

РЕФЕРАТ

Пояснительная записка 22 с., 5 рис., 1 табл., 5 источников ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗАЦИИ ОКАЗАНИЯ МЕДИЦИНСКОЙ ПОМОЩИ

Объектом исследования является программное средство автоматизации оказания медицинской помощи.

Ключевые слова:

NodeJS, JavaScript, PostgreSQL, медицина.

Цель работы - разработка программного средства, предоставляющего возможности медицинского консультирования на расстоянии, что позволит получать более квалифицированные медицинские услуги в отдаленных точках российской федерации.

Предлагаемое программное средство позволит предоставлять услуги медицинского консультирования без необходимости непосредственного физического присутствия рядом с врачом. Это уменьшит количество очередей в больницах, а так же позволит получать квалифицированную медицинскую консультацию в отдаленных точках страны.

Проведен анализ достоинств и недостатков существующих программных продуктов. С их помощью разработаны и спроектированы функциональные требования к приложению.

На основе функциональных требований разработана архитектура программного средства и модель базы данных.

Разработаны тесты для проверки соответствия функциональным требованиям и корректности работы приложения.

Приведено технико-экономическое обоснование эффективности разработки и использования программного средства.

СОДЕРЖАНИЕ

Введение	6
1 Анализ прототипов, литературных источников формирование требо- ваний к проектируемому программному средству	7
1.1 Аналитический обзор литературы	7
1.2 Обзор аналогов программных средств	8
1.3 Требования к проектируемому программному средству	9
2 Моделирование предметной области и разработка функциональных требований	13
2.1 Описание функциональности ПС	13
2.2 Спецификация функциональных требований	14
3 Проектирование программного средства	16
3.1 Разработка программной архитектуры	16
3.2 Разработка модели базы данных	17
3.3 Разработка схемы алгоритма работы с программой	19
3.4 Разработка алгоритма входа пользователя в систему	20

ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

В настоящей пояснительной записке применяются следующие определения и сокращения.

Пациент - лицо, изъявившее желание воспользоваться услугой медицинской консультации

Врач - лицо, оказывающее консультации по средствам использования системы

Телемедицина - использование компьютерных и телекоммуникационных технологий для обмена медицинской информацией.

Программа – данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма.

AJAX - Asynchronous Javascript and XML (асинхронный JavaScript и XML)

JSON – JavaScript Object Notation

SQL – Structured Query Language (язык структурированных запросов)

URL – Uniform Resource Locator (единообразный локатор ресурса)

ВВЕДЕНИЕ

медицинская консультация - наиболее часто используемая услуга в мед. учреждениях. Зачастую пациенты, чувствуя какой либо дискомфорт, не знают к какому врачу обратиться и идут на первичную медицинскую консультацию к педиатру(если это ребенок) или к терапевту. Терапевт в свою очередь делает общий осмотр и направляет пациента к более узконаправленному специалисту. А в случае если человек живет в удаленной точке РФ, то ввиду трудодоступности медицинских учреждений люди прибегают к самолечению, либо неквалифицированным рекомендациям на форумах и сайтах.

Целью данного проекта является разработка программного средства, позволяющего пациентам получать, а врачам оказывать, медицинскую помощь на расстоянии. Разработанное ПО должно быть доступно из любого места, подключенного к сети Интернет, а также иметь возможность расширения за счет модульной архитектуры.

Подобная разработка должна обладать повышенной надежностью, защищенностью и гибкостью. История общения пациента с врачом должна быть доступна только им обоим. Работа в сети Интернет сопряжена с риском перехвата данных, особенно если работа производится в публичной локальной сети. Чтобы этого избежать, можно использовать шифрование передаваемых данных, как клиентское, так и серверное. Поддерживаемый всеми современными браузерами протокол передачи данных SSL также обеспечивает дополнительную защиту от утечки информации.

Для удобства использования интерфейс разрабатываемого программного средства должен быть понятным и легко осваиваемым. Поэтому для решения задачи можно использовать современные возможности HTML5, CSS3 и популярного JavaScript фреймворка ReactJS.

Таким образом, необходимо разработать программное средство, которое должно сочетать в себе удобство использования, надежность хранения данных и отказоустойчивость, позволяющее оказывать и получать медицинские услуги по средствам сети интернет.

1 АНАЛИЗ ПРОТОТИПОВ, ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ПРОЕКТИРУЕМОМУ ПРО- ГРАММНОМУ СРЕДСТВУ

1.1 Аналитический обзор литературы

1.1.1 Федеральный закон “Об основах охраны здоровья граждан в Российской Федерации”

29.07.2017 в федеральный закон “Об основах охраны здоровья граждан в Российской Федерации”, который регулирует отношения, возникающие в сфере охраны здоровья граждан в Российской Федерации. Были внесены изменения, а именно закон был дополнен статьей 36.2 “Особенности медицинской помощи, оказываемой с применением телемедицинских технологий”, которая описывает основные правила применения телемедицинских технологий. А именно следующие пункты:

1) Медицинская помощь с применением телемедицинских технологий организуется и оказывается в порядке, установленном уполномоченным федеральным органом исполнительной власти, а также в соответствии с порядками оказания медицинской помощи и на основе стандартов медицинской помощи.

2) Консультации пациента или его законного представителя медицинским работником с применением телемедицинских технологий осуществляются в целях:

– профилактики, сбора, анализа жалоб пациента и данных анамнеза, оценки эффективности лечебно-диагностических мероприятий, медицинского наблюдения за состоянием здоровья пациента;

– принятия решения о необходимости проведения очного приема (осмотра, консультации).

3) При проведении консультаций с применением телемедицинских технологий лечащим врачом может осуществляться коррекция ранее назначенного лечения при условии установления им диагноза и назначения лечения на очном приеме (осмотре, консультации).

4) Дистанционное наблюдение за состоянием здоровья пациента назначается лечащим врачом после очного приема (осмотра, консультации). Дистанционное наблюдение осуществляется на основании данных о пациенте, зарегистрированных с применением медицинских изделий, предназначенных для мониторинга состояния организма человека, и (или) на основании данных, внесенных в единую государственную информационную систему в сфере здравоохранения, или государственную информационную систему в сфере здравоохранения субъекта Российской Федерации, или медицинскую информационную систему, или информационные системы, указанные в части 5 статьи 91 настоящего Федерального закона.

5) Применение телемедицинских технологий при оказании медицинской помощи осуществляется с соблюдением требований, установленных законодательством Российской Федерации в области персональных данных, и соблюдением врачебной тайны.

6) В целях идентификации и аутентификации участников дистанционного взаимодействия при оказании медицинской помощи с применением телемедицинских технологий используется единая система идентификации и аутентификации.

7) Документирование информации об оказании медицинской помощи пациенту с применением телемедицинских технологий, включая внесение сведений в его медицинскую документацию, осуществляется с использованием

усиленной квалифицированной электронной подписи медицинского работника.”;

1.2 Обзор аналогов программных средств

Для создания принципиально нового решения в виде программного продукта необходимо ознакомиться с существующими аналогами в данной сфере. Анализ достоинств и недостатков этих аналогов позволит сформировать требования к проектируемому программному средству, учитывающие опыт существующих разработок и внести в них улучшения или изменения.

В качестве исследуемых аналогов были выбраны программные продукты, связанные с оказанием услуг телемедицины.

В результате поиска были обнаружены ресурсы, представленные в таблице 1.1. В ней сведены данные по найденным аналогам и их существенным признакам.

В результате поиска были выявлены схожие технические решения, используемые при разработке исследуемых программных продуктов.

Проанализируем отличия разрабатываемого программного средства от выявленных устройств.

1) Программный продукт “Яндекс.Здоровье” (пункт 1 таблица 1.1). Это кроплатформенные продукт. Имеет свои реализации на таких платформах как Android, iOS, Web. Является платным. Схожесть продуктов обусловлена сферой применения (телемедицина).

Программный продукт обладает следующими достоинствами:

- Широкий выбор специалистов
- Возможность пользоваться системой не только с персонального компьютера, но и с мобильных устройств
- Ведение истории обращения пациента

Выявленные недостатки данного продукта:

- Оказание услуг платное
- Продукт принадлежит конкретной компании и не возможно его использовать на базе какого либо медицинского учреждения

2) Программный продукт “DOC+” (пункт 2 таблица 1.1). Это кроплатформенные продукт. Имеет свои реализации на таких платформах как Android, iOS, Web. Является платным. Схожесть продуктов обусловлена сферой применения (телемедицина).

Программный продукт обладает следующими достоинствами:

- Широкий выбор специалистов
- Возможность пользоваться системой не только с персонального компьютера, но и с мобильных устройств
- Ведение истории обращения пациента
- Широкий выбор услуг

Выявленные недостатки данного продукта:

- Оказание услуг платное
- Продукт принадлежит конкретной компании и не возможно его использовать на базе какого либо медицинского учреждения

3) Программный продукт “GetDoctor” (пункт 3 таблица 1.1). Это кроплатформенные продукт. Является платным. Схожесть продуктов обусловлена сферой применения (телемедицина).

Программный продукт обладает следующими достоинствами:

- Широкий выбор специалистов
- Выявленные недостатки данного продукта:

- Продукт принадлежит конкретной компании и не возможно его использовать на базе какого либо медицинского учреждения
- Нестабильная работа системы

Таблица 1.1 — Результаты поиска аналогов программных средств

Наименование ресурса и источник	Признаки выявленных аналогов
1. Программный продукт "Яндекс.Здоровье" ссылка[1]	Функции системы: <ul style="list-style-type: none"> - консультирование по видео, аудио, в чате - выбор типа врача - консультация как у терапевта так и у узкопрофильных специалистов - ведение медицинской карты пациента
2. DOC+ (https://docplus.ru/)[2]	Функции системы: <ul style="list-style-type: none"> - Онлайн консультация - Вызов врача на дом - ЭКГ на дому - Сдача анализов на дому
3. GetDoctor (http://getdoctor.com/) [3]	Функции системы: <ul style="list-style-type: none"> - Онлайн консультация - Получение рекомендаций

В результате поиска аналогов выявлены программные средства, схожие по назначению, однако обладающие функциональными отличиями и представляемыми возможностями. Каждый из сервисов предлагает услугу телемедицины, но возможности получить коробочный продукт и развернуть его на базе любого медицинского учреждения нет. Как следствие, патентная чистота разрабатываемого программного продукта является очевидной. Следовательно, разработка программного средства автоматизации приемной кампании обоснована и целесообразна.

1.3 Требования к проектируемому программному средству

1.3.1 Назначение разработки

Функциональное назначение программного средства

Функциональным назначением программного средства является предоставление возможности оказания и получения медицинских услуг по средствам сети интернет.

Эксплуатационное назначение программного средства

Программное средство может использоваться как для получения и оказания медицинских услуг, так и для упрощения мед. работниками поиска, хранения информации о пациентах.

Конечными пользователями программного средства могут являться врачи, пользователи получающие медицинские услуги, руководители врачей для оценки работы их подопечных.

1.3.2 Состав выполняемых функций

Программное средство должно обеспечивать возможность выполнения перечисленных ниже функций:

- возможность обмена сообщениями в режиме реального времени

- функциональность разграничения доступа к различным функциям систем
- функциональность просмотра истории общения пациента с врачом

1.3.3 Требования к организации входных данных

Входные данные для программного средства должны быть представлены в виде вводимого пользователем с клавиатуры текста: идентификатор; пароль; электронная почта. После процесса аутентификации клиенту предоставляются возможности работы с системой.

Данные, вводимые пользователем, должны проверяться на корректность, как в процессе аутентификации, так и перед осуществлением каких-либо действий в системе.

1.3.4 Требования к организации выходных данных

В качестве выходных данных будут выступать веб-страницы, отображающие пользовательские данные и результаты их обработки.

1.3.5 Требования к временным характеристикам

Требования к временным характеристикам должны зависеть от количества работающих в данный момент времени человек, т.к. программное средство предоставляет совместный доступ к системе.

1.3.6 Требования к надежности

Требования к обеспечению надежного (устойчивого) функционирования программы

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

- организацией бесперебойного питания технических средств;
- выполнением требований «ГОСТ 31078-2002. Защита информации.

Испытания программных средств на наличие компьютерных вирусов»;

- необходимым уровнем квалификации сотрудников.

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 времени, необходимого на перезагрузку операционной системы и запуск программы, 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.

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 конечного пользователя без предоставления ему ограниченного доступа.

Обеспечить максимально безотказную работу программы при любых входных данных, а также при любых действиях пользователя. Программный продукт должен соответствовать требованиям безопасности, установленным ГОСТ 27451-87, ГОСТ 26104-89.

1.3.7 Требования к составу и параметрам технических и программных средств

Требования к техническим средствам

Серверная часть программного средства должна функционировать на ЭВМ со следующими минимальными характеристиками:

- процессор Xenon с тактовой частотой 2 ГГц и более;
- 1 жесткий диска объемом в 100 гб;
- оперативная память 4 Гб и более;
- сетевая карта Ethernet 1 Гбит.

Клиентская часть программного средства должна функционировать на ЭВМ со следующими минимальными характеристиками:

- процессор Intel Pentium 4 с тактовой частотой 2 ГГц и более;
- оперативная память 2 Гб и более;
- сетевая карта Ethernet 10/100 Мбит.
- операционная система: OS Windows, Linux, Mac OS X.

1.3.8 Требования к информационной и программной совместимости

Требования к информационным структурам и методам решения

Требования к информационным структурам на входе и выходе, а также к методам решения не предъявляются.

Требования к исходным кодам и языкам программирования

Исходные коды программы должны быть реализованы на языке JavaScript. В качестве интегрированной среды разработки программы должна быть использована среда Visual Studio Code.

Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены локализованной версией операционной системы Windows, Linux или Mac OS X.

Требования к защите информации и программ

В системе должен быть обеспечен надлежащий уровень защиты информации в соответствии с законом о защите персональной информации и программного комплекса в целом от несанкционированного доступа – “Информационные технологии. Средства защиты информации от несанкционированного доступа в автоматизированных системах.

Требование к пользовательскому интерфейсу

Разработать удобный и интуитивно понятный пользовательский интерфейс. Программа будет состоять из серверной части и частей врача, руководителя врача и пациента. Цвет интерфейса: светло-синий, спокойный цвет.

Требования к патентной чистоте

Проектируемое решение программного продукта должно обладать патентной чистотой.

1.3.9 Обоснование выбора языка

Язык программирования JavaScript

JavaScript – прототипно-ориентированный сценарный язык программирования. Является диалектом языка ECMAScript [4].

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование,

функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами. Языком JavaScript не владеет никакая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

Название «JavaScript» является зарегистрированным товарным знаком компании Oracle Corporation.

JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам – функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания – что придаёт языку дополнительную гибкость.

Несмотря на схожий с Си синтаксис, JavaScript по сравнению с языком Си имеет коренные отличия:

- объекты, с возможностью интроспекции;
- функции как объекты первого класса;
- автоматическое приведение типов;
- автоматическая сборка мусора;
- анонимные функции.

В языке отсутствуют такие полезные вещи, как:

- модульная система: JavaScript не предоставляет возможности управлять зависимостями и изоляцией областей видимости;
- стандартная библиотека: в частности, отсутствует интерфейс программирования приложений по работе с файловой системой, управлению потоками ввода-вывода, базовых типов для бинарных данных;
- стандартные интерфейсы к веб-серверам и базам данных;
- система управления пакетами, которая бы отслеживала зависимости и автоматически устанавливала их.

Данный язык является основополагающим в реализации данного программного продукта, реализованного в рамках этого дипломного проекта.

В качестве формата передачи данных могут использоваться фрагменты простого текста, HTML-кода, JSON или XML.

2 МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

2.1 Описание функциональности ПС

Для представления функциональной модели была выбрана диаграмма вариантов использования UML [5], которая отражает отношения между актерами и прецедентами и позволяет описать систему на концептуальном уровне. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. UML предназначен для определения, визуализации, проектирования и документирования программных систем.



Рисунок 2.1 — Диаграмма вариантов использования

Диаграмма вариантов использования разрабатываемого программного средства представлена на рисунке 2.1. На диаграмме можно выделить два основных составляющих элемента – актер и прецедент. Актер – стилизованный человек, обозначающий набор ролей пользователя, взаимодействующего с некоторой сущностью. Прецедент – эллипс с надписью, обозначающий выполняемые системой действия, приводящие к наблюдаемым актером результатам.

На основании представленной диаграммы вариантов использования можно сделать вывод, что в системе будет существовать пять основных актеров: абитуриент, оператор ЭВМ, технический секретарь приемной комиссии, сотрудник приемной комиссии, ответственный секретарь приемной комиссии. Рассмотрим каждый из прецедентов более подробно для каждого актера.

2.1.1 Пациент

Пациенту предоставляются следующие возможности:

- 1) регистрация: новый пользователь, желающий пользоваться программным средством, вводит свои данные в соответствующую регистрационную форму (уникальный идентификатор; пароль; электронную почту), после чего заполненную форму отправляет на сервер;

- 2) авторизация: пользователь, являющийся полноценным пользователем программного средства, вводит собственный уникальный идентификатор и пароль, указанный при регистрации, в форму, после чего отправляет эти данные на удаленный сервер, который проверяет их и предоставляет доступ к системе;

3) отправка сообщения: авторизовавшийся пользователь имеет возможность отправлять мгновенные сообщения врачу.

4) просмотр рекомендаций, оставленных врачом: по завершении консультации врач оставляет рекомендацию пациенту. Пациент в свою очередь имеет возможность просматривать историю оставленных ему рекомендаций.

5) просмотр профиля врача: авторизовавшийся пациент имеет возможность посмотреть информацию о враче, который проводит консультацию

6) просмотр истории сообщений: пациент имеет возможность просматривать историю общения с врачом за прошедший период времени

2.1.2 Врач

1) авторизация: врач вводит собственный уникальный идентификатор и пароль, которые получает от администратора системы, после чего отправляет эти данные на удаленный сервер, который проверяет их и предоставляет доступ к системе;

2) отправка сообщения: врач имеет возможность отправлять сообщение пациенту. Все сообщения врача и пациента будут храниться в удаленной базе данных.

3) просмотр истории сообщений: врач может посмотреть историю общения с пациента за прошлый промежуток времени.

4) просмотр профиля пациента: врач имеет возможность посмотреть профиль пациента

5) отправлять рекомендацию пациенту: для завершения консультации врачу необходимо отправить рекомендацию, в которой врач на основании диалога с пациентом предоставляет список рекомендованных действий, которые необходимо сделать пациенту чтоб справиться с проблемой

2.1.3 Администратор

1) авторизация: администратор вводит адрес своей почты и пароль, которые указывает во время настройки системы, после чего отправляет эти данные на удаленный сервер, который проверяет их и предоставляет доступ к системе;

2) поиск врачей: администратор может искать врача в системе, для просмотра его профиля

3) поиск пациентов: врач может искать пользователей в системе, для просмотра их профиля

4) просмотр профиля пациента: администратор может просматривать профиль пациентов в системе

5) просмотр профиля врача: может просматривать профили врачей в системе

2.2 Спецификация функциональных требований

На основании анализа исходных данных для проектируемого программного средства можем выделить, что основной целью является создание качественного программного продукта, позволяющего решить существующие проблемы таких как:

- общение между врачом и пациентом на расстоянии;
- хранение информации о пациентах;
- хранение информации о врачах;

В ходе разработки будут реализованы следующие возможности:

- функция общения между пациентом и врачом

- функции хранения и поиска информации о врачах
- функции хранения и поиска информации о пациентах
- просмотр статистики работы врачей
- оценка работы врачей пациентами

Проект представляет собой онлайн систему с возможностью мгновенного обмена сообщениями.

Доступ к системе будет предоставляться трем категориям пользователей.

1) Пациент. Предоставляется возможность отправки сообщений врачам, просмотр информации о врачах, просмотр рекомендаций оставленных врачом.

2) Врач. Возможность обмена мгновенными сообщениями с пациентами, отправка рекомендаций пациентам, просмотр профиля пациентов.

3) Администратор. Регистрация врачей в системе. Просмотр профилей врачей и пациентов. Поиск профилей врачей и пациентов. Просмотр статистики работы врачей.

3 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

3.1 Разработка программной архитектуры

Прежде чем приступать к непосредственной реализации программного средства, необходимо определиться с архитектурой, а также компонентов, на основе которых будет построено конечное приложение.

В первую очередь, необходимо провести анализ необходимой аппаратной конфигурации, на которой будут работать части конечного программного средства, и описать их взаимодействие между собой. Для описания узлов и их связей будем использовать диаграмму развертывания (рисунок 3.1).

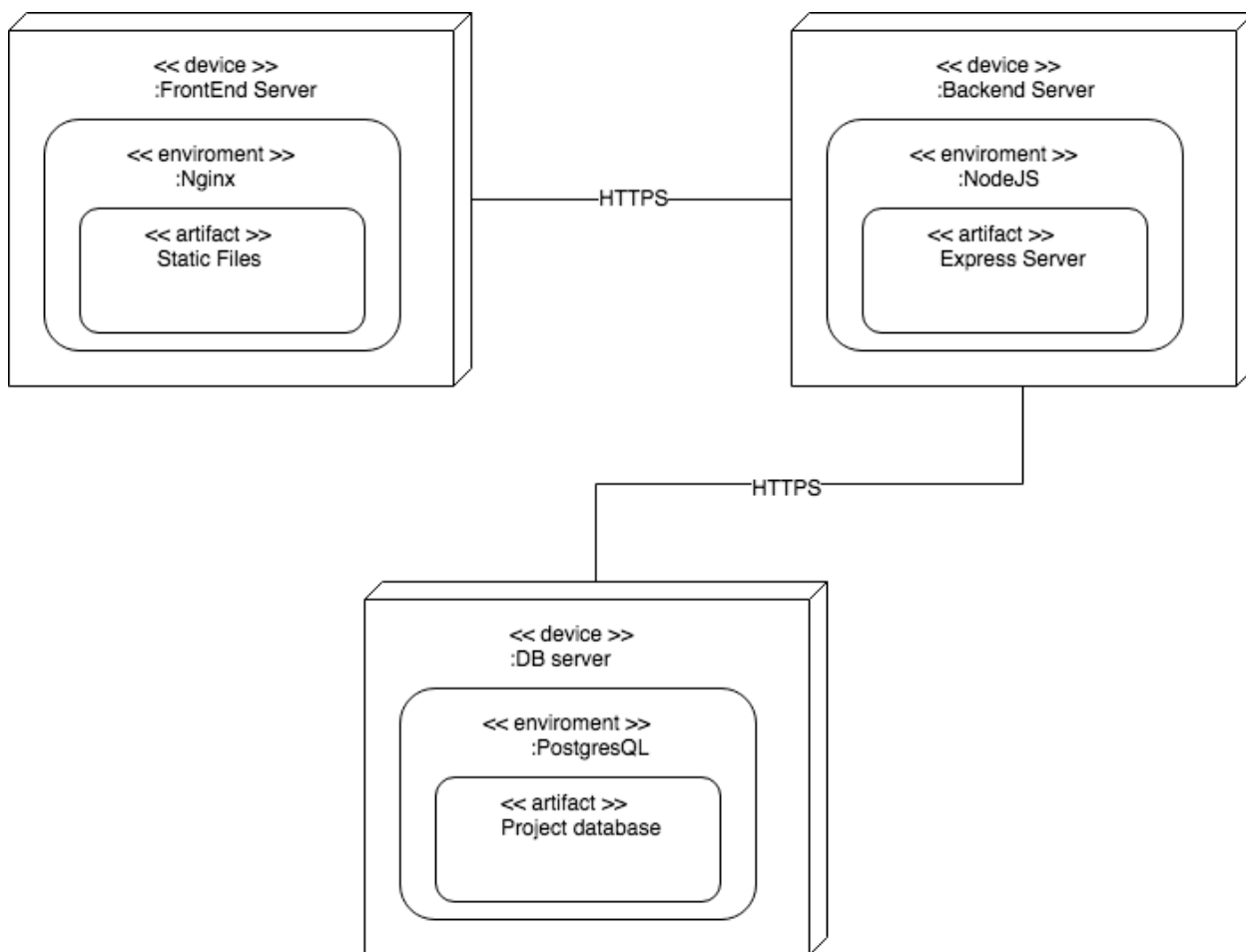


Рисунок 3.1 — Диаграмма развертывания

На основе вышеизображенной диаграммы можно сделать следующие выводы:

- 1) узлы могут располагаться в различных частях мира и взаимодействовать между собой через сеть Интернет;
- 2) сервер базы данных поддерживаются в рабочем состоянии отдельно от основного сервера;
- 3) клиент, осуществляющий работу с системой с помощью HTTPS;
- 4) HTTPS протокол применяется как клиентами, так и всевозможными серверами, с целью осуществления обмена запросами и ответами на них.

3.2 Разработка модели базы данных

Неотъемлемой частью конечного программного средства является база данных, используемая системой в процессе работы. Информационную модель предметной области можно представить на языке IDEF1X. Модель базы данных представлена на рисунке 3.2, а также на листе формата A1 (см. Графическое приложение).

TODO: переделать схему БД

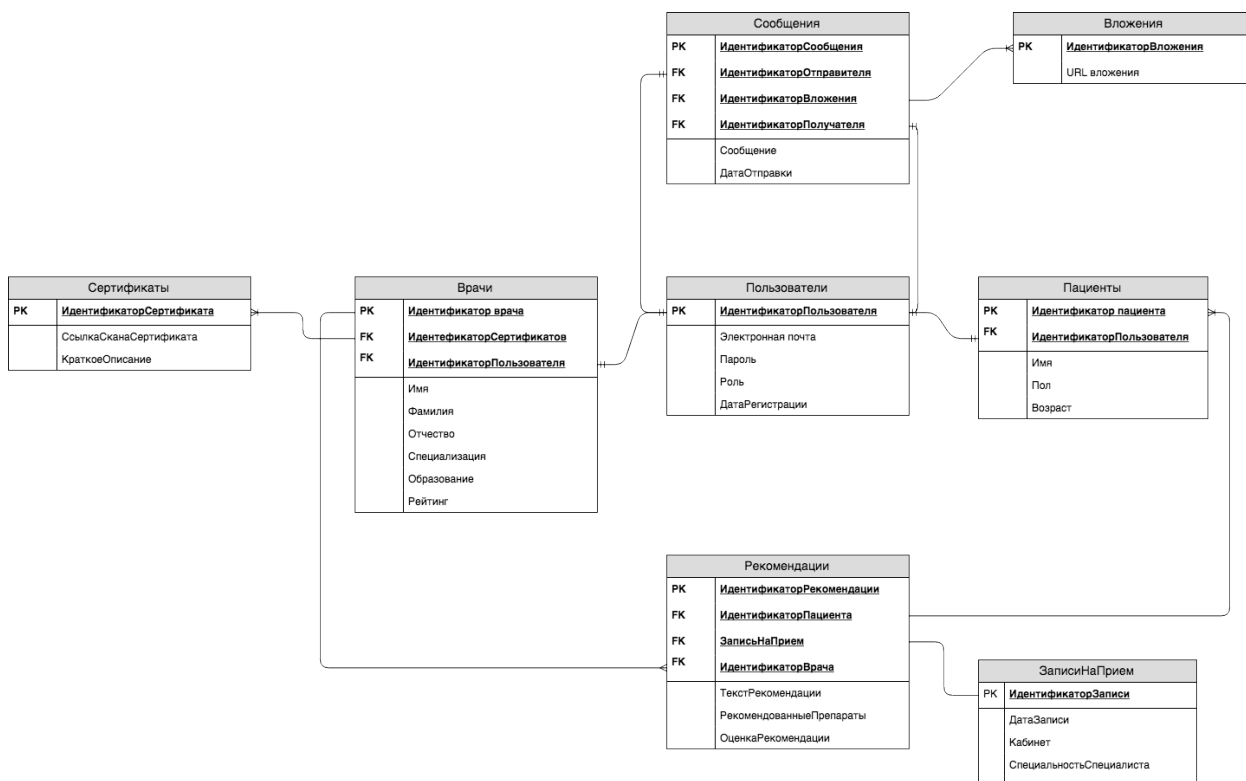


Рисунок 3.2 — Информационная модель предметной области

В модели использованы три типа связей: не идентифицирующая ноль-или-один-ко-многим, не идентифицирующая многие-ко-многим и идентифицирующая один-ко-многим. Первая обозначена штриховой линией с ромбом со стороны родительской сущности и кружком со стороны дочерней сущности, вторая – штриховой линией с ромбом со стороны родительской сущности и кружком со стороны дочерней сущности, третья – линией с кружком со стороны дочерней сущности. При наличии связи «один ко многим» одна запись в одной таблице связана с множеством записей в другой таблице. Связь между таблицами организуется на основе общего поля. На стороне «один» должно выступать ключевое поле, содержащее уникальные значения – такое поле называют внешним ключом. Значения на стороне «многие» могут повторяться.

Организация связей между таблицами обеспечивает целостность данных. Система не допустит, чтобы одноименные поля в разных таблицах имели разные значения. Ввод данных автоматически контролируется. Рассмотрим сущности по-отдельности.

3.2.1 Сущность «Пользователи»

Представляет всех пользователей в приложении со следующими полями – ИдентификаторПользователя - идентификатор пользователя (первич-

ный ключ);

- Пароль - пароль пользователя для авторизации в системе;
- Роль - роль пользователя в системе(администратор/пациент/врач);
- ДатаРегистрации - дата регистрации пользователя в системе;

Сущность имеет идентифицирующие связи «один к многим» с сущностью «Сообщения», один к одному с сущностями «Пациенты», «Врачи»

3.2.2 Сущность «Сообщения»

Представляет сообщения в системе со следующими полями:

– ИдентификаторСообщения – идентификатор сообщения (первичный ключ);

– ИдентификаторОтправителя – внешний ключ на сущность «Пользователи», представляет пользователя, отправившего сообщение;

– ИдентификаторПолучателя – внешний ключ на сущность «Пользователи», представляет пользователя, которому адресовано сообщение;

Сущность имеет связь «многие к одному» с сущностью «Пользователи», «один ко многим» с сущностью «Вложения»

3.2.3 Сущность «Вложения»

Представляет описания вложения в сообщение следующими полями:

– ИдентификаторВложения – идентификатор вложения (первичный ключ);

– ИдентификаторСообщения – внешний ключ на сущность «Сообщения»;

– URLвложения – путь до самого вложения на сервере;

Сущность имеет связь «многие к одному» с сущностью «Сообщения».

3.2.4 Сущность «Врачи»

Представляет врачей со следующими полями:

– ИдентификаторВрача – идентификатор врача (первичный ключ);

– ИдентификаторПользователя – внешний ключ на сущность «Пользователи»;

– Имя – имя врача;

– Фамилия – фамилия врача;

– Отчество – отчество врача;

– Специализация – область на которой специализируется врач;

– Образование - образовательные учреждения, законченные врачом;

– Рейтинг - средняя оценка врача на основе всех его рекомендаций;

Сущность имеет связь «один к одному» с сущностью «Пользователи», «один ко многим» с сущностями «Сертификаты», «Рекомендации».

3.2.5 Сущность «Сертификаты»

Представляет сертификаты, полученные врачом за время работы со следующими полями:

– ИдентификаторСертификата – идентификатор сертификата (первичный ключ);

– ИдентификаторВрача – внешний ключ на сущность «Врачи»;

– СсылкаСканаСертификата – ссылка на картинку со сканом сертификата;

– КраткоеОписание – описание сертификата.

Сущность имеет связь «многие к одному» с сущностью «Врачи».

3.2.6 Сущность «Пациенты»

Представляет пациентов со следующими полями:

- ИдентификаторПациента – идентификатор пациента (первичный ключ);
- ИдентификаторПользователя – внешний ключ на сущность «Пользователи»;
- Имя – имя пациента, по которому можно к нему обращаться;
- Пол – пол пациента;
- Возраст – возраст пациента.

Сущность имеет связь «один к одному» с сущностью «Пользователи» и «один к многим» с сущностью «Рекомендации».

3.2.7 Сущность «Рекомендации»

Представляет описания рекомендаций:

- ИдентификаторРекомендации – идентификатор рекомендации (первичный ключ);
- ИдентификаторПациента – внешний ключ на сущность «Пациенты»;
- ИдентификаторВрача – внешний ключ на сущность «Врачи»;
- Запись на прием – внешний ключ на сущность «ЗаписиНаПрием»;
- Рецепт - внешний ключ на сущность «Рецепты»;
- Текст рекомендации - сама рекомендация по лечению заболевания от врача;
- Оценка рекомендации - оценка рекомендации пользователем;

Сущность имеет связь «один ко многим» с сущностью «Абитуриенты». Сущность имеет связь «многие к одному» с сущностями «Врачи» и «Пациенты», «один к одному» с сущностями «ЗаписиНаПрием» и «Рецепты».

3.2.8 Сущность «ЗаписиНаПрием»

- ИдентификаторЗаписи - идентификатор записи на прием (первичный ключ);
- ДатаЗаписи - дата приема пациента
- СпециальностьСпециалиста - специальность специалиста к которому пациент записан на прием

3.2.9 Сущность «Рецепты»

- ИдентификаторРецепта - идентификатор рецепта (первичный ключ)
- ИдентификаторЛекарста - внешний ключ на сущность «Лекарства».
- ИдентификаторРекомендации - внешний ключ на сущность «Рекомендации»

Сущность имеет связь «один к одному» с сущностью «Рекомендации»; Сущность имеет связь «один к многим» с сущностью «Лекарства».

3.2.10 Сущность «Лекарства»

- ИдентификаторЛекарста - идентификатор лекарства (первичный ключ);
- НазваниеЛекарства - название лекарства;
- ОписаниеЛекарства - описание лекарства.

Сущность имеет связь «многие к одному» с сущностью «Рецепты»;

3.3 Разработка схемы алгоритма работы с программой

Схема алгоритма выполнена на листе формата А1 (см. Графическое приложение), а также на рисунке 3.5. Данная схема отображает алгоритм работы

с сайтом. Браузер пытается подключиться и загрузить страницу. Далее пользователь имеет ряд путей работы: переходы по ссылкам внутри сайта и использование его разделов. Если пользователь закрывает вкладку или переходит по внешней ссылке, то работа с сайтом закончена.

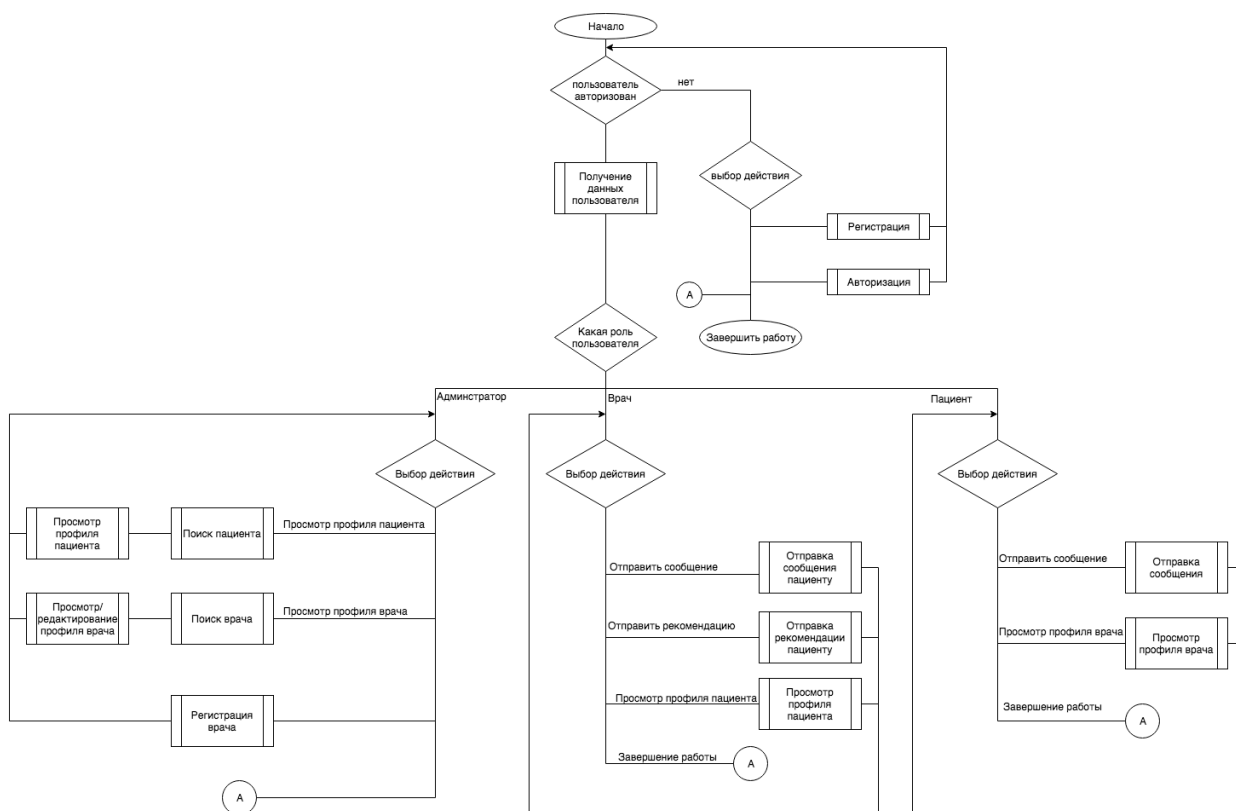


Рисунок 3.3 — Схема алгоритма работы с программой

3.4 Разработка алгоритма входа пользователя в систему

Вход пользователя в систему состоит из аутентификации и авторизации. Аутентификация представляет собой поиск соответствия пользователя названному им идентификатору. Авторизация – предоставление этому пользователю возможностей в соответствии с положенными ему правами. Алгоритм аутентификации представлен на рисунке 3.6.

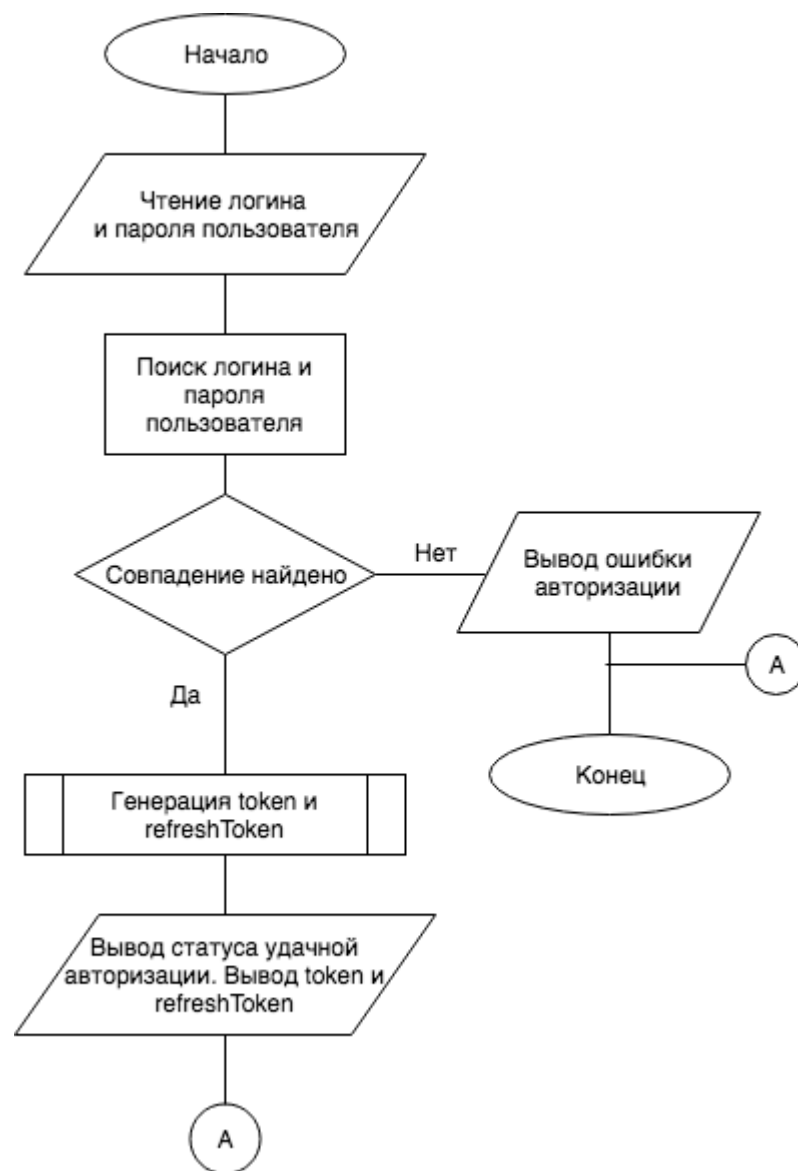


Рисунок 3.4 — Схема алгоритма аутентификации

Данный алгоритм реализует функцию аутентификации в системе. Сразу после входа на страницу аутентификации система предлагает ввести пользователю логин и пароль. В базе данных происходит поиск пользователя по его логину. В случае если пользователь с введенными логином не найден система возвращает ошибку. В случае если пользователь найден, то извлекается хешированный пароль этого пользователя из базы данных. Хешированный пароль сравнивается с хешированным паролем, который ввел пользователь. В случае их не совпадения система возвращает ошибку авторизации, иначе происходит генерация двух токенов. Основного токена, по которому будет проверяться авторизован пользователь или нет и “refresh” токена, на основании которого будет генерироваться основной токен в случае истечения срока его работы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Яндекс.Здоровье [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://health.yandex.ru/>
- [2] ДОС+ [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://docplus.ru/>
- [3] GetDoctor [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://getdoctor.com/>
- [4] Флэнаган, Д. JavaScript. Подробное руководство / Д. Флэнаган. – М.: Символ-Плюс, 2008. – 992 с.
- [5] Ларман, К. Применение UML 2.0 и шаблонов проектирования – 3-е издание / К. Ларман – М.: Вильямс, 2006. – 736 с.