МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт **математики, физики и информационных технологий**

Кафедра **«Прикладная математика и информатика»**

09.04.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

ПРИКЛАДНАЯ ИНФОРМАТИКА В ОБРАЗОВАНИИ И ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЯХ

# МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему «РАЗРАБОТКА МОДЕЛЕЙ И МЕТОДОВ МОНИТОРИНГА ЗАПРОСОВ К ИНФОРМАЦИОННЫМ РЕСУРСАМ ПОРТАЛА КОРПОРАТИВНЫХ ЗНАНИЙ»

Студент(ка) А.Ю. Губин _____

Научный руководитель д.т.н., профессор, А.И. Туищев _____

Руководитель программы д.пед.н., профессор, А.Н. Ярыгин

«_____»_____20_____г.

**Допустить к защите**

Заведующий кафедрой к.тех.н., доцент, А.В. Очеповский

«_____»_____20_____г.

Тольятти 2016

# Содержание

# ВВЕДЕНИЕ

На сегодняшний день во многих сферах деятельности наблюдается переизбыток информации, что требует поиска новых механизмов ее структурирования и хранения в удобном для пользователя виде. Многие крупные компании для решения указанной проблемы разрабатывают порталы корпоративных знаний.

Портал корпоративных знаний – это интегрированный портал, предназначенный для обеспечения коллективной работы сотрудников в едином информационном пространстве внутри компании и аккумулирования знаний сотрудников в структурированном виде [37].

Портал знаний компании является примером практического применения теории управления знаниями для получения конкурентных преимуществ путем более эффективного использования корпоративных знаний. Главная цель портала знаний - создание способностей, поддерживающих то или иное стратегическое направление развития компании.

Несмотря на то, что информация на порталах корпоративных знаний является структурированной, ее не всегда можно найти по поисковым запросам. Кроме того, порталы не всегда содержат искомую информацию. Поэтому существует потребность в проведении исследования процессов мониторинга запросов к информационным ресурсам портала корпоративных знаний.

Вопросом мониторинга информационных ресурсов занимались ученые В.И. Аверченков, С.М. Рощин, С.В. Кузнецов, Б.Г. Левинский, О.Б. Сладкова, С.К. Дулин, Т.Я. Ашихмина, Б.А. Баллод, А.А. Белов, П.А. Цуканов, Л.И. Калакутский, Ю.И. Шокин, Э.С. Манелис и др.

Наиболее точно и полно мониторинг определен как специально организованное, систематическое наблюдение за состоянием объектов, явлений, процессов с целью их оценки, контроля или прогноза [8].

В данной работе рассматривается мониторинг запросов к информационным ресурсам. В связи с этим под мониторингом будем понимать

процесс сбора необходимой для организации статистической информации из различных источников, который позволяет судить об актуальности информационных ресурсов и их состоянии в рассматриваемый момент времени с целью оценки и контроля.

**Актуальность работы** заключается в том, что процессы поиска и использования информационных ресурсов на предприятии, занимающемся разработкой информационных систем, являются достаточно затратными по времени и количеству усилий для их обработки. Для улучшения качества работы сотрудников с информацией требуется применить инновационные методы мониторинга информационных ресурсов, которые позволили бы упростить поиск и использование необходимых данных. Инструментом для обеспечения мониторинга может стать автоматизированная система, собирающая статистику использования страниц портала корпоративных знаний, а также пользовательских обращений к поисковой системе.

**Целью** диссертационной работы является теоретическое обоснование и практическая реализация моделей и методов мониторинга запросов к информационным ресурсам портала корпоративных знаний.

**Объектом** исследования является портал корпоративных знаний.

**Предметом** исследования являются методы мониторинга запросов к информационным ресурсам портала корпоративных знаний.

**Гипотеза** диссертационного исследования состоит в том, что эффективность функционирования портала корпоративных знаний будет выше, если будет производиться регулярный мониторинг пользовательских запросов и актуализация информации, направленные на повышение качества информационных ресурсов.

Для достижения цели и проверки сформулированной гипотезы необходимо решить следующие **задачи**:

1. Изучить понятие корпоративного портала знаний и необходимость его применения.

2. Исследовать существующие системы мониторинга запросов к информационным ресурсам портала корпоративных знаний.

3. Изучить различные методы мониторинга запросов к информационным ресурсам портала корпоративных знаний.

4. Проанализировать основные особенности корпоративного портала знаний, используемого на предприятии.

5. Осуществить моделирование системы мониторинга запросов к информационным ресурсам.

6. Спроектировать и разработать систему мониторинга запросов к информационным ресурсам, опираясь на изученные исследования;

7. Доказать экспериментальным путем эффективность реализованной системы мониторинга и описать результаты её внедрения на предприятии.

**Научная новизна** исследования состоит в том, что была обоснована и реализована методика мониторинга запросов к информационным ресурсам портала корпоративных знаний, позволяющая поддерживать ресурсы в актуальном состоянии, что позволяет повысить эффективность работы сотрудников с базой корпоративных знаний.

**Значимость** диссертационного исследования заключается в том, что была разработана система мониторинга запросов и поиска информационных ресурсов, позволяющая повысить эффективность работы сотрудников с корпоративным порталом знаний.

Теоретической основой исследований послужили научные труды отечественных и зарубежных исследователей в области информатики, посвященные проблемам мониторинга запросов и использования информационных ресурсов.

В процессе исследования были использованы следующие практические положения и методы: теория поиска и классификации информации, анализ научной и методической литературы по теме исследования, сравнение существующих аналогов систем, систематизация накопленной информации,

моделирование и проектирование системы, проведение эксперимента, проведение апробации системы и дальнейшая оценка результатов.

**Основные этапы исследования:** исследование велось с 2014 по 2016 гг. в три этапа:

На **первом** – *констатирующем* этапе исследования (2014 г.) – формулировалась тема, осуществлялся сбор информации из источников научной и методической литературы, формулировалась гипотеза, осуществлялась постановка цели, задач, предмета и объекта исследования, определялась проблематика исследования и вклад современников в ее разрешение.

**Второй** этап (2014-2015 гг.) – *поисковый.* В ходе него осуществлялось концептуальное моделирование системы мониторинга запросов к информационным ресурсам портала корпоративных знаний, проводилось обоснование средств реализации системы, разрабатывалась технология мониторинга запросов и дальнейшей актуализации ресурсов, корректировались задачи исследования, осуществлялась теоретическая апробация исследования в ходе выступлений на конференциях.

**Третий** этап (2015-2016 гг.) - э*кспериментальная апробация*. Осуществлялось внедрение системы мониторинга запросов к информационным ресурсам, проводилась оценка результатов, проверялась достоверность определенной гипотезы, формулировались выводы по проведенному исследованию.

**На защиту выносится:**

1. Мониторинг запросов к информационным ресурсам как форма сбора, хранения и обработки информации о работе пользователей с корпоративным порталом знаний.

2. Методология и основные концепции информационного поиска как средства для повышения эффективности работы с информационными ресурсами.

3. Модель системы мониторинга запросов к информационным ресурсам, представляющая собой совокупность взаимосвязанных моделей, позволяющих изучить основные возможности и функции мониторинга запросов.

4. Результаты апробации разработанной информационной системы мониторинга запросов, демонстрирующие эффективность внедрения.

Система мониторинга запросов к информационным ресурсам портала корпоративных знаний была реализована и внедрена в отделе «Управления корпоративными знаниями компании» ООО «НетКрэкер». Результаты исследования были заслушаны на Международной научно-практической конференции «Новая наука: от идеи к результату» (29 октября 2015 года, г. Стерлитамак).

В первой главе рассматриваются базовые принципы использования информационных ресурсов портала корпоративных знаний, основные понятия и методы мониторинга; проводится сравнительный анализ существующих систем, занимающихся информационным мониторингом; рассматриваются основные понятия информационного поиска.

Во второй главе представлена характеристика предприятия, для которого разрабатывается система мониторинга. Осуществляется моделирование деятельности по проведению мониторинга запросов. Производится обоснование выбора архитектуры и средств реализации информационной системы. Проектируются диаграммы вариантов использования, классов, последовательности и деятельности в нотации UML, отражающие суть системы мониторинга запросов и представившие её как в статическом состоянии, так и в динамике.

В третьей главе приводится процесс проектирования базы данных разрабатываемой системы - переход от концептуальной модели данных к физической. Происходит выбор системы управления базами данных и демонстрируется реализация системы мониторинга запросов к информационным ресурсам портала корпоративных знаний.

Четвертая глава посвящена процессу апробации системы и оценке эффективности разработанной системы.

В заключении подводятся итоги выполненной работы.

Диссертация состоит из введения, четырех глав, заключения, списка литературы и приложений. Работа изложена на 94 страницах и содержит 30 рисунков.

# ГЛАВА 1 Теоретические основы организации мониторинга запросов к информационным ресурсам портала корпоративных знаний

## 1.1 Использование корпоративных порталов знаний с целью структурирования информации

В настоящее время в компаниях работает достаточно большое количество сотрудников. Каждый из них обладает определенным набором знаний, использующихся в работе. В случае увольнения сотрудника часть знаний, которыми он обладал, может исчезнуть из скопа знаний компании.

Знания могут быть раскрыты и формализованы в виде статей, патентов, технологий, технической документации и т.д., так они превращаются в информацию. Но невозможно формализовать весь интеллект и интуицию людей и коллективов. Поэтому невозможно и организовать хорошую информационную систему, которая бы не опиралась на интеллект людей и их коллективную работу.

Но, в тоже время, интеллект коллектива опирается не только на интеллект участников, но и на тот интеллект, который уже формализован и накоплен другими людьми и коллективами - на уже формализованную информацию.

С целью структурирования информации разрабатываются системы сбора информации и формализации новых знаний, опирающиеся на:

- весь накопленный и формализованный ранее потенциал знаний (на входную информацию);
- коллективный интеллект и интуицию задействованных людей (на совместную работу);
- вычислительные приложения, называемые порталами корпоративных знаний.

Портал знаний комбинирует в себе три типа порталов:

- информационный портал, который соединяет людей с информацией;

- портал для совместной работы, который обеспечивает все мыслимые средства взаимодействия людей с использованием компьютерных технологий;
- портал экспертизы, который соединяет людей с другими людьми на основе таких критериев как опыт, область экспертизы и интересы.

Главная цель портала знаний - создание способностей, поддерживающих то или иное стратегическое направление развития компании [37].

Порталы корпоративных знаний базируются на информационных порталах предприятия (EIP). Корпоративные информационные порталы (Enterprise Information Portal) – EIP – это приложения, которые позволяют компаниям раскрывать информацию, хранящуюся внутри и вне организации, и предоставить каждому пользователю единую точку доступа к предназначенной для него информации, необходимой для принятия обоснованных управленческих решений [20]. Изначально этот вид порталов представлял собой сайт с информацией о компании, на одной из страниц которого мог быть организован простой внутренний форум. На портале можно было получить информацию о последних организационных изменениях в компании, ознакомиться с новостями. Помимо того, портал часто выполнял функции простого файлового хранилища.

EIP – это хороший способ упорядочить доступ к различной информации по интрасетям и в Интернете, но он не решает всех потребностей пользователей в доступе к информации. Доступ пользователей к системе, как правило, открыт только «для чтения». Информация на информационном портале зачастую однотипная и возможности работы с ней ограничены. Еще одним минусом EIP считается то, что для перехода к конкретному ресурсу необходимо знать всю иерархию страниц, т.к. возможности поиска на них достаточно скудны и ограничены.

В результате развития корпоративные порталы стали выполнять функции баз знаний, так как появились инструменты для структурирования, классификации и поиска данных. Пользователи портала получили возможность

самостоятельно наполнять портал контентом и редактировать его, а также искать необходимую информацию. Все это привело к появлению корпоративных порталов управления знаниями (Enterprise Knowledge Portal – EKP). EKP не только предоставляет средства доступа к информации, но и позволяет пользователям взаимодействовать друг с другом, помогая связывать информацию с коллективным пониманием, системой ценностей и опытом. Он дает возможность принимать оптимальные решения, поскольку сочетает приобретенные знания с информацией и служит "самодокументирующимся" центром обучения на опыте. Также корпоративный портал знаний обеспечивает выполнение транзакций, в частности, в процессе управления знаниями. Чем больше барьеров на пути к обретению нужных знаний, тем больше времени уходит на действие. Когда препятствий слишком много, на нужный ответ и реакцию требуется очень много времени, что сотрудник начинает действовать наугад – такой подход дает отнюдь не оптимальное решение. Портал знаний должен снижать временные затраты, давая пользователям мощную, непрерывно улучшаемую карту пути к корпоративным знаниям и информации [15]. Кроме того, порталы знаний снижают время на подготовку новых сотрудников, и, как следствие, снижают финансовые затраты компании и повышают эффективность работы других сотрудников.

Несмотря на то, что портал знаний аккумулирует информацию, она имеет свойство устаревать со временем. Кроме того, отсутствие регулярного обновления статей часто приводит к исчезновению их из поисковой выдачи и, как следствие, к проблемам в процессе обмена знаний. В связи с этим возникает необходимость актуализации текста статей и их статуса в поисковой системе. Отсюда следует, что необходим мониторинг запросов к информационным ресурсам на портале, который позволит выявлять часто используемые страницы корпоративного портала и обновлять их. Кроме того, результатом мониторинга может стать модификация поисковой выдачи таким образом, чтобы более релевантные значения имели больший рейтинг и, как следствие, были выше на странице.

## 1.2 Основные понятия и методы мониторинга запросов к информационным ресурсам портала корпоративных знаний

Мониторинг — систематический сбор и обработка информации, которая может быть использована для улучшения процесса принятия решения [8]. Он применяется тогда, когда возникает необходимость отслеживания процесса реализации какого-либо плана, проекта, развития каких-либо событий, явлений. Получение своевременной информации о ходе протекания какого-либо процесса позволяет лучше понять его суть, а если возникают отклонения, оперативно внести коррективы. Таким образом, мониторинг дает возможность своевременно обновлять накопленные знания.

Основными видами мониторинга информационных ресурсов являются:

1. Динамический – в качестве основания для анализа служат данные о динамике развития того или иного объекта. Ключевую роль играет предупреждение о возможной опасности, а выяснение причин носит второстепенный характер.

2. Конкурентный – в качестве основания для экспертизы выбираются результаты такого же обследования подобных объектов.

3. Сравнительный – в качестве основания для экспертизы выбираются результаты идентичного обследования одной или двух систем более высокого уровня.

4. Комплексный - использует несколько оснований для экспертизы.

5. Базовый – выявление новых проблем и опасностей до того, как они станут критическими.

6. Проблемный - выяснение закономерностей, процессов и опасностей тех проблем, которые носят глобальный характер.

Классификация методов мониторинга запросов представлена ниже:

1. Метод контент-анализа состоит из поиска в тексте определенных содержательных понятий (единиц анализа), выявления частоты их появления и соотношения с содержанием всего документа. Общепризнанным является распределение методологии контент-анализа на две ветви: качественную и

количественную. Основа количественного контент-анализа - частота появления в документах определенных характеристик содержания. Метод качественного контент-анализа базируется на самом факте присутствия или отсутствия в тексте одной или нескольких характеристик содержания. Данный метод удобен для мониторинга отдельных информационных ресурсов портала, но мониторинг ресурсов всего портала займет достаточно большой период времени.

2. Контент-мониторинг представляет собой постоянное выполнение узко-очерченного своими задачами контент-анализа беспрерывных информационных потоков [27]. Отличительная особенность такой работы состоит, прежде всего, в обслуживании узкого круга потребителей со специфической сферой задач, которые требуют оперативного решения. Это, в свою очередь, требует четкой постановки информационно-аналитических задач и тесного контакта между заказчиками и службами поиска и анализа информации.

3. Аналитические методы – основаны на различных операциях со статистическими данными.

4. Экспертные методы – методы на базе обобщения информации и оценок, представленных экспертами.

5. Метод линейного программирования – математический прием, используемый для определения лучшей комбинации ресурсов и действий необходимых для достижения оптимальных результатов, развития исследуемого объекта.

Целью информационного мониторинга портала корпоративных знаний является определение наиболее популярных ресурсов и актуализация данных. Отсюда следует, что наиболее приемлемыми методами мониторинга информационных ресурсов портала корпоративных знаний станут аналитические методы, т.к. они позволяют осуществить мониторинг портала целиком, не разбивая на отдельные ресурсы, причем этот процесс может быть автоматизирован. Остальные вышеперечисленные методы усложняют процесс

мониторинга, требуя дополнительные ресурсы, или не позволяют достигнуть намеченной цели.

В основу аналитических методов мониторинга ложатся статистические данные. Информационные ресурсы обладают перечнем свойств, по которым можно собрать статистику. Таковыми являются: число просмотров ресурса, количество редактирований, количество оставленных комментариев, число поисковых запросов к ресурсу. Число просмотров и редактирований ресурса играет важную роль при определении популярности ресурса. На основе этих данных должны перестраиваться поисковые индексы, чтобы оптимизировать поисковую выдачу.

## 1.3 Анализ существующих систем мониторинга запросов к информационным ресурсам

Для осуществления сравнительной характеристики существующих систем мониторинга запросов необходимо выделить ключевые характеристики для их оценки. Ими являются:

1. Незаметность работы системы мониторинга для пользователя портала.
2. Возможность сбора данных по всем типам запросов.
3. Идентификация пользователя, осуществившего запрос.
4. Многообразие отчетных документов.
5. Хранение собранных данных на локальном сервере.
6. Возможность осуществления актуализации поисковых индексов
7. Низкая цена.

На сегодняшний день существует три основных системы мониторинга запросов. Ими являются Google Analytics, Яндекс.Метрика и Piwik.

**Google Analytics** является продолжением аналитической системы Urchin on Demand компании Urchin Software [49]. Google Analytics является крупным сервисом сбора статистики и анализа посещаемости веб-сайтов. Данная система имеет достаточно большие возможности – она не только отслеживает количество уникальных посещений, но и осуществляет мониторинг массы

других операций. С помощью Google Analytics можно контролировать практически все действия, которые выполняются пользователями на сайте. Например, среднее время пребывания посетителей на сайте, наиболее и наименее популярные среди пользователей веб-страницы и т.д.

По своему функционалу и возможностям работы Google Analytics является ограниченно бесплатной системой. Данное ограничение зависит от популярности сайта – бесплатная версия сервиса действует для веб-сайтов с посещаемостью до 5 миллионов уникальных просмотров в месяц. Подключение Google Analytics достаточно простое – необходимо лишь установить на сайте счетчик с JavaScript кодом, а обработка информации и составление отчетов будет происходить автоматически на сайте системы. Информацию о посещаемости может отслеживаться как на панели инструментов в системе, так и в виде детализированных отчетов – всего в системе доступно около 80 видов различных настраиваемых отчетов.

В эти отчеты могут входить совершенно различные данные. Например, это могут быть:

- ссылки и ключевые слова, по которым на сайт приходят посетители,
- ссылки, по которым посетители уходят с сайта;
- среднее время пребывания пользователей на сайте в целом и на определенной странице в частности.

Также отдельным списком идет полный анализ посетителей сайта по различным критериям (их браузеры, географическое местоположение и т.д.). Также одним из плюсов данной системы является то, что ее можно интегрировать с корпоративными порталами знаний, а не только с обычными web-сайтами. Но особенностью данной системы является то, что все собранные данные хранятся на серверах Google, а клиент получает только результаты мониторинга в виде графиков и отчетов. Это может остановить компанию, которая дорожит конфиденциальностью своих данных, от внедрения этой системы мониторинга.

**Яндекс. Метрика** представляет собой бесплатный сервис сбора статистики посещений веб-сайтов. Он появился в свободном доступе не очень давно, в 2009 году, однако, за это время данный сервис сумел обрести высокую популярность среди вебмастеров. По своей сути, Яндекс. Метрика является счетчиком посещений, который устанавливается на сайт с помощью специального кода.

С помощью Яндекс. Метрики можно не только оценивать посещаемость сайта в целом, но и также получать данные о том, какие его веб-страницы пользуются большей популярностью среди посетителей, а какие – меньшей [49].

Важная особенность системы Яндекс. Метрика – высокая оперативность предоставляемых сведений о посещаемости сайта. Отчеты составляются без опозданий и обновляются раз в 5 минут. Также в системе существует такой полезный инструмент, как «Конструктор отчетов» – он позволяет получать детальные сведения по посещаемости. Имеется возможность сортировать посетителей сайта по таким параметрам, как их пол, возраст, а также географическое местоположение. Помимо этого, в системе Яндекс. Метрика есть функция фильтрации. С ее помощью можно устанавливать специальные фильтры, которые будут сортировать все поступающие в отчет исходные данные (такие, как URL веб-страницы, IP-адрес посетителей и т.д.)

**Piwik** — это бесплатная система веб-аналитики с открытым исходным кодом [42]. Piwik устанавливается на веб-сервер как обычная CMS. Все аналитические данные о посетителях сайта, которые собирает и структурирует Piwik, принадлежат только владельцу системы (веб-мастеру или компании, которые используют Piwik). Никакие третьи лица, будь то изначальные разработчики системы или крупные корпорации, не имеют никакого доступа к аналитике пользователей Piwik.

Piwik бесплатен для скачивания, использования, модификации и распространения, если это не нарушает требований лицензии GNU GPLv3. Открытый исходный код позволяет веб-мастерам и компаниям изменять Piwik

таким образом, чтобы он больше соответствовал их нуждам и представлениям о веб-аналитике [43]. Получаемая статистика обновляется в режиме реального времени (каждые 10 сек. по умолчанию). Количество сайтов, отслеживаемое в рамках одной системы, неограниченно. Число пользователей системы, имеющих различный доступ к статистике по сайтам, неограниченно.

Архитектура системы основана на плагинах. Можно создавать свои плагины и отключать ненужные. Система предоставляет возможности для конфиденциальности личных данных посетителей (анонимизация IP, очистка старых логов, отписка от учета в системе для посетителей).

На основе вышеперечисленного можно сформировать таблицу соответствия оценочным характеристикам (таблица 1.1).

Таблица 1.1 – Сравнительная характеристика аналогов

| Критерии | Google Analytics | Яндекс. Метрика | Piwik |
|---|---|---|---|
| Незаметность работы системы мониторинга для пользователя портала | + | + | + |
| Возможность сбора данных по всем типам запросов | + | - | + |
| Идентификация пользователя, осуществившего запрос | - (только IP адрес) | - (только IP адрес) | - (только IP адрес) |
| Многообразие отчетных документов | + | + | + |
| Хранение собранных данных на локальном сервере | - | - | + |
| Возможность осуществления актуализации поисковых индексов | - | - | - |
| Низкая цена | - | + | + |
| ИТОГО | 3 | 3 | 5 |

На основе анализа таблицы было выявлено, что существующие аналоги систем мониторинга запросов не обладают всеми требуемыми характеристиками, а также имеют плохую интеграцию с продуктом Atlassian Confluence. Отсюда, самым рациональным решением будет разработка нового модуля корпоративного портала - системы мониторинга запросов к информационным ресурсам, обладающей функциями сбора данных и прогнозирования, хранящей свои данные на локальном сервере и позволяющей влиять на результаты поисковой выдачи.

## 1.4 Информационно-поисковые системы как средство для актуализации информационных ресурсов портала корпоративных знаний

Пользователи корпоративного портала знаний вынуждены использовать поисковую систему, т.к. знать всю иерархию страниц крупного портала практически невозможно.

Информационно-поисковые системы появились на свет достаточно давно. Теории и практике построения таких систем посвящено довольно большое количество статей, основная масса которых приходится на конец 70-х - начало 80-х годов. Среди отечественных источников следует выделить научно-технический сборник "Научно-техническая информация. Информационные процессы и системы". На русском языке издана так же и "библия" по разработке этого рода систем - "Динамические библиотечно-информационные системы" Жерарда Солтона. Информационно-поисковые системы в сети Интернет - это признание того, что ни иерархическая модель Gopher, ни гипертекстовая модель World Wide Web не решают проблему поиска информации в больших объемах разнородных документов [33]. И на сегодняшний день нет другого способа быстрого поиска данных, кроме поиска по ключевым словам.

При использовании иерархической модели Gopher приходится довольно долго бродить по дереву каталогов, пока не встретишь нужную информацию. Эти каталоги должны кем-то поддерживаться и при этом их тематическое

разбиение должно совпадать с информационными потребностями пользователя. Учитывая огромное количество всевозможных интересов у пользователей Сети, понятно, что кому-то может и не повезти, и в сети не будет каталога, отражающего конкретную предметную область.

Документальным массивом ИПС в сети Интернет является все множество документов. Все это довольно разнородная информация, которая представлена в виде различных, никак несогласованных друг с другом форматов данных. Здесь есть и текстовая информация, и графическая информация, и аудио информация и вообще все, что есть в различных хранилищах. Естественно встает вопрос, как информационно-поисковая система должна со всем этим работать. В традиционных системах есть понятие поискового образа документа – ПОД [28]. ПОД (Поисковый Образ Документа) - это нечто, что заменяет собой документ и используется при поиске вместо реального документа. Поисковый образ является результатом применения некоторой модели информационного массива документов к реальному массиву. Наиболее популярной моделью является векторная модель, в которой каждому документу приписывается список терминов, наиболее адекватно отражающих его смысл. Если быть более точным, то документу приписывается вектор, размерность которого равна числу терминов, которыми можно воспользоваться при поиске. При булевой векторной модели элемент вектора равен 1 или 0, в зависимости от наличия термина в поисковом образе документа или его отсутствия. В более сложных моделях термины взвешиваются, т.е. элемент вектора равен не 1 или 0, а некоторому числу, которое отражает соответствие данного термина документу.

Таким образом, первая задача, которою должна решить информационно-поисковая система - это приписывание списка ключевых слов документу или информационному ресурсу. Именно эта процедура и называется индексированием. Часто, однако, индексированием называют составление файла инвертированного списка, в котором каждому термину индексирования ставится в соответствие список документов, в которых он встречается. Такая

процедура является только частным случаем, а точнее техническим аспектом создания поискового аппарата информационно-поисковой системы.

После того, как ресурсы проиндексированы, т.е. система составила массив поисковых образов документов, начинается построение поискового аппарата системы. Совершенно очевидно, что лобовой просмотр файла или файлов поискового образа займет много времени, что абсолютно не приемлемо для интерактивной системы, которой является Web. Для того, чтобы можно было быстро находить информацию в базе данных ПОД строится индекс. Индекс в большинстве систем - система связанных между собой файлов, которая нацелена на быстрый поиск данных по запросу пользователя. Структура и состав индексов различных систем могут отличаться друг от друга и зависят от многих факторов. К этим факторам можно отнести и размер массива поисковых образов, и информационно-поисковый язык системы.

При рассмотрении средств поиска нельзя пройти мимо процедуры коррекции запросов по релевантности. Релевантность - это мера соответствия найденного системой документа потребности пользователя [22]. Различают формальную релевантность и реальную. Формальная - это та, что вычисляет система и на основании чего ранжируется выборка найденных документов. Реальная - это та, как сам пользователь оценивает найденные документы. Некоторые системы имеют для этого специальное поле, где пользователь может отметить документ как релевантный. При следующей поисковой итерации запрос расширяется терминами этого документа. И выдача снова ранжируется. Так происходит до тех пор, пока результат не стабилизируется. Это означает, что ничего лучше, чем полученная выборка, от данной системы не добьешься.

Наличие и использование информационно-поисковых систем является одним из ключевых аспектов портала корпоративных знаний. При помощи них осуществляется навигация сотрудников между информационными ресурсами. Стоит отметить, что с развитием корпоративных порталов управления знаниями развивались и используемые на них поисковые системы. Самым первым появился классический полнотекстовый поиск, который позволял

искать лишь те ресурсы, заголовки которых полностью соответствуют запрошенной строке. Следующим этапом развития поисковых систем стало появление нечеткого поиска. Данный вид поиска зачастую использует метрики расстояний Левенштейна или Дамерау-Левенштейна. Оба этих расстояния описывают разницу между двумя строками символов. Данная мера рассчитывается как число операций, необходимых для перевода одной строки в другую. Использование нечеткого поиска предоставило более широкие возможности при поиске конкретного информационного ресурса. В поисковой выдаче начали появляться статьи с однокоренными словами. Дальнейшим шагом в развитии ИПС на порталах корпоративных знаний стала интеграция их с функционалом ЕКР. Таким образом, была добавлена возможность выбора разделов портала знаний для поиска, поиск по автору, дате создания или редактирования, использование ключевых слов информационного ресурса при поиске, выбор типа информационного ресурса и т.д. Последним внедренным шагом стало появление собственного языка поиска, что позволило пользователям самостоятельно формировать поисковую строку и добавлять в нее все необходимые критерии.

Современные информационно-поисковые системы позволяют расширять свой функционал, и, как следствие влиять на релевантность поисковых результатов. Вариантами расширения является модификация расчета поискового индекса по уже существующим полям документа или создание собственных уникальных полей. В совокупности с данными, полученными при мониторинге, можно модифицировать работу поисковой машины таким образом, чтобы результаты поиска были наиболее близки к искомому тексту, и косвенно актуализировать состояние информационных ресурсов.

В системе Atlassian Confluence используется поисковый движок Apache Lucene. Расчет поисковых индексов в данной системе строго неизменяем и модификация коэффициентов уже существующих значений достаточно сложна. Поэтому единственно возможный способ актуализации ресурсов – это создание

новых полей у каждого информационного ресурса на основе данных мониторинга.

## Вывод по главе

Проведенный анализ трудов в сфере информационного мониторинга позволил определить особенности работы с корпоративными порталами знаний; выявить методы, используемые при мониторинге, а также установить ключевые недостатки существующих систем. Помимо того, были выявлены возможности корпоративных порталов знаний по автоматизированной актуализации ресурсов. Все это в совокупности привело к формированию требований для системы мониторинга запросов к информационным ресурсам.

# ГЛАВА 2 Моделирование системы мониторинга запросов к информационным ресурсам портала корпоративных знаний

## 2.1 Характеристика деятельности предприятия по осуществлению запросов к информационным ресурсам портала корпоративных знаний

Порталы корпоративных знаний часто реализуются на базе вики-систем. Вики – веб-сайт, структуру и содержимое которого пользователи могут самостоятельно изменять с помощью инструментов, предоставляемых самим сайтом. Форматирование текста и вставка различных объектов в текст производятся с использованием вики-разметки. В компании Netcracker, где планируется внедрение разрабатываемой системы, существует портал B.A.S.S. «Business Analysis Support System» - это общая корпоративная wiki - система, разработанная для обмена знаниями и опытом, которая была основана в 2006 году. BASS реализован на базе одной из наиболее удобных платформ – Atlassian Confluence. Она сочетает в себе возможности создания онлайн-документов с помощью wiki-разметки и rich-text компонентов, а также интегрирована с Microsoft Office и Adobe Acrobat. Кроме того, поддерживается интеграция с Microsoft SharePoint, что позволяет полностью объединить информационное пространство организации. Confluence обладает открытым API, что не только обеспечивает наличие большого числа систем, расширяющих базовый функционал, но и позволяет разрабатывать их самостоятельно.

Основная задача портала корпоративных знаний в Netcracker – воплощение базы для сохранения бесценного опыта сотрудников и управления знанием. Управление знаниями — это механизм предоставления информации людям в удобном им виде для помощи в осуществлении их решений.

Цель системы - помочь каждому из сотрудников найти ответ на возникший в работе вопрос и возможность поделиться опытом. Только как

результат совместных усилий система обретёт ту базу знаний, которая станет ценностью не только компании, но и каждого из нас.

Существует несколько способов организации контента на BASS.

*Разделы* позволяют группировать контент по категориям. Каждый раздел представляет собой область внутри BASS, содержащую вики-страницы. Фактически, это сайт внутри другого сайта со своей домашней страницей.

*Страницы* являются основными средствами хранения информации на BASS, на их основе формируются разделы. Страницы используются для отдельных тем и структурируются в дерево страниц. *Дочерними страницами* называются вложенные страницы, созданные на родительских страницах. *Метки* – тэги, задаваемые пользователем для удобства навигации по страницам. Метки являются одним из лучших способов поиска информации по сходной тематике.

Во *Вложениях* размещаются изображения, мультимедиа, документы MS Office, PDF файлы и другие виды контента.

Пользователи могут добавлять записи в *блог* в любом разделе на BASS при наличии соответствующих прав доступа пользователя. Эти записи могут представлять собой уведомления, записи в журнале и прочую периодически обновляемую информацию. Пользователь имеет возможность просматривать записи в блогах пространства, выбрав секцию блогов.

Пользователи также могут добавлять комментарии − заметки, вопросы или любую другую информацию по теме, затронутой на странице. Любую доступную страницу или новость на BASS можно комментировать. Комментарии являются обязательным параметром портала управления знаниями, ведь при обсуждении конкретного информационного ресурса можно сформировать знание, которое будет полезно в дальнейшей деятельности компании.

В настоящее время мониторинг системы осуществляется только стандартными средствами платформы Confluence, которые позволяют просмотреть лишь число просмотров и редактирований разделов, не имея

дополнительной и развернутой информации об отдельных страницах, что является недостаточным объемом и не позволит выделить и актуализировать ресурсы. Кроме того, собираемые данные не содержат информации об использовании ресурсов портала конкретными пользователями. Предполагаемое решение данной проблемы заключается в сборе полной статистической информации о посещениях страниц и их редактированиях – в том числе подробной информации о пользователе и дате совершенного действия. Кроме того, будет полезно собирать информацию о поисковых запросах пользователей и переходах на определенные страницы. Это позволило бы администраторам портала выявлять потребности пользователей, и в случае отсутствия необходимого контента, дополнять им портал корпоративных знаний. Таким образом, были определены потенциальные возможности для расширения функционала мониторинга запросов к информационным ресурсам портала корпоративных знаний.

## 2.2 Концептуальное моделирование деятельности предприятия по осуществлению мониторинга запросов к информационным ресурсам

Для того чтобы выделить основные процессы, производимые при мониторинге, необходимо построить диаграмму IDEF0 TO-BE. Основной задачей мониторинга запросов можно считать повышение качества информационных ресурсов портала корпоративных знаний. Для ее решения необходимо выполнение различных подпроцессов, таких как сбор данных, формирование статистики и актуализация ресурсов. На основании этого можно определить элемент A0 диаграммы как «Осуществить мониторинг запросов к информационным ресурсам». Далее необходимо выделить входные и выходные параметры, механизмы и управление.

В качестве входного параметра выступают «Запросы пользователей». Выходными данными будут: «График», «Актуальные ресурсы на текущий момент», «Ресурсы, требующие актуализации» и «Отчет». Мониторинг запросов будет осуществляться администраторами портала с использованием

средств самого портала. Мониторинг запросов будет осуществляться на основании технической документации.

1. Построим контекстную диаграмму системы.

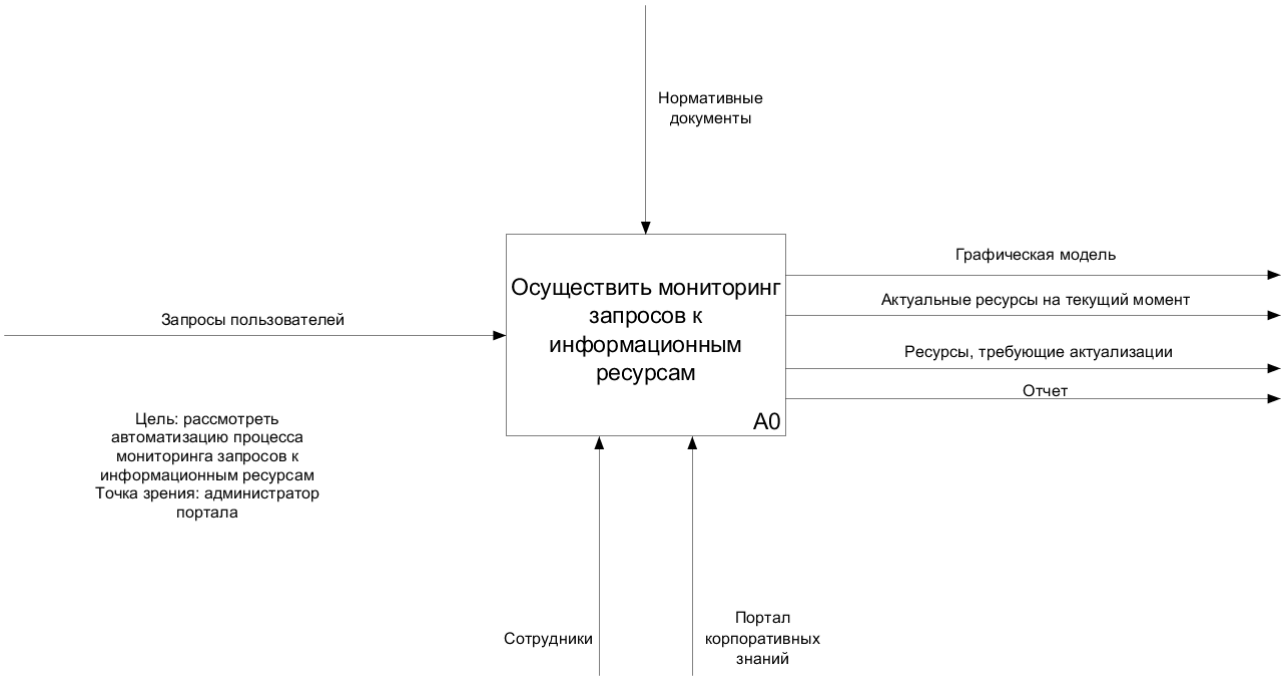


Рисунок 2.1 – Контекстная диаграмма TO-BE

2. Произведем декомпозицию контекстной диаграммы для определения основных процессов, происходящих при мониторинге запросов к информационным ресурсам.

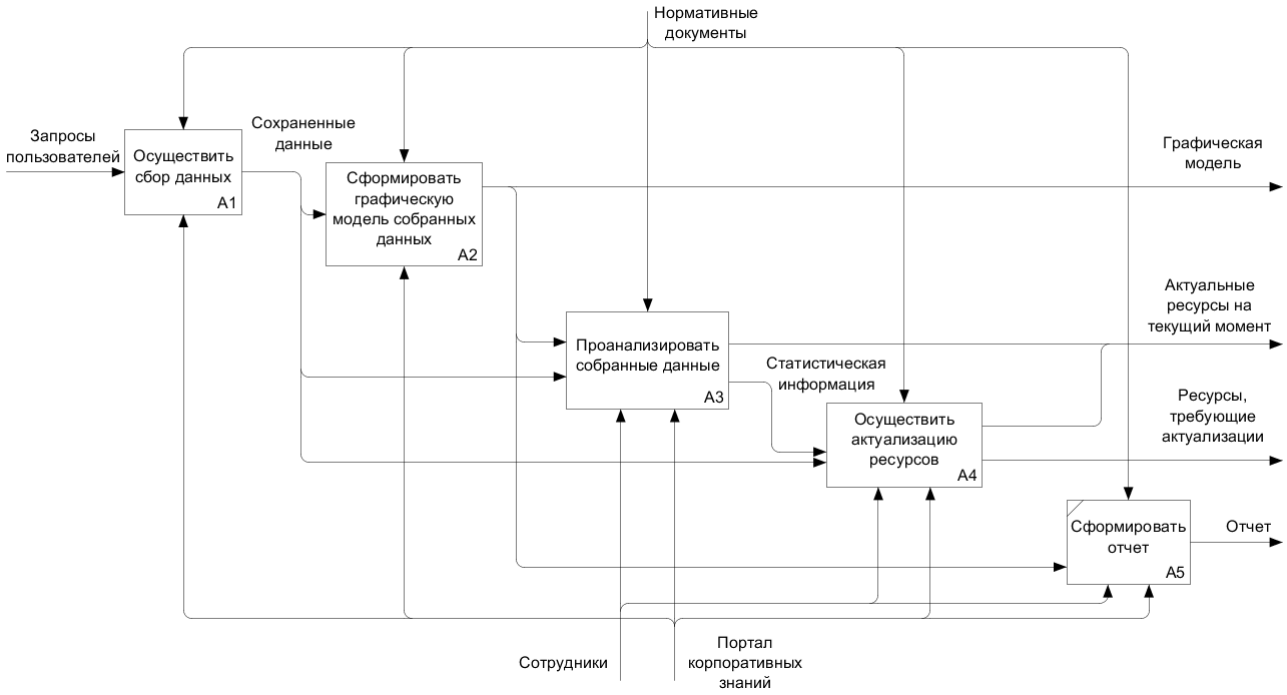


Рисунок 2.2 – Декомпозиция контекстной диаграммы TO-BE

На рисунке 2.2 представлена декомпозиция контекстной диаграммы, где видно, что процесс «Осуществить мониторинг запросов к информационным ресурсам» состоит из следующих процессов: «Осуществить сбор данных», «Сформировать графическую модель собранных данных», «Проанализировать собранные данные», «Осуществить актуализацию ресурсов», а также «Сформировать отчет». На диаграмме видно, что все параметры, представленные на контекстной диаграмме, отображены и на ее декомпозиции. Мониторинг запросов начинается с процесса «Осуществить сбор данных», входным параметром которого являются «Запросы пользователей». Результатом процесса станут собранные данные, процесс будет осуществляться полностью автоматически без вмешательства сотрудников организации. Следующим процессом является «Сформировать графическую модель данных. На основании данных, полученных из предыдущего процесса, формируется модель использования информационных ресурсов, на которой указаны наиболее популярные страницы. Процесс формирования графика также осуществляется без вмешательства сотрудников. На основе полученного графика и собранных данных из процесса А1 выполняется процесс «Проанализировать собранные данные». Выходными данными этого процесса являются «Актуальные ресурсы на текущий момент» и «Статистическая информация». Процесс осуществляется администраторами портала корпоративных знаний с применением средств портала. Далее, на основе статистической информации осуществляется автоматизированная актуализация, позволяющая изменить поисковую выдачу и косвенно актуализировать ресурсы. На основании всех полученных данных выполняется процесс «Сформировать отчет».

3. Для более глубокого анализа декомпозируем последовательно все блоки диаграммы.

Декомпозиция процесса «Осуществить сбор данных» представлена на рисунке 2.3. Процесс сбора данных полностью передан под контроль средств портала корпоративных знаний. При каждом запросе пользователя к

информационному ресурсу система мониторинга будет определять тип запроса, обрабатывать его определенным способом, в зависимости от типа и записывать данные.

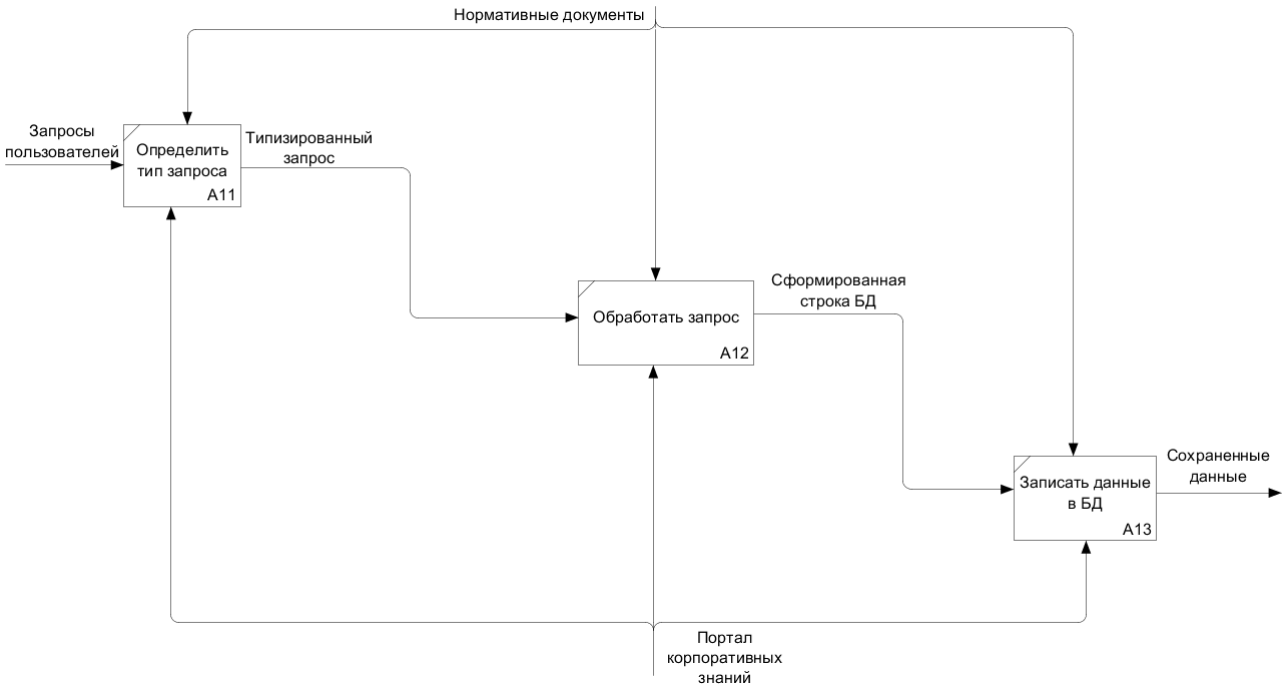


Рисунок 2.3 – Декомпозиция процесса «Осуществить сбор данных»

Процесс «Сформировать графическую модель собранных данных» декомпозируется на три процесса. Первым является группировка собранных данных по типу. В зависимости от этого будет происходить их дальнейшая обработка для построения модели. Последним процессом в этой декомпозиции станет построение модели. Она будет строиться с помощью средств портала корпоративных знаний на основе обработанных данных. Результатом данного процесса станет построенные графические модели, на основе которых администратор портала знаний сможет провести полноценный анализ использования ресурсов. В работе процесса формирования графической модели пользователь также не принимает участия. Подробная декомпозиция представлена на рисунке 2.4.

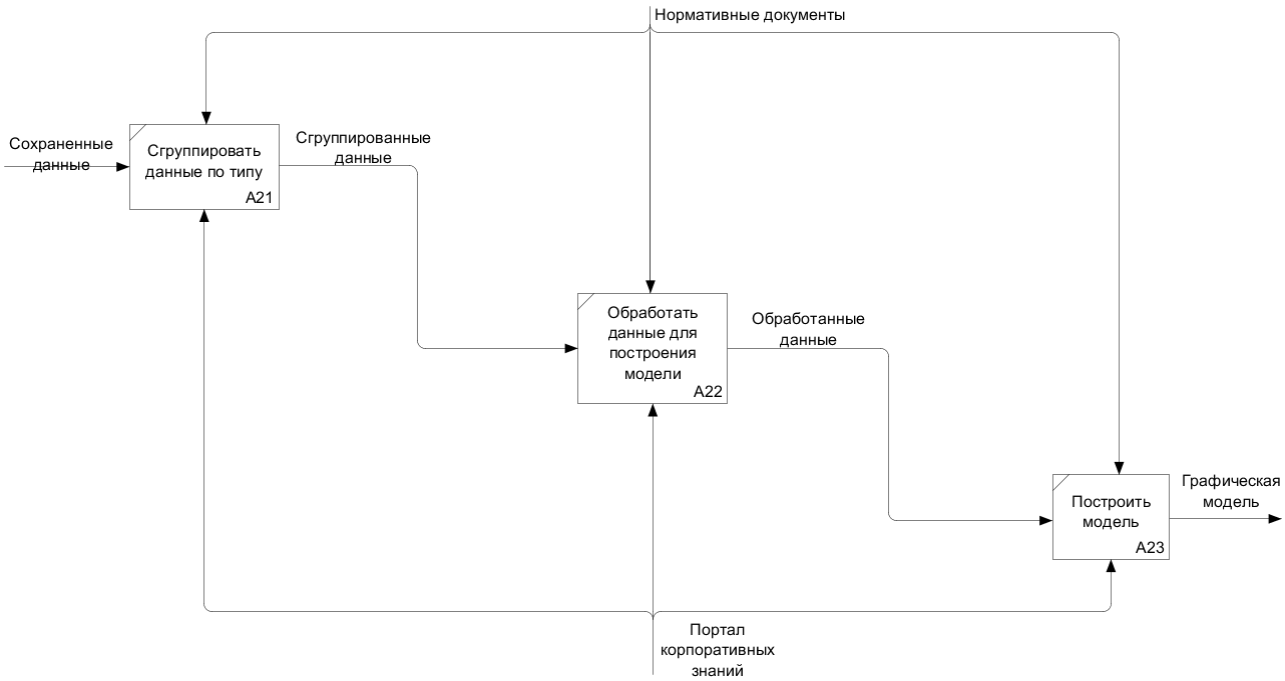


Рисунок 2.4 – Декомпозиция процесса «Сформировать графическую модель собранных данных»

На этапе анализа собранных данных необходимо определить ключевые характеристики для анализа. На их основе осуществляется анализ данных и формируется статистическая информация, которая в свою очередь используется для выделения актуальных ресурсов.

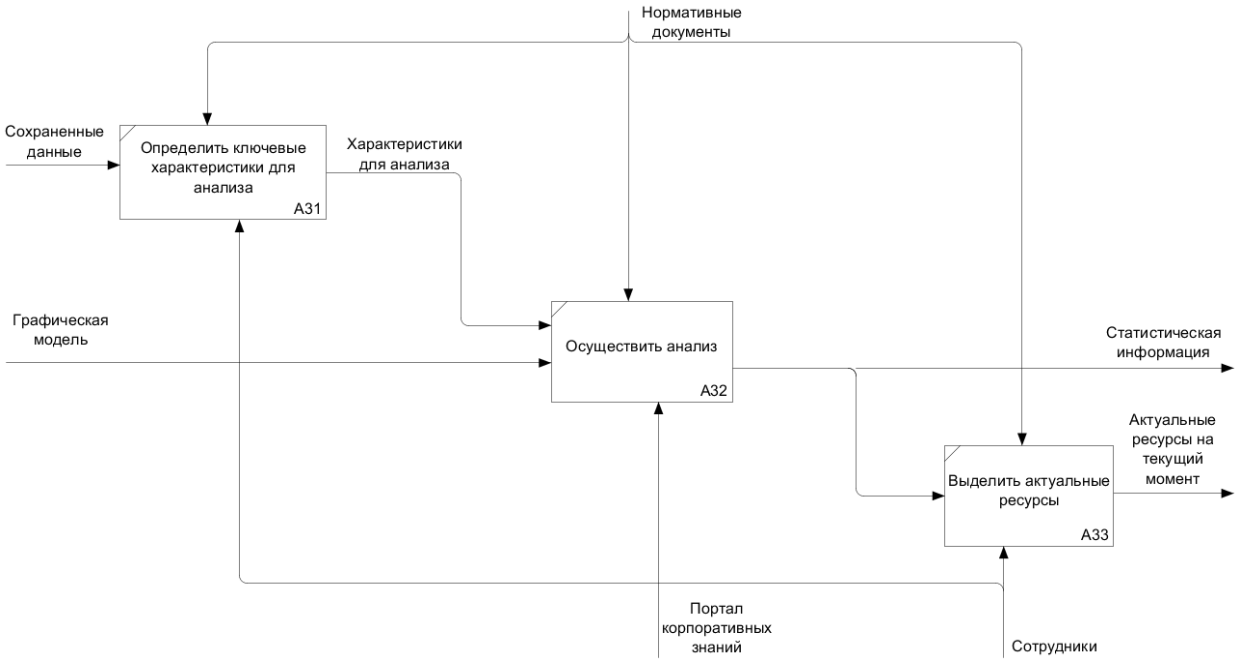


Рисунок 2.5 – Декомпозиция процесса «Проанализировать собранные данные»

Результирующими данными этих процессов являются «Статистическая информация», в дальнейшем используемая для проведения автоматизированной актуализации ресурсов, а также «Актуальные ресурсы», которые становятся результатом работы системы.

На следующем этапе при осуществлении актуализации, необходимо сформировать запрос для запуска этого процесса. Далее, на основе статистической информации и запроса непосредственно происходит процесс актуализации ресурсов. Результатом этого процесса становится список актуальных ресурсов. Список популярных ресурсов и данные о поисковых запросах пользователей являются входящими для процесса анализа статистических данных после процесса актуализации. Он позволит выявить ресурсы, требующие ручного обновления администраторами портала. На основании вышеперечисленных данных формируется отчет по неактуальным ресурсам.

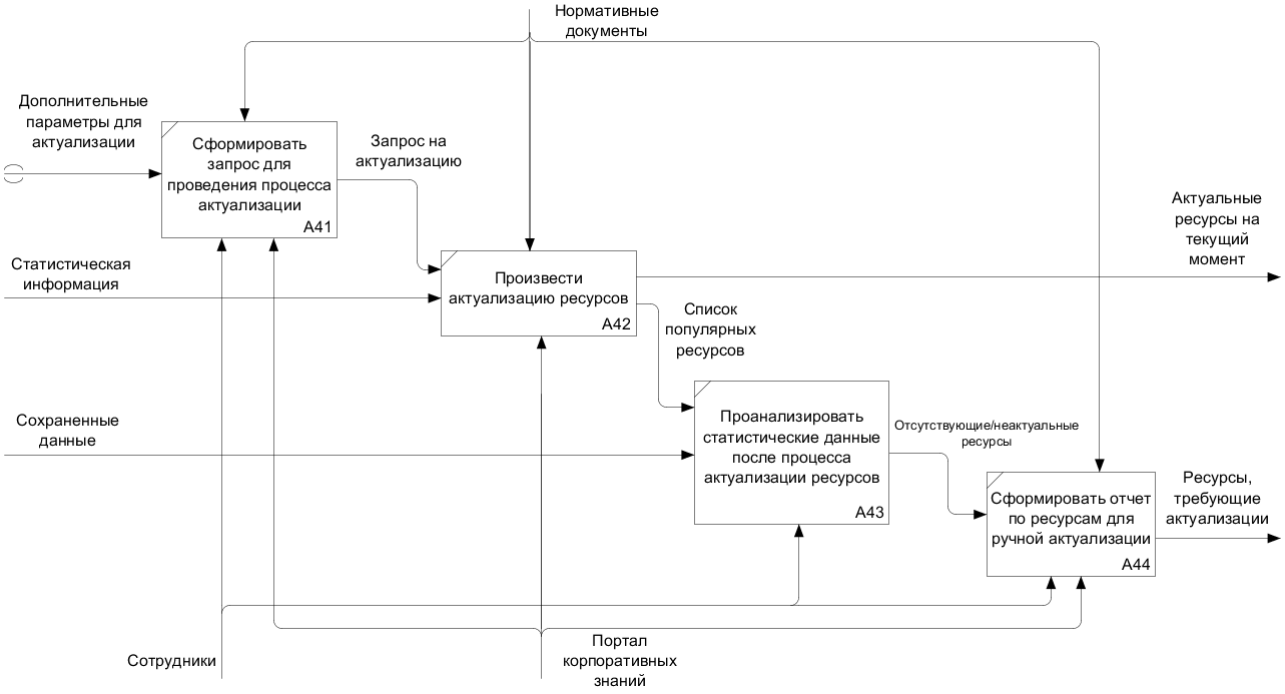


Рисунок 2.6 – Декомпозиция процесса «Осуществить актуализацию ресурсов»

На диаграммах нотации IDEF0 невозможно проследить движение потоков данных, вследствие чего отсутствует возможность полностью проанализировать процессы мониторинга с использованием автоматизируемых

функций. Поэтому представим процесс «Осуществить мониторинг запросов к информационным ресурсам» в нотации DFD.

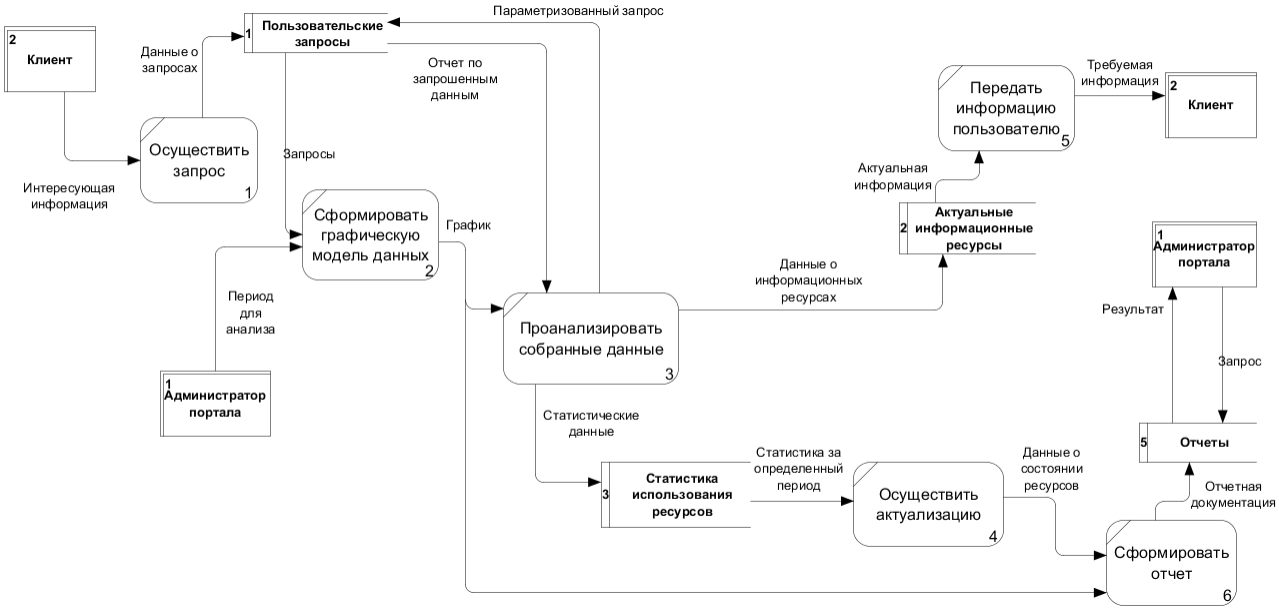


Рисунок 2.7 – Диаграмма потоков данных предметной области

На диаграмме видно, что при мониторинге запросов должны использоваться различные хранилища данных, такие как «Пользовательские запросы», которое используется для формирования графической модели и статистических данных, на основе хранящихся запросов. Хранилище «Статистика использования ресурсов» используется для дальнейшей автоматизированной актуализации поисковой выдачи согласно запросам пользователей. Для осуществления актуализации происходит выборка за определенный период. Кроме того, необходимо хранилище данных по актуальным ресурсам на текущий момент, а также тем ресурсам, которые необходимо актуализировать.

В результате построения диаграмм TO-BE и DFD были выявлены основные требования к системе мониторинга запросов к информационным ресурсам. Она должна собирать пользовательские запросы к информационным ресурсам, такие как запросы на поиск, просмотр, редактирование данных. В зависимости от типа запроса должна собираться дополнительная информация о нем. Так, например, при поисковом запросе должны собираться данные о теле запроса, его моменте, результате (о том ресурсе, на который перешел

пользователь), информация о пользователе. На основании собранных данных должна формироваться статистическая информация в виде отчетов и графических моделей, позволяющая администратору портала произвести анализ использования ресурсов.

## 2.3 Концептуальная модель системы мониторинга запросов к информационным ресурсам портала корпоративных знаний

Для анализа основных функций системы построим диаграмму потоков данных.

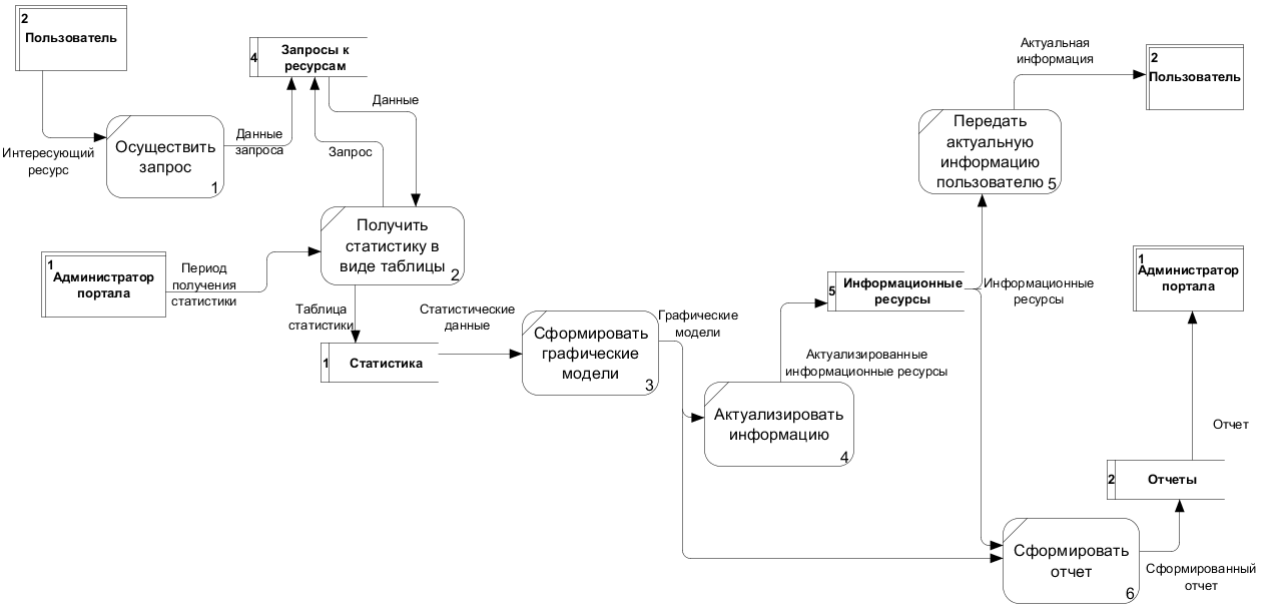


Рисунок 2.8 – Диаграмма потоков данных системы

На рисунке 2.8 отображены основные процессы, реализуемые системой с точки зрения администратора портала. Первым процессом является «Осуществить запрос». В результате осуществления запроса пользователем должен сработать обработчик событий, который определит тип запроса и запишет данные о нем в хранилище данных. Далее, на основе пользовательских запросов администратор портала получает статистическую информацию по пользовательским запросам за определенный период. На основе полученной статистики осуществляются процессы формирования графической модели и актуализации ресурсов. Результатами этих процессов, соответственно,

являются графическая модель и актуализированные ресурсы. На основании всех полученных данных формируется отчет по работе системы мониторинга.

При дальнейшем проектировании системы будет удобнее использовать объектно-ориентированный подход, в связи с тем, что он позволяет разбить систему на совокупность независимых сущностей - объектов и провести их строгую независимую спецификацию [23]. Используя диаграмму вариантов использования UML, определим основные функции и пользователей системы мониторинга. Основой для данной диаграммы будет диаграмма потоков данных системы.

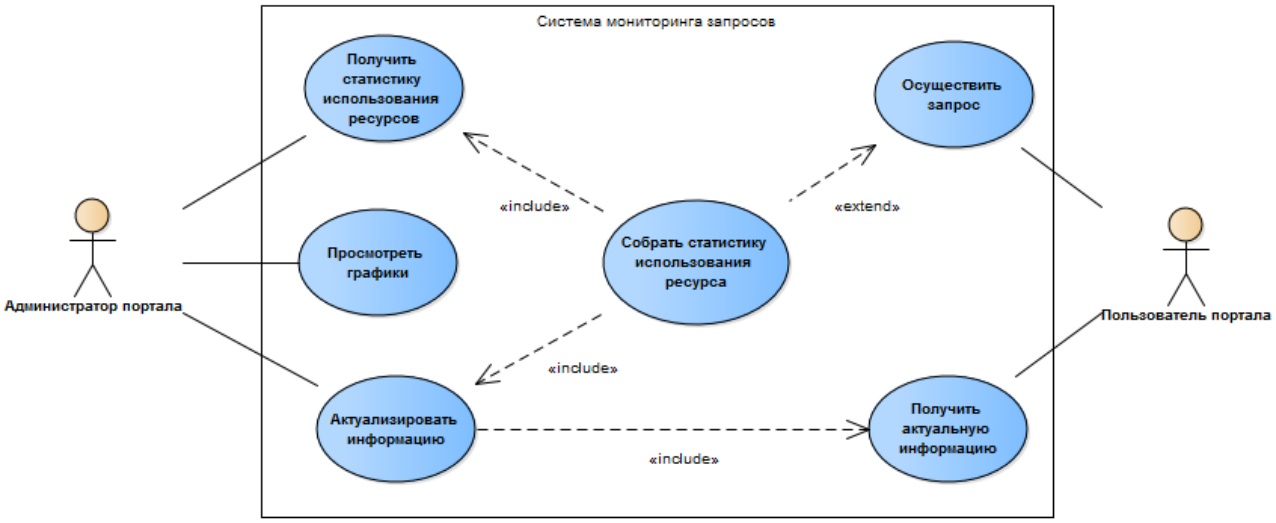


Рисунок 2.9 – Диаграмма вариантов использования

На рисунке 2.9 видно, что система имеет двух основных «актеров», осуществляющих действия: «Пользователь портала» и «Администратор портала». Прецедентами пользователя являются осуществление запроса и получение актуальной информации, а вариантами использования администратора являются получение статистики, просмотр графиков запросов к информационным ресурсам и актуализация информации.

На диаграмме видны некоторые зависимости между модулями системы. Например, пользователь не может получить актуальную информацию без ее актуализации администратором портала. В свою очередь, актуализация данных невозможна без статистики использования ресурсов.

На основе диаграммы вариантов использования можно построить диаграмму классов, которая позволит подробнее рассмотреть функции и аспекты каждого модуля разрабатываемой системы. Но прежде чем переходить к ее проектированию, необходимо определить архитектуру и средства реализации будущей системы.

## 2.4 Обоснование выбора архитектуры и средств реализации системы мониторинга запросов к информационным ресурсам

Портал корпоративных знаний планируется реализовывать в виде веб-приложения. Для реализации системы мониторинга запросов удобнее всего будет применить MVC-подход, который в настоящее время является одним из проверенных временем трендов в веб-разработке.

MVC (Model-View-Controller) – схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные [31]. Концепция MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента:

1. Модель. Предоставляет данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации как эти знания можно визуализировать. Зачастую, модель является объектным отображением таблиц базы данных.

2. Представление. Отвечает за визуализацию информации. Часто в качестве представления выступает форма с графическими элементами. В большинстве случаев, в представление передается обработанная контроллером модель для отображения запрашиваемого пользователем ресурса.

3. Контроллер. Обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

Важно отметить, что как представление, так и контроллер зависят от модели. Однако модель не зависит ни от представления, ни от контроллера. Тем самым достигается назначение такого разделения: оно позволяет строить модель независимо от визуального представления, а также создавать несколько различных представлений для одной модели.

Кроме того, в приложении должен присутствовать механизм кэширования. Необходимость в нем обусловлена большими объемами обрабатываемых данных, которые должны часто обновляться. В случае его отсутствия повысится нагрузка на базу данных, и могут возникнуть неполадки в работе портала корпоративных знаний от частичного снижения производительности до полного отказа работы сервера. В частности, данный механизм должен присутствовать в процессе актуализации данных.

Определившись с архитектурой приложения, необходимо выбрать средство реализации. Так как решение Atlassian Confluence, используемое в качестве основы для портала корпоративных знаний реализовано на языке Java, то и наиболее удобным способом расширения функционала портала для портала будет написание плагина на языке Java. Этот способ позволит воспользоваться всеми возможностями платформы и упростит разработку системы. Основной архитектурной особенностью приложения становится то, что портал использует платформу Spring. Он обеспечивает реализацию MVC подхода, а также принципа Inversion of Control (IoC). IoC - это абстрактный принцип, набор рекомендаций для написания слабо связанного кода, суть которого состоит в том, что каждый компонент системы должен быть как можно более изолированным от других, не полагаясь в своей работе на детали конкретной реализации других компонентов [31]. Также, для реализации web-приложения потребуется REST-модуль. Он необходим для построения веб-сервиса, с помощью которого будет происходить общение клиента с сервером. Еще одной архитектурной особенностью является использование регулярных задач на основе движка Quartz Job Scheduler. Он необходим для задания автоматического кэширования данных по заранее определенным временным

интервалам. Кроме того, его использование необходимо для очистки базы данных от переполнения.

Для работы поисковой машины в Atlassian Confluence используется движок Apache Lucene. Он предоставляет API для расширения функционала и именно на этом API будет реализован механизм актуализации данных.

Lucene - это свободная библиотека для высокоскоростного полнотекстового поиска, написанная на Java. Может быть использована для поиска в интернете и при решении различных задач вычислительной лингвистики. Основными преимуществами данного движка считается:

- высокоскоростная индексация;
- малый объем используемой RAM;
- высокая степень сжатия индекса;
- поиск, основанный на «полях»;
- возможность сортировать результаты поиска по различным полям;
- возможность одновременного поиска и обновления индекса.

Поиск в Apache Lucene осуществляется на основе проиндексированных данных. Одним из базовых понятий при индексации в Apache Lucene является «терм». **Терм** – объект, состоящий из двух элементов: строки со значением и поля, в котором это значение появляется в документе [52]. Именно по соответствию терма и текста запроса происходит формирование результирующего массива документов. **Документ** в Apache Lucene – это индексируемый объект [52]. Он содержит в себе набор полей, которые содержат в себе некоторые текстовые данные. **Поле** – элемент документа, содержащий некоторые данные [52]. Например, это может быть дата создания, автор, заголовок, тело документа, дата последнего редактирования. Не все поля являются индексируемыми, т.к. некоторые из них участвуют в процессе поиска лишь косвенно. Т.е. влияют лишь как дополнительные коэффициенты. Несмотря на то, что терм формируется на основе значений в поле документа, он

не является дочерним элементом поля, а представляет собой отдельный элемент, играющий важную роль в процессе поиска.

Расчет индексов в Lucene основан на комбинации двух моделей информационного поиска: бинарной и векторной. Бинарная модель информационного поиска основана на том, что документ должен соответствовать искомому запросу. При этом применяется алгебра логики.

Пусть имеется множество элементов T, содержащее в себе проиндексированные термы:

$$T = \{t_1, t_2, \dots, t_n\} \quad (2.1)$$

Также имеется множество элементов D, называемых документами, где каждый элемент – подмножество множества T.

Представим поисковый запрос как множество, объединенное в единую строку операциями И, ИЛИ, НЕ:

$$Q = (W_1\ AND\ W_2)\ OR\ W_3 \quad (2.2)$$

где $W_1 – t_i$, $W_2 –$ NOT $t_j$, $W_3 – t_k$,

$t_i$ – означает, что данный терм входит в документ $D_i$.

Операция информационного поиска в данном случае представлена в следующем виде [52]:

1. Формируется множество элементов S, которое содержит документы, содержащие или не содержащие терм $t_j$.
2. Эти документы объединяются в соответствии с исходным запросом пользователя, что приводит к получению результирующего множества данных.

Использование только бинарной модели поиска не оптимально, т.к. она не учитывает различные веса элементов и требует точного соответствия искомому запросу. Поэтому Lucene использует еще и векторную модель. Она представляет коллекцию документов векторами из одного общего для всей коллекции векторного пространства.

Документ в векторной модели рассматривается как неупорядоченное множество термов [59]. Различными способами можно определить вес терма в

документе — «важность» слова для идентификации данного текста. Например, можно просто подсчитать количество употреблений терма в документе, так называемую частоту терма, — чем чаще слово встречается в документе, тем больший у него будет вес. Если терм не встречается в документе, то его вес в этом документе равен нулю. Все термы, которые встречаются в документах обрабатываемой коллекции, можно упорядочить. Если теперь для некоторого документа выписать по порядку веса всех термов, включая те, которых нет в этом документе, получится вектор, который и будет представлением данного документа в векторном пространстве. Размерность этого вектора, как и размерность пространства, равна количеству различных термов во всей коллекции, и является одинаковой для всех документов. Располагая таким представлением для всех документов, можно, например, находить расстояние между точками пространства и тем самым решать задачу подобия документов — чем ближе расположены точки, тем больше похожи соответствующие документы. В случае поиска документа по запросу, запрос тоже представляется как вектор того же пространства — и можно вычислять соответствие документов запросу.

Механизм Apache Lucene, опираясь на две вышеперечисленные модели, предоставляет различные стратегии поиска, в том числе существует возможность как расширить базовые алгоритмы, так и использовать собственные. Основными при поиске являются TF-IDF Similarity и BM25 Similarity.

TF-IDF (Term frequency - Inverse Document Frequency) - статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса [59]. Вес некоторого слова пропорционален количеству употребления этого слова в документе, и обратно пропорционален частоте употребления слова в других документах коллекции. Вес документа при использовании этой стратегии рассчитывается по формуле:

$$s(q,d) = c(q,d) \cdot qN(q) \cdot \sum(tf(t\ in\ d) \cdot idf(t)^2\ \cdot t.Boost() \cdot norm(t,d)) \quad (2.3)$$

tf(t in d) – частота терма, описывающая частоту появлений терма в конкретном документе. Обычно рассчитывается по формуле:

$$tf(t\ in\ d) = \sqrt{frequency} \quad (2.4)$$

где frequency – число появлений терма t в документе d;

idf(t) - обратная частота документа.

Инверсия частоты, с которой некоторое слово встречается в документах коллекции. Данная мера предназначена для уменьшения влияния часто встречающихся слов, таких как предлоги или союзы, на общий вес документа [44]. IDF рассчитывается по формуле:

$$idf(t) = 1 + \log\left(\frac{numDocs}{docFreq+1}\right) \quad (2.5)$$

где numDocs – общее число документов в индексе,

docFreq – число документов, в которых встречается терм t,

t.Boost() – множитель терма. Это значение показывает насколько данный терм является значимым для всего текста запроса,

norm(t,d) – данное значение отвечает за значимость поля, в котором был найден терм. Оно рассчитывается при индексации всех документов портала и сохраняется в индекс.

Остальные значения рассчитываются уже на момент прихода поискового запроса в систему и перестраивают вес документа в соответствии со всем запросом:

- c(q,d) – масштабирующий коэффициент, основанный на том, сколько термов из искомого запроса q было найдено в документе d. Обычно, документ, содержащий большинство термов из запроса получает больший ранг, чем иной документ с меньшим значением числа входов термов в документ.
- qN(q) – коэффициент нормализации, используемый для того, чтобы рейтинги весов можно было сравнивать между собой корректным образом. Рассчитывается по формуле:

$$qN(q) = \frac{1}{\sqrt{sumOfSquaredWeights}} \quad (2.6)$$

где sumOfSquaredWeights - сумма квадратов весов, рассчитываемая объектом Weight поискового запроса.

Для разных типов запросов рассчитывается по-разному. Используемая формула для булева запроса:

$$sumOfSquaredWeights = q.getBoost()^2 \cdot \sum_{t\ in\ q}(idf(t) \cdot t.getBoost())^2 \quad (2.7)$$

где q.getBoost() – весовой коэффициент запроса,

idf(t) - обратная частота документа,

t.Boost() – множитель терма.

BM25 Similarity основана на функции ранжирования OKAPI BM25. Это – функция ранжирования, используемая поисковыми системами для упорядочивания документов по их релевантности данному поисковому запросу. Она основывается на вероятностной модели, разработанной в 1970-х - 1980-х годах. BM25 и его различные более поздние модификации (например, BM25F) представляют собой современные TF-IDF-подобные функции ранжирования, широко используемые на практике в поисковых системах [55].

BM25 — поисковая функция на неупорядоченном множестве термов («мешке слов») и множестве документов, которые она оценивает на основе встречаемости слов запроса в каждом документе, без учёта взаимоотношений между ними (например, близости) [52]. Это не одна функция, а семейство функций с различными компонентами и параметрами. Одна из распространенных форм этой функции описана ниже.

Пусть дан запрос Q, содержащий слова $q_1$, ..., $q_n$, тогда функция BM25 даёт следующую оценку релевантности документа D запросу Q:

$$s(D,Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D)+k_1\cdot\left(1-b+b\cdot\frac{|D|}{avgdl}\right)} \quad (2.8)$$

$f(q_i,D)$ - частота слова $q_i$ в документе D,

|D| - количество слов в документе,

Avgdl – средняя длина документа в индексе,

K1 и b – свободные коэффициенты. В платформе Confluence эти значения выбраны как k1 = 1.25, b = 0.3.

Значение IDF в алгоритме DM25 рассчитывается отличным от используемого в TF-IDF способа. Формула следующая:

$$IDF(q_i) = \log(1 + \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}) \quad (2.9)$$

Здесь N – общее число документов в индексе. $n(q_i)$ – число документов, содержащих терм $q_i$.

В Atlassian Confluence используется алгоритм BM25L. Это – развитие алгоритма BM25. Он является более продвинутым и позволяет нормализовать результаты поиска. В нем частота появления терма имеет меньшее значение, чем в TD-IDF алгоритме и позволяет появляться в поисковой выдаче другим значениям.

Помимо алгоритмов, используемых Lucene для поиска, Confluence накладывает дополнительные коэффициенты при помощи Boosting Query. Это – расширение для Lucene, позволяющее переопределить результаты поиска уже после того, как отработала основная логика поиска и сформировался финальный массив данных. Ключевыми особенностями здесь является то, что наличие текста запроса в заголовке статьи повышает ее коэффициент в 2 раза. Также, Confluence предполагает наличие модификатора по дате создания/редактирования статьи. Все вышеперечисленное никак не учитывает популярности статьи у пользователей, и как следствие, может нарушить релевантность поиска.

Рассмотрим пример расчета коэффициентов для документа «Welcome to Confluence» с суммарным весом 8,151545 по BM25L алгоритму по поисковой фразе «Welcome».

Оценка релевантности документа по алгоритму BM25L рассчитывается по следующей формуле:

$$s(D, Q) = \sum_{i=1}^{n} (IDF(q_i) \cdot tfNorm \cdot boost) \quad (2.10)$$

где tfNorm рассчитывается по формуле:

$$tfNorm = \frac{(k1+1) \cdot (cPrime + d)}{k1 + cPrime + d} \quad (2.11)$$

где cPrime рассчитывается по формуле:

$$cPrime = \frac{freq}{(1-b+b\cdot\frac{D}{avgdl})} \quad (2.12)$$

где freq – частота появления терма в документе,

D - количество слов в документе,

Avgdl – средняя длина документа в индексе,

k1, b и d – свободные коэффициенты. В данном случае равны: k1=1,25, b=0,3, d=0,5.

Рассчитаем меру IDF для этого терма по формуле (2.9). N имеет значение 65, а $n(q_i)$ – 2. Отсюда значение IDF будет равно:

$$IDF(q_i) = \log(1 + \frac{65-2+0.5}{2+0.5}) = \log(26.4) = 3.273 \quad (2.13)$$

Значение tfNorm при мерах freq=1, avgdl = 3.0307693, D=2.56 будет следующим:

$$cPrime = \frac{1}{(1-0,3+0,3\cdot\frac{2,56}{3,03})} = 1,048876 \quad (2.14)$$

Соответственно, tfNorm будет равно:

$$tfNorm = \frac{(1,25+1)\cdot(1,048876+0,5)}{1,25+1,048876+0,5} = 1,2451325 \quad (2.15)$$

Учитывая то, что искомый терм появился в заголовке ресурса, значение boost будет равно 2. Перемножив полученные значения из формул 2.14 и 2.15, получим 1,2151325*3,273*2=8,151545. Таким образом, было доказано что алгоритм расчета весов в Atlassian Confluence полностью соответствует алгоритму расчета BM25LSimilarity. Данный алгоритм учитывает значимость термов в индексе и количество его появлений, а также позволяет поисковому движку учитывать поля, в которых искомый запрос был найден.

Java предоставляет возможности только для написания логики серверной части системы мониторинга запросов. Для реализации клиентской части воспользуемся языком разметки HTML, т.к. Confluence – это именно WEB-портал. Интерактивность Web-приложений обычно обеспечивается языком JavaScript. Он позволяет добавить графическое отображение к текстовому

интерфейсу, тем самым повышая эффективность восприятия информации пользователем. Кроме того, JavaScript будет необходим для асинхронного сбора статистики о поисковых запросах, т.к. в случае, если данная операция будет производиться синхронно, это будет влиять на работу пользователей. Технология AJAX позволит реализовать данное требование.

Таким образом, были определены основные архитектурные особенности реализации системы мониторинга запросов к информационным ресурсам. Приложение будет реализовываться на языке Java с применением принципа IoC. Также будут использованы технологии REST, библиотека Quartz Job Scheduler, и API поискового движка Apache Lucene. Для отображения данных пользователю будет использована связка HTML + Javascript. Описание архитектуры системы и средств ее реализации позволяет нам перейти к построению логической модели системы мониторинга запросов.

## 2.5 Построение логической модели системы мониторинга запросов

Для того чтобы рассмотреть элементы системы в понятиях объектно-ориентированного программирования, т.е. представить их в виде интерфейсов и классов, необходимо построить диаграмму классов. Она служит для определения сущностей предметной области, описания их структуры и связей между ними, а также разделения приложения на основные слои.

На рисунке 2.10 представлена диаграмма классов модуля отчетности. Ключевым классом, как видно на диаграмме, является абстрактный класс Abstract Report. От него наследуются все классы отчетов. Также, в нем определены методы, которые должны быть переопределены в дочерних классах. В частности, метод getReport() предназначен для получения массива элементов для их дальнейшего преобразования в отчет. Кроме того, на диаграмме видно, что применен шаблон проектирования «Фабричный метод». Он позволяет снизить связанность кода, не привязываясь к конкретным классам, а оперируя лишь общим интерфейсом.

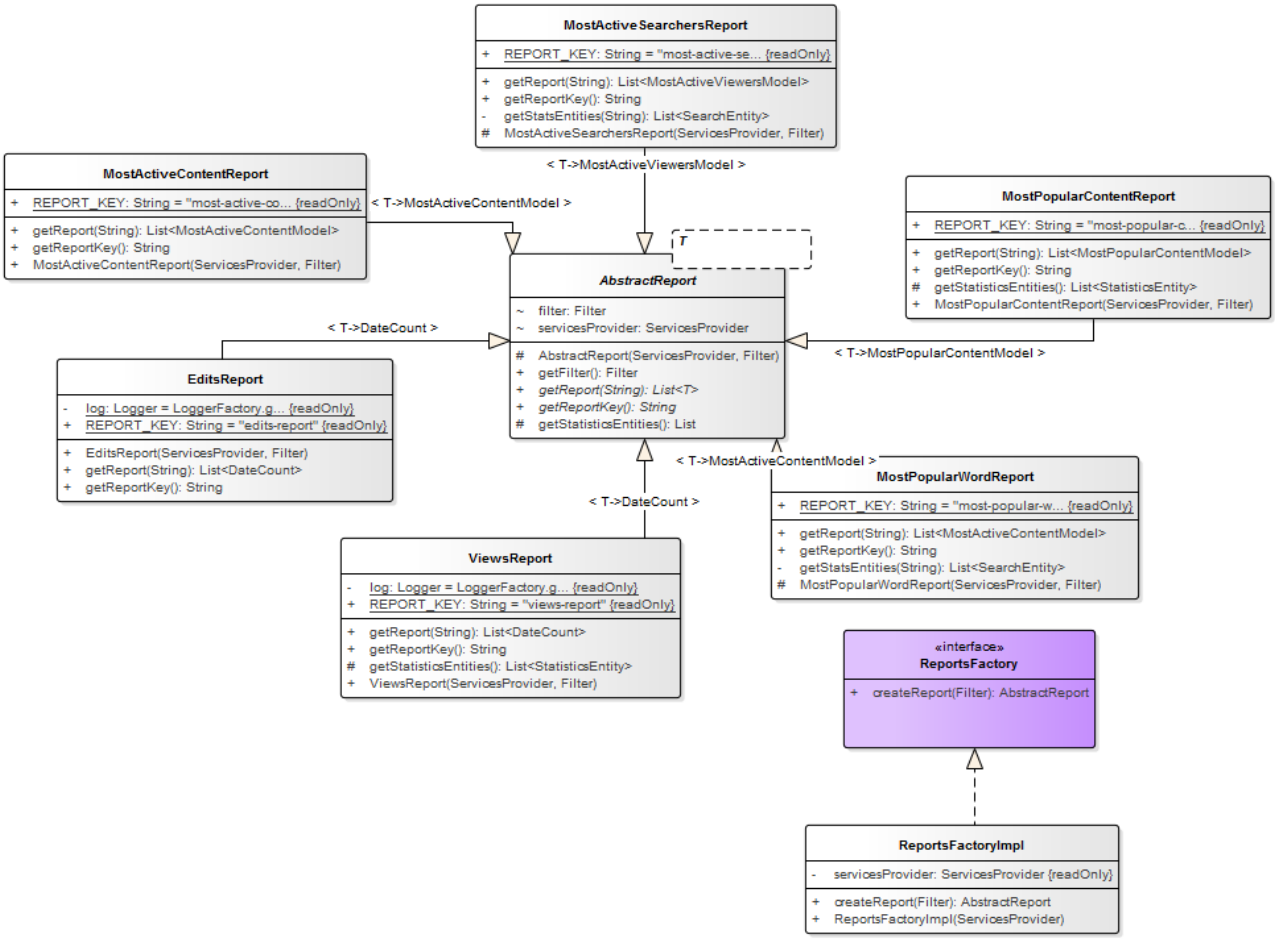


Рисунок 2.10 – Диаграмма классов (Модуль отчетности)

На рисунке 2.11 представлена диаграмма классов модуля, отвечающего за актуализацию ресурсов.

Актуализация ресурсов построена на движке Lucene. Класс Views Extractor позволит собрать статистику использования ресурсов из кэша, сформированного Cache View Statistics и поместить ее в индекс Lucene.

В дальнейшем использование класса Boost Popular Strategy позволит модифицировать результаты поиска, наложив коэффициенты на сформированный массив данных, базируясь на значениях в индексе Lucene.

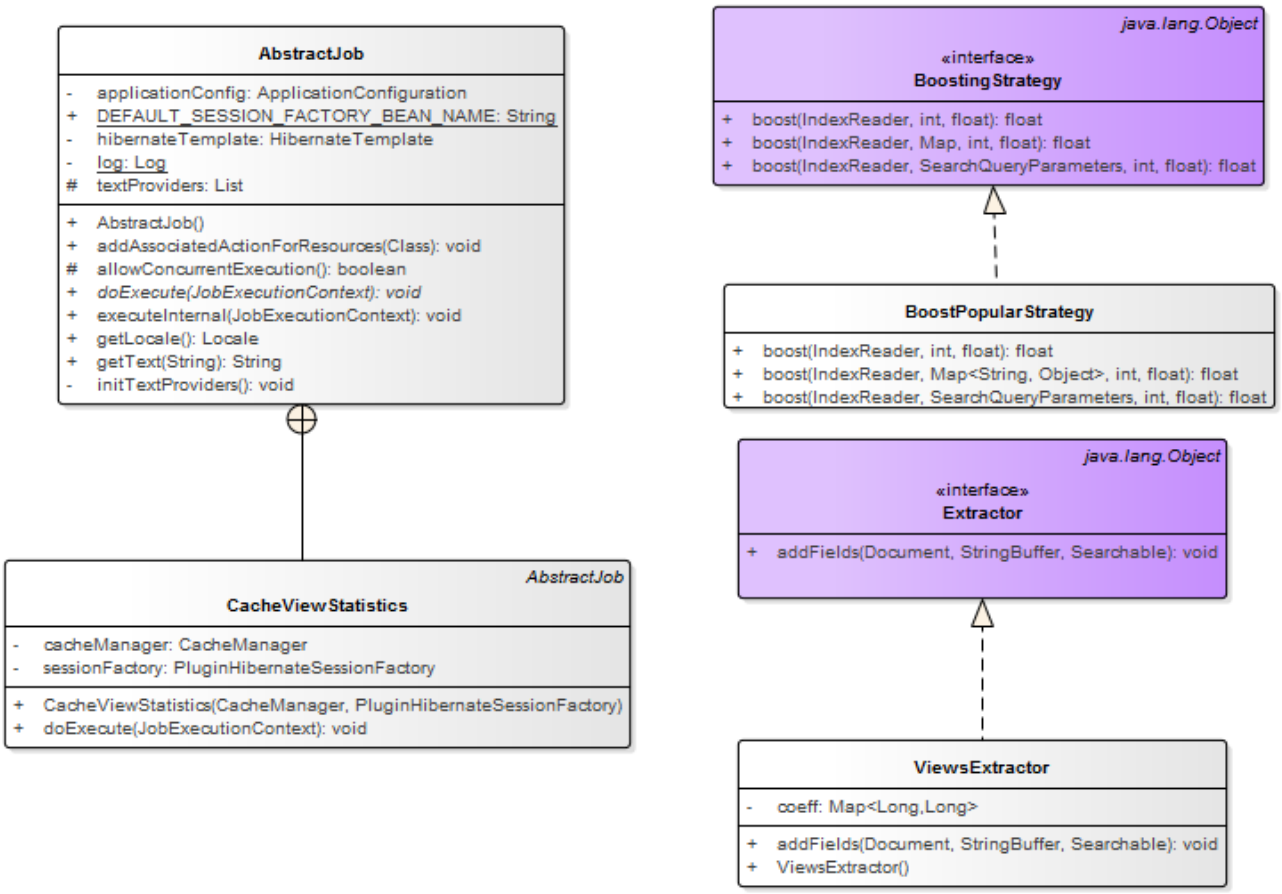


Рисунок 2.11 – Диаграмма классов (Модуль актуализации)

На рисунке 2.12 представлена диаграмма классов сервисного слоя приложения, т.е. слоя, на котором описывается бизнес-логика.

Основным интерфейсом, через который реализуется функционал разрабатываемой системы, является Confluence Statistics Service. Он описывает основные бизнес-операции, реализуемые приложением. Это – добавление статистических данных, их обработка, считывание согласно запросу, а также их удаление. Все эти операции лишь делегируются слою, работающему с базой данных. Эти операции на данной диаграмме не указаны.

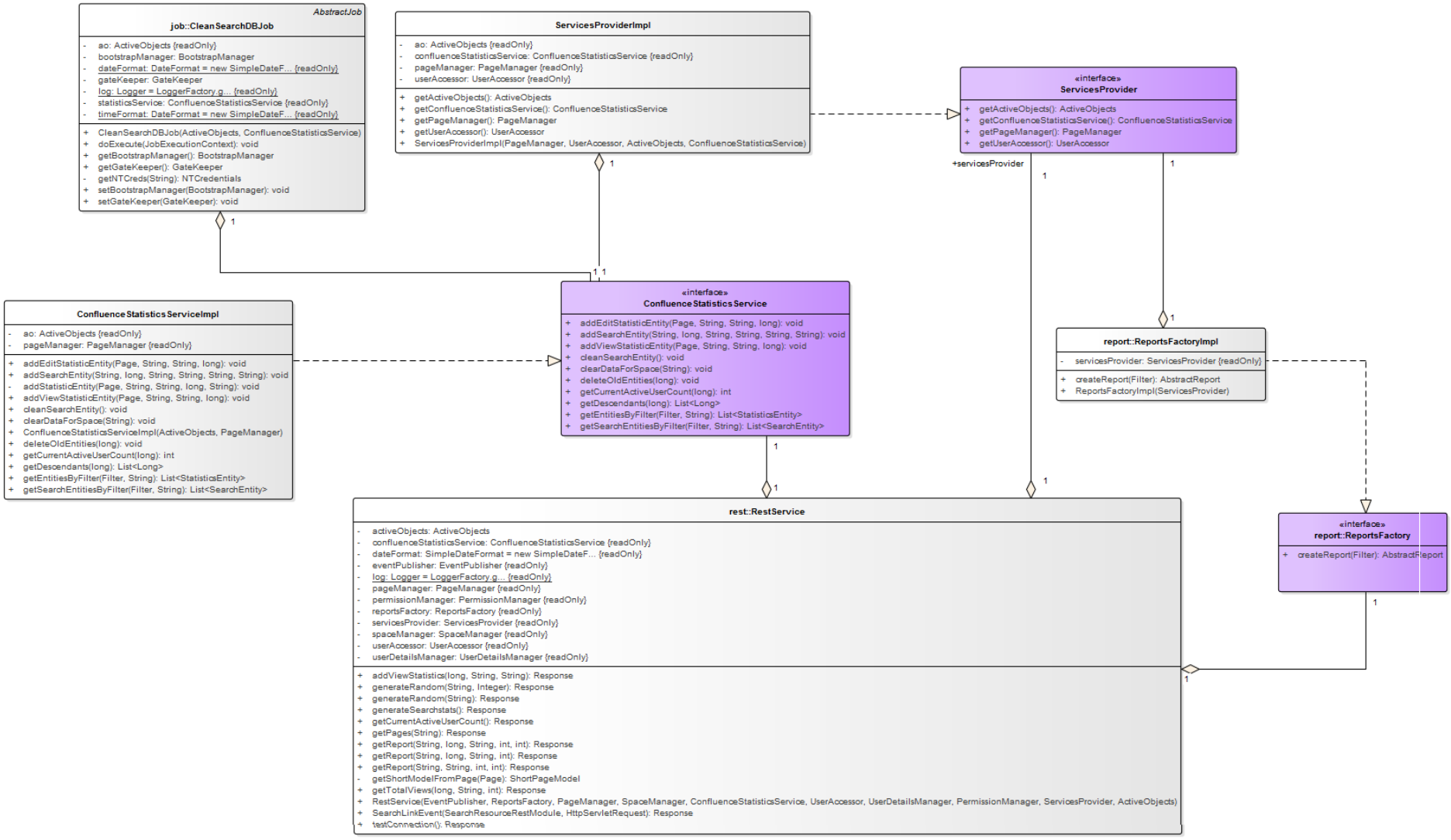


Рисунок 2.12 – Диаграмма классов (Сервисный слой)

Ключевым в данном случае является именно интерфейс, а не класс, в связи с тем, что применен принцип Inversion of Control, используемый для уменьшения связанности между классами в приложениях. Такое архитектурное решение упрощает расширение возможностей системы, при котором контроль над потоком управления программы остаётся за каркасом. В данном случае принцип реализован с помощью Dependency Injection. Это – принцип, позволяющий «положить» в специальный контейнер все необходимые зависимости, которые в дальнейшем можно вызывать, используя конструктор или метод класса. Именно DI обуславливает появление на диаграмме классов агрегаций между классами.

Диаграмма классов позволила рассмотреть основные элементы системы. Но для построения логической модели системы недостаточно рассмотреть систему в статике, поэтому существует потребность в диаграмме последовательности. Она позволит описать взаимодействие объектов и модулей системы, а также проследить основной поток процессов приложения. Диаграмма последовательностей необходима для обозначения очередности следования друг за другом различных сообщений, с помощью которых объекты взаимодействуют между собой. Главный акцент при построении этой диаграммы ставится на порядок и динамику поведения, т.е. как и в каком порядке происходят события.

На рисунке 2.13 представлены основные объекты системы и пути взаимодействия между ними в течение работы системы. На диаграмме видны два основных модуля, через которые идут потоки взаимодействия в системе, а также два хранилища данных: «Статистика использования ресурсов» и «Информационные ресурсы». Инициатором работы системы на данной диаграмме является сотрудник компании, обращающийся к корпоративному порталу знаний за определенным информационным ресурсом. Модуль навигации, обработав запрос, определяет его тип: переход по страницам или информационный поиск и передает данные о нем в модуль обработки запросов. Обработчик запросов перехватывает данные запроса и отправляет их в

хранилище статистики, после чего происходит обработка запроса и поиск релевантных данных. Результатом работы этого алгоритма становится массив информационных ресурсов, соответствующий пользовательскому запросу.

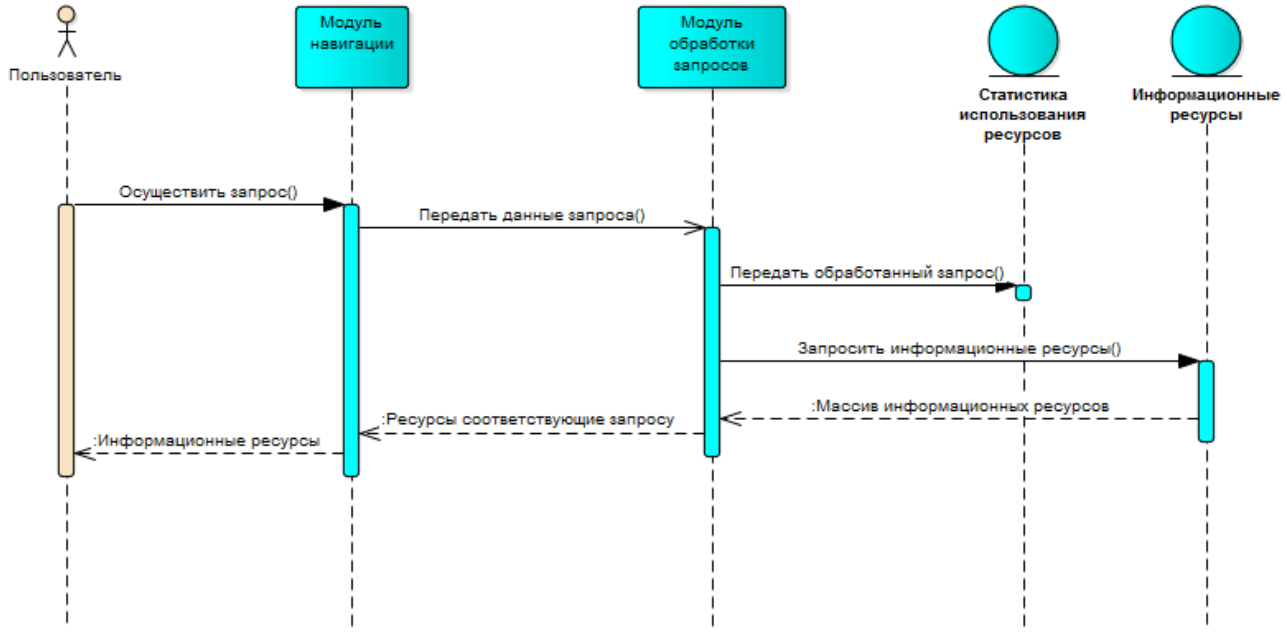


Рисунок 2.13 – Диаграмма последовательности (пользователь)

Рисунок 2.13 отобразил только одну сторону поведения системы. Для проведения полноценного анализа работы системы мониторинга запросов к информационным ресурсам, диаграмму последовательности с точки зрения администратора портала корпоративных знаний. Она представлена на рисунке 2.14.

Как видно из диаграммы последовательности, администратор несколько раз обращается к системе. Для выполнения основной задачи – актуализации ресурсов, необходимо вручную проанализировать статистику использования ресурсов, которая предоставляется в результате запроса к одноименному хранилищу данных. Следующим шагом является анализ графических моделей использования информационных ресурсов. Еще одним действием, выполняемым администратором портала, является асинхронный запрос к модулю актуализации данных. Этот модуль собирает данные из хранилища статистики и асинхронно актуализирует информационные ресурсы, при этом информируя администратора о прогрессе.

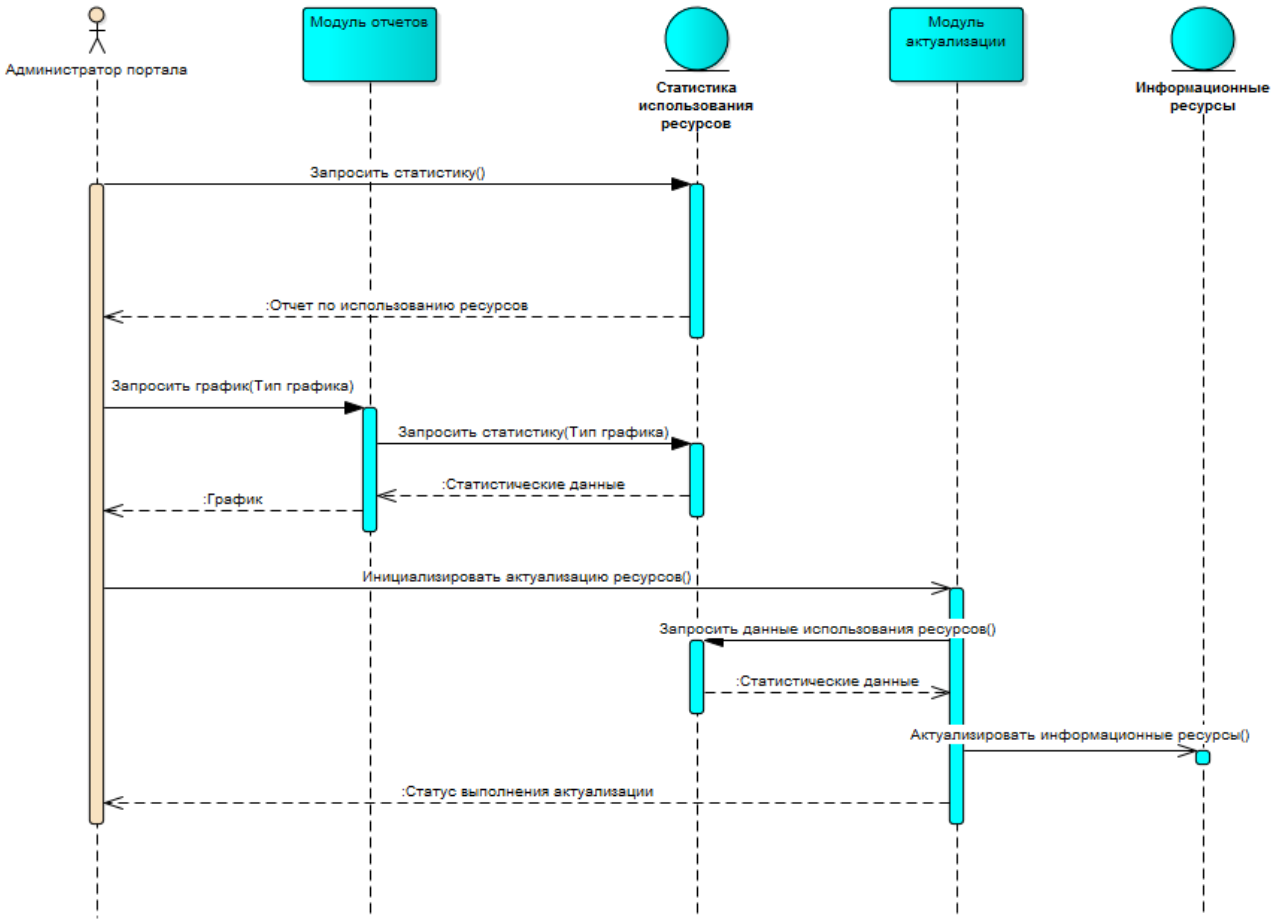


Рисунок 2.14 – Диаграмма последовательности (администратор портала)

В результате анализа построенных диаграмм были определены основные функции программы, и сформирована последовательность их выполнения. Для более подробного рассмотрения системы необходимо описать алгоритм ее работы.

## 2.6 Описание основных алгоритмов работы системы мониторинга запросов к информационным ресурсам портала корпоративных знаний

Построение диаграммы последовательности позволяет рассмотреть взаимодействие объектов друг с другом, но она представляет их в достаточно абстрактном виде, не учитывая особенностей и условий работы системы. Поэтому, необходимо построение диаграммы деятельности. Она позволит подробно рассмотреть динамику работы системы.

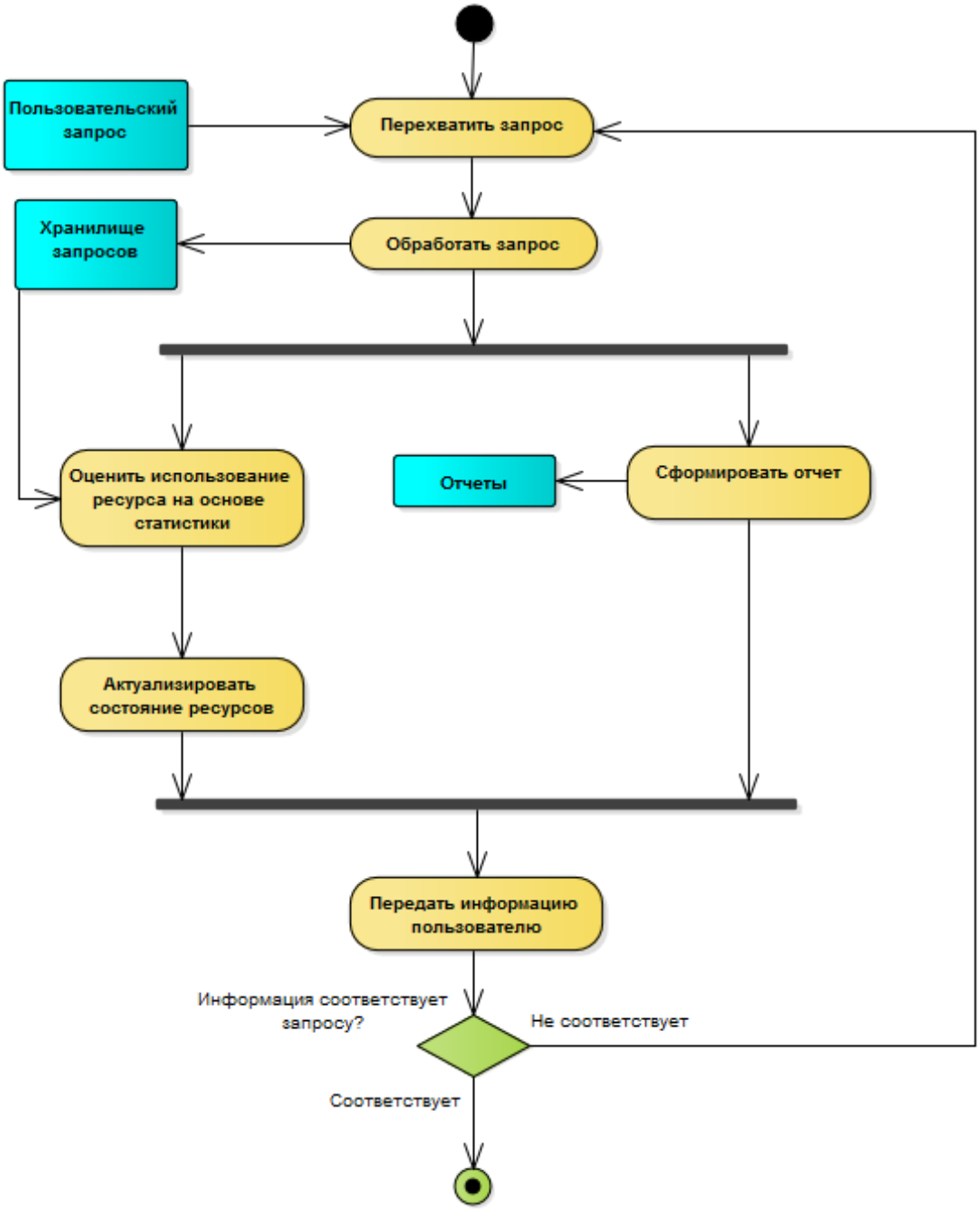


Рисунок 2.15 - Диаграмма деятельности

В результате моделирования системы был определен общий алгоритм работы приложения. Мониторинг запросов начинается с перехвата приложением пользовательского запроса. В дальнейшем, запрос обрабатывается и передается в хранилище запросов согласно его типу и заранее определенным критериям. Следующие процессы в системе происходят параллельно. В частности, выполняется задача по формированию отчетов для администратора системы. Она позволяет определить нагрузку на портал

знаний, а также определить потребность пользователей в определенных ресурсах. Еще одним процессом является оценка использования ресурсов на основе статистики запросов. На основе этой оценки происходит актуализация информационных ресурсов. После выполнения всех этих процессов, потоки приложения объединяются, и происходит передача пользователю информации согласно запросу. В зависимости от полученного результата, работа системы мониторинга завершается или продолжает работу до достижения информации, релевантной запросу.

## Вывод по главе

В ходе моделирования были определены основные аспекты разрабатываемой системы мониторинга запросов. Построены концептуальная и логическая модели системы, в которых были описаны элементы, с помощью которых будет осуществляться взаимодействие между модулями системы. В качестве архитектуры системы был выбран паттерн MVC, разделяющий между собой слои системы. Также, были определены технологии, которые будет применяться при разработке приложения. Описаны механизмы работы поисковой машины Apache Lucene. В ходе проектирования была определена логика работы системы при помощи диаграммы последовательности и диаграммы активности.

# ГЛАВА 3 Реализация системы мониторинга запросов к информационным ресурсам портала корпоративных знаний

## 3.1 Проектирование базы данных системы мониторинга запросов к информационным ресурсам портала корпоративных знаний

### 3.1.1 Концептуальное проектирование модели данных

Моделирование приложения необходимо, но недостаточно для охвата всего процесса проектирования системы, так как практически ни одна система не обходится без хранилища данных. В связи с этим, существует потребность в описании модели данных приложения. Начальной ступенью при проектировании базы данных является построение концептуальной модели, которая позволяет описать смысловую структуру данных. Концептуальная модель представляет собой описание основных сущностей (таблиц) и связей между ними без учета принятой модели БД и синтаксиса целевой СУБД [19]. Построение ER модели будет удобнее всего осуществить с помощью CASE-средства MySQL Workbench. Хоть оно и имеет в своем названии слово «MySQL», оно представляет полный спектр инструментов для проектирования моделей данных.

На рисунке 3.1 представлена концептуальная модель данных. Рассмотрев ее, можно заметить, что таблица «Content» имеет большое число связей, в том числе и рекурсивных. Рекурсивная связь – это связь сущности с этой же самой сущностью. В данном случае рекурсивная связь является иерархической [19]. Она является наиболее подходящей для хранения древовидной структуры ресурсов. Именно эта структура и используется в платформе Confluence. Кроме того, рекурсивная связь используется в том случае, если планируется большое число разнообразного контента, который имеет сходные свойства и родительские элементы. Также, на представленной схеме присутствует сущность «Spaces», являющаяся родительской для большинства элементов сущности «Content». Сущности «Space_stat_entity» и «Search_entity» хранят статистику пользовательских запросов, на основе которой и осуществляется

процесс актуализации ресурсов. Стоит обратить внимание на то, что подавляющее число связей между сущностями - 1:M.

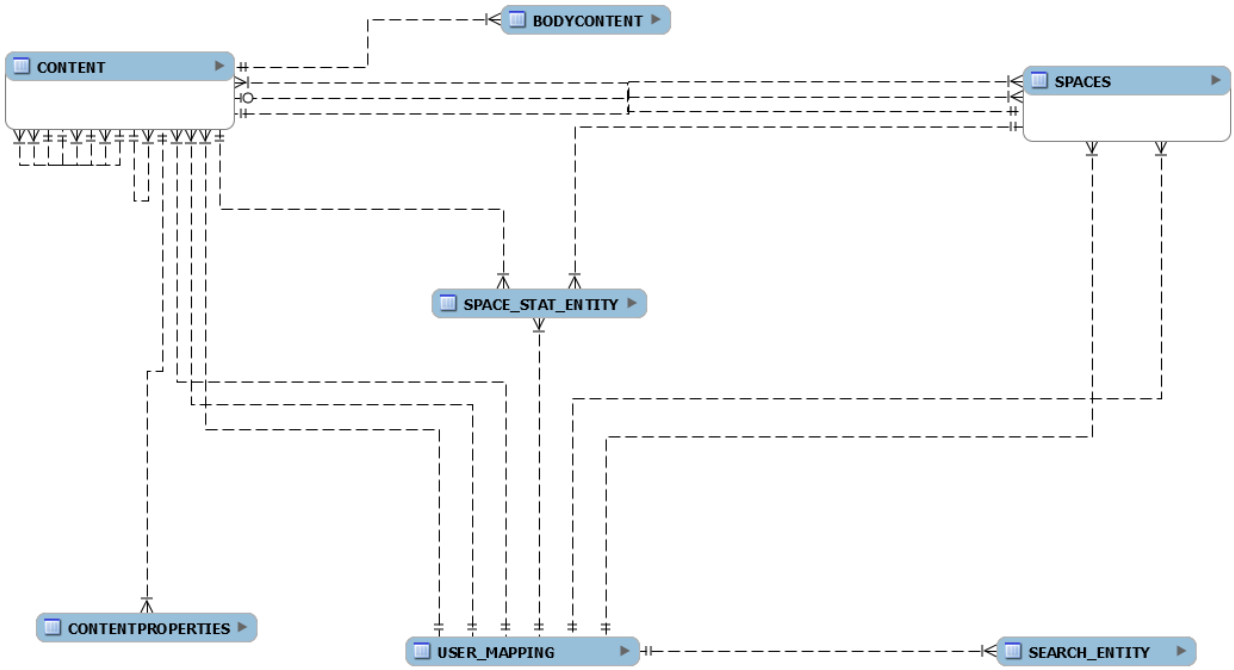


Рисунок 3.1 – Концептуальная модель данных

Построение концептуальной модели является первым шагом проектирования базы данных. Для полной демонстрации модели данных перейдем к этапу логического проектирования.

### 3.1.2 Логическое проектирование модели данных

Логическая модель описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью. Одним из важнейших аспектов логической модели данных является избавление от связей много-ко-многим [19]. Так как такой вид связи отсутствует на концептуальной модели данных, то и логическая модель тоже не поменяет свой вид. Для проектирования использовалось Case-средство ERWin Community Edition. Данная версия распространяется бесплатно для всех пользователей, но частично ограничена в функционале. ERWin представляет инструментарий для построения как логической, так и физической моделей данных, а также имеет возможности прямого и реверс-инжиниринга.

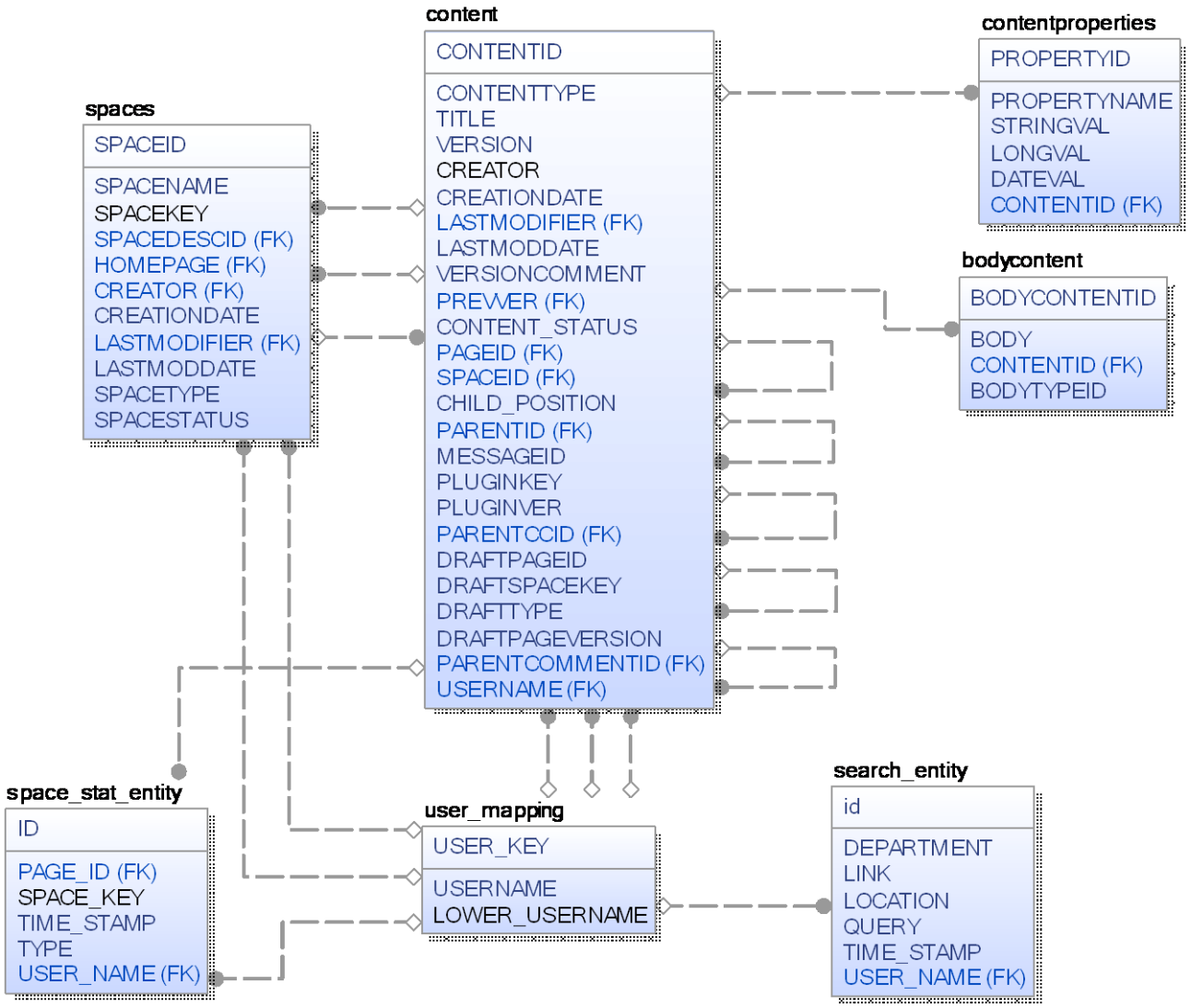


Рисунок 3.2 - Логическая модель данных

На логической модели данных просматриваются первичные и внешние ключи таблиц, на основе которых формируются связи между таблицами. Таблица «Content» представлена большим числом полей, в связи с тем, что она объединяет в себе около 10 различных типов данных. Таблица «Bodycontent» существует для уменьшения объема таблицы «Content», т.к. в поле «Body» помещается весь контент объекта базы знаний. Рассмотрев таблицу «user_mapping», можно обратить внимание, что в ней присутствует лишь первичный ключ и поле «username». Информация о пользователях в платформе Confluence хранится в отдельной таблице. На логической модели данных эта таблица не представлена в связи с ее незначимостью в контексте разработки системы мониторинга запросов к информационным ресурсам.

Следующим этапом проектирования базы данных является построение физической модели данных, которая позволит определить финальную структуру базы данных. Но перед этим необходимо определиться с систему управления базами данных.

### 3.1.3 Обоснование выбора системы управления базами данных

Вид физической модели данных зависит от выбранной СУБД. При разработке системы мониторинга запросов использовалось две СУБД: H2 и MySQL.

H2 - открытая кроссплатформенная СУБД, полностью написанная на языке Java. База данных имеет очень малый размер, но, несмотря на это, обладает большим количеством возможностей [45]. Одним из плюсов этой СУБД является то, что она может разворачиваться при запуске приложения. Эта возможность позволяет легко начать разработку, не акцентируя внимания на конфигурировании базы данных. Но H2 обладает и крупным минусом, являющимся платой за высокую производительность и не позволяющим использовать базу данных в промышленных нуждах. Это – низкая надежность. В частности, при отключении электропитания, часть данных в незавершенных транзакциях может быть потеряна. Это может быть критично при использовании системы множеством пользователей и частом обращении к информационным ресурсам. Поэтому, для промышленной эксплуатации была выбрана СУБД MySQL.

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems [41]. Выбор СУБД был осуществлен по ряду причин:

1. MySQL является решением для малых и средних приложений. Оценивая ожидаемые нагрузки на портал корпоративных знаний, было

выяснено, что не имеет смысла использовать такие крупные и дорогие решения, как Oracle DB или MSSQL Server.

2. MySQL является достаточно гибким решением, что позволяет конфигурировать ее в зависимости от решаемых задач [40].

3. Отказоустойчивость. Система позволяет настроить механизм репликации, что обеспечит целостность базы, вне зависимости от внешнего воздействия [40].

4. Быстродействие. Опираясь на значения ожидаемых нагрузок, можно считать, что MySQL является оптимальным решением по соотношению надежность/быстродействие.

5. СУБД является бесплатной. Затраты на эксплуатацию системы будут ниже, чем если бы использовались платные решения.

6. MySQL является одним из вариантов, поддерживаемых платформой Confluence и для которой предоставляются драйвера.

Таким образом, для реализации системы мониторинга была выбрана СУБД MySQL.

### 3.1.4 Физическое моделирование данных системы мониторинга пользовательских запросов

Исходя из выбранной базы данных, физическая модель будет выглядеть следующим образом, как представлено на рисунке 3.3.

Построение модели осуществлялось при помощи CASE-средства Enterprise Architect. EA обладает возможностью проектирования баз данных под конкретные СУБД. Это обеспечивается наличием встроенных драйверов, хранящих информацию о типах данных, используемых в конкретной СУБД. Кроме того, EA позволяет осуществлять проектирование полного стека задач, начиная от приложения и заканчивая базой данных. Библиотека проектов включает такие нотации, как UML, BPMN, DFD, IDEF, модели баз данных и т.д.

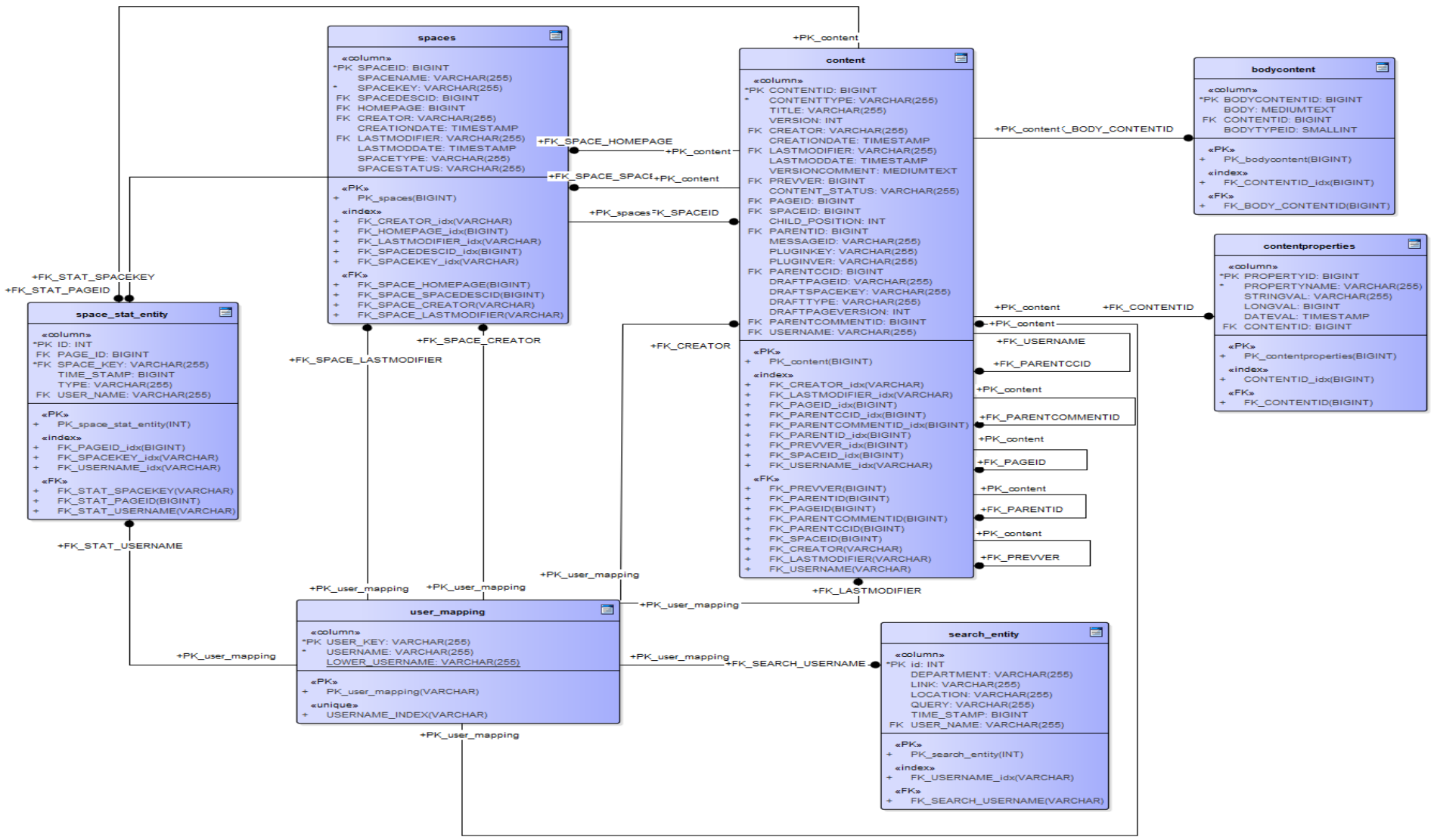


Рисунок 3.3 - Физическая модель данных

При сравнении физической и логической моделей данных видно, что структура базы данных не сменилась, но добавились типы данных полей, используемые в выбранной СУБД. В качестве типов полей преимущественно используются bigint, varchar и timestamp. Использование bigint в качестве типа данных для первичных ключей обусловлено тем, что количество данных портала корпоративных знаний может превысить размер, выделенный под int. На физической модели так же, как и на логической, связи определены в виде ассоциации, но на физической модели отображена мощность отношений. Отношение 1:0..* означает, что связь может отсутствовать, и в этом случае значение внешнего ключа будет null. Используемое CASE-средство позволило осуществить автоматическую генерацию таблиц в базе данных. Для этого потребовалось лишь корректно настроить соединение с базой через ODBC драйвер.

Помимо вышеперечисленных реляционных таблиц в системе мониторинга пользовательских запросов к информационным ресурсам будет использоваться документ-ориентированная база данных, используемая Apache Lucene. Она используется для хранения поисковых индексов. Работа с индексами является одной их важнейших при актуализации данных, так как ресурс, искомый пользователем отобразится ему именно на основе величины индекса документа.

В результате всего вышеперечисленного можно заключить, что было осуществлено проектирование системы мониторинга запросов к информационным ресурсам портала корпоративных знаний, позволившее определить основные функции и модули системы, а также ее пользователей.

## 3.2 Пример реализованной системы мониторинга запросов к информационным ресурсам портала корпоративных знаний

Основные функции системы мониторинга выполняются без участия пользователя. Сбор статистики по событиям просмотра и редактирования информации осуществляется при помощи модуля

ConfluenceStatisticsEventListener. Этот класс позволяет перехватывать события, происходящие с порталом корпоративных знаний. Код методов обработки событий просмотра и редактирования представлен ниже:

```
@EventListener
public void handleEvent(final PageUpdateEvent pageUpdateEvent) {
//Редактирование ресурса
//Создаем два объекта ConfluenceEntityObject, которые получены из события PageUpdateEvent
ConfluenceEntityObject oldEntityObject = pageUpdateEvent.getOld();
ConfluenceEntityObject newEntityObject = pageUpdateEvent.getNew();
long oldId = oldEntityObject != null ? oldEntityObject.getId() : 0;
long newId = newEntityObject != null ? newEntityObject.getId() : 0;
//Проверяем, что ID старого объекта и нового объекта идентичны
if (oldId != newId) {
Page updatedPage = pageUpdateEvent.getPage();
String spaceKey = updatedPage.getSpaceKey();
String userName = updatedPage.getLastModifier().getName();
long timeStamp = updatedPage.getLastModificationDate().getTime();
//создаем новую запись в базе данных
statisticsService.addEditStatisticEntity(updatedPage, spaceKey, userName, timeStamp);
//Логгируем событие
log.info("PageEditEvent: PageID = {}, SpaceKey = {}, UserName = {}, TimeStamp = {}",
        Arrays.asList(updatedPage.getId(), spaceKey, userName, timeStamp).toArray());
}
}
//Просмотр ресурса
@EventListener
```

```
public void handleEvent(final PageViewEvent pageViewEvent) {
//Получение объекта страницы из события просмотра
Page viewedPage = pageManager.getPage(pageViewEvent.getPageId());
//Проверка, что объект страницы не является null
if (viewedPage != null) {
String spaceKey = pageViewEvent.getSpaceKey();
String userName = pageViewEvent.getUserName();
long timeStamp = pageViewEvent.getTimeStamp();
//Создание новой записи в базе данных
statisticsService.addViewStatisticEntity(viewedPage, spaceKey, userName, timeStamp);
//Логгирование события
log.info("PageViewEvent: PageID = {}, SpaceKey = {}, UserName = {}, TimeStamp = {}",
        Arrays.asList(viewedPage.getId(), spaceKey, userName, timeStamp).toArray());
}
}
```

Оба метода аннотированы ключевым словом Event Listener, сигнализирующим JVM о том, что методы должны перехватывать события. Триггером исполнения метода является передаваемый аргумент. Для события просмотра страницы им является Page View Event, а для редактирования – Page Update Event.

На рисунке 3.4 представлена диаграмма активности, на которой представлен алгоритм работы данного модуля. На представленной диаграмме, также, как и в исходном коде, можно заметить, что обработчик сначала проверяет данные на приемлемость и только после этого вносит их в хранилище.

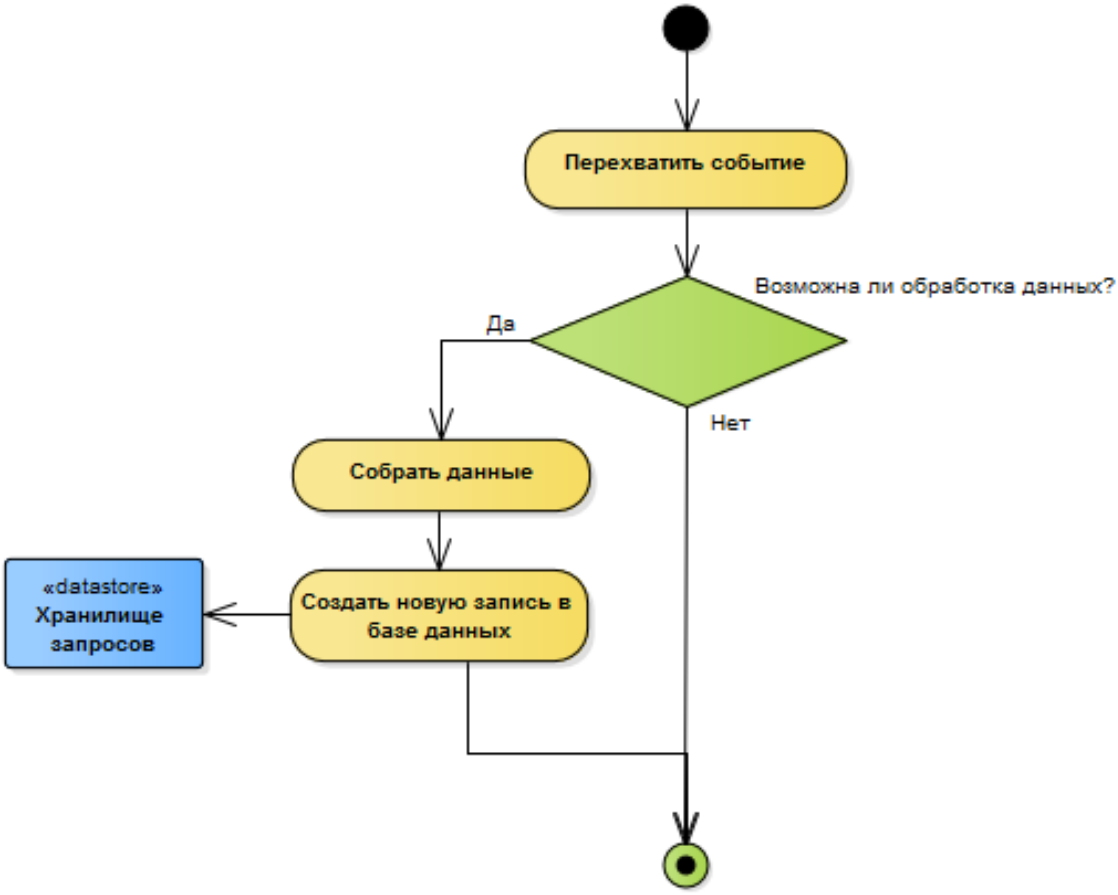


Рисунок 3.4 – Алгоритм работы модуля сбора данных о просмотрах и редактированиях ресурсов

Сбор статистики поисковых запросов пользователей осуществляется двумя способами, т.к. платформа Confluence предоставляет возможность использовать как полноценный поиск по всему порталу знаний, так и индивидуализированный быстрый поиск. В первом случае для перехвата событий поиска используется тот же подход, что и при обработке событий просмотра. Для второго случая применение такого способа сбора статистики не представляется возможным по причине того, что события поиска не происходит. Быстрый поиск отправляет REST запросы к системе, тем самым используя иной функционал. Поэтому для перехвата таких событий потребовалось написание функции-слушателя на JavaScript. Она обрабатывает событие перехода пользователя на страницу из блока быстрого поиска, одновременно с этим записывая всю необходимую статистическую информацию. Код данной функции представлен ниже:

```
AJS.toInit(function () {
jQuery(document).ready(function ($) {
//Обработчик события клика по элементу на странице
$("form#quick-search").delegate("div.aui-dropdown li a", "mousedown", function ()
{
   var $anchorLink = $(this);
   var $anchorLinkLi = $anchorLink.parent();
//получение имени пользователя через Confluence API
   var user = AJS.Data.get("current-user-fullname");
   var time = new Date().getTime();
   var query = $anchorLink.closest("form#quick-search").find("input#quick-search-
query").val();
   var href = $anchorLink.attr('href');
   var cssClasses = $anchorLink.attr('class');
   if (cssClasses.indexOf("search-for") == -1) {
   var data = {
   link: href,
   timeStamp: time,
   query: query,
   userName: user
   };
//Вызов функции асинхронной передачи данных
   send(data);
   }
});
var restPath = AJS.Data.get("context-path") +
"/rest/statistics/1.0/service/GOTOLinkEvent";
function send(data) {
   AJS.log("Your search statistics are being posted. Confluence will not record them
if you have set your preferences to disable them. ", data);
```

```
//Асинхронный POST запрос, данные передаются в формате JSON
   $.ajax({
   url: restPath,
   type: "POST",
   data: JSON.stringify(data),
   dataType: "json",
   contentType: "application/json; charset=utf-8",
   success: function (data) {
      // Do nothing
   },
   error: function (xhr, textStatus, errorThrown) {
//Вывод на консоль сообщения об ошибке в случае его появления
console.error("Error when submitting the search", xhr, textStatus, errorThrown);
   }
   });
}
});
});
```

Данный код собирает данные о пользователе, дате запроса, теле запроса и ссылке, по которой перешел пользователь и отправляет их Ajax запросом на адрес /rest/statistics/1.0/service/GOTOLinkEvent. Ajax обеспечивает незаметность сбора статистики для пользователя, так как передача данных осуществляется асинхронно.

Продемонстрируем работу системы по формированию статистических отчетов о работе пользователей с информационными ресурсами. График статистики по использованию ресурсов представлен на рисунке 3.5.

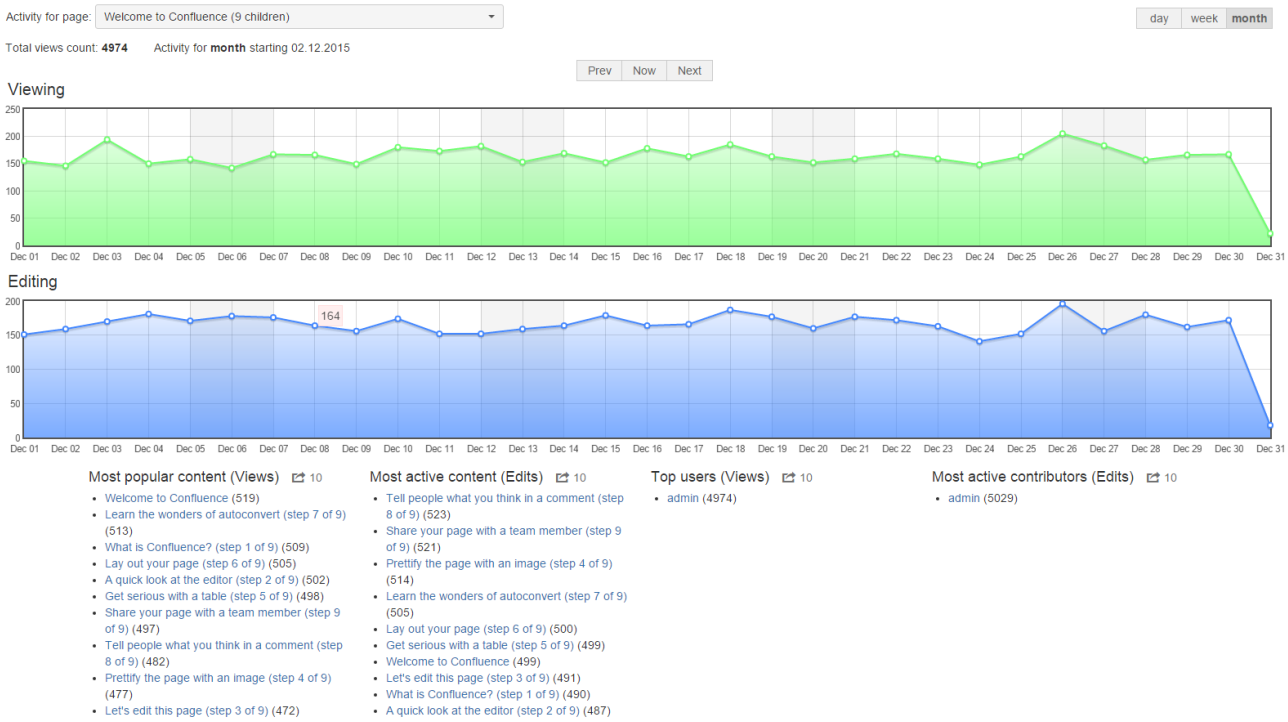


Рисунок 3.5 – Статистика использования ресурсов

Рисунок, представленный выше, можно разделить на несколько основных блоков. Посередине экрана представлены два графика. Они отражают статистику просмотров и редактирований информационных ресурсов согласно запросу. В верхней части отчета находится навигационный блок, который позволяет регулировать даты и период отчетности, а также выбирать то, для какой страницы будет представлен отчет. В нижней части экрана представлены 4 блока. На двух из них расположена статистика по популярным ресурсам в ключе просмотров и редактирований. Еще два отчета отображают активных пользователей и редакторов контента. Эта статистика позволит определить администратору портала наиболее востребованный ресурс и актуализировать его по мере необходимости.

Вторым используемым отчетом является статистика поисковых запросов. Она представлена на рисунке 3.6.

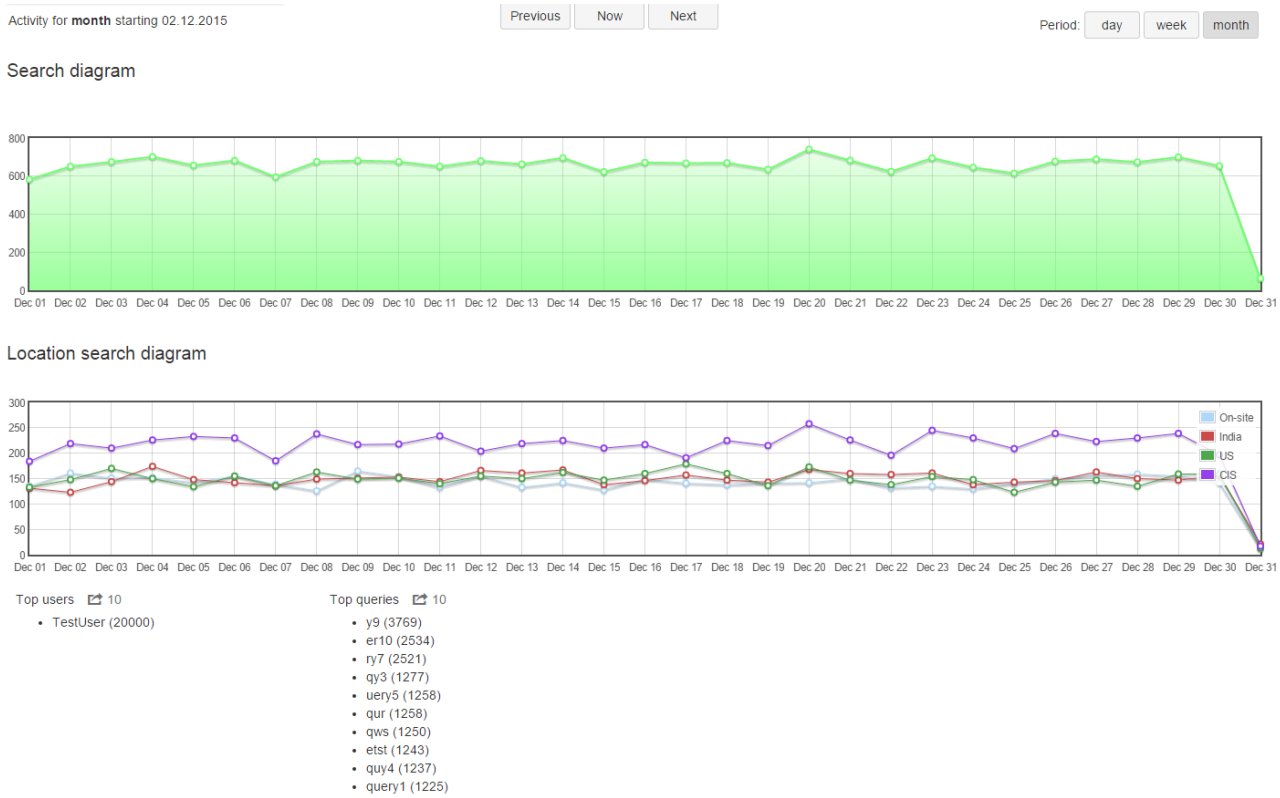


Рисунок 3.6 – Статистика поисковых запросов

Отчет по использованию поиска похож на отчет по статистике использования ресурсов в отношении элементов управления. Он также позволяет просмотреть статистику за определенный период. На верхнем графике отражается общее число поисковых запросов за запрашиваемую дату. Нижний график также отображает число поисковых запросов, но уже с делением на географические зоны. Это позволяет определить активность пользователей из разных регионов в разные моменты времени.

Два нижних отчета предоставляют информацию о наиболее часто запрашиваемых выражениях и о наиболее активном пользователе поисковой машины. Часто запрашиваемые выражения служат хорошим маркером для определения потребностей пользователей. Код отчета по географической активности пользователей представлен ниже:

```
public List<CitySearchModel> getReport(String offset) {
    long startTime = filter.getStartTime();
//Формирование временного массива
    List<SearchEntity> entities = getStatsEntities("");
```

```
    List<CitySearchModel> result = new ArrayList<>();
//Формирование объекта-фильтра по датам отчета
    DatesPeriod datesPeriod = DatesPeriod.valueOf(filter.getPeriod().toUpperCase());
    DateTime dateTime = new DateTime(startTime);
    List<String> cities = new ArrayList<>();
//Формирование массива городов, существующих в базе
    for (SearchEntity searchEntity : entities) {
        if (StringUtils.isNotBlank(searchEntity.getLocation()))
            cities.add(searchEntity.getLocation());
    }
    Set<String> uniqueCities = new HashSet<>(cities);
    uniqueCities.removeAll(CISList);
    uniqueCities.removeAll(USList);
    uniqueCities.removeAll(Indian);
//Заполнение массива данными по географической активности
    result.add(getTempResult(uniqueCities, "On-site", 1, datesPeriod, dateTime,
offset));
    result.add(getTempResult(Indian, "India", 2, datesPeriod, dateTime, offset));
    result.add(getTempResult(USList, "US", 3, datesPeriod, dateTime, offset));
    result.add(getTempResult(CISList, "CIS", 4, datesPeriod, dateTime, offset));
    return result;
}
//Метод, подготавливающий модель данных, которая в дальнейшем будет
передана пользователю
private CitySearchModel getTempResult(Set<String> uniqueCities, String listName,
int color, DatesPeriod datesPeriod, DateTime dateTime, String offset) {
    List<DateCount> tempResult;
    List<SearchEntity> entityList = new ArrayList<>();
    for (String uniqueCity : uniqueCities) {
        entityList.addAll(getStatsEntities(uniqueCity));
```

```
    }
//Блок, формирующий выборку по заданным датам
    switch (datesPeriod) {
        case WEEK:
            tempResult = DateUtils.populateOnlyDays(entityList, 7, dateTime, offset);
            break;
        case DAY:
            tempResult = DateUtils.populateWithHours(entityList, dateTime, offset);
            break;
        case MONTH:
        default:
            int daysInMonth = dateTime.dayOfMonth().getMaximumValue();
            tempResult = DateUtils.populateOnlyDays(entityList, daysInMonth,
dateTime, offset);
            break;
    }
    return new CitySearchModel(listName, color, tempResult);
}
```

Для наглядного отображения логики работы данного модуля, построим диаграмму активности. Она представлена на рисунке 3.7. На ней видно, что сначала формируется временный массив, на основе которого потом осуществляется работа данного модуля. После этого создается объект-фильтр по датам и формируется список городов. Далее – начинается обработка базы данных в цикле и помещение обработанных данных в отчет.

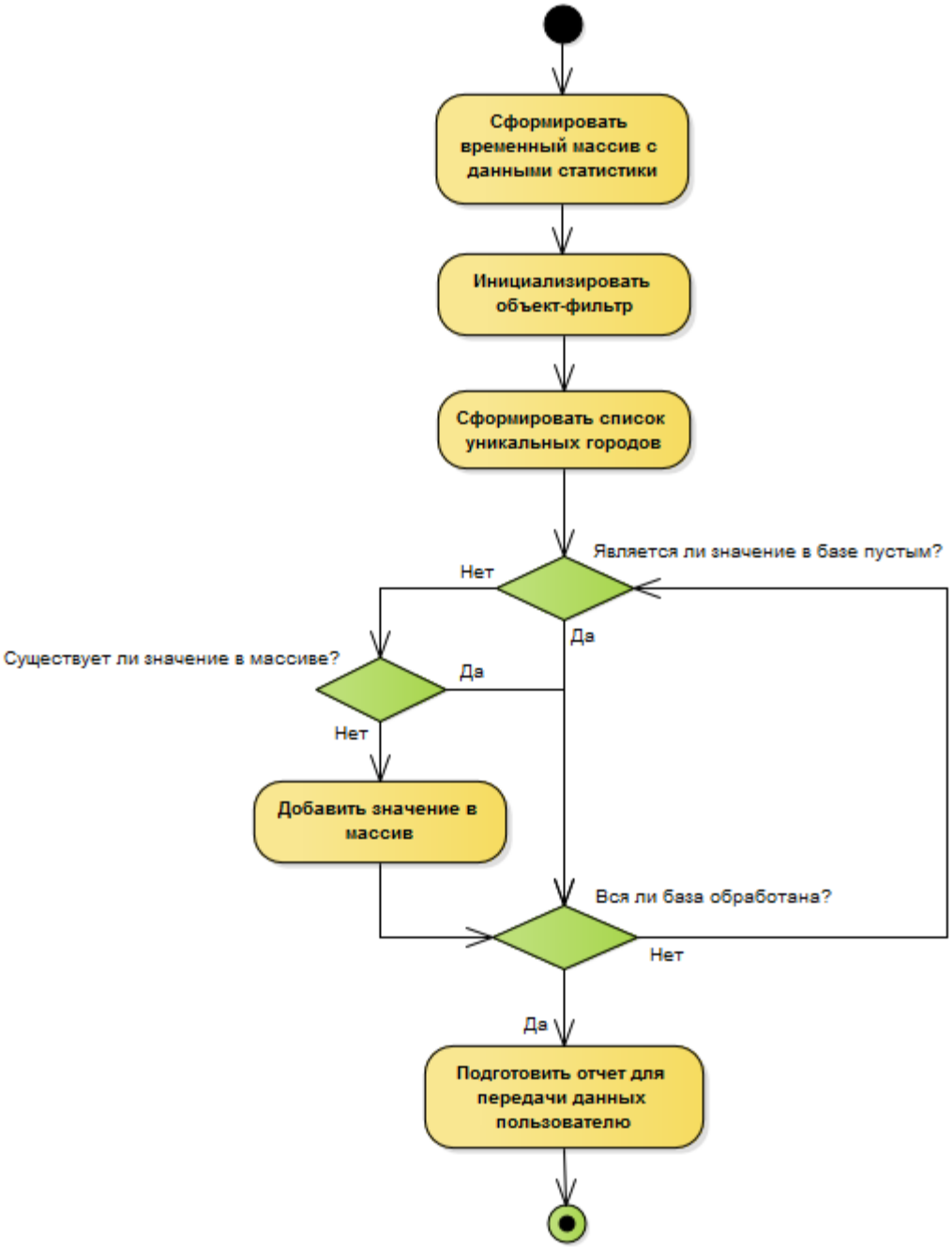


Рисунок 3.7 – Диаграмма деятельности (модуль построения отчетов)

Еще одним элементом модуля отчетности является форма выгрузки статистики в Microsoft Excel. Данный функционал реализован с целью расширения возможностей отчетов, а также для архивации данных. Форма выгрузки представлена на рисунке 3.8. На ней видно, что возможна выгрузка не только базы целиком, но и с наложением некоторых фильтров по дате, пользовательскому запросу и имени пользователя. Также, стоит обратить

внимание на то, что форма позволяет очистить базу данных после выгрузки ее в Excel.

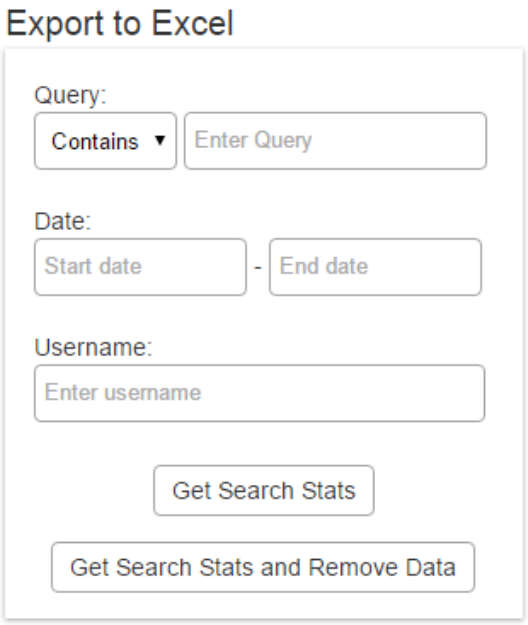


Рисунок 3.8 – Форма выгрузки в Excel

На рисунке 3.9 представлен образец рабочей книги Microsoft Excel с выгруженными в нее статистическими данными.

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Query | Date | Time | Username | Link | Department | Location |
| 2 | y9 | 23/05/2016 | 04:04:48 | TestUser | | | Australia |
| 3 | y9 | 03/05/2016 | 20:34:25 | TestUser | | | Cincinnati |
| 4 | y9 | 31/05/2016 | 13:57:47 | TestUser | | | Samara |
| 5 | Qur | 19/05/2016 | 17:09:11 | TestUser | | | Cincinnati |
| 6 | querY1 | 29/05/2016 | 11:49:31 | TestUser | | | Australia |
| 7 | Quy4 | 22/05/2016 | 02:20:22 | TestUser | | | Hyderabad |
| 8 | Quy4 | 23/05/2016 | 09:28:23 | TestUser | | | Hyderabad |
| 9 | y9 | 09/05/2016 | 22:01:25 | TestUser | | | Samara |
| 10 | Quy4 | 10/05/2016 | 00:29:44 | TestUser | | | Andorra |
| 11 | uery2 | 19/05/2016 | 03:07:16 | TestUser | | | Australia |
| 12 | uery5 | 06/05/2016 | 15:45:52 | TestUser | | | Waltham |
| 13 | er10 | 23/05/2016 | 12:29:49 | TestUser | | | Togliatti |
| 14 | Qy3 | 25/05/2016 | 23:27:42 | TestUser | | | Samara |
| 15 | Quy4 | 03/05/2016 | 15:20:25 | TestUser | | | Cincinnati |
| 16 | Quy4 | 30/05/2016 | 08:25:22 | TestUser | | | Cincinnati |
| 17 | querY1 | 04/05/2016 | 14:09:24 | TestUser | | | Cincinnati |

Рисунок 3.9 – Образец выгрузки данных в Microsoft Excel

На рисунке 3.10 представлена диаграмма деятельности, описывающая алгоритм работы функциональности, отвечающей за выгрузку статистики в Excel.

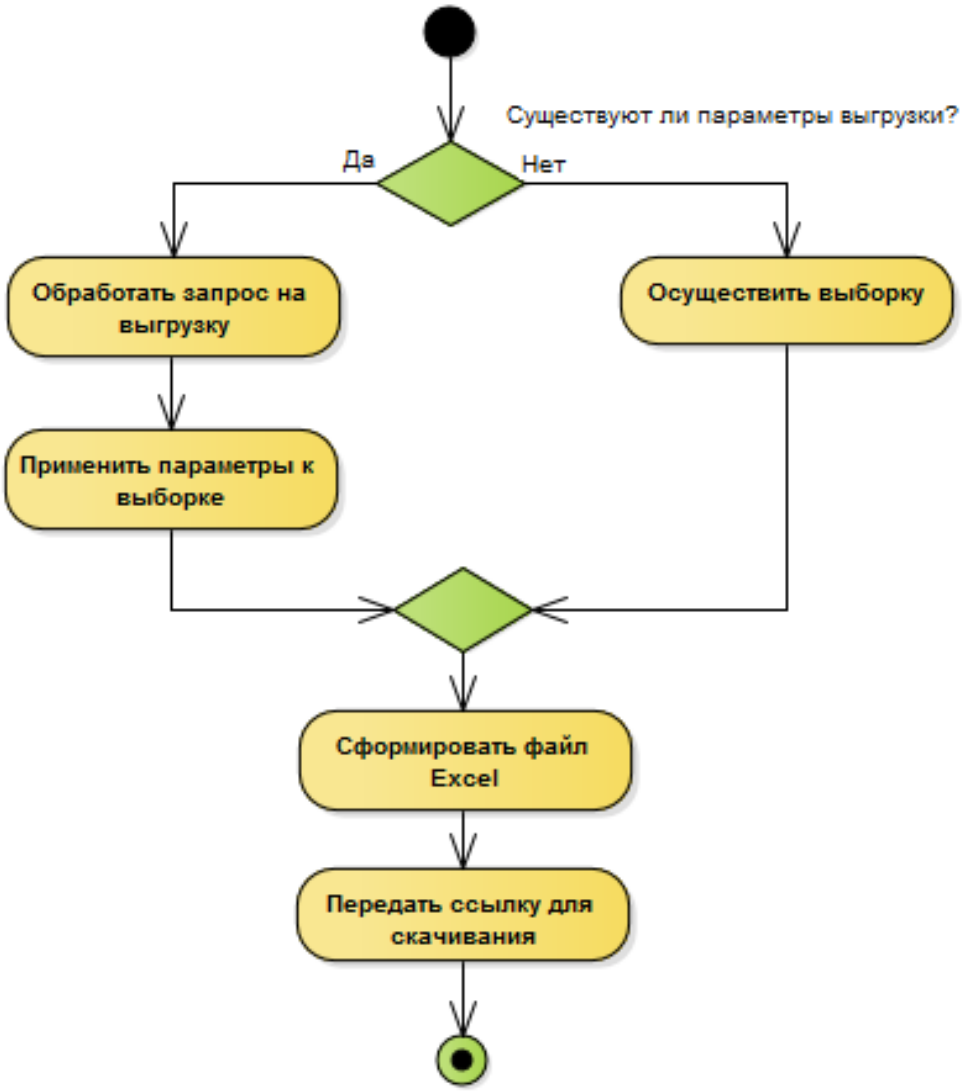


Рисунок 3.10 – Диаграмма деятельности (Выгрузка в Excel)

На диаграмме хорошо видно, что в зависимости от того, используются ли параметры для выгрузки данных, происходит выполнение различного кода.

Кроме того, система мониторинга интегрирована с внешней системой хранения документации, реализованной на платформе Microsoft Sharepoint. Интеграция выражается в возможности автоматической выгрузки статистических отчетов по заданному расписанию на внешний сервер. Загрузка реализована путем формирования PUT http запроса и побайтовой передаче файла. В ходе реализации данной функции возникла проблема, связанная с SSL

авторизацией. Система мониторинга запросов к информационным ресурсам не имела возможности проверить подлинность сертификата внешней системы, что прерывало передачу файла сразу после попытки HTTP подключения. Решением стало создание класса, который бы подтверждал правильность передаваемых HTTPS заголовков.

Так как основной задачей мониторинга запросов является повышение актуальности информационных ресурсов, то на следующих рисунках воспроизведена работа поискового движка до актуализации данных и после нее.

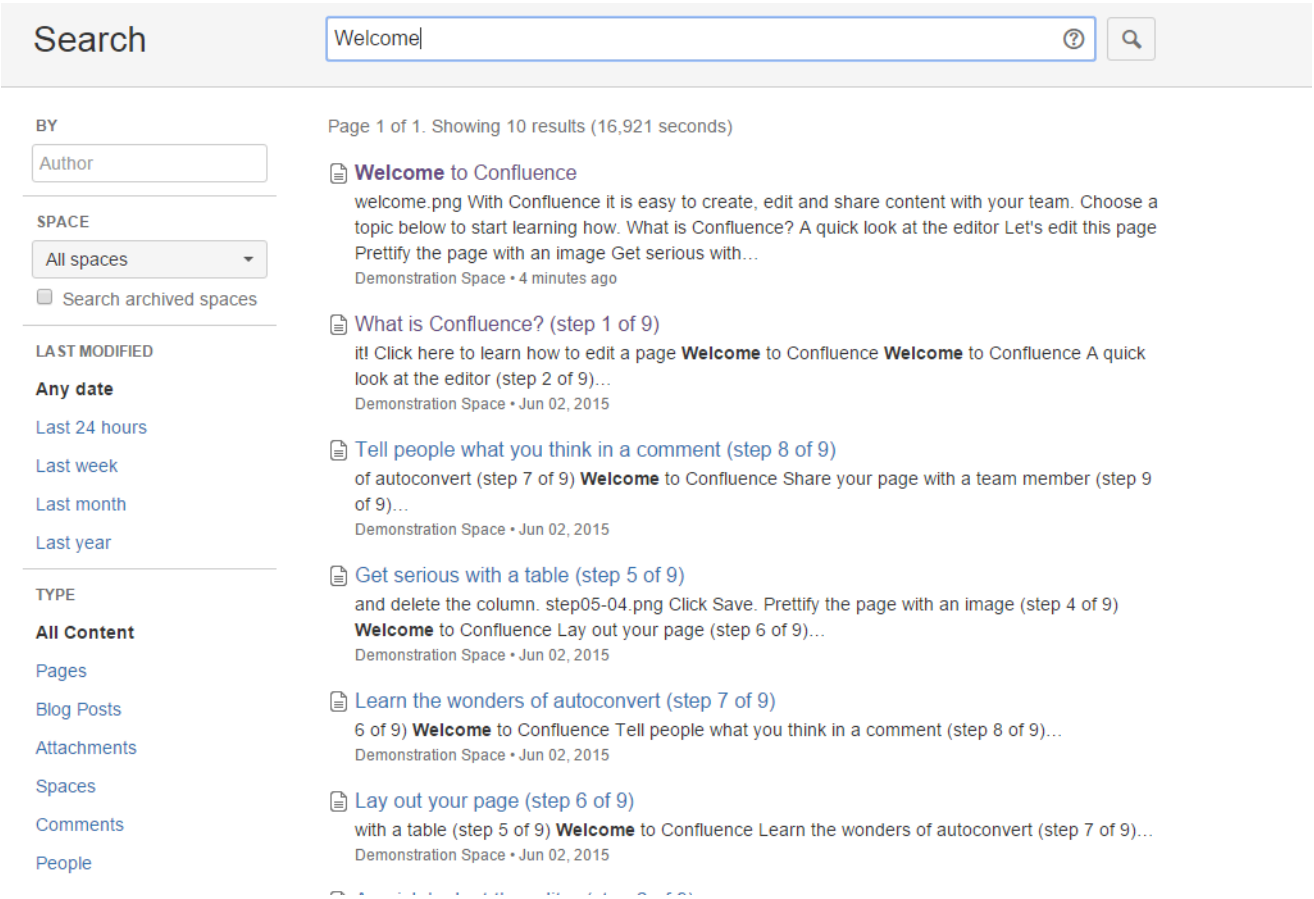


Рисунок 3.11 – Форма поиска до актуализации

На рисунке 3.11 видно, что наиболее релевантным ресурсом, согласно запросу пользователя, является ресурс «Welcome to Confluence». Причиной этого является то, что имеется сходство между заголовком ресурса и искомым запросом. Негативная сторона данного функционала заключается в том, что в нем полностью отсутствует учет статистических данных по использованию

портала корпоративных знаний. Код функции изменения коэффициентов представлен ниже:

```
public float boost(IndexReader indexReader, int i, float v) throws IOException {
   try {
//Получение списка полей у документа
      List abs = indexReader.document(i).getFields();
//Извлечение необходимого коэффициента из поля «views»
      float coeff =
Float.parseFloat(indexReader.document(i).getField("views").stringValue());
//Расчет даты создания страницы
      Date creationDate =
LuceneUtils.stringToDate(indexReader.document(i).getField("created").stringValue()
);
      Instant instant = Instant.ofEpochMilli(creationDate.getTime());
      LocalDateTime localDateTime = LocalDateTime.ofInstant(instant,
ZoneId.systemDefault());
      LocalDateTime currentDate = LocalDateTime.now();
//Проверка даты создания страницы
      if (currentDate.getDayOfYear()-localDateTime.getDayOfYear()<7) v *= 0.5;
      else v *= coeff;
   } catch (NullPointerException |NumberFormatException ex) {
   }
   return v;
}
```

Для подробного рассмотрения алгоритма актуализации рассмотрим диаграмму деятельности модуля актуализации. Она представлена на рисунке 3.12.

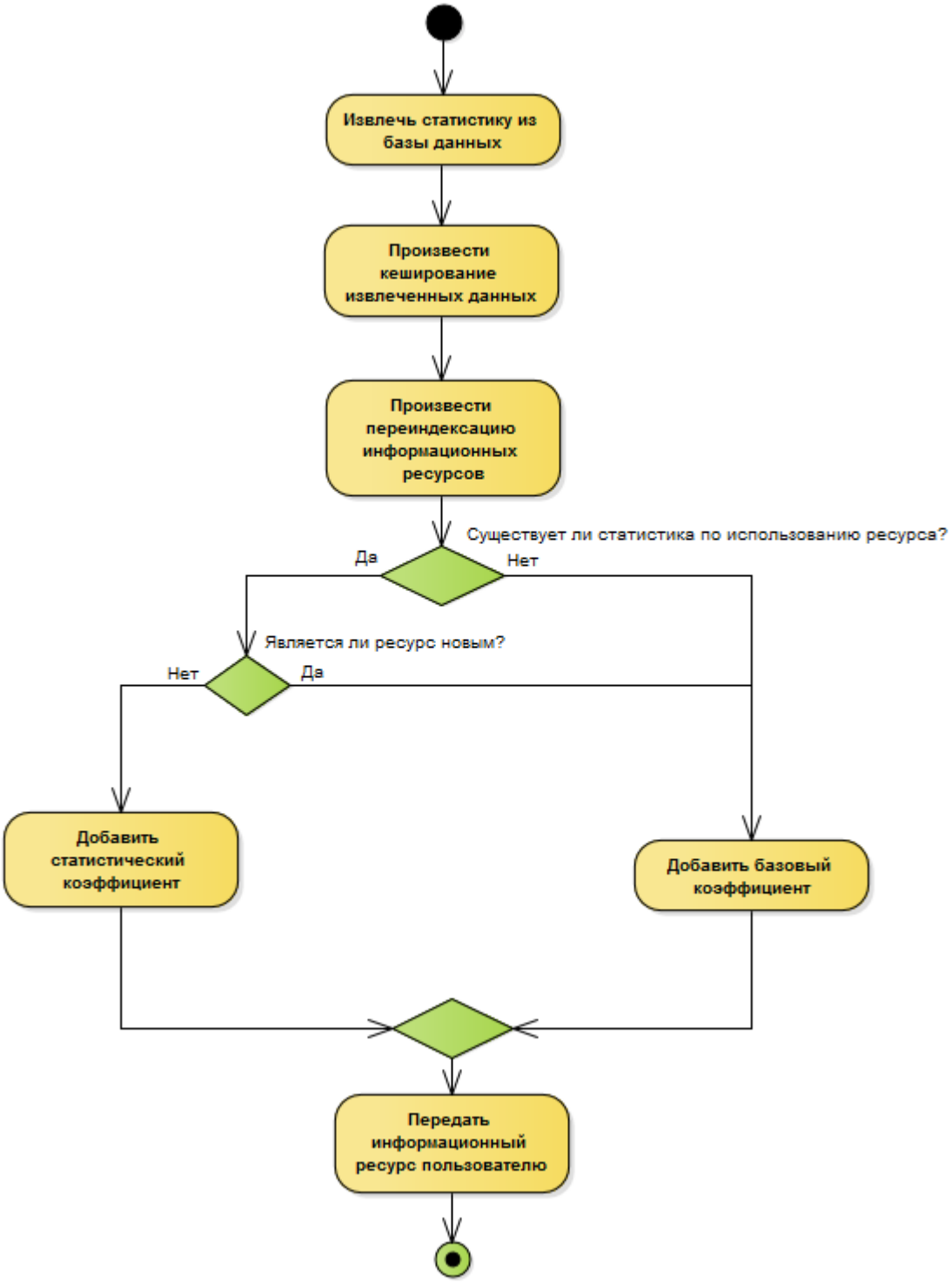


Рисунок 3.12 – Диаграмма деятельности (Модуль актуализации)

На диаграмме видно, что при актуализации данных используется кеширование, которое позволяет сократить число обращений к базе данных и снизить на нее нагрузку. В том случае, если информационный ресурс является новым, для него применяется базовый коэффициент в течение недели. Это необходимо, чтобы ресурс не затерялся в огромной базе знаний. Для тех

ресурсов, которые не могут иметь статистику просмотров (объекты пользователей, файлы приложений, и т.д.), также применяется базовый коэффициент, чтобы они также находились достаточно высоко в поисковой выдаче.

На рисунке 3.13 представлена картина поискового запроса после работы функционала, обеспечивающего актуализацию данных.

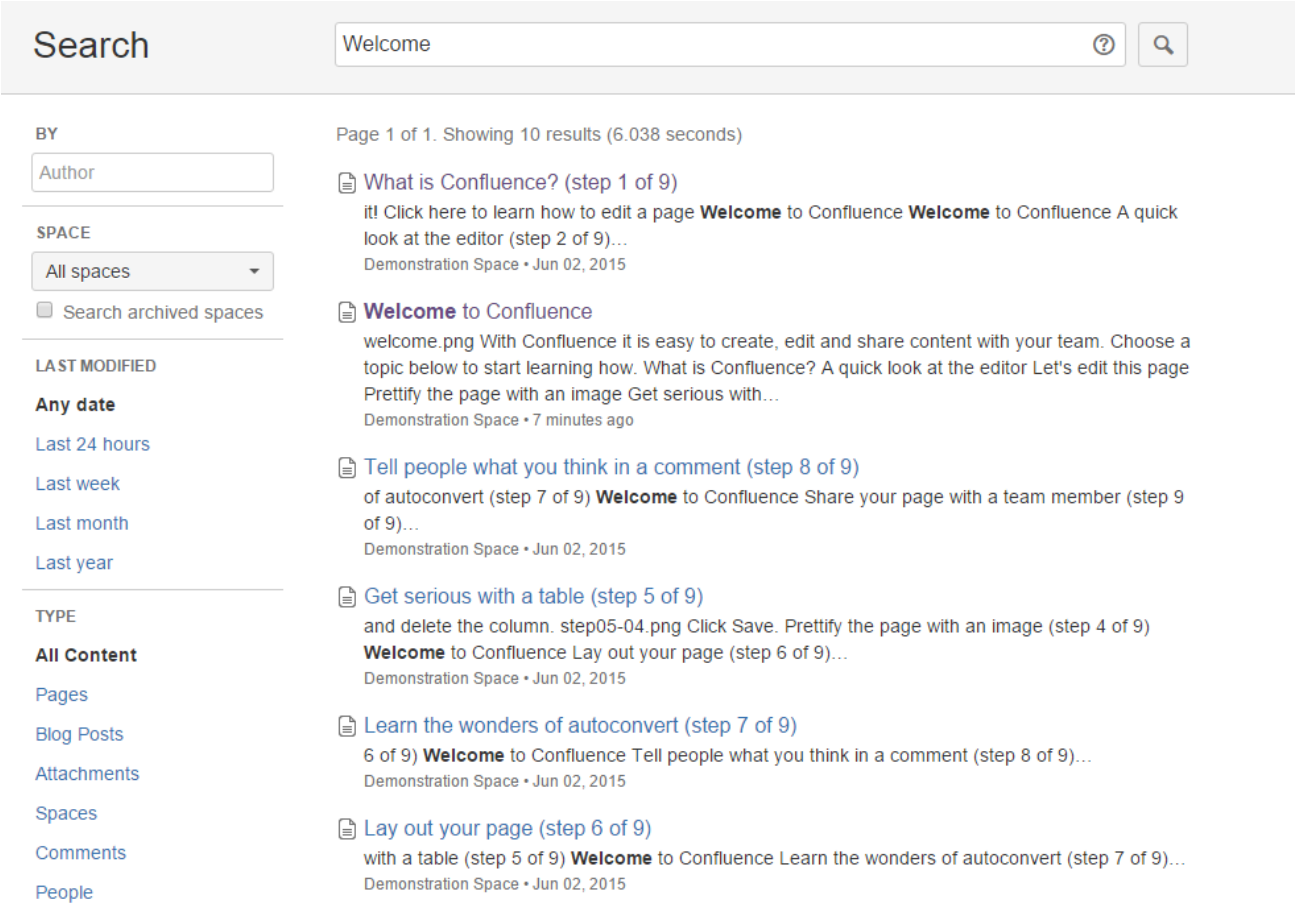


Рисунок 3.13 – Форма поиска после актуализации

Хорошо видно, что ресурс «What is Confluence?» вышел на первое место в результатах поисковой машины. Таким образом, видно, что механизм актуализации ресурсов выполняет свои функции. Для подтверждения данного вывода можно воспользоваться функцией Atlassian Confluence – explain.action, предоставляющей полный план расчета поисковых весов документов. В таблице 3.1 отображены веса документов до и после внедрения системы по поисковому запросу «Welcome».

Таблица 3.1 – Весовые коэффициенты документов при поиске

| Название статьи | Коэффициент до внедрения системы | Коэффициент после внедрения системы |
|---|---|---|
| (1) Official University Welcome | 5,6847653 | 7,4830074 |
| (2) Melbourne welcome | 5,644657 | 9,791752 |
| (3) Welcome Film | 4,1407576 | 5,8731155 |
| (4) Welcome to Confluence | 2,8985302 | 3,6379511 |
| (5) Holiday WSU | 1,6894137 | 1,7201575 |
| (6) WA Ambassador | 1,3360876 | 1,3093659 |
| (7) What is Confluence? (step 1 of 9) | 1,0527294 | 1,0742136 |
| (8) Tell people what you think in a comment (step 8 of 9) | 0,91655606 | 1,571239 |
| (9) Get serious with a table (step 5 of 9) | 0,91655606 | 0,9048755 |
| (10) Learn the wonders of auto convert (step 7 of 9) | 0,8657623 | 1,7315246 |

Рассмотрев эту сводную таблицу, можно увидеть, что позиции многих информационных ресурсов претерпели сильные изменения. Так, например, ресурс «Melbourne welcome» получил рейтинг 9,79 и находится на первом месте в поисковой выдаче. Вес некоторых ресурсов снизился, что свидетельствует об их низкой популярности среди пользователей. Исходя из вышесказанного, можно заключить вывод о том, что модуль актуализации запросов работает так, как и было запланировано.

## Вывод по главе

Результатом данной главы стало построение моделей данных системы мониторинга запросов. Анализ моделей данных в совокупности с представленными ранее концептуальной и логической моделями системы

позволил перейти к реализации программного продукта. Разработанное приложение предоставило возможность администраторам портала корпоративных знаний изучить статистику использования ресурсов и узнать потребности пользователей портала. Эти данные являются одними из основополагающих для процесса актуализации ресурсов и развития портала. Также, система в некоторой степени осуществляет автоматизацию процесса актуализации. Это происходит при помощи API, предоставляемого поисковым движком Apache Lucene. Для оценки эффективности информационной системы необходимо произвести экспериментальную апробацию, которая позволит выявить основные недостатки и исправить их перед вводом в промышленную эксплуатацию.

# ГЛАВА 4 Экспериментальная апробация и внедрение системы мониторинга запросов к информационным ресурсам портала корпоративных знаний

## 4.1 Процесс внедрения системы мониторинга запросов к информационным ресурсам портала корпоративных знаний

Процесс внедрения системы мониторинга запросов к информационным ресурсам в компании ООО «НетКрекер» предусматривает установку программного обеспечения на два окружения. Первое – стенд для тестирования функционала и демонстрации его возможностей заказчику системы. Второе подразумевает непосредственно ввод в промышленную эксплуатацию.

Первоначально при установке приложения на первый стенд – было проведено функциональное тестирование приложения. Оно проводилось методом черного ящика, т.к. доступа к коду у сотрудника, проводящего тестирование не было. Целью тестирования в данном случае ставится выяснение обстоятельств, в которых поведение программы не соответствует спецификации. Для обнаружения всех ошибок в программе необходимо выполнить исчерпывающее тестирование, то есть тестирование на всевозможных наборах данных. Для большинства программ такое невозможно, поэтому применяют разумное тестирование, при котором тестирование программы ограничивается небольшим подмножеством всевозможных наборов данных. При этом необходимо выбирать наиболее подходящие подмножества, подмножества с наивысшей вероятностью обнаружения ошибок. При таком подходе к тестированию приложения было выявлено несколько недостатков, которые, впрочем, были исправлены в дальнейшем. Одним из таких недостатков стала некорректная работа функционала отчетов в отношении отображения дат. Причиной этого было отсутствие поддержки различных часовых поясов. Еще одной крупной проблемой стала выгрузка отчетов в Microsoft Excel. При клике администратора на кнопку выгрузки отчета несколько раз, появлялась ошибка java.lang.OutOfMemoryError, которая

свидетельствовала о нехватке памяти на сервере. При запуске функционала отчетов в памяти формировался объемный файл Excel, который не выгружался из нее до тех пор, пока не закончится его формирование. Решением данной проблемы стало отключение кнопки выгрузки отчета после первого клика на нее.

Также при демонстрации возможностей приложения было проведено нагрузочное тестирование, позволившее выявить некоторые недочеты в алгоритмах актуализации ресурсов. В частности, изначально было предложено использовать индивидуальный индекс для каждого пользователя, что позволило бы сделать поисковую выдачу для каждого пользователя уникальной. Но при данном подходе было выявлено несколько минусов, такие как многократное увеличение объема поискового индекса, повышение временных и ресурсных затрат сервера на процесс индексации. Это привело к следующей итерации внедрения. От индивидуальных индексов было решено отказаться, и был использован общий индекс для всех пользователей на основе количества просмотров указанной страницы за определенный промежуток времени. Это привело к улучшению производительности системы, но исключило из расчета индексов объекты корпоративного портала, не являющиеся страницами. За исправлением этой ошибки последовала модификация коэффициентов для только что созданных статей, т.к. они выпадали из поисковой выдачи из-за отсутствия большого числа просмотров, несмотря на то, что содержали релевантный контент. В такой версии система была введена в эксплуатацию на промышленном сервере.

## 4.2 Оценка результатов апробации системы мониторинга запросов к информационным ресурсам

Для оценки эффективности работы модуля актуализации был проведен опрос среди сотрудников компании, активно использующих поисковый модуль портала корпоративных знаний. Пользователям был задан вопрос: «Является ли

релевантной поисковая выдача по вашему поисковому запросу?». Результаты опроса до и после актуализации были сведены в суммарную таблицу.

Таблица 4.1 – Результаты опроса пользователей

| | | Второй опрос | | Сумма |
|---|---|---|---|---|
| | | + | - | |
| Первый опрос | + | 2 | 6 | 8 |
| | - | 20 | 12 | 32 |
| | Сумма | 22 | 18 | 40 |

Для анализа полученных результатов опроса удобнее всего будет использовать критерий Макнамары в связи с тем, что он наиболее приспособлен к анализу данных, представленных в дихотомической шкале. Этот критерий предназначен для сравнения распределений объектов двух совокупностей по состоянию некоторого свойства на основе измерений этого свойства в двух зависимых выборках из рассматриваемых совокупностей [17].

Зададим две гипотезы: $H_1$ – о наличии существенных различий между результатами первого и второго опроса пользователей и $H_0$ – об отсутствии различий между результатами опросов.

Определим A=2, B=6, C=20 и D=12

где A – число участников опроса, которые до и после актуализации дали положительный ответ на поставленный вопрос,

B – число участников опроса, которые дали положительный ответ до актуализации, но отрицательный после,

C – число участников опроса, которые дали отрицательный ответ до актуализации, а после – положительный,

D – число участников опроса, которые до и после актуализации дали отрицательный ответ на поставленный вопрос.

Для расчета $M_{эмп}$ существуют два способа. В данном случае сумма значений B и C – больше 20, поэтому $M_{эмп}$ рассчитывается по формуле:

$$M_{эмп} = \frac{(B-C)^2}{B+C} \quad (4.1)$$

При имеющихся данных $M_{эмп}$ = 7,538.

Возьмем критические значения M из таблицы критических значений критерия Макнамары для уровней статистической значимости p<= 0.05 и p<=0.01. Уровень значимости – это вероятность того, что различия являются существенными, в то время как они на самом деле случайны.

Обычно в прикладной статистике используют 2 уровня значимости.

1. 1-й уровень значимости: $p \leq 0{,}05$.

Это 5% уровень значимости. До 5% составляет вероятность того, что был сделан ошибочный вывод о том, что различия достоверны, в то время как они на самом деле являются недостоверными. Можно сказать и по-другому: существует лишь 95% уверенность в том, что различия действительно достоверны.

2. 2-й уровень значимости: $p \leq 0{,}01$.

Это 1% уровень значимости. Вероятность ошибочного вывода о том, что различия достоверны, составляет не более 1%. Иначе – есть 99% уверенность в том, что различия действительно достоверны. В данном случае можно написать и так: P>0,99.

Критические значения при расчете критерия данным способом для первого и второго уровней значимости всегда постоянны и равны:

$$M_{кр} = \begin{cases} 3{,}841, для\ P \leq 0{,}05 \\ 6{,}635, для\ P \leq 0{,}01 \end{cases} \quad (4.2)$$

Построим ось значимости, изображенную на рисунке 4.1.

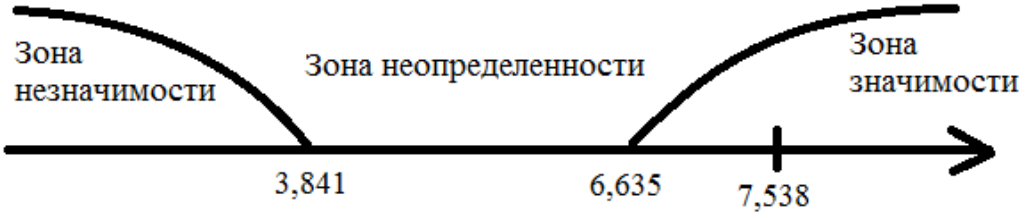


Рисунок 4.1 – Ось значимости

Полученное значение $М_{эмп}$ попало в зону значимости. Отсюда следует принять гипотезу $Н_1$ о наличии различий. Если при этом посмотреть на результаты опроса, то можно увидеть, что большинство из контрольной группы, опрошенных при первом опросе, ответили отрицательно, а при втором – положительно. В совокупности с принятой гипотезой можно говорить о том, что для опрошенной группы пользователей работа системы мониторинга была эффективной, и они получили более релевантные результаты после актуализации информационных ресурсов.

Была проведена еще одна оценка эффективности внедрения системы мониторинга запросов к информационным ресурсам портала корпоративных знаний. Был рассчитан объем времени, которое тратил администратор портала корпоративных знаний на получение статистических отчетов и последующую актуализацию информационных ресурсов. Для выполнения этой деятельности ему требовалось вручную выполнять SQL запросы к порталу, переносить данные в Microsoft Excel и строить необходимые отчеты. Минусом данного подхода можно считать то, что запросы осуществлялись к таблице Content, которая является основной в платформе Atlassian Confluence. Такие запросы могли вносить излишнюю нагрузку на сервер базы данных и как следствие нарушать нормальную работу с порталом знаний.

В таблице 4.2 представлены основные временные затраты за месяц опытного сотрудника портала на мониторинг и ручную актуализацию ресурсов.

Таблица 4.2 – Временные затраты на мониторинг

| Этап выполнения | Время выполнения, час |
|---|---|
| Написание и выполнение SQL запросов | 0,5 |
| Формирование рабочей книги Excel | 0,5 |
| Построение графиков и отчетов | 1 |
| Актуализация информационных ресурсов | 16 |
| Итого: | 18 |

Несмотря на то, что сотрудник портала мог выполнять некоторую работу по сбору статистики, проанализировать использование ресурсов было невозможно в полном объеме, т.к. отсутствовали данные по поисковой статистике и статистике просмотра ресурсов. Кроме того, события редактирования администратор мог отслеживать лишь косвенно по версиям информационных ресурсов. В результате внедрения системы мониторинга, сотрудник, работающий с порталом, получил исчерпывающую статистику об использовании информационных ресурсов, а также сократил время на получение статистической информации до 5 минут, т.к. вся необходимая информация уже представлена на выделенной странице портала. Причем у сотрудника осталась возможность строить персонифицированные отчеты из Microsoft Excel. Кроме того, временные затраты на процесс актуализации информационных ресурсов портала начали снижаться. Сводная информация представлена на рисунке 4.2.

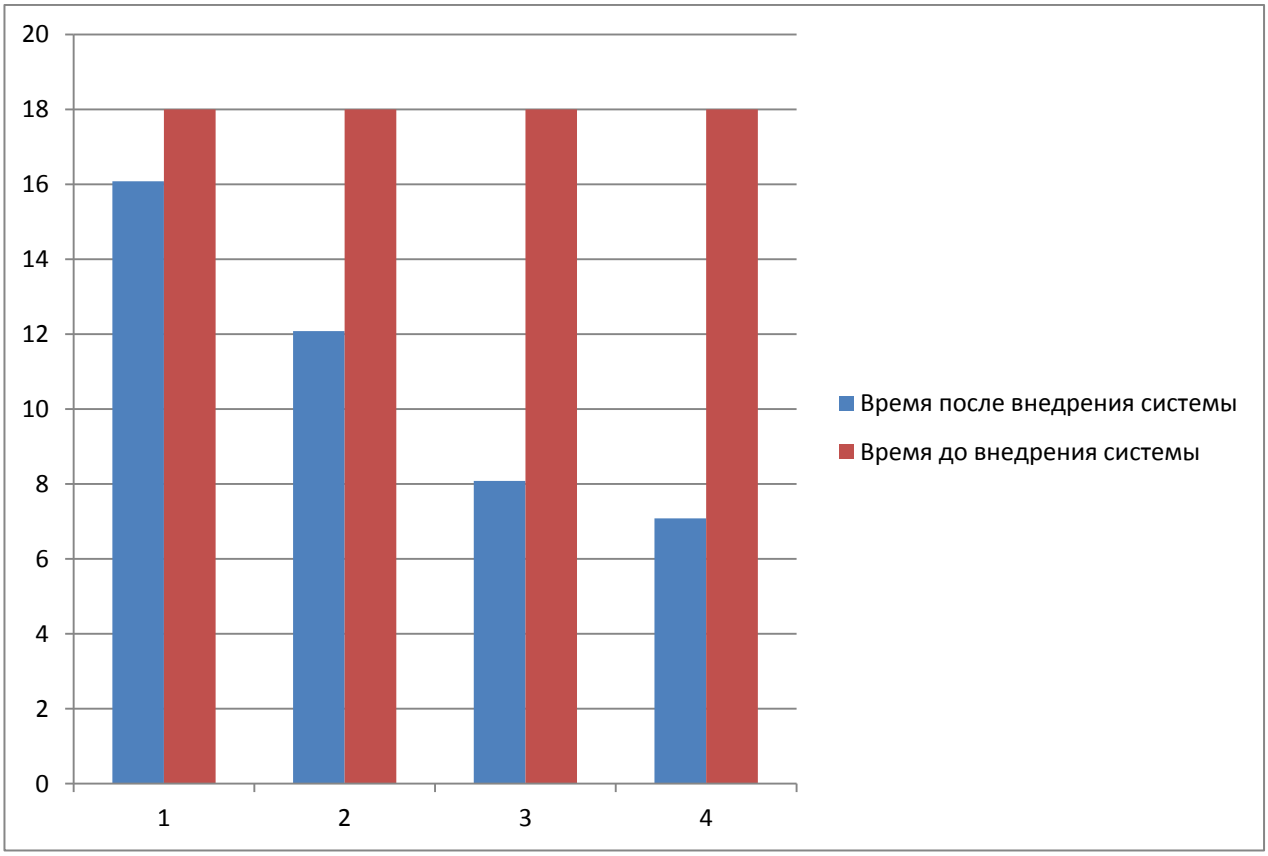


Рисунок 4.2 – График временных затрат

На диаграмме видно, что система мониторинга запросов к информационным ресурсам портала корпоративных знаний уже в первый месяц использования позволила сократить временные затраты на 2 часа. А к 4 месяцу использования системы, время на информационный мониторинг и актуализацию ресурсов сократилось более чем вдвое. Причиной этого стала работа системы мониторинга по актуализации поисковой выдачи. Это свидетельствует о высокой полезности системы для предприятия.

Еще одним способом оценки эффективности стало сравнение среднего времени, затрачиваемого пользователями на поиск необходимого информационного ресурса. Данные об этом сведены в таблицу 4.3.

Таблица 4.3 – Временные затраты сотрудников на поиск ресурсов

| | Среднее время на поиск до внедрения системы мониторинга, сек. | Среднее время на поиск после внедрения системы мониторинга, сек. |
|---|---|---|
| Сотрудник 1 | 15,7 | 13,2 |
| Сотрудник 2 | 20,9 | 20,1 |
| Сотрудник 3 | 18,1 | 17,6 |
| Сотрудник 4 | 20,4 | 16,8 |
| Сотрудник 5 | 25,4 | 22,8 |
| Сотрудник 6 | 17,1 | 17 |
| Сотрудник 7 | 22,4 | 21,6 |
| Сотрудник 8 | 15,6 | 14,9 |
| Сотрудник 9 | 17,9 | 17,5 |
| Сотрудник 10 | 19,2 | 18,5 |

Для анализа средних значений удобнее всего будет воспользоваться пакетом анализа Microsoft Excel. Он предоставляет возможность построения парного t-теста для средних (критерия Стьюдента). Определим гипотезу $H_1$ как «Среднее время, затрачиваемое сотрудником на поиск необходимой информации после внедрения разработанной системы, значительно отличается

от времени, затрачиваемого до внедрения системы». А гипотезу $H_0$ примем как гипотезу об отсутствии различий между двумя сравниваемыми значениями. Результаты выполнения этой функции представлены в таблице 4.4.

Таблица 4.4 – Парный двухвыборочный t-тест для средних

| | Среднее время до внедрения системы | Среднее время после внедрения системы |
|---|---|---|
| Среднее | 19,27 | 18 |
| Дисперсия | 9,497888889 | 8,462222222 |
| Наблюдения | 10 | 10 |
| Корреляция Пирсона | 0,92407588 | |
| Гипотетическая разность средних | 0 | |
| df | 9 | |
| t-статистика | 3,404908675 | |
| P(T<=t) одностороннее | 0,003906173 | |
| t критическое одностороннее | 1,833112933 | |
| P(T<=t) двухстороннее | 0,007812346 | |
| t критическое двухстороннее | 2,262157163 | |

Поскольку p-значение равно 0,004 и меньше $\alpha < 0,05$, нулевую гипотезу $H_0$ следует отклонить. Кроме того, гипотеза подтверждается за счет большого превышения t-статистики над t критическим двухсторонним. Это означает то, что среднее время, затрачиваемое сотрудником на поиск необходимой информации после внедрения разработанной системы, действительно значительно отличается от времени, затрачиваемого до внедрения системы.

Экспериментальная оценка эффективности проводилась по трем критериям: отзывам пользователей по эффективности работы системы мониторинга, по временным затратам сотрудников портала на анализ результатов мониторинга и дальнейшую ручную актуализацию ресурсов, а также по среднему времени, затрачиваемому пользователями портала на поиск необходимой информации.

В результате оценки эффективности внедрения информационной системы практическая реализация подтвердила правильность разработанной

теоретической модели и может быть внедрена в работу как система, позволяющая упростить рабочий процесс и улучшить качество обработки данных.

Таким образом, можно считать внедрение информационной системы мониторинга запросов к информационным ресурсам портала корпоративных знаний успешным.

## Вывод по главе

Результатом данной главы стало описание процесса внедрения системы мониторинга запросов к информационным ресурсам портала корпоративных знаний, основных особенностей, выявленных при передаче системы заказчику. Внедрение системы сопровождалось исправлениями в алгоритме актуализации, т.к. промышленная эксплуатация подразумевает высокую нагрузку, а обеспечить ее при разработке весьма затруднительно. Оценка эффективности данного алгоритма показала, что практическая реализация системы мониторинга подтвердила правильность теоретических выводов, и её внедрение способствовало улучшению качества работы пользователей с порталом корпоративных знаний.

# ЗАКЛЮЧЕНИЕ

В ходе диссертационного исследования была изучена и проанализирована научная литература, которая позволила понять специфику работы корпоративного портала и мониторинга запросов к его информационным ресурсам. Были рассмотрены возможности существующих систем мониторинга – преимущества и недостатки при их внедрении. Исследование этих систем позволило сделать вывод о необходимости разработки нового программного продукта, предоставляющего функции мониторинга, а также сформировать требования к ожидаемой системе.

Следующим шагом стало изучение особенностей платформы, для которой будет осуществляться разработка нового компонента. Исходя из полученной информации, было проведено проектирование системы: определение ее основных процессов, хранилищ данных, формулирование списка инструментов, с помощью которых будет осуществлена реализация, описание основных архитектурных особенностей. Были построены структурные диаграммы IDEF0 и DFD и осуществлен переход к объектной модели и построению диаграмм в нотации UML.

Третьим этапом стала разработка системы мониторинга. Была спроектирована база данных, построены логическая и физическая модели данных. Определена СУБД, используемая при передаче системы в промышленную эксплуатацию. На основе результатов моделирования реализована система мониторинга запросов, упрощающая процесс актуализации ресурсов администратору портала корпоративных знаний.

Финальным шагом диссертационной работы стало внедрение разработанной системы на предприятие и ее апробация. Этап внедрения выявил недоработки в системе, не проявившие себя при разработке и определил финальную конфигурацию механизма актуализации ресурсов. Апробация разработанного продукта показала, что его использование повышает качество работы пользователей с порталом корпоративных знаний.

Результатом исследования стало подтверждение выдвинутой гипотезы, заключающейся в том, что эффективность функционирования портала корпоративных знаний будет выше, если будет производиться постоянная актуализация и повышение качества информационных ресурсов на основе данных мониторинга пользовательских запросов.

Разработанная система полностью соответствует поставленным к ней требованиям и решает конкретные задачи, сформулированные при ее проектировании. Тестирование приложения прошло успешно и его функционал в полной мере используется на предприятии.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

*Нормативно-правовые акты*

1. ГОСТ 19.701-90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения (ИСО 5807-85) . Введ. 1992-01-01. – М. : Изд-во стандартов, 1992. – 14 с. – (Единая система программной документации)

2. ГОСТ 34.320-96. Информационная технология. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы. – Введ. 2001-07-01. – М. : Изд-во стандартов, 2001. – 46 с. – (Основополагающие стандарты)

3. ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание. Общие требования и правила сопоставления. – Введ. 2004-07-01. – М. : Изд-во стандартов, 2004. – 57 с. - (Система стандартов по информации, библиотечному и издательскому делу).

4. ГОСТ 7.32-2001. Отчет о научно-исследовательской работе. Структура и правила оформления. – М. : Изд-во стандартов, 2001. – 22 с. – (Система стандартов по информации, библиотечному и издательскому делу).

5. ГОСТ 7.82-2001. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления. – Введ. 2002-07-01. – Минск : Изд-во стандартов, 2001. – 35 с. – (Система стандартов по информации, библиотечному и издательскому делу).

*Литература*

6. Аверченков В.И., Мирошников В.В., Рощин С.М. Эффективное использование неструктурированной информации в процессе обучения // Новые информационные технологии в университетском образовании. Тезисы междунар. науч.-метод. конф., 6-8 июня 2001 г., – Новосибирск, 2001. – С. 205-206.

7. Аверченков В.И., Рощин С.М. Система формирования знаний // Материалы международной конференции. В 3-х т. Т.1./ ВолгГТУ. – Волгоград, 2004. – С. 10-15.

8. Аверченков, В.И. Мониторинг и системный анализ информации в сети Интернет: монография / В.И. Аверченков, С.М. Рощин. – 2-е изд., стереотип. – М. : ФЛИНТА, 2011. – 160 с.

9. Василенок И.И. Особенности восприятия пользователями электронных ресурсов / И.И. Василенок, Е.Е. Долгополова. – Минск : Национальная библиотека Беларуси, 2010.

10. Вора П. Шаблоны проектирования веб-приложений. – М.: Эксмо, 2011

11. Гасанов Э. Э. Теория хранения и поиска информации, / Э.Э. Гасанов. – М. : Фундамент. и прикл. матем, 2009.

12. Гасанов Э. Э., Кудрявцев В. Б., Теория хранения и поиска информации. – М. : ФИЗМАТЛИТ, 2002.

13. Гасанов Э. Э., Теория сложности информационного поиска, Изд-во механико-математического ф-та МГУ, М., 2005.

14. Гвоздев А.В. Программно-аппаратные решения анализаторов естественного языка для поиска угроз информационной безопасности электронного документооборота // Материалы 4-й научно-практической конференции «Информационная безопасность. Невский диалог — 2012» - «НП-Принт», 2012. - С. 33-35

15. Горохова Е. Социальная технология управления знаниями в многонациональной организации. – М.: Эдитус, 2013.

16. Ермаков Д. Г. Технология информационного мониторинга // Научное сообщество студентов XXI столетия. Тезисы междунар. науч.-практ. конф., 22 октября 2012 г., – Брянск, 2012.

17. Ермолаев О.Ю. Математическая статистика для психологов. Учебник. – М. : Флинта, 2003. – 336 с.

18. Козирацкий Ю. Модели информационного конфликта средств поиска и обнаружения. – М. : Радиотехника, 2013. – 232 с.

19. Коннолли, Т. Б. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. / Пер. с англ. Имамутдинова Р. Г., Птицын К. А.. М.: Вильямс. 2010. - 440с.

20. Кортнев А.В., Логинов В.И. Система мониторинга и анализа научно-исследовательской деятельности // Информ. технологии и вычисл. системы. – 2011.

21. Кудрявцев Д.В. Системы управления знаниями и применение онтологий: Учеб. пособие // Д.В. Кудрявцев. – СПб.: Изд-во Политехн. ун-та, 2010. – 343 с.

22. Ландэ Д. В., Снарский А. А., Безсуднов И. В. Интернетика: Навигация в сложных сетях: модели и алгоритмы. — М.: Либроком (Editorial URSS), 2010. — 264 с.

23. Ларман, К. Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку. – Пер. с англ. Шелестов А. – СПб. : Вильямс, 2013. – 736 с.

24. Максимчук Р.А., Нейбург. Э. Дж. UML для простых смертных. – М. : Лори, 2016. – 270 с.

25. Маннинг К., Рагхаван П., Шютце Х. Введение в информационный поиск. – СПб.: Вильямс, 2011. — ISBN 978-5-8459-1623-5.

26. Мартин Р. Чистый код. Создание, анализ и рефакторинг. Библиотека программиста. – СПб.: Питер, 2016. – 464 с.

27. Могилев А., Листрова Л. Технологии поиска и хранения информации. Технологии автоматизации управления. – СПб. : БХВ-Петербург, 2012. – 320 с.

28. Рощин С. М. Автоматизация мониторинга и системного анализа распределенной проблемно-ориентированной информации в среде Интернет: диссертация / С.М. Рощин. – Брянск: БГТУ, 2005.

29. Севостьянов И. Поисковая оптимизация. – СПб.: Питер, 2010.

30. Уринцов А. Управление знаниями. Теория и практика. Учебник. – М. : Юрайт, 2015. – 256 с.

31. Фримен Э., Фримен Э., Сиерра К. Паттерны проектирования. – СПб.: Питер, 2016. – 656 с.

32. Чернышов В.Н., Чернышов А.В. Теория систем и системный анализ: учебное пособие. - Тамбов : Изд-во Тамб. гос. техн. ун-та, 2009. - 96 с.

33. Шалафеева Е.А. Мониторинг информационных ресурсов жизнеспособной интеллектуальной программной системы // «Программная инженерия» – 2012 – №1, с. 10-15

34. Шокин Ю.И. Проблемы поиска информации / Ю.И. Шокин, А.М. Федотов, В.Б. Барахнин. Новосибирск: Наука, 2010. 220с.

35. Эккель, Б. Философия Java. – 4-е изд., – СПб. : Питер, 2016. 1168 с.

36. Ершеева Р. М. Обзор методов интеграции информационных ресурсов высших учебных заведений / Р. М. Ершеева // Молодой ученый. №12. Т.1., 2011. – С. 75-78.

*Периодические издания*

37. Ненахова А. Васильев П., Мониторинг информации // Директор информационной службы – 2007 – № 01, с.1-3.

38. Попов Э.В. Корпоративные системы управления знаниями // Новости ИИ, № 1, 2001.

39. Чеботарев В. Моделирование корпоративного портала знаний // PC Week/RE, №(284)14`, 2001.

*Электронные ресурсы*

40. Confluence. [Электронный ресурс] // TeamLead [Интернет-портал] URL: http://www.teamlead.ru/display/MAIN/Confluence

41. Oracle MySQL. [Электронный ресурс] URL: http://www.oracle.com/index.html

42. Piwik vs Google Analytics: Подробный обзор [Электронный ресурс] // Rebill.me [Интернет-портал] URL: http://rebill.me/showthread.php?t=3456

43. Piwik — бесплатная открытая альтернатива Google Analytics. [Электронный ресурс] // HabraHabr [Интернет-портал] URL: http://habrahabr.ru/post/26820/

44. TF-IDF Similarity. Apache Lucene [Электронный ресурс] URL: https://lucene.apache.org/core/4_2_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html

45. Wikipedia. [Электронный ресурс] // Wikipedia [Интернет-портал] URL: https://ru.wikipedia.org/

46. Баллод, Б.А. Информационная система проведения мониторинговых исследований общественного мнения «Monitoring»., 2001 [Электронный ресурс] / Б.А. Баллод, А.А. Белов, П.А. Цуканов. – http://ptsukanov.narod.ru/aticles/v1.html.

47. Духнич Ю. Корпоративный портал управления знаниями [Электронный ресурс] // Smart Education [Интернет-портал]. URL: http://www.smart-edu.com/korporativnyy-portal-upravleniya-znaniyami.html

48. Корпоративный портал управления знаниями EKP (Enterprise Knowledge Portal) [Электронный ресурс] // Портал о Корпоративных порталах Консалтинг, создание, внедрение и поддержка [Интернет-портал]. URL: http://corportal.ru/Encyclopedia/CorPortal/EKP/EKP%20.aspx

49. Кучаба Е. Яндекс.Метрика vs Google Analytics Сравнение возможностей [Электронный ресурс] // URL: http://www.slideshare.net/HRdepartment/vs-google-analytics-30484579

50. Мельников Н. Управление знаниями в ИТ-компаниях, а нужно ли? 2011 [Электронный ресурс] // Habrahabr [Интернет-портал] URL: http://www.habrahabr.ru/

51. Мониторинг и аналитика социальных медиа: кому и зачем? [Электронный ресурс] // Разработка приложений Microsoft Azure Материалы по облачному бизнесу [Интернет-портал]. URL: http://msdn.microsoft.com/ru-ru/dn133036.aspx

*Литература на иностранном языке*

52. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze. Introduction to Information Retrieval. – Cambridge University Press, UK, 2010.

53. Kohler S. Atlassian Confluence 5 Essential. – PACKT, UK, 2013.

54. Kudryavtsev, D. V., Gavrilova, T. A. Diagrammatic knowledge modeling for managers – ontology-based approach. – International Conference on Knowledge engineering and Ontology Development, France, 2011.

55. Nick Craswell, Hugo Zaragoza, Stephen Robertson. Microsoft Cambridge at TREC-14: Enterprise Track. In Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005). – Gaithersburg, USA, 2005.

56. Pariser E. The Filter Bubble: What The Internet Is Hiding From You. — Penguin Group, NY, 2011. — 257 с.

57. Rajendra Akerkar. Advanced Knowledge Based Systems: Model, Applications & Research, Vol. 1, pp 1 – 11, 2010

58. Rajendra Akerkar. Knowledge-Based Systems., Sardar Patel University, India, 2010.

59. Salton, G. and Buckley, Term-weighting approaches in automatic text retrieval. Information Processing & Management 24(5), 1988

60. Vladik Kreinovich. Interval Methods in Knowledge Representation., International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems Vol. 22, No. 06, pp. 941-942 (2014)

61. Wilson D., Sperder D. Meaning and Relevance. – Cambridge University Press, - UK, 2012