Министерство образования и науки Нижегородской области

Государственное бюджетное профессиональное образовательное учреждение «Нижегородский Губернский колледж»

Допущен к защите:

преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_ Е.А. Романова

\_\_\_\_\_\_\_\_\_\_\_\_ Ю.С. Мамшева

«03» июля 2024 г.

**ОТЧЕТ**

**ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

ПМ.02 ОСУЩЕСТВЛЕНИЕ ИНЕТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ

Руководитель

от предприятия: \_\_\_\_\_\_\_\_\_\_Шмелёва Е.С. 03.07.2024г.

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Чернов С.В. 03.07.2024г.

Специальность, группа: 09.02.07, 33П

Нижний Новгород

2024г.

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc170682515)

[**1.** **ТЕОРИТИЧЕСКАЯ ЧАСТЬ** 5](#_Toc170682516)

[**1.1.** **Организационная структура предприятия** 5](#_Toc170682517)

[1.1.1. Концепция предприятия 5](#_Toc170682518)

[1.1.2. Описание структуры предприятия 5](#_Toc170682519)

[**1.2.** **Анализ программного обеспечения и технического оснащения** 7](#_Toc170682520)

[1.2.1. Системное обеспечение 7](#_Toc170682521)

[1.2.2. Программное обеспечение 7](#_Toc170682522)

[1.2.3. Топология сети предприятия 8](#_Toc170682523)

[**1.3.** **Описание инструментальных средств разработки и сопровождения программных средств.** 9](#_Toc170682524)

[**2.** **ПРАКТИЧЕСКАЯ ЧАСТЬ** 10](#_Toc170682525)

[**2.1.** **Анализ предметной области** 10](#_Toc170682526)

[2.1.1. Анализ требований и информационных потребностей 10](#_Toc170682527)

[2.1.2. Выявление информационных объектов и связей между ними 12](#_Toc170682528)

[**2.2.** **Проектирование программного продукта** 13](#_Toc170682529)

[**2.3.** **Разработка программного модуля** 13](#_Toc170682530)

[**2.4.** **Тестирование программного продукта** 13](#_Toc170682531)

[**ЗАКЛЮЧЕНИЕ** 15](#_Toc170682532)

[**СПИСОК ИСПОЛЬЗОВАНОЙ ЛИТЕРАТУРЫ** 16](#_Toc170682533)

[**ПРИЛОЖЕНИЯ** 18](#_Toc170682534)

[**Приложение А** 18](#_Toc170682535)

[**Приложение Б** 19](#_Toc170682536)

[**Приложение В** 20](#_Toc170682537)

[**Приложение Г** 23](#_Toc170682538)

**ВВЕДЕНИЕ**

Производственная практика по ПМ.02 «Осуществление интеграции программных модулей» проходящая в организации АО «НЗ 70-летия Победы», предоставляет возможность применить знания, полученные в ходе обучения и учебной практики в реальном производственном процессе, а именно выполнения задач, поставленных руководителем практики от предприятия.

Целью производственной практики является, вышеупомянутое, применение знаний на практике посредством создания модуля продукта необходимого предприятию, который будет полу автоматизировать работу с заявками на исправление каких-либо нарушений работ систем или на улучшения в предприятии от его работников и необходимой документацией для составляемых заявок.

Установленная цель, при выполнении, решает проблему, которой, в данном случае, является создание модуля программного интерфейса приложения (API), которое будет встроено в общую систему в соответствии с требованиями заказчика.

Практическим результатом выполнения задания производственной практики является API для взаимодействия с базой данных и документами, созданный для дальнейшей разработки пользовательского интерфейса сторонними разработчиками.

API (англ. «Application Programming Interface» — программный интерфейс приложения) — это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными. Он встречается практически во всех проектах, связанных с работой с пользовательскими данными как общедоступными, так и закрытыми, поскольку позволяет разработчикам связывать разные программные модули и программы между собой, делая связь проще и безопаснее.

Для достижения цели были поставлены следующие задачи:

* повторить материал, пройденный на учебной практике;
* получить требования к разрабатываемому модулю;
* описать структуру системы в соответствии с требованиями к разрабатываемому модулю;
* описать структуру взаимодействия системы с пользователем;
* разработать модуль информационной системы;
* создать тестовую документацию и протестировать разработанный модуль информационной системы;
* провести защиту презентации продукта.

Основными методами решения задач будут являться: логический, практический, комбинаторный.

1. **ТЕОРИТИЧЕСКАЯ ЧАСТЬ**
   1. **Организационная структура предприятия**
      1. Концепция предприятия

Нижегородский завод 70-летия Победы, предприятие оборонной промышленности в Нижнем Новгороде.

Создан в 2014 г. на базе существовавшего с 1932 г. Нижегородского машиностроительного завода (специализация – ракетные комплексы, артиллерийские системы, атомные корабельные установки), входит в структуру Концерна воздушно-космической обороны «Алмаз-Антей» со штаб-квартирой в Москве.

Значимое машиностроительное предприятие Нижнего Новгорода (свыше 5 тыс. занятых в 2021), является одним из крупнейших производителей перспективного вооружения и военной техники, обеспечивающих выполнение государственного оборонного заказа.

Дополнительные специализации гражданского назначения – карьерная и сельскохозяйственная техника, крановое оборудование и системы хранения.

Основным заказчиком являются вооружённые силы России и другие государственные военно-исследовательские и сельскохозяйственные предприятия.

Лозунг предприятия: «Мирное небо – наша работа!».

* + 1. Описание структуры предприятия

Поскольку предприятие закрытое, то описать можно только структуру отдела, где проходила практика.

* + - 1. Деятельность отдела прохождения практики

Деятельностью отдела прохождения практики являются:

* разработка, внедрение, настройка и сопровождение информационных систем на основе экономико-математических методов, современных средств вычислительной техники, коммуникации и связи, элементов теории экономической кибернетики;
* исследование подсистем систем с целью определения возможности их формализации и целесообразности перевода соответствующих процессов на автоматический режим;
* подготовка планов проектирования и внедрения подсистем на предприятие и контроль их выполнения;
* организация работы по совершенствованию электронного документооборота по процессам соответствующих подсистем;
* осуществление организационного и технического взаимодействия подсистем предприятия;
* проектирование и разработка технологических схем обработки информации по всем задачам предприятия и технологических процессов обработки информации при помощи средств вычислительной техники;
* обеспечение бесперебойного функционирования информационных систем предприятия на основе анализа и изучения проблем обслуживания и принятие оперативных мер по устранению, возникающих в процессе работы нарушений.
* разработка инструкций, методических и нормативных материалов, связанных с информационным обеспечением подсистем предприятия.
  + - 1. Структура отдела и должности сотрудников

Отдел информационных систем (ОИС) является структурным подразделением управления информационных технологий (УИТ) и подчиняется начальнику управления информационных технологий.

ОИС возглавляет начальник отдела информационных систем, подчиняющийся начальнику УИТ.

Начальнику ОИС в установленном порядке подчиняется заместитель начальника отдела и руководители групп и бюро.

В соответствии с направлениями деятельности и выполняемыми задачами структура отдела следующая (наглядно представлена на рисунке А1):

* группа управления производством и развития;
* группа управления ресурсами предприятия;
* группа нормативно-справочной информации;
* группа сопровождения информационных систем;
* бюро программного обеспечения собственной разработки.

На время отсутствия начальника отдела, руководство отделом осуществляется работником, назначенным в установленном порядке.

ОИС взаимодействует со всеми структурными подразделениями предприятия, осуществляя сопроводительную и управленческую деятельности над информационными системами.

* 1. **Анализ программного обеспечения и технического оснащения**
     1. Системное обеспечение

Для решения задач, поставленных на предприятии, использовались компьютеры со следующими характеристиками:

* процессор: Intel(R) Core (TM) i7-6700 CPU @ 3.40GHz;
* оперативная память: 32.0 Гб, 2133 MHz (неизвестный производитель);
* твердотельный накопитель: SSD SPCC M.2 PCIe 477 Гб;
* видеокарта: NVIDIA Quadro M400 8 Гб;
* сетевая карта: Intel(R) Ethernet Connection (2) I219-V;
* мониторы: 24-27” 1980\*1080p 1-3 шт на рабочее место.

Данные характеристики являются средними для всех персональных компьютеров предприятия.

Также, на предприятии присутствуют принтеры и МФУ (многофункциональное устройство) (модели неизвестны), источники бесперебойного питания (APC) и IP-телефоны (модели неизвестны).

* + 1. Программное обеспечение

Основные виды программного обеспечения, которое используют сотрудники предприятия:

* Windows 10 Pro x 64;
* Kaspersky endpoint security;
* 7-zip;
* Adobe Acrobat;
* IPS;
* PDF24;
* SAP;
* Tessa;
* Yandex browser;
* 1С (различные конфигурации);
* Visual Studio;
* Microsoft SQL Server 2016;
* Microsoft SQL Server Manager 2016;
* Microsoft Office 2013;
* Docker;
* Git Bash.
  + 1. Топология сети предприятия

Топология всей сети предприятия состоит из следующих элементов:

* Звезда;
* Кольцо;
* Последовательная.

Но поскольку предприятие является военно-направленным, схему реальной топологии получить возможности нет, была составлена примерная топология сети информационного отдела предприятия (рисунок 1Б).

Совокупность программного и технического обеспечения удовлетворяет необходимым требованиям для решения поставленной задачи, поскольку техническая составляющая соответствует требованиям программной части.

* 1. **Описание инструментальных средств разработки и сопровождения программных средств.**

Инструментальными средствами разработки модуля программы являются следующие инструменты:

* Visual Studio 2022 – интегрированная среда разработки;
* Microsoft SQL Server 2016 – локальный сервер базы данных;
* Microsoft SQL Server Manager 2016 – система управления базами данных;
* Microsoft Office 2013 – набор программ для документирования и оформления отчётности;
* Git – система контроля версий;
* GitHub – удалённое хранилище репозиториев.
* Docker – программа для локального развёртывания создаваемого модуля для проверки его совместимости с другими модулями.

Инструментальные средства предприятия удовлетворяют всем необходимым потребностям для разработки модуля программного продукта и его сопровождения.

1. **ПРАКТИЧЕСКАЯ ЧАСТЬ**
   1. **Анализ предметной области**
      1. Анализ требований и информационных потребностей

Требования к разрабатываемому модулю представлены в виде требований к базе данных и реализации API.

* + - 1. Анализ требований пользователей к базе данных

Требования пользователей к базе данных предоставлена заказчиком в виде предметной области, описанной далее.

База данных должна содержать данные пользователей, необходимые для хода и регистрации, список заявок, а также ссылки (пути) на документы, необходимые для составления заявки.

Пользователь может входить в систему и выполнять действия в соответствии со своей ролью.

Обычный пользователь может создать заявку на решение какой-либо проблемы и прикрепить к данной заявке документы, оформленные с его стороны.

Администратор может зарегистрировать пользователя и выдать ему роль, а также обработать заявки от пользователя.

Управляющий должен изменить документы из заявки пользователя (изменить запись), заполнив документ с данными со своей стороны и прикрепить его обратно к заявке.

* + - 1. Анализ требований пользователей к API

Обращение со стороны клиентского приложения (сайта, программы, мобильного приложения) к базе данных происходит с помощью API, где реализованы следующие запросы к базе данных:

* регистрация пользователя администратором;
* авторизация пользователя в системе;
* изменение данных пользователя администратором;
* предоставление данных о пользователях;
* удаление пользователя администратором;
* создание заявки обычным пользователем;
* изменение заявки обычным пользователем;
* удаление заявки;
* изменение документа из заявки пользователем с ролью руководителя;
* изменение статуса заявки на выполненную исполнителем;
* предоставление документации для составления заявки;
* предоставление информации о заявках в соответствии с фильтрацией по выполнению, подтверждению, пользователям в разных ролях (создатель, подтверждающая сторона, исполнитель), дате и подобному.

Администратор владеет всеми правами и привилегиями относительно работы с данными API.

API должна быть легко масштабируема, а её функционал задокументирован с помощью «Swagger».

* + - 1. Составление функциональных требований
         1. Сущности базы данных

Судя по информации предметной области, можно сказать, что база данных для разрабатываемой информационной системы должна состоять из следующих таблиц:

* роли (описание ролей);
* пользователи (список данных пользователей для входа в систему);
* заявки (список заявок пользователей);
* статусы (описание статусов заявки);
* документы (описание документов в системе).

Вышеуказанные таблицы в дальнейшем будут связаны с клиентской стороной системы посредством какого-либо API.

* + - * 1. Задачи информационной системы

Основными задачами системы, которые она должна выполнять, являются:

* заполнение базы данных новыми данными (новые студенты, берущие книги, новые книги и т.п.);
* изменение информации в базе данных;
* удаление данных в базе данных;
* разграничение ролей пользователей системы.
  + - * 1. Выявление групп пользователей и их задач

Основными пользователями данной информационной системы будут:

* администратор информационной системы;
* простой пользователь;
* руководитель;
* исполнитель.

Администратор ИС – уровень доступа пользователя, который имеет доступ ко всем данным и привилегиям данной БД, поскольку его задачей будет её администрирование.

Простой пользователь – уровень доступа пользователя, который может использовать данные базы данных для составления заявки на какую-то услугу со стороны исполнителя, изменять данные заявки и удалить её.

Руководитель – уровень доступа пользователя, который может изменить документ, предоставленный для заполнения обычным пользователем, тем самым изменив статус заявки на «одобрено» и просмотреть данные пользователей, которые отправляют заявки.

Исполнитель – уровень доступа пользователя, который может изменить статус заявки на «в процессе» и «выполнена».

* + 1. Выявление информационных объектов и связей между ними

В разрабатываемой информационной системе основными объектами будут таблицы содержащие данные ролях, пользователях, заявках, статусах и документах структурированные посредством связей один к одному (1:1), один ко многим (1:М) и многие ко многим (М:М).

Структура базы данных информационной системы указана в таблицах с 1-ой по 7-мую в приложении «Приложение В», а схема самой базы данных на рисунке 1Г.

Так же, для создания указанного модуля информационной системы было создано техническое задание, описанное далее.

* 1. **Проектирование программного продукта**

Полное отображение функциональной составляющей разрабатываемой информационной системы представлено виде UML-диаграмм, которые будут представлены далее.

Так же стоит отметить, что взаимодействие с разрабатываемым модулем информационной системы происходит с помощью интерфейса «клиентской стороны» (интерфейс, с которым взаимодействует пользователь), который будет реализован другими разработчиками.

* + 1. Диаграмма использования

Диаграмма использования (рисунок 26) описывает варианты использования системы центра животных «Хвостатый дом» со стороны пользователя.

На данной диаграмме пользователь, как единственный актёр системы, может оформлять записи на приём (согласовывать оплату и время записи к доктору), просматривать информацию о приюте (список питомцев и их информацию), просматривать каталог услуг, создавать профиль (регистрироваться) или входить в свою учётную запись, где можно посмотреть и отредактировать личные данные, данные питомцев и их медицинские карточки.

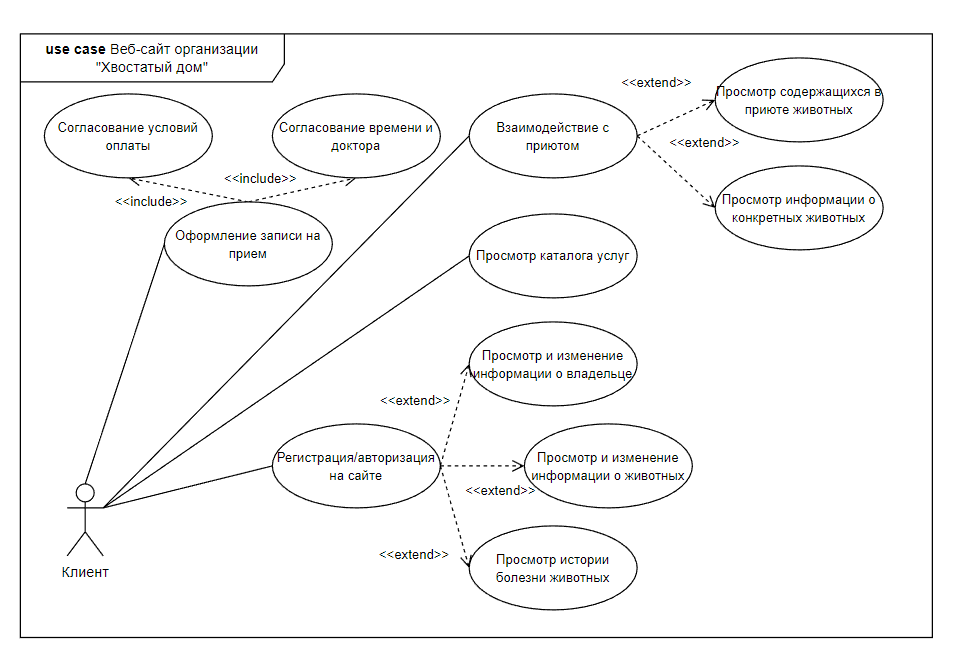


Рисунок 26. Диаграмма использования системы центра животных «Хвостатый дом» со стороны пользователя

* + 1. Диаграмма последовательности

Диаграмма последовательности (рисунок 27) отражает порядок действий, которые могут совершать пользователи по отношению к системе.

Для пользования системой пользователю необходимо авторизоваться или зарегистрироваться, что будет его первым действием, в котором система принимает и обрабатывает информацию. После входа в систему, пользователь может создать профиль своего питомца и запись его на приём к врачу, которую система внесёт в базу данных и подтвердит создание записи. Так же пользователь может просмотреть список питомцев приюта и ознакомится с их карточками.

В свою очередь, врач может редактировать созданную пользователем запись, просматривать различные данные питомцев (как имеющих хозяев, так и из приюта).

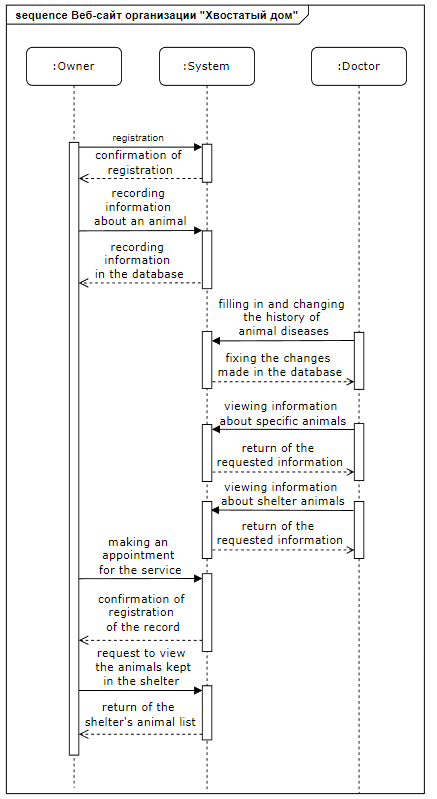


Рисунок 27. Диаграмма последовательности системы центра животных «Хвостатый дом» со стороны пользователя

* + 1. Диаграмма классов

Диаграмма классов (рисунок 28) описывает классы и их взаимосвязи в системе, что отражает структуру системы.

На данной диаграмме изображены основные сущности базы данных, с которыми пользователь может взаимодействовать с помощью интерфейса.

Данными сущностями являются:

«User», которая реализует одноимённую таблицу в базе данных (с помощью свойств), а также методы добавления, удаления, изменения своих данных и данных своих питомцев;

«Animals», которая реализует одноимённую таблицу в базе данных (с помощью свойств) и ассоциативно связана с пользователем («User»);

«Doctor», которая реализует одноимённую таблицу в базе данных (с помощью свойств) и расширяет сущность «Medicalhystory»;

«Services», которая реализует одноимённую таблицу в базе данных (с помощью свойств) и расширяет сущность «Medicalhystory»;

«Medicalhystory», которая включает в себя сущности (по полю «id») «Doctor» и «Services» и ассоциативно связана с питомцем («Animals»).

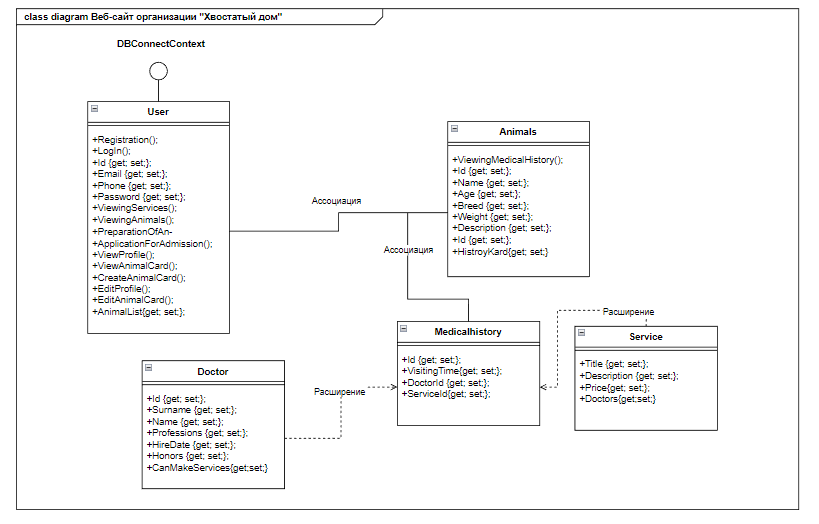


Рисунок 28. Диаграмма классов системы центра животных «Хвостатый дом» со стороны пользователя

* + 1. Диаграмма коммуникации

Диаграмма коммуникации (рисунок 29) иллюстрирует работу системы со стороны «общения» её модулей.

На данной диаграмме пользователь регистрируется или авторизуется в системе, просматривает информацию, представленную на веб-сайте, в личном кабинете добавляет своих питомцев, осуществляет онлайн запись на приём к врачу посредством упорядоченного обращения к каким-либо интерфейсам («The owner’s pet card», «Website», «Make an appointment»), которые в свою очередь так же упорядоченно «общаются» с базой данных.

Так же в системе представлена сущность «Doctor», которая «общается» с другими её частями с помощью стороннего интерфейса.

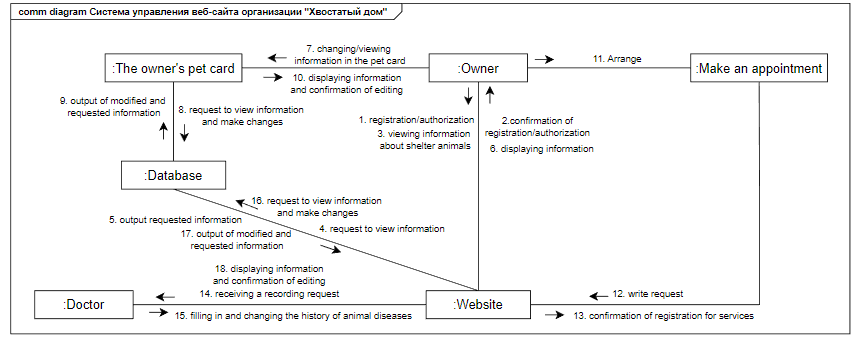


Рисунок 29. Диаграмма коммуникации системы центра животных «Хвостатый дом» со стороны пользователя

* + 1. Диаграмма деятельности

Диаграмма деятельности (рисунок 30) описывает поведение системы при запросе от пользователя на просмотр данных о питомце.

На данной диаграмме пользователь делает запрос у системы на просмотр данных о питомце. Система обрабатывает данный запрос и в зависимости от результата предлагает пользователю такие действия как завершить операцию, зарегистрировать питомца (создать профиль) или(и) возвращает данные о найденном (только что зарегистрированном) питомце.

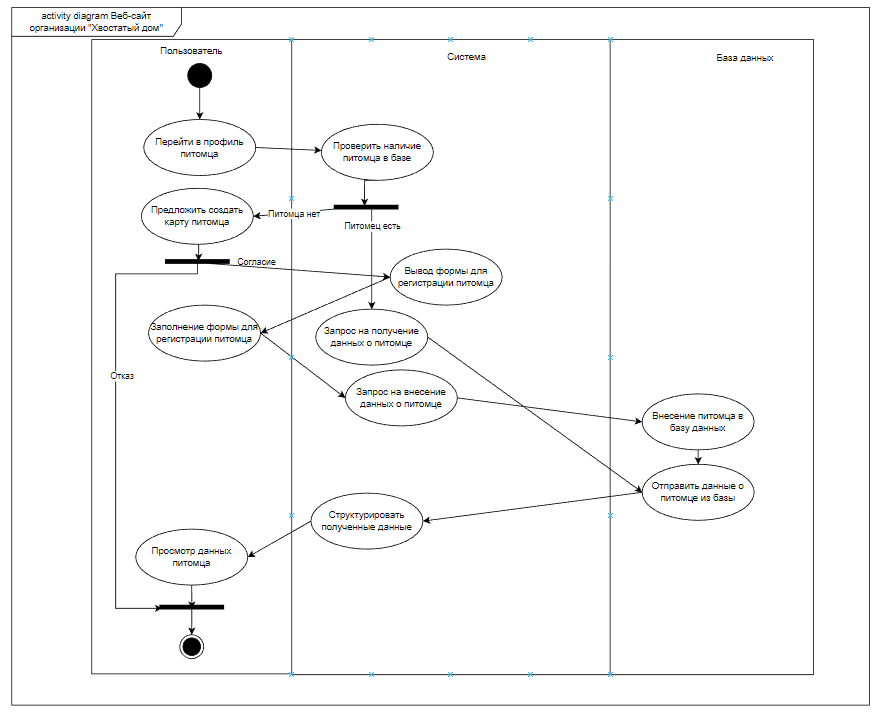


Рисунок 30. Диаграмма деятельности системы центра животных «Хвостатый дом» со стороны пользователя

* + 1. Диаграмма состояний

Диаграмма состояний (рисунок 31) отражает реализацию запроса на получение пользователем данных о питомце.

В начальном состоянии система ожидает запрос на получение данных о питомце от пользователя, а после их получения создаёт запрос к базе данных.

Если запрос моет быть обработан, то он переходит к ожидающей его базе данных, где по нему происходит выборка.

Если результат выборки вернул какие-то данные, то происходит обработка этих данных и после представления их пользователю, возврат системы в исходное состояние.

Если данных после запроса нет, то пользователю выводится сообщение о пустом результате и система возвращается в исходное состояние.

Если же запрос не может быть обработан, то пользователю выводится сообщение об ошибке, администраторам отправляется то же сообщение, а система прекращает свою работа (либо, если это предусмотрено системными администраторами, уходит в режим самовосстановления).

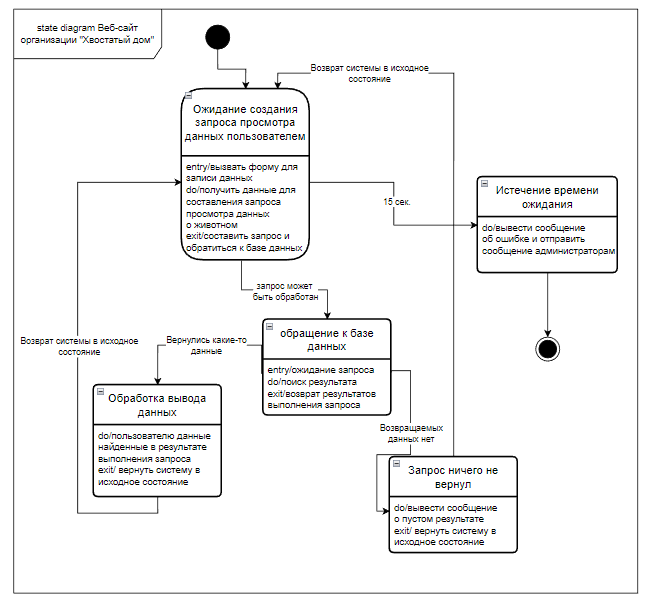


Рисунок 31. Диаграмма состояний системы центра животных «Хвостатый дом» со стороны пользователя

* + 1. Диаграмма развёртывания

Диаграмма развёртывания (рисунок 32) отражает размещение компонентов программного обеспечения на среде выполнения, а также визуализирует связи и взаимодействия между этими компонентами.

На диаграмме показано, что пользователь, через интерфейс сайта (main.html) на своём компьютере обращается посредством API к серверу (Application Server), на котором она размещена, в свою очередь которая производит необходимые манипуляции данными, получая их по протоколу TCP из базы данных (PostgreSQL Database), размещённой на собственном сервере (Database Server) на операционной системе Windows Server.

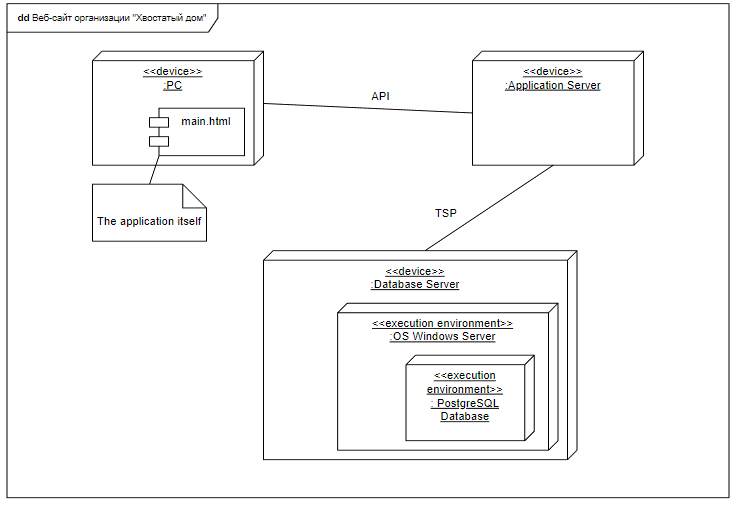


Рисунок 32. Диаграмма развёртывания системы центра животных «Хвостатый дом» со стороны пользователя

* + 1. Диаграмма компонентов

Диаграмма компонентов (рисунок 33) описывает физическую структуру и зависимостей компонентов, в роли которых может выступать исходный, бинарный и исполняемый код, в системе.

На диаграмме показана схема создания записи на приём.

Пользователь («Owner»/ «User») через порт (делегирующий соединитель) обращается к модулю авторизации, после выполнения которого (когда модуль авторизации передаёт требуемы данные основному модулю «Main.html») пользователь может с помощью различных интерфейсов (передавая данные по собираемому соединителю) обращается к модулю «Appointment.cs», который полученные данные (обработанные правильно) передаёт базе данных PostgreSQL через делегирующий соединитель.

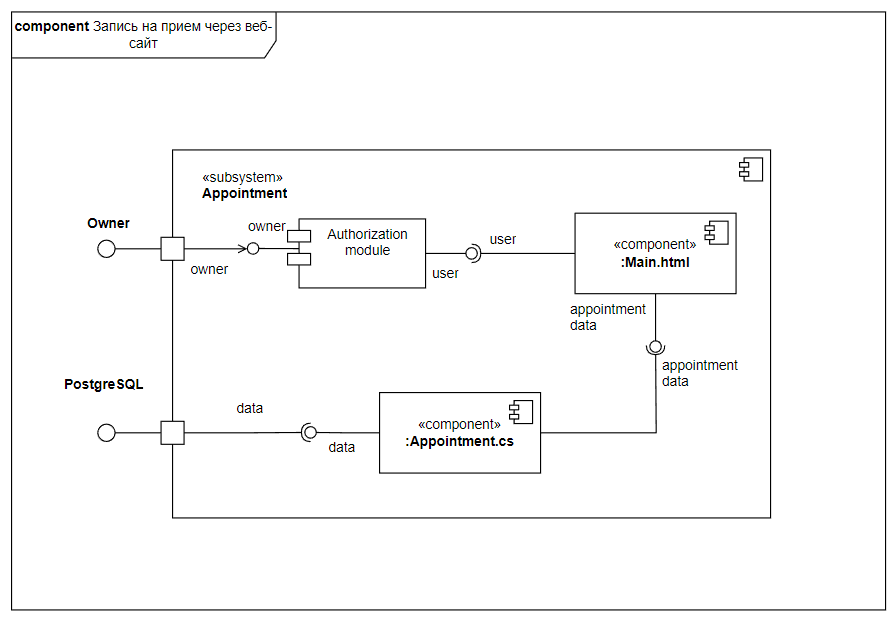


Рисунок 33. Диаграмма компонентов системы центра животных «Хвостатый дом» со стороны пользователя

* + 1. Диаграмма потоков данных

Диаграмма потоков данных (рисунок 34) иллюстрирует процесс оформления записи на приём врача, во время которого система предоставляет доступ к внешним источникам данных, потокам и хранилищам данных.

Владелец отправляет данные о своём питомце, которые обрабатываются компьютером (сервером) с сайтом «Хвостатый дом», как запись на приём к доктору. Эти данные отправляются на отдельный сервер и сохраняются до того момента, пока не понадобятся доктору, изменившему их и обратно сохранившему с помощью собственного интерфейса.

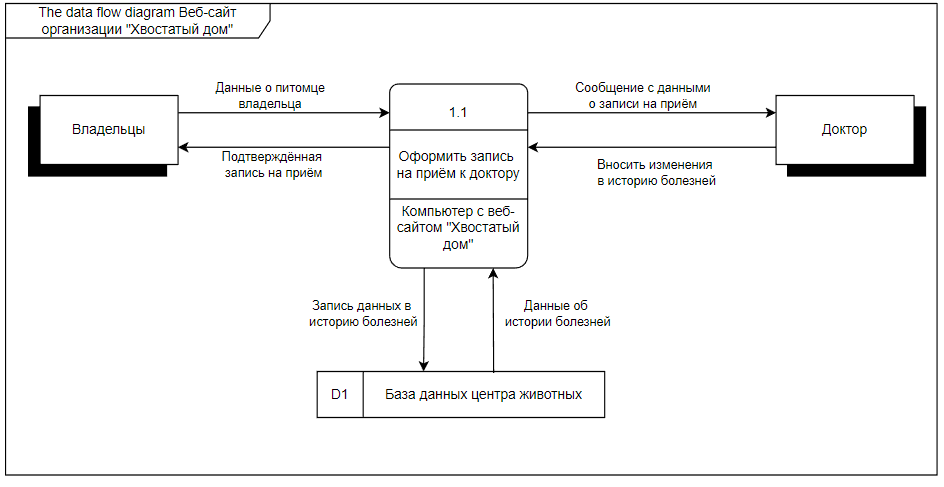


Рисунок 34. Диаграмма потоков данных системы центра животных «Хвостатый дом» со стороны пользователя

* 1. **Разработка программного модуля**
  2. **Тестирование программного продукта**

**ЗАКЛЮЧЕНИЕ**

Производственная практика по ПМ.02 «Осуществление интеграции программных модулей» проходящая в организации АО «НЗ 70-летия Победы», предоставляющая возможность применить знания, полученные в ходе обучения и учебной практики в реальном производственном процессе для решения проблемы обозначенной руководителем практики от предприятия, была успешно пройдена.

Знания, полученные в ходе обучения, были применены для создания модуля информационной системы, работающего с данными предприятия заказчика – API для системы обработки заявок на исправление каких-либо нарушений работ систем или на улучшения в предприятии от его работников, в соответствии с чем были достигнуты все поставленные цели и задачи.

Были приобретены следующие практические навыки:

* составление требований к разрабатываемому модулю и системе в целом;
* составление технического задания;
* составление описания структуры информационной системы и её взаимодействия с пользователем;
* разработка модуля информационной системы;
* составление тестовой документации и тестирование по ней разработанного модуля.

Из вышесказанного можно сделать вывод, что разработанный программный продукт удовлетворил требования заказчика.

**СПИСОК ИСПОЛЬЗОВАНОЙ ЛИТЕРАТУРЫ**

1. Галиаскаров, Э. Г.  Анализ и проектирование систем с использованием UML: учебное пособие для вузов / Э. Г. Галиаскаров, А. С. Воробьев. — Москва: Издательство Юрайт, 2024. — 125 с. — (Высшее образование). — ISBN 978-5-534-14903-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/544559 (дата обращения: 12.06.2024).
2. Черткова, Е. А.  Программная инженерия. Визуальное моделирование программных систем: учебник для среднего профессионального образования / Е. А. Черткова. — 3-е изд., испр. и доп. — Москва: Издательство Юрайт, 2024. — 146 с. — (Профессиональное образование). — ISBN 978-5-534-18094-7. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/539955 (дата обращения: 13.06.2024).
3. Советов, Б. Я.  Базы данных: учебник для вузов / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 4-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2024. — 403 с. — (Высшее образование). — ISBN 978-5-534-18479-2. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/535113 (дата обращения: 14.06.2024).
4. Маркин, А. В.  Программирование на SQL: учебное пособие для среднего профессионального образования / А. В. Маркин. — Москва: Издательство Юрайт, 2023. — 435 с. (дата обращения 15.06.2024).
5. Справочник по Transact-SQL (ядро СУБД) – URL: https://learn.microsoft.com/ru-ru/sql/t-sql/language-reference?view=sql-server-ver16 (дата обращения 15.06.2024).
6. Руководство по MS SQL Server 2022 – URL: https://metanit.com/sql/sqlserver/ (дата обращения 15.06.2024).
7. T-SQL Syntax and database concepts (Справочник по Transact-SQL) – URL: https://victorromeo.github.io/2020/01/02/TSQL-Syntax.html (дата обращения 15.06.2024).
8. Среда для разработки Microsoft Visual Studio – URL: <https://visualstudio.microsoft.com/ru/> (дата обращения 15.06.2024).
9. Документация по системе контроля версий Git – URL: https://git-scm.com/doc (дата обращения 15.06.2024).
10. Документация по языку программирования C# – URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/ (дата обращения 19.06.2024).
11. ASP.NET CORE (Создание API) – URL: https://metanit.com/sharp/aspnet6/2.11.php (дата обращения 20.06.2024).
12. Руководство по созданию веб-API с помощью ASP.NET Core – URL: https://learn.microsoft.com/ru-ru/aspnet/core/tutorials/first-web-api?view=aspnetcore-8.0&tabs=visual-studio (дата обращения 21.06.2024).
13. Документация по Git (Работа с системой контроля версий) – URL: https://devdocs.io/git/ (дата обращения 24.06.2024).
14. Документация по Unit-тестированию в языке программирования C# (Си Шарп) – URL: https://dev.to/bytehide/unit-testing-with-c-and-net-full-guide-5c7p (дата обращения 25.06.2024).
15. Документация по Unit-тестированию ASP.Net Web API 2 – URL: https://learn.microsoft.com/en-us/aspnet/web-api/overview/testing-and-debugging/unit-testing-with-aspnet-web-api (дата обращения 25.06.2024).

**ПРИЛОЖЕНИЯ**

**Приложение А**

**Структура предприятия**

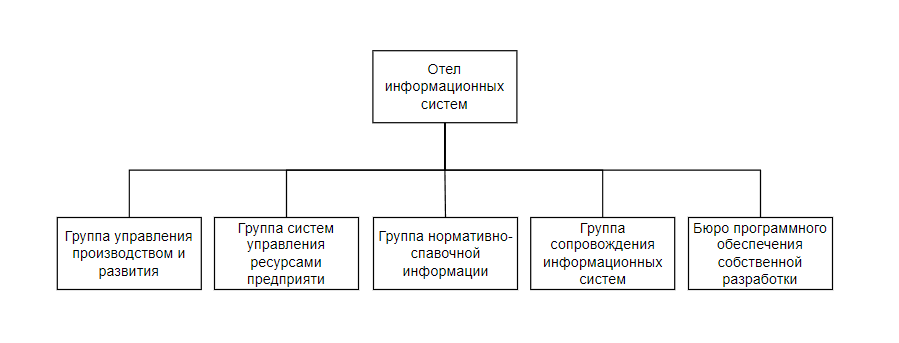


Рисунок 1. Структура информационного отдела АО «НЗ 70-летия Победы»

**Приложение Б**

**Топология сети предприятия**

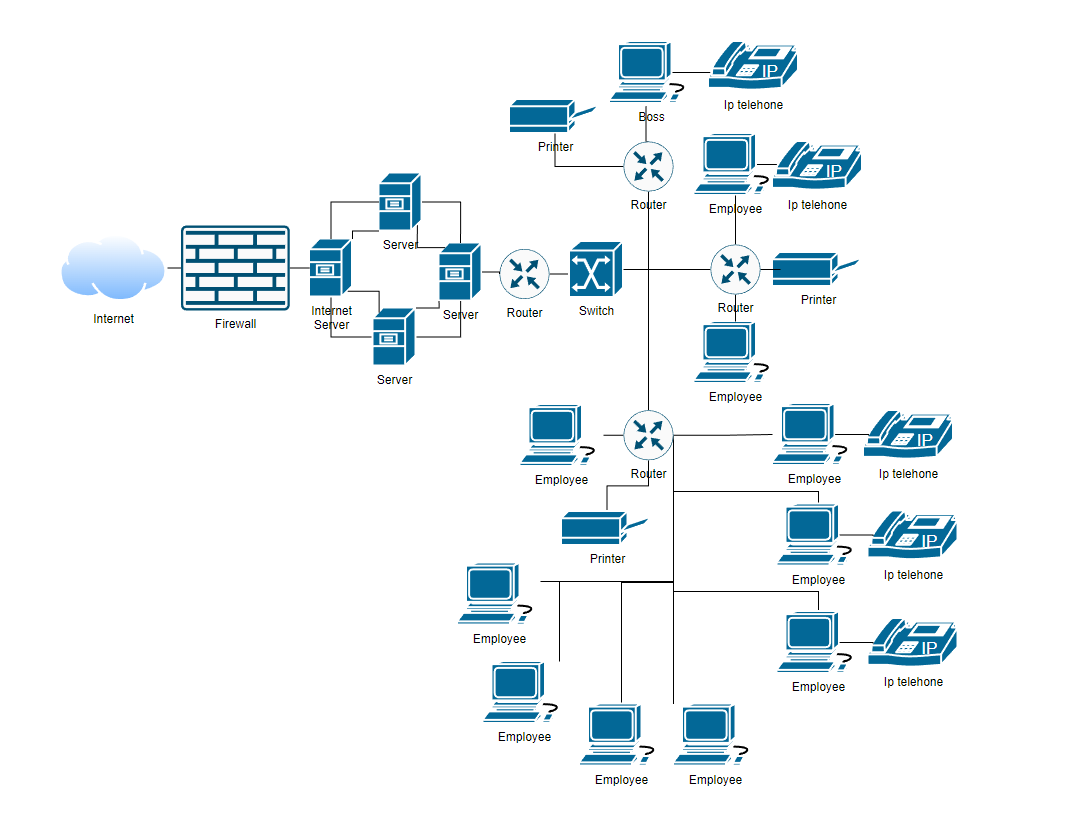


Рисунок 1. Примерная топология сети информационного отдела АО «НЗ 70-летия Победы»

**Приложение В**

**Структура базы данных информационной системы**

Таблица – 1 – Роли (Role)

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК (первичный ключ) | Код (role\_id) | Идентификация роли в виде числа |
|  | Название (tite) | Перечень названий ролей |

Таблица – 2 – Пользователи (User)

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК (первичный ключ) | Код (user\_id) | Идентификация пользователя в виде числа |
|  | Логин (login) | Логин пользователя для входа в систему |
|  | Пароль (password) | Пароль пользователя для входа в систему |
|  | Фамилия (surname) | Фамилия пользователя |
|  | Имя (name) | Имя пользователя |
| ФК (вторичный ключ) | Роль (role) | Числовой идентификатор роли, присвоенной пользователю |

Таблица – 3 – Статусы заявок (Status)

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК (первичный ключ) | Код (status\_id) | Идентификация статуса заявки в виде числа |
|  | Название (tite) | Перечень названий статусов заявок |

Таблица – 4 – Заявки (Appointment)

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК (первичный ключ) | Код (appointment\_id) | Идентификация заявки в виде уникальной хеш последовательности |
|  | Проблема (problem) | Краткое описание причины заявки пользователем |
|  | Описание проблемы (discription\_problem) | Более развёрнутое описание причины заявки |
|  | Место (place) | Фамилия пользователя |
|  | Дата создания (date\_create) | Дата, когда была создана заявка |
|  | Дата одобрения (date\_approve) | Дата, когда была одобрена заявка |
|  | Дата исправления (date\_fix) | Дата, когда были устранена причина заявки |
| ФК (вторичный ключ) | Статус (status) | Числовой идентификатор статуса заявки |

Таблица – 5 – Документы (Documents)

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК (первичный ключ) | Код (document\_id) | Идентификация документа в виде уникальной хеш последовательности |
|  | Название (tite) | Перечень названий документов |
| ФК (вторичный ключ) | Заявка (appointment) | Уникальная хеш последовательность - идентификатор заявки, к которой принадлежит документ |

Таблица – 6 – Заявки пользователя (Users\_appointment)

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ФК (вторичный ключ) | Заявка (appointment) | Уникальная хеш последовательность - идентификатор заявки, которую создал пользователь |
| ФК (вторичный ключ) | Пользователь (user) | Уникальная хеш последовательность - идентификатор пользователя, который создал заявку |

Таблица – 7 – Связи таблиц базы данных информационной системы

|  |  |
| --- | --- |
| Названия таблиц | Тип связи |
| Статусы - Заявки | 1 : М |
| Пользователи - Заявки | М : М |
| Роли - Пользователи | 1 : М |
| Заявки - Документы | 1 : М |

**Приложение Г**

**Схема базы данных для информационной системы**

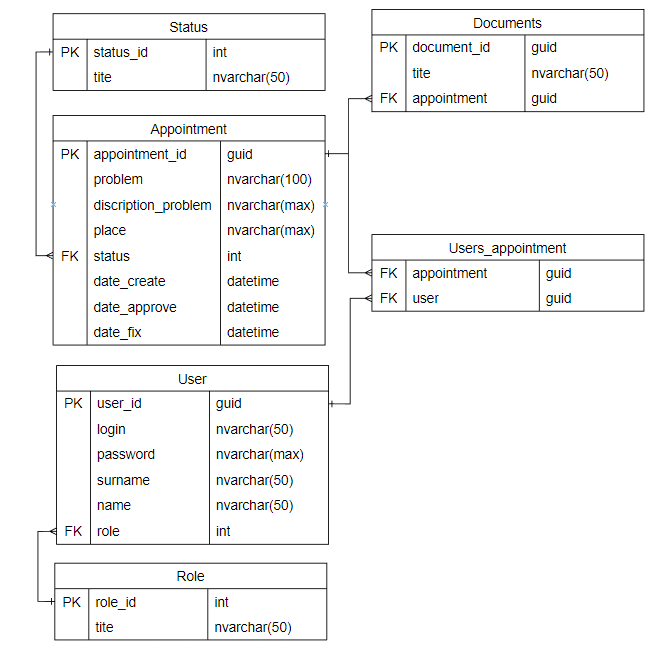


Рисунок 1. ER-диаграмма базы данных информационной системы

Таблица – 3 – Оценка возможных рисков эксплуатации

|  |  |  |  |
| --- | --- | --- | --- |
| Риск | Вероятность | Критичность | Алгоритм решения |
| Сбой алгоритма работы программы | Очень редко | Серьёзная | Перезапустить программу.  Если ошибка не будет исправлена перезапуском– переустановить программу и проверить свежеустановленную версию на работоспособность. |
| Не корректное отображение интерфейса | Редко | Маленькая | Перезапустить программу.  Если ошибка не будет исправлена перезапуском– переустановить программу и проверить свежеустановленную версию на работоспособность. |
| Поломка обработчика исключений | Очень редко | Средняя | Перезапустить программу.  Если ошибка не будет исправлена перезапуском– переустановить программу и проверить свежеустановленную версию на работоспособность. |
| Антивирус указывает на наличие в программе вирусов. | Никогда | Серьёзная | Удалить программу и обратиться в поддержку в месте приобретения программы. |