

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

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Разработка программного обеспечения

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка и внедрение цифрового меню для интеграции с чат-ботом в
программе 1С Коннект»

Обучающийся

С.Е. Бутакова

(Инициалы Фамилия)

(личная подпись)

Руководитель

В.В. Дружинкин

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

Аннотация

Тема выпускной квалификационной работы «Разработка и внедрение цифрового меню для интеграции с чат-ботом в программе 1С Коннект»

Ключевые слова: Чат бот, 1С Коннект, цифровое меню.

Предмет исследования – разработка пользовательского интерфейса цифрового меню и его внедрения с чат ботов в программе 1С Коннект, как инструмента автоматизации обслуживания и коммуникации клиентов со специалистами линии поддержки.

Цель выпускной квалификационной работы - повышение удобства и взаимодействия пользователей, автоматизация типовых задач и улучшение эффективности коммуникаций в рабочей среде (за счет внедрения в программу 1С Коннект).

Практическая значимость выпускной квалификационной работы заключается в разработке и внедрении прикладного решения, направленного на повышение эффективности взаимодействия клиентов с линией поддержки в программе 1С Коннект.

Данная работа состоит из введения, четырех разделов, заключения и списка используемых источников.

Выпускная квалификационная работа состоит из 51 страницы, 4 разделов, 22 рисунков, 21 листинга и 21 источника.

Содержание

Аннотация	2
Введение.....	4
1 Анализ предметной области	6
1.1 Постановка задачи и анализ требований.....	8
2 Проектирование программного продукта	12
2.1 Проектирование в UML.....	12
3 Техническое описание реализованного решения	15
3.1 Общее описание	15
3.2 Описание графической части.....	16
3.3 Структура разделов.....	17
3.4 Структура статей.....	17
3.5 Описание чат бота.....	19
3.6 Парсер статей.....	21
3.7 Вспомогательные средства	28
3.8 Готовая серверная часть чат бота.....	30
4 Документирование	32
4.1 Руководство пользователя для сотрудника.....	32
4.2 Руководство пользователя для клиента	35
4.3 Настройка DokuWiki.....	40
4.4 Установка DokuWiki на сервер.....	44
Заключение	48
Список используемых источников.....	50

Введение

Актуальность исследования. Разработка и внедрение чат бота становятся актуальными на фоне перехода современного бизнеса к цифровым решениям. Такой инструмент помогает автоматизировать общение с клиентами и снизить нагрузку на специалистов поддержки. Чат бот может быстро отвечать на однотипные вопросы, помогать сотрудникам с типовыми задачами и тем самым экономить время и ресурсы.

Чат бот включает в себя цифровое меню - удобный интерфейс, который помогает пользователю быстро находить нужную информацию или выполнять действия без лишних команд, при помощи кнопок меню. Вместо того, чтобы вводить текст вручную, клиент просто выбирает нужный пункт из списка. Это ускоряет работу, снижает количество ошибок и делает общение с ботом более понятным и комфортным. Особенно важно это где ценится скорость обработки вопроса и простота в использовании.

Предметом исследования является применение чат бота в программе 1С Коннект как инструмента автоматизации обслуживания клиентов и повышения эффективности коммуникаций с линией поддержки.

Цель выпускной квалификационной работы - повышение удобства и взаимодействия пользователей, автоматизация типовых операций и улучшение эффективности коммуникаций в рабочей среде (за счет внедрения в программу 1С Коннект). Проект направлен на создание удобного, интуитивно понятного интерфейса для пользователей, позволяющего взаимодействовать с чат ботом без ввода текстовых команд, а с помощью готовых кнопок и структурированного меню. Это должно повысить эффективность и скорость выполнения однотипных задач, снизить нагрузку на сотрудников поддержки, минимизировать количество ошибок при вводе текста обращения, а также улучшить общий пользовательский опыт в программу 1С Коннект.

Для достижения цели выпускной квалификационной работы были поставлены следующие задачи:

- определить, каким должно быть меню, чтобы им было удобно пользоваться;
- спроектировать структуру чат бота с цифровым меню;
- создать работающий прототип чат бота для 1С Коннект;
- настроить взаимодействие чат бота с базой знаний DokuWiki;
- выполнить разделение прав доступа для специалистов поддержки к базе знаний DokuWiki;
- провести подключение к линии поддержки «Сопровождение» в 1С Коннект, протестировать и убедиться, что чат бот работает корректно.

Практическая значимость данной бакалаврской работы [19] заключается в создании и внедрении чат бота в программу 1С Коннект, что позволяет сократить нагрузку на службу поддержки, при решении однотипных вопросов и улучшению коммуникации между поддержкой и клиентами.

1 Анализ предметной области

ООО «Миллениум сервис» – был основан в 2010 году и занимается внедрением и сопровождением программных продуктов на базе «1С:Предприятие», предоставляя широкий спектр услуг по автоматизации учётной работы, а также доработкой конфигураций 1С под требования клиентов.

ООО «Миллениум сервис» располагается в городе Тольятти и за свою 15 летнюю историю обзавелись более сотни постоянных организаций-клиентов. Являясь 1С Франчайзи ООО «Миллениум сервис» берет на себя обязательства соответствовать стандартам высокого уровня качества, принятого фирмой 1С. А именно:

- честно и корректно относиться к клиентам, сотрудникам и другим партнерам Фирмы "1С";
- конкурировать с другими 1С франчайзи качеством, а не ценой;
- обслуживать только лицензионное программное обеспечение;
- не использовать массовых рассылок сообщений по электронной почте, отправляемых без согласия владельца электронного адреса (спам).

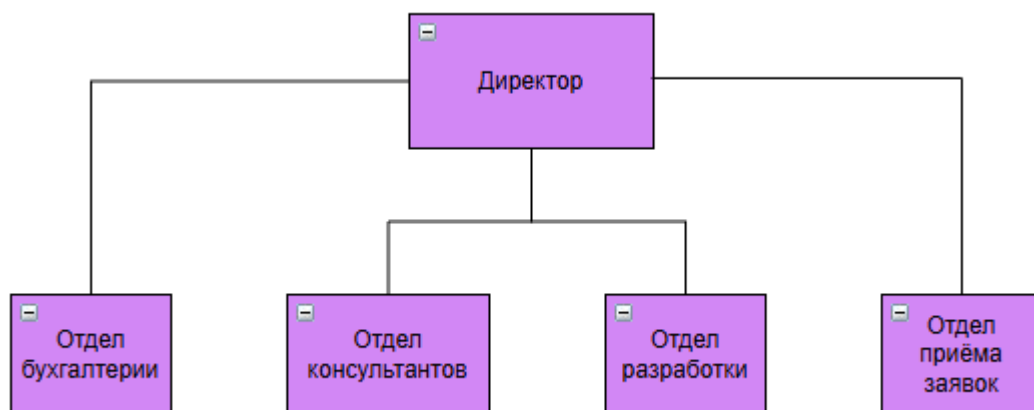


Рисунок 1 – организационная схема предприятия

Отделы предприятия:

- отдел бухгалтерии отвечает за финансовое управление и учетные процессы в организации. Он занимается ведением бухгалтерского учета, составлением финансовой отчетности, контролем за расходами и доходами, подготовкой налоговой отчетности и соблюдением финансовых стандартов. Отдел бухгалтерии регулируется законодательством о бухгалтерском учете и налогообложении;
- отдел разработки занимается разработкой и поддержкой информационных систем в организации. Он отвечает за разработку и настройку обработок в программных продуктах 1С, разработку программного обеспечения, установку и обслуживание компьютеров и сетей, а также управление веб-сайтом организации. Отдел разработки обеспечивает функциональность информационных систем, поддерживает их работоспособность, решает проблемы, связанные с технической поддержкой и обеспечением безопасности данных;
- отдел консультантов занимается консультированием клиентов по вопросам использования программных продуктов 1С. Консультанты отвечают на вопросы клиентов, предоставляют рекомендации по настройке и оптимизации работы с программами, помогают в решении проблем и проводят обучение пользователей. Они обладают обширными знаниями в области функциональности и возможностей программных продуктов 1С;
- отдел приема заявок отвечает за прием и обработку заявок от клиентов. Специалисты принимают входящие заявки клиентов, регистрируют их, проверяют и передают соответствующим отделам для дальнейшей обработки.

1.1 Постановка задачи и анализ требований

Задача:

Создать базу знаний для клиентов с быстрыми ответами на наиболее частые вопросы.

Функциональные требования:

- доступность для клиентов;
- простота получения информации для клиента;
- простота наполнения базы знаний для сотрудников;
- низкие требования к оборудованию.

Возможные решения:

- 1С ИТС [1] – большая база знаний от самой фирмы 1С, но при этом платная и больше рассчитана для специалистов;
- база знаний в 1С Коннект [2] – заполняемая вручную база на сайте 1С Коннект для отображения контекстных подсказок, при этом неудобное заполнение и вывод данных;
- 1С Коннект API – позволяет добавить в 1С Коннект произвольное контекстное меню, гибкое решение, но не примитивное;
- сайт с базой знаний – можно создать сайт, на котором публиковать наиболее частые ответы на вопросы, но не все клиенты захотят использовать что-то новое.

Для данной задачи было выбрано решение 1С Коннект API [3] из-за своей гибкости и доступности, большинство клиентов используют 1С Коннект. Данное API позволяет с помощью веб хука создать текстовое меню с произвольными действиями, например:

- вывод информации;
- создание заявки;
- перевод на специалиста;
- запуск произвольного действия на сервере;
- отправка клиенту файлов.

На данную роль подходит решение 1sconnect-text-bot который реализует произвольное конфигурируемое текстовое меню на заданных линиях поддержки. Основные характеристики:

- язык – GO [14];
- открытый исходный код;
- ОС – в приоритете linux, но и возможность запуска на windows;
- конфигурация – yml файлы, с возможностью менять меню, без остановки работы чат-бота и перезагрузки системы.

Yaml достаточно простой язык для редактирования человеком, но при большом количестве статей это неудобно, подробнее о yaml [17]. Поэтому для создания базы знаний был выбран DokuWiki – платформа для организации и управления документацией, которая отличается простотой использования, гибкостью и возможностью легко обновлять информацию без специальных технических навыков. Основные преимущества для выбора DokuWiki послужили:

- простота;
- открытый исходный код;
- низкие требования к ресурсам сервера;
- хранение статей в файлах;
- расширяемость с помощью плагинов.

Более подробно о платформе DokuWiki в статье [13].

Для того чтобы соединить lconnect-text-bot и DokuWiki необходимо промежуточное ПО, которое обрабатывает файловую систему на наличие файлов со статьями и на основе их генерировать умал файлы с контекстным меню для чат бота. Основные требования к парсеру:

- простота развёртывания;
- низкие требования к ресурсам;
- поддержка utf8 [9].

Для создания обработчика выбран язык C++ [12] из-за своих широких возможностей и быстроты.

В качестве операционной системы был выбран linux, основные преимущества linux для данного проекта:

- открытое ПО;
- безопасность;
- низкие требования к оборудованию;
- поддержка utf8 на уровне файловой системы;
- возможность работы без GUI [5].

Тем самым linux идеально подходит в качестве серверной операционной системы для данного проекта.

На рисунке 2 показано как сейчас клиент взаимодействует с фирмой если у него возникли вопросы или проблемы в 1с.

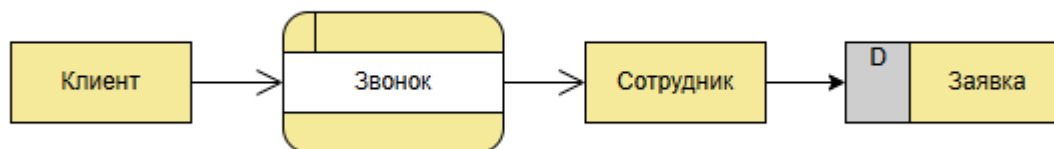


Рисунок 2 – модель «как есть»

На рисунке 3 показано что помимо звонка теперь есть возможность клиенту самому найти в базе знаний наиболее частые вопросы.

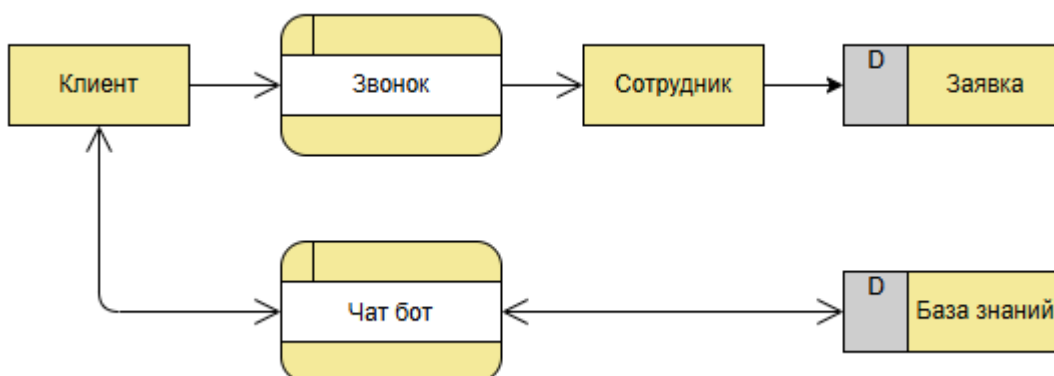


Рисунок 3 – модель «как должно быть»

Чат бот не заменяет собой сотрудника, но позволяет разгрузить сотрудников от самых частых и простых вопросов и заявок.

2 Проектирование программного продукта

2.1 Проектирование в UML

UML – унифицированный язык моделирования, который используется для отображения, описания, проектирования и документирования программных систем.

2.1.1 Диаграмма классов

Диаграмма классов – это один из главных типов диаграмм в UML, который показывает структуру системы через классы, их свойства (атрибуты), методы (операции) и связи между ними.

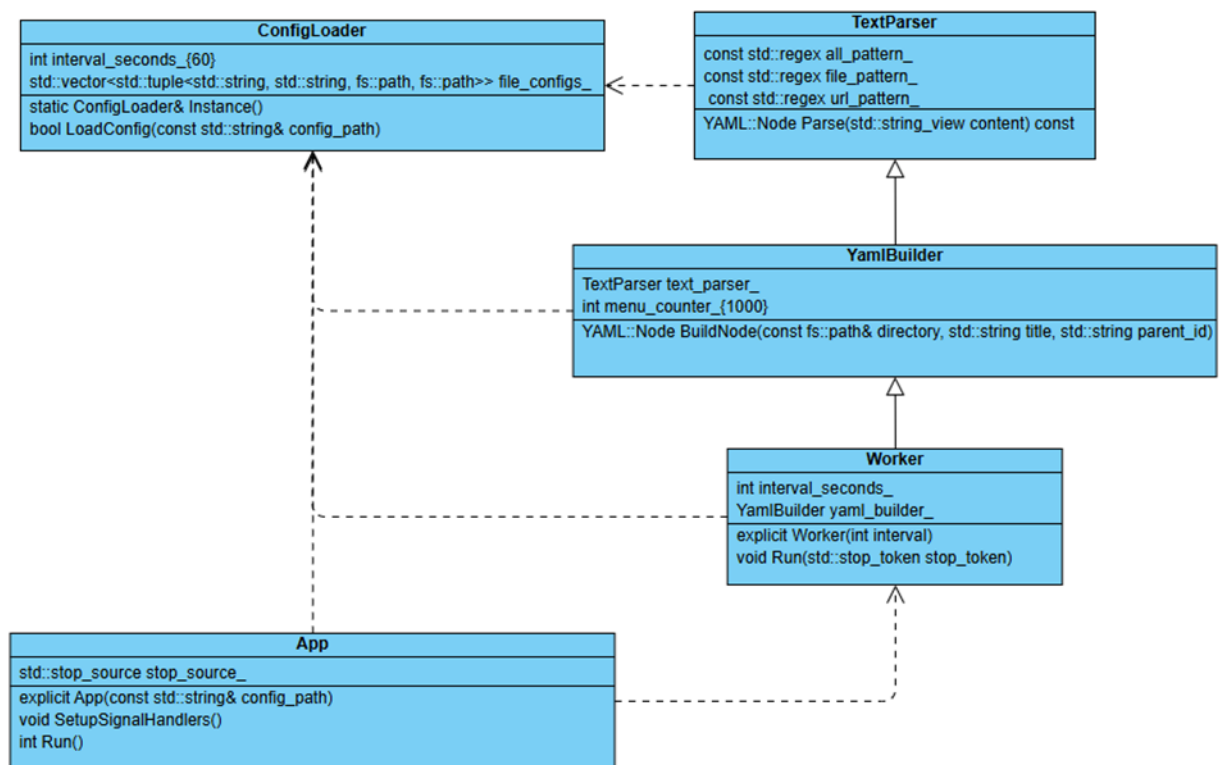


Рисунок 4 – Диаграмма классов

Описание классов показанных на рисунке 4:

- ConfigLoader это синглтон [22] который отвечает за настройки;
- App отвечает за запуск и завершение;
- Worker запускается в отдельном потоке и выполняет работу;
- YamlBuilder создаёт саму иерархию меню;
- TextParser парсит файлы статей и преобразует содержащийся текст в вид пригодный для yaml с учётом содержания картинок и ссылок.

2.1.2 Диаграмма вариантов использования

Диаграмма вариантов использования – один из видов UML-диаграмм, который показывает взаимодействие пользователей (акторов) с системой через сценарии использования (или «прецеденты»). Её основная цель – описать, какие функции выполняет система с точки зрения пользователя, а не как они реализованы изнутри.

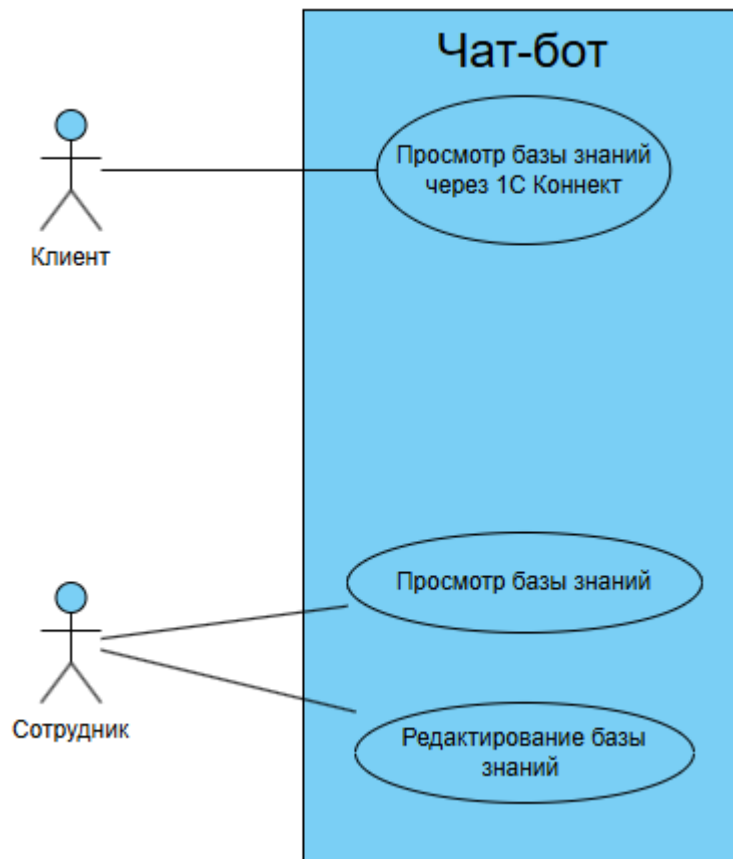


Рисунок 5 – диаграмма вариантов использования

Клиенты взаимодействуют только через 1С Коннект и только на просмотр базы знаний. Сотрудники же могут как просматривать, так и редактировать базу знаний через DokuWiki.

3 Техническое описание реализованного решения

В качестве системы используется Linux Ubuntu 24.04.1 LTS.

3.1 Общее описание

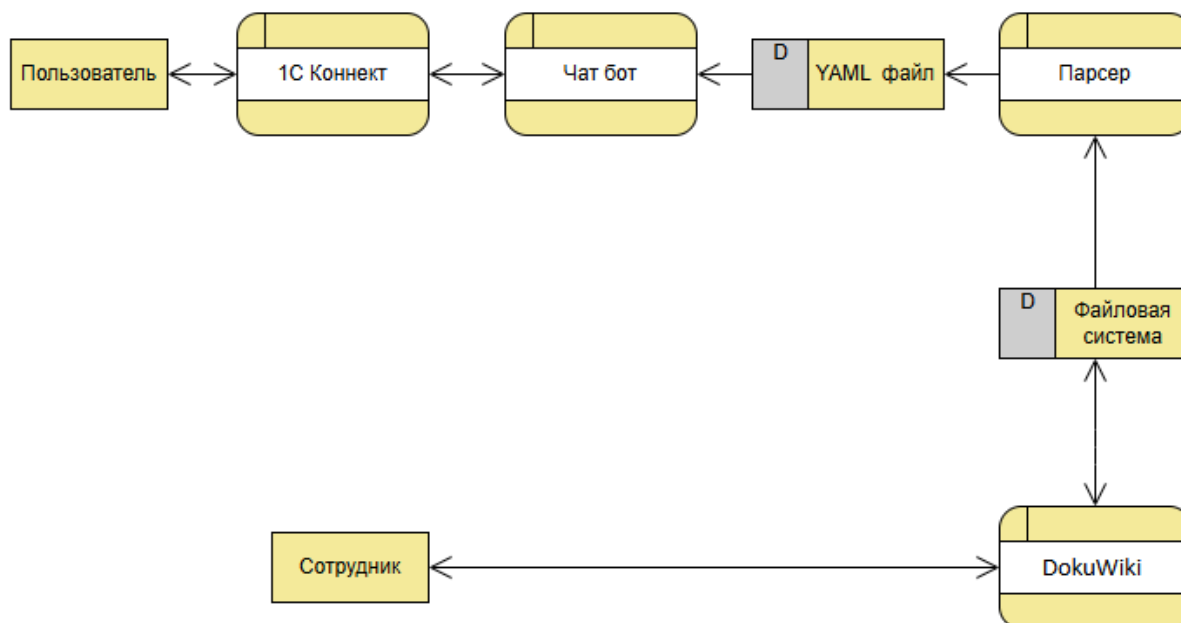


Рисунок 6 – описание взаимодействия

Сотрудник через веб интерфейс взаимодействует с DokuWiki который развёрнут в локальной сети. Парсер ищет в указанно директории статьи и из них генерирует yaml файл. Далее чат бот считывает yaml файл и на основе него создаёт меню, которое посредством веб хука считывает уже 1С Коннект.

3.2 Описание графической части

В качестве интерфейса для создания статей будет DokuWiki установленный на Nginx [7].

DokuWiki – это простой в использовании и мощный инструмент для ведения документации, написанный на PHP [8]. Он предназначен для создания вики-страниц без использования базы данных, храня статьи в виде текстовых файлов.

Основные особенности:

- легкость установки и настройки;
- поддержка системы управления версиями для отслеживания изменений;
- шаблоны и плагины для кастомизации;
- простая синтаксическая разметка для создания контента;
- управление доступом и правами пользователей;
- подходит для совместной работы и командных проектов;
- идеально подходит для внутренней документации компаний, ведения заметок или создания вики-сайтов.

Nginx – это высокопроизводительный веб-сервер и обратный прокси-сервер, который также может использоваться как почтовый прокси и балансировщик нагрузки.

Основные особенности:

- обработка большого количества одновременных соединений;
- поддержка статического и динамического контента;
- балансировка нагрузки и отказоустойчивость;
- реверс-прокси с кэшированием;
- поддержка HTTP/2, gRPC [15] и WebSocket [10];
- широко используется для ускорения веб-приложений, обслуживания статического контента и организации надежной инфраструктуры.

3.3 Структура разделов

DoKuWiki хранит статьи в текстовых файлах. Структура в файловой системе повторяет структуру в веб приложении DoKuWiki. Для того чтоб и название файлов повторяли названия статей необходимо в настройках DoKuWiki включить использование utf8 для названия файлов. Из минусов DoKuWiki не позволяет называть файлы и разделы с пробелами, тем самым нижние подчёркивания придётся заменять на пробелы после парсинга.

Данная структура используется в целях тестирования, в итоговом виде на предприятии используется похожая структура, но без лишних разделов.

Описание разделов:

- wiki – содержит справочную информацию о самом DoKuWiki;
- playground – используется для тестирования;
- sidebar – страница сайдбара;
- chat_bot – содержит все статьи и разделы что видит чат бот;
- часто_задаваемые_вопросы – тестовый раздел для проверки поддержки корневого раздела на кириллице.

3.4 Структура статей

DoKuWiki содержит весьма функциональный редактор с поддержкой широких возможностей форматирования текста, а благодаря плагинам возможна поддержка markdown.

Но из-за ограничений 1С Коннект в статьях используются только:

- простой текст без форматирования;
- картинки или файлы;
- ссылки.

Пример ссылки в тексте статьи показан в листинге 1.

```
[[http://example.com|Внешняя ссылка]]
```

Листинг 1 – пример ссылки в тексте статьи

Пример картинки в тексте статьи показан в листинге 2.

```
{{:chat_bot:бухгалтерия:снимок_экрана_от_2024-02-08_13-24-53.png?400|}}
```

Листинг 2 – пример картинки в тексте статьи

При этом DokuWiki в режиме просмотра отображает их в нормальном виде, как ссылку и картинку соответственно. Пример тестового файла содержащий текст, картинку и ссылку показан в листинге 3.

текст

```
{{:chat_bot:бухгалтерия:снимок_экрана_от_2024-02-08_13-24-53.png?400|}}
```

ещё текст

эта ссылка http: [[http://example.com|Внешняя ссылка]]

ещё один текст

```
{{:chat_bot:бухгалтерия:снимок_экрана_от_2024-02-08_13-24-53.png?400|}}
```

это ссылка https: [[https://example.com|Внешняя ссылка]]

Листинг 3 – текстовый файл содержащий текст, ссылку и картинку

3.5 Описание чат бота

Данный бот реализует произвольное конфигурируемое текстовое меню на заданных линиях поддержки. Техническая информация:

- операционная система Linux или Windows [20];
- язык GO;
- для сборки есть удобный скрипт build.sh.

Пример запуска бота показан в листинге 4.

```
./connect-text-bot --config=config.yml --bot=bot.yml
```

Листинг 4 – команда запуска чат бота

Параметры запуска:

- config – путь к конфигурации;
- bot – путь к конфигурации текстового меню;
- logger – путь к конфигурации логирования;
- debug – режим отладки.

Для удобства конфигурация меню разбита на несколько файлов:

- main.yml – главное меню;
- knowledge_base.yml – база знаний, которая будет генерироваться;
- connect_experts.yml – меню соединение с конкретными специалистами;
- create_request.yml – меню для создания заявки.

Файлы connect_experts.yml и create_request.yml на текущий момент не используются из-за ненужности организации, но оставлены на всякий случай. Для объединения yaml файлов используются анкера, например в файле knowledge_base.yml в самом начале файла есть строка с анкером &knowledge_base.

В main.yml в стартовом меню есть ссылка на файл с базой знаний.

```
start:
  answer:
    - chat: "Здравствуйте {{ .User.Name }}, добро пожаловать в чат
бот Миллениум сервис."
  buttons:
    - <<: *knowledge_base
    - button:
      close_button: true
```

Листинг 5 – стартовое меню

Так же в main.yml описано финальное меню.

```
final_menu:
  answer:
    - chat: "Могу ли я вам чем-то еще помочь?"
  buttons:
    - button:
      id: 1
      text: "Да"
      goto: "start"
    - button:
      id: 2
      text: "Нет"
      chat:
        - chat: "Спасибо за обращение!"
      close_button: true
```

Листинг 6 – Финальное меню

Более подробно о файле `knowledge_base.yml` и как он генерируется будет рассказано в следующем разделе.

3.6 Парсер статей

Для того чтобы из статей DokuWiki создать файл `knowledge_base.yml` с меню необходимо дополнительное приложение. Приложение парсит директорию со статьями DokuWiki и на основе этих файлов генерирует `knowledge_base.yml`. Название каждой директории или файла преобразуется в название кнопки, при этом нижнее подчёркивание преобразуется в пробелы, а первая буква в названии преобразуется в верхний регистр. Текст в файле преобразуется в текст как есть, а ссылки и файлы из формата DokuWiki преобразуются в формат понятный чат боту.

Пример кнопки с текстом, ссылкой и файлом показан на листинге 7.

```
- button:
    id: 2
    text: Файл
    chat:
      - chat: "текст\n\n"
      - file:
chat_bot/бухгалтерия/снимок_экрана_от_2024-02-08_13-24-53.png
      - chat: "\n\nещё текст\n\nэта ссылка http:
http://example.com\n\nещё один текст\n\n"
      - file:
chat_bot/бухгалтерия/снимок_экрана_от_2024-02-08_13-24-53.png
      - chat: "\n\nэто ссылка https:
https://example.com\n\n"
```

Листинг 7 – кнопка с текстом, ссылкой и файлом

В программе это реализуется через регулярные выражения [21], паттерны для поиска файлов, ссылки, листинг 8.

```
const std::string kFilePattern =  
R"(\{\{\s*:?s*([\^s\?\\|]+?)(?:\s*[\?\\|].*?)?\s*\}\})";  
const std::string kUrlPattern =  
R"(\[[\](https?:\|\/[\^\\|]+)\|[\^\\|]+\]\])";  
const std::string kAllPattern = R"([\[\[.*?\]\]|(\{\{\.*?\}\}))";
```

Листинг 8 – паттерны для поиска файлов и ссылок

Сначала через `kAllPattern` идёт поиск специальных последовательностей, а далее с помощью `kFilePattern` или `kUrlPattern` извлекается данные в нужном формате.

Из за того что названия файлов к в кодировке `utf8` для преобразования букв в верхний регистр используется сторонняя библиотека `utf8.h`. Для генерации `knowledge_base.yml` используется библиотека `yaml-cpp` [18]. Функция для рекурсивной генерации меню показана на листинге 9.

```
YAML::Node CreateYamlNode(const fs::path& dir_path, const std::string&  
text, const std::string& id_last_menu) {  
    YAML::Node node;  
    int button_id = 1;  
    int current_menu_id = menu_id;  
    node["id"] = menu_id++;  
    node["answer"] = YAML::Node(YAML::NodeType::Sequence);  
    node["buttons"] = YAML::Node(YAML::NodeType::Sequence);  
    YAML::Node chat;  
    chat["chat"] = text;  
    node["answer"].push_back(chat);  
  
    for (const auto& entry : fs::directory_iterator(dir_path)) {
```

```

if (entry.is_directory()) {
    YAML::Node sub_node;
    auto filename = entry.path().filename().string();
    sub_node["id"] = std::to_string(button_id++);
    sub_node["text"] = ButtonName(filename);
    sub_node["menu"] =
        CreateYamlNode(entry.path(), "Выберите раздел",
std::to_string(current_menu_id));
    YAML::Node button;
    button["button"] = sub_node;
    node["buttons"].push_back(button);
} else if (entry.is_regular_file() && entry.path().extension() ==
".txt") {
    YAML::Node article_node;
    article_node["id"] = button_id++;
    article_node["text"] = ButtonName(entry.path().stem().string());
    article_node["chat"] = ParseText(ReadFileContent(entry.path()));
    YAML::Node button;
    button["button"] = article_node;
    node["buttons"].push_back(button);
}
}
YAML::Node back_button;
back_button["button"]["id"] = button_id++;
back_button["button"]["text"] = "Назад";
back_button["button"]["goto"] = id_last_menu;
node["buttons"].push_back(back_button);
return node;
}

```

Листинг 9 – функция генерации меню

Если функция находит директорию [4], то она создаёт кнопку с названием директории и вызывает саму себя, если же попадается файл с расширением txt, то создаётся кнопка с названием файла и парсится содержимое файла. У каждой кнопки и меню есть свой идентификатор, для них используется сквозная нумерация.

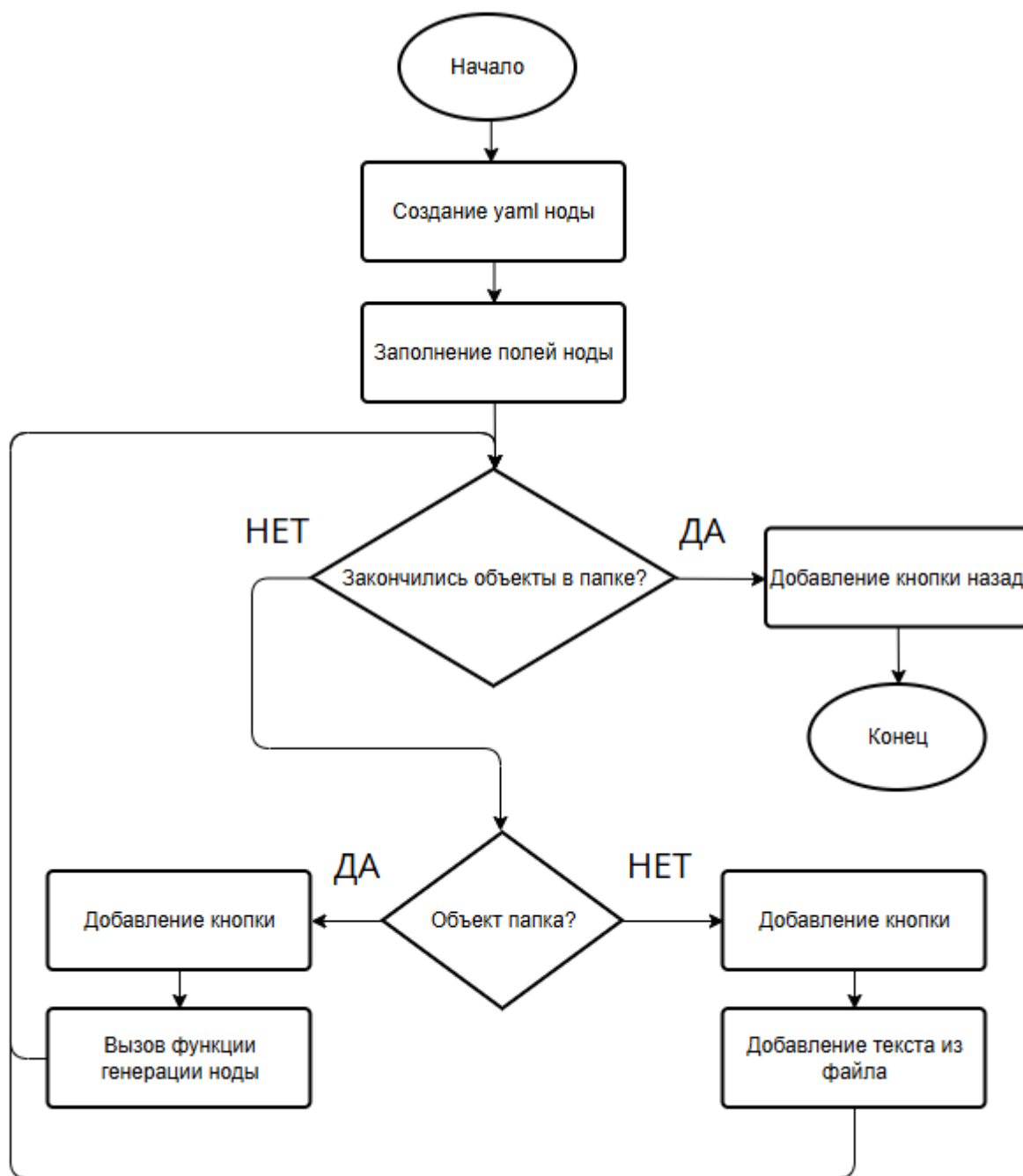


Рисунок 7 – блок схема функции генерации меню

В листинге 10 предоставлен исходный код функции парсинга текста.

```
YAML::Node ParseText(const std::string& content) {
    YAML::Node chatNode(YAML::NodeType::Sequence);
    std::regex pattern(kAllPattern);
    std::sregex_token_iterator iter(content.begin(), content.end(),
pattern, {-1, 1, 2});
    std::sregex_token_iterator end;
    std::string text;
    auto WriteText = [&text, &chatNode]() {
        if (text.empty())
            return;
        YAML::Node textNode;
        textNode["chat"] = text;
        chatNode.push_back(textNode);
        text.clear();
    };
    while (iter != end) {
        std::string part = *iter++;
        if (part.empty())
            continue;
        std::smatch match;
        if (std::regex_match(part, match, std::regex(kFilePattern))) {
            WriteText();
            std::string str = match.str(1);
            std::ranges::replace(str, ':', '/');
            YAML::Node fileNode;
            fileNode["file"] = str;
            chatNode.push_back(fileNode);
        } else if (std::regex_match(part, match, std::regex(kUrlPattern)))
        {
            text += match.str(1);
        } else {
            text += part;
        }
    }
}
```

```

    }
}
WriteText();
return chatNode;
}

```

Листинг 10 – функция парсинга текста

Каждый файл обрабатывается этой функцией и генерируется итоговый текст, который попадает уже в Node. Так же стоит обратить внимание что текст в кодировке utf8. Если utf8 текст из статьи можно записывать как есть, то название статей необходимо преобразовывать. Как это делается для создания названия кнопки по названию статьи показано в листинге 11.

```

std::string ButtonName(const std::string& input) {
    std::string result;
    result.reserve(input.size());
    utf8_int8_t* ptr =
reinterpret_cast<char8_t*>(const_cast<char*>(input.data()));
    utf8_int32_t codepoint;
    bool first_char = true;
    size_t size = utf8nlen(ptr, input.size());
    for (size_t i{0}; i < size; i++) {
        ptr = utf8codepoint(ptr, &codepoint);
        if (codepoint == '_') {
            result += ' ';
        } else {
            if (first_char) {
                codepoint = utf8uprcodepoint(codepoint);
                first_char = false;
            }
            utf8_int8_t temp[5] = {0};
            utf8catcodepoint(temp, codepoint, sizeof(temp));

```

```

    result += reinterpret_cast<char*>(temp);
}
}
return result;
}

```

Листинг 11 – преобразование текста utf8

Как можно заметить в функции преобразуется первый символ в верхний регистр, а нижнее подчёркивание в пробелы. Используется библиотека `utf8.h`. Перед запуском приложения считывает настройки с файла конфигурации содержащий в `yaml` такие данные как:

- интервал запуска парсинга;
- название раздела;
- пути, в которых находятся анализируемые директории;
- пути, в которые генерировать `yaml` файл.

Сам парсинг выполняется в отдельном потоке в бесконечном цикле, а для реакции программы на сигналы операционной системы и завершения цикла используется:

- `std::condition_variable` – для того что бы разбудить спящий поток;
- `std::atomic<bool>` – флаг для проверки что нужно завершаться;
- `std::mutex` – в данном контексте используется совместно с `condition_variable`;
- `std::signal` – используется для чтения сигнала операционной системой.

Пример кода, отвечающий за завершение приложения показан в листинге 12.

```

// переменные
std::atomic<bool> running{true};
std::condition_variable cv;
std::mutex cv_m;

```

```

// обработчик сигнала, который будет поток
void SignalHandler(int signal) {
    if (signal == SIGINT || signal == SIGTERM) {
        std::cout << "Получен сигнал завершения (" << signal << ").
Завершаем работу..." << std::endl;
        running = false;
        cv.notify_all();
    }
}
// считыватели сигналов ОС
std::signal(SIGINT, SignalHandler);
std::signal(SIGTERM, SignalHandler);
// засыпание потока между запуском парсинга
std::unique_lock<std::mutex> lock(cv_m);
cv.wait_for(lock, std::chrono::seconds(interval), [] { return
!running; });

```

Листинг 12 – часть кода, отвечающая за остановку приложения

3.7 Вспомогательные средства

Для удобства запуска, остановки и обновления созданы дополнительные bash скрипты. Для запуска и остановки используется скрипт start.sh, листинг 13.

```

#!/bin/bash

# Получение текущей директории скрипта
CURRENT_DIR="$(dirname "$0")"

case "$1" in
start)

```

```

# Запуск бота с конфигурационными файлами в фоне и фильтрация вывода
echo "Старт connect-text-bot, лог в bot.log"
nohup ./1connect-text-bot/connect-text-bot --
config=$CURRENT_DIR/config/config.yml --
bot=$CURRENT_DIR/config/bot/main.yml >bot.log 2>&1 &
echo "Старт YamlParser, лог в parser.log"
nohup ./YamlParser "$CURRENT_DIR/config/parser_config.yml"
>parser.log 2>&1 &

echo "Бот запущен в фоновом режиме."
;;
stop)
killall connect-text-bot
killall YamlParser
echo "Остановка приложений..."
;;
*)
echo "Неизвестная команда: $1"
echo "Используйте 'start' или 'stop'"
;;
Esac

```

Листинг 13 – bash [11] скрипт, отвечающий за запуск и остановку приложений

Скрипт в зависимости от параметра запускает как бота, так и парсер с нужными параметрами и перенаправляет вывод лога в файлы.

Для обновления чат бота используется скрипт, показанный в листинге 14.

```

#!/bin/bash

./start.sh stop &&
sudo rm -r 1connect-text-bot &&

```

```
git clone https://github.com/1C-Connect/1cconnect-text-bot.git &&
cd 1cconnect-text-bot &&
./build.sh &&
cd .. &&
./start.sh start
```

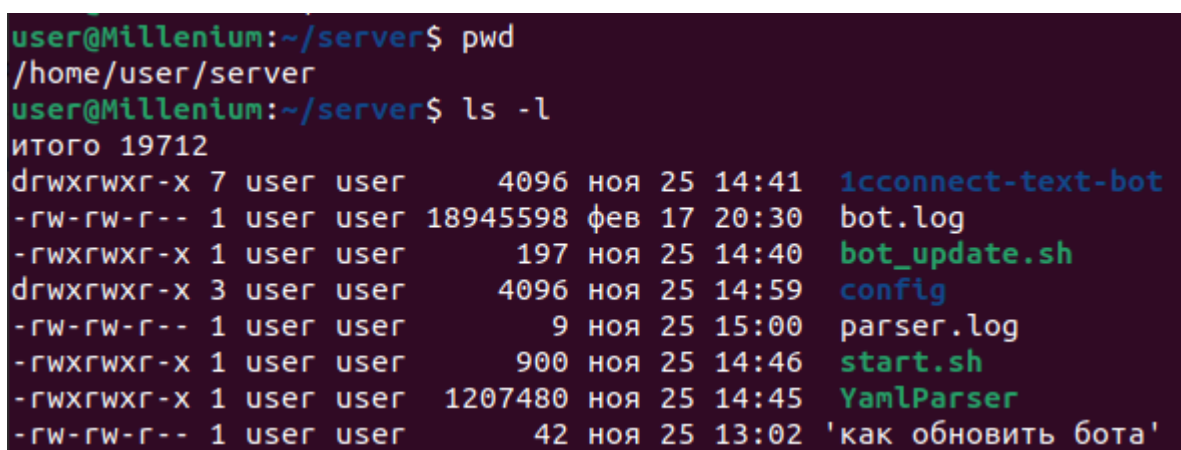
Листинг 14 – bash скрипт обновления чат бота

Описание действий скрипта:

- останавливает чат бот и парсер;
- удаляет папку с ботом;
- клонирует свежую версию бота с репозитория;
- запускает скрипт сборки бота;
- запускает скрипт запуска парсера и чат бота.

Обновление не затрагивает файлы конфигурации.

3.8 Готовая серверная часть чат бота



```
user@Millenium:~/server$ pwd
/home/user/server
user@Millenium:~/server$ ls -l
итого 19712
drwxrwxr-x 7 user user 4096 ноя 25 14:41 1cconnect-text-bot
-rw-rw-r-- 1 user user 18945598 фев 17 20:30 bot.log
-rwxrwxr-x 1 user user 197 ноя 25 14:40 bot_update.sh
drwxrwxr-x 3 user user 4096 ноя 25 14:59 config
-rw-rw-r-- 1 user user 9 ноя 25 15:00 parser.log
-rwxrwxr-x 1 user user 900 ноя 25 14:46 start.sh
-rwxrwxr-x 1 user user 1207480 ноя 25 14:45 YamlParser
-rw-rw-r-- 1 user user 42 ноя 25 13:02 'как обновить бота'
```

Рисунок 8 – директория сервера

На рисунке 8 можно увидеть готовый чат бот и парсер с конфигурацией и файлами логов.

```

user@Millenium:~/server/config$ pwd
/home/user/server/config
user@Millenium:~/server/config$ ls -l
итого 12
drwxrwxr-x 4 user user 4096 ноя 25 14:59 bot
-rw-rw-r-- 1 user user 724 окт 17 22:05 config.yml
-rw-rw-r-- 1 user user 604 ноя 25 14:57 parser_config.yml
user@Millenium:~/server/config$ cat parser_config.yml
# интервал обновления
interval: 60

# имя кнопки раздела, маркер уаml, папка для парсинга, выходной файл уаml
file_paths:
- name: 'Часто задаваемые вопросы'
  marker: '&faq'
  in: "/var/www/html/dokuwiki/data/pages/часто_задаваемые_вопросы/"
  out: "/home/user/server/config/bot/faq/faq.yml"
- name: 'База знаний'
  marker: '&knowledge_base'
  in: "/var/www/html/dokuwiki/data/pages/chat_bot/"
  out: "/home/user/server/config/bot/knowledge_base/knowledge_base.yml"

```

Рисунок 9 – файл конфигурации парсера

В папке config находятся конфигурационные файлы чат бота и парсера, а также директория с конфигурацией меню. На рисунке 9 показан уже готовый и в данный момент используемый файл конфигурации парсера статей.

4 Документирование

Данный программный комплекс реализует удобный интерфейс для заполнения базы знаний, а также простоту поиска информации клиентам.

4.1 Руководство пользователя для сотрудника

Сотрудники при заполнении базы знаний взаимодействуют только с DokuWiki. DokuWiki развернута в локальной сети, и каждый сотрудник имеет к ней доступ под своей учётной записью.

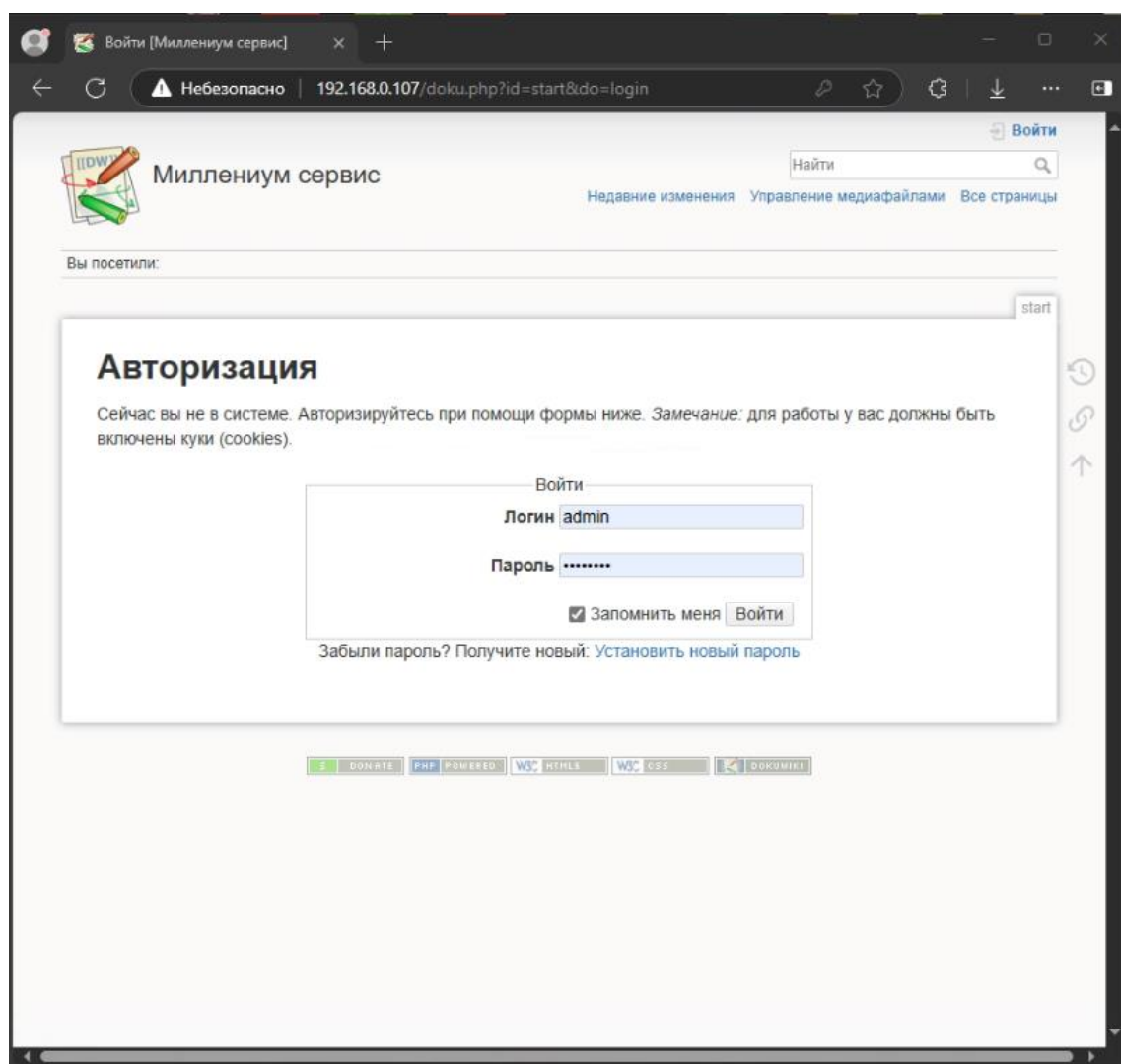


Рисунок 10 – страница авторизации

Каждый вошедший пользователь попадает на стартовую страницу. На ней кратко описано как взаимодействовать с DokuWiki, например создавать новые статьи и разделы. На рисунке 11 показана стартовая страница.

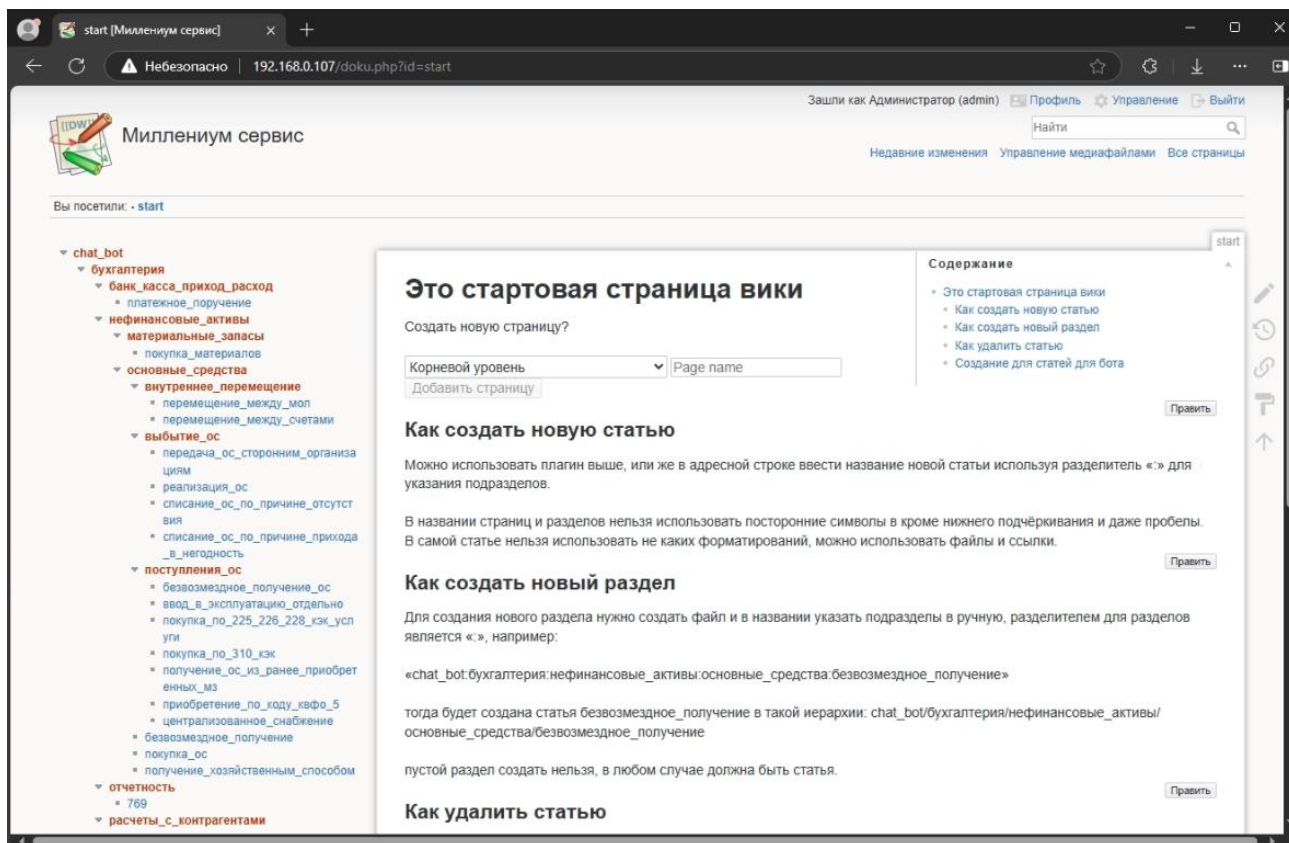


Рисунок 11 – стартовая страница

В левой части окна находится меню, в котором по разделам распределены все статьи. Стоит учитывать, что показ разделов зависит от прав пользователя.

Для создания страницы можно использовать плагин с главной страницы, который позволяет указать раздел и название новой статьи. На рисунке 12 показан выбор раздела для новой страницы из выпадающего меню.

Создать новую страницу?

Корневой уровень

- Корневой уровень
- chat_bot
- бухгалтерия
- банк_касса_приход_расход
- нефинансовые_активы
- материальные_запасы

е в адресной строке ввести н:

Рисунок 12 – создание новой страницы

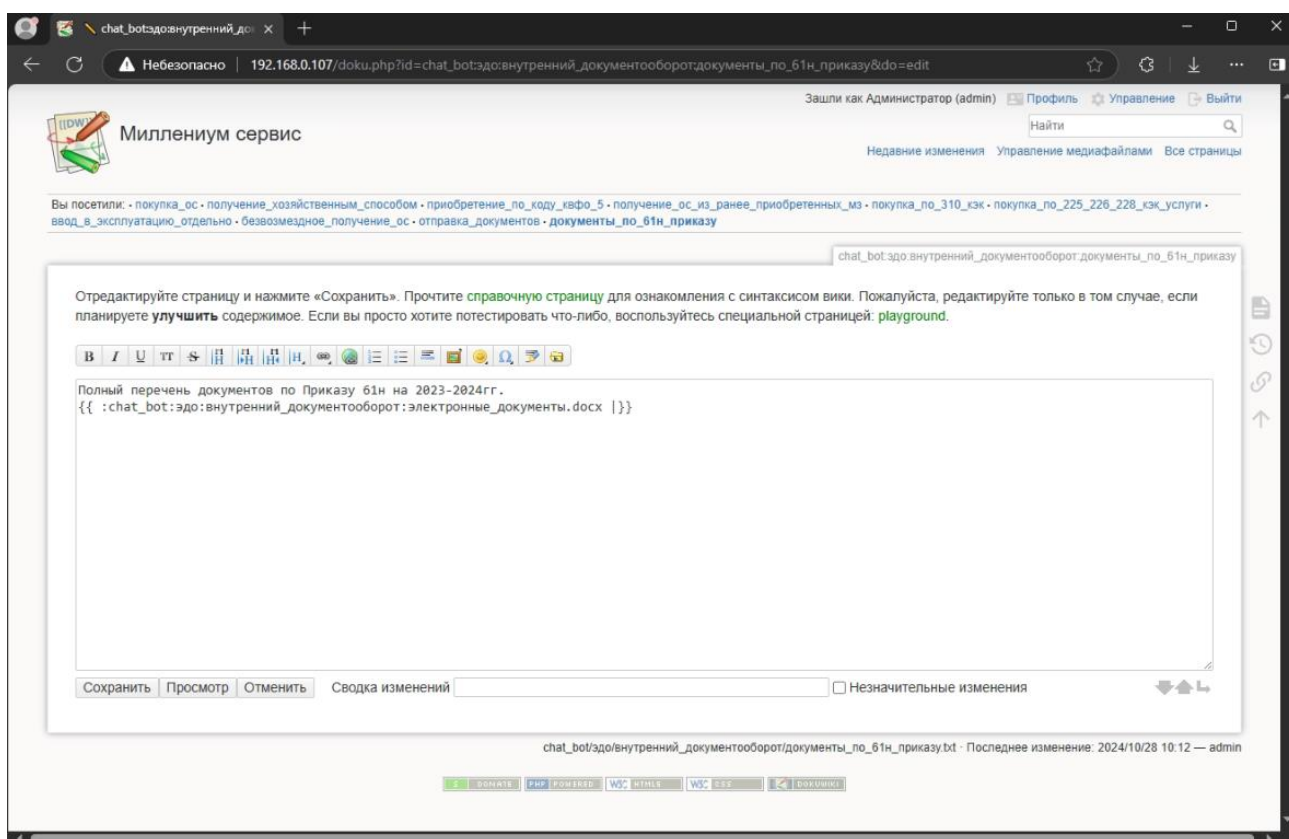


Рисунок 13 – редактор статьи

На рисунке 13 показан редактор статей. При многообразии средств форматирования текста стоит помнить, что 1С Коннект не поддерживает форматирование текста.

4.2 Руководство пользователя для клиента

Клиент взаимодействует только с программой 1С Коннект.

1С Коннект – это многофункциональная платформа для совместной работы и технической поддержки, которая позволяет компаниям общаться, сотрудничать и получать консультации, независимо от местоположения сотрудников. Платформа предоставляет средства для чатов, голосовых и видеозвонков, демонстрацию экрана, удаленного управления компьютерами, проведения конференций и вебинаров, а также обмена файлами. Также через данную платформу клиенты оставляют заявки для специалистов поддержки, либо общаются со специалистами на линиях поддержки «Сопровождение», где проводятся консультации по вопросам, касающимся бухгалтерского и зарплатного учета. На платформе также присутствует связь не только с клиентами, но и также внутри организации, в разделе «Коллеги», где доступны удаленный доступ, ауди и видео звонок.

В начале диалога пользователю показывается кнопка «Меню», которая отвечает за начало диалога с чат ботом. Мобильная версия приложения 1С Коннект показана на рисунке 14.

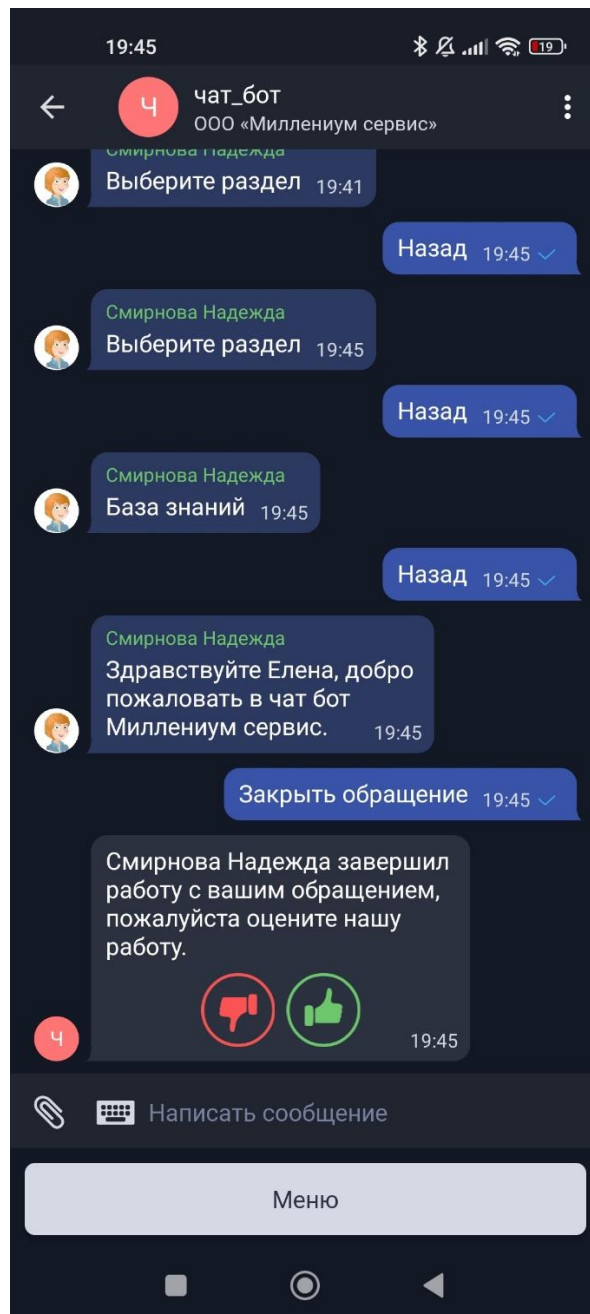


Рисунок 14 – начало диалога

На рисунке 14 показано открытое диалоговое окно с чат ботом. Бот под именем «Смирнова Надежда», после активации кнопки «Меню» выводит приветственное сообщение, которое прописано в конфигурации. Имя бота присваивается стандартно настройками 1С Коннект.

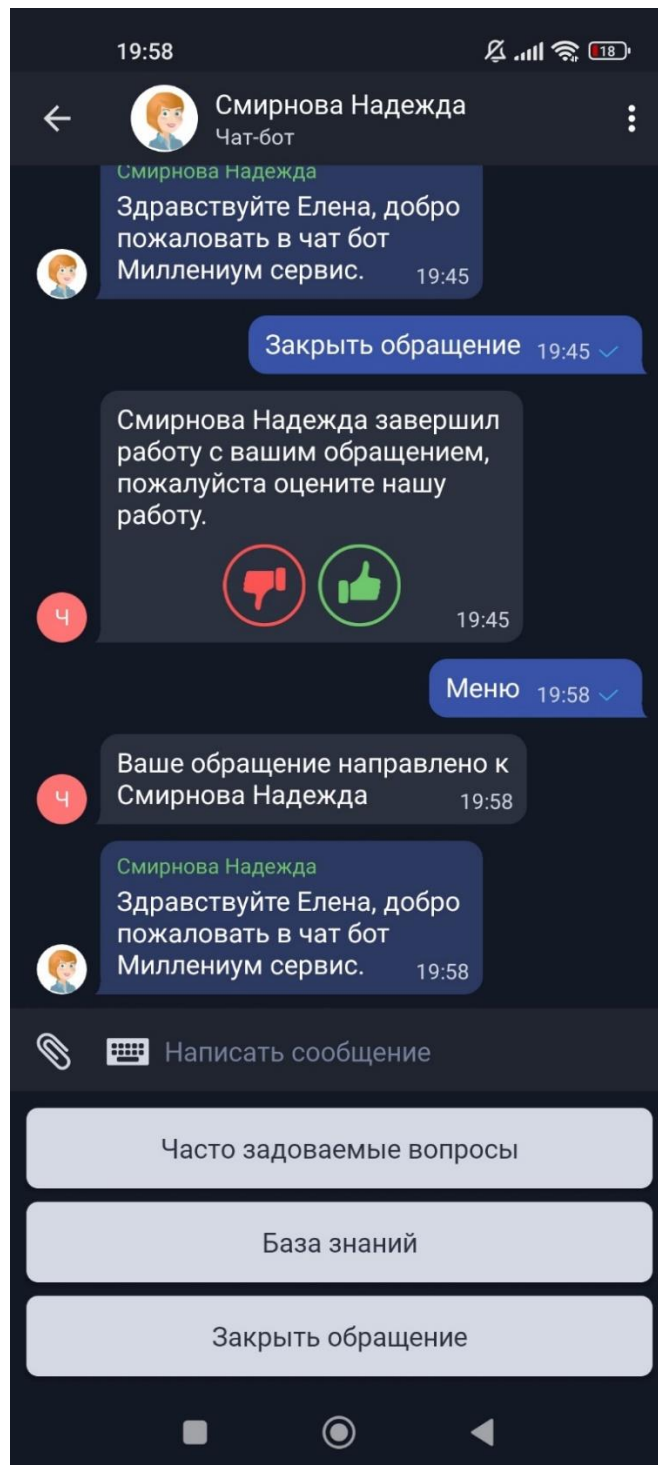


Рисунок 15 – главное меню

В главном меню пользователь может попасть в базу знаний, в часто задаваемые вопросы и закрыть обращение. Если же клиент напишет обращение, не обращаясь к цифровому меню, заявка сразу же будет переадресована на специалиста поддержки.

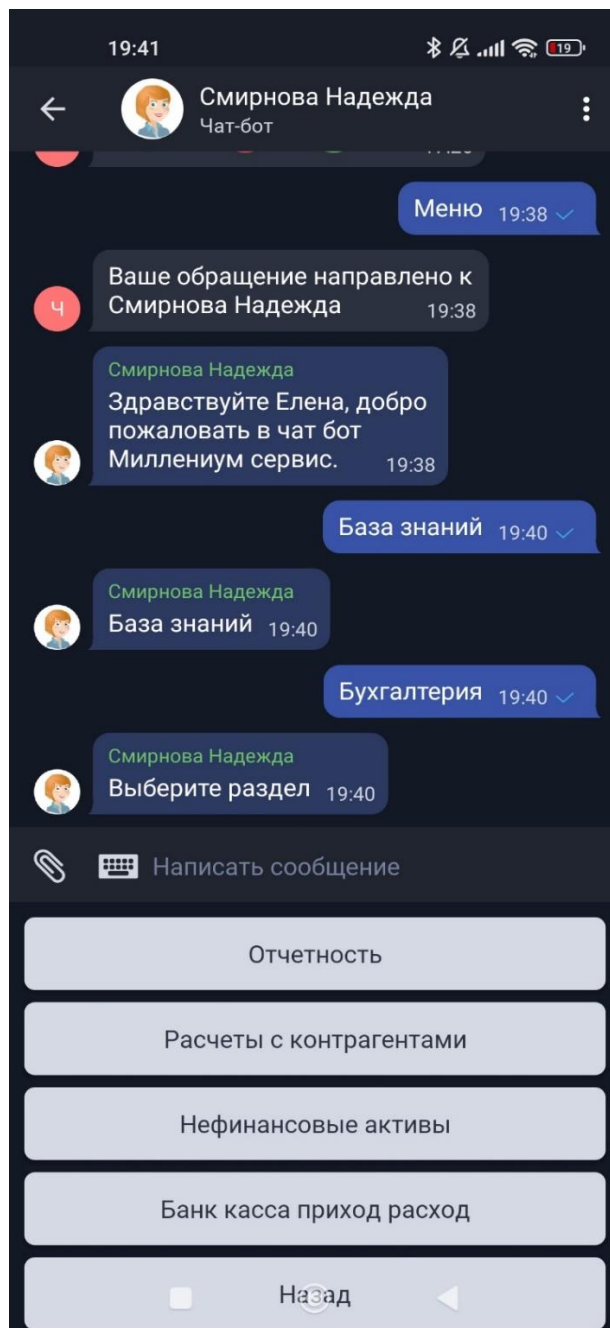


Рисунок 16 – меню

На рисунке 16 показано дальнейшее продвижение по меню, как можно заметить во всех меню кроме главного есть кнопка «Назад». Кнопка назад возвращает в предыдущий раздел вплоть до главного меню.

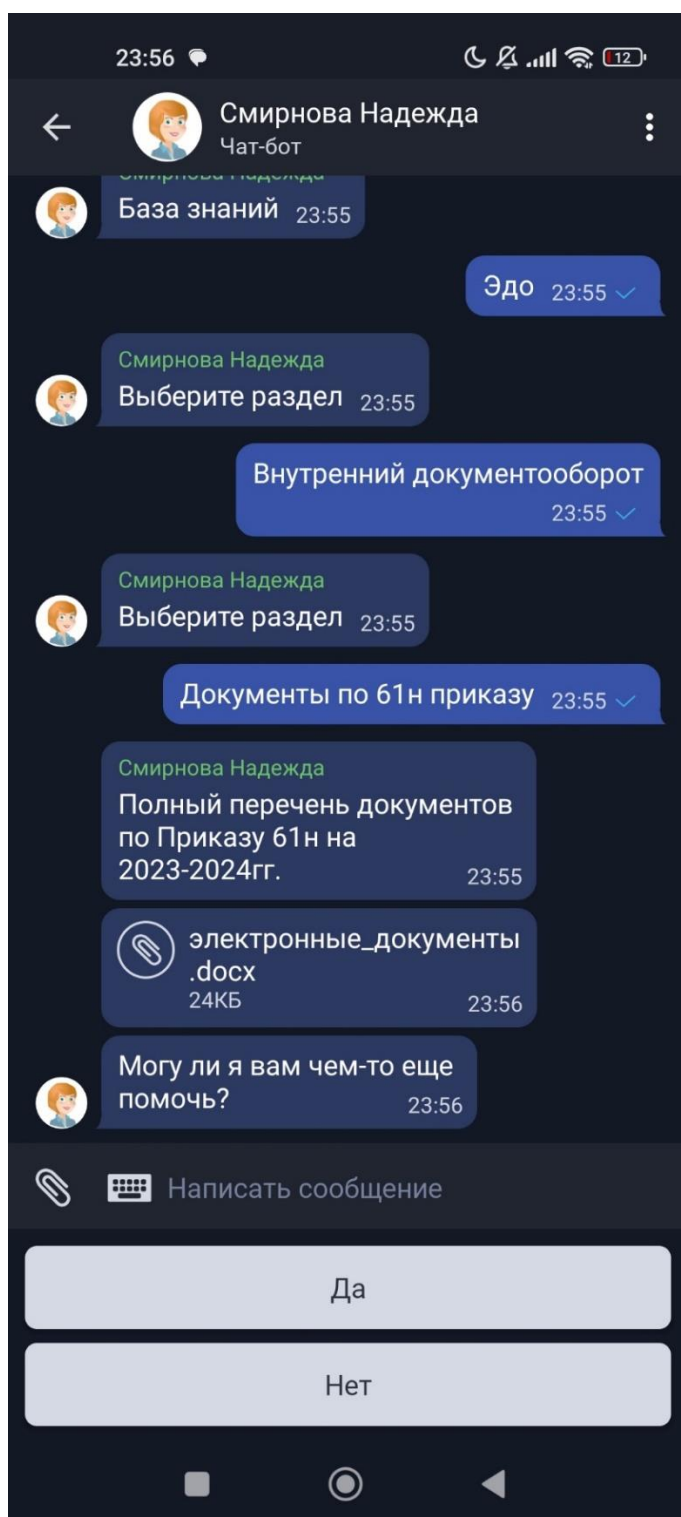


Рисунок 17 – вывод статьи и финальное меню

На рисунке 17 показано финальное меню и статья с прикреплённым файлом в тексте. Также показаны кнопки «Да», «Нет» если же у клиента остались еще вопросы, и он хочет вернуться к цифровому меню.

4.3 Настройка DokuWiki

Пользователь с правами администратора позволяет менять системные настройки вики системы, а также создавать пользователей в системе, которые в дальнейшем будут наполнять информацией базу знаний.

Управление

Ниже вы сможете найти список административных операций, доступных в «Докувики».

- ⚙️ Настройки вики
- ⚙️ Настройки стилей шаблона
- 📖 Просмотр журналов (logs)
- 🔗 Управление дополнениями
- 👤 Управление пользователями
- 🔍 Управление списками контроля доступа
- ↺ Менеджер откаток
- ▶️ Сбор информации о популярности (для загрузки может потребоваться некоторое время)

Дополнительные плагины

- 📄 Перемещение страниц и пространств имён

Рисунок 18 – основное меню управления

На рисунке 18 показано основное меню для управления DokuWiki для администратора.

Параметры «Докувики»

Основные параметры

title Название (заголовок) вики	<input type="text" value="Миллениум сервис"/>
start Имя страницы, используемой как стартовой в каждом пространства имён ⚠	<input type="text" value="start"/>
lang Язык интерфейса	<input type="text" value="ru"/>
template Шаблон (дизайн)	<input type="text" value="dokuwiki (нет других вариантов)"/>
tagline Слоган (если поддерживается шаблоном)	<input type="text"/>
sidebar Боковая панель (если поддерживается шаблоном); пустое поле отключает боковую панель.	<input type="text" value="sidebar"/>
license На условиях какой лицензии будет предоставляться содержимое вики?	<input type="text" value="0"/>
savendir Директория для данных ⓘ	<input type="text" value="/data"/>
basedir Корневая директория (например, /dokuwiki/). Оставьте пустым для автоопределения. ⓘ	<input type="text"/>
baseurl Корневой адрес (URL) (например, http://www.yourserver.ru). Оставьте пустым для автоопределения. ⓘ	<input type="text"/>
cookiedir Директория для cookie. Оставьте пустым для автоопределения. ⓘ	<input type="text"/>
dmode Права для создаваемых директорий	<input type="text" value="0755"/>

Рисунок 19 – основные параметры

На рисунке 19 предоставлена основное окно настроек базы знаний DoKuWiki. В основных настройках необходимо обязательно установить название к базе знаний DoKuWiki, а также язык интерфейса. После основных параметров нужно перейти к разделу «Тонкая настройка», которая показана на рисунке 20.

Тонкая настройка

<small>updatecheck</small> Проверять наличие обновлений и предупреждений о безопасности? Для этого «Докувики» потребует связываться с update.dokuwiki.org.	<input checked="" type="checkbox"/>
<small>userewrite</small> Удобочитаемые адреса (URL)	<input type="checkbox"/> (нет) ▾
<small>useslash</small> Использовать слэш разделителем пространств имён в URL	<input type="checkbox"/>
<small>sepchar</small> Разделитель слов в имени страницы	<input type="checkbox"/> _ ▾
<small>canonical</small> Полные канонические адреса (URL)	<input type="checkbox"/>
<small>fnencode</small> Метод кодирования имён файлов, записанных не ASCII-символами.	<input type="checkbox"/> utf-8 ▾
<small>autoplural</small> Проверять множественную форму имени страницы в ссылках	<input type="checkbox"/>
<small>compress</small> Сжимать файлы CSS и javascript	<input checked="" type="checkbox"/>
<small>cssdatauri</small> Размер в байтах, до которого изображения, указанные в CSS-файлах, должны быть встроены прямо в таблицу стилей для уменьшения избыточных HTTP-запросов. Значение между 400 и 600 байт является хорошим. Укажите 0, чтобы отключить.	512
<small>gzip_output</small> Использовать gzip-сжатие для xhtml (Content-Encoding)	<input type="checkbox"/>
<small>send404</small> Посылать «HTTP 404/Страница не найдена» для несуществующих страниц	<input type="checkbox"/>
<small>compression</small> Метод сжатия для архивных файлов	<input type="checkbox"/> gzip ▾
<small>broken_ja2</small> Возможно, функция ignore_user_abort не работает в вашей системе? Это может привести к потере функциональности индексирования поиска. Эта проблема присутствует, например, в IIS+PHP/CGI.	<input type="checkbox"/>
<small>xsendfile</small> Используете заголовок X-Sendfile для загрузки файлов на веб-сервер? Ваш веб-сервер должен поддерживать это.	<input type="checkbox"/> не используется ▾
<small>renderer_xhtml</small>	

Рисунок 20 – тонкая настройка

На рисунке 20 параметры для тонкой настройки, тут необходимо установить в пункте «Метод кодирования имён файлов» в utf8, если сервер на linux. Это позволит кодировать названия более естественно и без этой настройки не будет работать парсер статей.

Управление списками контроля доступа

Права доступа для Группа:

Пожалуйста, **введите пользователя или группу** в форме выше, чтобы просмотреть или отредактировать права на доступ к странице [start](#).

Краткая справка

На этой странице вы можете добавить или удалить права доступа к пространствам имён и страницам своей вики.

- На панели слева отображены доступные пространства имён и страницы.
- Форма выше позволяет вам просмотреть и изменить права доступа для выбранного пользователя или группы.
- Текущие права доступа отображены в таблице ниже. Вы можете использовать её для быстрого удаления или изменения правил.

Прочтение [официальной документации по правам доступа](#) может помочь вам в полном понимании работы управления

Текущие правила ACL

Страница /Пространство имён	Пользователь / Группа	Права доступа ¹⁾	Удалить
*	@ALL	<input checked="" type="radio"/> Нет доступа <input type="radio"/> Чтение <input type="radio"/> Правка <input type="radio"/> Создание <input type="radio"/> Загрузка <input type="radio"/> Удаление	<input type="checkbox"/>
chat_bot.*	@user	<input type="radio"/> Нет доступа <input type="radio"/> Чтение <input type="radio"/> Правка <input checked="" type="radio"/> Создание <input type="radio"/> Загрузка <input type="radio"/> Удаление	<input type="checkbox"/>
sidebar	@user	<input type="radio"/> Нет доступа <input checked="" type="radio"/> Чтение <input type="radio"/> Правка <input type="radio"/> Создание <input type="radio"/> Загрузка <input type="radio"/> Удаление	<input type="checkbox"/>
start	@user	<input type="radio"/> Нет доступа <input checked="" type="radio"/> Чтение <input type="radio"/> Правка <input type="radio"/> Создание <input type="radio"/> Загрузка <input type="radio"/> Удаление	<input type="checkbox"/>

¹⁾Более высокие права доступа включают в себя более низкие. Права доступа «Создание», «Загрузка» и «Удаление» относятся только к пространствам имён, а не к страницам.

Рисунок 21 – управление списком контроля доступа

На этой странице создаются группы пользователей чем позволяет разграничить доступ к статьям разным пользователям.

DoKuWiki содержит множество плагинов, которые можно установить прямо из страницы управления плагинов. Плагины позволяют упростить множество аспектов, таких как создание страниц, перемещение страниц, создание резервных копий и много другое. Стоит обязательно посетить страницу, показанную на рисунке 22 при первоначальной настройке так как базовый функционал весьма ограничен.

Управление дополнениями

Установленные плагины Установленные шаблоны Поиск и установка Ручная установка

Плагины, установленные в вашей «Докувики». Здесь вы можете их включить/отключить или даже полностью удалить. Также здесь показываются обновления плагинов; обязательно прочтите документацию плагина перед обновлением.

Показать как: включён отключён обновление









	ACL Manager  Andreas Gohr Manage Page Access Control Lists Описание Метки: Ibundled, acl, groups, users	
	Add New Page plugin  Damien Regad, Benjamin Santalucia, Sam Wilson and other contributors Adds a "new page form" to any wiki page. Описание Баг-трекер Метки: button, create, form Обновление: доступна новая версия 2024-12-24	Удалить Обновить Отключить
	Active Directory Auth Plugin  Andreas Gohr Provides user authentication against a Microsoft Active Directory Описание Метки: Ibundled, ad, authentication	Удалить Включить
	LDAP Auth Plugin  Andreas Gohr Provides user authentication against an LDAP server Описание Метки: Ibundled	Удалить Включить

Рисунок 22 – управление плагинами

На этой же странице возможно изменить базовый шаблон чтобы изменить внешний вид всех страниц.

4.4 Установка DokuWiki на сервер

Подразумевается, что на сервере установлен linux, например дистрибутив ubuntu [16]. Не важно есть ли gui или нет, на работу приложений это не влияет. Далее в описании будет подразумеваться что был выбран дистрибутив ubuntu.

Для начала необходимо установить PHP.

PHP – это скриптовый язык программирования, используемый для создания динамических веб-сайтов и веб-приложений. Он выполняется на сервере и генерирует HTML [6], который затем отправляется в браузер пользователя.

Основные особенности PHP:

- встраивается в HTML-код;
- поддерживает работу с базами данных (MySQL, PostgreSQL, SQLite и др.);
- имеет обширную стандартную библиотеку и множество фреймворков (Laravel, Symfony и др.);
- поддерживается большинством веб-серверов (Apache, Nginx);
- является открытым и кроссплатформенным.

DocuWiki это веб приложение и в качестве веб сервера в нашем случае используем Nginx.

Для установки Nginx достаточно выполнить в консоли команду, показанную в листинге 15.

```
sudo apt-get install nginx
```

Листинг 15 – установка Nginx

Далее идёт установка DocuWiki, процесс скачивания и установки показан в листинге 16.

```
cd /var/www
sudo wget https://download.dokuwiki.org/src/dokuwiki/dokuwiki-
stable.tgz
sudo tar -xvzf dokuwiki-stable.tgz
sudo mv dokuwiki-*/ dokuwiki
sudo chown -R www-data:www-data /var/www/dokuwiki
sudo chmod -R 755 /var/www/dokuwiki
```

Листинг 16 – скачивание и установка DocuWiki

Следующим шагом необходимо настроить Nginx, создание и содержимое конфигурационного файла показано в листингах 17 и 18.

```
sudo nano /etc/nginx/sites-available/dokuwiki
```

Листинг 17 – создание конфигурационного файла

```
server {
    listen 80;
    server_name your-domain.com;
    root /var/www/dokuwiki;
    index index.php index.html;

    location / {
        try_files $uri $uri/ @dokuwiki;
    }

    location @dokuwiki {
        rewrite ^/_media/(.*) /lib/exe/fetch.php?media=$1 last;
        rewrite ^/_detail/(.*) /lib/exe/detail.php?media=$1 last;
        rewrite ^/_export/([^/]+)/(.*) /doku.php?do=export_$1&id=$2
last;
        rewrite ^/(.*) /doku.php?id=$1&$args last;
    }

    location ~ /\.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php-fpm.sock;
        fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

```
location ~ /\.ht {
    deny all;
}
}
```

Листинг 18 – содержимое конфигурационного файла

Затем необходимо включить конфигурацию в Nginx, как это показано в листинге 19.

```
sudo ln -s /etc/nginx/sites-available/dokuwiki /etc/nginx/sites-enabled/
sudo nginx -t
```

Листинг 19 – включение конфигурации

Если ошибок нет, то нужно перезапустить Nginx, листинг 20.

```
sudo systemctl restart nginx
```

Листинг 20 – перезапуск Nginx

Установка завершается следующими шагами:

- открыть в браузере «<http://your-domain.com/install.php>»;
- следовать инструкциям установщика DokuWiki;
- после завершения удалить установочный файл, который больше не требуется, команда показана в листинге 21.

```
sudo rm /var/www/dokuwiki/install.php
```

Листинг 21 – удаление установочного файла

Заключение

Тема выпускной квалификационной работы посвящена актуальной проблеме, применение чат бота в системе 1С Коннект как инструмента автоматизации обслуживания клиентов и повышения эффективности коммуникаций с линией поддержки.

Цель выпускной квалификационной работы - повышение удобства и взаимодействия пользователей, автоматизация типовых операций и улучшение эффективности коммуникаций в рабочей среде (за счет внедрения в программе 1С Коннект). Проект направлен на создание удобного, интуитивно понятного интерфейса для пользователей, позволяющего взаимодействовать с чат ботом без ввода текстовых команд, а с помощью готовых кнопок и структурированного меню. Это должно повысить эффективность и скорость выполнения однотипных задач, снизить нагрузку на сотрудников поддержки, минимизировать количество ошибок при вводе текста обращения, а также улучшить общий пользовательский опыт в программе 1С Коннект.

Для достижения поставленной цели в процессе выполнения выпускной квалифицированной работы были решены следующие задачи:

- в первой главе был произведен анализ предметной области, было выбрано оптимальное решение по созданию чат бота, который будет интегрирован с базой знаний DokuWiki, через промежуточный парсер. Такое решение совмещает в себе гибкость, простоту и невысокие требования к ресурсам организации, максимальную доступность для клиентов с помощью применения программы 1С Коннект;
- во второй главе проведено проектирование работы с использованием средств унифицированного языка моделирования UML, что позволило чётко представить логику взаимодействия и внутреннюю

структуру системы чат бота, подтвердив соответствие проектного решения целям и требованиям;

- в третьей главе представлена реализация чат бота на языке Go, в связки с базой знаний DokuWiki, статьи которой парсер преобразует в yaml файл понятный для чата бота и выводит их в виде цифрового меню. Также был разработан bash скрипт, который позволяет обновлять чат бот автоматически;
- в четвертой главе представлена настройка и установка сервера для базы знаний Dokuwiki, а также проведено тестирование работоспособности чат бота на линии сопровождения. Показана графическая составляющая чат бота, а именно цифровое меню.

Разработанное решение позволило значительно упростить и ускорить процесс обработки обращений, обеспечивая оперативный доступ к базе знаний и возможность быстрого получения ответов на часто задаваемые вопросы. Интеграция чат бота в программу 1С Коннект поспособствовало снижению нагрузки на специалистов поддержки, улучшило качество обслуживания и повысило удовлетворённость клиентов.

Список используемых источников

1. 1С ИТС - ИТС 1С [Электронный ресурс]. – Режим доступа: <https://its.1c.ru/> (дата обращения: 23.05.2025).
2. 1С Коннект - 1С Коннект [Электронный ресурс]. – Режим доступа: <https://1c-connect.com/> (дата обращения: 23.05.2025).
3. 1С Коннект API - API 1С Коннект [Электронный ресурс]. – Режим доступа: <https://1c-connect.atlassian.net/wiki/spaces/PUBLIC/pages/1349877768/4.2.+API> (дата обращения: 23.05.2025).
4. Directory - Directory (computing) [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Directory_%28computing%29 (дата обращения: 23.05.2025).
5. GUI - Graphical User Interface [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Graphical_user_interface#:~:text=A%20graphical%20user%20interface%2C%20or,command%20labels%20or%20text%20navigation (дата обращения: 23.05.2025).
6. HTML - HTML [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/HTML> (дата обращения: 23.05.2025).
7. Nginx - Nginx [Электронный ресурс]. – Режим доступа: <https://nginx.org/> (дата обращения: 23.05.2025).
8. PHP - PHP [Электронный ресурс]. – Режим доступа: <https://www.php.net/> (дата обращения: 23.05.2025).
9. UTF-8 - UTF-8 [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/UTF-8> (дата обращения: 23.05.2025).
10. WebSocket - WebSocket [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/sandbox/171066/> (дата обращения: 23.05.2025).
11. bash - Bash [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/47163/> (дата обращения: 23.05.2025).

12. C++ - C++ [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/C%2B%2B> (дата обращения: 23.05.2025).
13. DokuWiki - DokuWiki [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/DokuWiki> (дата обращения: 23.05.2025).
14. Go - Go (язык программирования) [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Go> (дата обращения: 23.05.2025).
15. gRPC - gRPC [Электронный ресурс]. – Режим доступа: <https://grpc.io/>
16. Ubuntu - Ubuntu [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Ubuntu> (дата обращения: 23.05.2025).
17. yaml - YAML [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/YAML> (дата обращения: 23.05.2025).
18. yaml-cpp - yaml-cpp [Электронный ресурс]. – Режим доступа: <https://github.com/jbeder/yaml-cpp> (дата обращения: 23.05.2025).
19. Гущина О.М., Очеповский А.В., Рогова Н.Н. Прикладная информатика. Бизнес-информатика. Выполнение бакалаврской работы: электронное учебно-методическое пособие. Тольятти: Изд-во ТГУ, 2022.
20. Операционная система Linux или Windows - Сравнение Microsoft Windows NT и Linux [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Сравнение_Microsoft_Windows_NT_и_Linux (дата обращения: 23.05.2025).
21. Регулярные выражения - Регулярные выражения [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/545150/> (дата обращения: 23.05.2025).
22. Синглтон - Синглтон [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/811331/> (дата обращения: 23.05.2025).