

**МЕДГРАФ ИИ: семантические инструменты обработки медицинских баз  
данных для создания экспериментальной базы знаний на основе технологий  
искусственного интеллекта**

**(МЕДГРАФ ИИ: СЕМАНТИЧЕСКИЕ ИНСТРУМЕНТЫ)**

**Руководство системного программиста**

**Листов 26**

## АННОТАЦИЯ

Настоящий документ содержит руководство системного программиста «Семантических инструментов обработки медицинских баз данных для создания экспериментальной базы знаний на основе технологий искусственного интеллекта (ЭБ МЗ)», далее – «программа» или «Программный комплекс».

Организация-разработчик: АО «Авикомп Сервисез».

При разработке настоящего документа использовались ГОСТ 19.105-78 «Единая система программной документации. Общие требования к программным документам», ГОСТ 19.106-78 «Единая система программной документации. Требования к программным документам, выполненным печатным способом», ГОСТ 19.503-79 «Единая система программной документации. Руководство системного программиста. Требования к содержанию и оформлению».

## СОДЕРЖАНИЕ

1. Общие сведения о программе .....	4
1.1. Назначение Программного комплекса.....	4
1.2. Требования к техническим средствам.....	6
1.3. Требования к программным средствам.....	7
2. Структура программы.....	9
2.1. Состав Программной системы.....	9
2.1.1. Программа «Semantic Medical Search».....	9
2.1.2. Программа «Clinrek annotator».....	9
2.1.3. Программа «Mediteka».....	10
2.2. Связи между компонентами Программного комплекса.....	10
3. Установка и настройка программы .....	12
3.1. Установка дополнительного системного и специального программного обеспечения .....	12
3.1.1. Установка веб-сервера Apache на примере сервера Ubuntu 20.04. ....	12
3.1.2. Настройка виртуального хоста .....	12
3.1.3. Установка СУБД PostgreSQL.....	13
3.1.4. Установка СУБД ClickHouse.....	14
3.1.5. Установка Python 3.....	14
3.1.6. Установка PHP.....	15
3.1.7. Установка системы виртуализации приложений Docker CE в среду ОС Ubuntu. ....	15
3.1.8. Установка дополнительной утилиты docker-compose для запуска комплексных приложений .....	17
3.2. Установка и настройка Программного комплекса .....	19
4. Проверка программы .....	20
4.1. Вызов и загрузка.....	20
4.2. Завершение работы с интерфейсом Программного комплекса.....	21
5. Сообщения системному программисту .....	22
6. Архивирование и резервное копирование информации .....	23
6.1. Общее описание настройки архивирования и резервного копирования информации .....	23
6.2. Описание восстановления резервных копий из архивов .....	23
7. Мониторинг работоспособности программы.....	24
Перечень обозначений и сокращений .....	25

## 1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

### 1.1. Назначение Программного комплекса

Программный комплекс предназначен для реализации различных методов поиска информации в текстах и получения формализованного представления медицинских знаний (данных), извлекаемых из текстов. В качестве основного источника таких знаний используются клинические рекомендации Минздрава РФ и врачебных сообществ, информация из которых формализуется с использованием унифицированного языка медицинских систем UMLS. Также могут быть использованы другие текстовые источники, которые с экспертной точки зрения содержат релевантную для экспериментальной БД информацию.

Под формализованным представлением медицинской информации понимается граф, в котором в качестве вершин присутствуют выделенные в источниках концепты унифицированного языка медицинских систем UMLS, а в качестве ребер – связи между концептами по смыслу (семантике), соответствующие связям UMLS.

Программа «Semantic Medical Search» реализует различные методы поиска в текстах, в том числе фасетный поиск и использует методы кластеризации и категоризации наборов текстовых документов. Данный интерфейс позволяет по созданным категориям получать как списки релевантных документов, так и списки кластеров, относящихся к данной категории.

Программа «Аннотатор клинических рекомендаций (Clinrek annotator)» автоматизирует выделение в текстах концептов, соответствующих концептам UMLS, а также выделение связей между концептами. Полученный в результате выделения граф преобразуется в xml-представление (xml-файлы), которое далее может быть использовано в различных целях.

Для настройки извлечения концептов программой «Аннотатор клинических рекомендаций» в Программный комплекс включена программа «Редактор словаря разметки (MTD)».

В начальном базовом состоянии словарь разметки содержит значение следующих атрибутов описания концептов UMLS:

- тип концепта (TUI);
- принадлежность к определённому словарю, указанному в качестве значения этого атрибута;
- язык (RUS, ENG);
- вариант написания на данном языке (последовательность символов), обозначающих концепт.

Редактор словаря разметки предназначен для добавления к базовым вариантам других вариантов описания концептов с точки зрения значений атрибутов, перечисленных выше.

Частью Программного комплекса является «Программа ввода и анализа клинических данных – Mediteka», которая обеспечивает ввод клинических данных в Программный комплекс с целью предложения пользователю ранжированных вариантов диагноза, а также других связанных концептов на основе введенных клинических данных. Программа ввода и анализа клинических данных использует «Экспериментальную базу данных медицинской информации», которая является частью Программного комплекса.

Основными функциями Программного комплекса являются:

- доступ к загруженным данным из медицинских справочников;
- отбор и фильтрация медицинских данных по заданным пользователем критериям, а именно: типам концептов данных («Процедура», «Заболевание», «Травма или отравление» и т.п.), наименованиям самих концептов данных («Лающий кашель», «Астения», «Бессонница» и т.п.), группам заболеваний, нарушений, расстройств («Болезни легких обструктивные», «Болезни гортани» и т.п.), кодам МКБ-10;
- выделение в медицинских текстах и текстах клинических рекомендаций медицинских сущностей и семантических связей между ними;
- определения типов концептов и самих концептов, имеющих связь с заданным концептом, например, с концептом «Кашель»;

- отбор и фильтрация документов по заданным пользователем критериям, а именно: типам концептов данных, наименованиям самих концептов данных, листьям и узлам загруженных в базу данных медицинских справочников (реализация функционала семантического поиска);

- предоставление пользователям возможности добавления и корректировки словарных единиц, привязка загружаемых словарных единиц к концептам медицинских данных;

- кластеризация и категоризация медицинских текстов:

- ввод и обработка клинических данных пациента;

- выполнение запросов к базе знаний на основании введенных клинических данных пациента с целью нахождения предполагаемого диагноза и представление пользователю ранжированных извлечённых данных.

## 1.2. Требования к техническим средствам

Программный комплекс представляет собой ряд инструментов, использующих формат веб-приложения для взаимодействия с пользователем.

Программный комплекс функционирует на технических средствах, имеющих следующие характеристики:

1) рабочие станции для приложения-клиента, предназначенные для обеспечения пользователей и доступа к серверам, должны иметь:

- процессоры с частотой не менее 2 ГГц;

- объем оперативной памяти не менее 4 Гб;

- видеокарты с поддержкой OpenGL (при необходимости обеспечения возможности работы с графовым представлением данных);

2) сервер баз данных, совмещенный с сервером приложений – серверная платформа, совместимая с x86-64 и включающая:

- процессор Intel Xeon E5/E7 от v3 до v5 с восемью и более ядрами и тактовой частотой не менее 2 ГГц;

- накопители на жестких магнитных дисках не менее 1500 Гбайт SAS HDD 15000 rpm (или SSD-аналогов), не менее 2-х шт.;
- оперативная память – от 128 Гб;
- сетевой контроллер, совместимый с Gigabit Ethernet или Ethernet 10G ;
- источник бесперебойного питания;

3) локальная вычислительная сеть, предназначенная для обеспечения взаимодействия серверов и рабочих станций, средств резервного копирования и хранения резервных копий.

Серверы должны подсоединяться к локальной вычислительной сети Ethernet посредством типового коммуникационного оборудования, подобранного в соответствии с выбранным стандартом передачи данных. При необходимости получения информации из внешних источников в сети Интернет серверы должны иметь доступ к глобальной вычислительной сети Интернет через локальную вычислительную сеть Ethernet.

### 1.3. Требования к программным средствам

Система функционирует на следующем общем программном обеспечении (ОПО):

- операционная система на базе Linux (Ubuntu 18.04 и более поздних версий);
- Docker CE и Docker Compose — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации;
- СУБД PostgreSQL, версия не ниже 9.4;
- аналитическая СУБД ClickHouse, версия не ниже 19.1;
- веб-сервер Apache 2.4 и выше;
- интерпретатор языка PHP версии 7.0-7.3;
- интерпретатор языка программирования Python (версия 3.4 и выше).

Для обеспечения взаимодействия с Программным комплексом на рабочих станциях пользователей может быть использован браузер Google Chrome или любой другой браузер на ядре Chromium (версия не ниже 85).



## 2. СТРУКТУРА ПРОГРАММЫ

### 2.1. Состав Программной системы

Программный комплекс представляет собой совокупность следующих программных компонентов, разделённых различными визуальными интерфейсами:

- программа семантического поиска «Semantic Medical Search»;
- программа-аннотатор клинических рекомендаций «Clinrek annotator»;
- программа ввода и анализа клинических данных «Mediteka».

#### 2.1.1. Программа «Semantic Medical Search»

Программа «Semantic Medical Search» является набором скриптов, которые обеспечивают загрузку текстовых данных из СУБД PostgreSQL, их кластеризованную структуру и фильтрацию согласно дереву справочника АТХ и произвольно заданного текстового фильтра, созданного пользователем. Также в программе «Semantic Medical Search» предусмотрены механизмы разметки загруженных либо произвольных текстов медицинской направленности сущностями согласно таблице словарных соответствий из СУБД PostgreSQL.

#### 2.1.2. Программа «Clinrek annotator»

Программа «Clinrek annotator» состоит из нескольких подсистем, обеспечивающих разметку текстов документов клинических рекомендаций и дополнительную аналитику выполняемых действий, а именно:

- модуль загрузки и выбора клинических рекомендаций;
- модуль разметки текстов согласно таблице словарных соответствий из СУБД PostgreSQL;
- модуль визуального отображения семантических и фактически употребляемых связей медицинских терминов в рамках медицинских справочников;

– модуль отображения кодов МКБ-10 в соответствии с рабочим фрагментом текста.

### 2.1.3. Программа «Mediteka»

Программа «Mediteka» – это система, построенная на базе подпрограмм с использованием искусственного интеллекта и предназначенная для реализации анализа сочетаний медицинских понятий с возможностью построения справочных систем в области современной медицины. В состав программы «Mediteka» входят:

- модуль сложной фильтрации поиска в графовых структурах данных;
- модуль отображения результатов, формируемых с помощью запросов к СУБД ClickHouse;
- модуль подсказок по предполагаемым направлениям медицины, связанных с текущими результатами поиска;
- модуль отображения кодов МКБ-10 в соответствии с результатами поиска.

## 2.2. Связи между компонентами Программного комплекса

Компоненты Программного комплекса используют единые системы хранения данных (СУБД PostgreSQL и СУБД ClickHouse). Все текстовые документы попадают в реляционную систему, где они могут проходить несколько этапов обработки. Семантический индекс формирует данные, содержащие извлечённую метаинформацию о сформированных кластерах, найденных сущностях и других структурированных данных.

Для извлечения метаинформации из текстов и формирования базы знаний непосредственно используются словари, содержащие унифицированную, выровненную, интегрированную информацию из большого количества источников, единообразно представленную в соответствии с общими принципами представления информации.

Компоненты Программного комплекса являются самостоятельными программными инструментами для выполнения конкретных задач и не зависят от

работоспособности друг друга. Каждый компонент имеет набор API для связи интерфейсов компонентов с серверными методами, возвращающими структурированные данные, преимущественно в формате JSON.

### 3. УСТАНОВКА И НАСТРОЙКА ПРОГРАММЫ

3.1. Установка дополнительного системного и специального программного обеспечения

3.1.1. Установка веб-сервера Apache на примере сервера Ubuntu 20.04.

Для установки веб-сервера Apache необходимо:

- зайти на сервер Ubuntu под пользователем с привилегиями sudo;
- в терминальном окне выполнить команды:

```
sudo apt update
sudo apt install apache2
```

- после установки веб-сервера проверить статус его работы командой:

```
sudo systemctl status apache2
```

В случае удачной установки и запуска веб-сервера отобразится краткий отчет «The Apache HTTP Server» о запущенном веб-сервере.

3.1.2. Настройка виртуального хоста

Чтобы настроить отображение веб-интерфейсов Программного комплекса, необходимо произвести настройку виртуального хоста и выполнить подготовку директории для загрузки программного кода:

- создать директорию для загрузки программного кода командой:

```
sudo mkdir /var/www/mediteka
```

- установить разрешения на эту директорию командой:

```
sudo chmod -R 755 /var/www/mediteka
```

– с помощью любого текстового редактора создать файл «/etc/apache2/sites-available/mediteka.conf» и поместить в него данную запись:

```
<VirtualHost *:8050>
    ServerAdmin webmaster@localhost
    ServerName mediteka
    ServerAlias mediteka.serv
```

```

DocumentRoot /var/www/mediteka
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

```

– активировать виртуальный хост на сервере командой:

```
sudo a2ensite mediteka.conf
```

– в файле «/etc/apache2/ports.conf» прописать строку прослушивания нового порта «listen 8050»;

– перезагрузить сервер Apache командой:

```
sudo systemctl restart apache2
```

– повторить процедуру создания виртуального хоста для директории «/var/www/mediteka/server», назначив в качестве порта для виртуального хоста порт 8850.

### 3.1.3. Установка СУБД PostgreSQL

Для установки собранного пакета необходимо прежде всего выполнить следующие действия на сервере postgresql:

– установить СУБД PostgreSQL командой:

```
sudo apt install postgresql
```

– настроить доступ к СУБД PostgreSQL, для чего с помощью текстового редактора открыть файл «/etc/postgresql/9.6/main/postgresql.conf» (в данном примере предполагается, что идёт установка СУБД версии 9.6) и выставить параметр «listen\_addresses='\*'» для доступа из сети;

– настроить способ авторизации: в файле «/etc/postgresql/9.6/main/pg\_hba.conf» изменить метод аутентификации для локального подсоединения от имени пользователя postgres на «trust (local all postgres trust)», а в файле «/etc/postgresql/9.6/main/pg\_hba.conf» изменить метод аутентификации для всех пользователей из сети на «trust (host all all 0.0.0.0/0 trust)».

Для импорта рабочих таблиц в СУБД PostgreSQL из поставляемого dump-файла «Data-Demo.sql» можно использовать любой удобный менеджер управления

базами данных. Например, бесплатный менеджер с открытым кодом DBeaver (<https://dbeaver.io/>).

### 3.1.4. Установка СУБД ClickHouse

Для установки данной СУБД необходимо добавить в систему Ubuntu дополнительный репозиторий. Для этого в терминале сервера необходимо выполнить следующие команды:

```
sudo apt-get install -y apt-transport-https ca-certificates
dirmngr
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv
8919F6BD2B48D754
echo "deb https://packages.clickhouse.com/deb stable main" | sudo
tee \ /etc/apt/sources.list.d/clickhouse.list
sudo apt-get update
```

После того, как репозиторий успешно установлен в систему, для установки СУБД ClickHouse следует выполнить команды:

```
sudo apt-get install -y clickhouse-server clickhouse-client
clickhouse-common-static
sudo service clickhouse-server start
```

Для импорта рабочих таблиц в СУБД ClickHouse из поставляемого dump-файла «ClickHouse-Demo.tsv» можно использовать любой удобный менеджер управления базами данных (например, DBeaver).

### 3.1.5. Установка Python 3

Серверная платформа Ubuntu 18.04 и более поздние версии поставляются с уже предустановленными средствами программирования на Python 3. Чтобы убедиться, что данные средства присутствуют в системе, а также для внесения необходимых обновлений нужно выполнить в терминале несколько команд:

```
sudo apt update
sudo apt -y upgrade
python3 -V
```

После выполнения последней команды в окне терминала должна появиться версия установленного в системе программного средства Python.

Для управления программными пакетами в среде Python необходимо установить менеджер пакетов pip командой:

```
sudo apt install -y python3-pip
```

### 3.1.6. Установка PHP

Средства разработки на языке программирования PHP для веб-сервера Apache устанавливаются следующими командами:

```
sudo apt update
sudo apt install php libapache2-mod-php
sudo apt install php-pgsql libapache2-mod-authnz-pam php-cgi php-fpm
php-mbstring php-xml php-gd
```

Для применения настроек необходимо перезапустить веб-сервер Apache командой:

```
sudo systemctl restart apache2
```

3.1.7. Установка системы виртуализации приложений Docker CE в среду ОС Ubuntu.

Для установки системы виртуализации приложений Docker CE в среду ОС Ubuntu необходимо выполнить следующие действия:

- зайти на сервер под пользователем с привилегиями sudo;
- в терминальном окне выполнить команды:

```
sudo apt-get remove docker docker-engine docker.io conta.inerd
runc
```

(если все выполнено успешно, то из системы будут удалены ранее установленные версии программы);

- добавить необходимые зависимости для установки программы:

```
sudo apt-get update
sudo apt-get install \
    ca-certificates \
```

```
curl \  
gnupg \  
lsb-release
```

– добавить официальный GPG-ключ для проверки подлинности пакетов:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg  
| sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-  
keyring.gpg
```

– установить стабильный репозиторий в качестве основного, из которого будут производиться установка и обновление программы:

```
echo \  
"deb [arch=$(dpkg --print-architecture) signed-  
by=/usr/share/keyrings/docker-archive-keyring.gpg]  
https://download.docker.com/linux/ubuntu \  
$(lsb_release -cs) stable" | sudo tee  
/etc/apt/sources.list.d/docker.list > /dev/null
```

– установить программу Docker CE:

```
sudo apt-get update  
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

– проверить, что сервис в системе запущен (он запускается автоматически после успешной инсталляции программы):

```
sudo service docker status
```

при этом на экран будет выведена следующая информация:

```
docker.service - Docker Application Container Engine  
Loaded:loaded
```

Для проверки успешной установки программы необходимо выполнить команду:

```
sudo docker run hello-world
```

при этом на экране терминала должен появиться текст «hello-world».



### 3.1.8. Установка дополнительной утилиты docker-compose для запуска комплексных приложений

Для установки дополнительной утилиты docker-compose необходимо выполнить следующие действия:

- скачать утилиту, выполнив следующую команду:

```
sudo curl -L
```

```
"https://github.com/docker/compose/releases/download/1.29.2/docker-  
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

- применить требуемые атрибуты к скаченному бинарному файлу:

```
sudo chmod +x /usr/local/bin/docker-compose
```

- проверить, что команда доступна для выполнения из командной строки:

```
sudo docker-compose --version
```

Запуск модулей программы (компонентов) осуществляется при помощи виртуализации Docker CE. Программная система частично использует модули, поставляемые в виде файла-образа утилиты docker-compose. Чтобы их использовать, требуется скопировать поставляемый образ модуля программы «mediteka.img.gz» на целевой сервер, где установлены программа Docker CE и утилита docker-compose, а затем выполнить импорт образа модуля с последующим запуском образа.

Для импорта образа модуля требуется скопировать его на целевой сервер в любую доступную директорию и выполнить следующую команду:

```
cat mediteka.img.gz | docker load
```

Далее следует создать shell-скрипт, в котором напрямую прописать параметры запуска модуля, порты, параметры окружения, подключаемые папки файловой системы и другие доступные параметры запуска контейнера. Другой метод запуска образа (предпочтительный) – использование утилиты docker-compose, работающей на основе конфигурационного файла «docker-compose.yml», в котором задаются такие же параметры запуска, как и в случае их определения в shell-скрипте, а также предоставляется возможность множественного запуска нескольких связанных друг с другом контейнеров приложений, автоматический старт, перезапуск в случае сбоев и многое другое.

На рис. 1 приведен пример файла «docker-compose.yml» для совместного запуска БД ClickHouse (clickhouse-server) и прокси-кэша к ней (chproxy-service).

```

version: '3'

services:

  clickhouse-server:
    image: yandex/clickhouse-server
    volumes:
      - ./db/clickhouse-server-db:/var/lib/clickhouse
      - ./conf/config.xml:/etc/clickhouse-server/config.xml
      - ./conf/users.xml:/etc/clickhouse-server/users.xml
      - ./logs:/var/log/clickhouse-server
    ports:
      - 8128:8123
      - 9000:9000
    ulimits:
      nofile:
        soft: "262144"
        hard: "262144"
      shm_size: '2gb'
    restart: always

  chproxy-service:
    image: chproxy:dev
    ports:
      - 8123:8123
    volumes:
      - ./8123_proxy_cfg.yml:/proxy_cfg.yml
      - ./chcache:/cache
    restart: always
    depends_on:
      - clickhouse-server
  
```

Рисунок 1 - Пример файла «docker-compose.yml»

Выполнение команд должно осуществляться в рабочей директории, где расположен файл конфигурации.

Для запуска всех сервисов следует использовать команду:

```
sudo docker-compose up -d
```

Для проверки статуса всех сервисов, прописанных в файле «docker-compose.yml», выполнить команду:

```
sudo docker-compose ps
```

Остановить выполнение всех сервисов можно при помощи команды:

```
sudo docker-compose down
```

### 3.2. Установка и настройка Программного комплекса

Для обеспечения работы Программного комплекса необходимо, помимо развёртывания «дампов» для СУБД и установки образа в docker-контейнер (см. пункты 3.1.3, 3.1.4 и 3.1.7), разархивировать поставляемый файл «mediteka.tar.gz», поместив данный архив в папку «/var/www/mediteka» и выполнив команду:

```
tar -xvf mediteka.tar.gz
```

Далее необходимо проверить настройки IP-адресов и путей для API-запросов. Для этого с помощью наиболее удобного для пользователя редактора БД найти в СУБД PostgreSQL базу данных с именем «umls\_2021AA» и схему «meta». Затем отобразить данные таблицы «vars» и скорректировать данные согласно текущим настройкам сервера. Данными, подлежащими исправлениям, являются IP-адреса, которые необходимо будет заменить на IP рабочего сервера (рис. 2).

key	val	description
umls_server	10.15.1.30	Сервер с БД UMLS
umls_port	5432	Порт БД
umls_db	umls_2021AA	Имя БД
umls_user	postgres	Пользователь БД
umls_password	password	Пароль БД
umls_scheme	umls_2021aa	Схема с БД UMLS
main_vocabulary	med_terms_dictionary	Пользовательский словарь (имя таблицы)
api_diag	http://10.15.1.30:8501/	Получение диагнозов по симптомам + переводы
api_documentor	http://10.15.1.30:9085/v1/document/query	Семантический поиск
api_annotate	http://10.15.1.30:8999/	Функции аннотатора
api_links	http://10.15.1.30:5001/api/	Связи в предложении и абзаце

Рисунок 2 - Пример данных из БД

## 4. ПРОВЕРКА ПРОГРАММЫ

Проверка корректности установки программ с графическим пользовательским интерфейсом осуществляется путём выполнения следующих действий:

- запуск установленного на рабочей станции пользователя веб-браузера;
- набор необходимого адреса программного компонента в строке вызова и нажатие клавиши «Enter».

### 4.1. Вызов и загрузка

Проверка Программного комплекса заключается в тестировании работы интерфейсов, вызываемых по адресам в строке браузера:

- [http://\(IP-адрес сервера\):8050/semanticms](http://(IP-адрес сервера):8050/semanticms) (для вызова программного компонента «Semantic Medical Search»);
- [http://\(IP-адрес сервера\):8050/clinrek](http://(IP-адрес сервера):8050/clinrek) (для вызова программного компонента «Clinrek annotator»);
- [http://\(IP-адрес сервера\):8050/mediteka](http://(IP-адрес сервера):8050/mediteka) (для вызова программного компонента «Mediteka»).

Для вызова необходимого программного компонента необходимо выполнить следующие действия:

- запустить веб-браузер, установленный на рабочей станции пользователя;
- набрать один из адресов программного компонента в адресной строке браузера (например: [http://\(IP-адрес сервера\):8050/clinrek](http://(IP-адрес сервера):8050/clinrek)) и клавишу «Enter».

Откроется интерфейс выбранной страницы (рис. 3).

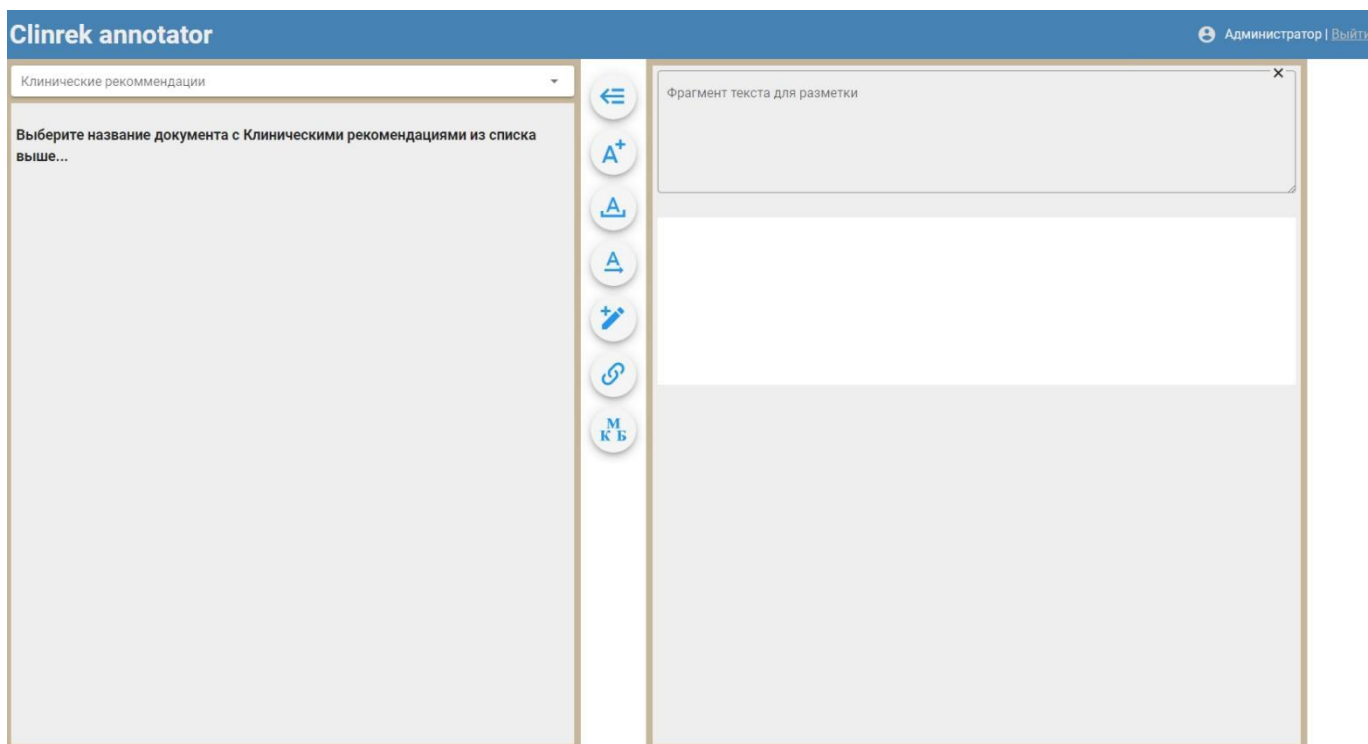


Рисунок 3 - Пример отображения интерфейса

#### 4.2. Завершение работы с интерфейсом Программного комплекса

Для выхода из интерфейса Программного комплекса необходимо закрыть вкладку браузера.

## 5. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

Во время работы Программный комплекс выводит сообщения о ходе работы в протокольные log-файлы, соответствующие запущенному процессу.

Ряд предупреждающих сообщений может быть выведен на интерфейс компонента программы или в консоль браузера.

## 6. АРХИВИРОВАНИЕ И РЕЗЕРВНОЕ КОПИРОВАНИЕ ИНФОРМАЦИИ

Для предотвращения глобальных сбоев в работе Программного комплекса необходимо выполнять резервное копирование информации перед использованием обновлений и изменений.

### 6.1. Общее описание настройки архивирования и резервного копирования информации

Архивы баз данных формируются стандартными процедурами СУБД PostgreSQL и СУБД ClickHouse либо инструментами, использующими стандартные процедуры создания архивных копий данных из БД.

Архивирование серверных скриптов и файлов, формирующих визуальные интерфейсы, выполняется командами в терминальном окне на сервере:

```
cd /var/www/mediteka  
tar -cvpzf mediteka_2022_03_22_14_59.tar.gz * .htaccess
```

(название архивного файла необходимо задавать в соответствии с текущей датой и временем).

После завершения архивирования в директории «/var/www/mediteka» появится архив с последними изменениями в визуальной части интерфейсов программных компонентов. Определив заранее место для хранения архивных данных, данный файл необходимо переместить в указанное место.

### 6.2. Описание восстановления резервных копий из архивов

Восстановление программных компонент и резервных копий БД из архивов следует выполнять согласно описания установки Программного комплекса (см. раздел 3 «Установка и настройка программы»).

## 7. МОНИТОРИНГ РАБОТОСПОСОБНОСТИ ПРОГРАММЫ

Автоматический мониторинг Программного комплекса не предусмотрен.



## ПЕРЕЧЕНЬ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

- API – (англ. Application Programming Interface) программный интерфейс приложения;
- CPU – (англ. Central Processing Unit) центральный процессор;
- JSON – (англ. JavaScript Object Notation) текстовый формат обмена данными, основанный на JavaScript;
- SQL – (англ. Structured Query Language) «язык структурированных запросов», язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных;
- БД – база данных;
- ОПО – общее программное обеспечение;
- ОС – операционная система;
- ПО – программное обеспечение;
- СУБД – система управления базами данных;
- ЭБ МЗ – экспериментальная база медицинских знаний с применением технологий искусственного интеллекта.

