL.

Приложение имеет встроенные функции для проведения всех необходимых работ - обслуживание, проверка и починка испорченных таблиц, резервирование и восстановление из архивов, экспорт данных из таблиц.

Программа позволяет администратору выполнять все SQL-запросы над базами, причем это можно делать как из специального, окна запросов, так и для каждой таблицы отдельно. На наиболее популярные запросы можно делать закладки и в дальнейшем получить доступ к ним по одному нажатию на закладку.

Строить запросы к базам можно в MySQL Query Browser, там же присутствует полноценный отладчик SQL, администрировать сервер и тонко управлять производительностью можно программой MySQL Administrator - такого множества функций и настроек сервера баз данных не предоставляет никакая из существующих утилит. Импорт и экспорт данных, создание резервных копий баз лучше всего получается утилитами компании EMS.

2.4 Разработка модели БД

В своей работе агентства недвижимости ведут множество записей, постоянно внося новые объекты или меняя старые записи. Для хранения информации выбрана СУБД MySQL и веб-приложение phptoadmin. При разработке автоматизированной системы необходимо разработать модель базы данных, показать, как будет происходить взаимодействие и хранение информации (см. рис. 2.3).

Логическая модель расширяет концептуальную путем определения для сущностей их атрибутов, описаний и ограничений, уточняет состав сущностей и взаимосвязи между ними. Целью построения логической модели является получение графического представления логической структуры исследуемой предметной области.

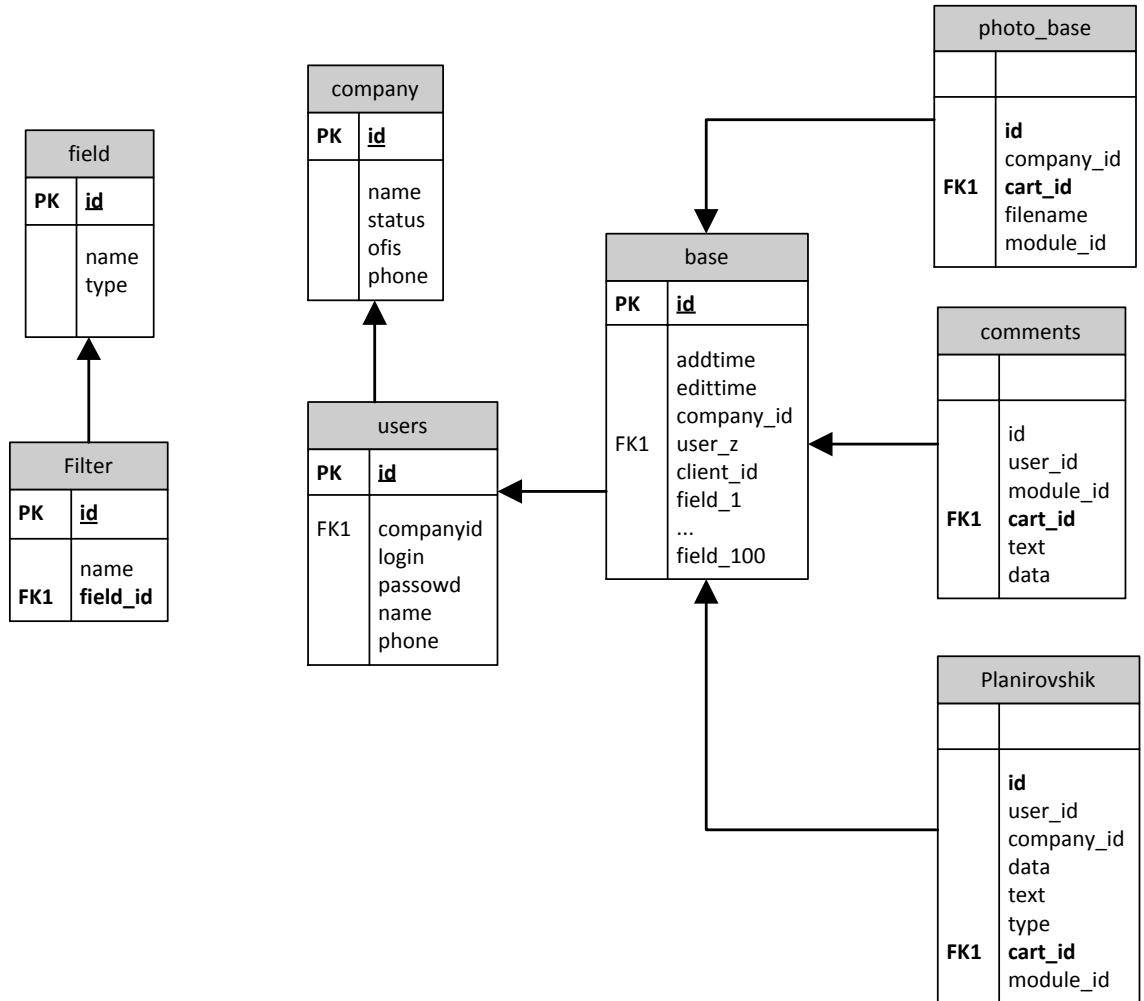


Рис. 2.3. Логическая модель базы данных

ГЛАВА 3 РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ

3.1 Разработка функционала для администрирования

Авторизация - процесс проверки (подтверждения) прав при попытке выполнения этих действий. Часто можно услышать выражение, что какой-то человек «авторизован» для выполнения данной операции - это значит, что он имеет на неё право.

Чтобы перейти на главную страницу сайта пользователь должен пройти процесс авторизации в системе, которая будет реализована с помощью HTML формы (см. рис. 3.1), методом POST. В случае успешной авторизации пользователь будет перенаправлен на стартовую страницу своего профиля.

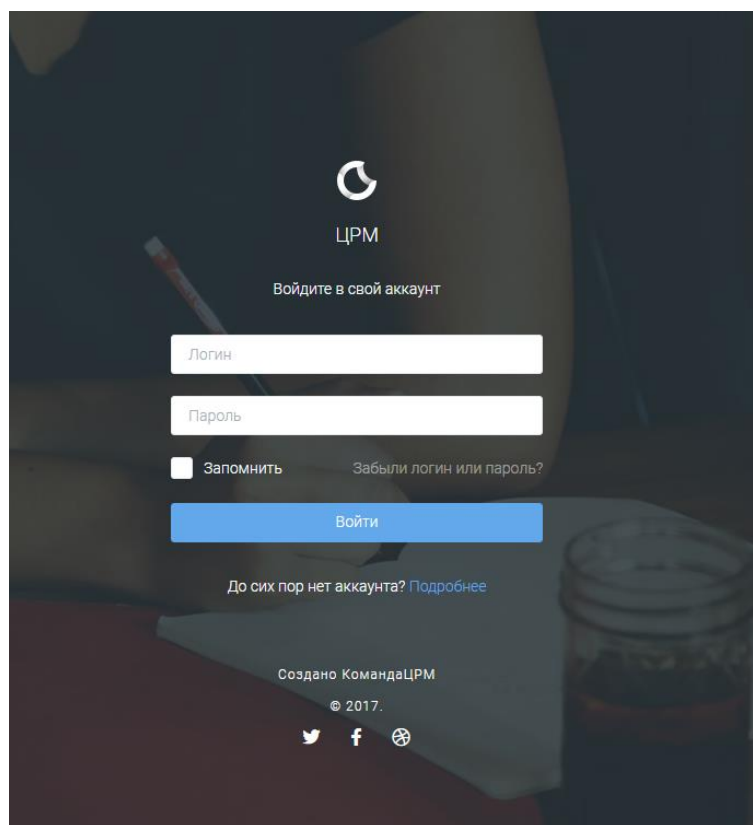


Рис. 3.1. Форма авторизации пользователя

При нажатии кнопки войти, выполняется PHP функция `checkPass` (см. листинг 3.1), которая возвращает 1 в случае успешной авторизации и 0 в случае если введенных данных нет в базе.

Листинг 3.1 Функция `checkPass`

```
function checkPass($login, $password) {
    if ((empty($login)) or (empty($password))) {return false;} else {
        $password = str_replace("'", "", $password);
        $login = str_replace("'", "", $login);
        $query      = "SELECT * FROM `users` WHERE `login` = '". $login.'" and
`password` = '". $password.'"";
        $result     = mysql_query($query);
        if(mysql_num_rows($result)==1) {
            $row=mysql_fetch_array($result);
            return $row;
        }
        return false;
    }
}
```

Чтобы исключить SQL инъекции при авторизации используется функция стандартная функция `str_replace`. Если авторизация прошла успешно, то выполняется функция `cleanMemberSession` (см. листинг 3.2), которая создает массив `$_SESSION`.

Листинг 3.2 Функция `cleanMemberSession`

```
function cleanMemberSession($login, $password) {
    $db_handle = new DBController();
    $query      = "SELECT * FROM `users` WHERE `login` = '". $login.'" and
`password` = '". $password.'"";
    $result     = mysql_query($query);
    $row1      = mysql_fetch_array($result);
    $_SESSION["id_user"]      = $row1['id'];
    $_SESSION["name"]        = $row1['name'];
    $_SESSION["login"]       = $login;
    $_SESSION["password"]    = $password;
}
```

```
$_SESSION["loggedIn"] = true;
$_SESSION["companyid"] = $row1['companyid'];
echo '<META HTTP-EQUIV="Refresh" CONTENT="0; URL=/html/index.php">';
```

Далее происходит перенаправление пользователя на стартовую страницу системы, где пользователь должен выбрать раздел базы в котором он хочет продолжить работу. После успешной авторизации система проверяет доступы пользователя и формирует навигационное меню [12, С. 352].

3.2 Разработка основного функционала

Как и во всех компаниях, агентства недвижимости ведут постоянный поиск новых сотрудников. Поэтому для полноценной автоматизации системы для агентств недвижимости необходимо, чтобы директор мог добавлять новых сотрудников с определенными правами, процесс создания сотрудника будет реализован с помощью HTML формы (см. рис. 3.2). Из общения со многими директорами агентств недвижимости, стали известны многие случаи так называемых "черных" сделок. Поэтому агентства крайне обеспокоены за сохранность своих данных - необходимы ограничения для сотрудников по доступам к базе компании. Создание сотрудника будет проходить в разделе доступном только директору компании. Для хранения данных о сотрудниках создадим таблицу users (см. рис. 2.3), в которую будем записывать данные о сотрудниках компании. После того как пользователь будет создан, ему можно будет назначить роля и доступы.

Важно реализовать возможность для ограничения работы во временном интервале, чтобы гарантировать, что новый сотрудник будет заходить в систему только в офисе, пока он проходит обучение, а не в свое личное время, например, из дома. Благодаря такой возможности, агентства недвижимости могут исключить "черные" сделки, проходящие скрытно.

Создание сотрудника

Права администратора
 Видит только свои объекты
 Видит только своих клиентов

ФИО сотрудника:

Логин и пароль:

Контакты:

Доступ к системе:

Рис. 3.2. Форма создания сотрудника

Автоматизированная система будет строиться на модульной основе, где модули представляют собой таблицы с данными в базе данных. Структура меню навигации представляет собой ссылки на файлы. Для удобства работы с системой предусмотрена настройка колонок таблицы, данные записываются в таблицу filter (см. рис. 2.3).

Большинство агентств недвижимости начинали вести свою базу в таблицах Excel, поэтому, внешний вид базы будет представлен в виде таблицы с настраиваемыми колонками. Для настройки колонок пользователю необходимо создать свой фильтр, где он сможет выбрать любое количество колонок из карточки, в зависимости от типа объекта.

Для перехода пользователя в режим создания колонок таблицы ему необходимо нажать на кнопку "Создать фильтр" (см. рис. 3.3), для удобства пользователя используется класс dropdown имитирующий выпадающий список, этот класс является стандартным и доступен в библиотеке bootstrap [13, С. 176].

Квартиры

[+ Добавить объект](#) [+ Поиск](#)

id	Объект	кол.	Общая	Населенный Пункт	Улица	№ Дома/Позиции
19	Квартира	1		Белгород	Богдана Хмельницкого ул.	21г
17				Белгород	2 Лесной пер.	2
14				Белгород	Спортивная ул.	11

Мой фильтр ▾

- Стандарт
- Мой фильтр
- [+ Создать фильтр](#)

Рис. 3.3. Функциональные действия с базой

При нажатии кнопки "Создать фильтр", срабатывает стандартный в библиотеке bootstrap плагин modal, которое разворачивает поверх экрана пользователя диалоговое окно создания фильтра (см. рис. 3.3). Пользователю необходимо отметить галочками колонки, которые он хочет использовать в работе.

Создание нового фильтра

Название фильтра

Населенный Пункт

№ Корпуса

Тип Этажа

Объект

Актуальность

Тип Квартиры

Планировка

Особенности

Улица

№ Этажа

ЖК, Застройка

Кол. Комнат

Дата

Цена от СБ

С/У

Ремонт

Общая

№ Дома/Позиции

№/под./ кв./домофона

Вопрос

Цена в Рекламе

Стр

Балкон

Благоустройство

Кухни

Рис. 3.3. Диалоговое окно создание фильтра

Поля в форме располагаются аналогично их порядку в карточке, чтобы пользователь не терялся среди большого количества данных. После того, как пользователь заполнил поле "Название фильтра", он должен отметить нужны поля, нажав на них левой кнопкой мышки. Чтобы создать фильтр, пользователю необходимо нажать кнопку "создать" расположенную в низу формы.

После этого данные отправляются методом POST в функцию-обработчик add_filter (см. листинг 3.3).

Листинг 3.3 Функция add_filter

```

if (isset($_POST["name"])) {
$name = $_POST["name"];
$module = $_POST["module"];
$userid = $_POST["userid"];
$companyid = $_POST["companyid"];
$defolt = $_POST["defolt"];
$add = mysql_query("INSERT INTO `lfilter` (module, userid, companyid,
name, defolt) values('".$module."', '".$userid."', '".$companyid."',
'".$name."', '".$defolt."')");
if (empty($_POST["defolt"])) { $defolt = 0; } else {
    $defolt = 1;
}
$sql = mysql_query("UPDATE `lfilter` SET `defolt` = '0' WHERE
`userid` LIKE '$_SESSION[id_user]' .'' AND module='".$module.'');
$sql = mysql_query("UPDATE `lfilter` SET `defolt` = '1' WHERE `id`
LIKE '$filterid.'" AND module='".$module.'');
}
$id_sql = mysql_query("SELECT MAX(id) as max FROM lfilter");
$row = mysql_fetch_array($id_sql);
$id = $row['max'];
$result = mysql_query("SELECT * FROM `lfield` WHERE deleted = '0'
AND (module = '".$module.'" OR module = '13')");
while ($field_str = mysql_fetch_assoc($result))
{
if (empty($_POST["field"].$field_str['id'])) { }
else {
$add = mysql_query("INSERT INTO `lfilter_list` (filter_id, fieldid)
values('".$id."', '$field_str[id].')");
}
}
}
}

```

Таким образом пользователь может создавать любое количество фильтров под разные нужды, выбирая нужные для визуального восприятия поля.

При работе с таблицами обычно пользователи создают для себя 3 фильтра: для объектов, для рекламы, для работы с районом.

Согласно выбранному фильтру генерируется таблица с данными. Для этого необходимо определить какой фильтр использует пользователь в текущий момент. Для этого передается GET параметр `filter`, который хранит в себе `id` фильтра.

Колонки таблицы для удобства пользователя располагаются вверху и дублируются с низу таблицы тем самым при выводе больших объемов данных пользователь будет всегда видеть к какой колонки относятся значения таблицы [14, С. 368].

После того как `id` фильтра известен, необходимо обратиться к таблице `filter_list`, которая хранит в себе колонки таблицы. После чего можно вывести эти колонки в таблицу с помощью цикла (см. листинг 3.4):

Листинг 3.3 Генерация колонок фильтра пользователя

```
while ($fields_str2 = mysql_fetch_assoc($filter_sql))
{
    $fields_row
    ='`'.$field_str2['module'].'base`.`field'.$field_str2['id'].'`,`';
    $field_name = mysql_query("SELECT id,name FROM `1field` WHERE id
    = '".$fields_str2['fieldid']."' AND (module = '".$module."' OR
    module = '13') "); $field_str2 = mysql_fetch_assoc($field_name);
    echo "<th>".$field_str2["name"]."</th>";
}
```

Так как, мы знаем, какие поля будут использоваться при выводе таблицы, для оптимизации MySQL запроса мы можем сформировать строку `$fields_row`, содержащую в себе нужные колонки. Благодаря этому объем выводимых данных можно снизить вплоть до 90%, а это в свою очередь влияет на скорость загрузки страницы.

У большинства агентств базы включают в себя информацию о тысячах объектов, поэтому для обеспечения максимально быстрой загрузки страниц будем использовать пагинацию, разбивая вывод информации по N записей на страницу. Для этого необходимо подключить обработчик пагинации (см. приложение)

```
require_once("/gen_files/auth/pagination.class.php");
```

Под пагинацией понимают показ ограниченной части информации на одной веб-странице. Она повсеместно используется в веб-приложениях для разбиения большого массива данных на странице и включает в себя навигационный блок для перехода на другие страницы. Благодаря пагинации агентства недвижимости могут создавать и хранить базы любого размера, не влияя на скорость работы системы.

После того как мы сформировали MySQL запрос к базе, вывели колонки можно преступить к выводу данных о объекте из базы данных (см. листинг 3.5).

Листинг 3.5 Вывод информации в таблицу

```
$objects = "SELECT $fields_row FROM `".$module."base`
WHERE `".$module."base`.`companyid`='".$_SESSION['companyid']."'
$poisk ";
$filename = basename(__FILE__, ".php");
$paginationlink = $filename.".php?page=";
$page = 1;
if(!empty($_GET["page"])) { $page = $_GET["page"]; }
$start = ($page-1)*$perPage->perpage;
if($start < 0) $start = 0;
$query = $objects . " LIMIT " . $start . "," . $perPage->perpage;
$result = $db_handle->runQuery($query);
$sql = mysql_query($objects);
if(empty($_GET["rowcount"])) { $_GET["rowcount"] =
mysql_num_rows($sql); }
$perpageresult = $perPage->perpage($_GET["rowcount"],
$paginationlink);
if(!empty($result)) { foreach($result as $k=>$v) {
echo "<tr>";
echo "<td>".str_replace(' |##| ',',',,$result[$k][$str])."</td> " ;
echo "<td><a
href='object.php?module=".$module."&id=".$result[$k]['id']."'
target='_blank' ><i class='icon wb-edit' aria-
hidden='true'></i></th>";
echo "</tr>";
```

Такой вывод информации позволяет быстро выводить большие объемы данных, для вывода ссылок перехода по страницам используется функция `$perpageresult`, которая вызывается в конце таблицы с данными.

В процессе вывода информации на экран алгоритм создает в строке ссылки на выводимые объекты, для повышения удобства пользователей и сокращения количества загрузки страниц используется HTML тэг `target='_blank'`, который сообщает браузеру пользователя о необходимости открытия данной ссылки в новой вкладке. По завершению работы с карточкой предполагается, что пользователь просто закроет вкладку с карточкой и продолжит работу с таблицей без каких-либо дополнительных действий. Следует отметить, что такой способ открытия карточки не закрывает уже загруженную таблицу с объектами, что сокращает количество загружаемого трафика пользователю и уменьшает нагрузку на сам сервер.

Одним из основных требований агентств недвижимости является быстрый и удобный поиск информации по своим базам данных. Основной особенностью автоматизированной системы для агентств недвижимости заключается в большом количестве полей необходимых риелтору для полноценной работы. Для решения любых задач по поиску информации система предоставляет возможность поиска по всем полям, существующим у выбранного вида объекта [15, С. 288].

Для удобства пользователя реализован поиск с помощью диалогового окна (см. рис. 3.4), которая отвечает за вызов класса `modal`. Следует отметить, что для удобства пользователя данные введенные после отправки формы будут подставлены в это поле с помощью `$_GET`, что позволит делать быструю коррекцию данных для поиска.

Рис. 3.4. Диалоговое окно поиска

Данные из формы поиска передаются методом GET, далее пока идет процесс вывода информации на экран (см. листинг 3.6) - в цикле происходит проверка типа поля и вывод формы для заполнения.

Листинг 3.6 Генерация полей карточки

```

$field = mysql_query("SELECT id,name,type,module,listid FROM
`field` WHERE `deleted` = '0' AND `module` = '". $module.'" AND
`fieldset` = '". $fieldset2_str['id']."' ORDER BY `row` ASC ");
while($field_str = mysql_fetch_assoc($field))
{
<select name="field<?php echo "' . $field_str["id"] . "' ?>" >
  <option value="">Выберите</option>
  $users = mysql_query("SELECT id,name FROM
`list". $field_str['listid'] . "`");
while ($users_str = mysql_fetch_assoc($users))
{
if ($_GET['field'. $field_str['id']] == $users_str['name'])
{
?>

```

```

    <option selected value="<?php echo $users_str['name'] ?>"><?php
echo $users_str['name'] ?></option>
    <?php
    } else {
    ?>
    <option value="<?php echo $users_str['name'] ?>"><?php echo
$users_str['name']; ?></option>
    <?php
    } }
</select>

```

После отправки формы, необходимо получить введенные пользователем данные. Для этого необходимо записать их в строку (см. листинг 3.7), которую в дальнейшем мы будем использовать при выводе информации из базы. В результате мы получим строку, в которой будут прописаны все необходимые MySQL запросы. Примером такого запроса к базе будет:

Листинг 3.7 Генерация MySQL запроса для поиска по базе

```

$get = "field".$field_str['id'];
$value = $_GET[$get];
$poisk .=" AND
`".$field_str['module']."base`.`field".$field_str['id']."` = '".$value.'";
}
SELECT
`1base`.`id`,`1base`.`addtime`,`1base`.`edittime`,`1base`.`userz`,`1base`.`co
mpanyid`,`1base`.`field188`,`1base`.`field189`,`1base`.`field192`,`1base`.`fi
eld219`,`1base`.`field220`,`1base`.`field221` FROM `1base
WHERE `1base`.`field219` = 'Белгород' AND `1base`.`field220` = '60 лет
Октября ул.' AND archive = "0" AND deleted = "0" AND `1base`.`companyid`='3'

```

Данный MySQL запрос возвращает из базы объекты по адресу "Белгород, 60 лет Октября ул.", которые принадлежат агентству с id = 3.

В своей работе агентства недвижимости каждый день получают информацию о новых объектах недвижимости, для занесения новой записи в базу данных пользователю необходимо нажать на кнопку "Добавить объект"

(см. рис. 3.3). После нажатия на кнопку пользователя переходит в раздел создания новой записи (см. рис. 3.5).

Поля, которые необходимо заполнить пользователю идентичный, как и в карточке объекта (см. рис. 3.7), но сам процесс создания объекта разделен на 4 логических этапа:

- Характеристики объекта;
- Характеристики дома, в котором находится этот объект;
- Фотографии объекта;
- Контакты собственника;

The screenshot shows a multi-step form for creating a new record. The first step, 'Параметры объекта' (Object parameters), is highlighted in blue and marked as 'Заполнение полей' (Field filling). Below it, there are several input fields: 'Адрес' (Address) with sub-fields for 'Населенный Пункт' (Settlement Point), 'Улица' (Street), '№ дом/поз/кор' (House/Flat/Corridor), and 'Секция' (Section); 'Ориентиры+Инфраструктура' (Landmarks+Infrastructure) with sub-fields for 'Ориентир' (Landmark) and 'Инфраструктура' (Infrastructure); and a 'Выберите' (Choose) button.

Рис. 3.5. Этапы создания новой записи в базе данных

После того, как пользователь внес все необходимые данные в поля, ему необходимо нажать на кнопку далее, которая перенаправит пользователя на следующий этап. Процесс создания новой записи, аналогичен процессу внесения изменений в объект, который будет описан ниже.

Для более детального просмотра информации об объекте используется карточка объекта (см. рис. 3.7).

Карточка объекта - это панель со сведениями об объекте, вид карточки объекта зависит от типа объекта, обычно карточка объекта включает в себя:

- информацию о объекте
- информацию о собственнике
- фотографии и документы
- заметки и комментарии для сотрудников компании

Информация о объекте подразумевает под собой раздел базы, которую риелтор должен заполнять в процессе работы, информация может быть рекламная, вымышленная и текущая. Информация о собственнике - это контакт, закрепленный за данным объектом, контактом может выступать сам риелтор если этот объект - рекламное объявление.

Фотографии могут разделяться на документы по объекту и общие фотографии. Во вкладке "Комментарии" (см. рис. 3.6) сотрудники могут делиться информацией по нюансам квартиры, особенностям характера собственника, информацией о желательном времени показа.

Для логического размещения информации в карточке используются вкладки (см. рис. 3.6) реализованные с помощью класса `tab` в библиотеке `bootstrap`. Переключение вкладок представляет собой скрытие всех вкладок и отображение только текущей активной вкладки. Активная вкладка выделяется синим цветом.

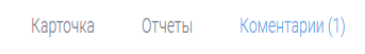


Рис. 3.6. Вкладки карточки объекта

Для переключения между основными вкладками по карточке используется панель, расположенная в левой части экрана (см. рис. 3.7). С помощью левой панели, пользователь может быстро переключаться между данными о объекте недвижимости и данными о собственнике объекта.

Вкладка **карточка** (см. рис. 3.6) отвечает за отображение полей, которые необходимы для работы риелтору. Вкладка **комментарии** (см. рис. 3.6) служит для обсуждения рабочих процессов.

The screenshot shows a web interface for a real estate object card. On the left is a sidebar with navigation links: Главная, Информация, Фотографии, and Клиент. The main content area is titled 'Карточка' and has sub-tabs for 'Отчеты' and 'Комментарии (0)'. The form includes the following fields:

- Агент:** A dropdown menu with 'Выберите' and two checkboxes: 'В рекламу' and 'В эксклюзив'.
- Вопрос, Актуальность, Объект, Цена:** A row of four fields: 'Вопрос' (dropdown), 'Дата' (text box with 'дд.мм.гггг' placeholder), 'Актуальность' (dropdown), and 'Цена от СБ' (text box).
- Вид Объекта:** A dropdown menu.
- Кол. Комнат:** A dropdown menu.
- Тип Квартиры:** A dropdown menu.
- Цена в Рекламе:** A text box.
- Договоренности с Продавцом:** A text box with placeholder 'Показы/Просмотры/Задаток/Услуги...'
- Дополнительное описание:** A large text area.
- Адрес:** A grid of dropdown menus: 'Населенный Пункт' (Белгород), 'Улица' (60 лет Октября ул.), '№ Дома/Позиции' (2), '№ Корпуса' (Выберите), '№/под./ кв./...', 'ЖК, Застройка' (Выберите), '№ Этажа' (Выберите), and 'Тип Этажа' (Выберите).
- Ориентир:** A text box.
- Инфраструктура:** A text box.

Рис. 3.7. Внешний вид карточки объекта

Переход в карточку будет осуществляться из таблицы с данными с помощью перехода по ссылке и передачи таких параметров как id объекта и модуль, в котором находится пользователь. После этого, обращаемся к базе и получаем массив с информацией об этом объекте. Вывод информации на экран будет реализован идентично с поиском объекта в базе (см. листинг 3.7), за исключением вставки информации из базы в заполняемое риелтором поле (см. листинг 3.8). Вывод информации на экран будет реализован идентично с поиском объекта в базе, за исключением вставки информации в поле (см. листинг 3.6).

Листинг 3.8 Вставка информации из базы в поле карточки

```
$object = mysql_query("SELECT * FROM `".$module."base` WHERE `id`
= '".$$id."'");
$object_str = mysql_fetch_assoc($object);
$field = mysql_query("SELECT id,name,type,listid FROM `1field`
WHERE `deleted` = '0' AND `module` = '".$$module."'");
while($field_str = mysql_fetch_assoc($field))
{
```

```
<input type="text" class="form-control" name="field"><?php echo
$field_str['id'];?>" value="<?php echo
$object_str['field'].$field_str['id']] ?>">
```

Отправка данных в базу будет осуществляться с помощью ajax-функции `AjaxFormRequest`(см. листинг 3.9) методом POST, в которую мы передаем id HTML формы, и скрипт обработчик `edit_object`. Для того чтобы обработчик мог изменить информацию в базе - передаем данные id и module, которые будут скрыты с помощью HTML. После этого необходимо получить данные из формы и при изменении информации необходимо занести данные в базу [16, С. 336].

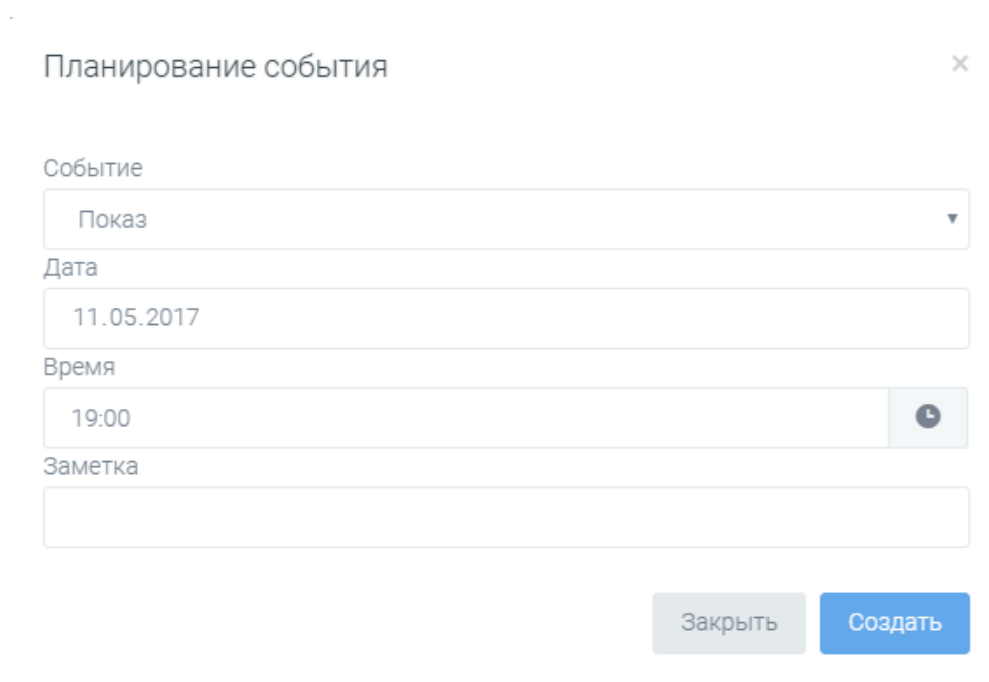
Листинг 3.9 Процесс изменения данных с помощью функции `AjaxFormRequest`

```
<a role="button"
onclick="AjaxFormRequest('result_edit_object', 'form_edit_object',
'funktions/edit_object.php')"> Сохранить объект</a>
<input type="text" class = "form-control" name="id"
style="display: none;" value="<?php echo ".$id." ?>">
<input type="text" class = "form-control" name="module"
style="display: none;" value="<?php echo ".$module." ?>">
$module = $_POST["module"];
$id = $_POST["id"];
$object = mysql_query("SELECT * FROM `".$module."base` WHERE
`id` = '".$id.'");
$object_str = mysql_fetch_assoc($object);
if $_POST["field".$field_str['id']] !=
$object_str["field".$field_str['id']])
{
$sql = mysql_query("UPDATE `".$module."base` SET
`field".$field_str['id']. "` = '".$stroka.'" WHERE `id` LIKE '".$id
.'");
}
```

Планировщик служит одним из основных инструментов в работе любого риелтора, поэтому необходима реализация такой функции в системе.

Принцип работы такой функции достаточно прост: пользователь планирует событие на определенную дату и время, а программа должна напомнить ему об этом событии.

Для разработки планировщика необходимо создать в базе данных таблицу, в которую будут записываться запланированные события. Данные будут попадать в эту таблицу через HTML форму (см. рис. 3.8), которую пользователь может отправить прямо из карточки объекта



The image shows a web form titled "Планирование события" (Event Planning) with a close button (x) in the top right corner. The form contains the following fields:

- Событие** (Event): A dropdown menu with "Показ" (Show) selected.
- Дата** (Date): A text input field containing "11.05.2017".
- Время** (Time): A time selection field showing "19:00" with a clock icon.
- Заметка** (Note): An empty text input field.

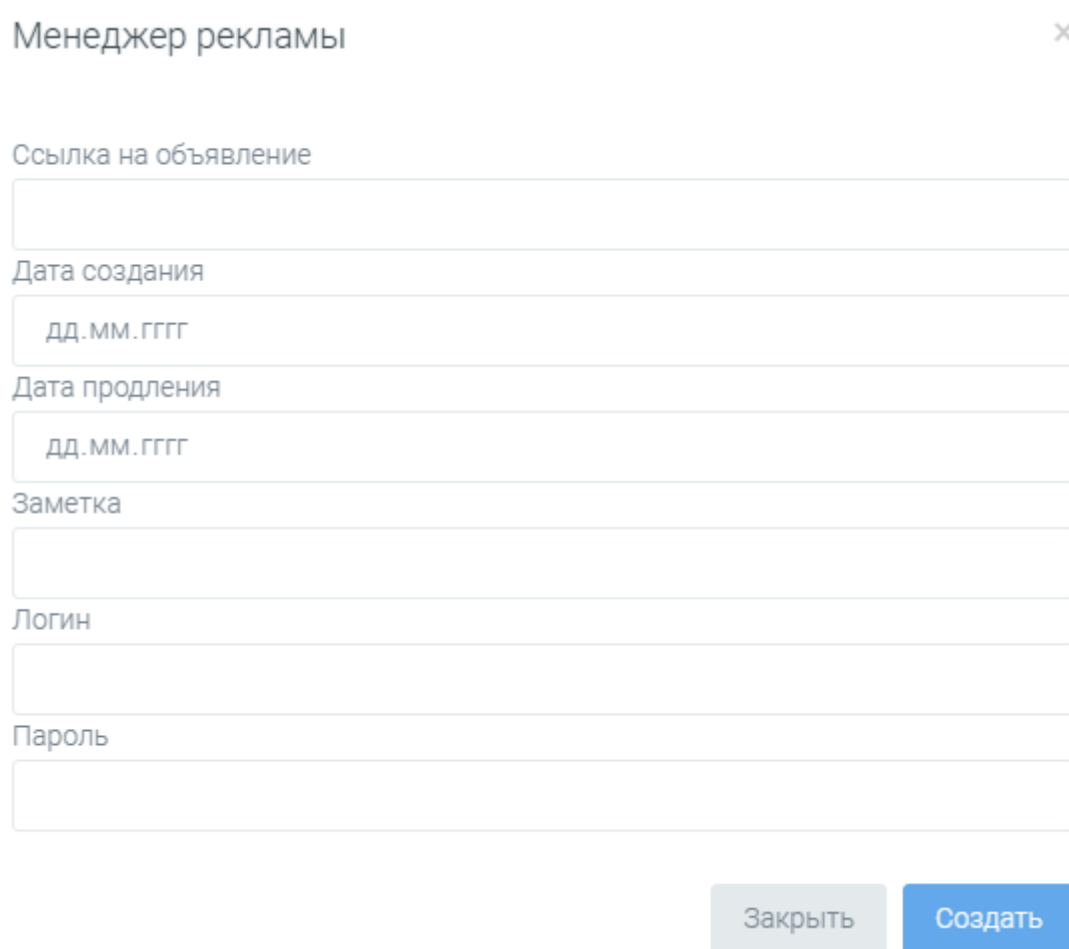
At the bottom right of the form are two buttons: "Закреть" (Close) in a light gray box and "Создать" (Create) in a blue box.

Рис. 3.8. Форма планирования события

Данные из формы будут отправляться методом POST в обработчик, так же скрытно от пользователя будут передаваться такие параметры, как id карточки и id модуля. После этого данные записываются в таблицу Planirovshik(см. рис. 2.3).

В процессе своей работы, любой агент по недвижимости сталкивается с необходимостью подачи объекта в рекламу на популярные доски объявлений, на которые запрещена массовая выгрузка XML. Такие доски как правило разрешают подавать бесплатно объявления только для частных лиц и в ограниченном количестве. Подача платной рекламы для агентств недвижимости может стать одной из самых больших статей расходов, чтобы

экономить деньги риелторы регистрируют себя, как частное лицо и подают 1-2 объявления. По истечению 30 дней, необходимо пройти процесс авторизации и продлить объявления. В связи с этим у агентов скапливаются десятки таких регистрационных данных, которые необходимо где-то хранить, а также необходимо оповещать риелтора о продлении объявлений, для таких задач необходимо создать **менеджер рекламы** (см. рис. 3.9), который будет работать по такому же принципу как и планировщик, но дополнительно будет иметь возможность хранения данных авторизации.



Менеджер рекламы x

Ссылка на объявление

Дата создания

Дата продления

Заметка

Логин

Пароль

Рис. 3.9. Форма менеджера рекламы

После нажатия кнопки "Создать", данные запаленной формы отправляются в обработчик, а после записываются в таблицу Planirovshik, структуру таблицы можно увидеть на рисунке 3.

Вкладка **комментарии** (см. рис. 3.10) служит для общения сотрудников и создания заметок по конкретному объекту, благодаря этому сотрудники могут не запоминать все тонкости и нюансы этого объекта, а просто записать их в комментарий. Для отправки комментария пользователь должен заполнить всего одно текстовое поле и нажать кнопку "Отправить". Отправка происходит с помощью аjax-функции `AjaxFormRequest` (см. листинг 3.9), что позволяет делать запись в базу без перезагрузки страницы. Для того чтобы пользователь увидел оставленный собой комментарий и не перезагружал страницу, после отправки формы вызывается javascript функции. `load`, которая перезагружает только раздел комментариев.

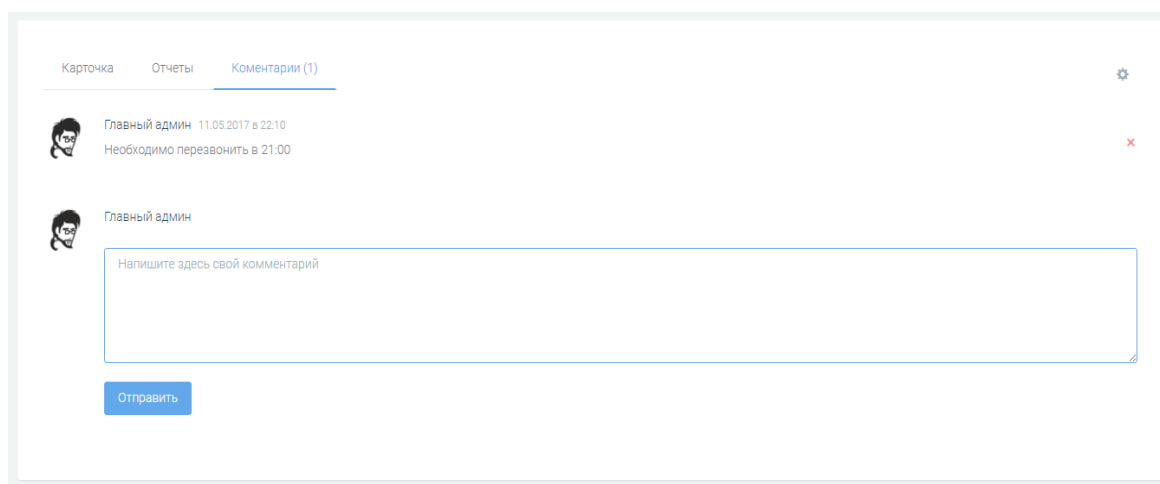


Рис. 3.10. Раздел комментариев

После полной загрузки вкладки комментариев, необходимо реализовать функцию удаления своих комментариев. При нажатии на красный крестик в правом углу комментария, будет отправлена аjax форма и вызвана функция. `load`.

3.3 Разработка дополнительного функционала

Важной функцией для работы любой организации являются сообщения между сотрудниками, многие сотрудники агентств недвижимости работают на дому, а также часто ездят на встречи с клиентами.

Помимо самого чата этого для офиса реализован целый ряд **дополнительных возможностей**. Первая из них – **система оповещений**. С ее помощью можно передать всем или только некоторым пользователям определенное сообщение и затребовать у них подтверждение прочтения. Это позволяет рассылать приказы для ознакомления, оповещения о проведении собраний и пр. Причем сотрудники уже не смогут сказать, что они не увидели или не прочитали сообщение.

Для реализации чата будет создана таблица, которая будет хранить сообщения между сотрудниками. Для отправки сообщения пользователю необходимо будет выбрать собеседника из сотрудников компании, заполнить сообщение и отправить форму (см. рис. 3.11). Отправка формы осуществляется аналогично отправки комментариев к объектам - методом POST, с использованием аjax-функции `AjaxFormRequest` (см. листинг 3.9) и функции `JavaScript.load`. После отправки сообщения, получателю приходит уведомление о получении нового сообщения.

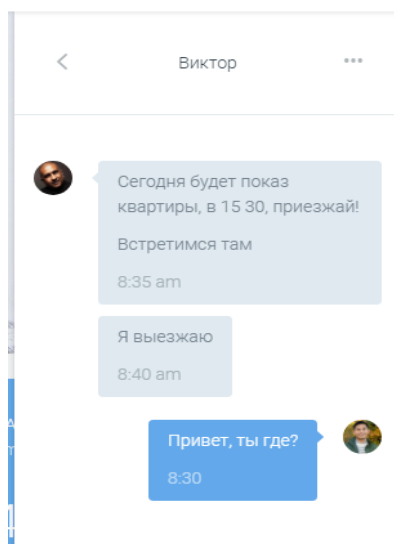


Рис. 3.11. Раздел комментариев

Каждый пользователь системы может настроить свои персональные данные, которые будут использоваться в списке сотрудников и данных при отправке рекламы. Изменить информацию о себе пользователь может в разделе "Мои настройки" (см. рис. 3.12). Данные личной переписки доступны

только пользователям системы, пользователи с правами администратора, также не имеют доступ к личной переписки сотрудников.

The screenshot displays a user profile interface. On the left, there is a profile card for 'Главный админ' (Main Admin) with a placeholder image, phone number 'Тел. 89192271212', and email 'Email MyProchta@mail.ru'. The main content area shows a form for editing personal information. The form has tabs for 'Личная информация', 'Фото', and 'Активность'. The 'Личная информация' tab is active. The form fields are: FIO (Главный админ), Login (admin), Password (admin), Main Phone (89192271212), Additional Phone (Телефон), Email (MyProchta@mail.ru), and Skype (Скайп). A blue 'Сохранить' (Save) button is located at the bottom left of the form.

Рис. 3.12. Персональный профиль пользователя

Заполнив нужные данные, пользователь отправляет форму обработчику методом POST. Блок расположенный с лева от формы служит для предварительного просмотра новой информации.

3.4 Тестирование автоматизированной системы

У автоматизированной системы может быть отличный дизайн и интерфейс, но внешний вид и функциональность не будут иметь никакого значения, если она долго загружается. Современные пользователи все более нетерпеливы, и уровень нетерпения будет расти вместе с увеличением доли мобильных устройств. Задача оптимизации скорости загрузки наиболее актуальна для автоматизированных систем с большим объемом данных.

По данным исследования американской компании Akamai:

- 47% пользователей ожидают, что веб-страница загрузится в течение 2 секунд;
- 40% посетителей могут уйти с сайта, который грузится более 3 секунд;

- 52% утверждают, что быстрая загрузка влияет на их лояльность;
- 3 секунды ожидания уменьшают лояльность клиентов примерно на 16%

В современных офисных центрах провайдеры предлагают стандартные тарифы со скоростью от 1 мб\с до 10 мб\с для юридических лиц. Так как система имеет возможность работы с помощью мобильного или планшетного устройства будут учтены скорости 3G и 4G соединений.

Замеры скорости загрузки страницы будут выполняться с помощью браузера Google chrome и инструмента network - доступном в режиме разработчика. Для удобства данные представлены в виде таблицы 3.1.

Таблица 3.1

Замеры скорости загрузки страниц системы

Скорость интернета	Таблица объектов 3500	Карточка объекта	Любая страница другая страница
2 мб\с	0,916	1,73	0,541
4 мб\с	0,835	1,47	0,523
5 мб\с	0,811	1,41	0,521
8 мб\с	0,775	1,34	0,499
10 мб\с	0,741	1,32	0,492
3G	0,875	1,51	0,578
4G	0,764	1,42	0,513
50 мб\с	0,687	1,12	0,487

Как видно из сравнительной таблицы автоматизированная система полностью соответствует современным требованиям, скорость загрузки страниц в большинстве случаев не превышает 1 секунды.

ЗАКЛЮЧЕНИЕ

В процессе выполнения дипломной работы мною были проанализированы существующие автоматизированные системы для агентств недвижимости, были выявлены их недостатки и рассмотрены основные функции. В процессе создания системы были проведены опросы агентств недвижимости, которые выявили серьезную потребность в наличии специализированной автоматизированной системы. В процессе опроса был сформирован список необходимого функционала системы, который будет учитывать особенности работы риелторов.

Исходя из предполагаемых нагрузок и объема обрабатываемой информации для корректной работы системы был арендован веб-сервер, со всем необходимым программным обеспечением. В процессе написания дипломной работы я обосновал выбор СУБД и приложения по администрированию баз данных на сервере. После чего была разработана модель базы данных и архитектура системы, для этого я использовал методологию IDEF3.

При разработке автоматизированной системы, большой уклон делался на оптимизацию и быстродействие системы, для этого было проведено множество операций по оптимизации программного кода. Для выявления проблем, связанных со скоростью работы программы, были проведены множественные тесты, как с мобильного, так и с кабельного подключения к интернету. Помимо скорости не мало внимания было уделено безопасности системы, как по хранению, так и по предоставлению доступов пользователям системы.

Руководствуясь анализом недостатков существующих автоматизированных систем были предопределены необходимые функции, которые легли в основу интерфейса программы, позволяющего

пользователям качественно и максимально быстро получать данные и вести работу с базой. После этого была осуществлена программная реализация автоматизированной системы с учетом требований всех агентств недвижимости.

Таким образом, поставленные задачи выполнены полностью.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Корджева, Елена Сделки с недвижимостью / Елена Корджева. - М.: Написано пером, 2016. - 390 с.
2. Данько, Т.П. Системы управления эффективностью бизнеса: Учебное пособие / С.Н. Брускин, Т.П. Данько; Под науч. ред. Н.М. Абдикеев, О.В. Китова. - М.: ИНФРА-М, 2012. - 282 с.
3. Попов В.М. и др. Глобальный бизнес и информационные технологии. Современная практика и рекомендации / В.М. Попов, Р.А. Маршавин, С.И. Лапунов; Под ред. В.М. Попова. – М.: Финансы и статистика, 2001. – 272 с.:
4. Голубков, Е.П. Основы маркетинга / Е. П. Голубков. – М.: Фин-Пресс, 2003. – 688 с.
5. Медведева, Е.В. Рекламная коммуникация / Е.В. Медведева. – М.: Едиториал УРСС, 2003. – 280 с.
6. Губанов В.А., Захаров В.В., Коваленко А.Н. Введение в системный анализ; Учебное пособие / Под ред. Л.А. Петросяна. - Л.: ЛГУ, 1988. - 232 с.
7. Багиев, Г.Л. Маркетинг: Учебник для вузов. 3-е изд. / Г. Л. Багиев, В. М. Тарасевич, Х. Анн. – СПб.: Питер, 2007. – 736с.
8. Менеджмент процессов / Под ред. Й. Беккера, Л. Вилкова, В. Таратухина, М. Кугелера, М. Роземанна; [пер. с нем.]. - М.: Эксмо, 2007. - 384 с. - (Качественный менеджмент).
9. Скляр, Дэвид РНР. Рецепты программирования / Дэвид Скляр , Адам Трахтенберг. - М.: Питер, 2015. - 784 с.
10. Тарасов, С. В. СУБД для программиста. Базы данных изнутри / С.В. Тарасов. - М.: Соломон, 2015. - 320 с.
11. Филиппов, В. А. Многозначные СУБД и XML базы данных / В.А. Филиппов. - М.: Ленанд, 2011. - 146 с.

12. Стивен Хольцнер. PHP в примерах. / Стивен Хольцнер. М.: 000 «Бином-Пресс», 2007 г. Пер. с англ. 352 с

13. Леонтьев, Борис PHP 5.0 для начинающих, или как создать динамический WEB-сайт / Борис Леонтьев. - М.: Новый издательский дом, 2005. - 176 с.

14. Шкрыль, А. PHP - это просто. Программируем для Web-сайта / А. Шкрыль. - М.: БХВ-Петербург, 2006. - 368 с.

15. Ларри Ульман. Ульман Л. Основы программирования на PHP:/Ларри Ульман. Пер. с англ. -М.: ДМК Пресс, 2001. -288 с.: ил. (Самоучитель).

16. Дари, Кристиан AJAX и PHP. Разработка динамических веб-приложений / Кристиан Дари и др. - М.: Символ-плюс, 2009. - 336 с.

ПРИЛОЖЕНИЕ

Листинг файла dbcontroller.php

```

<?php
# Запуск сессии
session_start();
# Служит для отладки, показывает все ошибки, предупреждения и т.д.
error_reporting(E_ALL);
# В этом массиве далее мы будем хранить сообщения системы, т.е. ошибки.
$message=array();
class DBController {
    private $host      = "localhost";
    private $user      = "user1192171";
    private $password  = "19934030aA";
    private $database  = "crm";

    function __construct() {
        $conn = $this->connectDB();
        if(!empty($conn)) {
            $this->selectDB($conn);
        }
    }
    function connectDB() {
        $conn = mysql_pconnect($this->host,$this->user,$this->password);
        return $conn;
    }

    function selectDB($conn) {
        mysql_select_db($this->database,$conn);
        mysql_query('SET NAMES utf8');
    }

    function runQuery($query) {
        $result = mysql_query($query);
        while($row=mysql_fetch_assoc($result)) {
            $resultset[] = $row;
        }
        if(!empty($resultset)){
            return $resultset;}else{
            printf('');
        }
    }
    function numRows($query) {
        $result = mysql_query($query);
        $rowcount = mysql_num_rows($result);
        return $rowcount;
    }
}
//-----
function checkAccses($module){
    $db_handle = new DBController();

```

```

$query2      = mysql_query("SELECT * FROM `role_funktions` WHERE `role` =
'".$_SESSION["role"]."' AND functionid='".$_.$module."'");
$row2       = mysql_num_rows($query2);
// echo $query3; echo $row2;
if ($row2 == 0) {
//      echo      '<META      HTTP-EQUIV="Refresh"      CONTENT="0;
URL=/templates/admin/errorAccses.php">';
}
}
function spy($module,$do){
$db_handle = new DBController();
$data = date("y-m-d H:i:s");
$history = mysql_query("INSERT INTO `spy` (userid,module,do,time)
values('".$_SESSION["id_user"]."', '".$_.$module."', '".$_.$do."', '".$_.$data.'')");
}

function displayErrors($messages) {
    foreach($messages as $msg){ print("$msg\n"); }
}

function checkLoggedIn($status){
    switch($status){
        case "yes":
            if(!isset($_SESSION["loggedIn"])){
                echo '<META HTTP-EQUIV="Refresh" CONTENT="0; URL=/index.php">';
                exit;
            }
            break;
        case "no":
            if(isset($_SESSION["loggedIn"]) && $_SESSION["loggedIn"] === true ){

                $db_handle = new DBController();
                $data = date("y-m-d H:i:s");
                $history = mysql_query("INSERT INTO `spy` (userid,module, time)
values('".$_SESSION["id_user"]."', 'Авторизация', '".$_.$data.'')");
                echo      '<META      HTTP-EQUIV="Refresh"      CONTENT="0;
URL=/templates/admin/index.php">';
            }
            break;
    }
    return true;
}

function checkPass($login, $password) {
    if ((empty($login)) or (empty($password))) {return false;} else {
        $db_handle = new DBController();
        $query      = "SELECT * FROM `users` WHERE `login` = '".$_.$login."' and
`password` = '".$_.$password.'";
        $result     = mysql_query($query);
        if(mysql_num_rows($result)==1) {
            $row=mysql_fetch_array($result);
            return $row;
        }
    }
    return false;
}

function cleanMemberSession($login, $password) {

```



```

    $query          = "SELECT * FROM `users` WHERE `login` = '". $login.'" and
`password` = '". $password.'"";
    $result         = mysql_query($query);
    $row1           = mysql_fetch_array($result);
    $id_user        = $row1['id'];
    $_SESSION["id_user"]      = $id_user;
    $_SESSION["first_name"]   = $row1['first_name'];
    $_SESSION["last_name"]    = $row1['last_name'];
    $_SESSION["login"]        = $login;
    $_SESSION["password"]     = $password;
    $sid                 = session_id();
    $_SESSION["sid"]         = $sid;
    $_SESSION["loggedIn"]    = true;
    $_SESSION["title"]       = $row1['title'];
    $_SESSION["phone_home"]  = $row1['phone_home'];
    $_SESSION["companyid"]   = $row1['companyid'];
    mysql_query("UPDATE `vtiger_users` set `email2` = '". $sid.'" WHERE
`id`='". $id_user);
    echo '<META HTTP-EQUIV="Refresh" CONTENT="0; URL=/">';
}

function flushMemberSession() {
    $db_handle = new DBController();
    $data = date("y-m-d H:i:s");
    $history = mysql_query("INSERT INTO `spy` (userid,module, time)
values('".$_SESSION["id_user"]."', 'Выход', '". $data.'")");
    unset($_SESSION["id_user"]);
    unset($_SESSION["first_name"]);
    unset($_SESSION["last_name"]);
    unset($_SESSION["login"]);
    unset($_SESSION["password"]);
    unset($_SESSION["sid"]);
    unset($_SESSION["loggedIn"]);
    unset($_SESSION["title"]);
    unset($_SESSION["phone_home"]);
    unset($_SESSION["companyid"]);
    session_destroy();
    echo '<META HTTP-EQUIV="Refresh" CONTENT="0;/">';
    return true;
}

//-----
?>

```

Листинг файла pagination.class.php

```

<?php
//echo "Цена мин ". $price_min."";
class PerPage {
    public $perpage;

    function __construct() {
        $this->perpage = 50;
    }

    function perpage($count,$href) {
        $output = '';
        $rooms = htmlspecialchars($_GET["rooms"]);
        $square = htmlspecialchars($_GET["square"]);

```

```

$adres = htmlspecialchars($_GET["adres"]);
$prise = htmlspecialchars($_GET["prise"]);
$floor = htmlspecialchars($_GET["floor"]);
$phone1 = htmlspecialchars($_GET["phone1"]);
$table = htmlspecialchars($_GET["table"]);
$type = htmlspecialchars($_GET["type"]);
$way = htmlspecialchars($_GET["way"]);
if (empty($rooms)) {} else {$stroka .= "&rooms=$rooms";}
if (empty($square)) {} else {$stroka .= "&square=$square";}
if (empty($adres)) {} else {$stroka .= "&adres=$adres";}
if (empty($floor)) {} else {$stroka .= "&floor=$floor";}
if (empty($prise)) {} else {$stroka .= "&prise=$prise";}
if (empty($phone1)) {} else {$stroka .= "&phone1=$phone1";}
$stroka .= "&type=$type&way=$way";

        if(!isset($_GET["page"])) $_GET["page"] = 1;
        if($this->perpage != 0)
            $pages = ceil($count/$this->perpage);
        if($pages>1) {
            if(($_GET["page"]-3)>0) {
                if($_GET["page"] == 1)
                {
                    // $output = $output . '<button id=1
type="button" class="btn btn-default btn-sm active">1</button>';
                    $output = $output. "<a href='". $href . $i.
$stroka."&table=".$table."'>". '<span id="'. $i.' class="btn btn-default btn-
sm">'. $i.'</span>'. "</a>";
                }
                else
                    // $output = $output . '<input type="button"
class="btn btn-default btn-sm" onclick="getResult(\'' . $href . '1\')'
value=1 />';
                    $output = $output. "<a href='". $href . $i.
$stroka."&table=".$table."'>". '<span id=1 class="btn btn-default btn-
sm">1</span>'. "</a>";
                }
                if(($_GET["page"]-3)>1) {
                    $output = $output . "<a href='". $href . "'>".
'<input type="button" class="btn btn-default btn-sm disabled" value="..."
/>';
                }
            }

            for($i=(($_GET["page"]-2); $i<=(($_GET["page"]+2); $i++)) {
                if($i<1) continue;
                if($i>$pages) break;
                if($_GET["page"] == $i)
                {
                    // $output = $output . '<span id="'. $i.'
class="btn btn-default btn-sm active">'. $i.'</span>';
                    $output = $output. "<a href='". $href . $i.
$stroka."&table=".$table."'>". '<span id="'. $i.' class="btn btn-default btn-sm
active" >'. $i.'</span>'. "</a>";
                }
                else {
                    // $output = $output . '<input type="button"
class="btn btn-default btn-sm" onclick="getResult(\'' . $href . $i . '\')'
value=' . $i . ' />';
                    $output = $output. "<a href='". $href . $i.
$stroka."&table=".$table."'>". '<span id="'. $i.' class="btn btn-default btn-
sm">'. $i.'</span>'. "</a>";
                }
            }
        }
    }
}

```

```

        if(($pages-($_GET["page"]+2))>1) {
            //$output = $output . '<input type="button" class="btn
            btn-default btn-sm disabled" value="..." />';
            $output = $output. "<a href='". $href .$.i.
            $stroka."&table=".$table.">". '<input type="button" class="btn btn-default
            btn-sm disabled" value="..." />';
        }
        if(($pages-($_GET["page"]+2))>0) {
            if($_GET["page"] == $pages)
                //$output = $output . '<span id=' . ($pages) .'
            class="btn btn-default btn-sm active">' . ($pages) . '</span>';
            $output = $output. "<a href='". $href .$.i.
            $stroka."&table=".$table.">". '<span id='.$pages.' class="btn btn-default
            btn-sm">' . $pages.'</span>'. "</a>";
            else
                //$output = $output . '<input type="button"
            class="btn btn-default btn-sm" onclick="getresult(\'' . $href . ($pages)
            .'\')"' value=' . $pages . ' />';
            $output = $output. "<a href='". $href .$.i.
            $stroka."&table=".$table.">". '<span id='.$pages.' class="btn btn-default
            btn-sm">' . $pages.'</span>'. "</a>";
        }
    }
    return $output;
}
}
?>

```

Листинг алгоритма генерации карточки

```

<form method="post" action="" id="edit_adres">
<input type="text" class = "form-control" name="cartid" style="display:
none;" value="<?php echo "".$id."" ?>">
<?php
$module = htmlspecialchars($_GET["module"]);
?>
<input type="text" name="module" style="display: none;" value="<?php echo
"".$module."" ?>">
<?php

$fieldset = mysql_query("SELECT id,name,col FROM `fieldset` WHERE `moduleid`
= '". $module.'" AND fieldsetid=0 ORDER BY `sortid` ASC "); // находим 1 ур.вл
while($fieldset_str = mysql_fetch_assoc($fieldset))
{
?>
<fieldset class="col-md-12">
<legend style="font-size: 17px;color: #01305d;margin-bottom: 4px;"><?php echo
$fieldset_str['name'];?> </legend>
<?php
$count_on_row = $fieldset_str['col'];
$count_now = 0;
$field = mysql_query("SELECT id,name,type,listid FROM `lfield` WHERE
`deleted` = '0' AND `module` = '". $module.'" AND `fieldset` =
'".$fieldset_str['id'].' ORDER BY `row` ASC ");
while($field_str = mysql_fetch_assoc($field))
{
if ((( $count_now % $count_on_row)) == 0)
{
?>

```

```

<div class="col-md-12" style="padding-bottom: 10px;">
<?php
}
$count_now ++;
?>
<div class="col-md-<?php echo 12/$fieldset_str['col'] ?>" style="padding-
bottom: 0px;">
<?php echo $field_str['name'];?>
<?php
if ($field_str['type']=="text")
{
?>
<input type="text" class = "form-control" name="field<?php echo
$field_str['id'];?>" value="<?php echo $object_str['field'].$field_str['id']
?>">
<?php
}
?>

<?php
if ($field_str['type']=="textarea")
{
?>
<textarea rows="3" class = "form-control" cols="160" name="field<?php echo
$field_str['id'];?>"><?php echo $object_str['field'].$field_str['id']
?></textarea>
<?php
}
?>

<?php
if ($field_str['type']=="date")
{
?>
<input type="date" class = "form-control" name="field<?php echo
$field_str['id'];?>" value="<?php echo $object_str['field'].$field_str['id']
?>">
<?php
}
?>

<?php
if ($field_str['type']=="select")
{
$listcounter = 21;
?>
<select <?php if ($listcounter>20) { ?> data-plugin="select2" <?php } ?>
class="form-control" name="field<?php echo ".$field_str["id"]." ?>">
<option value="">Выберите</option>
<?php
$list_count = mb_substr_count($field_str['listid'], ",");

$field_values_count =
mb_substr_count($object_str['field'].$field_str['id'],' |##| ');
$field_count = 0;
$field_value = explode(' |##| ', $object_str['field'].$field_str['id']);
$list_id = explode(",", $field_str['listid']);

$y = 0;

while ($y<=$list_count)

```

```

{
$users = mysql_query("SELECT id,name FROM `list".$list_id[$y]."` WHERE
(`deleted` = '0' OR `deleted` IS NULL) ORDER BY `sortid` ASC ");
while ($users_str = mysql_fetch_assoc($users))
{
if (($object_str['field'].$field_str['id'])==$users_str['name']) AND
($object_str['field'].$field_str['id']!='')
{
?>
<option selected value="<?php echo $users_str['name'] ?>"><?php echo
$users_str['name'] ?></option>
<?php
} else {
?>
<option value="<?php echo $users_str['name'] ?>"><?php echo
$users_str['name']; ?></option>
<?php
}
}
$y++;
}
?>
</select>
<?php
}
?>

<?php
if ($field_str['type']=="select_multi")
{
?>
<select class="form-control" data-plugin="select2" multiple
name="field<?php echo "".$field_str["id"]." ?>[]" class="full-width">
<?php
$list_count = mb_substr_count($field_str['listid'], ",");

$field_values_count =
mb_substr_count($object_str['field'].$field_str['id'],' |##| ');
$field_count = 0;
$field_value = explode(' |##| ', $object_str['field'].$field_str['id']);
$list_id = explode(",", $field_str['listid']);

$i=0;
while ($i<=$field_values_count)
{
$y = 0;
while ($y<=$list_count)
{
$users = mysql_query("SELECT id,name FROM `list".$list_id[$y]."` WHERE
(`deleted` = '0' OR `deleted` IS NULL) ORDER BY `sortid` ASC ");
while ($users_str = mysql_fetch_assoc($users))
{

if (($field_value[$i]==$users_str['name']) AND ($field_value[$i]!=''))
{
?>
<option selected value="<?php echo $users_str['name'] ?>"><?php echo
$users_str['name'] ?></option>
<?php
} else {

```

```

?>
<option value="<?php echo $users_str['name'] ?>"><?php echo
$users_str['name']; ?></option>
<?php
}
}
$y++;
}
$i++;
}
?>
</select>
<?php
}
?>
</div>
<?php
if ((( $count_now % $count_on_row) == 0)
{
?>
</div>
<?php
}
}
?>

</fieldset>
<?php
}
?>

<div id="result_div_form_sob_add3"></div>
</form>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-default
margin-0" data-dismiss="modal">Отмена</button>
<button type="button" class="btn btn-primary"
onclick="AjaxFormRequest('result_div_form_sob_add3',
'funktions/edit_adres.php')">Сохранить</button>
'edit_adres',
</div>
</div>
</div>
</div>
<div id="commentaries">
<?php
$module = htmlspecialchars($_GET["module"]);
$result2 = mysql_query("SELECT id,userid,text,data FROM `object_comments`
WHERE (moduleid = '". $module.'" AND cartid = '". $id.'" ) ORDER BY `data`
ASC");
while ($result2_str = mysql_fetch_assoc($result2))
{
$user_com = mysql_query("SELECT id,name FROM `users` WHERE (id =
'". $result2_str['userid']."' )");
$user_com_str = mysql_fetch_assoc($user_com);
?>

```

Листинг алгоритма вывода комментариев

```

<div class="comment media">
<div class="media-left">
<a class="avatar avatar-lg" href="javascript:void(0)">

```

```

    <?php
    $fotog = mysql_query("SELECT * FROM `user_photo_base` WHERE `userid` LIKE
    '". $result2_str['userid'] ."' LIMIT 1");
    if (mysql_num_rows($fotog)==0)
    { $avatar = '/crm2/html/photo/users_avatars/noavatar.png';}
    else { $fotog_row = mysql_fetch_assoc($fotog);
    $avatar = '/crm2/html/photo/users_avatars/'. $result2_str['userid'] .'/'.$.
    $fotog_row['filename'] .'; }
    ?>

        
        </a>
    </div>
    <div class="comment-body media-body">
        <a class="comment-author" href="javascript:void(0)"><?php
    echo "'".$user_com_str['name']."' ?></a>
        <div class="comment-meta">
            <span class="date"><?php echo date("d.m.Y B
    H:i",strtotime($result2_str['data'])); ?></span>
        </div>
    <?php
    if ($result2_str['userid']==$_SESSION['id_user'])
    {
    ?>

        <div class="comment-meta" style="float: right;">
        <form class="comment-reply" id="del_coment<?php echo
    $result2_str['id'] ?>" action="#" method="post">
    <input type="text" class = "form-control" name="id" style="display: none;"
    value="<?php echo "'".$result2_str['id']."' ?>">
        <span class="date"><i style="cursor: pointer;color:
    #fa8e8e;font-size: 15px;" class="icon wb-close-mini" aria-hidden="true"
    onclick="AjaxFormRequest('result_add_coment', 'del_coment<?php echo
    $result2_str['id'] ?>', 'funktions/del_coment.php');fync();" class="btn btn-
    primary"></i></span>
    </form>

            </div>
    <?php
    }
    ?>

        <div class="comment-content">

            <p><?php echo "'".$result2_str['text']."' ?></p>
        </div>

    </div>
    </div>
    <?php
    }
    ?>

        </div>

        <div class="comment media">
            <div class="media-left">
                <a class="avatar avatar-lg" href="javascript:void(0)">
                    <?php
                    $fotog = mysql_query("SELECT * FROM `user_photo_base` WHERE `userid` LIKE
                    '". $_SESSION['id_user'] ."' LIMIT 1");
                    if (mysql_num_rows($fotog)==0)
                    { $avatar = '/crm2/html/photo/users_avatars/noavatar.png';}
                    else { $fotog_row = mysql_fetch_assoc($fotog);

```

```

$avatar = '/crm2/html/photo/users_avatars/'. $_SESSION['id_user'] .'/'.
$fotog_row['filename'] .'; }
?>
        
        </a>
    </div>
    <div class="comment-body media-body">
        <a class="comment-author" href="javascript:void(0)"><?php
echo $_SESSION['name'] ?></a>
        <form class="comment-reply" id="add_coment" action="#"
method="post">
<input type="text" class = "form-control" name="id" style="display: none;"
value="<?php echo "$id." ?>">
<input type="text" class = "form-control" name="module" style="display:
none;" value="<?php echo "$module." ?>">
        <div class="form-group">
            <textarea class="form-control" name="text" rows="5"
placeholder="Напишите здесь свой комментарий"></textarea>
        </div>
        <div class="form-group">
            <a
                onclick="AjaxFormRequest('result_add_coment',
'add_coment', 'funktions/add_coment.php');fync();" class="btn btn-
primary">Отправить</a>
        </div>
        <div id="result_add_coment"></div>
        </form>
    </div>
</div>
<script type="text/javascript">
function fync() {
$("#commentaries").load("object.php?module=<?php echo "$module."
?>&id=<?php echo "$id." ?> #commentaries", function() {
});
$("#commentaries").load("object.php?module=<?php echo "$module."
?>&id=<?php echo "$id." ?> #commentaries", function() {
});
};
</script>

```

Реализация алгоритма .load

```

function fync3() {
$("#tab3").load("object.php?module=<?php echo "$module." ?>&id=<?php echo
"$id." ?> #klient_card", function() {
});
$("#tab3").load("object.php?module=<?php echo "$module." ?>&id=<?php echo
"$id." ?> #klient_card", function() {
});
document.getElementsByClassName('modal-backdrop fade in').remove();
};
</script>

<script>
$(function(){
    //код Javascript, предназначенный для того активировать вкладку по хэшу
    (#tab_tab1) в адресной строке
    var hash = document.location.hash;
    var prefix = "tab_";
    if (hash) {
        $('nav-tabs a[href="' + hash.replace(prefix, "") + '"]').tab('show');
    }
    //Изменить хэш страницы при открытии вкладки

```



```

$('.nav-tabs a').on('shown.bs.tab', function(e) {
    window.location.hash = e.target.hash.replace("#", "#" + prefix);
});
});
</script>

```

Листинг алгоритма создания фильтров

```

<div class="modal fade slide-up"
id="exampleNiftySlideFromRight<?php echo $result_str['id']; ?>" tabindex="-1"
role="dialog" aria-hidden="false">
    <div class="modal-dialog" style="width: 70%;">
        <div class="modal-content-wrapper">
            <div class="modal-content">
                <div class="modal-body m-t-20">
                    <h4 class="no-margin p-b-10">Редактирование фильтра <?php
echo $result_str['name']; ?></h4>
<form method="post" action="" id="form_idfilter<?php echo $result_str['id'];
?>">
<input type="text" class = "form-control" name="userid" style="display:
none;" value="<?php echo "'.$_SESSION["id_user"]."' ?>">
<input type="text" class = "form-control" name="companyid" style="display:
none;" value="<?php echo "'.$_SESSION["companyid"]."' ?>">
<input type="text" class = "form-control" name="filterid" style="display:
none;" value="<?php echo "'. $result_str["id"]."' ?>">
<?php
$module = htmlspecialchars($_GET["module"]);
?>

<input type="text" name="module" style="display: none;" value="<?php echo
"'".$_SESSION["module"]."' ?>">

<div class="col-md-12">
<div class="col-md-8">
Название фильтра
<input type="text" class = "form-control" name="name" value="<?php echo
$result_str['name']; ?>" required>
</div>
<div class="col-md-4">
    <div class="checkbox-custom checkbox-primary">
        <input type="checkbox" id="defolt<?php echo
$result_str['id']; ?>" name = "defolt" <? if ($result_str['defolt']==1) { ?>
checked <? } ?>/>
        <label for="defolt<?php echo $result_str['id']; ?>">По
умолчанию</label>
    </div>
</div>
</div>
<div class="col-md-12">
<fieldset class="menu-items">
<legend style="font-size: 21px;color: #263238;margin-bottom:
9px;">Поля</legend>
<?php
$module = htmlspecialchars($_GET["module"]);
if ($module == 13){
$result2 = mysql_query("SELECT id,name FROM `lfield` WHERE deleted = '0'
AND (module = '".$_SESSION["module"]."' OR module = '13') ORDER BY `module` ASC,
`fieldset` ASC , `id` ASC");
} else {
$result2 = mysql_query("SELECT id,name FROM `lfield` WHERE deleted = '0'
AND (module = '".$_SESSION["module"]."' OR module = '13')
AND (`id` != '136' AND `id` != '137' AND `id` != '138' AND `id` != '139' AND
`id` != '140' AND `id` != '141' AND `id` != '142' )

```

```

ORDER BY `module` ASC, `fieldset` ASC , `id` ASC");
}
while ($field_str = mysql_fetch_assoc($result2))
{
    $filter_chek = mysql_query("SELECT id FROM `filter_list` WHERE
filter_id = '". $result_str['id']. "' AND fieldid = '". $field_str['id']. "'");
?>
<div class="col-md-4">
    <div class="checkbox-custom checkbox-primary">
        <input type="checkbox" id="inputCheckedd<?php echo
$field_str['id'] ?>" name = "field<?php echo $field_str['id'] ?>" <?php if
(mysql_num_rows($filter_chek)>0) { ?> checked <?php } ?> />
        <label for="inputCheckedd<?php echo $field_str['id']
?>"><?php echo $field_str['name'] ?></label>
    </div>
</div>
<?php
}
?>
</fieldset>
</div>
<div id="result_div_filter<?php echo $result_str['id']; ?>"></div>

</form>
        <button type="button" class="btn btn-default
margin-0" data-dismiss="modal">Закрыть</button>
        <button type="button" class="btn btn-primary"
onclick="AjaxFormRequest('result_div_filter<?php echo $result_str['id']; ?>',
'form_idfilter<?php echo $result_str['id']; ?>',
'funktions/edit_filter.php')">Сохранить</button>
    </div>
</div>
</div>
<!-- /.modal-content -->

</div>
</div>
<?php
}
?>

```

Листинг алгоритма поиска по базе данных

```

<?php
$poisk = '';
$fieldset = mysql_query("SELECT id,name,col FROM `fieldset` WHERE `moduleid`
= '". $module. "' AND fieldsetid=0 ORDER BY `sortid` ASC "); // находим 1
ур.вл
while($fieldset_str = mysql_fetch_assoc($fieldset))
{
?>
<fieldset class="col-md-12">
<legend><?php echo $fieldset_str['name'];?></legend>
<?
$fieldset2 = mysql_query("SELECT id,name,col FROM `fieldset` WHERE `moduleid`
= '". $module. "' AND fieldsetid='". $fieldset_str['id']. "' ORDER BY `sortid`
ASC "); // находим 1 ур.вл

while($fieldset2_str = mysql_fetch_assoc($fieldset2))
{

?>

```

```

<fieldset class="col-md-6" style="border: 1px ridge #dadada;">
<legend><?php echo $fieldset2_str['name'];?></legend>
<?php
$field = mysql_query("SELECT id,name,type,module,listid FROM `lfield` WHERE
`deleted` = '0' AND `module` = '". $module.'" AND `fieldset` =
'".$fieldset2_str['id']."' ORDER BY `row` ASC ");
$count_on_row = $fieldset2_str['col'];
$count_now = 0;
while($field_str = mysql_fetch_assoc($field))
{
if (($count_now % $count_on_row) == 0)
{
?>
<div class="col-md-12" style="padding-bottom: 10px;">
<?php
}
$count_now ++;
?>
<div class="col-md-<?php echo 12/$fieldset2_str['col'] ?>" style="padding-
bottom: 0px;">
<?php echo $field_str['name']; ?>

<?php
if ($field_str['type']=="text")
{
if (!empty($_GET['field'].$field_str['id'])) { $poisk .= " AND
`".$field_str['module']."base`.`field".$field_str['id']."' =
'".$_GET['field'].$field_str['id']."'"; }
?>
<input type="text" class = "form-control" name="field<?php echo
$field_str['id'];?>" value="<?php echo $_GET['field'].$field_str['id'] ?>">
<?php
}
?>

<?php
if ($field_str['type']=="textarea")
{
if (!empty($_GET['field'].$field_str['id'])) { $poisk .= " AND
`".$field_str['module']."base`.`field".$field_str['id']."' =
'".$_GET['field'].$field_str['id']."'"; }
?>
<textarea rows="3" class = "form-control" cols="160" name="field<?php echo
$field_str['id'];?>"><?php echo $_GET['field'].$field_str['id'] ?></textarea>
<?php
}
?>

<?php
if ($field_str['type']=="date")
{
if (!empty($_GET['field'].$field_str['id'])) { $poisk .= " AND
`".$field_str['module']."base`.`field".$field_str['id']."' =
'".$_GET['field'].$field_str['id']."'"; }
?>
<input type="date" class = "form-control" name="field<?php echo
$field_str['id'];?>" value="<?php echo $_GET['field'].$field_str['id'] ?>">
<?php
}
?>

<?php

```

```

if ($field_str['type']=="select")
{
if (!empty($_GET['field'].$field_str['id'])) { $poisk .= " AND
`".$field_str['module']."base`.`field".$field_str['id']."' =
'".$_GET['field'].$field_str['id']."'"; }
?>
<select data-plugin="select2" class="form-control" name="field<?php echo
".$field_str["id"]."' ?>" >
<option value="">Выберите</option>
<?php
$list_count = mb_substr_count($field_str['listid'], ",");

$field_values_count =
mb_substr_count($object_str['field'].$field_str['id'],' |##| ');
$field_count = 0;
$field_value = explode(' |##| ', $object_str['field'].$field_str['id']);
$list_id = explode(",", $field_str['listid']);

$y = 0;

while ($y<=$list_count)
{
$users = mysql_query("SELECT id,name FROM `list".$list_id[$y]."` WHERE
(`deleted` = '0' OR `deleted` IS NULL) ORDER BY `sortid` ASC ");
while ($users_str = mysql_fetch_assoc($users))
{
if ($_GET['field'].$field_str['id']==$users_str['name'])
{
?>
<option selected value="<?php echo $users_str['name'] ?>"><?php echo
$users_str['name'] ?></option>
<?php
} else {
?>
<option value="<?php echo $users_str['name'] ?>"><?php echo
$users_str['name']; ?></option>
<?php
}
}
$y++;
}
?>
</select>
<?php
}
?>

<?php
if ($field_str['type']=="select_multi")
{
?>
<select class="form-control" data-plugin="select2" multiple
name="field<?php echo ".$field_str["id"]."' ?>[]">
<?php
$list_count = mb_substr_count($field_str['listid'], ",");

$field_values_count =
mb_substr_count($object_str['field'].$field_str['id'],' |##| ');
$field_count = 0;
$field_value = explode(' |##| ', $object_str['field'].$field_str['id']);
$list_id = explode(",", $field_str['listid']);

```

```

$get = "field".$field_str['id'];
$value = $_GET[$get];
$value_count = count($value);

if (!empty($value)) {
$poisk .= " AND (";
$count = 0;
while ($field_count<$value_count)
{
if ($field_count==0)
{
$poisk .= "      `".$field_str['module']."base`.`field".$field_str['id']."` =
'".$value[$field_count]."'";
} else {
$poisk .= " OR `".$field_str['module']."base`.`field".$field_str['id']."` =
'".$value[$field_count]."'";
}

$field_count++;
}
$poisk .= " )";

} else {
$value_count = 1;
}

$y = 0;

while ($y<=$list_count)
{
$users = mysql_query("SELECT id,name FROM `list".$list_id[$y]."` WHERE
(`deleted` = '0' OR `deleted` IS NULL) ORDER BY `sortid` ASC ");
while ($users_str = mysql_fetch_assoc($users))
{
$i=0;
while ($i<$value_count)
{
if (($value[$i]==$users_str['name']) AND (!empty($value)))
{
?>
<option selected value="<?php echo $users_str['name'] ?>"><?php echo
$users_str['name'] ?> </option>
<?php
} else {
?>
<option value="<?php echo $users_str['name'] ?>"><?php echo
$users_str['name']; ?></option>
<?php
}
}
}
}
$y++;
}
?>
</select>
<?php

```

```

}
?>
</div>
<?php
if ((( $count_now % $count_on_row )) == 0)
{
?>
</div>
<?php
}
}
?>

</fieldset>
<?php
}
?>
<?php
$count_on_row = $fieldset_str['col'];
$count_now = 0;
$field = mysql_query("SELECT id,name,type,module,listid FROM `lfield` WHERE
`deleted` = '0' AND `module` = '". $module.'" AND `fieldset` =
'".$fieldset_str['id']."' ORDER BY `row` ASC ");
while($field_str = mysql_fetch_assoc($field))
{
if ((( $count_now % $count_on_row )) == 0)
{
?>
<div class="col-md-12" style="padding-bottom: 10px;">
<?php
}
$count_now ++;
?>
<div class="col-md-<?php echo 12/$fieldset_str['col'] ?>" style="padding-
bottom: 0px;">
<?php echo $field_str['name'];?>
<?php
if ($field_str['type']=="text")
{
if (!empty($_GET['field'].$field_str['id'])) { $poisk .= " AND
`".$field_str['module']."base`.`field".$field_str['id']."'`
='".$_GET['field'].$field_str['id']."'"; }
?>
<input type="text" class = "form-control" name="field<?php echo
$field_str['id'];?>" value="<?php echo $_GET['field'].$field_str['id'] ?>">
<?php
}
?>

<?php
if ($field_str['type']=="textarea")
{
if (!empty($_GET['field'].$field_str['id'])) { $poisk .= " AND
`".$field_str['module']."base`.`field".$field_str['id']."'`
='".$_GET['field'].$field_str['id']."'"; }
?>
<textarea rows="3" class = "form-control" cols="160" name="field<?php echo
$field_str['id'];?>"><?php echo $_GET['field'].$field_str['id'] ?></textarea>
<?php
}
?>

```

```

<?php
if ($field_str['type']=="date")
{
if (!empty($_GET['field'].$field_str['id'])) { $poisk .= " AND
`".$field_str['module']."base`.`field".$field_str['id']."`
'".$_GET['field'].$field_str['id']."'"; }
?>
<input type="date" class = "form-control" name="field<?php echo
$field_str['id'];?>" value="<?php echo $_GET['field'].$field_str['id'] ?>">
<?php
}
?>

<?php
if ($field_str['type']=="select")
{
if (!empty($_GET['field'].$field_str['id'])) { $poisk .= " AND
`".$field_str['module']."base`.`field".$field_str['id']."`
'".$_GET['field'].$field_str['id']."'"; }
?>
<select data-plugin="select2" class="form-control" name="field<?php echo
".$field_str["id"]."' ?>">
<option value="">Выберите</option>
<?php
$list_count = mb_substr_count($field_str['listid'], ",");

$field_values_count =
mb_substr_count($object_str['field'].$field_str['id'],' |##| ');
$field_count = 0;
$field_value = explode(' |##| ', $object_str['field'].$field_str['id']);
$list_id = explode(",", $field_str['listid']);

$y = 0;

while ($y<=$list_count)
{
$users = mysql_query("SELECT id,name FROM `list".$list_id[$y]."` WHERE
(`deleted` = '0' OR `deleted` IS NULL) ORDER BY `sortid` ASC ");
while ($users_str = mysql_fetch_assoc($users))
{
if ($_GET['field'].$field_str['id']==$users_str['name'])
{
?>
<option selected value="<?php echo $users_str['name'] ?>"><?php echo
$users_str['name'] ?></option>
<?php
} else {
?>
<option value="<?php echo $users_str['name'] ?>"><?php echo
$users_str['name']; ?></option>
<?php
}
}
$y++;
}
?>
</select>
<?php
}

```

```

?>

<?php
if ($field_str['type']=="select_multi")
{
?>

    <select      class="form-control"      data-plugin="select2"      multiple
name="field<?php echo "". $field_str["id"]." ?>[]" class="full-width">
    <?php
    $list_count = mb_substr_count($field_str['listid'], ",");

    $field_values_count =
mb_substr_count($object_str['field'].$field_str['id'],' |##| ');
    $field_count = 0;
    $field_value = explode(' |##| ', $object_str['field'].$field_str['id']);
    $list_id = explode(",", $field_str['listid']);

    $get = "field".$field_str['id'];
    $value = $_GET[$get];
    $value_count = count($value);

    if (!empty($value)) {
    $poisk .=" AND (";
    $count = 0;
    while ($field_count<$value_count)
    {
    if ($field_count==0)
    {
    $poisk .="      `".$field_str['module']."base`.`field".$field_str['id']."` =
'".$value[$field_count]."'";
    } else {
    $poisk .=" OR `".$field_str['module']."base`.`field".$field_str['id']."` =
'".$value[$field_count]."'";
    }

    $field_count++;
    }
    $poisk .=" )";

    } else {
    $value_count = 1;
    }

    $y = 0;

    while ($y<=$list_count)
    {
    $users = mysql_query("SELECT id,name FROM `list".$list_id[$y]."` WHERE
(`deleted` = '0' OR `deleted` IS NULL) ORDER BY `sortid` ASC ");
    while ($users_str = mysql_fetch_assoc($users))
    {
    $i=0;
    while ($i<$value_count)
    {
    if (($value[$i]==$users_str['name']) AND (!empty($value)))
    {
    ?>

```



```

<option selected value="<?php echo $users_str['name'] ?>"><?php echo
$users_str['name'] ?> </option>
<?php
} else {
?>
<option value="<?php echo $users_str['name'] ?>"><?php echo
$users_str['name']; ?></option>
<?php
}
$i++;
}
}
$y++;
}
?>
</select>
<?php
}
?>
</div>
<?php
if ((( $count_now % $count_on_row) == 0)
{
?>
</div>
<?php
}
}
?>

</fieldset>
<?php
}
$sort = $_GET["sort"];
$sort2 = $_GET["sort2"];
if (empty($sort)) { $sortby = "edittime"; } else { $sortby = "".$sort.""; }
if (empty($sort2)) { $sortby .= " DESC"; $sort2 = 'DESC'; } else { $sortby .=
".$sort2.""; }
?>
<input id="sortir" type="text" name="sort" style="display: none;"
value="<?php echo $sort ?>">
<input id="sortir2" type="text" name="sort2" style="display: none;"
value="<?php echo $sort2 ?>">
<div class="modal-footer">
<button type="button" class="btn btn-default
margin-0" data-dismiss="modal">Закрыть</button>
<button type="submit" class="btn btn-
primary">Найти</button>
</div>
</form>
</div>
</div>
</div>
<!-- /.modal-content -->

</div>
</div>

```

Листинг алгоритма генерации таблицы базы данных

```

<div id="baza">
<div id="product_add_result"></div>

```

```

        <div class="panel-body" id="exampleTableAdd">

        <?
        $sort = $_GET["sort"];
        $sort2 = $_GET["sort2"];

        $module = htmlspecialchars($_GET["module"]);
        $filter = htmlspecialchars($_GET["filter"]);
        $filename2 = $_SERVER["REQUEST_URI"];

        $filename2
        str_replace('&sort='.$sort.'&sort2='.$sort2.'','',',$filename2);
        $filename2 = str_replace('&&','',',$filename2);
        if (empty($sort)) {$sort = 'edittime'; }
        if (empty($sort2)) {$sort2 = 'DESC'; }
            if ($sort2=='DESC') { $sort22 = str_replace('DESC','ASC',$sort2);}
            if ($sort2=='ASC') { $sort22 = str_replace('ASC','DESC',$sort2); }

        if (!empty($_GET["userz"])) {$userz = ' AND `'.$module.'base`.`userz` =
        `'.$_GET["userz"].'`; }
        if (!empty($_GET["archive"])) {$archive = ' AND `'.$module.'base`.`archive` =
        `'.$_GET["archive"].'`; } else { $archive = ' AND
        `'.$module.'base`.`archive` = "0"'; }
        if (!empty($_GET["deleted"])) {$deleted = ' AND `'.$module.'base`.`deleted` =
        `'.$_GET["deleted"].'`; } else { $deleted = ' AND
        `'.$module.'base`.`deleted` = "0"'; }
        ?>

        <table class="table table-hover dataTable table-striped width-full"
        >
            <thead>
                <tr>
                    <th style="width: 50px; <? if ($sort=='id') { ?> color: #46be8a; <? } ?>"
                    onclick="location.href = '<?php echo "$filename2.'" ?>&sort=id&sort2=<?php
                    echo $sort22 ?>';">id</th>
                    <th style="width: 150px;">Компания</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>
                        <?php
                        if (empty($filter))
                        {
                            $filter = mysql_query("SELECT id FROM `filter` WHERE `defolt` = 1 AND
                            userid = '".$_SESSION['id_user']."' AND module = '".$_module.'");
                            $count = mysql_num_rows($filter);
                            if (($count==0) or (empty($count))) {
                                $filter = mysql_query("SELECT id FROM `filter` WHERE module =
                                '".$_module.'" LIMIT 0,1");
                            }
                            $filter_str = mysql_fetch_assoc($filter);
                            $filter = $filter_str['id'];
                            $filter_sql = mysql_query("SELECT id,fieldid,sortid,width,name FROM
                            `filter_list` WHERE filter_id = '".$filter.'" ORDER BY `sortid` ASC ");
                        } else {
                            $filter_sql = mysql_query("SELECT id,fieldid,sortid,width,name FROM
                            `filter_list` WHERE filter_id = '".$filter.'" ORDER BY `sortid` ASC ");
                        }
                        $j = 0;
                        $id = htmlspecialchars($_GET['id']);
                        if (empty($id)) { $str = ""; } else { $str .= "AND id LIKE '$id'"; }
                        $module = htmlspecialchars($_GET["module"]);
                        $col_count = mysql_num_rows($filter_sql)+1;
                        $with_col = 100/$col_count;
                        while ($fields_str2 = mysql_fetch_assoc($filter_sql))
                    </td>
                </tr>
            </tbody>
        </table>
    </div>

```

```

{
    $field_name = mysql_query("SELECT id,name FROM `lfield` WHERE id =
'".$fields_str2['fieldid']."' AND (module = '".$module.'" OR module = '13')
"); $field_str2 = mysql_fetch_assoc($field_name);
?>
        <th onclick="location.href = '<?php echo "'.$filename2.'"
?>&sort=field<?php echo "'.$field_str2['id'].'"' ?>&sort2=<?php echo $sort2
?>';"
            style="width: <?php if (empty($fields_str2["width"]))
{ echo $with_col."%"; } else { echo $fields_str2["width"]."px"; } ?>;
            <? if ($sort=='field'.$field_str2['id']) { ?> color:
#46be8a; <? } ?>"><?php if (empty($fields_str2["name"])) { echo
$field_str2["name"]; } else { echo $fields_str2["name"]; } ?></th>
<?Php
}
?>
<th style="width: 50px;"><a href="#" data-toggle="modal" data-
target="#exampleNiftySlideFromRight_edit_filter"><i class="icon wb-edit"
aria-hidden="true"></i></a></th>

        <div class="modal fade slide-up"
id="exampleNiftySlideFromRight_edit_filter" tabindex="-1" role="dialog" aria-
hidden="false">
            <div class="modal-dialog" >
                <div class="modal-content-wrapper">
                    <div class="modal-content" style="width: 900px;min-height:
700px;">
                        <div class="modal-body m-t-20">
                            <h4 class="no-margin p-b-10">Редактирование
порядка
фильтра <?php echo $filter_str['name']; ?></h4>
<form method="post" action="" id="form_filter_sort">

<?php
$module = htmlspecialchars($_GET["module"]);
$filter = htmlspecialchars($_GET["filter"]);
if (empty($filter))
{
    $filter = mysql_query("SELECT id FROM `lfilter` WHERE `defolt` = 1 AND
userid = '".$SESSION['id_user']."' AND module = '".$module."'");
    $count = mysql_num_rows($filter);
    if (($count==0) or (empty($count))) {
        $filter = mysql_query("SELECT id FROM `lfilter` WHERE module =
'".$module.'" LIMIT 0,1");
    }
    $filter_str = mysql_fetch_assoc($filter);
    $filter = $filter_str['id'];
    $filter_sql = mysql_query("SELECT id,fieldid,sortid,width,name FROM
`lfilter_list` WHERE filter_id = '".$filter.'" ORDER BY `sortid` ASC ");
} else {
    $filter_sql = mysql_query("SELECT id,fieldid,sortid,width,name FROM
`lfilter_list` WHERE filter_id = '".$filter.'" ORDER BY `sortid` ASC ");
}
?>
<input type="text" class = "form-control" name="filterid" style="display:
none;" value="<?php echo "'.$filter_str["id'].'"' ?>">
<input type="text" class = "form-control" name="module" style="display:
none;" value="<?php echo "'.$module.'" ?>">
<?php
while ($fields_str2 = mysql_fetch_assoc($filter_sql))
{
    $field_name = mysql_query("SELECT name FROM `lfield` WHERE id =
'".$fields_str2['fieldid']."' AND (module = '".$module.'" OR module = '13')
"); $field_str2 = mysql_fetch_assoc($field_name);

```

```

?>
<div class="col-lg-12 form-group">
<div class="col-md-2">
Homep
<input type="text" name="field<?php echo $fields_str2["id"] ?>"
value="<?php echo $fields_str2["sortid"] ?>" class="form-control">
</div>
<div class="col-md-6">
Имя
<input type="text" name="field_name<?php echo $fields_str2["id"] ?>"
value="<?php if (empty($fields_str2["name"])) { echo $field_str2["name"]; }
else { echo $fields_str2["name"]; } ?>" class="form-control">
</div>
<div class="col-md-4">
Ширина px
<input type="text" name="field_width<?php echo $fields_str2["id"] ?>"
value="<?php echo $fields_str2["width"] ?>" class="form-control">
</div>
</div>

<?php
}
?>
<div id="result_div_filter_sort"></div>
</form>
<button type="button" class="btn btn-default
margin-0" data-dismiss="modal">Закрыть</button>
<button type="button" class="btn btn-primary"
onclick="AjaxFormRequest('result_div_filter_sort',
'form_filter_sort',
'funktions/edit_filter_sort.php')">Сохранить</button>

</div>
</div>
</div>
<!-- /.modal-content -->

</div>
</div>

</tr>
</thead>
<tfoot>
<tr>
<th style="width: 50px;">id</th>
<th style="width: 150px;">Компания</th>
<?php
$module = htmlspecialchars($_GET["module"]);
$filter = htmlspecialchars($_GET["filter"]);
if (empty($filter))
{
$filter = mysql_query("SELECT * FROM `lfilter` WHERE `defolt` = 1 AND
userid = '". $_SESSION['id_user']."' AND module = '". $module."'");
$count = mysql_num_rows($filter);
if (($count==0) or (empty($count))) {
$filter = mysql_query("SELECT * FROM `lfilter` WHERE module =
'". $module."' LIMIT 0,1");
}
$filter_str = mysql_fetch_assoc($filter);
$filter = $filter_str['id'];
$filter_sql = mysql_query("SELECT * FROM `lfilter_list` WHERE filter_id =
'". $filter."' ORDER BY `sortid` ASC ");
} else {

```

```

$filter_sql = mysql_query("SELECT * FROM `1filter_list` WHERE filter_id =
'".$filter.'" ORDER BY `sortid` ASC ");
}
$j = 0;
$id = htmlspecialchars($_GET['id']);
if (empty($id)) { $str = ""; } else {$str .= "AND id LIKE '$id'";}
$module = htmlspecialchars($_GET["module"]);
$col_count = mysql_num_rows($filter_sql)+1;
$with_col = 100/$col_count;
while ($fields_str2 = mysql_fetch_assoc($filter_sql))
{
    $field_name = mysql_query("SELECT * FROM `1field` WHERE id =
'".$fields_str2['fieldid']."' AND (module = '".$module.'" OR module = '13')
"); $field_str2 = mysql_fetch_assoc($field_name);
?>
        <th style="width: <?php if (empty($fields_str2["width"])) {
echo $with_col."%"; } else { echo $fields_str2["width"]."px"; } ?>";><?php
if (empty($fields_str2["name"])) { echo $field_str2["name"]; } else { echo
$fields_str2["name"]; } ?></th>
<?Php
}
?>
<th style="width: 50px;"><a href="#" data-toggle="modal" data-
target="#exampleNiftySlideFromRight_edit_filter"><i class="icon wb-edit"
aria-hidden="true"></i></a></th>
        </tr>
    </tfoot>
<tbody>
<?php
$module = htmlspecialchars($_GET["module"]);
$filter = htmlspecialchars($_GET["filter"]);
$fields_row
=''. $module.'base`. `mjidid`, `'. $module.'base`. `id`, `'. $module.'base`. `addtime
`, `'. $module.'base`. `edittime`, `'. $module.'base`. `userz`, `'. $module.'base`. `c
ompanyid`, `';
if (empty($filter))
{
    $filter = mysql_query("SELECT * FROM `1filter` WHERE `defolt` = 1 AND
userid = '".$SESSION['id_user']."' AND module = '".$module.'"");
    $count = mysql_num_rows($filter);
    if (($count==0) or (empty($count))) {
    $filter = mysql_query("SELECT * FROM `1filter` WHERE module =
'".$module.'" LIMIT 0,1");
    }
    $filter_str = mysql_fetch_assoc($filter);
    $filter = $filter_str['id'];
    $filter_sql = mysql_query("SELECT * FROM `1filter_list` WHERE filter_id =
'".$filter.'" ORDER BY `sortid` ASC ");
    } else {
    $filter_sql = mysql_query("SELECT * FROM `1filter_list` WHERE filter_id =
'".$filter.'" ORDER BY `sortid` ASC ");
    }
    $j = 0;
    $id = htmlspecialchars($_GET['id']);
    if (empty($id)) { $str = ""; } else {$str .= "AND id LIKE '$id'";}
    $module = htmlspecialchars($_GET["module"]);
    while ($fields_str2 = mysql_fetch_assoc($filter_sql))
    {
        $field_name = mysql_query("SELECT * FROM `1field` WHERE id =
'".$fields_str2['fieldid']."' AND (module = '".$module.'" OR module =
'13') "); $field_str2 = mysql_fetch_assoc($field_name);
        $fields_row
        .= `'. $field_str2['module']. `base`. `field`. $field_str2['id']. ``, `';

```

```

}
$fields_row = substr($fields_row, 0, -1);

$objects = "SELECT $fields_row FROM `".$module."base`,`13base`
WHERE
(
$privyazka
) $poisk $userz $archive $deleted AND
`".$module."base`.`companyid`='".$_SESSION['companyid']."' ORDER BY
".$sortby."
";

$like = htmlspecialchars($_GET["like"]);
if ($like==1)
{
$likes = " AND (`".$module."base`.`id` LIKE `likes`.`cartid` AND
`likes`.`moduleid` = '".$module.'" AND `likes`.`userid` =
'".$_SESSION['id_user']."'");
}

$objects = "SELECT $fields_row FROM `".$module."base`,`13base`,`likes`
WHERE
(
$privyazka
) $poisk $userz $archive $deleted $likes AND
`".$module."base`.`companyid`='".$_SESSION['companyid']."' ORDER BY
".$sortby."
";
}

$objects = "SELECT $fields_row
FROM `".$module."base`
LEFT JOIN 13base ON `".$module."base`.mjdid = 13base.id
where `".$module."base`.id !=0
$poisk $userz $archive $deleted AND `".$module."base`.`exclusive`='1' ORDER
BY ".$sortby."
";

$paginationlink = $filename.".php?page=";
$page = 1;
if(!empty($_GET["page"])) { $page = $_GET["page"]; }

$start = ($page-1)*$perPage->perpage;
if($start < 0) $start = 0;

$query = $objects . " LIMIT " . $start . "," . $perPage->perpage;
//echo $query;
$result = $db_handle->runQuery($query);
$sql = mysql_query($objects);
if(empty($_GET["rowcount"])) { $_GET["rowcount"] = mysql_num_rows($sql); }
$perPageResult = $perPage->perpage($_GET["rowcount"], $paginationlink);

$objects_i = 0; // счетчик объектов, которые не подошли

if(!empty($result)) { foreach($result as $k=>$v) {

$company = mysql_query("SELECT name,phone FROM `company` WHERE id =
'".$result[$k]['companyid']."'");
$company_str = mysql_fetch_assoc($company);

echo "<tr>";
?>

```

```

<td><?php echo $result[$k]['id'] ?></td>
<td><?php echo $company_str['name'] ?> <?php echo $company_str['phone']
?></td>
<?Php
if (empty($filter))
{
$filter = mysql_query("SELECT * FROM `lfilter` WHERE `defolt` = 1 AND
userid = '". $_SESSION['id_user']."' AND module = '". $module."");
$count = mysql_num_rows($filter);
if (($count==0) or (empty($count))) {
$filter = mysql_query("SELECT * FROM `lfilter` WHERE module =
'".$module."' LIMIT 0,1");
}
$filter_str = mysql_fetch_assoc($filter);
$filter = $filter_str['id'];
$filter_sql = mysql_query("SELECT * FROM `lfilter_list` WHERE filter_id =
'".$filter."' ORDER BY `sortid` ASC ");
} else {
$filter_sql = mysql_query("SELECT * FROM `lfilter_list` WHERE filter_id =
'".$filter."' ORDER BY `sortid` ASC ");
}

while ($filter_sql_str = mysql_fetch_assoc($filter_sql))
{
$str = "field".$filter_sql_str['fieldid'];
$field_name = mysql_query("SELECT * FROM `lfield` WHERE id = '". $res_list."'
AND AND (module = '". $module."' OR module = '13') "); $field_name_str =
mysql_fetch_assoc($field_name);
if (($field_name_str['type']=="date") OR
($field_name_str['type']=="datetime"))
{
echo "<td>".date('d.m.Y H:i', strtotime($result[$k][$str]))."</td>" ;
} else {
echo "<td>".str_replace(' |##| ', ', ', $result[$k][$str])."</td>" ;
}

}
echo "<td>";
echo "<a href='e_object.php?module=".$module."&id=".$result[$k]['id']."'
target='_blank' ><i class='icon wb-edit' aria-hidden='true'></i></a>";
if ($result[$k]['mjid']==0) {
?>
<button type="button" class="btn btn-default" data-
toggle="tooltip" data-placement="top" style="padding: 0;background: none;
border: none;"
data-trigger="click" data-original-title="Нет связи с МЖД"
title=""><i class='icon wb-extension' aria-hidden='true' style='color:
#a2a2a2;'></i></button>
<?
}

echo "</td>";

echo "</tr>";
$objects_i++;
} } else {
echo "<tr>";
echo "<td><h3></h3></td>";
echo "<td colspan='7'><p style='font-size: 16px;color: #8c8c8c;text-align: -
webkit-center;'>Подходящие заявки не найдены, но вы можете <a href =
'add.php?module=".$module."'>создать новую</a>.</p></td>";
echo "</tr>";
}

```

```

?>
        </tbody>
    </table>
    <div class="btn-group" style="width: 100%;text-align: -webkit-center;">
    <?php if(isset($perpageresult)) { echo $perpageresult; } ?>
</div>
    </div>
        </div>
    </div>
    </div>
</div>

```

Листинг алгоритма создания нового объекта в базе данных

```

$id_sql = mysql_query("SELECT MAX(id) as max FROM `".$module."base`"); //
получаем последний айди в базе
$row = mysql_fetch_assoc($id_sql);
$id = $row['max'];
$id = $id + 1; // и увеличиваем его на 1

$str_values = ("'.$id.'",");
$str_fields = ('`id`,`');

$result = mysql_query("SELECT id,type FROM `1field` WHERE deleted = '0' AND
(module = '".$module."')");
while ($field_str = mysql_fetch_assoc($result))
{
if (!empty($_POST["field".$field_str['id']]))
{
if (($field_str['type'] == 'select_multi'))
{
$znachenie = $_POST["field".$field_str['id']];
$znachenie_count = count($znachenie);
if (($znachenie_count == 1) OR (empty($znachenie[1]))) {$stroka =
$znachenie[0];} else {
$i=0;
$stroka = "";
while ($i<$znachenie_count)
{
$stroka .= $znachenie[$i];
$stroka .= " |##| ";
$i++;
}
$stroka = substr($stroka, 0, -6);
}
} else {
$stroka = $_POST["field".$field_str['id']];
}
}
}
$str_values .= "'.$stroka.'",";
$str_fields .= "`field".$field_str['id']."`,`";
}
}
}
$str_values = substr($str_values, 0, -1);
$str_fields = substr($str_fields, 0, -1);

```



```

$str_values .= '>';
$str_fields .= '>';
$add = mysql_query("INSERT INTO `".$module."base` $str_fields values
$str_values");

$data = date("y-m-d H:i:s");
$sql = mysql_query("UPDATE `".$module."base` SET `addtime` = '".$data."'
WHERE `id` LIKE '".$id."'");
$sql = mysql_query("UPDATE `".$module."base` SET `edittime` = '".$data."'
WHERE `id` LIKE '".$id."'");
$sql = mysql_query("UPDATE `".$module."base` SET `companyid` =
'".$_SESSION['companyid']."' WHERE `id` LIKE '".$id."'");
$sql = mysql_query("UPDATE `".$module."base` SET `userid` =
'".$_SESSION['id_user']."' WHERE `id` LIKE '".$id."'");
$sql = mysql_query("UPDATE `".$module."base` SET `userz` =
'".$_SESSION['id_user']."' WHERE `id` LIKE '".$id."'");

//echo "Все";
echo "<meta http-equiv='refresh' content='1;URL=add-step-
2.php?module=".$module."&id=".$id."' />";
} else {
echo $errors;
}
}

```

Листинг алгоритма внесения изменений в базу данных

```

$object = mysql_query("SELECT * FROM `".$module."base` WHERE `id` =
'".$id."'");
$object_str = mysql_fetch_assoc($object);
$result = mysql_query("SELECT id,type FROM `1field` WHERE deleted = '0' AND
(module = '".$module."'");
while ($field_str = mysql_fetch_assoc($result))
{
if (($field_str['type'] == 'select_multi'))
{
$znachenie = $_POST["field".$field_str['id']];
$znachenie_count = count($znachenie);
if (($znachenie_count == 1) OR (empty($znachenie[1]))) {$stroka =
$znachenie[0];} else {
$i=0;
$stroka = "";
while ($i<$znachenie_count)
{
$stroka .= $znachenie[$i];
$stroka .= " |##| ";
$i++;
}
$stroka = substr($stroka, 0, -6);
}
} else {
$stroka = $_POST["field".$field_str['id']];
}
if ($stroka != $object_str["field".$field_str['id']])
{
$sql = mysql_query("UPDATE `".$module."base` SET `field".$field_str['id']."'
= '".$stroka."' WHERE `id` LIKE '".$id."'");
}
//echo "<br>".$stroka;
}

```

```

//$add = mysql_query("INSERT INTO `".$module."base` $str_fields values
$str_values");

$data = date("y-m-d H:i:s");
$sql = mysql_query("UPDATE `".$module."base` SET `createdate` = '".$data."'
WHERE `id` LIKE '". $id ."'");
$sql = mysql_query("UPDATE `".$module."base` SET `editdate` = '".$data."'
WHERE `id` LIKE '". $id ."'");
$sql = mysql_query("UPDATE `".$module."base` SET `companyid` =
'".$_SESSION['companyid']."' WHERE `id` LIKE '". $id ."'");
//$sql = mysql_query("UPDATE `".$module."base` SET `userid` =
'".$_SESSION['id_user']."' WHERE `id` LIKE '". $id ."'");
$sql = mysql_query("UPDATE `".$module."base` SET `userz` =
'".$_POST['userz']."' WHERE `id` LIKE '". $id ."'");

$rek = $_POST["rek"];
$eks = $_POST["eks"];

if ($rek != 'on') { $rek = 0; } else{ $rek = 1; }
if ($eks != 'on') { $eks = 0; } else{ $eks = 1; }

$sql = mysql_query("UPDATE `".$module."base` SET `reklama` = '".$rek."'
WHERE `id` LIKE '". $id ."'");
$sql = mysql_query("UPDATE `".$module."base` SET `exclusive` = '".$eks."'
WHERE `id` LIKE '". $id ."'");

$users = mysql_query("SELECT * FROM `".$module ."base` WHERE (`id` = '". $id
.'"') AND `companyid`='".$_SESSION['companyid']."' LIMIT 0 , 1");
$object_str = mysql_fetch_assoc($users);

```

Выпускная квалификационная работа выполнена мной самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

«___» _____ г.

(подпись)

(Ф.И.О.)