

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «ПРОГРАММА ДЛЯ ПОМОЩИ В
ДИФФЕРЕНЦИАЛЬНОЙ ДИАГНОСТИКЕ GALENOS.AI»**

**ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ**

ОПИСАНИЕ ТЕХНИЧЕСКОЙ АРХИТЕКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

**ОПИСАНИЕ ПРОЦЕССОВ, ОБЕСПЕЧИВАЮЩИХ ПОДДЕРЖАНИЕ
ЖИЗНЕННОГО ЦИКЛА, В ТОМ ЧИСЛЕ УСТРАНЕНИЕ НЕИСПРАВНОСТЕЙ И
СОВЕРШЕНСТВОВАНИЕ, А ТАКЖЕ ИНФОРМАЦИЮ О ПЕРСОНАЛЕ**

**ИНСТРУКЦИЯ ПО РАЗВЁРТЫВАНИЮ ЭКЗЕМПЛЯРА ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ «ПРОГРАММА ДЛЯ ПОМОЩИ В ДИФФЕРЕНЦИАЛЬНОЙ
ДИАГНОСТИКЕ GALENOS.AI»**

ИНСТРУКЦИЯ ПО ИСПОЛЬЗОВАНИЮ СИСТЕМЫ

**Ценовая политика программного обеспечения «ПРОГРАММА ДЛЯ ПОМОЩИ В
ДИФФЕРЕНЦИАЛЬНОЙ ДИАГНОСТИКЕ GALENOS.AI»**

ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

“Программа для помощи в дифференциальной диагностике Galenos.AI”

Перечень сокращений, терминов и определений

Сокращение	Определение
ПО	Программное обеспечение
СУБД	Система управления базами данных
ИИ	Искусственный интеллект
ЗНО	Злокачественное новообразование
ЭМК	Электронная медицинская карта
МИС	Медицинская информационная система
МО	Медицинская организация
Заказчик	Владелец «права на передачу неисключительных (пользовательских) лицензионных прав на программное обеспечение»
СППВР	Система поддержки принятия врачебных решений
ЭВМ	Электронно-вычислительная машина

ГОСТ	Государственный стандарт
JSON	Текстовый формат обмена данными, основанный на языке программирования JavaScript
API	Программный интерфейс приложения
HTTP(s)	Протокол передачи данных между веб-страницей и ЭВМ

Описание программного обеспечения «Программа для помощи в дифференциальной диагностике Galenos.AI»

Программное обеспечение «Программа для помощи в дифференциальной диагностике Galenos.AI» (далее могут использоваться следующие наименования медицинского изделия: Программа, ПО Galenos.AI, Изделие) предназначено для поддержки проявления онкологической настороженности по данным ЭМК, анамнеза и жалоб пациента, которые предоставляются в виде протокола анкетирования, обрабатываются и Программа определяет необходимость направления пациента на диагностику ЗНО.

В основе ПО лежит алгоритм ИИ, базирующийся на методе градиентного бустинга. Предварительно обученная предсказательная модель, которая обрабатывает внесенные пользователем данные пациента, выполняет предсказание диагнозов в виде вероятностей, выраженных в процентах. Чем больше вероятность наличия того или иного ЗНО, тем более выражен с точки зрения Программы указанный риск. Изделие способно классифицировать на 17 классов (самые часто встречающиеся 16 видов ЗНО и вероятность отсутствия ЗНО). Значение риска, равное пороговому значению или более него, должно подразумевать риск, достаточный для проявления настороженности как в отношении данной группы патологий, так и в целом любых нозологических форм ЗНО, и для необходимости внесения ЗНО в список дифференциального диагноза. Величина порогового значения изменяется в настройках ПО и изначально установлена в значение 30%. Изменение значения данной величины производится Администратором системы по указанию администрации Заказчика.

ПО Galenos.AI функционирует как отдельная система, так и в составе программных комплексов при интеграции с МИС.

Основные требования к ПО Galenos.AI:

1. Производительность: ПО Galenos.AI оптимизирована для обеспечения высокой скорости выполнения операций, таких как поиск, запись и обновление данных.
2. Надежность: ПО Galenos.AI должна быть стабильной и надежной, чтобы обеспечить сохранность данных и минимальные простои.
3. Безопасность: ПО Galenos.AI должна обеспечивать механизмы безопасности для защиты данных от несанкционированного доступа, изменений и утечек. ПО Galenos.AI не обрабатывает и не хранит персональные данные пациентов.
4. Мобильность: ПО Galenos.AI обеспечивает удобство установки.
5. Сопровождаемость: интерфейс соответствует современным эргономическим требованиям и обеспечивать удобный доступ к функциям и операциям ПО Galenos.AI.

Основными функциональными возможностями ПО Galenos.AI являются:

1. Взаимодействие пользователя с ПО Galenos.AI, осуществляющееся через клиентский компонент:
 - ПО Galenos.AI имеет визуальный графический интерфейс, реализованный как одностраничное веб-приложение на платформе Angular, либо клиентская часть реализуется в МИС, используемой МО, и является внешней по отношению к сервису.
2. Поддержка реализации алгоритма ИИ:
 - ПО Galenos.AI позволяет получать входные данные, внесённые пользователем, обрабатывать запрос, обращаясь к предварительно обученной предсказательной модели; выходные данные для ПО Galenos.AI представляют собой интерпретированные результаты, полученные в ответ на запрос (рассчитанные

Программой с использованием алгоритмов ИИ), а также сообщения о возможных ошибках.

3. Поддержка работы со структурированными данными:

- ПО Galenos.AI обеспечивает хранение, организацию и управление структурированными данными, за исключением персональных данных пациентов.

4. Предоставление механизмов для обеспечения безопасности данных, таких как аутентификация, авторизация и шифрование:

- ПО Galenos.AI осуществляет различные механизмы безопасности данных. Встроенная аутентификация позволяет проверять и подтверждать личность пользователей, имеющих доступ к базе данных. Авторизация позволяет определить права доступа пользователей к различным коллекциям и операциям в базе данных. ПО Galenos.AI также поддерживает шифрование данных, включая шифрование в покое и во время передачи данных по сети, чтобы обеспечить конфиденциальность и защиту данных.

Архитектура программного обеспечения «Программа для помощи в дифференциальной диагностике Galenos.AI»

Изделие устроено по принципу клиент-серверной архитектуры: компоненты ПО должны быть отнесены либо к клиентской, либо к серверной части.

ПО Galenos.AI должно включать, как минимум, следующие основные сервисы (не считая СУБД, которая не является составной частью программы):

1. ПО Galenos.AI UI;
2. Сервис аутентификации;
3. Сервис НСИ;
4. Сервис данных анкет пациентов.

Серверные компоненты должны быть реализованы на основе микросервисного подхода с обеспечением взаимодействия между компонентами путем обмена сообщениями по сети. Сервисные компоненты должны предоставлять программный интерфейс (API) по протоколу HTTP(s) (формат данных – JSON) для обеспечения возможности интеграции с внешними системами и автоматизации решения административных задач.

2.1 Клиентская часть

Взаимодействие пользователя с ПО Galenos.AI должно осуществляться через клиентский (пользовательский) компонент – визуальный графический интерфейс, реализованный как одностраничное веб-приложение на платформе Angular, либо клиентская часть реализуется в МИС, используемой МО, и является внешней по отношению к сервису. Содержит реализацию пользовательского интерфейса всех компонентов.

Клиентская часть ПО Galenos.AI должна иметь интерфейс, отвечающий следующим требованиям: отображение в виде единой web-страницы, возможность динамического заполнения анкеты на русском языке, наличие экранных форм в виде модальных окон, наличие руководства пользователя. Веб-приложение должно использовать адаптивную верстку для работы на различных пользовательских устройствах с разными экранными разрешениями. Пользовательский интерфейс должен проектироваться с учетом отечественных и наиболее авторитетных международных стандартов унификации и эргономики. Для облегчения работы пользователей должно соблюдаться единство наименований и расположения элементов управления.

Пользовательская часть согласно ГОСТ Р ИСО/МЭК ТО 12182 может быть отнесена:

- по типу функции – к СППВР;
- по классу прикладной области – к научным;
- по классу режима эксплуатации – с обработкой данных в режиме реального времени;
- по классу представления данных – к сетевым;

- по классу критичности – к не влияющим на безопасность человеческой жизни;
- по классу пользователя – для специалистов;
- по классу стабильности – с дискретным внесением изменений;
- по классу готовности – к запатентованным (оригинальной разработке);
- по классу использования – для множества пользователей.

2.2 Сервис аутентификации

Доступен по HTTP/JSON. Предназначен для управления пользователями / группами пользователей, аутентификацией, правами доступа. Развертывание осуществляется с использованием контейнерной виртуализации (технологии Docker), поставляется Заказчику в виде Docker-образа и развертывается в системе виртуализации.

2.3 Сервис нормативно-справочной информации

Доступен по HTTP/JSON. Предназначен для управления справочными и нормативными данными. Развертывание осуществляется с использованием контейнерной виртуализации (технологии Docker), поставляется Заказчику в виде Docker-образа и развертывается в системе виртуализации.

2.4 Сервис данных анкет пациентов

Доступен по HTTP/JSON. Серверные компоненты должны быть реализованы на основе микросервисного подхода с обеспечением взаимодействия между компонентами путем обмена сообщениями по сети. Развертывание осуществляется с использованием контейнерной виртуализации (технологии Docker), поставляется Заказчику в виде Docker-образа и развертывается в системе виртуализации.

Основными рабочими характеристиками ПО Galenos.AI являются скорость обработки запросов и количество сохраненных анкет. Программа должна удовлетворять следующим

показателям – время обработки запроса на получение данных пациента не должно превышать 2 секунд, количество сохраненных анкет должно быть неограниченно технологически, технически – объемом дискового пространства и возможностями СУБД.

Пример входных для алгоритма ИИ данных (результат анкетирования пациента):

```
{  
  "part1": {  
    "gender": "female",  
    "age": "18-25",  
    "part1Symptoms": [  
      "noSymptoms"  
    ]  
  },  
  "part2": {  
    "part2Symptoms": [  
      "noSymptoms"  
    ]  
  },  
  "part3": {  
    "part3Symptoms": [  
      "noSymptoms"
```



```
]
},
"part4": {
  "part4Symptoms": [
    "noSymptoms"
  ]
},
"part5": {
  "part5Symptoms": [
    "noSymptoms"
  ]
},
"part6": {
  "part6Symptoms": [
    "noSymptoms"
  ]
},
"part7": {
  "part7Symptoms": [
    "noSymptoms"
  ]
}
```

}

2.5 База данных

База данных ПО Galenos.AI должна находиться под управлением документоориентированной СУБД. Отдельных требований к размерности базы данных не предъявляется.

База данных ПО Galenos.AI выполняет следующие основные функции: загрузка и хранение данных, обработка поисковых запросов.



Перечень сокращений, терминов и определений

Сокращение	Определение
БД	База данных
ПО	Программное обеспечение
СУБД	Система управления базами данных
UI	Пользовательский интерфейс

ИИ	Искусственный интеллект
ЗНО	Злокачественное новообразование
ЭМК	Электронная медицинская карта
МИС	Медицинская информационная система
МО	Медицинская организация
Заказчик	Владелец «права на передачу неисключительных (пользовательских) лицензионных прав на программное обеспечение»
СППВР	Система поддержки принятия врачебных решений
ЭВМ	Электронно-вычислительная машина
ГОСТ	Государственный стандарт
JSON	Текстовый формат обмена данными, основанный на языке программирования JavaScript
API	Программный интерфейс приложения
HTTP(s)	Протокол передачи данных между веб-страницей и ЭВМ

1. Описание программного обеспечения «Программа для помощи в дифференциальной диагностике Galenos.AI»

Программное обеспечение «Программа для помощи в дифференциальной диагностике Galenos.AI» (далее могут использоваться следующие наименования медицинского изделия: Программа, ПО Galenos.AI, Изделие) предназначено для поддержки проявления онкологической настороженности по данным ЭМК, анамнеза и жалоб пациента, которые предоставляются в виде протокола анкетирования, обрабатываются и Программа определяет необходимость направления пациента на диагностику ЗНО.

В основе ПО лежит алгоритм ИИ, базирующийся на методе градиентного бустинга. Предварительно обученная предсказательная модель, которая обрабатывает внесенные пользователем данные пациента, выполняет предсказание диагнозов в виде вероятностей, выраженных в процентах. Чем больше вероятность наличия того или иного ЗНО, тем более выражен с точки зрения Программы указанный риск. Изделие способно классифицировать на 17 классов (самые часто встречающиеся 16 видов ЗНО и вероятность отсутствия ЗНО). Значение риска, равное пороговому значению или более него, должно подразумевать риск, достаточный для проявления настороженности как в отношении данной группы патологий, так и в целом любых нозологических форм ЗНО, и для необходимости внесения ЗНО в список дифференциального диагноза. Величина порогового значения изменяется в настройках ПО и изначально установлена в значение 30%. Изменение значения данной величины производится Администратором системы по указанию администрации Заказчика.

ПО Galenos.AI функционирует как отдельная система, так и в составе программных комплексов при интеграции с МИС.

Основные требования к ПО Galenos.AI:

1. Производительность: ПО Galenos.AI оптимизирована для обеспечения высокой скорости выполнения операций, таких как поиск, запись и обновление данных.
2. Надежность: ПО Galenos.AI должна быть стабильной и надежной, чтобы обеспечить сохранность данных и минимальные простои.

3. Безопасность: ПО Galenos.AI должна обеспечивать механизмы безопасности для защиты данных от несанкционированного доступа, изменений и утечек. ПО Galenos.AI не обрабатывает и не хранит персональные данные пациентов.

4. Мобильность: ПО Galenos.AI обеспечивает удобство установки.

5. Сопровождаемость: интерфейс соответствует современным эргономическим требованиям и обеспечивать удобный доступ к функциям и операциям ПО Galenos.AI.

Основными функциональными возможностями ПО Galenos.AI являются:

1. Взаимодействие пользователя с ПО Galenos.AI, осуществляющееся через клиентский компонент:
 - ПО Galenos.AI имеет визуальный графический интерфейс, реализованный как одностраничное веб-приложение на платформе Angular, либо клиентская часть реализуется в МИС, используемой МО, и является внешней по отношению к сервису.
2. Поддержка реализации алгоритма ИИ:
 - ПО Galenos.AI позволяет получать входные данные, внесённые пользователем, обрабатывать запрос, обращаясь к предварительно обученной предсказательной модели; выходные данные для ПО Galenos.AI представляют собой интерпретированные результаты, полученные в ответ на запрос (рассчитанные Программой с использованием алгоритмов ИИ), а также сообщения о возможных ошибках.
3. Поддержка работы со структурированными данными:
 - ПО Galenos.AI обеспечивает хранение, организацию и управление структурированными данными, за исключением персональных данных пациентов.
4. Предоставление механизмов для обеспечения безопасности данных, таких как аутентификация, авторизация и шифрование:

– ПО Galenos.AI осуществляет различные механизмы безопасности данных. Встроенная аутентификация позволяет проверять и подтверждать личность пользователей, имеющих доступ к базе данных. Авторизация позволяет определить права доступа пользователей к различным коллекциям и операциям в базе данных. ПО Galenos.AI также поддерживает шифрование данных, включая шифрование в покое и во время передачи данных по сети, чтобы обеспечить конфиденциальность и защиту данных.

1.1 Архитектурные принципы

Ключевые особенности архитектуры программного обеспечения ПО Galenos.AI:

1. Изделие устроено по принципу клиент-серверной архитектуры: компоненты ПО должны быть отнесены либо к клиентской, либо к серверной части. Серверные компоненты должны быть реализованы на основе микросервисного подхода с обеспечением взаимодействия между компонентами путем обмена сообщениями по сети. Сервисные компоненты должны предоставлять программный интерфейс (API) по протоколу HTTP(s) (формат данных – JSON) для обеспечения возможности интеграции с внешними системами и автоматизации решения административных задач.
Клиентская часть - ПО Galenos.AI UI. Построен на Angular. Содержит реализацию пользовательского интерфейса всех компонентов.
2. Документоориентированная модель данных: база данных ПО Galenos.AI должна находиться под управлением документоориентированной СУБД, где информация хранится в виде JSON-подобных документов. Такая модель позволяет гибко организовывать данные и упрощает их обработку. Благодаря этому ПО Galenos.AI подходит для работы с различными типами данных, включая сложные иерархические структуры.

1.2 Компонентная структура программного обеспечения «Программа для помощи в дифференциальной диагностике Galenos.AI»

ПО Galenos.AI работает на двух уровнях:

- Прикладной уровень;
- Уровень данных.

На прикладном уровне ПО Galenos.AI включает клиентские библиотеки.

Клиентские библиотеки предоставляют различные языковые интерфейсы для взаимодействия с сервером ПО Galenos.AI. Они позволяют приложениям выполнять операции чтения и записи данных.

На уровне данных ПО Galenos.AI включает сервер базы данных, механизм хранения данных и конфигурационные файлы.

Сервер базы данных отвечает за обработку запросов и управление данными. Он хранит данные и обрабатывает операции чтения и записи.

Механизм хранения данных отвечает за хранение данных на диске, используя схему хранения данных, основанную на JSON.

Конфигурационные файлы позволяют настроить параметры работы сервера ПО Galenos.AI, такие как порт прослушивания, размер кэша и другие. Они определяют поведение сервера и его взаимодействие с клиентскими приложениями.

Взаимодействие между компонентами происходит следующим образом: клиентские приложения используют клиентские библиотеки для отправки запросов и получения ответов от сервера ПО Galenos.AI. Сервер базы данных обрабатывает запросы от клиентских приложений, выполняет операции чтения и записи данных, а также хранит данные. Конфигурационные файлы определяют параметры работы сервера ПО Galenos.AI и его взаимодействие с клиентскими приложениями.

2. Архитектурные модели

Архитектурные модели в ПО определяют основные принципы организации и взаимодействия компонентов системы. Основная архитектурная модель, которая применяется в программном обеспечении ПО Galenos.AI - клиент-серверная архитектура, то есть компоненты ПО отнесены либо к клиентской, либо к серверной части.

2.1 Клиент-серверная архитектура

ПО Galenos.AI должно включать, как минимум, следующие основные сервисы (не считая СУБД, которая не является составной частью программы):

1. ПО Galenos.AI UI (клиентская часть - пользовательский интерфейс);
2. Сервис аутентификации;
3. Сервис НСИ;
4. Сервис данных анкет пациентов.

Серверные компоненты должны быть реализованы на основе микросервисного подхода с обеспечением взаимодействия между компонентами путем обмена сообщениями по сети. Сервисные компоненты должны предоставлять программный интерфейс (API) по протоколу HTTP(s) (формат данных – JSON) для обеспечения возможности интеграции с внешними системами и автоматизации решения административных задач.

Их развертывание осуществляется с использованием контейнерной виртуализации (технологии Docker), поэтому Сервисные компоненты поставляются Заказчику в виде Docker-образа и развертываются в системе виртуализации.

База данных ПО Galenos.AI должна находиться под управлением документоориентированной СУБД. Отдельных требований к размерности базы данных не предъявляется.

База данных ПО Galenos.AI выполняет загрузку и хранение данных, обработку поисковых запросов.

Взаимодействие пользователя с ПО Galenos.AI должно осуществляться через клиентский (пользовательский) компонент – визуальный графический интерфейс, реализованный как одностраничное веб-приложение на платформе Angular, либо клиентская часть реализуется в МИС, используемой МО, и является внешней по отношению к сервису. Он содержит реализацию пользовательского интерфейса всех компонентов.

Описание процессов, обеспечивающих поддержание жизненного цикла Программного обеспечения «Программа для помощи в дифференциальной диагностике Galenos.AI», в том числе устранение неисправностей и совершенствование, а также информацию о персонале, необходимом для обеспечения такой поддержки

Перечень сокращений, терминов и определений

Сокращение	Определение
ЖЦ	Жизненный цикл
Заказчик	Владелец «права на передачу неисключительных (пользовательских) лицензионных прав на программное обеспечение»
БД	База данных
СУБД	Система управления базами данных
ПО	Программное обеспечение
МО	Медицинская организация
ОС	Операционная система

1. Жизненный цикл программного обеспечения «Программа для помощи в дифференциальной диагностике Galenos.AI»

Настоящий документ является описанием процессов жизненного цикла программного обеспечения «Программа для помощи в дифференциальной диагностике Galenos.AI» и содержит сведения о жизненном цикле программного обеспечения, в том числе устранение неисправностей, выявленных в ходе эксплуатации программного обеспечения, а также информацию о персонале, необходимом для обеспечения такой поддержки.

1.1. Описание жизненного цикла

Жизненный цикл (ЖЦ) включает период создания и использования программного обеспечения «Программа для помощи в дифференциальной диагностике Galenos.AI», начиная с момента возникновения потребности в продукте, заканчивая разработкой, тестированием и отладкой, поставкой программной продукции, ее эксплуатацией на объектах Заказчика (МО) и технической поддержкой.

ЖЦ определен с учетом положений следующих стандартов:

- ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»;

- ГОСТ Р 56939-2016 «Защита информации. Разработка безопасного программного обеспечения. Общие требования».

Основными процессами ЖЦ программной продукции являются:

- Исследование требований к продукции;
- Формирование и оценка задач;
- Проектирование и разработка;
- Тестирование и отладка;
- Эксплуатация и сопровождение.

Разработка ПО ведется по итеративной модели с выпуском новой версии в среднем каждые две недели. Каждая итерация включает в себя этапы формирования и оценки задач,

разработки и тестирования по задачам, обновления документации. Процессы управления конфигурацией ПО осуществляются с использованием репозитория дистрибутивов, стенда сборки и системы контроля версий.

2. Процессы жизненного цикла

2.1 Исследование требований к продукции

2.1.1 Определение требований, относящихся к продукции

Требования к продукции определяются путем маркетингового исследования и анализа требований потенциального потребителя и рынка ПО, а также учитываются нормативные и законодательные требования, применимые к данному виду продукции. В компании формируется рыночная стратегия, из которой определяется приоритетная стратегия в

составе стратегических инициатив на разработку новых ПО, планов производства и налаживание новых каналов сбыта, продвижения продукции на рынок.

2.1.2 Анализ требований, относящихся к продукции

Под анализом требований потребителя подразумевается выяснение способности ООО «ТехЛаб» реализовать требования потребителя в виде изучения данных, полученных в ходе маркетинговой деятельности. Руководством определены продуктивные механизмы и инструменты взаимодействия с потребителями на этапах маркетингового исследования.

Разработанный механизм установления требований потребителей позволяет адекватно определить и документально зафиксировать требования потребителя для предотвращения возникновения разногласий между потребителем и исполнителем.

2.1.3 Изменение требований к продукции

В случае изменений требований к продукции, ведению технологического процесса и др. соответствующая документированная информация актуализируется в соответствии с требованиями внутренних процедур системы управления качеством, до персонала изменения доводятся посредством обучения.

2.2 Формирование и оценка задач

Процессом формирования и оценки задач определен порядок:

- ответственный за продукт аналитик определяет задачи, которые должны быть выполнены в итерацию;
- ответственный за продукт аналитик определяет требования к задачам;
- технический менеджер, ответственный за продукт аналитик совместно с командой разработки проводят оценку задач;
- технический менеджер устанавливает ответственность за разработку, распределяя задачи между членами команды разработки;
- технический менеджер осуществляет мониторинг этого процесса (установлены критерии и методы оценки результативности процесса).

Управление задачами ведётся с использованием выпусков (далее, по тексту - релизов), т.е. все задачи, запланированные в релиз, помечаются соответствующим образом. Задачи, не включенные в какой-либо релиз, остаются в списке задач по продукту (далее, по тексту - бэклог) и не участвуют в разработке, а используются только для планирования будущих работ.

Результатами этапа являются:

1. Сформирован бэклог на итерацию и релиз;
2. Подготовлена спецификация требований и/или техническое описание по каждой задаче;
3. Назначен исполнитель по каждой задаче.

2.3 Проектирование и разработка

Процессом проектирования и разработки задач определен порядок:

- разработка программной архитектуры, а также разработка решений по построению всех составных компонент;
- разработка исходных текстов, написание файлов спецификации для сборки пакетов прикладного ПО;
- сборка пакетов прикладного ПО и добавление их в репозиторий программного обеспечения;
- сборка дистрибутивов из репозитория ПО.

Разработка по задачам идёт до тех пор, пока все задачи в рамках итерации не будут закрыты.

2.4 Тестирование и отладка

Проведение тестирования является обязательным перед передачей новой версии потребителю. Тестирование проводится лицами, ответственными за проведение тестирования программной продукции (отдел тестирования). Для тестирования и отладки программной продукции собирается стенд, выдается задание на тестирование. По результатам тестирования осуществляется устранение ошибок и осуществляется (при необходимости) доработка ПО.

Процессом тестирования и отладки определен порядок:

- формируется тест-план;
- распределяются задачи по тестированию между сотрудниками отдела тестирования;
- подготавливаются/изменяются тестовые сценарии.

Передача планового релиза в тестирование производится согласно графику выхода версий.

По окончанию тестирования формируется файл отчетности о проведенной работ.

2.5 Эксплуатация и сопровождение

2.5.1 Техническая поддержка

Техническая поддержка пользователей осуществляется путем:

- регистрации обращений пользователей по различным вопросам эксплуатации программного обеспечения,

- координации усилий как своих специалистов, так и сотрудников прочих структурных подразделений, при необходимости – и внешних соисполнителей,
- подготовку ответов на эти обращения.

Указанные действия осуществляются по электронным каналам связи (электронной почте info@teh-lab.ru). В рамках технической поддержки сервиса оказываются следующие услуги:

- помощь в установке ПО;
- помощь в настройке и администрировании ПО;
- помощь в установке обновлений ПО;
- помощь в поиске и устранении проблем в случае некорректной установки обновления ПО;
- описание функционала ПО, консультирование;
- фиксация обнаруженных пользователями проблем, сбойных ситуаций, выявленных несоответствий документации;
- предоставление актуальной документации по установке/настройке/работе ПО.

В заявке на техническую поддержку пользователь должен указать следующую информацию:

- описание проблемы;
- предпринятые попытки решения проблемы;
- любая релевантная дополнительная информация.

После доставки ответа пользователю запрос считается завершенным, и находится в статусе “завершен, требует подтверждения пользователя”. В случае аргументированного несогласия пользователя с завершением запроса, выполнение запроса продолжается.

Завершенный запрос переходит в состояние закрытого после получения подтверждения от пользователя о решении запроса. В случае отсутствия ответа пользователя о завершении

запроса в течение 14 рабочих дней, в случае если иное не оговорено в соглашении о расширенной технической поддержки, запрос считается закрытым. Закрытие запроса может инициировать пользователь, если надобность в ответе на запрос по каким-либо причинам более не требуется.

2.5.2 Персонал, обеспечивающий работу на местах

Для ПО “Программа для помощи в дифференциальной диагностике Galenos.AI” определены следующие категории пользователей:

1. Пользователь – медицинские работники (врачи общей практики, терапевты, другие врачи амбулаторно-поликлинического звена, фельдшеры, средний медицинский персонал), предварительно ознакомившиеся с руководством пользователя.

Пользователи должны иметь опыт работы с персональным компьютером и веб-браузером.

2. Системный администратор – как правило, сотрудник организации Заказчика, обеспечивающий бесперебойное функционирование программы и проведение регламентных мероприятий по ее обслуживанию. Может выполнять функции роли администратора, что подразумевает расширенные права доступа к базовым инструментам управления информационными объектами: данными о пользователях, группах пользователей, правах доступа и т.п.

Системный администратор должен иметь опыт администрирования следующих системных и базовых программных средств:

1. Linux-совместимая ОС с docker-совместимыми средствами контейнеризации
2. Документо-ориентированная СУБД;

Кроме того, системный администратор должен быть знаком со службами безопасности ОС Linux в объеме, позволяющем устанавливать сертификаты цифровой подписи и ключей шифрования в соответствующие хранилища сертификатов.

2.5.3 Персонал, обеспечивающий техническую поддержку и развитие системы

Специалисты, обеспечивающие техническую поддержку и развитие ПО “Программа для помощи в дифференциальной диагностике Galenos.AI”, должны

обладать следующими знаниями и навыками:

- Владение персональным компьютером на уровне уверенного пользователя;

- Знание функциональных возможностей ПО «Программа для помощи в дифференциальной диагностике Galenos.AI» и особенностей работы с ними;
- Знание языков программирования;
- Знание реляционных БД;
- Значение протоколов обмена данными;
- Знание средств восстановления баз данных и мониторинга производительности серверов.

Коллектив разработчиков (программисты, консультанты, технические специалисты)

обладают необходимым набором знаний для работы со всеми компонентами, входящими в состав ПО, при решении прикладных задач, соответствующих функционалу программ.

3. Фактический адрес размещения разработчиков ПО «Программа для помощи в дифференциальной диагностике Galenos.AI»

191186, Санкт-Петербург, улица Итальянская, дом 17, литер А, офис 258. Тел.: 8 800 302 5510

4. Фактический адрес размещения производителей ПО «Программа для помощи в дифференциальной диагностике Galenos.AI»

191186, Санкт-Петербург, улица Итальянская, дом 17, литер А, офис 258. Тел.: 8 800 302 5510

5. Фактический адрес размещения службы поддержки ПО «Программа для помощи в дифференциальной диагностике Galenos.AI»

191186, Санкт-Петербург, улица Итальянская, дом 17, литер А, офис 258. Тел.: 8 800 302 5510

ИНСТРУКЦИЯ ПО ИСПОЛЬЗОВАНИЮ СИСТЕМЫ
«Программа для помощи в дифференциальной диагностике Galenos.AI»

1. Описание программного обеспечения «Программа для помощи в дифференциальной диагностике Galenos.AI»

Программное обеспечение «Программа для помощи в дифференциальной диагностике Galenos.AI» (далее могут использоваться следующие наименования медицинского изделия: Программа, ПО Galenos.AI, Изделие) предназначено для поддержки проявления онкологической настороженности по данным ЭМК, анамнеза и жалоб пациента, которые предоставляются в виде протокола анкетирования, обрабатываются и Программа определяет необходимость направления пациента на диагностику ЗНО.

В основе ПО лежит алгоритм ИИ, базирующийся на методе градиентного бустинга. Предварительно обученная предсказательная модель, которая обрабатывает внесённые пользователем данные пациента, выполняет предсказание диагнозов в виде вероятностей, выраженных в процентах. Чем больше вероятность наличия того или иного ЗНО, тем более выражен с точки зрения Программы указанный риск. Изделие способно классифицировать на 17 классов (самые часто встречающиеся 16 видов ЗНО и вероятность отсутствия ЗНО). Значение риска равное пороговому значению или более него должно подразумевать риск, достаточный для проявления настороженности как в отношении данной группы патологий, так и в целом любых нозологических форм ЗНО, и для необходимости внесения ЗНО в список дифференциального диагноза. Величина порогового значения изменяется в настройках ПО и изначально установлена в значение 30%. Изменение значения данной величины производится Администратором системы по указанию администрации Заказчика.

ПО Galenos.AI функционирует как отдельная система, так и в составе программных комплексов при интеграции с МИС.

Основные требования к ПО Galenos.AI:

1. Производительность: ПО Galenos.AI оптимизирована для обеспечения высокой скорости выполнения операций, таких как поиск, запись и обновление данных.
2. Надежность: ПО Galenos.AI должна быть стабильной и надежной, чтобы обеспечить сохранность данных и минимальные простои.
3. Безопасность: ПО Galenos.AI должна обеспечивать механизмы безопасности для защиты данных от несанкционированного доступа, изменений и утечек. ПО Galenos.AI не обрабатывает и не хранит персональные данные пациентов.

4. Мобильность: ПО Galenos.AI обеспечивает удобство установки.
5. Сопровождаемость: интерфейс соответствует современным эргономическим требованиям и обеспечивать удобный доступ к функциям и операциям ПО Galenos.AI.

Основными функциональными возможностями ПО Galenos.AI являются:

1. Взаимодействие пользователя с ПО Galenos.AI, осуществляющееся через клиентский компонент:
 - ПО Galenos.AI имеет визуальный графический интерфейс, реализованный как одностраничное веб-приложение на платформе Angular, либо клиентская часть реализуется в МИС, используемой МО, и является внешней по отношению к сервису.
2. Поддержка реализации алгоритма ИИ:
 - ПО Galenos.AI позволяет получать входные данные, внесённые пользователем, обрабатывать запрос, обращаясь к предварительно обученной предсказательной модели; выходные данные для ПО Galenos.AI представляют собой интерпретированные результаты, полученные в ответ на запрос (рассчитанные Программой с использованием алгоритмов ИИ). К выходным данным ПО Galenos.AI относятся первоочередная и пояснительная информация, а также сообщения об ошибках.
3. Поддержка работы с структурированными данными:
 - ПО Galenos.AI обеспечивает хранение, организацию и управление структурированными данными, за исключением персональных данных пациентов.
4. Предоставление механизмов для обеспечения безопасности данных, таких как аутентификация, авторизация и шифрование:
 - ПО Galenos.AI осуществляет различные механизмы безопасности данных. Встроенная аутентификация позволяет проверять и подтверждать личность

пользователей, имеющих доступ к базе данных. Авторизация позволяет определить права доступа пользователей к различным коллекциям и операциям в базе данных. ПО Galenos.AI также поддерживает шифрование данных, включая шифрование в покое и во время передачи данных по сети, чтобы обеспечить конфиденциальность и защиту данных.

2. Авторизация в системе

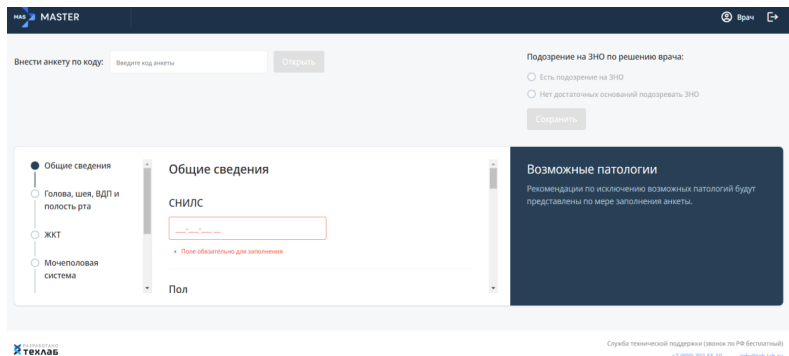
Для авторизации в системе необходимо ввести в поля «Логин» и «Пароль» данные для входа, затем нажать кнопку «Войти». Данные для входа предоставляет администратор системы. Обратите внимание, пароль чувствителен к регистру букв, т.е. password и PASSWORD — это разные пароли.

3. Описание интерфейса системы

В зависимости от роли (врач-терапевт, врач-онколог) пользователю доступен разный интерфейс системы:

- **врачу-терапевту** доступны несколько таблиц для ввода и вывода информации:

- поиск анкеты
- общие сведения о пациенте
- возможные патологии
- подозрение на ЗНО по решению врача

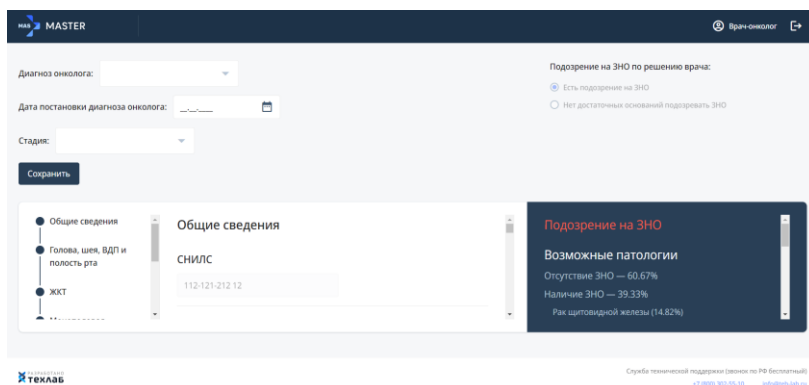


- **врачу-онкологу** доступны следующие таблицы для ввода информации:

- **диагноз онколога**

а также только для вывода:

- **общие сведения о пациенте**
- **подозрение на ЗНО по решению врача**
- **подозрение на ЗНО**



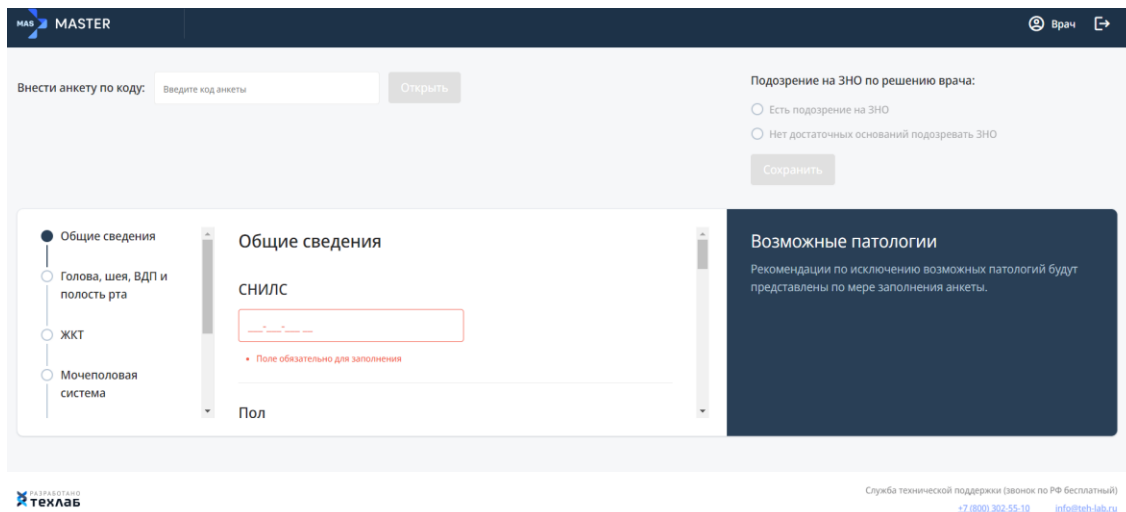
4. Работа под ролью «врач-терапевт»

Для начала работы пользователю необходимо авторизоваться под ролью «врач-терапевт».

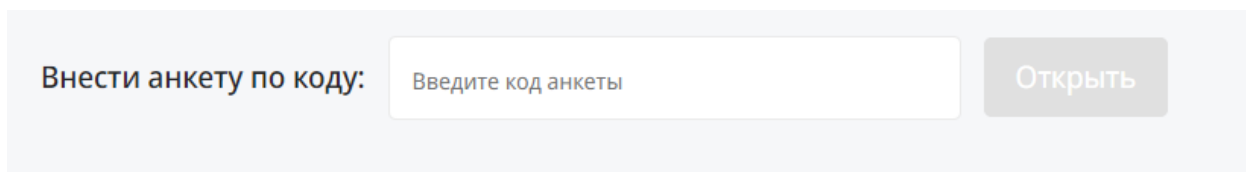
Перед пользователем откроется окно со следующими таблицами:

- **поиск анкеты**
- **общие сведения о пациенте**
- **возможные патологии**

- подозрение на ЗНО по решению врача



Если у пользователя есть код анкеты пациента, он может ввести его в соответствующую строку и нажать кнопку «открыть», как это указано ниже



Для работы с новым пациентом необходимо перейти в таблицу «Общие сведения» и заполнить данные:

- СНИЛС
- Пол
- Возраст
- Общие симптомы

● Общие сведения

● Голова, шея, ВДП и полость рта

● ЖКТ

● Мочеполовая система

Общие сведения

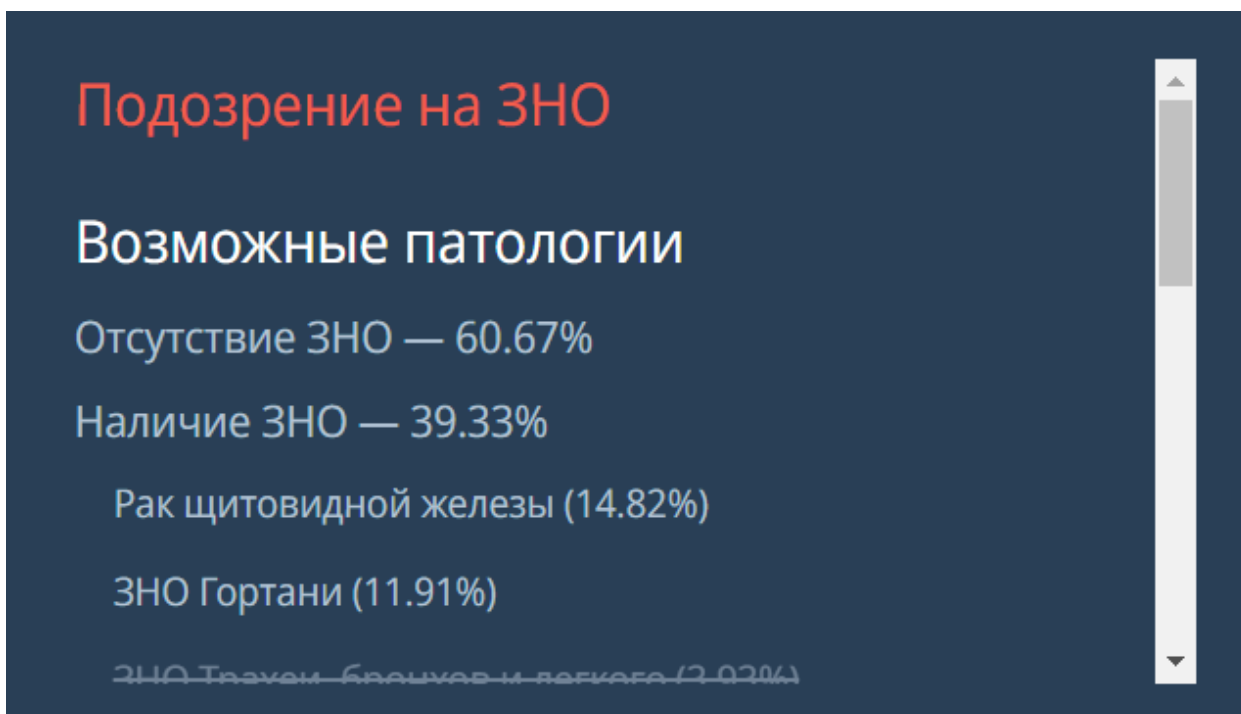
СНИЛС

112-121-212 12

Пол

☒ Мужской

На основании заполненных данных система предположит варианты подозрений на ЗНО.



Важно! Система только предполагает вероятности подозрения на ЗНО в соответствии с заполненными врачом-терапевтом данными. Окончательное решение о наличии или отсутствии подозрения на заболевание принимает врач-терапевт:

- если врач-терапевт считает, что подозрение на ЗНО имеется, он нажимает флажок в соответствующей строке и сохраняет данные анкеты

Подозрение на ЗНО по решению врача:

☒ Есть подозрение на ЗНО

☐ Нет достаточных оснований подозревать ЗНО

Сохранить

- если врач-терапевт считает, что нет достаточных оснований подозревать ЗНО, он нажимает флажок в соответствующей строке и сохраняет данные анкеты

Подозрение на ЗНО по решению врача:

☐ Есть подозрение на ЗНО

☒ Нет достаточных оснований подозревать ЗНО

Сохранить

5. Работа под ролью «врач-онколог»

Для начала работы пользователю необходимо авторизоваться под ролью «врач-онколог».

Перед пользователем откроется окно с поиском анкеты пациента по номеру СНИЛС.

Для дальнейшей работы необходимо ввести номер СНИЛС и нажать кнопку «Открыть»

MAS MASTER

Врач-онколог

Открыть анкету по СНИЛС:

техлаб

Служба технической поддержки (звонок по РФ бесплатный)
[+7 \(800\) 362-55-10](tel:+78003625510) info@teh-lab.ru

Перед пользователем откроется окно со следующими таблицами:

- диагноз онколога
- общие сведения о пациенте
- подозрение на ЗНО по решению врача
- подозрение на ЗНО

Диагноз онколога:

Дата постановки диагноза онколога:

Стадия:

Подозрение на ЗНО по решению врача:
☐ Есть подозрение на ЗНО
☒ Нет достаточных оснований подозревать ЗНО

Общие сведения

Голова, шея, ВДП и полость рта

ЖКТ

☐ Увеличение прочих лимфатических узлов

☐ Анемия

☐ Кровохарканье

☐ Боль за грудиной

☐ Боль в грудной клетке

☐ Бледность кожи с землистым оттенком

Подозрение на ЗНО

Возможные патологии

Отсутствие ЗНО — 60.67%

Наличие ЗНО — 39.33%

Рак щитовидной железы (14.82%)

Таблица «общие сведения» (рис.ниже) носит информационный характер, данные в нее вносит «врач-терапевт». Врач-онколог не может вносить данные в эту таблицу, но может изучить все необходимые сведения о пациенте.

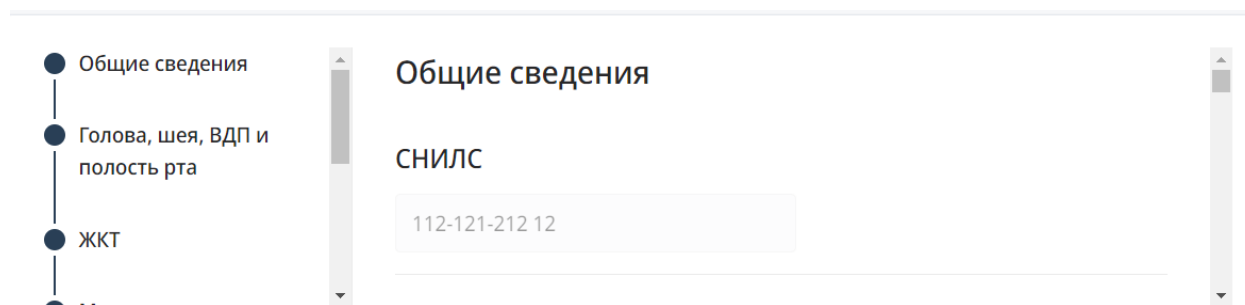


Таблица «Подозрение на ЗНО по решению врача» (рис.ниже) носит информационный характер, данные в этой таблице были заполнены врачом-терапевтом, на основании симптомов, отмеченных у пациента, и рекомендации системы.

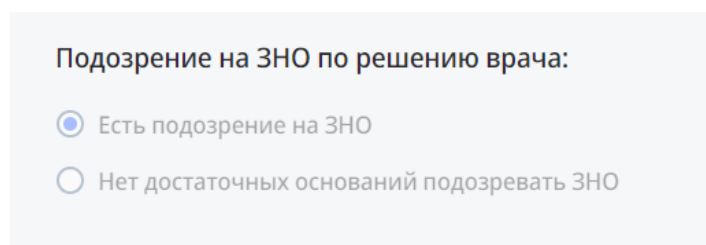


Таблица «Подозрение на ЗНО» (рис.ниже) носит информационный характер, данные в этой таблице сгенерированы системой на основании введенных врачом-терапевтом симптомов пациента. Информация в данной таблице – справочная, решение о постановке диагноза принимает врач-онколог.

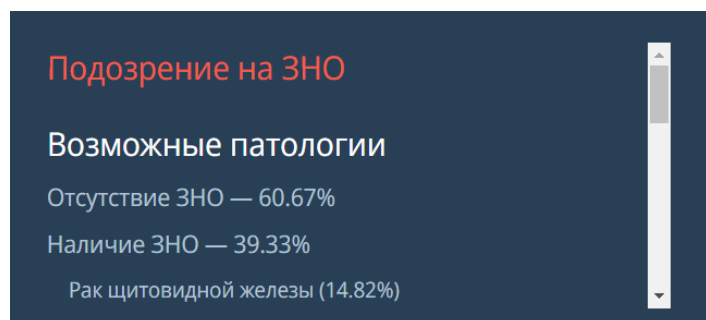



Таблица «Диагноз онколога» (рис.ниже) заполняется врачом-онкологом после изучения информационных таблиц, описанных выше. Врач-онколог может принять во внимание рекомендации системы о подозрении на ЗНО, но диагноз ставит самостоятельно.

Диагноз онколога:

Дата постановки диагноза онколога:

__-__-__




Стадия:

Для постановки диагноза врачу-онкологу необходимо заполнить следующие данные в этой таблице:


- диагноз онколога,
- дата постановки диагноза,
- стадия

После заполнения этих данных (рис. ниже), необходимо нажать на кнопку «сохранить»


Диагноз онколога:

БОЛЕЗНИ ЩИТОВИДНОЙ ... 

Дата постановки диагноза онколога:

26.01.2024 

Стадия:

Ia 

Сохранить

**ИНСТРУКЦИЯ ПО РАЗВЁРТЫВАНИЮ ЭКЗЕМПЛЯРА ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ «ПРОГРАММА ДЛЯ ПОМОЩИ В ДИФФЕРЕНЦИАЛЬНОЙ
ДИАГНОСТИКЕ GALENOS.AI»**

1. Перечень сокращений, терминов и определений

Сокращение	Определение
ПО	Программное обеспечение
Дистрибутив	Набор файлов, приспособленный для распространения ПО

Требования к стенду

Для корректной работы программы необходима Linux-совместимая операционная система с поддержкой среды контейнеризации Docker. Для указанных ниже шагов использовалась Ubuntu 22.04.

Перед началом установки ПО убедитесь, что ваша система соответствует следующим требованиям:

- установка должна проводиться от имени учетной записи root или записи с правами sudo. Для шагов ниже использовалась учетная запись root;
- дистрибутив находится в домашней директории на сервере, где осуществляется установка, она же является текущей директорией.

2. Подготовительные мероприятия к установке ПО

Установка обновлений системы:

```
apt update
```

Установка Docker:

```
apt install -y docker.io unzip
```

Создание сети Docker (для инструкции используем 192.168.90.0/24, можно использовать любую другую, внося соответствующие правки в команды ниже):

```
docker network create --subnet=192.168.90.0/24 oncoalertness
```

Создание директории с настроечными и рабочими файлами компонентов:

```
mkdir -p /data
```

Распаковка дистрибутива:

```
unzip oncoalertness-2024.1.zip
```

```
cd oncoalertness-2024.1
```


3. Установка общего программного обеспечения

3.1. Nginx

Можно использовать сторонний nginx, для инструкции используем входящий в дистрибутив:

```
mkdir -p /data/0_proxy
cp -rv ./oncoalrtness_0_infra_0_nginx/config /data/0_proxy
cp -rv ./oncoalrtness_0_infra_0_nginx/data /data/0_proxy
docker load -i ./oncoalrtness_0_infra_0_nginx/oncoalrtness_0_infra_0_nginx.tar.gz
docker run --restart=always --name 0_proxy -v
/data/0_proxy/config/nginx.conf:/etc/nginx/nginx.conf -v
/data/0_proxy/config/ssl:/etc/nginx/ssl -v /data/0_proxy/config/instances:/etc/nginx/instances -v
/data/0_proxy/data:/usr/share/nginx/html --network=oncoalrtness --ip=192.168.90.200 -p 80:80
--restart=always --log-driver json-file --log-opt max-size=10m -d nginx:1.21.0-alpine
```

3.2. Конфигурация Nginx

Если используем сторонний nginx, то надо адаптировать конфигурацию из дистрибутива (oncoalrtness_0_infra_0_proxy):

```
cp -rv ./oncoalrtness_0_infra_0_proxy/config /data/0_proxy
```

Отредактировать /data/0_proxy/config/instances/oncoalrtness-demo.conf, заменить oncoalrtness-demo.teh-lab.ru на IP адрес сервера или домен:

```
docker exec 0_proxy nginx -s reload
```

3.3. MongoDB

Можно использовать сторонний MongoDB, для инструкции используем входящий в дистрибутив:

```
mkdir -p /data/1_mongo
cp -rv ./oncoalrtness_0_infra_1_mongo/data /data/1_mongo
docker load -i oncoalrtness_0_infra_1_mongo/oncoalrtness_0_infra_1_mongo.tar.gz
docker run --restart=always --name 1_mongo -v /data/1_mongo/data:/data/db --
network=oncoalrtness --ip=192.168.90.202 --restart=always --log-driver json-file --log-opt
max-size=10m -d mongo:4.2.6

cp ./oncoalrtness_forms_auth.dump /data/1_mongo/data/
docker exec 1_mongo mongorestore --archive=/data/db/oncoalrtness_forms_auth.dump
cp ./oncoalrtness_forms_metadata.dump /data/1_mongo/data/
docker exec 1_mongo mongorestore --archive=/data/db/oncoalrtness_forms_metadata.dump
cp ./oncoalrtness_forms_objects.dump /data/1_mongo/data/
docker exec 1_mongo mongorestore --archive=/data/db/oncoalrtness_forms_objects.dump
```

4. Установка компонентов системы

4.1. Auth

```
mkdir -p /data/oncoalertness/forms/auth
cp -rv ./oncoalertness_forms_auth-2.25.0/config /data/oncoalertness/forms/auth/
отредактировать /data/oncoalertness/forms/auth/config/config.json
- исправить соединение с базой данных, если использовалась сторонняя
- исправить адрес с http://oncoalertness-demo.teh-lab.ru на используемый адрес сервера
docker load -i ./oncoalertness_forms_auth-2.25.0/oncoalertness_forms_auth-2.25.0.tar.gz
docker run --name oncoalertness_forms_auth_1 --restart=always --log-driver json-file --log-
opt max-size=10m --network=oncoalertness --ip=192.168.90.2 -v
/data/oncoalertness/forms/auth/config:/workspace/config -d oncoalertness_forms_auth:2.25.0
```

4.2. Metadata

```
mkdir -p /data/oncoalertness/forms/metadata
cp -rv ./oncoalertness_forms_metadata-1.22.0/config /data/oncoalertness/forms/metadata/
- отредактировать /data/oncoalertness/forms/metadata/config/config.json
- исправить соединение с базой данных, если использовалась сторонняя
- исправить адрес с http://oncoalertness-demo.teh-lab.ru на используемый адрес сервера
docker load -i ./oncoalertness_forms_metadata-1.22.0/oncoalertness_forms_metadata-
1.22.0.tar.gz
docker run --name oncoalertness_forms_metadata_1 --restart=always --log-driver json-file --
log-opt max-size=10m --network=oncoalertness --ip=192.168.90.7 -v
/data/oncoalertness/forms/metadata/config:/workspace/config -d
oncoalertness_forms_metadata:1.22.0
```

4.3. Objects

```
mkdir -p /data/oncoalertness/forms/objects
cp -rv ./oncoalertness_forms_objects-1.36.0/config /data/oncoalertness/forms/objects/
- отредактировать /data/oncoalertness/forms/objects/config/config.json
- исправить соединение с базой данных, если использовалась сторонняя
- исправить адрес с http://oncoalertness-demo.teh-lab.ru на используемый адрес сервера
docker load -i ./oncoalertness_forms_objects-1.36.0/oncoalertness_forms_objects-1.36.0.tar.gz
docker run --name oncoalertness_forms_objects_1 --restart=always --log-driver json-file --log-
opt max-size=10m --network=oncoalertness --ip=192.168.90.8 -v
/data/oncoalertness/forms/objects/config:/workspace/config -d
oncoalertness_forms_objects:1.36.0
```

4.4. Calculation

```
mkdir -p /data/oncoalrtness/forms/calculation
cp -rv ./oncoalrtness_forms_calculation-1.5.0/config /data/oncoalrtness/forms/calculation/
- отредактировать /data/oncoalrtness/forms/calculation/config/config.json
- исправить адрес с http://oncoalrtness-demo.teh-lab.ru на используемый адрес сервера
docker load -i ./oncoalrtness_forms_calculation-1.5.0/oncoalrtness_forms_calculation-1.5.0.tar.gz
docker run --name oncoalrtness_forms_calculation_1 --restart=always --log-driver json-file --log-opt max-size=10m --network=oncoalrtness --ip=192.168.90.4 -v /data/oncoalrtness/forms/calculation/config:/workspace/config -d oncoalrtness_forms_calculation:1.5.0
```

4.5. Frontend (Административной панели)

```
mkdir -p /data/oncoalrtness/forms/frontend
cp -rv ./oncoalrtness_forms_frontend-4.7.0/config /data/oncoalrtness/forms/frontend/
- отредактировать /data/oncoalrtness/forms/frontend/config/config.json
- исправить адрес с http://oncoalrtness-demo.teh-lab.ru на используемый адрес сервера
docker load -i ./oncoalrtness_forms_frontend-4.7.0/oncoalrtness_forms_frontend-4.7.0.tar.gz
docker run --name oncoalrtness_forms_frontend_1 --restart=always --log-driver json-file --log-opt max-size=10m --network=oncoalrtness --ip=192.168.90.6 -v /data/oncoalrtness/forms/frontend/config/config.json:/usr/share/nginx/html/assets/config.json -v /data/oncoalrtness/forms/frontend/config/branding:/usr/share/nginx/html/assets/branding -d oncoalrtness_forms_frontend:4.7.0
```

4.6. AI

```
mkdir -p /data/oncoalrtness/oncoalrtness/ai
docker load -i ./oncoalrtness_oncoalrtness_ai-1.3.1/oncoalrtness_oncoalrtness_ai-1.3.1.tar.gz
docker run --name oncoalrtness_oncoalrtness_ai_1 --restart=always --log-driver json-file --log-opt max-size=10m --network=oncoalrtness --ip=192.168.90.130 -v /data/oncoalrtness/oncoalrtness/ai/data:/workspace/data -d oncoalrtness_oncoalrtness_ai:1.3.1
```

4.7. Backend

```
mkdir -p /data/oncoalrtness/oncoalrtness/backend
cp -rv ./oncoalrtness_oncoalrtness_backend-1.3.0/config /data/oncoalrtness/oncoalrtness/backend/
- отредактировать /data/oncoalrtness/oncoalrtness/backend/config/production.config
```

- исправить адрес с <http://oncoalrtness-demo.teh-lab.ru> на используемый адрес сервера
docker load -i ./oncoalrtness_oncoalrtness_backend-1.3.0/oncoalrtness_oncoalrtness_backend-1.3.0.tar.gz
docker run --name oncoalrtness_oncoalrtness_backend_1 --restart=always --log-driver json-file --log-opt max-size=10m --network=oncoalrtness --ip=192.168.90.131 -v /data/oncoalrtness/oncoalrtness/backend/config:/workspace/config -v /data/oncoalrtness/oncoalrtness/backend/config/backend.env:/workspace/.env -d oncoalrtness_oncoalrtness_backend:1.3.0

4.8. Frontend (Онконастороженности)

mkdir -p /data/oncoalrtness/oncoalrtness/frontend
cp -rv ./oncoalrtness_oncoalrtness_frontend-4.7.0/config /data/oncoalrtness/oncoalrtness/frontend/
- отредактировать /data/oncoalrtness/oncoalrtness/frontend/config/config.json
- исправить адрес с <http://oncoalrtness-demo.teh-lab.ru> на используемый адрес сервера
docker load -i ./oncoalrtness_oncoalrtness_frontend-4.7.0/oncoalrtness_oncoalrtness_frontend-4.7.0.tar.gz
docker run --name oncoalrtness_oncoalrtness_frontend_1 --restart=always --log-driver json-file --log-opt max-size=10m --network=oncoalrtness --ip=192.168.90.132 -v /data/oncoalrtness/oncoalrtness/frontend/config/config.json:/usr/share/nginx/html/assets/config.json -v /data/oncoalrtness/oncoalrtness/frontend/config/branding:/usr/share/nginx/html/assets/branding -d oncoalrtness_oncoalrtness_frontend:4.7.0

5. Проверка работоспособности

Перейдите на <http://oncoalertness-demo.teh-lab.ru> (заменить oncoalertness-demo.teh-lab.ru на IP адрес сервера или доменное имя, которое использовали при настройке nginx)

Войдите в систему пользователей с ролями «врач-терапевт» и «врач-онколог»:

Терапевт:

Логин: doctor

Пароль: Qt']1IS&

Онколог:

Логин: oncologist

Пароль: r^W=Gq5[

Перейдите на <http://oncoalertness-demo.teh-lab.ru/forms>

Войдите в систему пользователя с ролью администратора:

Администратор:

Логин: admin

Пароль: mgS6#)k&

**Ценовая политика программного обеспечения «ПРОГРАММА ДЛЯ ПОМОЩИ В
ДИФФЕРЕНЦИАЛЬНОЙ ДИАГНОСТИКЕ GALENOS.AI»**

Минимальная стоимость развертывания решения составляет 100 000 рублей.
Стоимость лицензии на компонент, группу компонентов или весь программный комплекс зависит от типа лицензии. Дополнительная информация предоставляется по запросу на адрес электронной почты: info@teh-lab.r