Техническое задание на разработку
системы мониторинга и управления ПМВИ Вертикаль

от 18.07.2023

1 Введение

1.1 Наименование программы

Система мониторинга и управления ПМВИ «Вертикаль»

1.2 Краткая характеристика области применения

Система мониторинга и управления ПМВИ «Вертикаль» представляет собой веб-приложение для управления и является пользовательским интерфейсом, предназначенным для человеко-машинного взаимодействия между оператором нескольких физических серверов с предустановленной виртуальной машиной, называемой «Гостевая операционная система» (гостевая ОС). Приложение предназначено для управления серверами и гостевой ОС по сети (включение, выключение, мониторинг) как единой системой.

2 Основания для разработки

Основанием для разработки является служебное задание (приказ) №

Согласно заданию Разработчик обязан разработать и передать код, исполняемые файлы и сопроводительную документацию.

Наименование темы разработки: «Проектирование и разработка системы мониторинга и управления (программное обеспечение, совокупность динамических страниц, веб-приложение)»

Условное обозначение темы разработки (шифр темы) – «_____».

3 Назначение разработки

Система мониторинга и управления ПМВИ «Вертикаль» (веб-приложение) будет использоваться в демонстрационных целях либо как часть программного комплекса для внедрения и развертывания больших систем (программно определяемых серверов, ПОС) на оборудовании Заказчика.

3.1 Функциональное назначение

Для специалиста по развертыванию результатом работы будет папка, содержащая исполняемый код веб-приложения, которая может поставляться как часть пакета, предназначенного для установки ПМВИ «Вертикаль». Данная папка размещается по заранее заданному пути.

Для оператора управления результатом является веб-приложение, позволяющее выполнять над рабочими узлами и программно-определяемыми серверами набор действий, описанный в «Руководстве пользователя ПМВИ «Вертикаль».

Для пользователя веб-приложения результатом работы является часть системы, позволяющей осуществлять мониторинг состояния ПОС и физических рабочих узлов, находящихся под управлением ПМВИ «Вертикаль».

Для инженера по разработке ПО результатом работы является исходный код продукта, включающий в себя инструкцию по сборке.

Веб-приложение работает на компьютере под управлением операционной системы Linux, взаимодействует с серверной частью, расположенной на той же машине локально использования цифровой платформы является виртуальный компьютер с согласованной операционной системой семейства Linux, обладающий аппаратными ресурсами, составляющими не менее 70% аппаратных ресурсов рабочих узлов как единым целым с сохранением полного функционала ПО, установленного на данной операционной системе.

Требования к информационной и программной совместимости для данной платформы являются определяющими и перечислены в разделе 4.5 данного технического задания.

3.2 Эксплуатационное назначение

Программа будет эксплуатироваться в системе больших данных и программ работы с ними. Специфика заключается в исполнении операций над данными, объем которых в памяти может превышать объем оперативной памяти каждого из отдельно взятых рабочих узлов.

4 Требования к программе или программному изделию

4.1 Требования к функциональным характеристикам

4.1.1 Требования к составу выполняемых функций

Развертывание системы мониторинга и управления осуществляется согласно «Руководству по эксплуатации».

Работа с системой осуществляется посредством Web-интерфейса, описанного в «Руководстве пользователя».

Все изменения, выполняемые пользователем должны сохраняться в системе.

Функционал веб-приложения полностью определяется графическими элементами взаимодействия пользователя с веб-приложением для управления демонстрационной серверной частью, располагающейся в закрытом контуре Заказчика.

Приложение содержит 15 основных страниц-форм взаимодействия с пользователем и несколько дополнительных окон-форм (не вызывающих переходов между страницами).

Графический интерфейс управления включает в себя:
1. Страница входа (/login)
2. Основная страница приложения (/home)
3. Страница активности (/activity)
4. Страница сообщений (/alerts)
5. Страница настроек (/settings)
6. Страница списка пулов (/pool/list)
7 Страница создания пулов (/pool/create)
8. Страница описания пула (/pool/:uuid)
9. Страница редактирования пула (/pool/edit/:uuid)
10. Страница списка узлов (/node/list)
11. Страница описания узла (/node/:uuid)
12. Страница списка ПОС (/sds/list)
13. Страница создания ПОС (/sds/create)
14. Страница редактирования ПОС (/sds/edit/:uuid)
15 Страница расширенного редактирования ПОС (/sds/edit/advanced/:uuid)

Навигация между страницами осуществляется по следующим правилам:

Стартовой страницей является страница авторизации. При успешном входе устанавливается WebSocket-соединение.

При старте приложения создается подписка для получения уведомлений о состоянии входа. В случае отсутствия активности в течение X минут (X — параметр, может принимать значения от 1 до 240 минут) приложение принимает сообщение о скором завершении и выдает пользователю соответствующее предупреждение. В случае отсутствия реакции на сообщение требуется повторный вход (т. е. с любой страницы автоматически происходит переадресация на страницу входа). В случае подтверждения отсчёт времени начинается заново.

В ситуации, когда вход успешно осуществлён, допускается любая последовательность переходов между страницами, в том числе прямое указание требуемой страницы в адресной строке браузера.

Взаимодействие с серверной частью приложения

Всё взаимодействие с серверной частью платформы осуществляется в рамках установленного соединения по технологии WebSocket (WebSocket API) между веб-приложением и приложением, установленном на той же машине.

Описание протокола взаимодействия между веб-приложением и серверной частью является частью разработки веб-приложения и предоставляется одновременно со сдачей веб-приложения.

Взаимодействие включает в себя последовательную отправку запросов для осуществления управления платформой с помощью графических элементов управления и отображение сообщений, получаемых от сервера (как в ответ на запрос, так и служебных, поступающих по сокету по подписке).

Все сообщения (как запросы, так и ответы) имеют формат JSON согласно схеме, входящей в состав серверной части ПО.

Каждое сообщение со стороны веб-приложения к серверу включает в себя команду-запрос и уникальный в рамках сессии идентификатор, состоящий из 10 десятичных цифр (correlator).

Получаемый от сервера ответ должен соответствовать json-схеме, предоставляемой вместе с серверной частью платформы, и содержать идентификатор запроса. Получаемое от сервера информационное сообщение должно соответствовать json-схеме и включать в себя идентификатор типа сообщения.

Схемы и наполнение страниц

1. Страница входа (/login)

Содержит два поля ввода (логин, пароль) и кнопку «Войти».

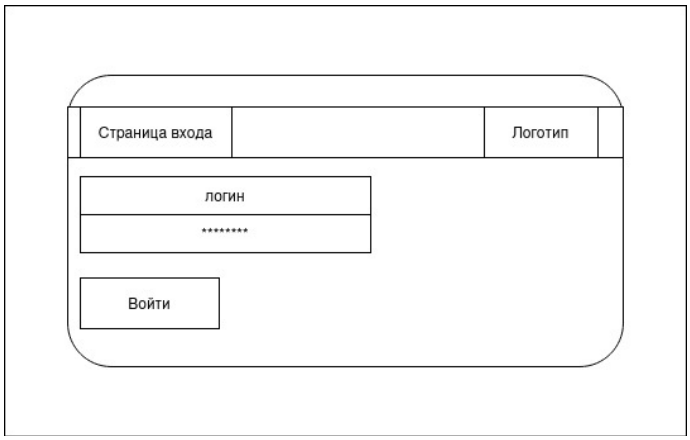
При нажатии на кнопку устанавливается соединение с сервером, отправляется запрос, содержащий пару логин-пароль и осуществляется проверка ответа.

В случае неуспешной проверки пользователь получает сообщение о том, что логин или пароль неверен.

В случае успешного входа компонентом страницы входа осуществляется мониторинг статуса соединения от сервера и срока «жизни» авторизации. При необходимости осуществляется автоматический повторный вход.

При разрыве соединения (как со стороны сервера, так и со стороны клиента) пользователь получает сообщение о потере связи. После этого необходимо осуществить повторный вход.

Макет страницы входа



2. Основная страница приложения (/home)

Включает в себя разделы, изображенные на рис.: верхнее меню, боковая панель, основной раздел с описанием текущей конфигурации системы — ПОС и узлов, подразделом для сообщений и лога активности, нижнюю панель с версией ПО.

Для получения данных для отображения на этой форме отправляется запрос(ы) к серверной части ПО. Наполнение разделов осуществляется на основании ответного сообщения.

После того, как выполнен запрос на чтение всей доступной информации о системе, веб-приложение получает изменение (дельту) статуса системы в виде последовательных сообщений.

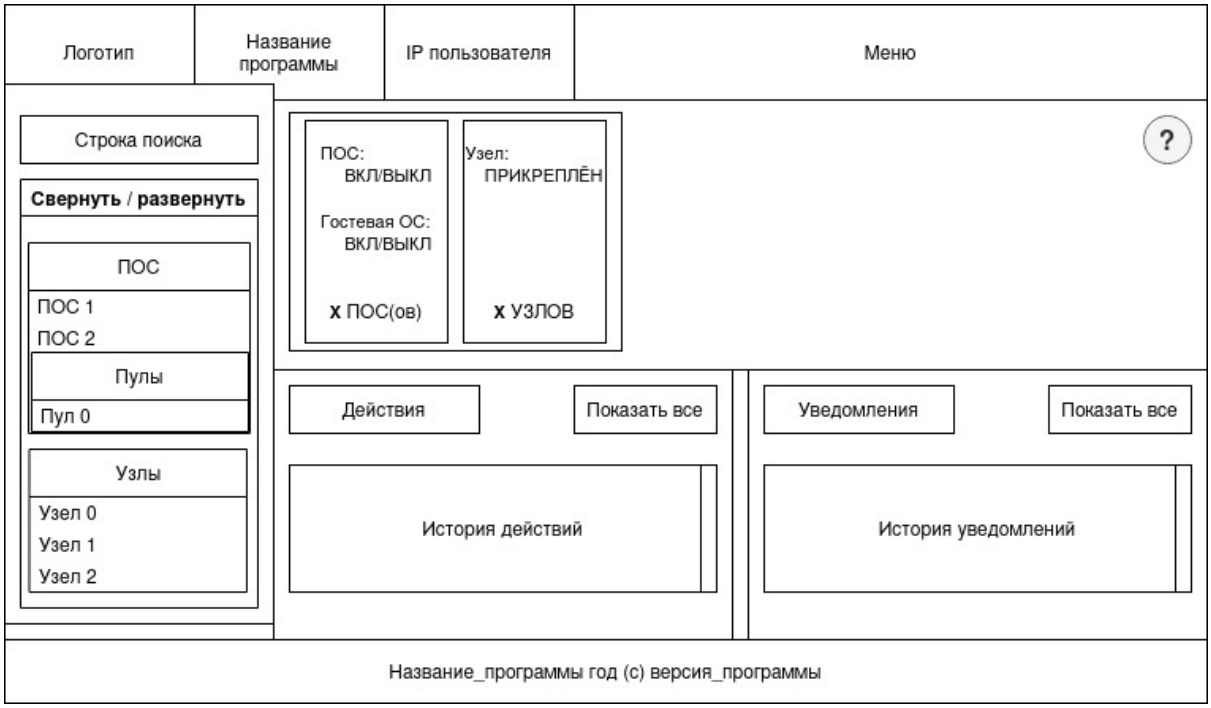
Помощь к странице размещается в отдельном разделе специального файла и содержит как русскую, так и английскую версии размеченного текста.

Действия доступные с основной страницы:

- просмотр текущего состояния;
- поиск по ключевым словам, содержащимся в ответе (строка поиска размещается на боковой панели);
- просмотр помощи (при нажатии на специальную иконку);
- переход на страницы списков узлов, ПОС, пулов (при нажатии в соответствующей области или на ссылку);
- переход на страницы детализации узлов, ПОС, пулов (при нажатии на область объекта или ссылку);
- переход к окну настроек;

- ❖ просмотр лога событий;
- ❖ просмотр лога активности;
- ❖ просмотр версии клиентской и серверной части ПО;
- ❖ просмотр IP, с которого осуществлён вход;
- ❖ выход из системы.

Макет основной страницы



3. Страница активности (/activity)

Замещает основной раздел. В верхней части содержат форму для фильтрации событий включая строку поиска, и даты начала и конца периода поиска.

Каждое событие содержит ссылку на страницу объекта, к которому оно относится.

Данные об активности поступают по websocket в виде json-сообщений, соответствующих схеме.

4. Страница сообщений (/alerts)

Замещает основной раздел. Содержит информацию о количестве сообщений, а так же о соотношении новых относительно всех. Содержит фильтры. Списки фильтров заполняются автоматически на основе информации, полученной от сервера по факту обращения к элементу управления посредством отправки запроса серверу или используя информацию, сохранённую в локальном хранилище приложения.

Каждый элемент (сообщение) содержит информацию о дате и времени получения сообщения, принадлежности объекту, а так же статус, который может быть изменён (прочитано, не прочитано, удалить).

Предусмотрено групповое выделение сообщений.

Данные о событиях поступают по websocket в виде json-сообщений, соответствующих схеме.

5. Страница настроек (/settings)

Содержит окно для ввода единственного параметра: времени, через которое пользователю будет сообщено об отсутствии активности и произойдёт автоматический выход из системы.

6. Страница списка пулов (/pool/list)

Переход на страницу отображения списка пулов осуществляется с боковой панели или при клике в соответствующей части основного раздела.

Содержит кнопку перехода на страницу _*создания пула*_.

Включает в себя информацию о количестве ПОС, рабочих узлов и рабочих узлов, незадействованных в перечисленных ПОС.

7. Страница создания пулов (/pool/create)

Переход на страницу создания пулов осуществляется при нажатии на единственную кнопку «Создать пул» со страницы *Список пулов*. При этом открывается модальное окно (см. рис.)

Форма содержит элементы управления:

* Поле ввода «Название пула» — строка ввода, обязательное поле не менее 5 символов (проверка), допускаются как латиница, так и кириллица;
* Поле ввода «Описание» — необязательное текстовое поле;
* Выпадающий список «Тип узла» — список из двух вариантов: на железе (bare metal) и виртуальная машина/эмуляция (demo);
* Флаг «самосохранение» со всплывающей подсказкой при нажатии на иконку «помощь»;
* Кнопки «Отмена» и «Сохранить». При нажатии на кнопку «Сохранить» будет отправлен запрос на сервер с командой записи и новый пул будет создан. При этом поступит ответное сообщение от сервера, на основании которого приложение должно обновить соответствующие разделы формы (списки и количество пулов). Так же становится доступна страница с идентификатором вновь созданного пула. При переходе на неё открывается страница *Детали пула*.

8. Страница описания пула (/pool/:uuid)

Содержит:

* иконку открытия помощи;
* 4 (четыре) кнопки: настройки (редактирование пула), удаление пула, созадание ПОС, импорт узлов;
* два списка: список ПОС, ассоциированных с данным пулом, и список узлов в этом пуле.

По нажатию кнопки «Настройки» открывается страница редактирования пула.
По нажатию кнопки «Удалить» проверяется, что пул не содержит активные ПОС и рабочие узлы. Только в этом случае пул может быть удалён.
Импорт узлов открывает модальное окно, предназначенное для обнаружения узлов по сети BMC (без перехода на новую страницу). Добавление обнаруженных узлов в пул осуществляется путём отправки соответствующего запроса с командой записи на сервер. В ответном сообщении поступит обновление данных для отображения. После успешного добавления узлов они должны появиться во всех списках узлов.

9. Страница редактирования пула (/pool/edit/:uuid)

Повторяет по своей структуре страницу создания пула за исключением того, что в окне редактирования отсутствует выбор типа узла.

10. Страница списка узлов (/node/list)

Содержит:

* иконку открытия помощи;
* 4 (четыре) кнопки: настройки узла (редактирование деталей узла), настройка управляющего интерфейса для узла, удаление удаление узла и расширенные настройки узла;
* визуализацию параметров узла (статус: ВКЛ/ВЫКЛ, использован в ПОС/не используется; кол-во ядер CPU, количество оперативной памяти, объем доступного хранилища (дисков))
* 4 (четыре) страницы-”таба” с параметрами:
    ** состояние узла – параметры мониторинга “здоровья” с помощью, получаемое с помощью ipmitool в формате json (память, сетевые интерфейсы - статус линка, статистика по интерфейсам, параметры температуры, напряжений и системы охлаждения); для этих данных указывается дата и время их последнего получения, так же возможен запрос на принудительное обновление данных;
    ** детализация информации по обнаруженным сетевым интерфейсам (управляющая, интерконнект, гостевые);
    ** системная информация: идентификатор узла, параметры cpu и памяти;
    ** информация о дисках в системе (вендор, модель, полный объем).

В описанных разделах предоставляются данные, которые возможно получить станаджртными утилитами системы FreBSD (in-memory), под управлением которой находится каждый рабочий узел.

По нажатию кнопки «Настройки» открывается *_окно редактирования узла_*, аналогичная странице *_ создание узла_*.
По нажатию кнопки «Удалить» проверяется, что узел не содержится в активном ПОС. Только в этом случае узел может быть удалён. В противном случае пользователь получает сообщение об ошибке.
«Расширенная детализация» открывает новое модальное окно, содержащее редактор json (на основе ace-jsoneditor). Структура json соответствует json-схеме серверной части, данные - ответу от сервера. При редактировании данных на сервер поступает запрос на запись, после чего в ответном сообщении обновленные данные или сообщение об ошибке.

11. Страница редактирования узла (/pool/edit/:uuid)

Повторяет по своей структуре страницу создания пула за исключением того, что в окне редактирования отсутствует выбор типа узла.

11. Страница списка ПОС (/sds/list)

Содержит:

* иконку открытия помощи;
* 1 (одну) кнопку: озадание ПОС;

* описание ПОС (количество, статус, состояние гостевой JC)
* 1 список ПОС, содержащий информацию о гостевой системе, работающей под его управлением

12. Страница создания ПОС (/sds/create)
Аналогично странице создания пулов

13. Страница описания ПОС (/sds/:uuid)

Содержит:

* иконку открытия помощи;
* 4 (четыре) кнопки: настройки ПОС (редактирование деталей ПОС, включая расширенные настройки), удаление удаление узла, добавление узла к существующему ПОС, запуск терминала гостевой системы (терминал представляет собой компонент подключения к узлу по ssh, консольный ввод и вывод осуществляются посимвольно через специальную подписку поверх websocket);
* визуализацию параметров ПОС игоствеой ОС (кол-во ПОС, статус: ВКЛ/ВЫКЛ, кол-во ядер CPU, количество оперативной памяти, объем доступного хранилища) ;
* список узлов (аналогично старнице _список_узлов_)

14. Страница расширенного редактирования ПОС (/sds/edit/advanced/:uuid)
Представляет собой ace_jsoneditor аналогично расширенной настройке узлов

15. Страница редактированияПОС (/sds/edit/:uuid)

Позволяет осуществлять добавление образа гостевой операционной системы и автоматическое обнаружение его на файловой системе. Данные поступают в ответ на запрос при просмотре доступных образов (выпадающий список).

Детализация этой страницы будет уточнена в процессе разработки и тестирования версии приложения.

Дополнительные разделы/окна (не имеющие собственной точки входа)

1. Выбор языка

Реализован в виде переключателя EN/RU. По умолчанию используется русский язык.

Переключение языка осуществляется при помощи изменения файла конфигурации языка и дополнительного файла соответствий ключевых слов из схемы ответов отображаемым в приложении подписям.

На этапе создания серверной части функционал перевода сообщений может быть перенесен из веб-приложения на сторону сервера.

2. Окно помощи

Реализуется в виде окна, всплывающего при нажатии на иконку «Вопрос». Содержимое этого окна зависит от того, с какой страницы осуществляется вызов. Помощь выдается на выбранном языке. По умолчанию русский.

3. Окно обнаружения узлов

Модальное окно, открывающееся при импорте узлов в пул. Содержит 4 поля ввода (диапазон IP-адресов + логин и пароль доступа к физическому узлу или узлам по IPMI), флаг «перевключения питания» при обнаружении включенного узла, кнопку «Обнаружить».

Нажатие на кнопку «Обнаружить» инициирует появление окна выбора обнаруженных узлов (без перехода на новую страницу). После выбора хотя бы одного узла можно добавить эти узлы в пул. После добавления форма обнаружения узла будет закрыта.

4. Окно конфигурации сетей при импорте узлов

Модальное окно со списками доступных настроек. Позоволяет сконсфигурировать сеть для узла. В случае, если узел несконфигурирован, кнопка добавления узла неактивна.

5. Окно редактирования узла

Модальное окно, содержащее Ace json editor. Позволяет осуществлять “сырое” редактирование файла, который размещается на сервере с попыткой применения изменений путём отправки запроса типа write по websocket.

По мере тестирования возможно появление страницы редактирования, аналогично странице редактирования ПОС.

4.1.2 Требования к организации входных и выходных данных

Взаимодействие с пользователем осуществляется посредством графического интерфейса, который на основе действий пользователя формирует запросы к серверу в формате JSON и получает на них ответы, а так же служебные сообщения так же в формате JSON по заданной серверной частью схеме. В качестве канала взаимодействия WebSocket API.

4.1.3 Требования к временным характеристикам

Задержки при взаимодействии пользователя с веб-интерфейсом и между сервером и веб-приложением не должны превышать ожидаемые от данного типа взаимодействия. Допускаются задержки при обработке информации до 2с при чтении данных от сервера, до 1с при человеко-машинном взаимодействии.

4.2 Требования к надежности

Вероятность безотказной работы системы должна составлять не менее вероятности безотказной работы оборудования, использующегося в системе.

4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- организацией бесперебойного питания технических средств;
- использованием лицензионного программного обеспечения;
- регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
- соблюдением правил размещения информации на серверах в соответствии с требованиями продуктов, использующихся на основе платформы.

4.2.2 Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30 минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

4.2.3 Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий пользователя при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу пользователя без предоставления ему административных привилегий.

4.3 Условия эксплуатации

4.3.1 Климатические условия эксплуатации

Специальные условия не требуются.

4.3.2 Требования к видам обслуживания

Программа не требует проведения каких-либо видов обслуживания.

4.3.3 Требования к численности и квалификации персонала

При установке и настройке системы необходим специалист по развертыванию. К квалификации пользователя системы специальные требования не предъявляются.

4.4 Требования к составу и параметрам технических средств

Программа выполняет свои функции на любом физическом или виртуальном компьютере с установленной системой Linux. Тестирование осуществляется в ОС Linux Debian 11.

К серверной составляющей, с которой работает система мониторинга и управления, предъявляются специальные требования, перечисленные ниже.

4.4.1. Требования к узлу управления

Административный сервер управления (минимум 1 на инсталляцию). Допускается установка как на физическом компьютере, так и в виртуальную машину (при условии корректной настройки сети).

4.4.2. Требования к рабочим узлам

Каждый рабочий узел должен иметь двухсокетную материнскую плату Intel с поддержкой IPMI 2.0 и сетевые интерфейсы 10Gb (не менее 3х).

4.4.3 Требования к коммуникационному оборудованию

Коммутатор сети Interconnect уровня L3 – коммутатор с портами 10GbE или выше. Необходимое количество портов равно числу рабочих узлов + 1.

Коммутатор(ы) управления / загрузки (сети BMC, PXE и Corporate) – обеспечивают управление, бездисковую загрузку / сеть SAN (iSCSI) и доступ из внешних сетей. Число портов, в случае неиспользования тегированных VLAN, 3 x (n + 1), где n – число рабочих узлов +1.

4.5 Требования к информационной и программной совместимости

4.5.1. Требования к серверной части

Система управления и мониторинга ПМВИ «Вертикаль» должна быть совместима хотя бы с одним продуктом, представленным на рынке и обеспечивать возможность ее протестировать как минимум в режиме эмуляции.

4.5.2 Поддерживаемые браузеры

Браузеры, входящие в дистрибутивы современных систем Linux (официальный релиз не старше 4 лет от даты согласования данного ТЗ): Mozilla FireFox, Google Chrome/Chromium

4.5.3 Языки программирования, используемые технологии и библиотеки.

Основным рабочим языком приложения является java script. Для создания приложения используется фреймворк Angular (версии 16/18), имеющий на дату согласования ТЗ открытый исходный код и распространяющийся бесплатно (лицензия MIT). Основные используемые библиотеки: angular material (https://material.angular.io/, MIT), RxJS(https://rxjs.dev, MIT), Ace editor/jsoneditor (https://ace.c9.io/, BSD).

Взаимодействие с серверной частью осуществляется по WebSocket API (https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API).

4.6 Требование к маркировке и упаковке

Программное изделие передается в виде установочного пакета для развертывания посредством операционной системы из семейства Linux. Предполагаемый формат пакета развертывания – deb. Специальных требований к маркировке не предъявляется. Для проверки подлинности программного обеспечения рекомендуется проверять контрольные суммы загруженных файлов со значениями, со значениями, указанными в документации к поставляемой версии.

4.7 Требования к транспортированию и хранению

Специальных требований не предъявляется.

4.8 Специальные требования

Программа должна обеспечивать взаимодействие с пользователем посредством графического пользовательского Web-интерфейса в заданной цветовой схеме. Взаимодействие пользователя с программой осуществляется на русском языке.

5 Требования к программной документации

Предварительный состав программной документации, по содержанию, но не по форматированию соответствующие указанным стандартам:

- техническое задание (ГОСТ 19.201-78);
- программа и методика испытаний (ГОСТ 19.301-79);
- руководство системного программиста (ГОСТ 19.503-79);
- руководство оператора (ГОСТ 19.505-79);
- руководство программиста (ГОСТ 19.504-79);
- ведомость эксплуатационных документов (ГОСТ 19.507-79);
- формуляр (ГОСТ 19.501-78).

6 Технико-экономические показатели

Экономический эффект может быть обеспечен за счет платной установки в составе программного комплекса на оборудовании покупателей.

7 Стадии и этапы разработки

Этапы разработки

I. Изучение функциональных возможностей приложения. Анализ ТЗ.
II. Работа над архитектурой приложения. Выделение объектов и функционала, необходимых для демонстрации работы приложения. Создание минимального работающего прототипа (MVP).
III. Локализация интерфейса управления.
IV. Создание всех страниц приложения.
V. Тестирование логики переходов между страницами.
VI. Демонстрация предварительной версии.
VII. Согласование форм и функционального наполнения версии.
VIII. Наполнение страниц приложения
IX. Детальное описание протокола взаимодействия веб-приложения с серверной частью.
X. Создание списка (чек-листа) основного функционала для приемо-сдаточных испытаний.
XI. Создание и согласование “Программы и методики испытаний” (на основе ГОСТ 19.301-79)
XII. Создание и согласование сопроводительной программной документации.
XIII. Тестирование согласованного функционала, проведение приемо-сдаточных испытаний.

8 Порядок контроля и приемки

Приемо-сдаточные испытания программы должны проводиться согласно разработанной Исполнителем в процессе работы над продуктом и согласованной Заказчиком «Программой и методикой испытаний».

Ход проведения приемо-сдаточных испытаний Заказчик и Исполнитель документируют в Протоколе Испытаний.

На основании Протокола Испытаний Исполнитель совместно с Заказчиком подписывают Акт приемки-сдачи программного продукта в эксплуатацию.

Список используемой литературы

1. ГОСТ 19.201-78 Единая система программной документации. Техническое задание. Требования к содержанию и оформлению. 1978. Режим доступа: http://protect.gost.ru/document.aspx?control=7&id=155153
2. ГОСТ 19.301-79 Единая система программной документации. Программа и методика испытаний. Требования к содержанию и оформлению. 1979. Режим доступа: https://protect.gost.ru/document.aspx?control=7&id=153589
3. ГОСТ 19.101-77 Единая система программной документации. Виды программ и программных документов. 1979. Режим доступа: https://protect.gost.ru/v.aspx?control=7&id=158686
4. ГОСТ 24.701-86. Единая система стандартов автоматизированных систем управления. Надежность автоматизированных систем управления. Основные положения. М.: Издательство стандартов, 1987. — 17 с.