

РЕФЕРАТ

Выпускная квалификационная работа состоит из 125 страниц, 16 рисунков, 39 таблиц, 23 источников литературы.

Объектом исследования является централизованная генерация сложных отчётов.

Цель работы – разработка унифицированного сервера отчётов для корпоративных информационных систем в рамках создания ЕАМ-системы.

В процессе работы был проведён обзор существующих на рынке информационных систем серверов отчётов и разработан унифицированный сервер отчётов для корпоративных информационных систем. Приведены результаты его использования.

Разработанный унифицированный сервер отчётов может быть применён в корпоративных информационных системах для решения задач централизованной генерации сложных отчётов

Экономическая значимость работы: предоставление широкому кругу заинтересованных лиц инструментального средства, который позволяет генерировать, кэшировать и строить по расписанию сложные отчёты.

Выпускная квалификационная работа выполнена в текстовом редакторе Microsoft Word 2007 и представлена на компакт-диске (в конверте на обороте обложки).

Для разработки программного обеспечения использованы: службы Microsoft SQL Reporting Services, язык программирования SQL, среда для доступа и управления базами данных Management Studio, интегрированная среда разработки для создания приложений для Windows: Microsoft Visual Studio.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	7
1.1 Oracle Reports Developer.....	7
1.2 Crystal Reports.....	9
1.3 Stimulsoft Reports.....	10
1.4 Fast Reports.....	11
1.5 Microsoft SQL Reporting Services.....	12
1.6 Недостатки рассмотренных серверов отчётов	13
1.7 Анализ функциональных требований серверов отчётов	15
2 ПРОЕКТИРОВАНИЕ СЕРВЕРА ОТЧЁТОВ	18
2.1 Цель и функции сервера отчётов.....	18
2.2 Разработка структуры универсального сервера отчётов	18
2.2.1 Подсистема администрирования сервера отчётов	19
2.2.2 Обработчик запросов сервера отчётов	20
2.2.3 Кэш сервера отчётов.....	21
2.2.4 Система расписаний сервера отчётов	22
2.2.5 Драйверы генераторов отчётов	22
2.3 Разработка алгоритма построения отчётов для клиентского приложения	23
2.4 Разработка алгоритма построения отчётов по расписанию	25
3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ УНИФИЦИРОВАННОГО СЕРВЕРА ОТЧЁТОВ.....	27
3.1 Требования к аппаратным и программным средствам	27
3.2 Подсистема администрирования сервера отчётов.....	27
3.2.1 Настройка параметров кэширования	28
3.2.2 Настройка планирования по расписанию	30
3.3 API унифицированного сервера отчётов	31
3.4 Генератор отчётов	35

4 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОСБЕРЕЖЕНИЕ И РЕСУРСОЭФФЕКТИВНОСТЬ.....	Ошибка! Закладка не определена.
5 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ....	Ошибка! Закладка не определена.
ЗАКЛЮЧЕНИЕ	40
СПИСОК ПУБЛИКАЦИЙ	41
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	Ошибка! Закладка не определена.
ПРИЛОЖЕНИЕ А	42
ПРИЛОЖЕНИЕ Б.....	53
ПРИЛОЖЕНИЕ В	54
ПРИЛОЖЕНИЕ Г.....	55
ПРИЛОЖЕНИЕ Д	57
ПРИЛОЖЕНИЕ Е.....	58
ПРИЛОЖЕНИЕ Ж	59
ПРИЛОЖЕНИЕ И	60
ПРИЛОЖЕНИЕ К	61
ПРИЛОЖЕНИЕ Л	62
ПРИЛОЖЕНИЕ М	64

ВВЕДЕНИЕ

Актуальность работы. Во всех корпоративных информационных системах актуальна задача построения отчётности. Одна из серьёзных проблем подсистем отчётности – это проблема ресурсоёмких отчётов. Отчёты являются ресурсоёмкими, если они обрабатывают большие объёмы данных, потребляя много вычислительных ресурсов, и/или используют сложные расчётные алгоритмы.

Длительное построение ресурсоёмких отчётов для конечных пользователей выглядит как недостаток самой информационной системы. В случае, когда нужно построить не один, а несколько ресурсоёмких отчётов за определённый промежуток времени, причём одновременно некоторому количеству конечных пользователей, создаётся недопустимо большая нагрузка на систему.

Основными способами ускорить получение отчётов является использование в системе отчётности кэширования, построение отчётов по расписанию и оптимизация запросов. Все они могут использоваться как по отдельности, так и совместно.

Таким образом, наилучшим решением построения ресурсоёмких отчётов является использование централизованных серверов приложений для построения отчётов, называемых серверами отчётов.

Главной **целью** магистерской диссертации является разработка унифицированного сервера отчётов (далее УСО) для корпоративных информационных систем в рамках создания ЕАМ-системы.

Цель работы включает в себя решение таких задач, как:

- изучение предметной области;
- поиск и анализ существующих систем отчётности;
- исследование требований основных функциональных возможностей сервера отчётов;

- анализ и выбор программных средств для реализации унифицированного сервера отчётов;

- проектирование API и унифицированного сервера отчётов;

- разработка УСО и API для взаимодействия с ним.

Научная новизна разработки состоит в концепции универсальности сервера отчётов, так как подобных идей для решения проблем ресурсоёмких отчётов, как и готовых существующих генераторов отчётов (далее ГО) с возможностью кэширования отчётов и построения их по расписанию, с последующим помещением в долговременный кэш, не существует.

Практическая значимость результатов научно-исследовательской работы заключается в том, что они могут использоваться организациями и предприятиями, работающими в сфере корпоративных информационных технологий.

Апробация работы. Результаты работы обсуждались и были одобрены на конференциях:

- III Международная научная конференция «Информационные технологии в науке, управлении, социальной сфере и медицине», 23 – 26 мая 2016 года, г. Томск;

- VII Фестиваль Microsoft в ТПУ, 22 мая – 23 апреля 2016 года, г. Томск;

- XIII Международная конференция студентов, аспирантов и молодых учёных «Перспективы развития фундаментальных наук», 26 – 29 апреля 2016 года.

Также результаты работы использовались при разработке корпоративной информационной системы ОАО «Сибирский центр высоких технологий».

Содержание диссертации.

- **Первая глава** посвящена анализу предметной области: обзору существующих на рынке информационных систем программных средств, предназначенных для генерации отчётов, методов их кэширования и планирования по расписанию.

- **Во второй главе** приведено описание программных средств, необходимых для разработки унифицированного сервера отчётов, и составлена архитектура УСО.
- **Третья глава** содержит этапы разработки API для взаимодействия с унифицированным сервером отчётов.
- **В четвёртой главе** приведены результаты анализа и расчётов финансового менеджмента, ресурсоэффективности и ресурсосбережения научно-технического проекта.
- **Пятая глава** освещает вопросы социальной безопасности и защиты в чрезвычайных ситуациях и содержит результаты анализа вредных и опасных факторов в вычислительных центрах.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Среди имеющихся решений на рынке информационных систем по построению ресурсоёмких отчётов, заключающихся в использовании централизованных серверов приложений для построения отчётов, существует множество генераторов отчётов, предоставляющих для этого незначительные возможности, как малоизвестных, которые обычно используются в рамках конкретной системы, так и достаточно мощных и популярных, которые используются для различных систем, например: Oracle Reports, Crystal Reports, Stimulsoft Reports, Fast Reports, Microsoft SQL Reporting Services.

Рассмотрим каждую из них с точки зрения наличия средств для решения задачи ресурсоёмких отчётов.

1.1 Oracle Reports Developer

Программный продукт Oracle Reports Developer, разработанный корпорацией Oracle имеет серверное решение построения отчётов – Oracle Reports Server. Oracle Reports Server используют в случае, когда генерация отчётов требует длительного времени и немалых ресурсов и поэтому проводится в нерабочее время и/или автоматически по расписанию с возможностью доставки отчётов пользователям по электронной почте, и/или в случае необходимого предоставления одних и тех же данных в виде отчётов разным пользователям. Это в свою очередь требует кэширования результатов выполнения запросов.

Идея кэширования в Oracle Reports Server заключается в том, что часто запрашиваемые пользователями данные сохраняются в кэше, а не извлекаются заново при каждом запросе [1]. Он позволяет кэшировать как статические, так и динамические данные и содержит процедуры для задания условий, при которых данные в кэше следует обновлять. В Oracle Reports Server при известных зависимостях между данными и отдельными пользователями или приложениями возвращается та или иная информация с учётом конкретной ситуации. Oracle

Reports Server позволяет сжимать HTML-страницы отчётов, что ускоряет их доставку клиентам.

Данные, обрабатываемые Oracle Reports Server, для повторного использования помещаются в разделяемый пул, который кэширует различные конструкции, использующиеся несколькими пользователями повсеместно, в том числе предъявляемые пользователями SQL-запросы и их фрагменты, а также результаты их выполнения. Объём разделяемого пула может автоматически расширяться. Чем меньше объём разделяемого пула, тем сильнее это отражается на его производительности, ощущаемой всеми пользователями. Это приводит к тому, что:

- В кэш помещается мало информации о создаваемых отчётах, что влечёт за собой частое обращение к диску для запроса и обновления информации в словаре.
- Не удаётся кэшировать достаточное количество SQL команд, что приводит к «перетряхиванию» памяти, когда одни отчёты вытесняются из кэша для освобождения места другим отчётам. Если памяти недостаточно для кэширования, то одни и те же отчёты приходится разбирать, кэшировать и выталкивать снова и снова, бессмысленно растрачивая ресурсы процессора и увеличивая накладные расходы на выполнение каждой транзакции.
- Не удаётся кэшировать достаточное количество запросов, что также приводит к «перетряхиванию» памяти и снижению производительности при выполнении кода, хранящегося в базе данных.

Кэшировать отчёты в Oracle Reports Server можно в течение заданного времени, когда в ответ на очередной запрос будет возвращён уже готовый отчёт вместо повторного выполнения необходимых для его генерации обращений к базе данных. Кроме того, Oracle Reports Server позволяет не только создавать отчёты, но и запускать их по расписанию в периоды низкой загрузки, после чего конечные пользователи в любой момент могут просмотреть полученные результаты [2].

1.2 Crystal Reports

Входящий в Crystal Reports механизм планировщика заданий позволяет пользователям автоматически генерировать отчёты, в соответствии с заданным расписанием, и предназначен в первую очередь[3]:

- для выполнения сложных отчётов, требующих значительных серверных ресурсов, в том числе отчётов с раскрытыми подотчётами как версии для печати.
- для выполнения регулярных (циклических) отчётов, которые необходимо формировать в определённое время и/или день (недели, месяца).

Также механизм планировщика заданий позволяет выполнять отчёты практически в любом формате и для любого места назначения (файл, принтер, электронная почта, ftp) и события, которые могут быть трёх типов:

- файловые события, возникающие при появлении заданных файлов в заданном каталоге;
- события планировщика, позволяющие организовать взаимозависимое выполнение отчётов;
- события по команде, которые передаются из консоли управления системой (Crystal Management Console), возникающие при запуске на выполнение всех отчётов, ожидающих это событие, лицом, имеющим административные привилегии.

Благодаря входящему в состав Crystal Reports серверному решению, Crystal Reports Server, пользователь имеет возможность кэшировать отчёты. Кэширование отчётов происходит следующим образом: Crystal Reports Server поддерживает эффективную обработку отчётов с использованием сервера страниц, тот в свою очередь несёт ответственность за отклики на запросы страниц, выполняя по мере необходимости обработку отчётов и их генерацию. При этом пользователю выдаётся только требуемая страница, а не весь отчёт целиком. Это значительно увеличивает производительность и уменьшает сетевой трафик для больших отчётов, однако позволяет пользователю только

просмотреть отчёт, а не распечатать его. Crystal Reports Server также поддерживает активное совместное использование данных, которое означает, что отчёты могут использовать совместные данные при обращении к одним и тем же данным. При этом такие отчёты могут генерироваться без дополнительных запросов к базе данных (далее БД), даже при отличии одного отчёта от другого. Результатом является значительный рост производительности в рамках всей системы, включая базу данных. Основным преимуществом активного совместного использования данных является уменьшение нагрузки и интенсивности использования, более полное кэширование данных, с обеспечением более эффективной работы системы.

Сервер кэширования, Crystal Reports Server, несёт ответственность за обработку всех запросов на просмотр отчётов. Он проверяет, можно ли выполнить запрос с использованием кэшированной страницы отчёта. Если это невозможно, он запрашивает у сервера страниц генерацию данной страницы.

1.3 Stimulsoft Reports

Stimulsoft Reports также имеет своё серверное решение для задач построения ресурсоёмких отчётов: Stimulsoft Report Server. При кэшировании отчётов Stimulsoft Report Server использует временные файлы для хранения промежуточных результатов, благодаря чему отчёты строятся без излишнего потребления оперативной памяти. Благодаря применению многопоточности построение отчётов с кэшированием происходит значительно быстрее, так как ядро генератора отчётов запускает несколько потоков: основных и вспомогательных [4]. В основном потоке выполняется построение отчёта, параллельно во вспомогательном осуществляется работа с временными файлами кэша. В результате время построения отчёта, с кэшированием или без него, практически совпадает.

Запланированная обработка данных по расписанию в Stimulsoft Report Server происходит при помощи встроенного планировщика, который позволяет запрограммировать любое событие по времени – от простого однократного

запуска по времени до национальных праздников, со смещением по времени. При этом с его помощью можно настроить циклические повторения отчётов и их исключения. Объект планировщика «календарь» позволяет собрать список не связанных логически друг с другом дат. При этом он учитывает часовые пояса и имеет крайний срок исполнения действий, в качестве которых можно запускать отчёты, копировать объекты, отсылать почтовые сообщения для информирования пользователей о событиях, и запускать вложенные планировщики. Все эти действия могут создаваться в рамках одного планировщика и последовательно выполняться без малейшего вмешательства конечного пользователя или администратора сервера отчётов.

Кроме того, важной особенностью Stimulsoft Report Server является возможность работать с многостраничными документами. Такие документы Stimulsoft Report Server приводит к компактному виду.

1.4 Fast Reports

Кэширование отчётов и их данных в Fast Reports происходит как в памяти, что позволяет увеличить быстродействие, так и в файлах на диске, что необходимо для экономии ресурсов ОЗУ[5].

В Fast Reports существуют следующие виды кэширования:

1. Весь текст и объекты построенного отчёта сохраняются во временный файл на диске, при этом указывается, сколько мегабайт отводится под шаблон в ОЗУ.

2. В памяти может храниться как целый отчёт, так и некоторые его страницы, что существенно ускоряет работу предварительного просмотра, но расходует много памяти (особенно если в шаблоне есть рисунки).

3. Все картинки построенного отчёта сохраняются во временный файл на диске, что приводит к уменьшению потребления памяти в отчётах с большим количеством рисунков, но снижает быстродействие.

Кроме функции кэширования Fast Report предоставляет пользователям возможность автоматизированной работы с отчётами и их планового

составления. К основным его возможностям относятся: регулярность формирования отчётов, их сохранение и рассылка по e-mail. В конфигураторе планировщика задач пользователь может задать свойства и действия, связанные с запуском нужного отчёта, а также дату и время его выполнения и отправить отчёт по e-mail в одном из удобных форматов (PDF, HTML, TXT, RTF, GIF, BMP). Кроме того, в планировщике присутствует возможность настройки периодического запуска задания.

1.5 Microsoft SQL Reporting Services

Службы Reporting Services позволяют пользователям переносить часть нагрузки на сервер БД на другой сервер для улучшения производительности всей системы [6]. Кроме того, эти службы позволяют отделять базовую обработку отчёта, включая получение и размещение данных, от преобразования отчёта в требуемый формат[7]. Этот процесс является кэшированием.

Кэширование в службах Reporting Services позволяет сократить время, необходимое для поиска нужных отчётов, объём которых велик либо к которым часто обращаются[7]. Содержимое кэша энергонезависимо, это означает, что оно может измениться при замене, добавлении или удалении отчётов. Службы Reporting Services хранят временные файлы в БД, благодаря чему обеспечивается поддержка пользовательских сеансов и обработка отчётов. Эти файлы кэшируются для внутреннего использования.

Кэшируемый экземпляр отчёта основан на промежуточном формате отчёта. Сервер отчётов в целом кэширует только один экземпляр отчёта на основе его имени, но если он будет содержать данные, которые зависят от параметров запроса, то в этом случае кэшироваться будут уже несколько версий отчёта в нужное время.

Кэшированный отчёт заменяется более новой версией, когда пользователь выбирает отчёт, у которого срок действия кэшированной копии истёк [8]. Отчёты, которые настроены для выполнения в виде кэшируемых экземпляров, удаляются из кэша через определённые промежутки времени,

заданные в параметрах настройки истечения срока действия. При этом, если пользователь не использует APISOAP, то напрямую удалить отчёт из кэша он не сможет.

Службы Microsoft SQL Reporting Services позволяют создавать отчёты по расписанию, что особенно важно при обработке больших объёмов данных. Множество отчётов могут распределять общий планировщик обработки, например, выполнения генерации месячной или квартальной отчётности. Пользователь может задать расписание только для определённого отчета или же использовать общее расписание для нескольких подписок [9]. При этом для отчётов с параметрами он должен определить значения, присваиваемые параметрам при запуске подписки.

Срок действия расписания истекает без уведомления[10]. После даты его окончания отчёты больше не выполняются. При этом расписания с истёкшим сроком действия сами не удаляются, пользователь может их удалить только вручную. Таким образом, если необходимо продлить расписание, то нужно увеличить дату его окончания.

1.6 Недостатки рассмотренных серверов отчётов

Возможности рассмотренных серверов отчётов недостаточны для решения всех проблем, которые связаны с построением отчётов. Перечислим эти проблемы.

1. Наличие проблемы валидности кэша. Поскольку данные, извлекаемые из базы данных и используемые в отчётах, изменяются вне систем отчётности сторонними приложениями. Сервер отчётов отследить этот процесс изменения не может и не может установить изменённые данные, которые в нём затем будут использоваться.

2. Отсутствие возможности прямого управления кэшированием отчётов. Ресурсоёмкие отчёты, сохранённые в кэше, могут быть актуальны только в течение определённого времени, по истечению которого они вытесняются из кэша другими, более новыми отчётами. Поэтому отчёты

требуется строить регулярно, а построение в свою очередь может продолжаться от нескольких часов до нескольких дней, что тормозит работу пользователя. Такая проблема периодически возникает в Oracle Reports Server, где пользователи ещё некоторое время могут работать с неактуальной версией отчёта, а затем должны либо сохранить новую версию отчёта, либо удалить отчёт из кэша, и в Microsoft SQL Reporting Services, где для доставки в кэш новых копий отчётов необходимо истечение срока действия старых копий отчётов.

3. Отсутствие возможности построения отчёта по расписанию с последующим его сохранением в кэш. Отчёты, построенные по расписанию, в этих системах отчётности, минуя сохранения в кэш, рассылаются всем заявленным пользователям, причём как пользователям, которым эти отчёты нужны в данный момент, так и пользователям, которым они не нужны. При этом во всех системах отчётности заранее неизвестно, когда, кем и каким образом были изменены данные, используемые в отчётах, отчего сложно определить, актуальна построенная по расписанию версия отчёта или нет. В службах Microsoft SQL Reporting Services построенные по расписанию отчёты сохраняются в кэш в промежуточном формате, что позволяет пользователям, которым они необходимы, самостоятельно извлекать отчёты из кэша.

4. Поддержка не всех комбинаций частоты событий. К примеру, если пользователю нужно запускать отчёт каждый четверг в 14:00 и в 18:00, то необходимо создать два ежедневных расписания, запускаемых в четверг, и для одного указать время запуска 14:00, а для другого – 18:00. В Crystal Reports построенные по расписанию отчёты хоть и сохраняются в кэше, однако интервал их хранения недолговечен и варьируется от 20 до 60 минут.

5. Наличие возможности сохранения отчётов в форматах офисных приложений Microsoft (Word, Excel и др.) не во всех системах отчётности, тогда как она является важным условием для построения ресурсоёмких отчётов, позволяющим предоставлять для них большие возможности в области печати и отправки по почте. Это особенно актуально для продуктов Oracle Reports Developer и Fast Reports, в котором присутствует только собственный формат

сохранения отчётов. В программных продуктах Stimulsoft Reports, Crystal Reports и службах Microsoft SQL Reporting Services отчёты хоть и не сохраняются в форматах офисных приложений, но могут конвертироваться в них. Конвертация форматов отчётов также не обходится без недостатков: как в Crystal Reports, так и в Microsoft SQL Reporting Services при конвертации в отчётах теряются некоторые данные об оформлении, как например, информация, используемая при построении графиков и диаграмм, и некоторые внутренние данные, такие как, формулы в вычисляемых полях.

6. Невозможность рассматриваемых систем отчётности быть какими-либо унифицированными решениями, которые можно интегрировать с желаемым генератором отчётов. У разработчика может возникнуть необходимость самостоятельного выбора для себя генератора отчётов, подходящего ему по каким-либо критериям.

Таким образом, поскольку в существующих системах отчётности имеются недостатки, связанные с отсутствием управления кэшем и невозможностью построить отчёт по расписанию с последующим его сохранением в кэш, нет никакой гарантии, что один построенный ресурсоёмкий отчёт не будет вытеснен из кэша другим построенным отчётом. Поэтому эти технологии в полной мере не позволяют решить проблему ресурсоёмких отчётов.

1.7 Анализ функциональных требований серверов отчётов

После анализа возможностей существующих систем отчётностей, был составлен список функциональных требований к создаваемому серверу отчётов, который объединит в себе не только достоинства этих систем, но и исправит недостатки. Опишем эти требования.

1. Отчёты в кэше должны храниться долговременно, при этом объём и расписанием кэша должен управлять администратор. Построение ресурсоёмких отчётов может варьироваться в интервалах от нескольких часов до нескольких дней, а у пользователя может не быть такой возможности строить отчёты заново

и регулярно. При этом невозможно определить, когда, как и кем были изменены данные в отчёте, поэтому контроль администратора за построением отчётов и сохранением его в кэше на длительный срок просто необходим.

2. Пользователь должен иметь возможность гибкой настройки планирования построения отчётов по расписанию, в том числе в часы минимальной загрузки (например, в ночное время суток или в праздничные дни). При этом учитывается не расписание, при котором сервер построил отчёт и отослал его пользователю в течение определённого времени, а то, что отчёт, построенный по расписанию, сохраняется в кэше и затем извлекается из него пользователем при необходимости. Поскольку заранее неизвестно, кому отчёт может понадобиться в течение заданного времени, а кому нет. Более того, отчёт может отправиться к человеку, которому этот отчёт не нужен, а человеку, которому, наоборот, этот отчёт необходим, так и не прийти. Поэтому контроль администратора за построением отчётов здесь тоже необходим, чтобы построенный по расписанию отчёт сохранялся в кэше, а затем выгружался только тем пользователям, которым он нужен.

3. Пользователь должен иметь возможность самостоятельного выбора между тем, хочет ли он построить новый отчёт и тем, хочет ли он взять готовый отчёт из системы, то есть при построении отчёта сервер отчётов должен проверить его наличие в кэше: если отчёта там нет, то сервер должен построить его и выдать пользователю, если же отчёт в кэше уже есть, то сервер должен уведомить пользователя о действиях, которые необходимо будет произвести с отчётом (перезаписать старый, добавить новый, отменить добавление). При этом построение отчёта в кэше может существовать, а может не существовать, поэтому пользователь должен быть заранее оповещён о наличии кэша. Хотя проблему валидности кэша нельзя решить полностью, мы пытаемся тем самым переложить часть решения на пользователя.

4. Пользователь должен иметь возможность сохранять отчёты в форматах офисных приложений Microsoft (Excel, Word и др.), максимально удобных для печати и отправки отчётов по почте. Ведь возможностей для печати отчётов,

предоставляемых офисными приложениями, больше, нежели возможностей для печати отчётов, предоставляемых системами отчётностей. Например, Microsoft Excel может предложить пользователю печать как всего документа, так и отдельных его страниц, диапазона значений, отфильтрованных данных, примечаний в ячейках и т.д. Также Excel предоставляет возможность печати широкоформатных документов, таких, как А3 или А1, целиком или по частям.

5. Сервер отчётов должен иметь возможность интеграции с каким-либо существующим генератором отчётов, который будет строить отчёты, потому что сам УСО должен заниматься решением вопросов взаимодействия между клиентским приложением и генератором отчётов, в частности осуществляя подключение к генератору отчётов и предоставляя решение задач, связанных с кэшированием отчётов и построением их по расписанию.

Таким образом, в рамках проектной работы необходимо разработать унифицированный сервер отчётов, который будет решать задачи прямого управления кэшированием и построения отчётов по расписанию с помещением их в кэш, решая при этом проблему ресурсоёмких отчётов, и иметь возможность интеграции с каким-либо существующим генератором отчётов, необходимым для построения отчётов.

В качестве генератора отчётов будет использоваться XLFormBuilder, подключение к которому будет осуществляться за счёт разрабатываемого драйвера (промежуточного приложения), который будет перенаправлять к нему запросы от УСО с некоторым набором параметров для построения отчёта. Сгенерированный отчёт XLFormBuilder будет отправлять обратно УСО в виде некоторого файла, а тот, в свою очередь, будет решать: поместить отчёт в кэш или отправить пользователю.

2 ПРОЕКТИРОВАНИЕ СЕРВЕРА ОТЧЁТОВ

2.1 Цель и функции сервера отчётов

Целью работы является разработка унифицированного сервера приложений для построения отчётов, который должен стать промежуточным звеном между клиентским приложением и генератором отчётов.

Создаваемый сервер отчётов должен обладать следующими функциональными характеристиками:

- на сервере должна быть реализована технология подключения различных генераторов отчётов;
- сервер должен предоставлять настраиваемые возможности кэширования и построения отчётов по расписанию;
- сервер должен предоставлять возможность сохранять отчёты в форматах офисных приложений;
- при построении отчёта сервер должен проверить его наличие в кэше: если отчёта там нет, то сервер должен построить его и выдать пользователю, если же отчёт в кэше уже есть, то сервер должен уведомить пользователя о действиях, которые необходимо будет произвести с отчётом (перезаписать старый, добавить новый, отменить добавление).

Кроме того, сервер отчётов должен отправлять с помощью сервиса предупреждений/оповещений по почте заданным пользователям письма с сообщением о заданном событии при выполнении того или иного отчёта.

2.2 Разработка структуры универсального сервера отчётов

Структура создаваемого универсального сервера отчётов представлена на рисунке 2.2.1.

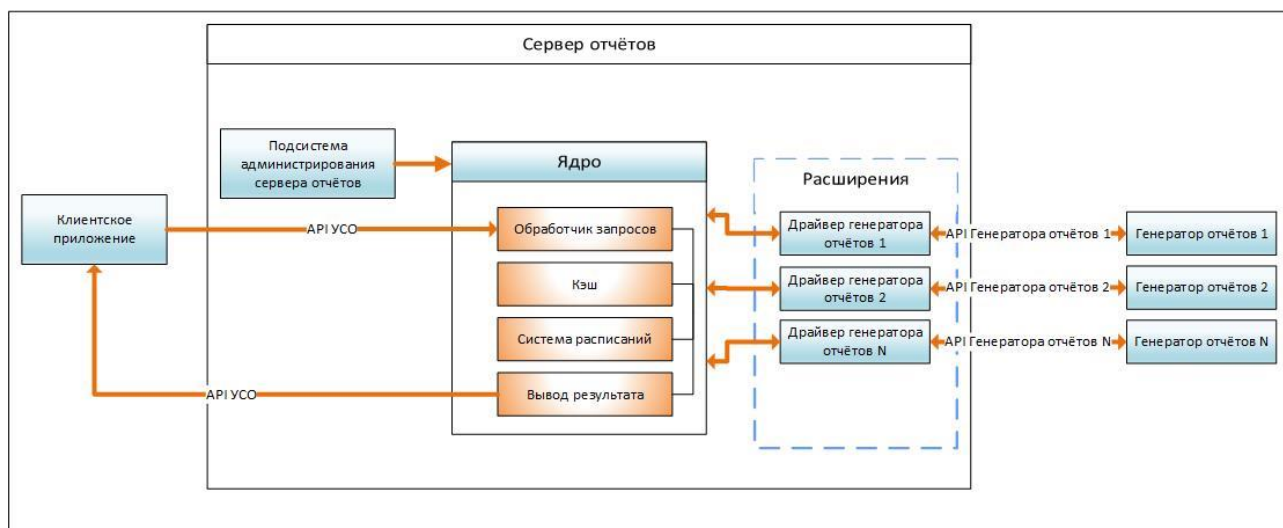


Рисунок 2.2.1 – Структура универсального сервера отчётов

Унифицированный сервер отчётов включает в себя следующие компоненты:

1. Подсистему администрирования сервера отчётов;

2. Ядро, которое также содержит:

- обработчик запросов;
- кэш;
- систему расписаний;
- компонент, отвечающий за выдачу результата.

3. Расширения, включающие различные драйверы генераторов отчётов.

2.2.1 Подсистема администрирования сервера отчётов

Подсистема администрирования сервера отчётов, представляет собой приложение, которое предназначено для внесения изменений в сервер отчётов.

Подсистема администрирования позволяет решать следующие задачи:

- настройка параметров кэша:
 - объём кэша;
 - длительность хранения отчётов в кэше;
 - период очистки кэша.

- настройка параметров системы расписаний:
 - время и дата планируемого создания отчёта;
 - тип повторения: ежемесячно, еженедельно или ежедневно. В пределах одного типа повторений можно указать интервалы и диапазон, определяющий частоту повторения события.
 - информирование пользователей об истечении времени построения отчётов;
 - сохранение запланированного отчёта в кэш;
 - отправка отчётов, построенных по расписанию, пользователям.

2.2.2 Обработчик запросов сервера отчётов

Обработчик запросов позволяет обрабатывать следующие запросы:

- Запрос для построения отчёта по требованию. Клиентское приложение посылает запросу серверу отчётов для создания нового отчёта. На сервере отчётов инициируется обработка запроса. Обработчик запросов получает определение отчёта, проверяет, имеется ли отчёт в кэше, соединяет определение отчёта с данными кэша, и в случае, если данные не совпадают, формирует файл с параметрами этих данных и передаёт генератору отчётов для создания нового отчёта, в необходимом формате по API. После того, как отчёт будет готов, генератор отчётов посылает его обратно серверу отчётов, где обработчик запросов обращается к компоненту, отвечающему за выдачу результата, для передачи готового для просмотра отчёта пользователю. После этого он посылает пользователю запрос для выбора действия «Сохранить отчёт».
- Запрос для получения отчёта из кэша. Обработка отчётов из кэша является разновидностью обработки отчёта по требованию, выполняемой в сервере отчётов. Клиентское приложение посылает серверу запрос на получение отчёта из кэша. На сервере отчётов инициируется обработка отчёта. В случае, если срок действия кэшированной копии истёк, отчёт повторно обрабатывается прежде, чем он будет вновь добавлен в кэш, а затем направляется по API в клиентское приложение. Иначе обработчик запросов обращается к компоненту,

отвечающему за выдачу результата, для передачи готового для просмотра отчёта пользователю. После этого он посылает пользователю запрос для выбора одного из двух действий «Создать новый отчёт» или «Изменить отчёт, полученный им при запросе из кэша».

- Запрос из системы расписаний. Запланированная обработка создания отчётов по расписанию и доставки их пользователям инициируется сервером отчётов и выполняется в фоновом режиме. Весь принудительный доступ к обработке и доставке отчётов производится через расписания и события, обрабатываемые компонентом «Система расписаний» сервера отчётов. При этом запланированные отчёты обязательно сохраняются в кэш. При получении запроса на запланированное построение отчётов, обработчик получает их определение, отправляет кэшу запрос на данные, соединяет определение отчёта с данными и сохраняет скомпилированный отчёт в базе данных сервера отчётов.

2.2.3 Кэш сервера отчётов

Кэш представляет собой временное хранилище отчётов, получаемых сервером отчётов от клиентского приложения. Его основная задача заключается в создании кэшируемых копий отчётов, которое позволит:

- защитить отчёт от случайного его удаления пользователями;
- обеспечить доступ к отчёту одновременно несколькими пользователями;
- сократить время, необходимое для поиска отчётов, что в особенности касается ресурсоёмких отчётов и отчётов, к которым часто обращаются;
- восстановить отчёты при перезагрузке сервера.

Динамический объём кэша позволяет также решать задачу ресурсоёмких отчётов, обеспечивая долговременное хранение их копий и планирование построения по расписанию.

2.2.4 Система расписаний сервера отчётов

Система расписаний позволяет планировать построение отчёта в определённый период времени и информировать об этом событии пользователей, которым он необходим. Кроме того, с помощью системы расписаний можно:

- запланировать доставку отчётов в стандартной или управляемой данными подписке;
- запланировать отчёты, создаваемые по расписанию, так, что изменения, вносимые в них пользователями во время запланированного построения, будут добавляться к этим отчётам через регулярные промежутки времени;
- запланировать время обновления данных о кэшированной копии отчётов;
- запланировать истечение срока действия кэшированного отчёта в стандартное время, чтобы впоследствии его можно было обновить.

2.2.5 Драйверы генераторов отчётов

Драйверы генераторов отчётов, входящие в состав сервера отчётов, позволяют преобразовывать унифицированный интерфейс, который требуется ядру сервера отчётов в определённые вызовы для конкретного генератора отчётов с использованием его собственного API, предоставляемый генератором отчётов. Это означает, что при необходимости построения отчёта сервер отчётов будет посылать генератору отчётов по его API файл с определёнными параметрами данных, посредством использования драйвера отчётов. Сами драйверы генераторов отчётов будут преобразовывать файл с параметрами данных в определённый формат соответствующих им генераторов отчётов.

При этом, поскольку отчёты строятся генератором отчётов, интегрированным с сервером отчётов посредством драйвера, сервер отчётов не может предоставлять возможность сохранять отчёты в формате офисных

приложений. Эту функцию должен осуществлять генератор отчётов, подключённый к серверу, на который накладывается следующее ограничение: генератор отчётов должен уметь выдавать готовый отчёт в виде файла формата Office. Если он не умеет сохранять отчёт в офисном формате, то он должен, хотя бы выдавать его на экран, чтобы пользователь мог просмотреть, сохранить или распечатать отчёт прямо с экрана. Поэтому уникальность унифицированного сервера отчётов базируется на предположении, что генератор отчётов будет отдавать серверу отчётов файл формата офисных приложений.

API генератора отчётов должен обладать двумя характеристиками:

- умение передавать параметры данных для построения отчётов;
- умение получать файл в формате офисных приложений.

API сервера отчётов также должен обладать двумя характеристиками:

- возможность передавать параметры данных для построения отчётов;
- возможность передавать идентификатор отчёта;
- возможность передавать готовый отчёт в виде файла формата офисных приложений.

2.3 Разработка алгоритма построения отчётов для клиентского приложения

На рисунке 2.3.1 представлена схема построения отчётов для клиентского приложения.

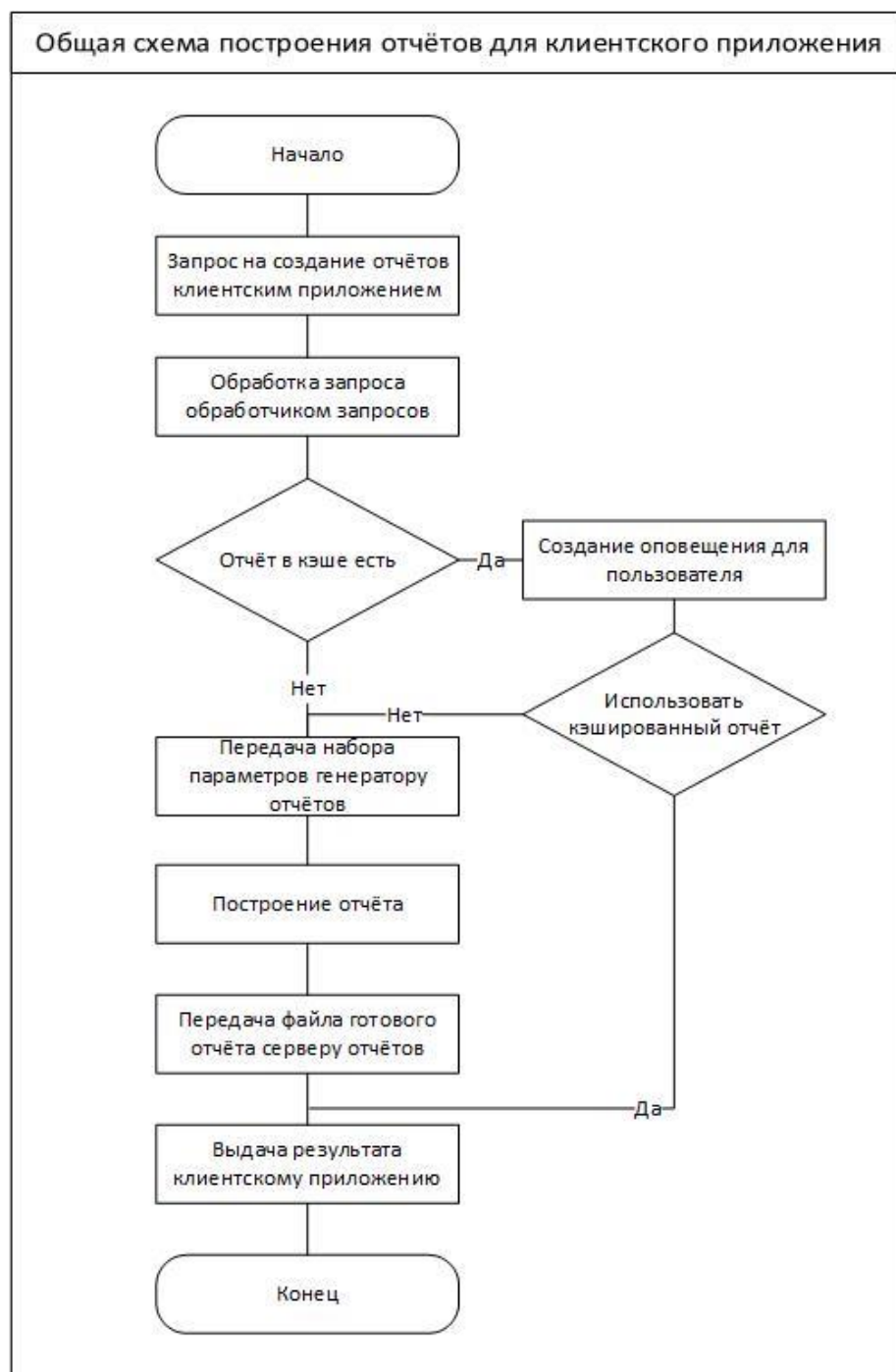


Рисунок 2.3.1 – Общая схема построения отчётов для клиентского приложения

При необходимости построения отчёта клиентское приложение передаёт серверу отчётов набор параметров для его построения для проверки наличия отчёта в кэше. Обработчик запросов сервера отчётов проверяет, если отчёта в кэше нет, то сервер отчётов передаёт набор параметров генератору отчётов, а тот в свою очередь строит отчёт своими средствами и выдаёт его в готовом виде в

формате офисных приложений обратно серверу отчётов. Затем сервер отчётов передаёт по API результат построенного отчёта клиентскому приложению.

Если же отчёт в кэше есть, то сервер отчётов предлагает пользователю одно действие из двух на выбор: или пользователь создаёт новый отчёт, или изменяет отчёт, найденный в кэше по заданным параметрам.

2.4 Разработка алгоритма построения отчётов по расписанию

Схема построения отчётов по расписанию представлена на рисунке 2.4.1.

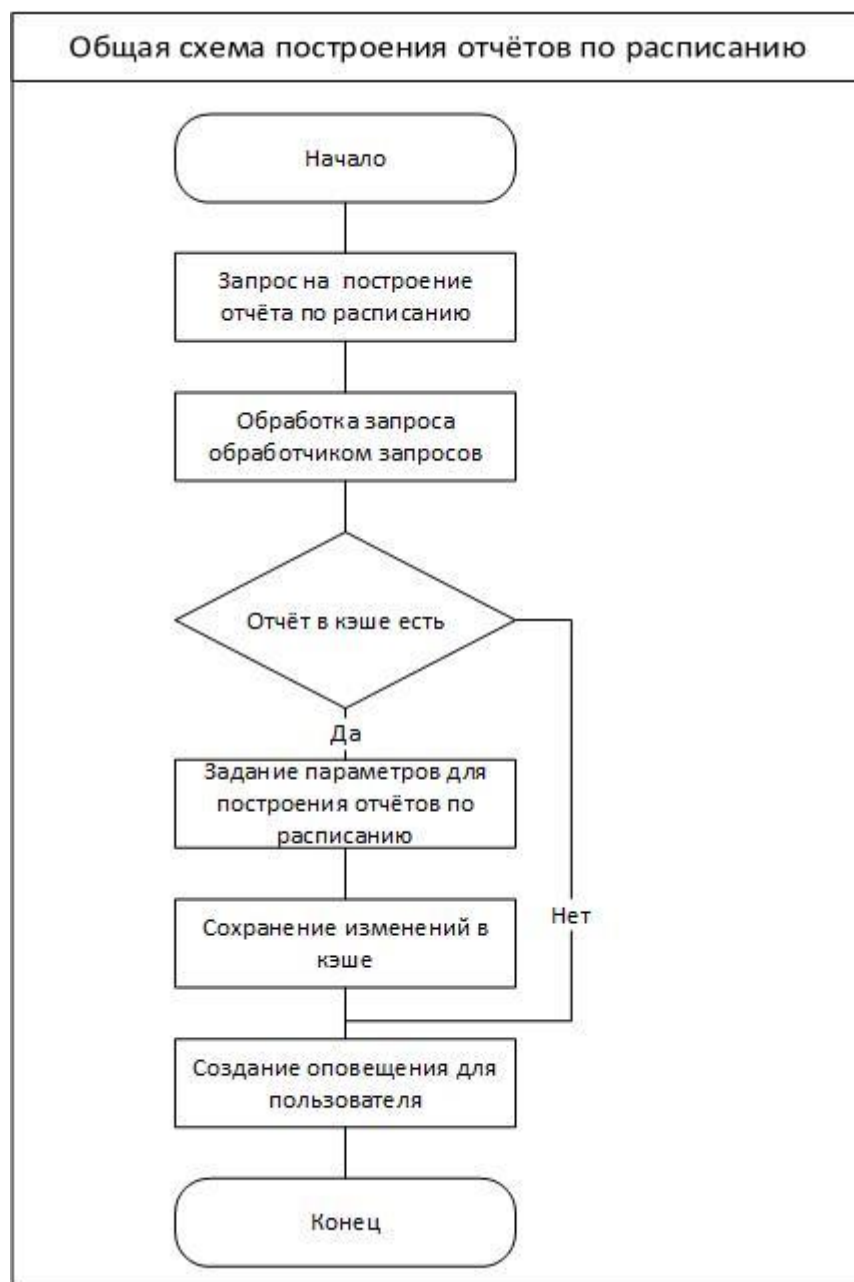


Рисунок 2.4.1 – Общая схема построения отчётов по расписанию

При необходимости планирования построения отчёта по расписанию клиентское приложение передаёт запрос серверу отчётов. Обработчик запросов сервера обрабатывает запрос и проверяет истечение срока действия кэшированной копии запрашиваемого отчёта. Если срок действия кэша не истёк, то система расписаний задаёт для отчёта необходимые параметры для его построения по расписанию (время, дата, тип обновления и др.), иначе, система расписаний задаёт параметры для истечения срока действия кэшированной копии отчёта. Затем отчёт, построение которого запланировано по расписанию, сохраняется обратно в кэш, и создаётся оповещение для пользователя с параметрами запланированного отчёта, которое он сможет получить по электронной почте. После получения оповещения пользователь самостоятельно решает, хочет ли он только просмотреть полученный отчёт или хочет его сохранить на ПК.

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ УНИФИЦИРОВАННОГО СЕРВЕРА ОТЧЁТОВ

3.1 Требования к аппаратным и программным средствам

Для функционирования унифицированного сервера отчётов необходимо наличие:

- 1) Генератора отчётов XLFormBuilder;
- 2) Библиотеки ADO;
- 3) Microsoft Excel версии 2003 или выше;
- 4) Клиентского приложения, через которое можно вызывать построение отчёта.

3.2 Подсистема администрирования сервера отчётов

Администратор системы должен иметь возможность войти в клиентское приложение под своей учётной записью Windows, чтобы защитить от других лиц используемую информацию.

Главное окно подсистемы администрирования сервера отчётов, представленное на рисунке 3.2.1, состоит из следующих элементов:

- главное меню программы;
- панель инструментов;
- вкладка выбора пользователя **Дерево пользователей**;
- вкладка выбора действий для пользователя **Дерево действий**.

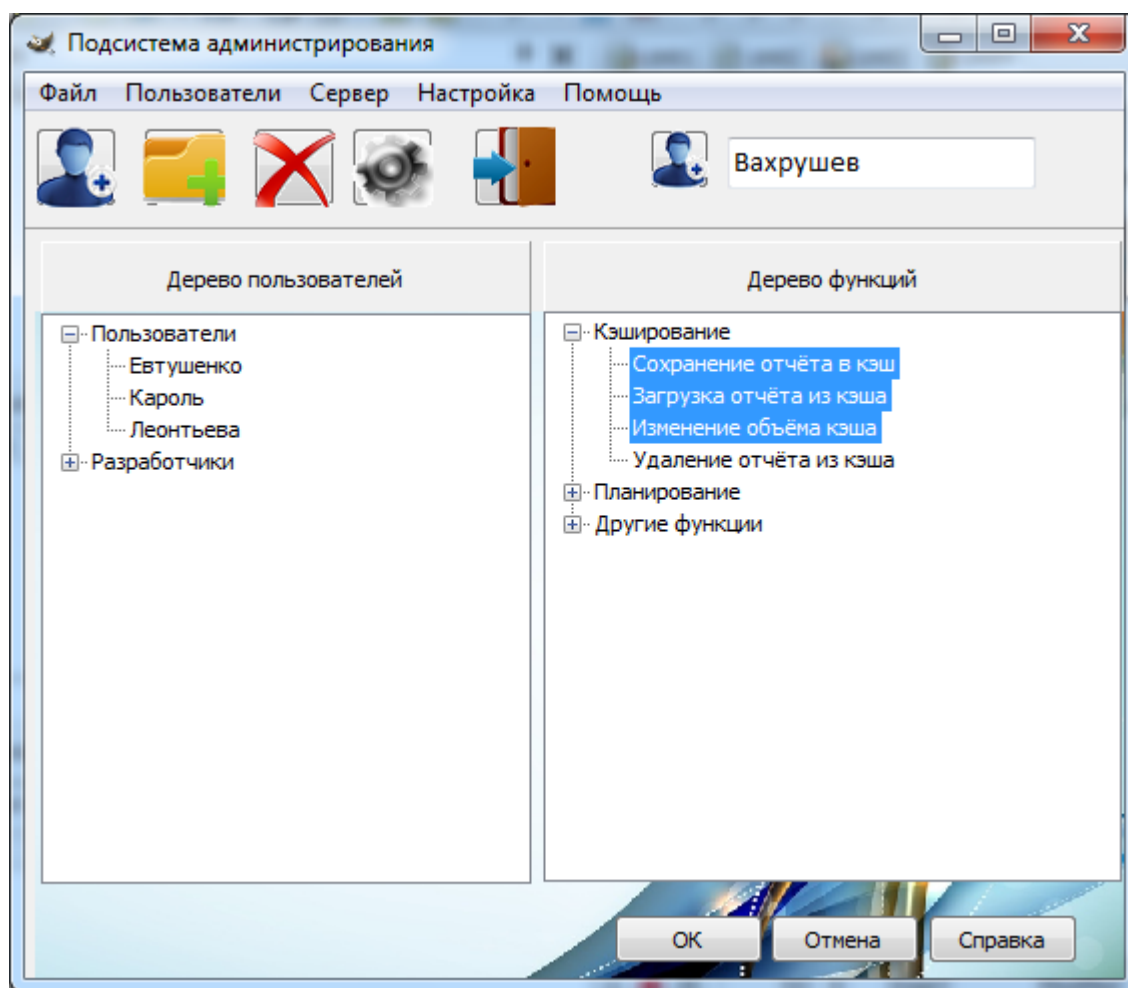


Рисунок 3.2.1 – Главное окно подсистемы администрирования сервера отчётов

Основными функциями подсистемы администрирования являются: настройка параметров кэширования и настройка планирования построения отчётов по расписанию.

3.2.1 Настройка параметров кэширования

Окно параметров функции кэширования представлено на рисунке 3.2.2.

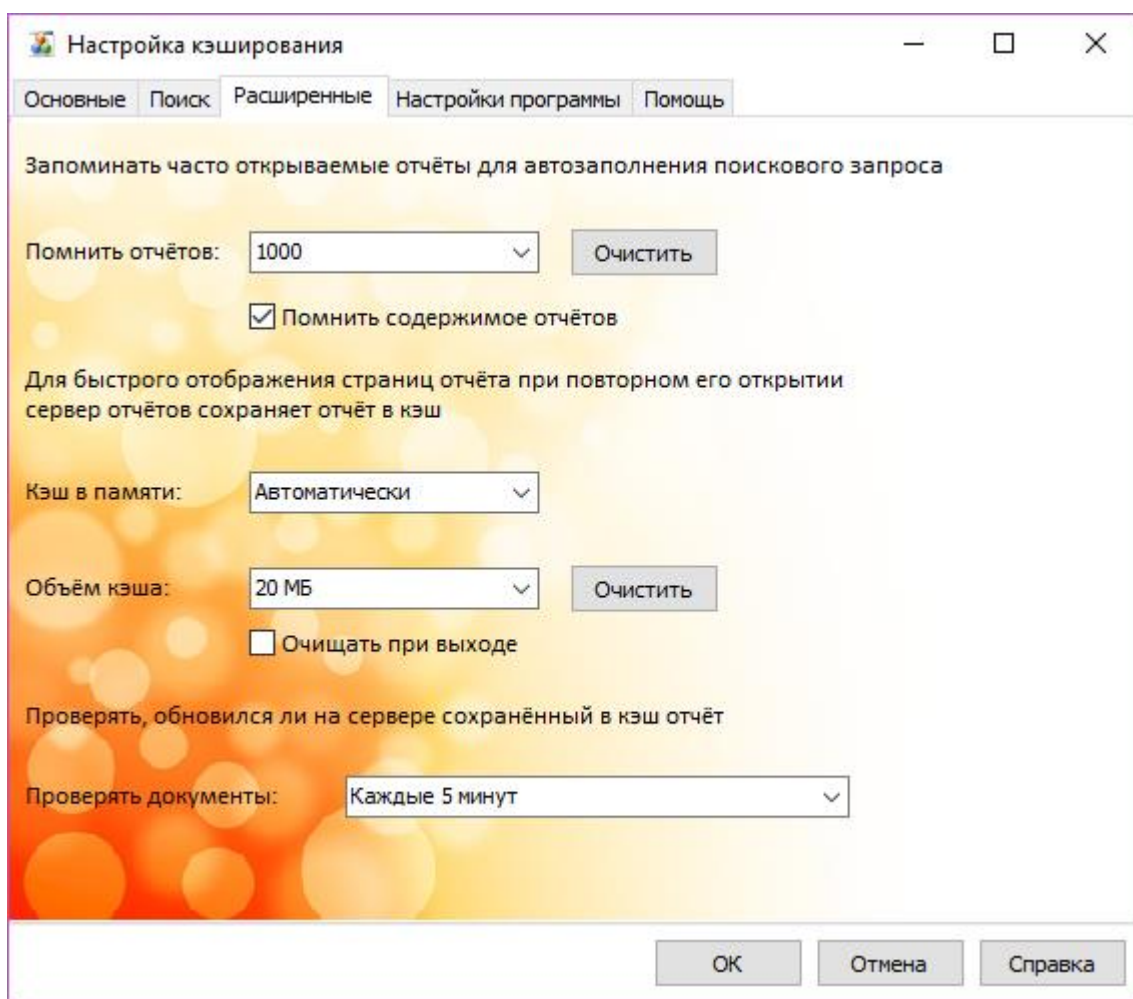


Рисунок 3.2.2 – Окно параметров функции кэширования

Во вкладке **Основные** можно просмотреть имеющиеся у пользователя отчёты и выбрать из них нужные для сохранения в кэш.

Во вкладке **Поиск** можно осуществить поиск по кэшу необходимого отчёта для его дальнейшего просмотра, скачивания или удаления из кэша. Кроме того, здесь можно увидеть часто просматриваемые кэшированные копии отчётов и их параметры, установить срок их действия и время обновления.

Во вкладке **Расширенные** можно указать следующие параметры кэширования:

- объём используемого кэша;
- длительность хранения отчётов в кэше;
- период очистки кэша.

Также во вкладке **Расширенные** доступны функции незамедлительной очистки кэша и проверки его обновления.

3.2.2 Настройка планирования по расписанию

На рисунке 3.2.3 представлено окно параметров функции планирования по расписанию.

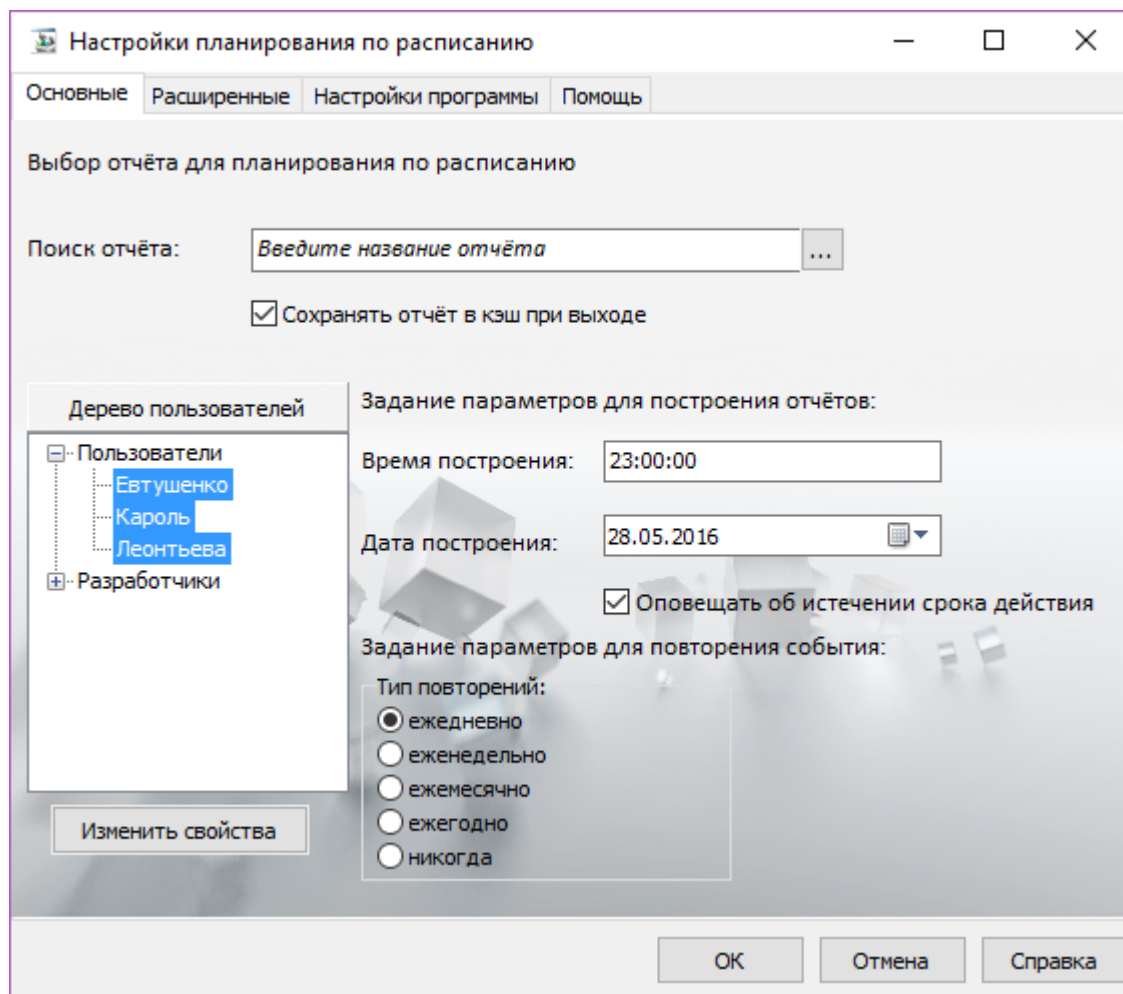


Рисунок 3.2.3 – Окно параметров функции планирования по расписанию

Во вкладке **Основные** можно осуществить поиск необходимого отчёта для дальнейшего планирования его построения по расписанию. Также здесь можно выбрать пользователей из **Дерева пользователей**, для которых необходимо запланировать построение отчёта, и указать их почту для доставки отчёта, либо пользователей, которых нужно информировать об истечении времени построения отчётов.

Кроме того, во вкладке **Основные** доступны функции:

- задания времени и даты построения отчёта;
- планирование истечения срока действия кэшированного отчёта;
- тип повторения (ежемесячно, еженедельно или ежедневно).

Во вкладке **Расширенные** имеются функции, позволяющие:

- запланировать доставку отчётов в стандартной или управляемой данными подписке;
- запланировать отчёты, создаваемые по расписанию, так, чтобы изменения, вносимые в них пользователями во время запланированного построения, добавлялись к этим отчётам через регулярные промежутки времени;
- запланировать время обновления данных кэша.

3.3 API унифицированного сервера отчётов

Реализация API унифицированного сервера отчётов будет сводиться к двум основным этапам:

- Реализация клиент-серверного API для взаимодействия клиентского приложения и сервера отчётов;
- Исполнение запроса от клиентского приложения на сервере отчётов.

После исполнения запросов клиентское приложение должно только получать либо ответ об успешном выполнении, либо информацию об ошибке.

API реализовано с помощью именованных каналов PIPE, которые используются в операционных системах Windows для передачи данных между процессами. Они позволяют организовать передачу данных между локальными процессами и между процессами, запущенными на различных рабочих станциях в сети. Кроме того, PIPE являются каналами для взаимодействия одного сервера и нескольких клиентов, подключаемых к нему по имени канала. Передача данных в PIPE будет происходить как синхронно, так и асинхронно.

Структурная схема передачи запроса от клиентского приложения серверу отчётов через API по каналам PIPE представлена на рисунке 3.3.1.



Рисунок 3.3.1 – Структурная схема передачи запроса от клиентского приложения серверу отчётов

Клиентское приложение отправляет данные с идентификатором и параметрами отчёта серверу для создания отчёта по API УСО в виде файла формата XML, синтаксис которого имеет вид:

```

<?xml version="1.0" encoding="UTF-8"?>
<request>
  <action type="list" id="80085">
    <structure name="references.reports ">
      <fields>
        <id /> <!--идентификатор отчёта -->
        <name /> <!-- название отчёта -->
      </fields>
    </structure>
  </action>
</request>
  
```

```

        <date /> <!--дата создания отчёта -->
    </fields>
</structure>
</action>
</request>

```

Каждый запрос серверу отчётов на выборку данных будет содержать следующие параметры:

format	Формат ответа, который будет принимать значение " xml ".
fields	Перечисленные через запятую наименования полей, которые необходимо вернуть в результате запроса. Т.е., например, если вам необходимо получить только ID и название рассылок, достаточно указать fields=id,title в запросе к интерфейсу управления рассылками. По умолчанию возвращаются все поля записей.
fields_exclude	Перечисленные через запятую наименования полей, которые необходимо исключить из результата запроса. Т.е., например, если вам необходимо получить все поля из списка шаблонов, кроме достаточно объемных html_body и text_body , необходимо указать в запросе параметр fields_exclude=html_body,text_body .

Ответ сервера также будет получен в формате **XML**. Формат ответа:

```

<response>
  <result></result>
  <error code="" message=""/>
  <данные>
    ...
  <данные>
</response>

```

result	Результат исполнения запроса. 1 - запрос исполнен удачно (не возникло системных ошибок) 0 - запрос не исполнен или исполнен с ошибками
error	В случае если во время исполнения запроса возникли ошибки, это поле содержит код ошибки (атрибут code) и текст сообщения об ошибке (атрибут message). Если запрос исполнен без ошибок, поле error в ответе отсутствует.
данные	Набор полей, содержащих данные ответа API. Набор специфичен для каждого интерфейса (может отсутствовать, если интерфейс не возвращает никаких данных).

Для всех входных и выходных данных API используется кодировка UTF-

8. Данные передаются через именованные каналы PIPE. При передаче данных от клиентского приложения серверу отчётов создаётся новый канал PIPE и проверяется, дошёл ли запрос серверу отчётов. В случае, если он дошёл, канал закрывается, и пользователю приходит сообщение об успешном выполнении, иначе, происходит подключение к серверу отчётов. Если подключение выполнено успешно, то запрос от клиентского приложения серверу отчётов отправляется заново, иначе канал закрывается, а клиентскому приложению приходит сообщение об ошибке.

В таблице 3.3.1 перечислены основные функции интерфейса именованного канала PIPE.

Таблица 3.3.1 – Основные функции интерфейса именованного канала PIPE

Функция	Назначение
CallNamedPipe	Выполнить одну транзакцию
ConnectNamedPipe	Соединить сервер с каналом
CreateFile	Открыть канал
CreateNamedPipe	Создать именованный канал
DisconnectNamedPipe	Закончить обмен данными
GetNamedPipeHandleState	Получить состояние канала
GetNamedPipeInfo	Получить информацию об именованном канале
PeekNamedPipe	Копировать данные канала
ReadFile	Читать данные из канала
SetNamedPipeHandleState	Изменить характеристики канала
TrasactNamedPipe	Писать и читать данные канала
WaitNamedPipe	Определить доступность канала
WriteFile	Писать данные в канал

Все функции именованного канала PIPE можно разбить на три группы:

- функции управления каналом (создать канал, соединить сервер с каналом, открыть канал, получить информацию об именованном канале, получить состояние канала, изменить характеристики канала);
- функции обмена данными (писать в канал, читать из канала, копировать данные канала);
- функции для работы с транзакциями.

3.4 Генератор отчётов

В рамках данной работы для примера генератора отчётов был выбран программный продукт XLFormBuilder, разработанный лабораторией ГИС. Он предназначен для создания шаблонов отчётных форм и формирования на их основе отчётов, служащих для вывода необходимой информации, хранящейся в БД, в документы Microsoft Excel в заданном виде.

Шаблон Microsoft Excel, создаваемый с помощью XLFormBuilder, представляет собой электронную таблицу, которая, помимо обычных констант, формул и форматирования Excel, содержит ссылки на извлекаемые данные и параметры, определяемые пользователем, а также опции книги, листов, и отдельных областей отчета.

Для доступа к данным в программе FormBuilder используется технология Microsoft ActiveX Data Objects (ADO), которая обеспечивает единый способ доступа к источникам данных различных типов. Для выборки данных из источников данных, используемых в шаблоне, создаются SQL-запросы на диалекте языка SQL, применяемом в конкретной СУБД.

Непосредственно для генерации отчётов в приложении FormBuilder применяются бесплатные компоненты из набора TMS FlexCel Studio. Этим определяется набор возможностей, доступных пользователю при проектировании шаблона.

Файл со сгенерированным отчётом XLFormBuilder передаёт серверу отчётов через соответствующий драйвер по своему API в виде конфигурационного файла сеанса, синтаксис которого представлен ниже:

XLFormBuilder.exeConfigurationFile

ConfigurationFile — полное (с путём) имя конфигурационного файла сеанса с расширением **.ini**.

Конфигурационный файл сеанса по формату является стандартным ini-файлом Windows в кодировке Windows 1251. В таблице 3.4.1 описываются разделы и параметры разделов.

Таблица 3.4.1– Разделы и параметры разделов

Раздел, параметр раздела	Описание
[Common]	Общие параметры
InputFile	Полный путь к файлу шаблона (*.xls). Если задан, XLFormBuilder при запуске начнёт генерацию отчёта, иначе запустится в полноэкранном режиме.
OutputFile	Полный путь к файлу отчёта (*.xls) для сохранения. Если задан, XLFormBuilder при генерации отчёта сохранит его в указанный файл, иначе построенный отчёт будет открыт в Microsoft Excel.
SessionID	Произвольная строка, значением которой в запросах шаблона будет заменена псевдопеременная @SessionID
UserInfo_Name	Имя пользователя, который генерирует отчёт. Если не задано в конфигурационном файле сеанса, то подставляется имя текущего пользователя Windows. Может использоваться в теле шаблона с помощью параметра #.UserInfo_Name, который можно поместить в ячейку Excel, а также в любую область колонтитула листа Excel. Может использоваться в теле запроса через псевдопеременную @UserInfo_Name, которая при выполнении запроса будет заменена строковым значением параметра (в апострофах).
UserInfo_Login	Login пользователя, который генерирует отчёт. Если не задано в конфигурационном файле сеанса, то подставляется login текущего пользователя Windows в формате SAM-Account-Name, он же Down-LevelLogonName, то есть 'NetBIOSDOMAIN\UserName'. Может использоваться в теле шаблона с помощью параметра #.UserInfo_Login, который можно поместить в ячейку Excel, а также в любую область колонтитула листа Excel. Может использоваться в теле запроса через псевдопеременную @UserInfo_Login, которая при выполнении запроса будет заменена строковым значением параметра (в апострофах).
ProgressWindowHandle	Дескриптор окна (HWND), в которое XLFormBuilder должен посылать сообщения об этапе построения отчёта:
	PostMessage(ProgressWindowHandle, 1025, EstimatedSeconds , Stage)
	где EstimatedSeconds — примерное количество секунд для построения отчёта; Stage — код этапа построения: 1 = подготовка; 2 = запрос пользовательских параметров; 3 = начато выполнение (очередного) запроса; 4 = окончено выполнение (очередного) запроса; 5 = открытие.

Раздел, параметр раздела	Описание
	Если параметр ProgressWindowHandle не задан, XLFormBuilder пытается определить его путём поиска окна с классом 'TReportWaitingForm' или заголовком 'Построение отчёта'.
[ADOConnection]	Параметры ADO-соединения, с которыми будет выполняться подключение к базе данных. Если не задано, используются настройки именованного соединения
Provider	Provider=SQLOLEDB.1 (Для Microsoft SQL Server)
Integrated Security	Integrated Security=SSPI (аутентификация Windows)
Data Source	Data Source=Имя сервера
Initial Catalog	Initial Catalog=Имя БД
...	И другие допустимые для ADO-соединения параметры при необходимости
[UIParameters]	Значения для подстановки в пользовательские параметры отчёта, если таковые есть. Если эти значения заданы, у пользователя они не запрашиваются. Должны быть заданы значения для всех пользовательских параметров, либо не задано ни одно. Подстановка лишь некоторых значений недопустима. Значения пользовательских параметров в разделе [UIParameters] должны быть заданы в том же порядке, что и сами пользовательские параметры в шаблоне. Пример: [UIParameters] 1 Месторождение=6EC0E412-26E4-499D-B194-142C995E2A81 Год=2015 Месяц=5
[Selection]	Используется для передачи из клиента в шаблон информации, которую по какой-либо причине неудобно или невозможно передавать через пользовательские параметры, например, если нужно передать целый список значений, размер которого заранее неизвестен. Ключи и значения в данном разделе формируют наполнение таблицы-переменной @Selection, которую можно использовать в запросах шаблона. Таблица-переменная @Selection имеет два поля: [Key] VARCHAR(100), [Value] VARCHAR(100) Эти поля автоматически заполняются из названий и значений параметров данного раздела, например: [Selection] 7C5954CA-F7BA-4E90-988E-B5A13C275163=План 71C50668-5776-4B97-94B6-43CD79E80070=Объект Следующий запрос даст эти же ключи и значения в виде таблицы:

Раздел, параметр раздела	Описание
	SELECT [Key], [Value] FROM @Selection

Конструктор генератора отчётов XLFormBuilder также учитывает возможность форматирования содержимого отчёта: размера, стиля шрифтов, их цвета, цвета фона, положения на листе и т.д.

Форматирование листа XLS производится автоматически перед выдачей его пользователю: лист отчёта вписан в область печати, т.е. данные равномерно распределены по ширине листа, если не указаны другие параметры.

В рамках разработки корпоративной информационной системы были созданы шаблоны отчётов, использование которых проверялась с помощью сервера отчётов и генератора отчёта. Пример созданного отчёта по определённому набору параметров, переданному от сервера отчётов генератору отчётов XLFormBuilder, с использованием шаблона отчёта, представлен на рисунке 3.4.1.

МЭР по добыче нефти — новый1 [Режим совместимости] - Microsoft Excel

Гарнизон	Вставка	Разметка страницы	Формулы	Данные	Рецензирование	Вид	Тема	Условное форматирование	Форматировать как таблицу	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	Стили	
----------	---------	-------------------	---------	--------	----------------	-----	------	-------------------------	---------------------------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

Рисунок 3.4.1– Пример созданного отчёта

ЗАКЛЮЧЕНИЕ

На сегодняшний день во всех корпоративных информационных одной из серьёзных проблем построения отчётности является проблема ресурсоёмких отчётов, построение которых занимает длительное время, что создаёт недопустимо большую нагрузку на систему.

В рамках научно-исследовательской работы были выполнены следующие задачи:

- Проведён анализ предметной области;
- Проанализирована проблема ресурсоёмких отчётов;
- Сформулированы функциональные требования к унифицированному серверу отчётов, способному решить проблему ресурсоёмких отчётов;
- Разработан унифицированный сервер отчётов;
- Выполнена апробация унифицированного сервера отчётов в рамках создания ЕАМ-системы;
- Проведена оценка полных денежных затрат на исследования и приближенная экономическая оценка результатов их внедрения;
- Рассмотрены вопросы выполнения требований к безопасности и гигиене труда, к промышленной безопасности, охране окружающей среды и ресурсосбережения.

СПИСОК ПУБЛИКАЦИЙ

Список публикаций по теме магистерской диссертации:

1. Костенко К.А. Development of the server of reports for corporate information systems // VIII Фестиваль Microsoft в ТПУ. XIII Всероссийская научно-практическая конференция студентов, аспирантов и молодых ученых «Технологии Microsoft в теории и практике программирования», г. Томск, 22 марта – 23 апреля 2016 г.

2. Костенко К.А. Унифицированный сервер приложений для построения отчётов // Информационные технологии в науке, управлении, социальной сфере и медицине. III Международная научная конференция, г. Томск, 23- 26 мая 2016г.

3. Костенко К.А. Применение серверов отчётов в корпоративных информационных системах. Перспективы развития фундаментальных наук. Сборник трудов XIII Международной конференции студентов, аспирантов и молодых ученых «Перспективы развития фундаментальных наук». Томск, 26-29 апреля 2016 г. – Томск: Изд-во ТПУ. – Т. 7 – 78-80с.

Другие публикации:

4. Kostenko K.A. Statistical analysis of individual tasks on probability theory // Информационные технологии в науке, управлении, социальной сфере и медицине. III Международная научная конференция. Томск, 23 – 26 мая 2016г.

5. Костенко К.А. Статистический анализ вариантов индивидуальных контрольных заданий по теории вероятностей // Молодежь и современные информационные технологии: сборник трудов XIII Международной научно-практической конференции студентов, аспирантов и молодых ученых. Томск, 9-13 ноября 2015 г.– Томск: Изд-во ТПУ. – Т.1 – 67-68с.

6. Костенко К.А., Брезгулевский Е.Д. Компетенции системного инженера. // Молодежь и современные информационные технологии: сборник трудов XIII Международной научно-практической конференции студентов, аспирантов и молодых ученых. Томск, 9-13 ноября 2015 г.– Томск: Изд-во ТПУ. – Т.2 – 28-29с.

ПРИЛОЖЕНИЕ А

Раздел 1. Обзор предметной области (Overview of subject domain)

Студент:

Группа	ФИО	Подпись	Дата
8ИМ4А	Костенко Карина Алексеевна		

Консультант кафедры ВТ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Мирошниченко Евгений Александрович	к.т.н.		

Консультант – лингвист кафедры ИЯИК:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ст. преп. каф. ИЯИК	Шепетовский Денис Владимирович			

1 ANALYSIS OF THE DOMAIN

Among the available decisions in the market of information systems for the resource-intensive reports creation using centralized application servers for reports creation there is a set of reports generators giving for this purpose insignificant opportunities as little-known which are usually used within particular system, and enough powerful and popular systems which are used for various systems, for example: Oracle Reports, Crystal Reports, Stimulsoft Reports, Fast Reports, Microsoft SQL Reporting Services.

We will consider each of them from the point of view of existence of means for the solution of resource-intensive reports' problem.

1.1 Oracle Reports Developer

The Oracle Reports Developer software product developed by Oracle corporation has a server solution for creation of reports – Oracle Reports Server. Oracle Reports Server is used when generation of reports demands a long time and considerable resources and therefore it is carried afterhours and/or automatically according to the schedule with a possibility to deliver the reports to users by e-mail, and/or in case of necessary providing the same data in the form of reports to different users. It in turn demands a caching of results of performance of inquiries.

The idea of a caching in Oracle Reports Server is that the data, which are often requested by users, is saved in a cache, but are not taken anew at each inquiry. It allows caching both static and dynamic data and contains procedures for a task of conditions under which data in a cache should be updated. This or that information taking into account a concrete situation comes back to Oracle Reports Server at the known dependences between data and certain users or applications. Oracle Reports Server allows compressing HTML pages of reports that accelerates their delivery to clients.

The data processed by Oracle Reports Server for reuse are located in the divided pool, which caches various designs, which are used by several users everywhere,

including SQL inquiries shown by users and fragments of SQL inquiries, and results of their performance. The volume of the divided pool can automatically extend.

The small volume of the divided pool is the stronger, it is reflected in its productivity as felt by all users. It leads to the fact that:

- Not enough information about the created reports is located to a cache that involves the frequent appeal to a disk for inquiry and updating of information in the dictionary.
- It is not possible to cache enough SQL commands that leads to "Shake-up" of memory when one reports are forced out from a cache for release of the place to other reports. If memory hasn't enough for a caching, then the same reports should be sorted, cached and pushed out again and again, senselessly spending resources of the processor and increasing overhead costs of performance of each transaction.
- It is not possible to cache enough inquiries that also leads to "Shake-up" of memory and decline in production when performing the code, which is stored in the database.

It is possible to cache reports in Oracle Reports Server during the set time when in response to the next inquiry already ready report is returned instead of repeated implementation of the addressing the database, necessary for its generation.

Besides, Oracle Reports Server allows not only to create reports, but also to start them according to the schedule during the periods of low loading then end users at any time can see the received results [2].

1.2 Crystal Reports

Entering mechanism of the Crystal Reports task scheduler allows users automatically to generate reports, according to the set schedule, and is intended first of all [3]:

- for implementation of the difficult reports demanding considerable server resources, including reports with the opened subreports as printable versions.
- for implementation of regular (cyclic) reports which need to be formed in certain time and/or day (weeks, month).

Also the mechanism of the scheduler of tasks allows to carry out reports practically in any format and for any destination (the file, the printer, e-mail, ftp) and events which can be three types:

- the file events arising at emergence of the set files in the set catalog;
- the scheduler's events allowing to organize interdependent implementation of reports;
- events on command which are transferred from the console of the control of system (Crystal Management Console) arising at start on implementation of all reports expecting this event, by the person having administrative privileges.

Thanks to a server solution, Crystal Reports Server, which is a part of Crystal Reports the user has an opportunity to cache reports. The caching of reports occurs as follows: Crystal Reports Server supports effective processing of reports with use of the server of pages, that in turn bears responsibility for responses to requests of pages, carrying out as required processing of reports and their generation. At the same time the user receives only the required page, but not all report. It considerably increases productivity and reduces a network traffic for big reports, however allows the user only to check the report, but not to unpack it.

Crystal Reports Server also supports active sharing of data which means that reports can use joint data at the addressing the same data. At the same time such reports can be generated without additional inquiries to the database (further by a DB), even at difference of one report from another. Significant increase of productivity within all system, including the database is result. The main advantage of active sharing of data is reduction of loading and intensity of use, fuller caching of data, with ensuring more effective system work.

The caching server, Crystal Reports Server, bears responsibility for processing of all requests for viewing of reports. It checks whether it is possible to execute inquiry with use of the cached page of the report. If it is impossible, it requests generation of this page from the server of pages.

1.3 Stimulsoft Reports

Stimulsoft Reports also has the server solution for problems of creation of resource-intensive reports: Stimulsoft Report Server.

At a caching of reports Stimulsoft Report Server uses temporary files for storage of intermediate results thanks to what reports are under generating without excessive consumption of random access memory. Thanks to application of a multithreading creation of reports with caching happens much quicker as the kernel of the generator of reports starts several streams: the main and auxiliary [4]. Creation of the report is carried out in the main stream; work with temporary files of a cache is carried out in auxiliary stream. As a result, time of the report creation, with a caching or without, practically coincides.

The planned data processing according to the schedule in Stimulsoft Report Server happens by means of the built-in scheduler. The scheduler of Stimulsoft Report Server allows programming any event on time – from simple single start on time to national holidays, with shift on time. At the same time with its help it is possible to adjust cyclic repetitions of reports and their exception. The object of the scheduler "calendar" allows collecting the list of the dates, which are not connected logically with each other. At the same time it considers time zones and has a deadline of execution of actions as which it is possible to start reports, to copy objects, to send post messages for informing users on events and to start the enclosed schedulers. All these actions can be created within one scheduler and be carried out consistently without the slightest intervention of the end user or the administrator of the server of reports.

Besides, important feature of Stimulsoft Report Server is an opportunity to work with multipage documents. Stimulsoft Report Server leads to a compact look such documents.

1.4 Fast Reports

The caching of reports and their data in Fast Reports occurs as in memory that allows increasing speed, and in files on a disk that is necessary for economy of resources of the RAM [5].

There are following types of a caching in Fast Reports:

1. All text and objects of the constructed report are saved to the temporary file on a disk, at the same time it is specified how many megabyte is taken away under a template in the RAM.

2. In memory it can be stored both the whole report, and some of its pages, that significantly accelerates preview work, but spends a lot of memory (especially if there are drawings in a template).

3. All pictures of the constructed report are saved to the temporary file on a disk that leads to reduction of consumption of memory in reports with a large number of drawings, but reduces speed.

Except function of a caching Fast Report gives to users an opportunity of the automated work with reports and their planned drawing up. The main opportunities includes a regularity of formation of reports, their saving and mailing. In a configurator of the scheduler of tasks, the user can set the properties and actions connected with start of the necessary report as well as date and time of his performance and send the report on e-mail in one of convenient formats (PDF, HTML, TXT, RTF, GIF, BMP). Besides, at the scheduler there is a possibility of control of periodical start of a task.

1.5 Microsoft SQL Reporting Services

Reporting Services allow users to transfer part of load of the DB server to other server for improvement of productivity of all system [6]. Besides, these services allow to separate basic processing of the report, including receiving and placement of data, from transformation of the report to the required format [7]. This process is a caching.

The caching in Reporting Services allows reducing time necessary for search of the necessary reports which volume is big or which often address [7]. Contents of a

cache are nonvolatile; it means that it can change during the replacing, addition or removal of reports. Reporting Services store temporary files in a DB thanks to what support of the user sessions and processing of reports is provided. These files are cached for internal use.

The cached copy of the report is based on an intermediate format of the report. The server of reports in general caches only one copy of the report on the basis of its name, but if it contains data which depend on inquiry's parameters, then in this case several versions of the report in due time will be cached already.

The cached report is replaced with newer version when the user chooses the report, the cached copy's period of validity has expired [8]. Reports that are adjusted for performance in the form of the cached copies are removed from a cache through the certain periods set in settings of expiration. At the same time, if the user does not use SOAP API, then directly he will not be able to remove the report from a cache.

Microsoft SQL Reporting Services allow creating reports on the schedule that is especially important when processing large volumes of data. A set of reports can distribute the general scheduler of processing, for example, of performance of generation of the monthly or quarterly reporting. The user can set the schedule only for a certain report or use the general schedule for several subscriptions [9]. At the same time for reports with parameters, he has to define the values appropriated to parameters at start of a subscription.

Period of validity of the schedule expires without notice [10]. After date of its termination, reports aren't carried out any more. At the same time schedules with the expired validity period aren't removed, the user can remove them only manually. Thus, if it is necessary to prolong the schedule, then it is necessary to extend date of his termination.

1.6 Disadvantages of the considered servers of reports

Opportunities of the considered servers of reports are insufficient for the solution of all problems, which are connected, with creation of reports. We will list these problems.

1. Problem of the cache's validity. As the data taken from the database and used in reports change out of systems of the reporting third-party applications. The server of reports can't monitor this process of change and can't establish the changed data which in it then will be used.

2. Lack of a possibility of a direct control of reports caching. The resource-intensive reports kept in a cache can be actual only during certain time on which expiration they are forced out from a cache by other, newer reports. Therefore, reports are required to be generate regularly, and its generation in turn can continue from several hours to several days that slows down work of the user. Such problem periodically arises in Oracle Reports Server where users still can work with the irrelevant version of the report, and then have to either keep the new version of the report, or remove the report from a cache. Situation is similar in Microsoft SQL Reporting Services where for delivery to a cache of new copies of reports requires expiration of old copies of reports.

3. Lack of a possibility of the report creation on the schedule with the saving it to a cache. The reports generated according to the schedule in these systems of the reporting, passing preservations in a cache, are sent to all declared users and as to users whom these reports are necessary at present, and to users whom they aren't necessary. At the same time in all systems of the reporting it is in advance unknown when, by whom and how the data used in reports have been changed, that is why it is difficult to define the version of the report constructed according to the schedule is actual or not. In Microsoft SQL Reporting Services the reports generated according to the schedule remain in a cache in an intermediate format that allows users for whom they are necessary, independently to take reports from a cache.

4. There is not all necessary combinations of events' frequency. For example, if the user needs to start the report every Thursday at 14:00 and at 18:00, then it is necessary to create two daily schedules starting on Thursday and for one to specify time of start 14:00, and for another – 18:00. In Crystal Reports the reports generated according to the schedule though remain in a cache, however the interval of their storage is short-lived and varies from 20 to 60 minutes.

5. Lack of a possibility of saving reports to formats of Microsoft Office applications (Word, Excel, etc.) is not in all reporting systems though it is the important condition for resource-intensive reports creation allowing giving great opportunities for printing and sending by mail. It is especially actual for products of Oracle Reports Developer and Fast Reports where there is only own format of saving reports. In software products Stimulsoft Reports, Crystal Reports and Microsoft SQL Reporting Services reports though don't remain in formats of office applications, but can be converted into them. Converting of formats of reports get disadvantages too both in Crystal Reports, and in Microsoft SQL Reporting Services lost some data at registration, such as information using at creation of schedules and charts, and some internal data, such as formulas in the calculated fields at converting of reports.

6. Considered reporting systems cannot serve as any unified solution, which can be integrated with the desirable reports generator. The developer can get a necessity of the independent choice of the reporting generator suitable him by any criteria.

Thus, as in the existing systems of the reporting there are disadvantages connected with lack of management of a cache and impossibility to construct the report on the schedule with the subsequent its preservation in a cache, there is no guarantee that one constructed resource-intensive report won't be forced out from a cache by other constructed report. Therefore, these technologies fully don't allow solving a problem of resource-intensive reports.

1.7 Analysis of functional requirements of servers of reports

After the analysis of opportunities of the existing systems of reporting, the list of functional requirements to the created server of reports that will unite in itself not only advantages of these systems, but also will correct disadvantages, has been made. We will describe these requirements.

1. Reports have to be stored in a cache for a long time and the administrator can change the volume and the schedule of a cache. The resource-intensive reports creation can vary in intervals from several hours to several days, and at the user cannot be such opportunity to build reports anew and regularly. At the same time it is

impossible to define when as well as who has changed data in the report therefore control of the administrator of creation of reports and saving it in a cache for long term is simply necessary.

2. The user has to have a flexible possibility of schedule, for example during idle time in hours of the minimum load. At the same time, there may be or may not be previously created report in cache, and then is taken from it by the user if necessary. As in advance unknown whom the report can be necessary during the set time and whom can not. Moreover, the report can go to the person whom this report isn't necessary, and it not to come to the person, whom the report is necessary. Therefore, control of the administrator over reporting creation here too is necessary that the report built on the schedule saved to a cache, and then was unloaded only to those users whom it is necessary.

3. The user has to have a possibility to choice between report server is to generate the new report and whether he gets the report from the cache. At the report creation reports server has to check its availability in a cache: if there is not report, then the server must to creation it and give out to the user, if there is the report in a cache already, then the server has to notify the user about actions which will need to be made with the report (to rewrite old, to add new, to cancel addition). At the same time there is or there is not the reporting creation in a cache, therefore the user must to be in advance notified about availability of a cache.

4. The user has to have an opportunity to save reports in formats of Microsoft Office applications (Excel, Word, etc.), which are the most convenient for printing and sending by e-mail. Opportunities for the printing of reports given by office applications are more, than opportunities for the printing of the reports given by systems of reporting. For example, Microsoft Excel can offer the user the printing of both all document, and his separate pages, range of values, the filtered data, notes in cells, etc. Also Excel gives an opportunity to the printing of large-format documents, such as A3 or A1, entirely or in parts.

5. The report server has to have an opportunity to been integrated with reports generator. USR has to be engaged in the solution of questions of interaction between

the client application and the generator of reports, in particular carrying out connection to the generator of reports and providing the solution of the tasks connected with a caching of reports and construction them according to the schedule.

Thus, within research work it is necessary to develop the unified server of reports which will solve problems of a direct control of a caching and reports creation on the schedule with the moving its to a cache, solving at the same time resource-intensive reports' problem and to have ability to integrate with any necessary existing reports generator.

As the reports generator we will use XLFormBuilder, where connection will be carried out at the expense of the developed driver (the intermediate application) which will redirect to them inquiries from USR with some set of parameters for the reporting creation. The generated report of Reporting Services will be sent back by USR in the form of some file, and that, in turn, will solve: to place the report in a cache or to send to the user.

ПРИЛОЖЕНИЕ Б

Таблица Б.1 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Б _ф	Б _{к1}	Б _{к2}	К _ф	К _{к1}	К _{к2}
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Повышение производительности труда пользователя	0,09	5	2	5	0,45	0,2	0,5
2. Удобство в эксплуатации (соответствует требованиям потребителей)	0,08	4	2	5	0,45	0,3	0,5
3. Помехоустойчивость	0	0	0	0	0	0	0
4. Энергоэкономичность	0	0	0	0	0	0	0
5. Надежность	0,04	3	1	3	0,3	0,1	0,2
6. Уровень шума	0	0	0	0	0	0	0
7. Безопасность	0	0	0	0	0	0	0
8. Потребность в ресурсах памяти	0,04	3	1	3	0,3	0,15	0,3
9. Функциональная мощность (предоставляемые возможности)	0,1	5	2	4	0,5	0,2	0,4
10. Простота эксплуатации	0,1	5	2	5	0,5	0,2	0,3
11. Качество интеллектуального интерфейса	0,1	5	2	5	0,5	0,1	0,3
Экономические критерии оценки эффективности							
1. Конкурентоспособность продукта	0,05	4	2	4	0,25	0,15	0,25
2. Уровень проникновения на рынок	0,05	4	2	5	0,25	0,15	0,25
3. Цена	0,1	5	2	4	0,5	0,3	0,5
4. Предполагаемый срок эксплуатации	0,1	5	2	4	0,5	0,15	0,25
5. Послепродажное обслуживание	0	0	0	0	0	0	0
6. Финансирование научной разработки	0,1	5	2	4	0,5	0,3	0,5
7. Срок выхода на рынок	0,05	5	4	4	0,25	0,15	0,25
8. Наличие сертификации разработки	0	0	0	0	0	0	0
Итого	1	58	26	52	5,25	2,45	4,5

ПРИЛОЖЕНИЕ В

Таблица В.1 – Определение стоимости функций, выполняемых ИС

Выполняемая функция	Показатели				
	Трудоём- кость функции, нормо-ч	Стоимость функции, руб.	Заработ- ная плата, руб. (100 руб.час)	Себесто- имость, руб.	k
A1. Импорт/экспорт данных из БД	60	2200	6000	8200	0,124
A2. Создание отчёта в Excel по шаблону	70	2400	7000	9400	0,143
A3. Добавление/редактирование/удаление данных	50	2000	5000	7000	0,106
A4. Поиск совпадений	20	1400	2000	3400	0,052
A5. Вставка в отчёты диаграмм	2	200	200	400	0,006
A6. Вставка в отчёты рисунков	2	200	200	400	0,006
A7. Запрос отчёта из кэша	90	2800	9000	11800	0,179
A8. Сохранение отчёта в кэш	90	2800	9000	11800	0,179
A9. Составление графика построения отчётов по расписанию	70	2400	7000	9400	0,143
A10. Создание резервной копии отчёта	10	1200	1000	3200	0,049
A11. Восстановление отчёта из резервной копии	5	400	500	900	0,014
Итого (Общая себестоимость функций):				65900	1

Примечание: k – коэффициент относительных затрат на функцию.

ПРИЛОЖЕНИЕ Г

Таблица Г.1– SWOT-матрица

	Сильные стороны научно-исследовательского проекта: F1. Расширенный функционал при работе с БД: функции удаления, добавления и редактирования данных; F3. Дружественный гибкий и современный интерфейс пользователя; F4. Экономичность и гибкость технологии; F6. Низкие технологические и технико-эксплуатационные требования к функционированию системы; F7. Масштабируемость БД, созданная в бесплатной версии СУБД MS SQL Server 2008; F8. Наличие контроля целостности и актуальности при генерации отчёта; F9. Наличие API. F10. Использование служб MS SQL Reporting Services для гибкой генерации отчётов и работы с ними. F11. Написание представлений на языке SQL для улучшения визуального представления данных.	Слабые стороны научно-исследовательского проекта: L1. Локальное подключение к БД; L2. Ограниченность размера БД, созданной в бесплатной версии СУБД MS SQL Server 2008; L3. Обязательное требование наличия установленного бесплатного продукта MS SQL Reporting Services и среды разработки отчётов Business Intelligence на ПК для функционирования ИС, занимающие много свободного места на ПК.
Возможности: P1. Появление дополнительного спроса на продукт. P2. Снижение стоимости необходимых программных средств	Демонстрация клиентам возможностей, точности и удобства системы. Настройка многопользовательского режима при работе с данными, которая позволит одновременно нескольким пользователям выполнять операции.	Снижение цены на продукт. Проведение рекламной компании Расширение функциональных возможностей системы Применение новых технологий и продуктов.
Угрозы: D1. Отсутствие спроса на научную разработку; D2. Относительно высокая стоимость коммерческих лицензионных программных продуктов (MS SQL Server 2008, CrystalReport); D3. Ограничение во времени для реализа-	Продвижение продукции с акцентированием на достоинствах системы. Использование нелегальных версий продуктов. Повышение опыта и знаний у разработчиков для создания крупной и продвинутой ИС. Увеличение сроков реализации системы.	Поиск дополнительных опытных разработчиков для завершения реализации ИС. Поиск новых путей реализации технологии. Пересмотр ценовой политики.

ции расширенных возможностей ИС; D4. Несовместимость программных продуктов и технологий; D5. Отсутствие достаточного опыта в реализации корпоративных приложений для разработки крупной ИС.		
--	--	--

ПРИЛОЖЕНИЕ Д

Таблица Д.1 – Бланк оценки степени готовности ИС к коммерциализации

	Наименование	Степень проработанности научного проекта	Уровень имеющихся знаний у разработчика
1	Определен имеющийся научно-технический задел	5	5
2	Определены перспективные направления коммерциализации научно-технического задела	4	4
3	Определены отрасли и технологии (товары, услуги) для предложения на рынке	4	4
4	Определена товарная форма научно-технического задела для представления на рынок	5	5
5	Определены авторы и осуществлена охрана их прав	5	5
6	Проведена оценка стоимости интеллектуальной собственности	4	4
7	Проведены маркетинговые исследования рынков сбыта	4	4
8	Разработан бизнес-план коммерциализации научной разработки и определены пути продвижения научной разработки на рынок	4	4
9	Определены пути продвижения научной разработки на рынок	4	4
10	Разработана стратегия (форма) реализации научной разработки	5	4
11	Проработаны вопросы международного сотрудничества и выхода на зарубежный рынок	3	2
12	Проработаны вопросы использования услуг инфраструктуры поддержки, получения льгот	3	2
13	Проработаны вопросы финансирования коммерциализации научной разработки	4	3
14	Имеется команда для коммерциализации научной разработки	3	3
15	Проработан механизм реализации научного проекта	3	3
	Итого баллов	60	56

ПРИЛОЖЕНИЕ Е

Таблица Е.1 – Цель и результат проекта

Цель проекта:	Разработка унифицированного сервера отчётов для корпоративных ИС
Ожидаемые результаты проекта:	1. Внедрение ИС в ОАО «Сибирский центр высоких технологий» 2. Успешная защита магистерской диссертации 3. Опубликование результатов разработки в научных журналах 4. Демонстрация результатов разработки на конференциях, конкурсах
Критерии приёмки результата проекта:	Тестирование ИС на стороне Заказчика. Приёмка осуществляется с требованиями технического задания, согласованного с Заказчиком.
Требования к результату проекта:	Требование:
	1. Должна быть разработана БД, в которой будут храниться данные для создания отчётов
	2. Должно быть разработано клиентское приложение, предоставляющее API для управления такими данными
	3. Должен быть разработан сервер отчётов для хранения отчётов в кэше и предоставления их пользователям в требуемом виде
	4. В составе клиентского приложения должны быть разработаны 5 основных подсистем: импорт и экспорт данных, управление отчётами, планирование расписаний отчётов, запрос отчётов у сервера отчётов

Таблица Е.2 – Контрольные события проекта

№	Контрольное событие	Даты	Результат
1	Разработка технического задания	01.02.2016 – 05.02.2016	Разработанное техническое задание
2	Согласование и утверждение технического задания	06.02.2016 – 08.02.2016	Техническое задание, согласованное и утверждённое заказчиком
3	Выбор технологии для разработки ИС	09.02.2016 – 17.02.2016	Выбранные ПО, технологии и библиотеки для разработки ИС
4	Исследование предметной области	18.02.2016 – 25.02.2016	Результаты анализа существующих ИС
5	Календарное планирование работ по теме	26.02.2016 – 28.02.2016	Составленный план работ
6	Проектирование клиентского приложения	29.02.2016 – 13.03.2016	Эскизы интерфейсов пользователя, прототип клиентского приложения
7	Разработка клиентского приложения	14.03.2016 – 02.05.2016	Рабочая версия программы
8	Тестирование	03.05.2016 – 22.05.2016	Конечная версия программы с исправленными ошибками
9	Создание документации	23.05.2016 – 04.06.2016	Руководство пользователя, пояснительная записка

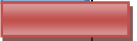
ПРИЛОЖЕНИЕ Ж

Таблица Ж.1– Временные показатели для построения диаграммы Ганта

Этап	Исполнители	Продолжительность работ, дни			Длительность работ, чел/дн.			
					Т _{рд}		Т _{кд}	
		t_{min}	t_{max}	$t_{ож}$	НР	И	НР	И
Разработка технического задания	НР	2	4	2,8	3,36	-	4,97	-
Согласование и утверждение технического задания	НР, И	1	2	1,4	1,44	1,44	2,13	2,13
Выбор технологии для разработки ИС	И	6	8	6,8	-	8,16	-	12,08
Исследование предметной области	И	4	7	5,2	-	6,24	-	9,24
Календарное планирование работ по теме	НР, И	1	2	1,4	1,44	1,44	2,13	2,13
Проектирование клиентского приложения	НР, И	7	14	9,8	3,924	7,836	5,81	11,6
Разработка клиентского приложения	И	35	49	43,4	-	52,08	-	77,08
Тестирование	НР, И	14	19	17	6,804	13,596	10,07	20,12
Создание документации	НР, И	10	12	11,2	4,476	8,964	6,62	13,27
Итого:		80	117	99	21,44	99,76	31,73	147,65

ПРИЛОЖЕНИЕ И

Таблица И.1 – Календарный план-график проведения НИОКР

№	Ткд НР	Ткд И	Февраль			Март			Апрель			Май			Июнь	
			1	2	3	1	2	3	1	2	3	1	2	3	1	2
1	4,97	-														
2	2,13	2,13	 													
3	-	12,08														
4	-	9,24														
5	2,13	2,13			 											
6	5,81	11,6				 										
7	-	77,08														
8	10,07	20,12										 				
9	6,62	13,27												 		

Примечание:  - И. 

ПРИЛОЖЕНИЕ К

Таблица К.1 – Реестр рисков

№	Риск	Потенциальное воздействие	Вероятность наступления (1-5)	Влияние риска (1-5)	Уровень риска	Способы смягчения	Условия наступления
1	Неправильно спроектированная БД	Неверная работа ИС	5	5	Высокий	Привлечение специалистов, задействованных в работе по созданию отчётов	Самостоятельное проектирование БД без привлечения специалистов
2	Отсутствие средств для приобретения программных инструментов разработки ИС	Низкое качество ИС	4	4	Высокий	Использование версий инструментов с ограниченными функциями	Отсутствие финансовой поддержки со стороны Заказчика
3	Неправильно спроектированный пользовательский интерфейс	Неверная работа ИС, неудобный интерфейс	4	4	Высокий	Демонстрация набросков эскизов пользовательского интерфейса Заказчику на протяжении всего проекта	Отсутствие согласования эскизов интерфейса с Заказчиком
4	Отсутствие обратной связи с Заказчиком	Неверно разработанная ИС	3	3	Средний	Регулярные встречи, обсуждения	Несогласованность технического задания, непредвиденные обстоятельства со стороны Заказчика

ПРИЛОЖЕНИЕ Л

Таблица Л.1 – План денежных потоков

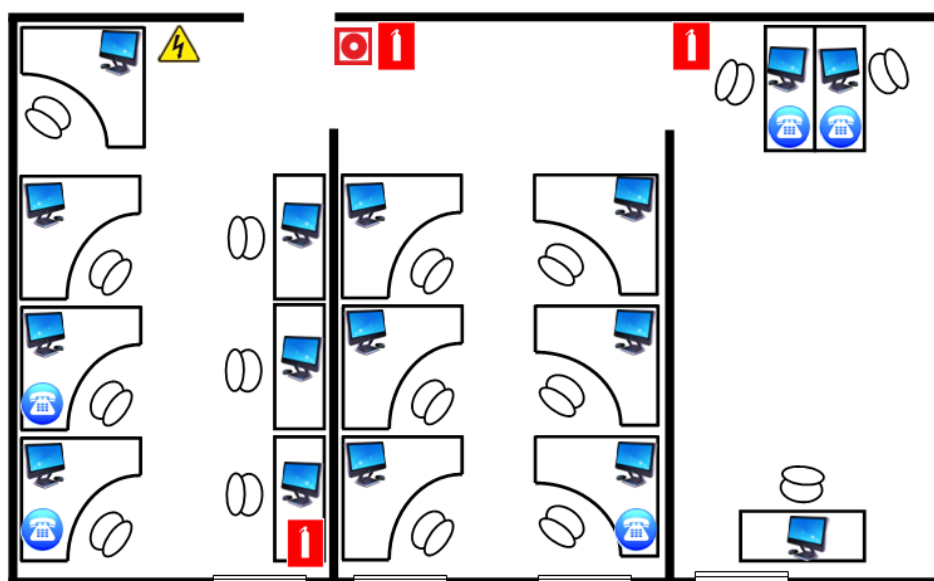
№	Показатель	Номер шага (периода) расчёта t			
		0	1	2	3
Операционная деятельность					
1	Выручка без НДС	0,0	272000	272000	136000
2	Полные текущие издержки, в том числе (сумма пунктов 3-7):	0,0	- 177933,3	- 177933,3	- 93133,3
3	ФОТ основных рабочих, включая взносы во внебюджетные фонды	0,0	7000*8 = - 56000	7000*8 = - 56000	7000*4 = - 28000
4	Амортизационные отчисления	0,0	- 8333,3	- 8333,3	- 8333,3
5	Командировочные расходы	0,0	12000*8 = -96000	12000*8 = -96000	12000*4 = -48000
6	Прочие расходы	0,0	500*8 = - 4000	500*8 = - 4000	500*4 = - 2000
7	Затраты на подготовку лицензии (установочный диск, комплект документов, обучение пользователей)	0,0	1700*8 = - 13600	1700*8 = - 13600	1700*4 = - 6800
8	Прибыль до налогообложения	0,0	94066,7	94066,7	42866,7
10	Налог на прибыль (УСН: 15% с разницы выручки и затрат)	0,0	14110,005	14110,005	6430,005
11	Чистая прибыль (Выручка – издержки –все налоги)	0,0	79956,695	79956,695	36436,695
12	Денежный поток от производственной деятельно-сти ДП (ОД) (п. 11 + п.4)	0,0	88289,995	88289,995	44769,995
Инвестиционная деятельность					
13	Поступление инвестиций	0,0	0,0	0,0	0,0
14	Капиталовложения, обслуживание инвестиций	- 179592,335	0,0	0,0	0,0
15	Сальдо от инвестиционной деятельности (п.13 + п.14)	- 179592,335	0,0	0,0	0,0
16	Сальдо суммарного потока (п.12 + п.15)	- 179592,335	88289,995	88289,995	44769,995
17	Коэффициент дисконтирования при ставке дохода 10%	1,0	0,909	0,826	0,751
18	Дисконтированное сальдо суммарного потока (п.16* п.17)	- 179592,335	80255,6	72927,5	33622,3
19	Накопленное дисконтированное сальдо	- 179592,335	- 99336,735	- 26409,235	7213,1

Примечания:

$ДП\ (ОД) = ЧП + |А|$, где ЧП – чистая прибыль и А – расходы на амортизацию.

$ЧП = (Выручка - З) * 0,15$, где В – выручка за период без НДС и З – сумма всех затрат (пункт 2 = п.3 + п.4 + п.5 + п.6 + п.7), 0.8 – налоги 15%.

ПРИЛОЖЕНИЕ М



-  – Огнетушитель
  – Электроцит
  – Телефон
 – Кнопка включения средств и систем пожарной автоматики
 – Компьютер

Рисунок М.1 – Схема помещения ГИС лаборатории

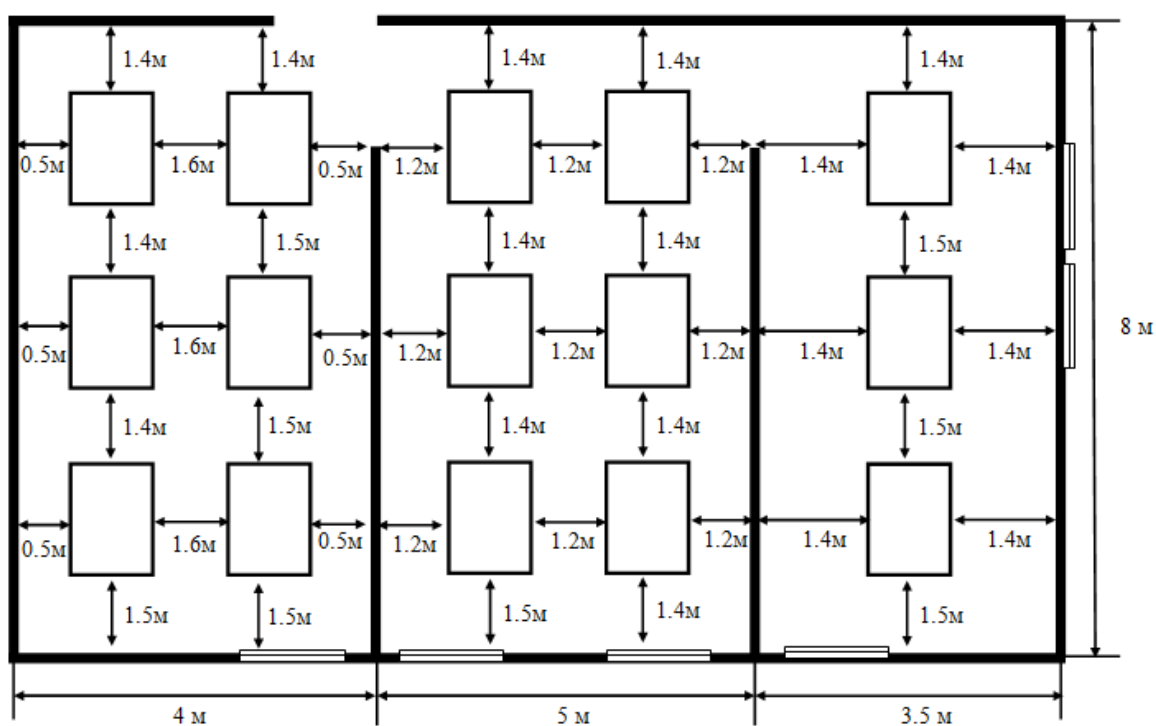


Рисунок М.2 – Схема размещения светильников в лаборатории НИ ТПУ

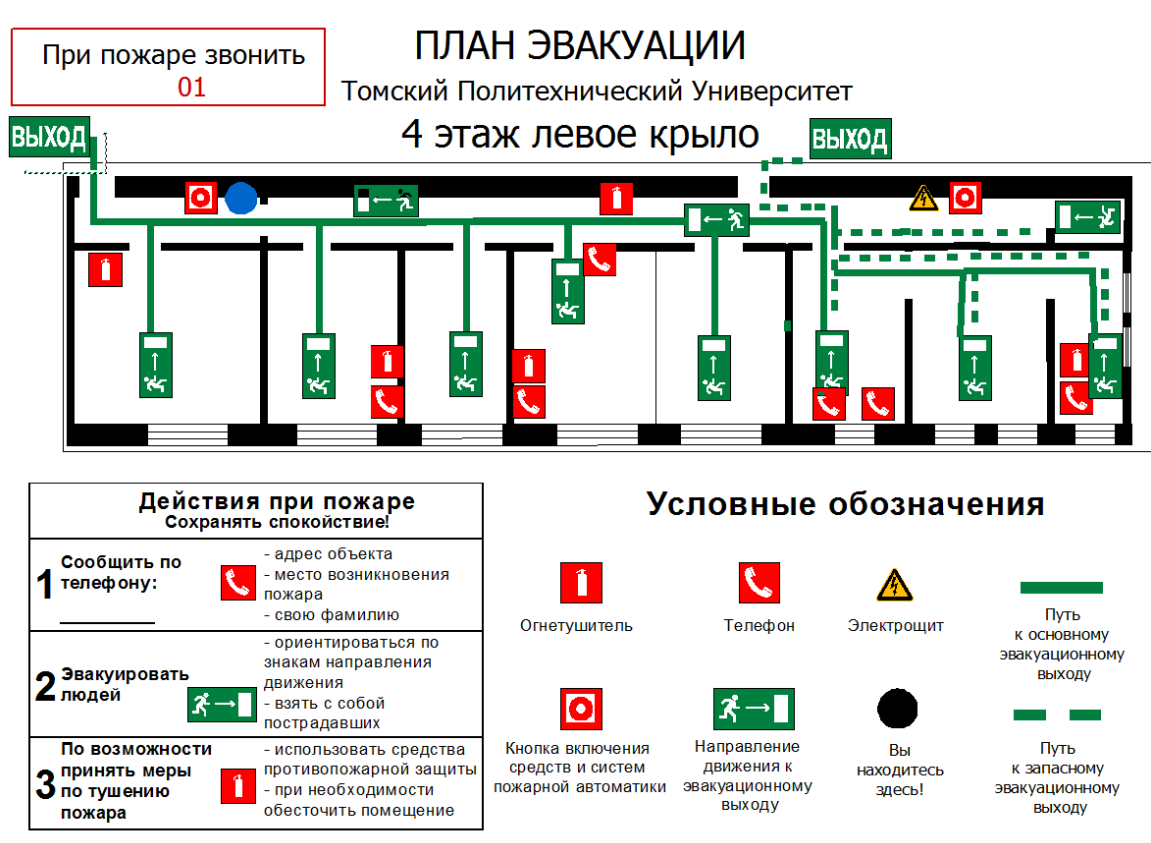


Рисунок М.3 – План эвакуации при пожаре