МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа электроники и компьютерных наук
Кафедра «Информационно-аналитическое обеспечение управления в социальных и экономических системах»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой,
д.т.н., профессор

_____/О.В.Логиновский/
«___»_____ 2018 г.

Оптимизация базы данных о клиентах ООО «УралГИС»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ-09.03.01.2018.114. ПЗ ВКР

Руководитель ВКР,
к.п.н., доцент
_____/В.Н Максимова/
«___»_____2018 г.

Автор ВКР
Студент группы КЭ-486
_____/А.В. Игнатьева/
«___»_____2018 г.

Нормоконтролер,
к.т.н., доцент
_____/В. Н. Любицын/
«___»_____2018 г.

Челябинск 2018

АННОТАЦИЯ

Игнатьева А.В. Оптимизация базы данных о клиентах ООО «УралГИС» – Челябинск: ЮУрГУ, ВШ ЭКН; 2018, 65 с., 10 ил., 2 таблиц, , библиогр. список - 15 наим.

В рамках выполнения квалификационной работы были изучены делопроизводственные процессы ООО «УралГИС».

В ходе анализа были обнаружены недостатки в организации системы учета клиентов и обоснована целесообразность оптимизации существующей системы.

Были разработаны требования к оптимизации системы. В результате сравнительного анализа представленных на рынке систем управления взаимоотношениями с клиентами было принято разрабатывать собственную систему. Разработка подкреплена экономической эффективностью оптимизации системы учета клиентов.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 8
1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СИСТЕМ УПРАВЛЕНИЯ ВЗАИМООТНОШЕНИЯМИ С КЛИЕНТАМИ 9
1.1. Понятие автоматизации учета клиентов 9
1.2. Общая классификация 13
1.3 Обзор основных систем управления взаимоотношениями с клиентами 10
Выводы по разделу один 16
2. ХАРАКТЕРИСТИКА ООО «УралГИС» 19
2.1. Организационно-правовая форма и вид деятельности 20
2.2. Организационная структура 21
Выводы по разделу два 21
3. РАЗРАБОТКА ТРЕБОВАНИЙ К ОПТИМИЗАЦИИ БАЗЫ ДАННЫХ О КЛИЕНТАХ ООО «УралГИС» 22
3.1 Анализ делопроизводственных процессов ООО «УралГИС» 24
3.2 Требования к оптимизации базы данных о клиентах 28
Выводы по разделу три 29
4. ОПТИМИЗАЦИЯ БАЗЫ ДАННЫХ 30
4.1 Описание процесса создания базы данных 30
4.2 Детальное описание разработки требований к оптимизации системы 37
4.3 Расчет экономической эффективности от оптимизации базы данных 59
Выводы по разделу четыре 60
ЗАКЛЮЧЕНИЕ 61
БИБЛИОГРАФИЧЕСКИЙ СПИСОК 62
ПРИЛОЖЕНИЕ А Существующая система учета клиентов ООО «УралГИС» 64
ПРИЛОЖЕНИЕ Б Структурная схема базы данных 65

ВВЕДЕНИЕ

Актуальность проблемы. В настоящее время автоматизация рабочих процессов предприятия напрямую зависит от его дальнейшего развития, и дохода. Особое значение занимает грамотно реализованная система учета клиентов, так как она затрагивает такие процессы предприятия как точное планирование, работа с имеющимися клиентами и их удержание. Не имея автоматизированной системы учета клиентов, компания теряет существенные средства из-за недостаточно эффективной работы отделов обслуживания, вызванной медленной обработкой поступающих данных, а также их разрозненностью.

Целью данной работы является оптимизация существующей системы учета клиентов предприятия, а именно разработка удаленной базы данных и web-интерфейса для учета клиентов. При такой реализации нагрузка распределяется между сервером и рабочими станциями, что позволяет увеличить скорость работы программы, а переход к реляционной базе данных снижает риски возникновения ошибок и повышает безопасность системы.

1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СИСТЕМ УПРАВЛЕНИЯ ВЗАИМООТНОШЕНИЯМИ С КЛИЕНТАМИ

1.1. Общие понятия

Система управления взаимоотношениями с клиентами (CRM-система) – это программное обеспечение для компании, функцией которого является улучшенное обслуживание заказчиков, путём обработки информации о заказчиках и истории взаимоотношений с ними, улучшения бизнес-процедур и последующего анализа результатов. В настоящее время CRM – система имеет множество различных определений.

В переводе с английского клиента Управление взаимоотношениями с клиентами (CRM) - это управление взаимоотношениями с клиентами. Из этого следует, что CRM - это прежде всего не компьютерная программа, а технология работы компании на рынке. И эта технология во многом зависит от типа рынка, типа бизнеса и т. д. В этой связи CRM-система - это инструмент, который помогает реализовать эту технологию.

Анализ различных информационно-технологических и методологических решений, называемых системой управления взаимоотношениями с клиентами (так называемый CRM), позволяет сделать вывод о том, что CRM - это не программный продукт, а не набор программных продуктов или даже специальных информационных технологий. CRM - это прежде всего сложная организационная модель, суть которой «ориентирована на клиента». Этот подход предполагает высокий уровень информационно-технологических и организационных решений.

CRM - это модель взаимодействия, которая считает, что центром всей философии бизнеса является клиент, а основными видами деятельности являются меры по поддержке эффективного маркетинга, продаж и обслуживания клиентов. Поддержка этих бизнес-целей включает сбор, хранение и анализ информации о потребителях, поставщиках, партнерах, а также о внутренних процессах компании. Функции для поддержки этих бизнес-целей включают продажи, маркетинг, поддержку клиентов. Цель CRM-систем заключается в привлечении и

удержании прибыльных клиентов путем установления и улучшения отношений с ними.

Развитие стратегии взаимоотношений с клиентами стало возможным благодаря прорыву в области информационных технологий. Сегодня компании могут улучшить свою работу с клиентами, используя ряд технологий управления базами данных, а также растущее число приложений для CRM.

Такое развитие событий позволяет создавать обширные массивы клиентских данных, гарантирует обширную обратную связь от клиентов и позволяет анализировать, интерпретировать и конструктивно использовать данные.

Кроме того, наряду с ростом преимуществ, предоставляемых все более мощным компьютерным оборудованием, программами и электронными услугами, стоимость их услуг снижается. Этот избыток доступных инструментов для CRM позволяет компаниям более эффективно ориентироваться на наиболее перспективные области.

Основной задачей CRM-систем является повышение эффективности проектов и услуг, сосредоточенных в «фронт-оф-энд» и направленных на привлечение и удержание клиентов - в маркетинге, продажах, обслуживании и обслуживании, независимо от канала, с помощью которого осуществляется контакт с клиентом. [2]

Внедрение CRM принесет ощутимую пользу, если в компании есть следующие проблемы: разрозненная клиентская база, история общения с клиентами делится или не зарегистрированы вообще, потери информации при передаче между подразделениями приводит к сбоям в основных бизнес-процессов компании, регламентированы и утверждены бизнес-процессы не автоматизированы и не реализуются в "рабочая среда" фирмы, компании нужен инструмент прогнозирования продаж для активного управления бизнес-процессом продаж, нет возможности анализа клиентской базы, построения комплексных отчетов по продажам, закупкам и истории общения с клиентами, жалобы клиентов теряются, не рассматриваются вовремя; составление типового

договора или коммерческого предложения требуется Менеджер больше чем за полчаса; руководитель вынужден половину своего рабочего дня, контроль за работой сотрудников; сотрудники отдела маркетинга, продаж и обслуживания департаментов растет гораздо быстрее, чем компании, если эти проблемы возникают в компании, CRM-система может помочь в их решении.

Внедрение CRM-системы, скорее всего, не принесет ощутимой пользы для следующих предприятий: если организация имеет ограниченное количество клиентов и контактных лиц, и она не заинтересована в дальнейшем росте клиентской базы, если нет конкуренции в сегменте рынка или его уровень низок; если клиенты разовые, случайные, и продавец не беспокоиться о качестве оказываемых услуг, о сохранении постоянных клиентов, когда все работает так, как считает необходимым, и не нужно обмениваться информацией с коллегами. [3]

Таким образом, в целом CRM система направлена на привлечение и удержание потенциальных потребителей. Здесь важную роль играет система управления маркетингом, которая подразумевает под собой поиск количества потребителей, достаточного для данного уровня производства компании. Управление спросом означает управление клиентами, спрос на продукцию компании исходит от двух групп потребителей: новых и постоянных клиентов.

Теория и практика традиционного маркетинга были сосредоточены на привлечении новых клиентов и продажи своих товаров и услуг. Сегодня ситуация изменилась и современный маркетинг нацелен не только на привлечение новых клиентов, но и на поддержание хороших отношений с существующими клиентами.

Ситуации, в которых нецелесообразно внедрение CRM-системы приведены в Таблице 1.

Таблица 1 - Ситуации, в которых нецелесообразно внедрение CRM-системы

№ п/п	Ситуации	Пояснения
1	организация имеет ограниченное количество клиентов и контактных лиц, и она не заинтересована в дальнейшем росте клиентской базы	сознательное ограничение роста небольшой компании по управлению
2	в сегменте рынка нет конкуренции или ее уровень низок	на рынке нет конкурентов, а потребность в товаре компании высока, клиенты будут вынуждены приобретать его без дополнительных усилий со стороны продавца
3	клиенты разовые, случайные, и продавец не беспокоится о качестве услуг, о сохранении постоянных клиентов	стиль работы характерен для коммивояжеров. Если потенциальные клиенты длятся десятилетиями, то зачем беспокоиться о каждом из них?
4	каждый работает так, как считает нужным, и не обменивается информацией с коллегами	адвокатское бюро – сообщество независимых профессионалов, объединенных в одно юридическое лицо только формальными или техническими аспектами ведения бизнеса.

1.2 Общая классификация

Система управления взаимодействием с клиентами основана на использовании передовых управленческих и информационных технологий, с помощью которых компания строит взаимовыгодные отношения со своими клиентами.

В свете информационных технологий CRM-система представляет собой набор специфического программного обеспечения, позволяющего автоматизировать и усовершенствовать бизнес-процессы, связанные с управлением продажами, маркетингом и обслуживанием клиентов. Это программное обеспечение позволяет координировать не только действия различных департаментов, взаимодействующих с клиентом (продажи, маркетинг, сервис), но и работу различных каналов взаимодействия с клиентом – личное взаимодействие, телефон, Интернет – в целях установления долгосрочных и взаимовыгодных отношений.

CRM-система-это единое информационное хранилище, где информация о взаимодействии с клиентами, собираются, в ней предусматривается использование многих каналов взаимодействия: обслуживание на точках продаж, телефонные звонки, Электронная почта, мероприятия, встречи, регистрационные формы на веб-сайтах, рекламные ссылки, чаты, социальные сети; анализ собранной информации о клиентах и подготовка данных для принятия соответствующих решений-например, сегментация клиентов на основе их значимости для компании, потенциальном отклике на те или иные промо-акции, прогноз потребности в продукции компании.

Следовательно, CRM-решения помимо сбора и хранения информации делают ее доступной для всех сервисов компании, позволяют анализировать информацию, делать прогнозы и учитывать предпочтения клиентов и принимать решения по каждому конкретному клиенту.

В настоящее время CRM-системы можно классифицировать по следующим параметрам: функциональность (управление продажами, Управление маркетингом, управление клиентским обслуживанием и колл-центрами), уровни

обработки информации (операционный CRM, аналитический CRM, совместный CRM) по отраслям (Банковское дело, Финансы, фармацевтика, туризм и др.); Размер предприятий (малые, средние и крупные предприятия); производители (Российская, Зарубежная система).

Рассмотрим более подробно классификацию CRM-систем по уровням обработки информации. CRM-системы управления решают задачи быстрого ввода и хранения контактной информации, истории общения с клиентами. Операционные системы охватывают продажи, маркетинг, сервис. Данный тип CRM-системы полезен для многоэтапных и долгосрочных транзакций с несколькими участниками. Есть небольшой функционал в аналитике в операционных CRM системах-можно получить аналитику для клиентов, для сотрудников, для транзакций.

Аналитические CRM-системы, помимо решения операционных задач, позволяют сегментировать базу клиентов, оценить успешность проведения маркетинговой компании, оценить прибыльность клиентов, провести анализ продаж, анализ закупок и складских условиях, для прогнозирования объема продаж. Аналитические CRM-системы используются компаниями, у которых много" коротких"," быстрых " сделок, большой ассортимент и большое количество клиентов. Данный тип CRM-системы подходит для широкого круга задач. Аналитические CRM-системы в основном распространяются на малый и средний бизнес. Помимо операционных и аналитических возможностей, совместные CRM-системы позволяют клиенту принимать непосредственное участие в деятельности компании, например, в процессе разработки продукта или оказания услуг. На данный момент данный тип CRM-системы не распространен в России. [5]

Классификацию CRM-систем по уровню обработки информации можно обобщить в виде таблицы 2.

Таблица 2. Классификация CRM – систем по уровням обработки информации

Виды	Характеристика	Функциональные возможности	Примеры реализации
Операционный	регистрация и оперативный доступ к первичной информации по событиям, компаниям, проектам, контактам, документам	формализация всех бизнес-процессов взаимодействия с клиентом; контроль прохождения длительных и сложных сделок; анализ этапов сделок и проектов; планирование и контроль коммуникаций с клиентами; сбор и классификация максимальной информации о клиенте.	Для малых предприятий: ACT, GoldMine, Maximaizer, Sales Expert. Для средних: Clientele, Onyx, Sales Logix. Для крупных: Oracle, SAP, Siebel, BAAN, «Управление деловыми процессами. Парус-Клиент»
Аналитический	отчетность по первичным данным, более глубокий анализ информации в различных разрезах	анализ продаж в любых разрезах, ассортиментный и ценовой анализ, классификация клиентов по произвольным признакам, анализ конкурентной и рыночной среды, анализ закупок и склада, оценка маркетинговых мероприятий и других влияющих факторов; интеграция с учетными системами.	1C:CRM, Brio, Business Objects, Broadbase, E.Piphany, Hyperion, MicroStrategy, SAS, Marketing analytic
Коллаборационный	уровень организации тесного взаимодействия с конечными потребителями, клиентами, вплоть до влияния клиента на внутренние процессы компании	опросы, для изменения качеств продукта или порядка собслуживания, web-страницы для отслеживания клиентами состояния заказа, уведомление по SMS о проведённых транзакциях по банковскому счету, возможность для клиента самостоятельно скомплектовать и заказать в online, к примеру, автомобиль или компьютер из доступных блоков и опций и др.	IntraNet Solutions, Plumtree, Symon, Vignette, Aspect, Broadvision, Cisco

Вместе эти три компонента CRM поддерживают и питают друг друга; их интеграция просто необходима для успешной работы CRM, выражающейся в улучшении опыта взаимоотношений с клиентами. Совместный CRM позволяет клиентам контактировать с предприятием посредством целого спектра различных каналов и получать одинаковое впечатление от контакта с компанией по всем этим каналам.

1.3 Обзор основных систем управления взаимоотношениями с клиентами

РосБизнесСофт CRM - самая популярная российская web-CRM система для малого и среднего бизнеса. Это универсальный программный продукт для управления взаимоотношениями с клиентами, комплексной автоматизации бизнеса, а также успешного развития корпоративной информационной системы (КИС). Удобный и понятный интерфейс позволяет эффективно работать с web-CRM программой из любой точки мира. Внедрение РосБизнесСофт Онлайн CRM позволит значительно повысить производительность менеджеров, а также увеличит объем продаж на 30-40%.

Программный продукт «1С:Предприятие 8. Управление торговлей и взаимоотношениями с клиентами (CRM)» предназначен для комплексной автоматизации бизнес-процессов предприятия, оперативного и управленческого учета, анализа и планирования торговых операций, создания эффективных механизмов взаимоотношений с клиентами. «1С:Управление торговлей и взаимоотношениями с клиентами (CRM)» будет полезно для компаний, которые стремятся управлять бизнесом в соответствии с концепцией CRM: организаций оптово-розничной торговли; проектных организаций; сервисных организаций; организаций сферы услуг; выделенных торговых подразделений производственных компаний и т.д. [6]

Программа «1С: Управление торговлей и взаимоотношениями с клиентами (CRM)» автоматизирует следующие направления торговой деятельности: управление бизнес-процессами; автоматизация процессов продаж, начиная с момента первого обращения клиента; управление заказами покупателей;

управление продажами (включая оптовую, розничную и комиссионную торговлю); управление поставками; планирование продаж и закупок и др.

Основными результатами внедрения «1С:Управление торговлей и взаимоотношениями с клиентами (CRM)» являются: минимизация риска
Программы «1С: Управление торговлей и взаимоотношениями с клиентами (CRM)» или подразделениями компании (отделами продаж, закупок, склада, маркетинга, сервисного обслуживания, службы качества и т.д.); сегментация клиентской базы позволяет выявить перспективных клиентов и сократить время на неэффективные переговоры; у сотрудников появляется больше времени на работу с клиентами за счет повышения эффективности работы с информацией и автоматизации рутинных операций; сокращение времени обслуживания в расчете на одного клиента позволяет снизить общие издержки по продажам; снижение количества и размера скидок, до продажи и кросс-продажи позволяют получить максимум от работы с каждым клиентом и увеличить ценность сделок; персональное обслуживание клиентов (памятные даты, специальные предложения и программы лояльности), быстрое предоставление необходимой клиенту информации, точность и оперативность в работе повышает уровень лояльности клиентов, способствует стабильности и росту объема продаж. Благодаря гибкости и легкой настройке программа «1С: Управление торговлей и взаимоотношениями с клиентами (CRM)» реализует функции учета от ведения справочников и ввода первичных документов до получения различных аналитических отчетов. Конфигурация позволяет вести управленческий учет по торговому предприятию в целом. Для предприятия холдинговой структуры документы могут оформляться от имени нескольких организаций, входящих в холдинг. [7]

Рассмотрим другие виды программных продуктов. CRM-система ASoft CRM Lite разработана специально для представителей малого и среднего бизнеса с численностью пользователей, не превышающей 20 человек. Функциональные возможности ASoft CRM Lite в основном схожи с теми возможностями, которые дает стандартная версия системы ASoft CRM. В Lite версию не вошли следующие

разделы, которые упростили работу с системой: склад; маркетинг; техническая поддержка; IT структура.

BLUEJET web CRM - это система для отдела продаж, менеджеров по продажам, торговых агентов, ассистентов, операторов отделов телемаркетинга и сервиса, которая предоставляется по модели SaaS. Система BLUEJET web CRM предназначена для тех, кто понимает, что в бизнесе дорого время, что информация о клиентах принадлежит компании, что клиенты являются единственным источником средств существования и развития компании, что CRM-система для делового человека как наличие мобильного телефона, что адресная маркетинговая кампания дешевле и более эффективна для привлечения новых клиентов. BLUEJET web CRM охватывает основные аспекты деятельности компании и основные бизнес-процессы, т.к. повседневная поддержка работы Ваших специалистов не менее важна, чем контроль их деятельности: управление компаниями; управление контактами; управление продажами; управление проектами; управление маркетингом; организация совместной работы. [6]

АПЕК CRM Lite -это мощная CRM-система, которая позволяет централизовано управлять информацией о клиентах, продажах, закупках, финансах, задачах и кадрами. АПЕК CRM Lite является незаменимым помощником менеджеров, способствуя повышению эффективности их работы, а также дает возможность руководителю осуществлять анализ и контроль работы предприятия. Основные функции АПЕК CRM Lite: ведение базы клиентов; управление контактами; ведение всей историю взаимодействий с клиентом; напоминания о событиях, встречах и т.д.; управление задачами, статус выполнения задач; управление рабочим временем.

Еще одна наиболее популярная программа по работе с клиентами- система - Простой бизнес. Модуль управления взаимоотношениями с клиентами в составе комплекса «Простой бизнес» включает в себя: управление клиентской базой (анкетирование клиентов); управление контактами клиентов; обмен информацией с клиентами по электронной почте и факсу; отслеживание комментариев и

пожеланий клиента; архив переписки с клиентами; осуществление звонков средствами системы; ведение маркетинговой работы.

Программа «Простой бизнес» –это облачный сервис для построения целостного информационного пространства бизнеса. Комплекс совмещает в себе актуальные инструменты управления проектами, задачами, персоналом, счетами, продажами и другое. Это простая программа для решения сложных задач, которую просто установить и просто использовать.

Выводы по разделу один

Проведенное выше исследование позволяет сделать вывод, что в настоящее время существует достаточно большое предложение программных продуктов для системы управления взаимоотношениями с клиентами. Каждая из программ отличается своими функциями и выполнениям задач. Важно отметить, что при выборе программного продукта CRM-системы необходимо сопоставлять стратегию предприятия, ее цели, задачи, которые желает получить компания от внедрения данной системы, и ее отраслевую принадлежность. Не всегда дорогостоящая система CRM может дать желаемый результат и с точностью до наоборот, это касается и CRM-систем низкой и средней ценовой категории. При выборе программного продукта по управлению клиентами на предприятии, необходимо учитывать подготовленность персонала, корпоративную культуру и другие факторы, важные при выборе автоматизации системы взаимоотношения клиентами в организации.

2. ХАРАКТЕРИСТИКА ООО «УралГИС»

2.1 Организационно-правовая форма и вид деятельности

ООО «УралГИС» была образована в 2013 году. Является малым инновационным предприятием ЮУрГУ.

Малое инновационное предприятие (высокотехнологичная или наукоемкая фирма) – это предприятие, которое разрабатывает и внедряет высокотехнологические изделия. Такое предприятие обычно связывает науку и производство, тем самым превращая знание в готовый продукт.

Существует три способа создания МИП:

- ученые (инженеры) разработали продукт, который будет иметь коммерческий успех, для внедрения продукта создается МИП, чтобы найти потребителя;
- ученые временно покидают научную деятельность, для того, чтобы заработать стартовый капитал и создать МИП;
- заинтересованный в науке предприниматель, самостоятельно изучает рынок, находит программистов, развивает НИОКР, создает МИП.

Направление деятельности ООО «УралГИС»**:**

- разработка геоинформационных технологий и внедрение их в социально-экономической и промышленной сферах;
- работы по созданию цифровых карт для администрации Челябинской области, мониторинг объектов областного значения;
- проведение аэрофотосъемки, изготовление ортофотопланов местности;
- определение рельефа местности, создание изолиний поверхности;
- определение береговых линий водных объектов;
- кадастровые работы: межевание земельных участков, постановка на кадастровый учет;
- геодезические работы (вынос границ земельных участков в натуру, проведение топосъемки масштаба 1:500 – 1:5000);
- проведение судебных и досудебных экспертиз по спорным вопросам определения границ земельных участков;

2.2. Организационная структура

В ООО «УралГИС» принята функциональная структура управления, подразумевающий собой группирование конкретных должностей в отделы, основывается на основе общих видов деятельности.

В зависимости от задач организации эти виды деятельности могут быть разными. Ниже представлена функциональная схема организационной структуры предприятия. (Рисунок 1)

Рисунок 1 – Функциональная схема структуры предприятия

Преимущества этой организационной структуры:

- концентрированный учет достижений результатов стратегии;
- наибольший эффект управления для организаций, работающих в одной и той же деловой сфере;
- возможность достижения масштабного эффекта для функциональных разграничений;
- сокращение дублирования работы;

Недостатки этой организационной структуры:

- контроль за подразделениями ложится на директора;
- узкая специализация;
- вероятность возникновения конфликтов между отделами;
- передача ответственности за эффективность директору;
- длительный процесс принятия решений.

Выводы по разделу два

В данном разделе был проведен анализ деятельности ООО «УралГИС». Была определена организационно – правовая форма предприятия, и обозначены виды деятельности. Также была определена организационная структура предприятия: функциональная структура управления. Из приведенных выше преимуществ и недостатков данной структуры управления можно сделать вывод о том, что данный вид является подходящим для ООО «УралГИС» так как предприятие не имеет большого штата сотрудников, а группирование конкретных должностей в отделы, основанных на основе общих видов деятельности, является наиболее благоприятным для малых предприятий, работающих в одной сфере.

3. РАЗРАБОТКА ТРЕБОВАНИЙ К ОПТИМИЗАЦИИ БАЗЫ ДАННЫХ О КЛИЕНТАХ ООО «УралГИС»

3.1 Анализ делопроизводственных процессов ООО «УралГИС»

ООО «УралГИС» - это современное предприятие со своими бизнес-процессами и потоками данных. Одной из направлений деятельности компании является оказание услуг по межеванию земельных участков. Процесс межевания выглядит следующим образом:

1. Отбор и сканирование необходимых для постановки на кадастровый учет документов у заказчика работы;
2. Определение границы земельного участка в специализированном программном продукте «Quantum GIS».
3. Поиск и распечатка необходимых шаблонов документов (заявление на межевание, согласие на обработку персональных данных, договор, акт согласования границ смежных земельных участков, фрагмент ортофотоплана, акт приема-передачи документов, квитанция на оплату).
4. Заказчик заполняет документы.
5. Заказчик оплачивает квитанцию, и получает фрагмент ортофотоплана и акт согласования для последующего письменного сбора согласий с собственниками смежных участков.
6. После принятия заполненных актов согласования кадастровый инженер начинает подготовку межевого плана. Подготовка происходит в специализированном продукте «Межевой план».
7. Заказчик получает готовый межевой план и относит межевой план в многофункциональный центр для постановки на кадастровый учет.
8. Кадастровый инженер ожидает постановки на учет, проверяя статус участка на портале «Публичная кадастровая карта».

8.1 В случае успешной постановки участка на кадастровый учет, заказчик подписывает акт выполненных работ.

8.2 В случае возникновения замечаний кадастровый инженер делает исправления в межевом плане, и заказчик повторяет пункт 7.

9. В течении 30 дней кадастровый инженер обязан отправить оригиналы актов согласования в кадастровую палату, для регистрации результатов работы инженера, для этого заполняется шаблон документа (сопроводительное письмо).

Каждый этап процесса межевания регистрируется в таблице учета работ кадастрового инженера. Учет происходит в программном продукте Microsoft Office Excel. (Приложение А)

Таблица содержит следующие столбцы:

- дата приема документов (тип ячейки: дата)
- ФИО Заказчика (тип ячейки: текст)
- дата окончания действия договора (тип ячейки: дата)
- СНТ, наименование садового товарищества, в котором состоит заказчик (тип ячейки: текст)
- ФИО сотрудника, принявшего документы (тип ячейки: текст)
- дата приема акта согласования (тип ячейки: дата)
- ФИО кадастрового инженера (тип ячейки: текст)
- дата начала подготовки межевого плана (тип ячейки: дата)
- примечание, комментарии сотрудников о процессе работы (тип ячейки: текст)
- статус, состояние процесса изготовления межевого плана (тип ячейки: текст):

а) НАДО ДЕЛАТЬ! – заказчик сдал заполненные акты согласования, можно приступать к подготовке межевого плана;

б) ГОТОВО! – межевой план подготовлен;

в) ПРИОСТАНОВКА – межевой план ожидает исправления.

- результат, состояние межевого плана (тип ячейки: текст):

а) ОЖИДАЕТ КЛИЕНТА (межевой план подготовлен, но заказчик его не забрал);

б) МФЦ (заказчик отнес межевой план в регистрирующий орган);

в) НА УЧЕТЕ! – участок зарегистрирован;

- дата окончания подготовки межевого плана (тип ячейки: дата)
- дата сдачи актов согласования, дата сдачи кадастровым инженером актов согласования после завершения межевого плана (тип ячейки: дата)

- кадастровый номер (уникальный номер земельного участка) (тип ячейки: число)
- приостановка, указывается дата заявления, в котором пришло приостановления дела от регистрирующего органа (тип ячейки: дата)
- подпись, кадастровый инженер работает под уникальной цифровой подписью, для учета целевого использования ведется учет поставленных дел (тип ячейки: текст)
- номер заявления указывается номер заявления, в котором пришло приостановления дела от регистрирующего органа (тип ячейки: число)
- дата постановки на учет (тип ячейки: дата)

Достоинства системы учета клиентов на базе Microsoft Office Excel:

- низкая требовательность к программному обеспечению, с базой можно работать на маломощных компьютерах;
- доступность, Excel входит в стандартный пакет офис, предустановлена;
- простота использования (не требует специальных навыков в разработке и настройке баз данных);
- хранение больших объемов данных в одном файле;
- наглядность (оформление ячеек по цвету, построение диаграмм);
- встроенная математическая база (возможность производить вычисления над столбцами, для последующего анализа работы).

Недостатки:

- отсутствие многозадачности, с базой может обращаться только один сотрудник, что понижает общее время изготовления межевого плана;
- большие базы склонны к возникновению фатальных ошибок, что при отсутствии ежедневного резервного копирования ведет к большим потерям данных;
- человеческий фактор, на практике пользования Excel было выявлено, что пользователи не всегда могут сохранить результаты своей работы, путают очередность заполнения полей, что приводит к потери данных;
- невозможность отследить историю редактирования определенным пользователем, что часто приводит к возникновению конфликтов в компании;

3.2 Требования к оптимизации базы данных о клиентах

В результате анализа работы имеющейся системы были выявлены основные требования к оптимизации базы данных о клиентах:

1. Поддержка личных учетных записей у сотрудников.
2. Резервное копирование базы данных.
3. Создание отчетов по запросу пользователя.
4. Создание шаблона документов

Из соображений целевого использования бюджета компании, было принято разработать систему в следующей связи:

1. База данных: Microsoft SQL Server 2008 r2

Предназначена для управления, безопасного доступа к данным и работе с информацией. Особенностью инновационного продукта является наличие широкого набора инструментов, предназначенных для анализа бизнес-процессов и подготовки отчетной документации.

Microsoft SQL Server 2008 R2 обеспечивает высокий уровень надежности и производительности систем управления базами данных и оборудования локальной сети. Встроенный модуль Power Pivot позволяет производить загрузку данных в создаваемые проекты из любых внешних источников, информационных систем и электронных таблиц.

Предусмотрена возможность записи анализируемых сведений в виде бизнес-терминов, установки связей между ними, добавления формул и внешней информации. Microsoft SQL Server 2008 R2 сокращает временные и эксплуатационные затраты на создание аналитической документации посредством высокой степени быстродействия, наличия всех необходимых инструментов и упреждения применения сторонних приложений при осуществлении расчетов.

Простота использования функций программы позволяет производить анализ бизнес-данных даже сотрудниками, не имеющими специальных навыков в программировании. Microsoft SQL Server 2008 R2 содержит в своем составе компонент Reporting Services, обеспечивающий генерирование комплексных систем корпоративной отчетности для обслуживания тысяч пользователей.

Имеется надежный, простой и удобный инструмент для проектирования отчетности Report Designer и средства визуализации, облегчающие работу с данными конечных пользователей предприятия. Microsoft SQL Server 2008 R2 содержит встроенный OLAP-сервер для осуществления интеллектуального анализа многомерных данных в режиме текущего времени.

Система фильтрации дает возможность производить выделение значимой для заданных бизнес-процессов информации. Microsoft SQL Server 2008 R2 осуществляет одновременный учет данных из множества различных источников,

используя технологию комплексного анализа происходящих событий и обрабатывая более 10 000 записей в секунду.

Служба управления нормативно-справочными сведениями Master Data Services является высокоэффективным инструментом, предназначенным для осуществления централизованного мониторинга данных. Microsoft SQL Server 2008 R2 в полной мере соблюдает установленные регламентом предприятия процедуры, упреждая внесение несанкционированных изменений.

Для добавления информации в справочники в ручном режиме имеется широкий набор инструментов, позволяющих описывать структурные взаимосвязи отдельных компонентов при внедрении данных из различных источников. Microsoft SQL Server 2008 R2 значительно облегчает процесс консолидации информации из разрозненных приложений.

Программный комплекс может применяться в качестве платформы для развертывания баз данных. Microsoft SQL Server 2008 R2 дает возможность работать с большими объемами информации, консолидируя бизнес-сведения из различных источников и обеспечивая высокий уровень производительности и эффективности работы конечных пользователей.

Обслуживание баз данных и запросы пользователей выполняются с наивысшей скоростью, за счет чего повышается качество и результативность функционирования информационной системы предприятия. Microsoft SQL Server 2008 R2 интегрируется с платформой Windows Server 2008 R2, обеспечивая запуск виртуальных дата-центров и баз данных на 256 логических процессоров.

Предусмотрена возможность миграции в облачную, виртуализованную инженерно-техническую инфраструктуру, представляемую по требованию. Microsoft SQL Server 2008 R2 поддерживает 64-разрядные операционные системы, за счет чего повышается скорость работы приложений и баз данных.

Наличие служб мастер-данных SQL Server Master Data Services обеспечивает поддержку целостности информации в гетерогенном окружении. Microsoft SQL Server 2008 R2 производит консолидацию и виртуализацию серверного оборудования в масштабе всей корпоративной сети.

Объем оперативной памяти ограничивается только возможностями установленной операционной системы. Microsoft SQL Server 2008 R2 поддерживает методику сложной обработки событий.

Программный пакет дает возможность производить создание масштабных информационных хранилищ совместно с серверными системами Bull, Dell, EMC, HP, Unisys и IBM. В Microsoft SQL Server 2008 R2 значительно усовершенствована

бизнес-аналитическая составляющая, повышающая качество обслуживания и работы с информационными хранилищами.

Имеется множество инструментов для генерирования стандартной и нерегламентированной отчетной документации различными категориями сотрудников предприятия. Microsoft SQL Server 2008 R2 поддерживает клиентскую систему отчетности в формате Excel и интегрируется с бизнес-аналитическим модулем Sharepoint.

Предусмотрена возможность скоростного создания необходимой отчетной документации. Microsoft SQL Server 2008 R2 позволяет осуществлять публикацию компонентов отчетов в общую библиотеку для их многоразового использования.

Программное решение обеспечивает целостность и доступность всех сохраняемых сведений. Microsoft SQL Server 2008 R2 дает возможность легко находить и повторно использовать требуемые сведения.

Инновационной функцией при составлении геопространственной отчетной документации является применение картографической информации, пользовательских областей и маршрутов. За счет поддержки геопространственных типов Microsoft SQL Server 2008 R2 интегрируется с приложением Microsoft Virtual Earth.

2. Сервер: Windows Server 2008 R2

Microsoft Windows Server 2008 обеспечивает организациям наиболее эффективную платформу для виртуализации рабочих нагрузок, поддержки приложений и защиты сетей. Это безопасная и легко управляемая платформа для разработки и надежного размещения веб-приложений и сервисов. Новые полезные возможности и значительные усовершенствования в ОС Windows Server 2008 по сравнению с предыдущими версиями операционных систем будут востребованы как в небольших рабочих группах и в больших вычислительных центрах.

Сервер 2008 виндовс обеспечивает ИТ-специалистов с больше контроля над серверами и сетевой инфраструктурой, позволяющ они сфокусировать на критических потребностях их организаций. Расширенные возможности сценариев и автоматизации задач, такие как Windows PowerShell, позволяют автоматизировать стандартные ИТ-задачи. Диспетчер серверов позволяет выполнять установку и управление на основе ролей, что облегчает задачи управления и обеспечения безопасности различных ролей сервера на предприятии. В консоли диспетчера сервера унифицировано управление конфигурацией сервера и системной информацией. Возможна установка только необходимых ролей и функций, и многие рутинные задачи развертывания avtomatiziruete мастеров.

Улучшенные инструменты управления системой, такие как монитор производительности и надежности, предоставляют информацию о системах и предупреждают ИТ-персонал о потенциальных проблемах.

В Windows Server 2008 представлен ряд новых и улучшенных технологий безопасности, которые повышают безопасность операционной системы и обеспечивает прочную основу для организации предприятия. Новые функции безопасности, такие как технология PatchGuard, уменьшают площадь контакта с ядром и повышают безопасность и стабильность серверной среды. Ограниченный режим служб Windows предотвращает нарушение работы критически важных служб сервера из-за аномальной активности в файловой системе, реестре или сети и помогает обеспечить безопасность системы. Такие технологии, как защита доступа к сети (NAP), контроллер домена только для чтения (RODC), улучшения инфраструктуры открытых ключей (PKI), ограниченные службы Windows, новый двунаправленный брандмауэр Windows и поддержка шифрования следующего поколения также повышают безопасность операционной системы Windows Server 2008.

Операционная система Windows Server 2008 разработана таким образом, чтобы администраторы могли легко изменять инфраструктуру для адаптации к изменяющимся потребностям организации, сохраняя при этом гибкость инфраструктуры. Возможности мобильных пользователей также улучшены благодаря таким технологиям, как remoteapp и Terminal services gateway, которые позволяют запускать программы из любого удаленного местоположения. Служба развертывания Windows (WDS) в Windows Server 2008 ускоряет развертывание и обслуживание ИТ-систем, а виртуализация Windows server (WSv) упрощает объединение серверов. Контроллер домена только для чтения (RODC) - это новая функция в Windows Server 2008 для установки контроллера домена в удаленном филиале, который позволяет защитить учетные записи пользователей при компрометации контроллера домена.

3. Веб-интерфейс: ASP.NET MVC

Платформа ASP.NET MVC является основой для создания веб-сайтов и веб-приложений путем реализации шаблона MVC.

Концепция pattern (template) MVC (model - view - controller) предполагает разделение приложения на три компонента:

Контроллер (controller) - класс, обеспечивающий связь между пользователем и производительностью системы и хранилищем данных. Он получает введенные пользователем данные и обрабатывает их. И в зависимости от результатов

обработки отправляет пользователю определенный вывод, например, в виде представления.

Представление-это визуальная часть или пользовательский интерфейс самого приложения. Как правило, html-страницы, которые видит пользователь, заходя на сайт.

Модель представляет класс, описывающий логику используемых данных. Общую схему взаимодействия этих компонентов можно представит следующим образом (Рисунок 2):

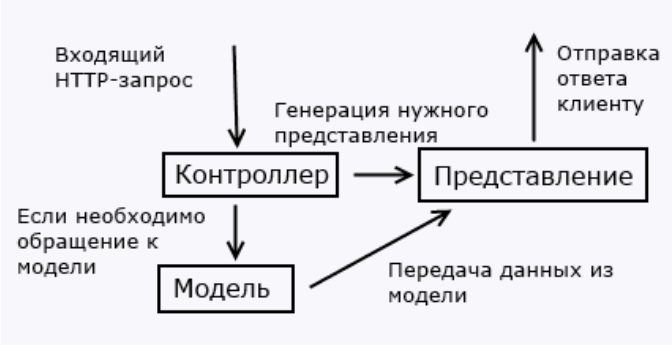

Рисунок 2 – Схема взаимодействия компонентов ASP.NET

Выводы по разделу три

В данном разделе была проанализирована существующая система управления взаимоотношения с клиентами ООО «УралГИС», выявлены основные достоинства и недостатки системы, из проведенного исследования были разработаны основные требования к оптимизации существующей системы, и было принято решение оптимизировать систему самостоятельно, так как ООО «УралГИС» является малым инновационным предприятием ЮУргУ, что предполагает самостоятельную разработку проектов студентами, для получения опыта работы с реальными объектами автоматизации.

4. ОПТИМИЗАЦИЯ БАЗЫ ДАННЫХ

4.1 Описание процесса создания базы данных

Разработанная база данных основана на реляционной модели данных. В данной модели используется набор данных с предопределенными связями между ними. Данные организованы в виде набора таблиц, состоящих из столбцов и строк. Таблица хранит информацию об объекте, представленном в базе данных, например, о заказчике межевых работ, таблица хранит тип данных, такие как текст, дата, число, уникальное значение. Каждая строка в таблице может иметь уникальный идентификатор, называемый первичным ключом, и иметь связь с другими таблицами, с помощью внешнего ключа.

Для того, чтобы исключить дублирование, противоречия, возникающие при обработке данных, таблицы приводятся к нормальной форме Бойса – Кодда, в такой форме таблица имеет два или более потенциальных ключа, которые являются составными, и имеет хотя бы один атрибут.

Создание таблиц в базе данных представлено в Листинге 1:

Листинг 1 – Создание таблиц:

```
USE [uralgis]
GO


CREATE TABLE [Сад_ЮЛ]
(
[Сад_ID] [int] IDENTITY (1,1) NOT NULL,
[Тип_ID] [int] NOT NULL,
[Область] [nvarchar] (30) NOT NULL,
[Город] [nvarchar] (20) NOT NULL,
[Поселок] [nvarchar] (20) NULL,
[Район] [nvarchar] (30) NOT NULL,
[Название] [nvarchar] (20) NOT NULL,
[ОГРН] [nvarchar] (20) NULL,
[Председатель] [nvarchar] (100) NOT NULL,
[Телефон] [nvarchar] (20) NULL,
[Почта] [nvarchar] (50) NULL,
[Статус_ID] [int] NULL,
        CONSTRAINT [PK_Сады] PRIMARY KEY ([Сад_ID] ASC)
Продолжение Листинга 1:
);

CREATE TABLE [Тип_Сад]
(
[Тип_ID] [int] IDENTITY (1,1) NOT NULL,
[Название] [nvarchar] (10) NOT NULL,
        CONSTRAINT [PK_Тип_Сады] PRIMARY KEY ([Тип_ID] ASC)
)
```

Продолжение Листинга 1:

```
CREATE TABLE [Физическое_лицо]
(
[ФЛ_ID] [int] IDENTITY (1,1) NOT NULL,
[Сад_ID] [int] NOT NULL,
[ФИО] [nvarchar] (100) NOT NULL,
[Паспорт_Номер] [nvarchar] (15) NOT NULL,
[Паспорт_Выдан] [nvarchar] (50) NOT NULL,
[Паспорт_Дата] [datetime]  NOT NULL,
[Телефон] [nvarchar] (15) NOT NULL,
[Прописка_НП] [nvarchar] (50) NOT NULL,
[Прописка_Адрес] [nvarchar] (100) NOT NULL,
[Статус_ID] [int] NULL,
        CONSTRAINT [PK_Заказачик] PRIMARY KEY ([ФЛ_ID] ASC)
);

CREATE TABLE [Документы_ФЛ]
(
   [Прием_ID] [int] NOT NULL,
[Договор] [nvarchar] (10) NULL,
[Квитанция] [int] NULL,
[Квитанция_Газета] [int] NULL,
[ГосАкт_Серия] [nvarchar] (20) NULL,
[ГосАкт_Номер] [nvarchar] (20) NULL,
[СвидетельствоЗУ_Серия] [nvarchar] (10) NULL,
[СвидетельствоЗУ_Номер] [nvarchar] (10) NULL,
[СвидетельствоДом_Серия] [nvarchar] (10) NULL,

[СвидетельствоДом_Номер] [nvarchar] (10) NULL,
[СвидетельствоГараж_Серия] [nvarchar] (10) NULL,
[СвидетельствоГараж_Номер] [nvarchar] (10) NULL,
[СвидетельствоХоз_Серия] [nvarchar] (10) NULL,
[СвидетельствоХоз_Номер] [nvarchar] (10) NULL,
);


CREATE TABLE [Должность]
(
[Должность_ID] [int] IDENTITY (1,1) NOT NULL,
[Название] [nvarchar] (20) NOT NULL,
        CONSTRAINT [PK_Должность] PRIMARY KEY ([Должность_ID] ASC)
);

CREATE TABLE [Подпись]
(
[Подпись_ID] [int] IDENTITY (1,1) NOT NULL,
[ФИО] [varchar] (100) NOT NULL,
[Паспорт_Номер] [nvarchar] (15) NOT NULL,
[Паспорт_Выдан] [nvarchar] (50) NOT NULL,
[Паспорт_Дата] [datetime]  NOT NULL,
[Телефон] [varchar] (15) NOT NULL,
```

Продолжение Листинга 1:

```
[Прописка] [nvarchar] (100) NOT NULL,
[Связь_Адрес] [nvarchar] (100) NOT NULL,
[Связь_Почта] [nvarchar] (50) NOT NULL,
[Номер_реестр] [varchar] NOT NULL,
[СНИЛС] [nvarchar] (20) NOT NULL,
[Юридическое_лицо] [nvarchar] (100) NULL,
[СРО] [nvarchar] (100) NOT NULL,
[ГСРОКИ_номер] [int] NOT NULL,
[ГСРОКИ_дата] [datetime]  NOT NULL,
[Свидетельство_СРО_Номер] [int] NOT NULL,
[Свидетельство_СРО_Дата] [datetime]  NOT NULL,
```

Продолжение Листинга 1:

```
[Номер_аттестат] [nvarchar] (15) NOT NULL,
        CONSTRAINT [PK_Подпись] PRIMARY KEY ([Подпись_ID] ASC)
);

CREATE TABLE [Сотрудник]
(
[Сотрудник_ID] [int] IDENTITY (1,1) NOT NULL,
[ФИО] [nvarchar] (100) NOT NULL,
[Должность_ID] [int] NOT NULL,
[Телефон] [varchar] (15) NOT NULL,
        CONSTRAINT[PK_Сотрудник] PRIMARY KEY ([Сотрудник_ID] ASC)
);

CREATE TABLE [Статус]
(
[Статус_ID] [int] IDENTITY (1,1) NOT NULL,
[Статус] [nvarchar] (50) NOT NULL,
        CONSTRAINT [PK_Статус] PRIMARY KEY ([Статус_ID] ASC)
);


CREATE TABLE [Прием]
(
[Прием_ID] [int] IDENTITY (1,1) NOT NULL,
[Сотрудник_ID] [int] NOT NULL,
[ФЛ_ID] [int] NOT NULL,
[Сад_ID] [int] NOT NULL,
[Дата] [datetime] NOT NULL CONSTRAINT [DF_Прием_Дата] DEFAULT (getdate()),
[Комментарий] [nvarchar] (200) NULL,
[Координаты] [nvarchar] (1000) NULL,
        CONSTRAINT [PK_Прием] PRIMARY KEY ([Прием_ID])
);



CREATE TABLE [Прием_АС]
(
```

Продолжение Листинга 1:

```
[Сотрудник_ID] [int] NOT NULL,
[Прием_ID] [int] NOT NULL,
[Дата] [datetime] NOT NULL CONSTRAINT [DF_Прием_АС_Дата] DEFAULT (getdate())
);

CREATE TABLE [Межевание]
(
[Сотрудник_ID] [int]  NOT NULL,
[Прием_ID] [int] NOT NULL,
[Подпись_ID] [int] NOT NULL,
[Дата_ЗКП]  [datetime]  NULL  CONSTRAINT  [DF_Межевание_Дата_ЗКП]  DEFAULT
(getdate()),
[Дата_Приостановка]  [datetime]  NULL  CONSTRAINT
[DF_Межевание_Дата_Приостановка] DEFAULT (getdate()),
[Номер_Заявление]  [varchar] (30)  NULL,
[Дата_Учет]  [datetime]   NULL  CONSTRAINT  [DF_Межевание_Дата_Учет]  DEFAULT
(getdate()),
[Оплата_ID] [int]  NULL,

);


CREATE TABLE [Выписки]
(
[Выписка_ID] [int] IDENTITY (1,1) NOT NULL,
[Номер] [varchar] (10) NULL,
[Код] [varchar] (10) NULL,
[Прием_ID] [int]  NULL,
[Сотрудник_ID] [int]  NULL,
[Дата_Заказ] [datetime]  NOT NULL CONSTRAINT [DF_Выписки_Дата_Заказ] DEFAULT
(getdate()),


[Дата_Готовность]  [datetime]   NULL  CONSTRAINT  [DF_Выписки_Дата_Готовность]
DEFAULT (getdate()),
[Готовность] [int]  NOT NULL,
                CONSTRAINT[PK_Выписки] PRIMARY KEY ([Выписка_ID] ASC)
);


CREATE TABLE [Газета]
(
[Газета_ID] [int] IDENTITY (1,1) NOT NULL,
[Сотрудник_ID] [int]  NULL,
[Подпись_ID] [int]  NULL,
[Прием_ID] [int]  NULL,
[Дата] [datetime]  NOT NULL CONSTRAINT [DF_Газета_Дата] DEFAULT (getdate()),
[Готовность] [int]  NOT NULL,
                CONSTRAINT[PK_Газета] PRIMARY KEY ([Газета_ID] ASC)
);
```

Продолжение Листинга 1:

```
ALTER TABLE [Физическое_лицо]
	ADD CONSTRAINT [FK_Физическое_лицо_Сад_ЮЛ] FOREIGN KEY ([Сад_ID])
		REFERENCES [Сад_ЮЛ] ([Сад_ID]);

ALTER TABLE [Заказчик]
	ADD CONSTRAINT [FK_Заказчик_Результат] FOREIGN KEY ([Результат_ID])
		REFERENCES [Результат] ([Результат_ID]);

ALTER TABLE [Заказчик]
	ADD CONSTRAINT [FK_Заказчик_Статус] FOREIGN KEY ([Статус_ID])
		REFERENCES [Статус] ([Статус_ID]);
```

Продолжение Листинга 1:

```
ALTER TABLE [ЗОП]
	ADD CONSTRAINT [FK_ЗОП_Сады] FOREIGN KEY ([Сад_ID])
		REFERENCES [Сады] ([Сад_ID]);

ALTER TABLE [ЗОП]
	ADD CONSTRAINT [FK_ЗОП_Сотрудник] FOREIGN KEY ([Сотрудник_ID])
		REFERENCES [Сотрудник] ([Сотрудник_ID]);

ALTER TABLE [Сотрудник]
	ADD CONSTRAINT [FK_Сотрудник_Должность] FOREIGN KEY
([Должность_ID])
		REFERENCES [Должность] ([Должность_ID]);

ALTER TABLE [Прием]
	ADD CONSTRAINT [FK_Прием_Сотрудник] FOREIGN KEY ([Сотрудник_ID])
		REFERENCES [Сотрудник] ([Сотрудник_ID]);
ALTER TABLE [Прием]
	ADD CONSTRAINT [FK_Прием_Заказчик] FOREIGN KEY ([Заказчик_ID])
		REFERENCES [Заказчик] ([Заказчик_ID]);

ALTER TABLE [Прием_АС]
	ADD CONSTRAINT [FK_Прием_АС_Сады] FOREIGN KEY ([Сотрудник_ID])
		REFERENCES [Сотрудник] ([Сотрудник_ID]);

ALTER TABLE [Прием_АС]
	ADD CONSTRAINT [FK_Прием_АС_Заказчик] FOREIGN KEY ([Заказчик_ID])
		REFERENCES [Заказчик] ([Заказчик_ID]);

ALTER TABLE [Межевание]
```

Продолжение Листинга 1:

```
	ADD CONSTRAINT [FK_Межевание_Сотрудник] FOREIGN KEY
([Сотрудник_ID])
		REFERENCES [Сотрудник] ([Сотрудник_ID]);
```

Окончание Листинга 1:

```
ALTER TABLE [Межевание]
	ADD CONSTRAINT [FK_Межевание_Подпись] FOREIGN KEY ([Подпись_ID])
		REFERENCES [Подпись] ([Подпись_ID]);

ALTER TABLE [Межевание]
	ADD CONSTRAINT [FK_Межевание_Оплата] FOREIGN KEY ([Оплата_ID])
		REFERENCES [Оплата] ([Оплата_ID]);

ALTER TABLE [Выписки]
	ADD CONSTRAINT [FK_Выписки_Заказчик] FOREIGN KEY ([Заказчик_ID])
		REFERENCES [Заказчик] ([Заказчик_ID]);

ALTER TABLE [Выписки]
	ADD CONSTRAINT [FK_Выписки_Сотрудник] FOREIGN KEY ([Сотрудник_ID])
		REFERENCES [Сотрудник] ([Сотрудник_ID]);


ALTER TABLE [Газета]
	ADD CONSTRAINT [FK_Газета_Сотрудник] FOREIGN KEY ([Сотрудник_ID])
		REFERENCES [Сотрудник] ([Сотрудник_ID]);

ALTER TABLE [Газета]
	ADD CONSTRAINT [FK_Газета_Подпись] FOREIGN KEY ([Подпись_ID])
		REFERENCES [Подпись] ([Подпись_ID]);



ALTER TABLE [Газета]
	ADD CONSTRAINT [FK_Газета_Заказчик] FOREIGN KEY ([Заказчик_ID])
		REFERENCES [Заказчик] ([Заказчик_ID]);
```

Здесь конструкция CREATE TABLE отвечает за создание таблицы, CONSTRAINT за создание первичного ключа, а ALTER TABLE за создание внешнего ключа в базе данных.

Для того, чтобы свести к минимальному значению ошибки, создаваемые пользователем, некоторым значениям в таблице присваиваются уникальные значения, при помощи конструкции CREATE UNIQUE INDEX. Создание уникальных значений описывается в Листинге 2.

Листинг 2 – Создание уникальных значений:

```
USE [uralgis]
GO

CREATE UNIQUE INDEX [IX_Название] ON [Сады]
(
[Название] ASC
);
```

Продолжение Листинга 2:

```
CREATE UNIQUE INDEX [IX_ОГРН] ON [Сады]
(
[ОГРН] ASC
);

CREATE UNIQUE INDEX [IX_Паспорт_Номер] ON [Заказчик]
(
[Паспорт_Номер] ASC
);

CREATE UNIQUE INDEX [IX_Результат] ON [Результат]
(
[Результат] ASC
);
```

Продолжение Листинга 2:

```
CREATE UNIQUE INDEX [IX_Статус] ON [Статус]
(
[Статус] ASC
);

CREATE UNIQUE INDEX [IX_Номер] ON [Выписки]
(
[Номер] ASC
);

CREATE UNIQUE INDEX [IX_Код] ON [Выписки]
(
[Код] ASC
);

CREATE UNIQUE INDEX [IX_Договор] ON [Прием]
(
[Договор] ASC
);

CREATE UNIQUE INDEX [IX_Номер_заявление] ON [Межевание]
(
[Номер_заявление] ASC
);

CREATE UNIQUE INDEX [IX_Оплата] ON [Межевание]
(
[Оплата] ASC
);

CREATE UNIQUE INDEX [IX_Название] ON [Должность]
(
[Название] ASC
);
```

ОкончаниеЛистинга 2:

```
CREATE UNIQUE INDEX [IX_Паспорт] ON [Подпись]
(
[Паспорт_номер] ASC
);

CREATE UNIQUE INDEX [IX_Номер_реестр] ON [Подпись]
(
[Номер_реестр] ASC
);

CREATE UNIQUE INDEX [IX_СНИЛС] ON [Подпись]
(
[СНИЛС] ASC
);
CREATE UNIQUE INDEX [IX_ГСРОКИ_номер] ON [Подпись]
(
[ГСРОКИ_номер] ASC
);
CREATE UNIQUE INDEX [IX_ГСРОКИ_дата] ON [Подпись]
(
[ГСРОКИ_дата] ASC
);

CREATE UNIQUE INDEX [IX_Свидетельство_СРО_Номер] ON [Подпись]
(
[Свидетельство_СРО_Номер] ASC
);

CREATE UNIQUE INDEX [IX_Свидетельство_СРО_Дата] ON [Подпись]
(
[Свидетельство_СРО_Дата] ASC
);
```

4.2 Детальное описание разработки требований к оптимизации системы

Web-приложение UralGISApp – ключевое приложение, объединяющее все компоненты. Через него осуществляется работа с данными, их анализ и формирование веб-страниц для пользователя, создание и редактирование данных. Более подробно структура приложения представлена на рисунке 3.

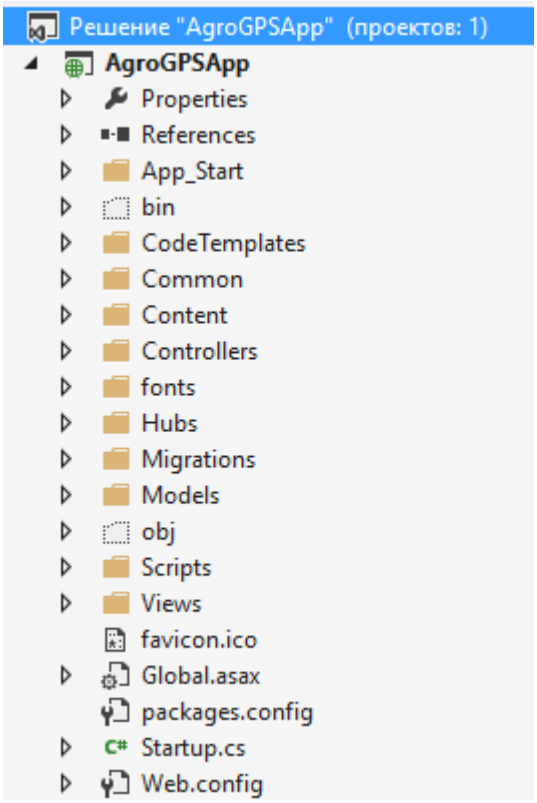

Рисунок 3 – Структура веб-приложения

Веб-приложение UralGIS написано на языке C#. Среда разработки – Visual Studio 2013 for Web (но может быть использована и более новая Visual Studio 2017 Community Edition). Для сборки и внесения изменений требуется .NET Framework версии 4.5, а также фреймворк ASP.NET MVC – это следует учитывать при установке среды разработки.

Конфигурация системы подробно представлена в Листинге 3:

Листинг 3 – Конфигурационный файл системы:

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=301880
  -->
<configuration>
  <configSections>
```

Продолжение Листинга 3:

```
  <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->

  <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />

 </configSections>

 <connectionStrings>

  <add name="DefaultConnection" connectionString="Data Source=(LocalDb)\v11.0;AttachDbFilename=|DataDirectory|\aspnet-uralGIS-20180612075738.mdf;Initial Catalog=aspnet-uralGIS-20180612075738;Integrated Security=True"

    providerName="System.Data.SqlClient" />

 </connectionStrings>

 <appSettings>

  <add key="webpages:Version" value="3.0.0.0" />

  <add key="webpages:Enabled" value="false" />

  <add key="ClientValidationEnabled" value="true" />

  <add key="UnobtrusiveJavaScriptEnabled" value="true" />

 </appSettings>

 <system.web>

  <authentication mode="None" />

  <compilation debug="true" targetFramework="4.5" />

  <httpRuntime targetFramework="4.5" />

 </system.web>

 <system.webServer>

  <modules>

   <remove name="FormsAuthentication" />

  </modules>

 </system.webServer>

 <runtime>

  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">

   <dependentAssembly>
```

Продолжение Листинга 3:

```
      <assemblyIdentity name="Microsoft.Owin" publicKeyToken="31bf3856ad364e35" />
      <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="Microsoft.Owin.Security.OAuth" publicKeyToken="31bf3856ad364e35" />
      <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="Microsoft.Owin.Security.Cookies" publicKeyToken="31bf3856ad364e35" />
      <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="Microsoft.Owin.Security" publicKeyToken="31bf3856ad364e35" />
      <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="Newtonsoft.Json" culture="neutral" publicKeyToken="30ad4fe6b2a6aeed" />
      <bindingRedirect oldVersion="0.0.0.0-6.0.0.0" newVersion="6.0.0.0" />
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="System.Web.Helpers" publicKeyToken="31bf3856ad364e35" />
      <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="System.Web.Mvc" publicKeyToken="31bf3856ad364e35" />
      <bindingRedirect oldVersion="1.0.0.0-5.2.2.0" newVersion="5.2.2.0" />
    </dependentAssembly>
    <dependentAssembly>
```

Окончание Листинга 3:

```
      <assemblyIdentity name="System.Web.Optimization" publicKeyToken="31bf3856ad364e35" />
      <bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="1.1.0.0" />
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="System.Web.WebPages" publicKeyToken="31bf3856ad364e35" />
      <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="WebGrease" publicKeyToken="31bf3856ad364e35" />
      <bindingRedirect oldVersion="1.0.0.0-1.5.2.14234" newVersion="1.5.2.14234" />
    </dependentAssembly>
  </assemblyBinding>
</runtime>
<entityFramework>
  <defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory, EntityFramework">
    <parameters>
      <parameter value="mssqllocaldb" />
    </parameters>
  </defaultConnectionFactory>
  <providers>
    <provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
  </providers>
</entityFramework>
</configuration>
```

Рассмотрим детально реализацию требований к оптимизации системы.

1. Поддержка личных учетных записей у сотрудников.

Реализацию личных учетных записей описывают компоненты (Контроллер, Модель, Представление). Их назначение описано в разделе 3.3, а реализация представлена на Рисунке 4:

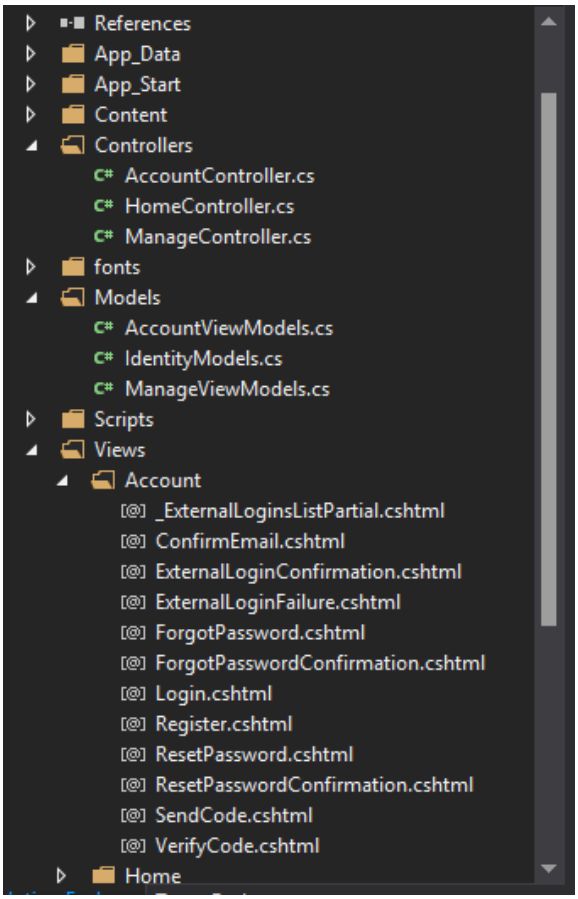

Рисунок 4 – Реализация поддержки личных учетных записей

Рассмотрим реализацию регистрации нового пользователя. В листинге 4 описано создание модели, отвечающей за логику используемых данных:

Листинг 4 – Модель регистрации нового пользователя системы:

```
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace uralGIS.Models
{
    public class ExternalLoginConfirmationViewModel
    {
        [Required]
        [Display(Name = "Email")]
        public string Email { get; set; }
    }

    public class ExternalLoginListViewModel
    {
        public string ReturnUrl { get; set; }
    }
```

Продолжение Листинга 4:

```
public class SendCodeViewModel
{
    public string SelectedProvider { get; set; }
    public ICollection<System.Web.Mvc.SelectListItem> Providers { get; set; }
    public string ReturnUrl { get; set; }
    public bool RememberMe { get; set; }
}

public class VerifyCodeViewModel
{
    [Required]
    public string Provider { get; set; }

    [Required]
    [Display(Name = "Code")]
    public string Code { get; set; }
    public string ReturnUrl { get; set; }

    [Display(Name = "Запомнить браузер?")]
    public bool RememberBrowser { get; set; }

    public bool RememberMe { get; set; }
}

public class ForgotViewModel
{
    [Required]
    [Display(Name = "Email")]
    public string Email { get; set; }
}

public class LoginViewModel
{
    [Required]
    [Display(Name = "Email")]
    [EmailAddress]
    public string Email { get; set; }

    [Required]
    [DataType(DataType.Password)]
    [Display(Name = "Пароль")]
    public string Password { get; set; }

    [Display(Name = "Запомнить меня?")]
    public bool RememberMe { get; set; }
}

public class RegisterViewModel
{
    [Required]
```

Окончание Листинга 4:

```
        [EmailAddress]
        [Display(Name = "Email")]
        public string Email { get; set; }

        [Required]
        [StringLength(100, ErrorMessage = "Максимальная {0} длина пароля {2} символа.",
MinimumLength = 6)]
        [DataType(DataType.Password)]
        [Display(Name = "Пароль")]
        public string Password { get; set; }

        [DataType(DataType.Password)]
        [Display(Name = "Подтвердить пароль")]
        [Compare("Password", ErrorMessage = "Пароли не совпадают.")]
        public string ConfirmPassword { get; set; }
    }

    public class ResetPasswordViewModel
    {
        [Required]
        [EmailAddress]
        [Display(Name = "Email")]
        public string Email { get; set; }

        [Required]
        [StringLength(100, ErrorMessage = "Максимальная {0} длина пароля {2} символа.",
MinimumLength = 6)]
        [DataType(DataType.Password)]
        [Display(Name = "Пароль")]
        public string Password { get; set; }

        [DataType(DataType.Password)]
        [Display(Name = "Подтвердить пароль")]
        [Compare("Password", ErrorMessage = "Пароли не совпадают.")]
        public string ConfirmPassword { get; set; }

        public string Code { get; set; }
    }

    public class ForgotPasswordViewModel
    {
        [Required]
        [EmailAddress]
        [Display(Name = "Email")]
        public string Email { get; set; }
    }
}
```

Далее рассмотрим описание создания контроллера регистрации, обеспечивающего связь между пользователем и системой (Листинг 5):

Листинг 5 – Описание контроллера регистрации нового пользователя системы:

```
namespace uralGIS.Controllers
{
   [Authorize]
   public class AccountController : Controller
   {
Продолжение Листинга 4:
      private ApplicationSignInManager _signInManager;
      private ApplicationUserManager _userManager;

      public AccountController()
      {
      }

      public AccountController(ApplicationUserManager userManager, ApplicationSignInManager
signInManager )
      {
         UserManager = userManager;
         SignInManager = signInManager;
      }

      public ApplicationSignInManager SignInManager
      {
         get
         {
            return_signInManager HttpContext.GetOwinContext().Get<ApplicationSignInManager>();
         }
         private set
         {
            _signInManager = value;
         }
      }

      public ApplicationUserManager UserManager
      {
         get
         {
            return_userManager
HttpContext.GetOwinContext().GetUserManager<ApplicationUserManager>();
         }
         private set
         {

            _userManager = value;
         }
      }

      //
      // GET: /Account/Login
```

Продолжение Листинга 5:

```
 [AllowAnonymous]
     public ActionResult Login(string returnUrl)
     {
        ViewBag.ReturnUrl = returnUrl;
        return View();
     }

     //
     // POST: /Account/Login
     [HttpPost]
     [AllowAnonymous]
     [ValidateAntiForgeryToken]
     public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)
     {
        if (!ModelState.IsValid)
        {
           return View(model);
        }

        var result = await SignInManager.PasswordSignInAsync(model.Email, model.Password,
model.RememberMe, shouldLockout: false);
        switch (result)
        {
           case SignInStatus.Success:
              return RedirectToLocal(returnUrl);
           case SignInStatus.LockedOut:

              return View("Lockout");
           case SignInStatus.RequiresVerification:
              return RedirectToAction("SendCode", new { ReturnUrl = returnUrl, RememberMe =
model.RememberMe });
           case SignInStatus.Failure:
           default:
              ModelState.AddModelError("", "Invalid login attempt.");
              return View(model);
        }
     }

     //
     // GET: /Account/VerifyCode
     [AllowAnonymous]
     public async Task<ActionResult> VerifyCode(string provider, string returnUrl, bool
rememberMe)
     {
      login
        if (!await SignInManager.HasBeenVerifiedAsync())
        {
           return View("Error");
        }
```

Продолжение Листинга 5:

```
        return View(new VerifyCodeViewModel { Provider = provider, ReturnUrl = returnUrl,
RememberMe = rememberMe });
    }

    //
    // POST: /Account/VerifyCode
    [HttpPost]
    [AllowAnonymous]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult> VerifyCode(VerifyCodeViewModel model)
    {
      if (!ModelState.IsValid)
      {

        return View(model);
      }

      var result = await SignInManager.TwoFactorSignInAsync(model.Provider, model.Code,
isPersistent: model.RememberMe, rememberBrowser: model.RememberBrowser);
      switch (result)
      {
        case SignInStatus.Success:
          return RedirectToLocal(model.ReturnUrl);
        case SignInStatus.LockedOut:
          return View("Lockout");
        case SignInStatus.Failure:
        default:
          ModelState.AddModelError("", "Invalid code.");
          return View(model);
      }
    }

    //
    // GET: /Account/Register
    [AllowAnonymous]
    public ActionResult Register()
    {
      return View();
    }

    //
    // POST: /Account/Register
    [HttpPost]
    [AllowAnonymous]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult> Register(RegisterViewModel model)
    {
      if (ModelState.IsValid)
```

Продолжение Листинга 5:

```
            {
                var user = new ApplicationUser { UserName = model.Email, Email = model.Email };
                var result = await UserManager.CreateAsync(user, model.Password);
                if (result.Succeeded)
                {
                    await SignInManager.SignInAsync(user, isPersistent:false, rememberBrowser:false);

                    // string code = await UserManager.GenerateEmailConfirmationTokenAsync(user.Id);
                    // var callbackUrl = Url.Action("ConfirmEmail", "Account", new { userId = user.Id, code
= code }, protocol: Request.Url.Scheme);
                    // await UserManager.SendEmailAsync(user.Id, "Confirm your account", "Please confirm
your account by clicking <a href=\"" + callbackUrl + "\">here</a>");

                    return RedirectToAction("Index", "Home");
                }
                AddErrors(result);
            }

            // If we got this far, something failed, redisplay form
            return View(model);
        }

        //
        // GET: /Account/ConfirmEmail
        [AllowAnonymous]
        public async Task<ActionResult> ConfirmEmail(string userId, string code)
        {
            if (userId == null || code == null)
            {
                return View("Error");
            }
            var result = await UserManager.ConfirmEmailAsync(userId, code);
            return View(result.Succeeded ? "ConfirmEmail" : "Error");
        }

        //
        // GET: /Account/ForgotPassword
        [AllowAnonymous]
        public ActionResult ForgotPassword()
        {
            return View();
        }

        //
        // POST: /Account/ForgotPassword
        [HttpPost]
        [AllowAnonymous]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult> ForgotPassword(ForgotPasswordViewModel model)
```

Продолжение Листинга 5:

```
        {
            if (ModelState.IsValid)
            {
                var user = await UserManager.FindByNameAsync(model.Email);
                if (user == null || !(await UserManager.IsEmailConfirmedAsync(user.Id)))
                {
                    return View("ForgotPasswordConfirmation");
                }

                // string code = await UserManager.GeneratePasswordResetTokenAsync(user.Id);
                // var callbackUrl = Url.Action("ResetPassword", "Account", new { userId = user.Id, code =
code }, protocol: Request.Url.Scheme);
                //  await UserManager.SendEmailAsync(user.Id, "Reset Password", "Please reset your
password by clicking <a href=\"" + callbackUrl + "\">here</a>");
                // return RedirectToAction("ForgotPasswordConfirmation", "Account");
            }

            return View(model);
        }

        //
        // GET: /Account/ForgotPasswordConfirmation
        [AllowAnonymous]
        public ActionResult ForgotPasswordConfirmation()
        {
            return View();
        }

        //
        // GET: /Account/ResetPassword
        [AllowAnonymous]
        public ActionResult ResetPassword(string code)
        {
            return code == null ? View("Error") : View();
        }

        //
        // POST: /Account/ResetPassword
        [HttpPost]
        [AllowAnonymous]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult> ResetPassword(ResetPasswordViewModel model)
        {
            if (!ModelState.IsValid)
            {
                return View(model);
            }
            var user = await UserManager.FindByNameAsync(model.Email);
            if (user == null)
```

Продолжение Листинга 5:

```
            return RedirectToAction("ResetPasswordConfirmation", "Account");
        }
        var result = await UserManager.ResetPasswordAsync(user.Id, model.Code, model.Password);
        if (result.Succeeded)
        {
            return RedirectToAction("ResetPasswordConfirmation", "Account");
        }
        AddErrors(result);
        return View();
    }

    //
    // GET: /Account/ResetPasswordConfirmation
    [AllowAnonymous]
    public ActionResult ResetPasswordConfirmation()
    {
        return View();
    }

    //
    // POST: /Account/ExternalLogin
    [HttpPost]
    [AllowAnonymous]
    [ValidateAntiForgeryToken]
    public ActionResult ExternalLogin(string provider, string returnUrl)
    {
        // Request a redirect to the external login provider
        return new ChallengeResult(provider, Url.Action("ExternalLoginCallback", "Account", new {
ReturnUrl = returnUrl }));
    }

    //
    // GET: /Account/SendCode
    [AllowAnonymous]
    public async Task<ActionResult> SendCode(string returnUrl, bool rememberMe)
    {
        var userId = await SignInManager.GetVerifiedUserIdAsync();
        if (userId == null)
        {
            return View("Error");
        }
        var userFactors = await UserManager.GetValidTwoFactorProvidersAsync(userId);
        var factorOptions = userFactors.Select(purpose => new SelectListItem { Text = purpose, Value
= purpose }).ToList();
        return View(new SendCodeViewModel { Providers = factorOptions, ReturnUrl = returnUrl,
RememberMe = rememberMe });
    }

    //
    // POST: /Account/SendCode
```

Продолжение Листинга 5:

```
    [HttpPost]
     [AllowAnonymous]
     [ValidateAntiForgeryToken]
     public async Task<ActionResult> SendCode(SendCodeViewModel model)
     {
        if (!ModelState.IsValid)
        {
           return View();
        }

        // Generate the token and send it
        if (!await SignInManager.SendTwoFactorCodeAsync(model.SelectedProvider))
        {
           return View("Error");
        }
        return RedirectToAction("VerifyCode", new { Provider = model.SelectedProvider, ReturnUrl =
model.ReturnUrl, RememberMe = model.RememberMe });
     }

     //
     // GET: /Account/ExternalLoginCallback
     [AllowAnonymous]
     public async Task<ActionResult> ExternalLoginCallback(string returnUrl)
     {
        var loginInfo = await AuthenticationManager.GetExternalLoginInfoAsync();
        if (loginInfo == null)
        {
           return RedirectToAction("Login");
        }

        // Sign in the user with this external login provider if the user already has a login
        var result = await SignInManager.ExternalSignInAsync(loginInfo, isPersistent: false);
        switch (result)
        {
           case SignInStatus.Success:
              return RedirectToLocal(returnUrl);
           case SignInStatus.LockedOut:
              return View("Lockout");
           case SignInStatus.RequiresVerification:
              return RedirectToAction("SendCode", new { ReturnUrl = returnUrl, RememberMe = false
});
           case SignInStatus.Failure:
           default:
              // If the user does not have an account, then prompt the user to create an account
              ViewBag.ReturnUrl = returnUrl;
              ViewBag.LoginProvider = loginInfo.Login.LoginProvider;
              return View("ExternalLoginConfirmation", new ExternalLoginConfirmationViewModel {
Email = loginInfo.Email });
        }
     }
```

Продолжение Листинга 5:

```
// POST: /Account/ExternalLoginConfirmation
    [HttpPost]
    [AllowAnonymous]
    [ValidateAntiForgeryToken]
    public                                  async                                  Task<ActionResult>
ExternalLoginConfirmation(ExternalLoginConfirmationViewModel model, string returnUrl)
    {
      if (User.Identity.IsAuthenticated)
      {
        return RedirectToAction("Index", "Manage");
      }

      if (ModelState.IsValid)
      {
        // Get the information about the user from the external login provider
        var info = await AuthenticationManager.GetExternalLoginInfoAsync();
        if (info == null)
        {
          return View("ExternalLoginFailure");
        }
        var user = new ApplicationUser { UserName = model.Email, Email = model.Email };
        var result = await UserManager.CreateAsync(user);
        if (result.Succeeded)
        {
          result = await UserManager.AddLoginAsync(user.Id, info.Login);
          if (result.Succeeded)
          {
            await SignInManager.SignInAsync(user, isPersistent: false, rememberBrowser: false);
            return RedirectToLocal(returnUrl);
          }
        }
        AddErrors(result);
      }

      ViewBag.ReturnUrl = returnUrl;
      return View(model);
    }

    //
    // POST: /Account/LogOff
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult LogOff()
    {
      AuthenticationManager.SignOut();
      return RedirectToAction("Index", "Home");
    }

    //
    // GET: /Account/ExternalLoginFailure
    [AllowAnonymous]
```

Продолжение Листинга 5:

```
public ActionResult ExternalLoginFailure()
        {
            return View();
        }

        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                if (_userManager != null)
                {
                    _userManager.Dispose();
                    _userManager = null;
                }
                if (_signInManager != null)
                {
                    _signInManager.Dispose();
                    _signInManager = null;
                }
            }

            base.Dispose(disposing);
        }

        #region Helpers
        private const string XsrfKey = "XsrfId";

        private IAuthenticationManager AuthenticationManager
        {
            get
            {
                return HttpContext.GetOwinContext().Authentication;
            }
        }

        private void AddErrors(IdentityResult result)
        {
            foreach (var error in result.Errors)
            {
                ModelState.AddModelError("", error);
            }
        }

        private ActionResult RedirectToLocal(string returnUrl)
        {
            if (Url.IsLocalUrl(returnUrl))
            {
```

Окончание Листинга 5:

```
            return Redirect(returnUrl);
        }
        return RedirectToAction("Index", "Home");
    }

    internal class ChallengeResult : HttpUnauthorizedResult
    {
        public ChallengeResult(string provider, string redirectUri)
            : this(provider, redirectUri, null)
        {
        }

        public ChallengeResult(string provider, string redirectUri, string userId)
        {
            LoginProvider = provider;
            RedirectUri = redirectUri;
            UserId = userId;
        }

        public string LoginProvider { get; set; }
        public string RedirectUri { get; set; }
        public string UserId { get; set; }

        public override void ExecuteResult(ControllerContext context)
        {
            var properties = new AuthenticationProperties { RedirectUri = RedirectUri };
            if (UserId != null)
            {
                properties.Dictionary[XsrfKey] = UserId;
            }
            context.HttpContext.GetOwinContext().Authentication.Challenge(properties,
LoginProvider);
        }
    }
    #endregion }}
```

Процесс создания представления, интерфейса, который видит пользователь описан в Листинге 6:

Листинг 6 – Создание представления для регистрации нового пользователя системы:

```
@model uralGIS.Models.RegisterViewModel
@{
   ViewBag.Title = "Register";
}

<h2>@ViewBag.Title.</h2>

@using (Html.BeginForm("Register", "Account", FormMethod.Post, new { @class = "form-
horizontal", role = "form" }))
```

Окончание Листинга 6:

```
{
   @Html.AntiForgeryToken()
   <h4>Create a new account.</h4>
   <hr />
   @Html.ValidationSummary("", new { @class = "text-danger" })
   <div class="form-group">
      @Html.LabelFor(m => m.Email, new { @class = "col-md-2 control-label" })
      <div class="col-md-10">
         @Html.TextBoxFor(m => m.Email, new { @class = "form-control" })
      </div>
   </div>
   <div class="form-group">
      @Html.LabelFor(m => m.Password, new { @class = "col-md-2 control-label" })
      <div class="col-md-10">
         @Html.PasswordFor(m => m.Password, new { @class = "form-control" })
      </div>
   </div>
   <div class="form-group">
      @Html.LabelFor(m => m.ConfirmPassword, new { @class = "col-md-2 control-label" })
      <div class="col-md-10">
         @Html.PasswordFor(m => m.ConfirmPassword, new { @class = "form-control" })
      </div>
   </div>
   <div class="form-group">
      <div class="col-md-offset-2 col-md-10">
         <input type="submit" class="btn btn-default" value="Register" />
      </div>
   </div>
}

@section Scripts {
   @Scripts.Render("~/bundles/jqueryval")
```

После сборки программы пользователь в браузере видит следующую страницу, представлена на рисунке 5:

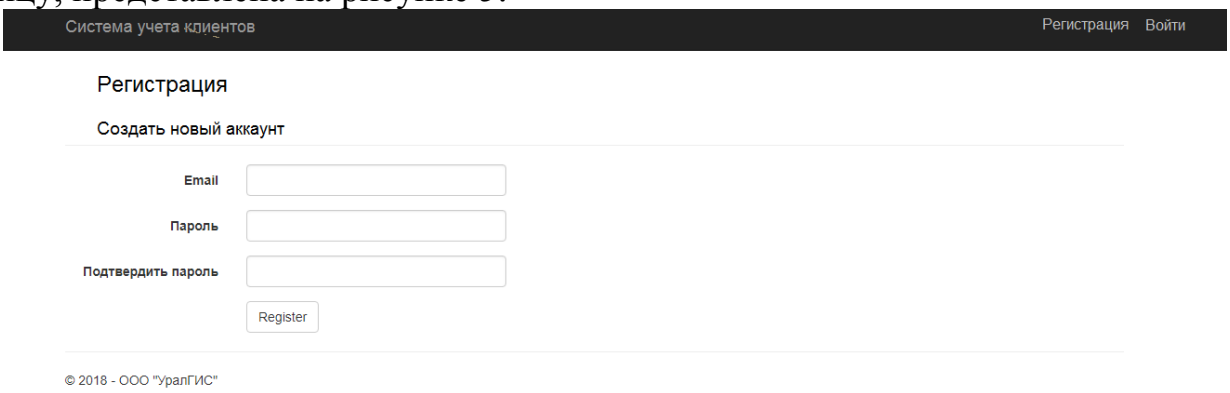

Рисунок 5 – Страница регистрации

При неверных действиях отображаются системные сообщения (Рисунок 6):

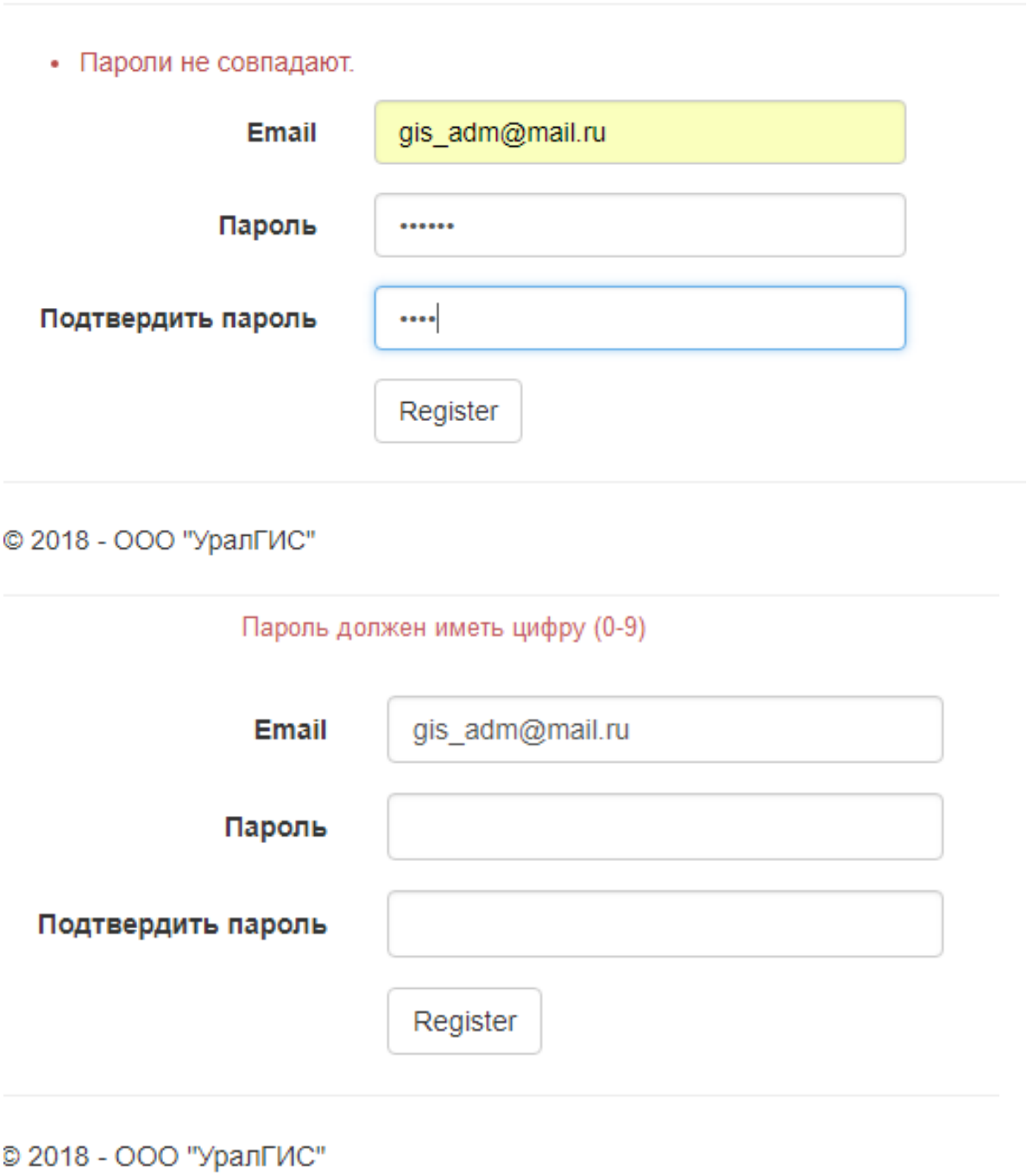

Рисунок 6 – Системные сообщения

При успешной регистрации пользователь видит шапку сайта, со своим именем (Рисунок 7):

Рисунок 7 – Успешная регистрация

2. Резервное копирование базы данных.

Резервная копия базы данных создается в Micrsoft SQL Server (Листинг 7):

Листинг 7 – Создание резервной копии базы данных:

```
USE UralGIS;
GO
BACKUP DATABASE UralGIS
TO DISK = 'Z:\SQLServerBackups\UralGIS.Bak'
  WITH FORMAT,
    MEDIANAME = 'Z_SQLServerBackups',
    NAME = 'UralGIS13062018';
GO
```

- значение database является базой данных для резервного копирования;
- значение backup_device указывает список устройств резервного копирования, используемых для создания резервной копии. Можно указать как физическое устройство резервного копирования, так и соответствующее логическое устройство, если оно уже определено. Для указания физического устройства резервного копирования используется параметр DISK или TAPE;
- значение NAME указывает имя резервной копии;
- значение FORMAT указывает на очистку накопителя для резервной копии;

3. Создание отчетов по запросу пользователя.

Реализация генерации отчетов представлена в Листинге 8:

Листинг 8 – Генерация отчетов появления новых клиентов по садам

```
 {
<tr>
   <td>
      @Html.DisplayFor(modelItem => item.N)
   </td>
   <td>
      @Html.DisplayFor(modelItem => item.Customer)
   </td>
   <td>
      @Html.DisplayFor(modelItem => item.SNT)
   </td>
   <td>

</tr>
}
<tfoot>
```

Окончание Листинга 8:

```
            <tr>
                <th></th>
                <th></th>
                <th>
                    @Html.Raw(String.Format("{0:N3} га", group.Records.Sum(r => r.N)))
                </th>
                <th>
                    @Html.Raw(String.Format("{0:P2}", 1))
                </th>
                <th>
                    @Html.Raw(String.Format("{0}", group.Records.Sum(r => r.SNTt)))
                </th>
            </tr>
        </tfoot>
    </table>
    <div class="col-md-4">
        <label class="chart-label">Процент новых клиентов %</label>
        <canvas id="@string.Format("percentChart{0}", group.GetHashCode())" width="300" height="300" class="chart"></canvas>
        <script>
            if (typeof(chartData) == "undefined") {var chartData = {}; };
            chartData["@string.Format("percentChart{0}", group.GetHashCode())"] = {};
            chartData["@string.Format("percentChart{0}", group.GetHashCode())"].data = @Html.Raw(Json.Encode(group.PercentChartData));
            chartData["@string.Format("percentChart{0}", group.GetHashCode())"].options = { tooltipTemplate: "<%= value %>%", segmentStrokeWidth: 0.5 };
        </script>
    </div>
    <div class="col-md-4" id="legend-@string.Format("chart{0}", group.GetHashCode())"></div>
}
```

В результате пользователю отображается страница с отчета (Рисунок 8):

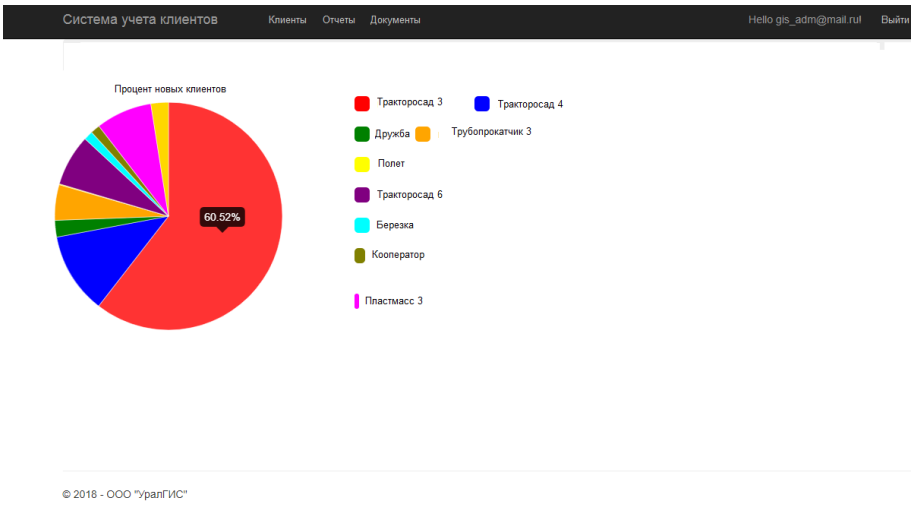

Рисунок 8 – Количество новых клиентов, при распределении по садам

4. Создание шаблонов документа

Фрагмент генерации шаблона документа представлена в листинге 9:

Листинг 9 – Генерация шаблона документа

```
var wordApp = new Word.Application ();
wordApp.Visible = false;

try
{
var wordDocument = wordApp.Documents.Open(TemplateFileName);
ReplicationWordStub (“{ФИО}”, FullName.Customer, WordDocument);
ReplicationWordStub (“{Телефон}”, Phone.Customer WordDocument);
ReplicationWordStub (“{Адрес}”, Adress.View, WordDocument);.
ReplicationWordStub (“{DateTime.Now }”, DateTime.Now(), WordDocument);
wordDocument.SaveAs (@C:\Downloads\zayavlenie.doc”);

wordApp.Visible = true;
}
catch
{
MessageBox.Show (“Произошла ошибка”);
}
}
```

Принцип работы заключается в том, что в системе хранятся шаблоны документов, со встроенными значениями из базы данных. Пример документа представлен на рисунке 9.

Директору ООО «УралГИС»

Максимовой В.Н.

От {ФИО}

заявление

Прошу провести межевание (уточнение) участка, расположенного по адресу: {Адрес} на договорной основе.

Тел. {Телефон} Дата {DateTime.Now}

Подпись_____

Рисунок 9 – Шаблон документа

Принимая клиента, сотрудник для экономии времени может воспользоваться заполненными ранее данными о заказчике и получить заполненный документ. (Рисунок 10).

Директору ООО «УралГИС»

Максимовой В.Н.

От Тышкевич Н.А.

заявление

Прошу провести межевание (уточнение) участка, расположенного по адресу: г.Челябинск, Траткорозаводский район, СНТ «Тракторосад - 3», дорога 13, участок 54 на договорной основе.

Тел. 772-39-52 Дата 18.05.2018

Подпись_____

Рисунок 10 – Заполненный документ

4 Расчет экономической эффективности от оптимизации базы данных.

Как и любому коммерческому предприятию, ООО «УралГИС» важна экономическая выгода от оптимизации системы учета клиентов предприятия. Преимущества, получаемые при оптимизации системы учета клиентов предприятия, можно представить в формализованном виде. Так, нетрудно рассчитать прямой эффект от оптимизации системы, то есть непосредственные выгоды, связанные с экономией рабочего времени сотрудников. Однако сложнее произвести количественное измерение преимуществ стратегического характера, связанных с автоматизацией клиентского учета. Косвенная эффективность - это эффект, получаемый от обработки данных, не поддающийся прямому счёту, а требующий рассмотрения факторов, на которые влияет автоматизация процессов управления.

Для оценки эффекта от оптимизации системы рассмотрим несколько факторов, лежащих на поверхности и легко просчитываемых:

- сокращение сроков поиска информации о клиенте;
- сокращение времени на передачу и согласование документов;

Сокращение сроков поиска информации о клиенте.

Кадастровой деятельностью занимается 3 работника (за исключением вспомогательного персонала). Все они для выполнения работы используют персональный компьютер.

Средняя заработная плата сотрудника ООО «УралГИС» - 15000 руб.

Количество рабочих дней в месяце – 21.

Средняя продолжительность рабочего дня – 8 часов.

Таким образом, средняя стоимость одной минуты рабочего времени составляет – 1,4 руб.

В среднем один сотрудник тратит в день на поиск нужной информации о клиенте от 10 до 60 минут своего времени. Примем 20 минут, как среднее значение. Из этого следует, что каждый сотрудник в год тратит на поиск документов в среднем 3,5 рабочих дня, что для предприятия составляет 2502 руб. расходов на его

заработную плату. Для 3 сотрудников эта величина составит - 5 рабочих дней и соответственно 3571 руб. затрат на их заработную плату.

При оптимизации системы эти затраты сократятся в 5 раз.

Сокращение времени на передачу и согласование документов.

В существующих на предприятии условиях при необходимости согласования или передачи какого-либо документа между отделами либо подразделениями тратится значительное количество времени, поскольку эти процессы осуществляются доставкой документации непосредственно «из рук в руки». Таким образом, один сотрудник тратит в среднем 10-15 минут на доставку документа, а другой сотрудник 10 минут ожидает этот документ. При согласовании одного документа таких звеньев может быть около 5-6. То есть согласование одного проекта, с учетом изучения и внесения изменений, может занять около 8 часов.

При внедрении системы сотрудник автоматически будет получать документ на согласование и сразу же сможет приступить к изучению. Учитывая сокращение времени на передачу документа, согласование может сократиться до 1-2 часов.

При переводе в финансовый показатель, экономия 6 часов одного сотрудника равняется 504 руб. А в расчете на 3 сотрудников показатель равен 1512 руб. Таким образом, предприятие с согласования одного документа получает экономию 1512 руб. Если в среднем за месяц согласование проходят около 100 документов, то экономия составит 151200 руб. А за год – 181440 руб.

Выводы по разделу четыре

В данном разделе была произведена оптимизация системы согласно требованиям задания, и рассчитан экономический эффект от оптимизации системы.

ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе в качестве объекта исследования выступает ООО «УралГИС».

Проводилось исследование и анализ делопроизводственных процессов работы предприятия, в ходе чего была выявлена проблема: неэффективная организация системы ведения учета клиентов, и, как следствие, потеря документов, длительный процесс согласования, отсутствие контроля исполнения, длительный процесс поиска документов, что в свою очередь приводит к трудовым и временным затратам.

В результате была определена основная цель работы – оптимизация системы учета клиентов предпиятия для автоматизаци и процессов делопроизводства.

В первой главе дана характеристика деятельности и организационной структуры предприятия.

В первой главе описаны задачи, выполняемые системами управления взаимоотношениями с клиентами.

Во второй главе дана характеристика деятельности и организационной структуры предприятия.

В третьей главе была проанализирована существующая система ведения учета клиентов, выявлены недостатки и, в соответствии с ними, разработаны функциональные требования к оптимизации системы.

В четвертой главе была предложена оптимизация системы и детально описана ее разработка.

В конце работы был проанализирован показатель экономической эффективности оптимизации системы. В результате анализа было доказано, что оптимизация системы не просто решает поставленные задачи, но еще и экономически целесообразно.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Молино, П. Технологии CRM: экспресс-курс / Молино П. – М.: ФАИР-ПРЕСС, 2011г. – 120 с.
2. О программных продуктах CRM-систем //http://www.crmonline.ru [Электронный ресурс]. Режим доступа: – //http://www.crmonline.ru, свободный.
3. Коровин, А.М. Новые информационные технологии в административном и корпоративном управлении: учебное пособие / А.М. Коровин. – Челябинск: Изд-во ЮУрГУ, 2003. – 113 с.
4. Коровин, А.М. оформление курсовых и дипломных работ по специальности 230102 – «автоматизированные системы обработки информации и управления»: учебное пособие / А.М. Коровин, В.Н. Любицин. – Челябинск: Изд-во ЮУрГУ, 2006. – 39 с.
5. ГОСТ 7.1-84. Система стандартов по информации, библиотечному и издательскому делу, библиографическое описание документа. Общие требования и правила составления.
6. ГОСТ 19.001-77. ЕСПД. Общие положения.
7. ГОСТ 19.101-77. ЕСПД. Виды программ и программных документов.
8. ГОСТ 19 103-77. ЕСПД. Обозначения программ и программных документов.
9. ГОСТ 19.201-78. ЕСПД. Техническое задание. Требования к содержанию и оформлению.
10. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
11. ГОСТ 34.003-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения.

12. ГОСТ 34.602-89. Техническое задание на создание автоматизированной системы.
13. Коренная, К.А. Интегрированные информационные системы промышленных предприятий: моногр. / К.А. Коренная, О.В. Логиновский, А.А. Максимов; под ред. д-ра техн. наук проф. А.Л. Шестакова. – Челябинск: Издат. центр ЮУрГУ, 2012. – 315 с.
14. Логиновский, О.В. Корпоративное управление / О.В. Логиновский, А.А. Максимов. – М.: Машиностроение, 2007. – 624 с.
15. Логиновский, О.В. Управление промышленным предприятием / О.В. Логиновский, А.А. Максимов. – М.: Машиностроение, 2006. – 603 с.

ПРИЛОЖЕНИЕ А

Существующая система учета клиентов ООО «УралГИС» (Рисунок 1А)

Дата принятия документ	ФИО заказчика	Дата окончания действия договор	Наименование СНТ	ФИО приемщика документо	Дата приемки акта согласования	ФИО кадастрового инженера	Дата начала подготовки межевого плана	Примечание	Статус	Результат	Дата подготовки МП	Дата сдачи акта согласовани я в ЗКГ	Ка
01.01.2000	Андрюков		Тракторосад №3			Глазков И.А.	ГОТОВО	Поставлен на учет 28.03.17		НА УЧЕТЕ			
01.01.2000	Ахметова Татьяна Васильевна		Тракторосад №3			Валецкая О.С.		Подписан акт выполненных работ (16.09.2016), отсканирован кадастровый паспорт	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Бахарева Р.П		Оптимист	Нечунаева К.		Аляпсков М.Н.	05.09.2016	заявитель забрал МП	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Вавилова Ирина Александровна		Тракторосад №3			Глазков И.А.	14.09.2016	Подписан акт выполненных работ (10.11.2016 г, отсканирован кадастровый паспорт	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Валиев Марат Ахатович (8-908-821-08-95)		Шахты Центральная			Валецкая О.С.	20.09.2016	Подписан акт выполненных работ (11.11.2016), отсканирован кадастровый паспорт	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Веденяпин Николай Сергеевич (8-912-313-71-20)		Тракторосад №3			Валецкая О.С.	22.02.2017	Акт выполненных работ подписан, кадастровый паспорт отсканирован	ГОТОВО	НА УЧЕТЕ		03.04.2017	
01.01.2000	Верещагина Ксения Игоревна		Тракторосад №3	Нечунаева К.		Нечунаева К.		Акт согласования подписан 27.10.16, кадастровый паспорт отсканирован		НА УЧЕТЕ			
01.01.2000	Власова Наталья Витальевна		Тракторосад №3	Глазков И.А.		Глазков И.А.	21.09.2016	Поставлен на учет в ГКН 23.11.2016		НА УЧЕТЕ			
01.01.2000	Давыдова Светлана Викторовна (8-952-504-45-47, раб.775-21-26		Тракторосад №4			Валецкая О.С.	31.08.2016	Подписан акт выполненных работ (17.11.2016), отсканирован кадастровый паспорт	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Дмитриева Татьяна Ивановна		Тракторосад №3			Валецкая О.С.	23.09.2016	Подписан акт выполненых работ 16.11.201, отсканирован кадастровый паспорт	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Ершов Александр Николаевич		Тракторосад №6			Валецкая О.С.	30.06.2016	Подписан акт выполненных работ 1.08.2016	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Желиба Г.Г.		Тракторосад №4			Аляпсков М.Н.	06.09.2016	заявитель забрал МП	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Зябкина Ольга Александровна 8-951-12-89-394		Тракторосад №3			Валецкая О.С.	21.09.2016	Акт выполненных работ подписан, кадастровый паспорт отсканирован	ГОТОВО	НА УЧЕТЕ		19.05.2017	
01.01.2000	Калачёва Алевтина Борисовна		Тракторосад №4			Валецкая О.С.	02.09.2016	Подписан акт выполненных работ (29.09.2016), отсканирован кадастровый паспорт	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Каткова О.В		Дружба			Аляпсков М.Н.	15.10.2016	заявитель забрал МП	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Кочегаров А.В.		Тракторосад №4			Карпкин М.		Кадастровый паспорт отсканирован, акт выполненных работ 11.01.17		НА УЧЕТЕ			
01.01.2000	Кравченко Ирина Казимировна		Оптимист			Валецкая О.С.		Подписан акт выполненных работ (15.09.2016), отсканирован кадастровый паспорт	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Кузьменко Анатолий Семенович		Оптимист			Валецкая О.С.		Подписан акт выполненных работ (15.09.2016), отсканирован кадастровый паспорт	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Кулькова В.И		Оптимист	Нечунаева К.		Аляпсков М.Н.	08.09.2016	заявитель забрал МП	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Куминова Татьяна Ивановна (8-951-772-19-60, 790-10-11)		Любитель-2			Валецкая О.С.	06.09.2016, 13.02.2017	Заказчик забрал МП 22.02.2017, Приостановка: указать зону территориальной проекции	ГОТОВО	НА УЧЕТЕ		21.04.2017	
01.01.2000	Курочкина Е.А		Тракторосад №6			Аляпсков М.Н.	09.02.2017	исправлена кад ошибка	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Мелашич Иван Николаевич		Оптимист			Валецкая О.С.	-	Подписан акт выполненных работ (27.09.2016), отсканирован кадастровый паспорт	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Новгородская Светлана Юрьевна (8-912-793-08-15)		Тракторосад №3			Валецкая О.С.	16.09.2016	Подписан акт выполненных работ (07.11.2016), отсканирован кадастровый паспорт	ГОТОВО	НА УЧЕТЕ			
01.01.2000	Осипова Наталья Валентиновна		Тракторосад №3			Валецкая О.С.	08.04.2017	Подписан акт выполненных работ, отсканирована выписка из ЕГРН	ГОТОВО	НА УЧЕТЕ			

Рисунок 1А - Существующая система учета клиентов ООО «УралГИС»

ПРИЛОЖЕНИЕ Б

Структурная схема базы данных (Рисунок 1Б)

Рисунок 1Б - Структурная схема базы данных