

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра социальной информатики

**РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО
ПОИСКА ДАННЫХ В БАЗЕ ЗНАНИЙ С ИСПОЛЬЗОВАНИЕМ
ЧАТ-БОТА В VK**

(автореферат бакалаврской работы)

студента 5 курса 531 группы
направления 09.03.03 - Прикладная информатика
профиль Прикладная информатика в социологии
Социологического факультета
Красивского Дмитрия Сергеевича

Научный руководитель

кандидат физико-математических наук, доцент _____ М.Г. Плешаков
подпись, дата

Зав. кафедрой

кандидат социологических наук, доцент _____ И.Г. Малинский
подпись, дата

Саратов 2026

ВВЕДЕНИЕ

Актуальность темы. В современных условиях стремительного роста объёмов научной и технической информации организации всё острее сталкиваются с проблемой управления корпоративными базами знаний. Исследовательские группы, учебные заведения и аналитические подразделения накапливают значительные коллекции документов — диссертаций, отчётов, статей, технических руководств, — доступ к которым зачастую затруднён из-за отсутствия удобных инструментов поиска и классификации. Традиционные способы хранения файлов в папках и таблицах не обеспечивают ни быстрого поиска по содержанию, ни автоматической систематизации поступающих материалов.

Параллельно с этим мессенджеры прочно вошли в повседневную профессиональную коммуникацию. ВКонтакте занимает одну из лидирующих позиций среди российских платформ по числу активных пользователей, а интерфейс чата стал для многих привычнее и удобнее, чем специализированные веб-приложения. Разработка инструментов, встроенных непосредственно в привычную среду общения пользователя, позволяет снизить порог входа и повысить практическую востребованность системы.

Появление доступных языковых моделей открыло новые возможности для автоматизации работы с текстовыми данными. Системы, способные извлекать смысловые метаданные из неструктурированного текста — определять тематику, год, тип документа, формулировать краткое описание, — ранее требовали значительных ресурсов для разработки. Сегодня эти задачи решаются интеграцией с языковыми моделями через API, что существенно расширяет возможности небольших систем.

Таким образом, актуальность данной работы определяется сочетанием трёх факторов: потребностью в удобных инструментах управления документальными коллекциями, широким распространением мессенджеров как рабочей среды и доступностью языковых моделей для интеллектуальной обработки текста. Разработанная система представляет собой чат-бот на

платформе ВКонтакте, обеспечивающий загрузку документов различных форматов, их автоматическую классификацию с использованием GigaChat API и полнотекстовый поиск по накопленной базе знаний. Практическая ценность работы состоит в создании готового инструмента, не требующего специальных технических знаний от конечного пользователя и не предполагающего установки дополнительного программного обеспечения.

Степень изученности работы: Проблематика автоматизации доступа к корпоративным базам знаний через диалоговые интерфейсы находится на пересечении нескольких научных направлений: информационного поиска, разработки программного обеспечения и социальной информатики.

Теоретические основы диалоговых интерфейсов и их роль в информационном обмене исследованы в работах отечественных учёных. Андриюшина А. А. и Мириманов Д. А. рассматривают чат-боты как инструмент профессиональной коммуникации и выявляют новые возможности их применения в организационной среде¹. Параскевов А. В., Каденцева А. А. и Мороз С. И. анализируют архитектурные особенности и перспективы разработки чат-ботов, указывая на значимость детерминированных алгоритмов поиска в задачах справочного обслуживания².

Вопросы архитектуры программных систем, применяемой в данной работе, проработаны в фундаментальных трудах зарубежных авторов. Мартин Р. в монографии «Чистый код» формулирует принципы построения слоистых архитектур с инверсией зависимостей, которые составляют основу проектирования разработанной системы³. Фаулер М. в работе по рефакторингу описывает паттерны разделения ответственности между компонентами,

¹ Андриюшина А. А., Мириманов Д. А. Чат-боты: новые возможности и вызовы для редакций // Российская школа связей с общественностью. 2025. Т. 38. С. 175—194.

² Параскевов А. В., Каденцева А. А., Мороз С. И. Перспективы и особенности разработки чат-ботов // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. 2017. Т. 130. С. 395—404.

³ Мартин Р. Чистый код: создание, анализ и рефакторинг. Москва : Питер, 2019. С. 464.

реализованные в доменном слое системы¹. Клеппман М. исследует подходы к построению высоконагруженных систем, обосновывая применение асинхронной обработки и механизмов отказоустойчивости².

Практические аспекты разработки серверных приложений на языке Go освещены в работах Цукалоса М.³, а также Хайруллина Д. А., Чирковой К. Е. и Богданова М. Р.⁴, рассматривающих эффективные паттерны конкурентного программирования, применённые при реализации параллельной обработки файлов.

Вопросы разработки REST API на Go исследованы Кряжевой Е. В. и Амелиным Е. В.⁵. Архитектурные стандарты микросервисных систем систематизированы в трудах Ньюмена С.⁶ и Перальты Х. А.⁷

Методологические аспекты организации процесса разработки рассмотрены в работах Зараменских Е. П.⁸, Диденко Н. И.⁹ и Цыренова Д. Д.¹⁰, посвящённых жизненному циклу информационных систем и применению Agile-методологий.

Вместе с тем в изученной литературе практически отсутствуют работы, посвящённые комплексной реализации систем поиска по базе знаний, интегрированных непосредственно в экосистему социальной сети «ВКонтакте» с поддержкой загрузки и хранения документов различных форматов.

¹ Фаулер М. Рефакторинг: улучшение существующего кода. Москва : Диалектика, 2020. С. 448.

² Клеппман М. Высоконагруженные приложения. Программирование, масштабирование, поддержка. Санкт-Петербург : Питер, 2019. С. 640.

³ Цукалос М. Golang для профи: работа с сетью, многопоточность, структуры данных и машинное обучение с Go. Москва : Питер, 2021. 720 с.

⁴ Хайруллин Д. А., Чиркова К. Е., Богданов М. Р. ЭФФЕКТИВНЫЕ ПАТТЕРНЫ КОНКУРЕНТНОГО ПРОГРАММИРОВАНИЯ В GOLANG // Международный журнал гуманитарных и естественных наук. 2025. Т. 5—1 (104). С. 355—360.

⁵ Кряжева Е. В., Амелин Е. В. Разработка REST API микросервиса с использованием стандартного пакета NET/HTTP в GOLANG // Заметки ученого. 2021. № 6—1. С. 40—44.

⁶ Ньюмен С. Создание микросервисов. 2-е изд. Санкт-Петербург : Питер, 2025. С. 624.

⁷ Перальта Х. А. Микросервисы и API. 2-е изд. Санкт-Петербург : Питер, 2025. С. 464.

⁸ Зараменских Е. П. Информационные системы: управление жизненным циклом: учебник и практикум для среднего профессионального образования. Москва : Юрайт, 2024. С. 497.

⁹ Диденко Н. И., Скрипнюк Д. Ф., Дементьев И. И. Жизненный цикл сложных систем в среде бизнес-инжиниринга: учебное пособие для вузов. Москва : Юрайт, 2024. С. 210.

¹⁰ Цыренов Д. Д., Табинаев И. М. Agile-методологии в управлении бизнесом: от теории к практике // Естественно-гуманитарные исследования. 2025. Т. 3 (59). С. 957—963.

Большинство существующих решений либо ориентированы на англоязычные платформы (Telegram, Slack), либо используют крупные языковые модели в качестве основного поискового механизма, что создаёт зависимость от внешних API и снижает предсказуемость результатов. Разработанная система заполняет данный пробел, предлагая детерминированный подход к поиску с использованием тегово-скорингового алгоритма и полнотекстового поиска средствами SQLite FTS5, что обеспечивает высокую точность и независимость от сторонних сервисов.

Цель работы:

Разработка и программная реализация автоматизированной системы поиска данных в локальной базе знаний с использованием чат-бота в социальной сети «ВКонтакте».

Для достижения поставленной цели поставлены **следующие задачи:**

1. Провести системный анализ предметной области и сформировать перечень функциональных и нефункциональных требований к разрабатываемому чат-боту и алгоритмам поиска.

2. Спроектировать и реализовать серверную логику приложения, реализующую механизмы сопоставления пользовательских запросов со структурированными записями базы знаний.

3. Осуществить интеграцию приложения с инфраструктурой «ВКонтакте» с помощью официального VK SDK, настроив бесперебойный прием сообщений и отправку релевантных ответов пользователю.

Структура работы. ВКР состоит из введения, двух глав (по 4 и 8 параграфов соответственно), заключения и списка использованных источников и трёх приложений.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

В первой главе **«Теоретические основы разработки интеллектуальных систем поиска информации»** обобщены положения, необходимые для обоснования разрабатываемой системы.

Мессенджеры и социальные сети стали устойчивой частью профессиональной коммуникации, поэтому размещение поискового интерфейса внутри привычной среды снижает «цифровой барьер» по сравнению с отдельными веб-приложениями. Чат-бот в такой системе выступает как «тонкий клиент»: платформа обеспечивает доставку сообщений и интерфейс, а серверное приложение выполняет обработку запросов, поиск и формирование ответа. Ключевые преимущества подхода заключаются в низком пороге входа, кроссплатформенности и асинхронном взаимодействии с пользователем.

Для проекта с уточняющимися требованиями выбран итеративный подход с элементами Kanban: задачи выполнялись последовательно, без жёстких спринтов, с возможностью оперативно изменять приоритеты. В качестве основного языка использован Go, поскольку он сочетает компилируемость, строгую типизацию и встроенную поддержку конкурентной обработки через горутины, что важно для параллельной загрузки и анализа документов.

Разработка чат-бота для поиска по базе знаний обоснована ростом документальных коллекций, распространением мессенджеров и доступностью языковых моделей. Выбранные организационные и программные средства соответствуют требованиям простоты внедрения, производительности и сопровождаемости системы.

Во второй главе **«Разработка интеллектуальной системы»** рассматриваются архитектурные решения, алгоритмы и практические аспекты реализации системы.

Архитектура системы построена по принципам чистой и гексагональной архитектуры с учётом SOLID. Логика приложения разделена на доменный

слой, слой вариантов использования и слой адаптеров. Доменный слой содержит бизнес-сущности и не зависит от внешних библиотек; use cases реализуют прикладные операции; адаптеры обеспечивают работу с SQLite, парсерами файлов, HTTP API и VK Long Poll. Такое разделение позволяет заменять внешние сервисы и расширять поддержку форматов без изменения бизнес-логики.

Доменный слой включает сущности Entry, Intent и Job. Они описывают документ базы знаний, разобранный пользовательский запрос и состояние асинхронной обработки файла. Зависимости от инфраструктуры вынесены в интерфейсы-порты, что позволяет менять СУБД или транспорт без изменения бизнес-логики.

Система предоставляет два транспортных канала, работающих через общие use cases: HTTP REST API для программного взаимодействия и VK Long Poll для пользователей. В REST API загрузка файлов выполняется асинхронно: сервер возвращает идентификатор задачи, а клиент отслеживает её состояние через отдельный эндпоинт.

VK Long Poll выбран как основной пользовательский интерфейс, поскольку не требует публичного IP-адреса и TLS-сертификата. Входящие сообщения маршрутизируются по типу: файлы, команды и поисковые запросы. Несколько файлов из одного сообщения обрабатываются параллельно, а отправка ответов выполняется через единый метод с повторными попытками при сетевых ошибках.

Таблица 1 – Основные эндпоинты HTTP REST API

Метод	Путь	Назначение
POST	/chat	Поисковый запрос и получение ответа
POST	/upload	Загрузка документа в базу знаний
GET	/upload/{jobID}	Проверка статуса асинхронной загрузки
GET	/entries	Список всех документов (метаданные)
GET	/entries/{id}	Получение одного документа с полным текстом

DELETE	/entries/{id}	Удаление документа
GET	/health	Проверка работоспособности сервера

Система поддерживает PDF, DOCX/DOC, JSON, CSV и текстовые файлы.

Парсер выбирается автоматически компонентом CompositeParser; для PDF используется pdftotext, для документов Word — LibreOffice. Защита от дубликатов выполняется по SHA-256 хэшу и имени файла до начала ресурсоёмкой обработки. Оригинал хранится на диске, а извлечённый текст и метаданные — в SQLite.

Для автоматического обогащения документов метаданными система интегрирована с GigaChat API — большой языковой моделью от Сбера, обученной на обширном корпусе русскоязычных текстов. При обогащении в запрос к модели передаются первые 3000 символов текста документа; промпт предписывает вернуть валидный JSON с пятью полями: заголовок, описание, теги, год и категория.

Разработан механизм отказоустойчивости: при недоступности GigaChat система автоматически переключается на резервный компонент под названием HeuristicEnricher, определяющий метаданные непосредственно из структуры текста. Теги извлекаются из секции «Ключевые слова» или методом частотного анализа; категория — по ключевым словам в заголовке; описание — каскадным поиском аннотации, введения или первого достаточно длинного абзаца. Такая архитектура гарантирует работоспособность системы без потери основного функционала при любом состоянии внешних сервисов.

Для корректировки автоматически определённых метаданных реализованы команды /meta и /описание, позволяющие пользователю уточнить любое поле непосредственно в чате.

Поиск и ранжирование. Поиск реализован как последовательность трёх операций: классификация запроса, выборка кандидатов из базы данных и ранжирование результатов по релевантности.

Классификатор намерений IntentParser последовательно анализирует запрос: ищет год регулярным выражением, затем явные теги в синтаксисе #тег или [тег], наконец извлекает ключевые слова после фильтрации стоп-слов. Тип намерения определяет стратегию запроса к базе данных: поиск по году — прямой фильтр по индексированному полю, по тегу — поиск через LIKE по JSON-полю, по ключевым словам — полнотекстовый поиск SQLite FTS5. При наличии в запросе одновременно года и ключевых слов результаты обоих запросов объединяются без дубликатов.

Ранжирование выполняется компонентом TagScorer по балльной системе (таблица 2). Тематическое соответствие через теги намеренно весит выше, чем формальное совпадение слов в тексте, поскольку теги отражают авторскую интерпретацию тематики документа.

Таблица 2 – Критерии расчёта релевантности в TagScore

Критерий	Баллы
Совпадение тега	+3
Совпадение года	+2
Ключевое слово в заголовке	+2
Ключевое слово в тексте	+1

Для каждого найденного документа дополнительно извлекаются контекстные сниппеты — фрагменты текста вокруг вхождений ключевых слов (окно 150 символов с каждой стороны, не более трёх сниппетов на документ). Итоговый ответ содержит до пяти наиболее релевантных документов с заголовком, годом, категорией, тегами, сниппетом и процентом релевантности.

Для подключения бота создаётся сообщество ВКонтакте и генерируется API-токен с необходимыми правами доступа. Пользователь начинает взаимодействие, перейдя на страницу сообщества и нажав кнопку «Сообщение». Загрузка документа осуществляется отправкой файла боту; в ответ пользователь получает сводное уведомление с автоматически определёнными метаданными и подсказкой для их уточнения. Поиск

выполняется отправкой произвольного текстового запроса; скачивание оригинального файла — через команду /документы.

Разработанная система реализует разделение бизнес-логики и инфраструктуры, поддерживает загрузку пяти форматов файлов, защищает базу от дубликатов и выполняет автоматическое обогащение метаданных через GigaChat API с эвристическим резервным механизмом. Поиск сочетает SQLite FTS5 и балльное ранжирование, что позволяет формировать релевантные ответы на пользовательские запросы.

ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы были успешно решены поставленные задачи и достигнута цель — была реализована система автоматизированного поиска данных в базе знаний с использованием чат-бота, используя платформу «ВКонтакте». Разработан функционал поиска данных в базе знаний, отправка оригинальных файлов по запросу пользователя, получение и обработка полученных документов. Произведена интеграция GigaChat API для более корректного создания описаний, для присланных боту, работ. Разработан модуль для работы чат-бота с платформой «ВКонтакте», которая позволяет удобно использовать весь функционал.

Изучение современных архитектурных подходов — чистой архитектуры, гексагональной архитектуры и принципов SOLID — позволило спроектировать систему с чётким разделением ответственности между компонентами. Это решение обеспечило независимость бизнес-логики от конкретных технологий и упростило возможность замены отдельных компонентов без модификации остального кода.

Исследование механизмов интеграции с платформой ВКонтакте позволило реализовать удобный пользовательский интерфейс на основе Long Poll API, не требующий от пользователя установки дополнительного программного обеспечения или освоения новых инструментов. Параллельно разработан HTTP REST API, предоставляющий программный доступ к

функциям системы для интеграции со сторонними сервисами и автоматизированной загрузки документов.

Реализация конвейера обработки документов охватила поддержку пяти форматов файлов: PDF, DOCX, DOC, CSV, JSON и текстовых файлов. Каждый формат обрабатывается специализированным парсером, выбираемым автоматически по расширению файла. Двухуровневая защита от дубликатов на основе SHA-256 хэша и проверки имени файла исключает повторное сохранение одного и того же документа.

Интеграция с GigaChat API обеспечила автоматическое обогащение загружаемых документов структурированными метаданными: тематическими тегами, кратким описанием, годом создания и типом документа. Разработанный резервный механизм на основе эвристического анализа текста гарантирует работоспособность системы при недоступности внешнего API, реализуя принцип отказоустойчивости без потери основной функциональности.

Реализация поиска объединила полнотекстовый поиск средствами SQLite FTS5, тегово-скоринговый алгоритм ранжирования и классификатор намерений пользователя, распознающий запросы по году, тегу и ключевым словам. Совокупность этих механизмов обеспечивает точный и быстрый поиск по накопленной коллекции документов.

Таким образом, работа не только достигла поставленных целей, но и продемонстрировала практическую применимость современных архитектурных паттернов при разработке систем управления знаниями. Разработанная система представляет собой готовый инструмент, способный повысить эффективность работы с документальными коллекциями в исследовательских группах и организациях, использующих ВКонтакте как рабочую коммуникационную платформу.