

Требования к интеграции сервиса

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

версия 1.1.0

Содержание

Введение	2
1. Назначение программного продукта	4
2. Архитектура Программного продукта	5
3. Ограничения работы Программного продукта	7
4. Требования к установке программного продукта	8
5. Информация о протоколах обмена данными МИС и Программного продукта	9
6. Функциональные требования к МИС, интегрируемой с Программным продуктом	9
7. Пример интерфейса интегрированной МИС с ПО	10
8. Описание API	12
8.1. Получение предсказания	13
8.1.1. Входные данные	13
8.1.2. Выходные данные	13
8.1.3. Пример запроса и ответа	14
8.2. Передача обратной связи	15
8.2.1. Входные данные	15
8.2.2. Выходные данные	15
8.2.3. Пример запроса	15
9. Техническая поддержка	17

Введение

Настоящий документ содержит краткое описание сервиса «Система поддержки принятия врачебных решений ТОП-3 диагнозов на основе данных

электронной истории болезни» и способов взаимодействия с ним посредством программного интерфейса.

В основе сервиса 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 (по нашим оценкам такое количество самых распространённых диагнозов покрывает больше 90% всех обращений). 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 [arXiv:2007.07562](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 симптомов, жалоб и анамнеза 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

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

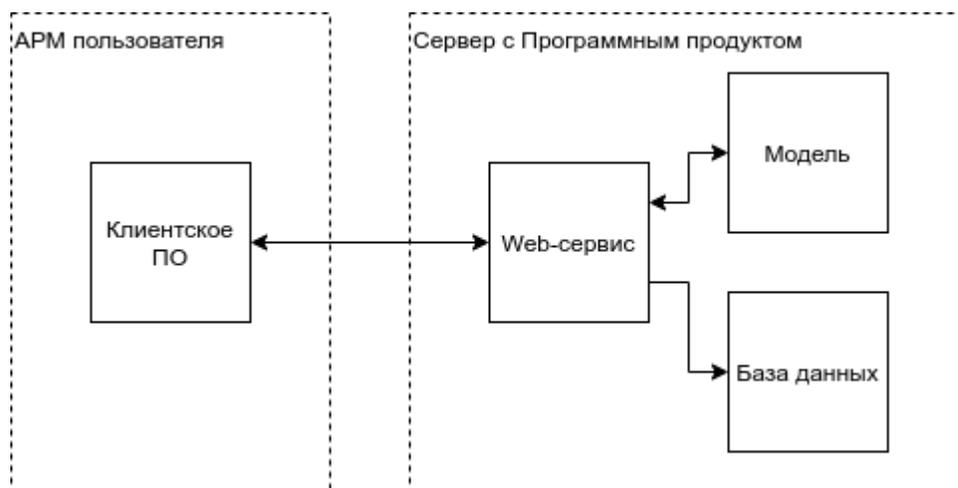


Рис. 1 - Архитектура

2. Клиентская часть реализуется в МИС, используемой мед. учреждением, и является внешней по отношению к сервису.
3. 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.

4. Требования к установке программного продукта

Сервис является программным обеспечением по модели распространения SaaS «программное обеспечение как услуга» и не предусматривает наличие физических носителей и его инсталляцию в качестве клиентского приложения на автоматизированное рабочее место сотрудников медицинской организации или в качестве серверной части на серверные электронно-вычислительные машины таких медицинских организаций.

5. Информация о протоколах обмена данными МИС и Программного продукта

Сетевой обмен данными между МИС и ПО строится на базе протокола HTTP(S) с использованием JSON в качестве формата передаваемых данных. Аутентификация МИС в ПО происходит посредством аутентификационного ключа.

6. Функциональные требования к МИС, интегрируемой с Программным продуктом

Функциональные требования к МИС, прошедших интеграцию с Сервисом:

- МИС должна поддерживать в ЭМК текстовые поля с описанием жалоб и анамнеза заболевания пациента.
- МИС должна в автоматическом фоновом режиме до выбора пользователем из списка диагноза переадресовывать из ЭМК пациента в Сервис текстовую информацию о жалобах и анамнезе заболевания пациента в кодировке UTF-8 в формате JSON через API системы.
- МИС должна в автоматическом фоновом режиме принимать результат работы Сервиса в виде 3-х наиболее вероятных спрогнозированных диагнозов и обновлять поле с выбором списка диагнозов.
- МИС после интеграции с Сервисом должна обеспечивать хранение результатов работы Сервиса.
- МИС должна опционально предусматривать функцию передачи обратной связи (фактически выбранного врачом диагноза).

7. Пример интерфейса интегрированной МИС с ПО

Сервис не имеет собственного графического интерфейса. После интеграции Сервиса в МИС, которые используются в медицинских организациях, взаимодействие с Сервисом со стороны пользователей осуществляется в автоматическом фоновом режиме без непосредственного участия пользователя. Пользователь получает от Сервиса готовый результат без необходимости вручную загружать ЭМК пациента.

Ниже представлен пример внешнего вида интерфейса интегрированной МИС (Рис. 2). В каждом конкретном случае внешний вид может отличаться от представленного.

The screenshot displays a web interface for creating a medical protocol. On the left is a sidebar with the following menu items: Жалобы (highlighted in yellow), Анамнез заболевания, Общий осмотр, Диагноз, Опровержение диагноза, Степень тяжести по COVID-19, Лекарственные назначения, and Рекомендации. The main area is titled 'Создание протокола' and contains a dropdown menu with the text 'Выберите шаблон из списка'. Below this is a section for 'На момент осмотра жалобы*' with two radio buttons: 'не предъявляет' and 'предъявляет' (the latter is selected and highlighted in yellow). A text input field labeled 'Жалоба' contains the text 'кашель, насморк'. Below the input field is a red-bordered box containing the following text: 'Возможные диагнозы J06 - Острые инфекции верхних дыхательных путей множественной и неуточненной локализации', 'J00 - Острый назофарингит (насморк)', and 'J20 - Острый бронхит'. A note below the diagnoses states: 'Возможные диагнозы выбраны системой как наиболее часто встречающиеся при данных жалобах'. At the bottom of the form are three expandable sections: '+ Анамнез заболевания', '+ Общий осмотр', and '+ Диагноз'. A 'Сохранить' button is located at the bottom right of the interface.

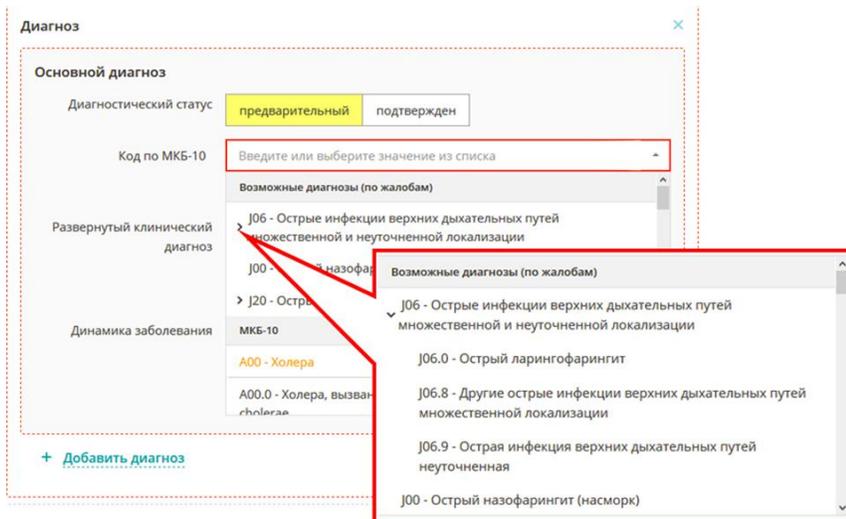


Рис. 2 – Примеры пользовательского интерфейса ПО.

8. Описание API

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

Базовый URL для всех запросов к облачному сервису - <https://top3.sbermed.ai/api>.

8.1. Получение предсказания

Для получения одиночного предсказания необходимо сделать *POST* запрос на адрес */calls*.

8.1.1. Входные данные

1. Обязательными заголовками при этом являются:
 - *Authorization*: аутентификационный ключ клиента, для его получения необходимо обратиться к администраторам;
 - *Content-Type*: принимается только *application/json*
2. Текст запроса (payload) должен быть в формате JSON и иметь следующую структуру:

```
{
  "user_id": "med-1-doctor-1"
  "symptoms": "насморк, головная боль, температура, кашель,
ломота в теле, покраснения глаз"
}
```

3. Значение текстового поля *user_id* - это идентификатор пользователя (врача), *symptoms* - симптомы и жалобы пациента в свободной форме.

8.1.2. Выходные данные

1. Результатом вызова будет являться JSON документ следующей структуры:

```
{
  "call_id": "19c46978-26ae-4e76-b500-de2dc97d42ce",
  "diag": [
    ["J06", "Острые инфекции верхних дыхательных путей
множественной и неуточненной локализации"],
    ["J00", "Острый назофарингит (насморк)"],
    ["J01", "Острый синусит"]
  ]
}
```

2. Значение поля *call_id* - это уникальный идентификатор запуска, который может в дальнейшем использоваться для сбора статистики и любой другой информации, которая может быть полезна для усовершенствования предсказательной модели и сервиса.
3. Значения массива *diag* - это предсказанные моделью наиболее вероятные диагнозы.
4. Возможны следующие коды результатов работы для запросов:
 - 201 - модель и сервис успешно отработали;
 - 401 - аутентификация с указанным ключом запрещена;
 - 500 - произошёл сбой в работе модели или сервиса.

8.1.3. Пример запроса и ответа

1. Пример запроса с использованием утилиты curl:

```
curl -d '{"user_id":"med-1-doctor-1", "symptoms":"насморк, головная боль, температура, покраснения глаз, кашель, ломота в теле"}' -H 'Authorization: test30bf87b2472b710492437d03e1b35fbd7d42333a3d9bb489d0a103062ce' -H 'Content-Type: application/json' https://top3.sbermed.ai/api/calls
```

2. Пример ответа на запрос

```
{"diag":[[{"J06","Острые инфекции верхних дыхательных путей множественной и неуточненной локализации"}, {"J00","Острый назофарингит (насморк)"}, {"J01","Острый синусит"}]], "call_id":"85cf2465-ee2c-438e-836b-ecb6f3fca327"}
```

8.2. Передача обратной связи

Для передачи обратной связи по предсказанию необходимо сделать *POST* запрос на адрес */results*.

8.2.1. Входные данные

1. Обязательными заголовками при этом являются:
 - *Authorization*: аутентификационный ключ клиента;
 - *Content-Type*: принимается только *application/json*
2. Текст запроса (payload) должен быть в формате JSON и иметь следующую структуру:

```
{
  "call_id": "19c46978-26ae-4e76-b500-de2dc97d42ce",
  "diag": [
    ["J00", "Острый назофарингит (насморк)"]
  ]
}
```

3. Значение поля *call_id* - это уникальный идентификатор запуска, по которому передается обратная связь.
4. Значения массива *diag* - это фактические диагнозы, которые были выбраны врачом в данном случае.

8.2.2. Выходные данные

1. Возможны следующие коды результатов работы для запросов:
 - 201 - сервис успешно отработал;
 - 401 - аутентификация с указанным ключом запрещена;
 - 500 - произошёл сбой в работе сервиса.

8.2.3. Пример запроса

1. Пример запроса с использованием утилиты *curl*:

```
curl -d '{"call_id":"85cf2465-ee2c-438e-836b-ecb6f3fca327", "diag":
```

```
[[{"J00", "Острый назофарингит (насморк)"}]]' -H 'Authorization:  
test30bf87b2472b710492437d03e1b35fbd7d42333a3d9bb489d0a103062ce' -H  
'Content-Type: application/json' https://top3.sbermed.ai/api/results
```

9. Техническая поддержка

Если при работе с API у вас возникнут ошибки, вопросы или пожелания, их можно направить на helpdesk@sbermed.ai.