

# Руководство по установке программного продукта

«Система поддержки принятия врачебных решений ТОП-3  
диагнозов на основе данных электронной истории болезни»

версия 1.1.0

# Содержание

<b>Содержание</b>	2
<b>Введение</b>	3
<b>1. Назначение Программного продукта</b>	4
<b>2. Архитектура Программного продукта</b>	5
<b>3. Ограничения работы Программного продукта</b>	6
<b>4. Локальная инсталляция программного продукта</b>	7
4.1. Установка контейнера	7
4.2. Обновление контейнера	9
<b>5. Описание API</b>	10

# Введение

Настоящий документ содержит краткое описание программного продукта «Система поддержки принятия врачебных решений ТОП-3 диагнозов на основе данных электронной истории болезни» и способов взаимодействия с ним.

В основе программного продукта лежит предсказательная модель, которая обрабатывает входной текст в свободной форме в кодировке UTF-8 (симптомы, жалобы, анамнез и т.п.) и выполняет предсказание диагнозов.

В качестве результата модель возвращает 3 наиболее вероятных диагноза из 265 возможных вариантов. Диагнозы соответствуют кодам второго уровня МКБ-10, т. е. идентификатор кода до точки. С более подробным описанием процедуры разработки модели и проведённых экспериментов можно ознакомиться в статье <https://arxiv.org/abs/2007.07562>. Стоит отметить, что текущая версия модели не производит дополнительных проверок полноты и “тематической корректности” входного текста и всегда возвращает наиболее вероятные диагнозы.

Целевая аудитория сервиса:

- врачи общей практики;
- терапевты;
- амбулаторно-поликлиническое звено.

# 1. Назначение Программного продукта

Программный продукт предназначен для поддержки принятия врачебного решения по предварительному диагнозу для врача.

Программный продукт обрабатывает описание жалоб пациента в текстовом формате в кодировке UTF-8, выполняет оценку вероятности диагнозов.

В качестве результата Программный продукт возвращает 3 наиболее вероятных диагноза из 1277 возможных вариантов 265 групп заболеваний (именно такое количество самых распространённых диагнозов покрывает 95% причин обращений в поликлиники).

Диагнозы соответствуют кодам второго уровня МКБ-10, т. е. идентификатор кода до точки. Текущая версия Программного продукта не производит дополнительных проверок полноты и «тематической корректности» входного текста и всегда возвращает наиболее вероятные диагнозы. Точность результатов не превышает 80%.

Программный продукт охватывает все виды нозологий, по которым пациенты обращаются в городские поликлиники, и покрывает 95% причин обращений в поликлиники Москвы.

Полученный список диагнозов ранжирован по частоте выставленных диагнозов.

Программный продукт разработан на основе данных из различных информационных баз (базы государственных и негосударственных медицинских учреждений), использованы данные о 3 976 888 визитов 1 123 541 пациентов, в период за 15 лет.

## 2. Архитектура Программного продукта

Верхнеуровневая архитектура Программного продукта и схема взаимодействия с Клиентским ПО изображена на рисунке 1.

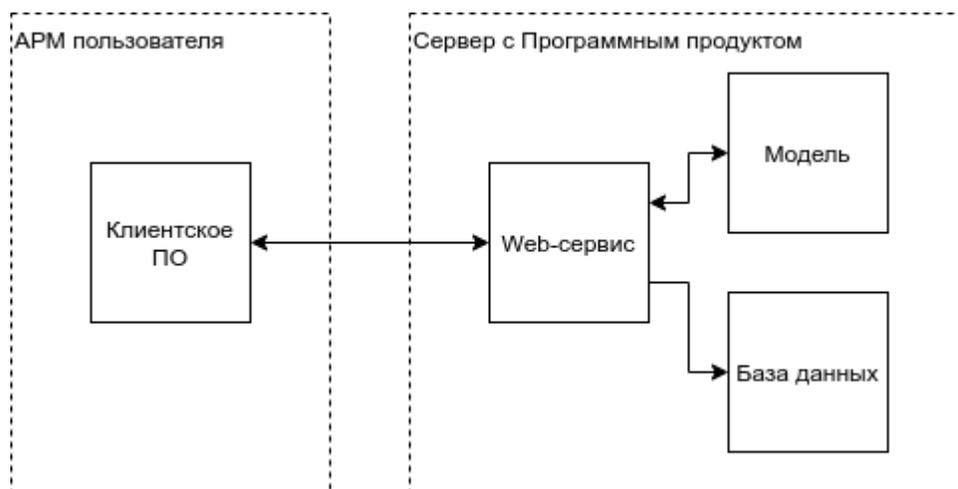


Рисунок 1

Программный продукт включает в себя следующие блоки, характеристики которых перечислены в таблице 1 ниже.

Таблица 1

Название блока	Описание
Web-сервис (далее по тексту - сервис)	Блок, реализующий логику работы с API Программного продукта - получение запросов от Клиентского ПО и передачу ответов на них, сохранение служебной информации в базу данных.
Модель	Блок, реализующий функционал предсказания предварительных диагнозов по описанию симптомов, жалоб и анамнеза.
База данных	Временное локальное хранилище, в котором собираются Данные для их дальнейшей обработки перед выгрузкой в Систему в рамках Запросов.

### 3. Ограничения работы Программного продукта

Использование Программного продукта врачами возможно только в случае интеграции Программного продукта и МИС, в результате интеграции процедура работы врача в МИС не должна измениться.

Для корректной работы Программного продукта врачу перед выбором диагноза МКБ необходимо заполнить поле в МИС: жалобы. Рекомендуемое кол-во жалоб - не менее 6.

Далее, в ходе приема пользователь может в любой удобный ему момент (зависит от реализации в МИС) запросить вероятные диагнозы у программного продукта и выбрать диагноз из предложенного списка.

## 4. Локальная инсталляция программного продукта

Минимальные технические требования к серверу:

- a. CPU – 3 ГГц x 4;
- b. RAM – 16 ГБ;
- c. HDD – 100 ГБ;
- d. ОС – Ubuntu 18 LTS 64-бит и выше;

Все действия описанные ниже, также представлены в **README.md** файле который прилагается к скриптам.

### 4.1. Установка контейнера

1. Необходимо скачать архив **top3.zip** по ссылке <https://files.sbermed.ai/s/NG99cnHxWSiyjmb>
2. Создать папку **scripts**, распаковать в неё файлы из загруженного архива. Перейти в папку scripts и открыть в ней терминал. Далее необходимо выдать скриптам права на исполнение, выполнив команду:

```
sudo chmod +x *.sh
```

3. Для установки контейнера необходимо выполнить команду:

```
./prepare.sh
```

Дождитесь выполнения команды, это может занять несколько минут. В ответ будет выведен результат (см. рисунок №2) :

```
Loaded image: top3-wrapper_web:latest
Loaded image: top3-wrapper_top3:latest
Loaded image: top3-wrapper_db:latest
Image DCMGATE (top3-wrapper_web) loaded successful
Image TOP3 (top3-wrapper_top3) loaded successful
Image DB (top3-wrapper_db) loaded successful
#####
Preparation was successful
#####
```

Рисунок №2 Результат выполнения команды

4. Далее остается запустить скрипт для старта контейнеров.

```
./start_container.sh
```

Будет выведен результат выполнения команды (см. рисунок №3)

```
Start containers
Building with native build. Learn about native build in Compose here: https://docs.docker.com/go/compose-native-build/
Creating network "top3_default" with the default driver
Creating top3_top3_1 ... done
Creating top3_db_1 ... done
Creating top3_web_1 ... done
```

Рисунок №3 Результат выполнения команды

5. После запуска последнего скрипта, необходимо убедиться, что контейнер находится в состоянии *UP*:

```
sudo docker-compose ps
```

В ответ будет выведен результат (см. рисунок №4):

Name	Command	State	Ports
top3_db_1	docker-entrypoint.sh postgres	Up	5432/tcp
top3_top3_1	gunicorn -b 0.0.0.0:50001 ...	Up	0.0.0.0:8000->50001/tcp, :::8000->50001/tcp, 6006/tcp, 8888/tcp
top3_web_1	/usr/src/app/entrypoint.sh ...	Up	0.0.0.0:50001->8000/tcp, :::50001->8000/tcp

Рисунок №4 Результат выполнения команды

## 4.2. Обновление контейнера

В случае, если модель была обновлена и вы получили новый дистрибутив, необходимо обновить систему. Для этого последовательно выполните все шаги, описанные в главе **4.1. Установка контейнера**.

В случае, если произошла перезагрузка системы, то достаточно запустить команду:

```
./start_container.sh
```

## 5. Описание API

Взаимодействие с сервисом происходит удаленно через интерфейс REST API по протоколу HTTP. Сервис предусматривает обязательную аутентификацию пользователей и ведёт историю обращений с фиксацией входных и выходных данных.

Базовый URL для всех запросов - `http://localhost/api`, если запросы отправляются с сервера, на котором развернут сервис, или на адрес `http://{ip адрес вашего сервера}/api`.

Подробнее про описание API см. документе - **Требования к интеграции сервиса «Система поддержки принятия врачебных решений ТОП-3 диагнозов на основе данных электронной истории болезни».**