

Таблица 3 – Результаты моделирования (на частоте 5,5 ГГц)

Конструкция	КНД, дБ	Ширина главного лепестка по - 3 дБ, °	УБЛ, дБ	КСВ
Шестислойная линз сплошные слои	14,85	4,23	-12,6	2
Аппроксимированная линза	14,70	9,39	-10,67	1,5

В ходе проделанной работы:

а) было исследовано несколько способов имитации структуры линзы Люнеберга. Показана работоспособность моделей таких линзовых антенн;

б) было рассмотрено два закона изменения коэффициента преломления (1) и (2), а также получены радиальные структуры линз из шести слоев для каждого из рассмотренных законов;

в) получена универсальная аппроксимирующая функция, позволившая получить плавный закон изменения эффективной диэлектрической проницаемости.

Список литературы

1. Конструкции антенн на основе линзы Люнебурга [Офиц. сайт] URL: <http://farragsat.meximas.com/homearab.html> (дата обращения: 25.05.2018)
2. Volakis J.L. Antenna engineering handbook / Volakis J.L. – McGraw Hill, 2007. – 1755 p.
3. Luneburg R.K. The mathematical theory of optics / Luneburg R.K. – Providence, RI: Brown Univ. Press., 1944. – 478 p.

СОЗДАНИЕ СИСТЕМЫ ПОИСКА МЕДИЦИНСКОЙ ИНФОРМАЦИИ В ЭЛЕКТРОННОЙ МЕДИЦИНСКОЙ КАРТЕ ДЛЯ ПОМОЩИ В ПРИНЯТИИ РЕШЕНИЯ ВРАЧОМ

Хрушков А.Е.

Уральский федеральный университет, Екатеринбург, Россия

Deagle1@mail.ru

Аннотация. Данная статья посвящена проблеме работы с электронными медицинскими картами. Так же предложен путь решения данной проблемы, путем реализации программного продукта для работы с данными и дальнейшей визуализации.

Ключевые слова: врач, API, электронная медицинская карта, JSON, автоматизация, объект.

CREATION OF MEDICAL INFORMATION SEARCH SYSTEM IN AN ELECTRONIC MEDICAL RECORD TO ASSIST A DOCTOR IN DECISION-MAKING

Khrushkov A.

Ural Federal University, Ekaterinburg, Russia

Abstract. This article is devoted to the problem of working with electronic medical records. Also, implementation of software product for working with data and its further visualization was proposed as a way to solve this problem.

Key words: doctor, API, electronic medical card, JSON, automation, entity.

В современной медицине эффективность работы персонала и оказание медицинской помощи зависит от лечебных назначений и рекомендаций врача, принимающего решения. Поэтому для обеспечения качества медицинских услуг особое внимание уделяется на проблему рациональной постановки клинического диагноза заболевания.

Не всегда врач видит всю картину целиком, поэтому для принятия решения или постановки верного диагноза, нужно наглядно посмотреть на данные, которые были собраны в ходе осмотра пациента. Будь то температурный график, количество сахара в крови или другие анализы.

Применение системы оказания информационной поддержки для оказания помощи врачам в процессе диагностики с использованием новых знаний и технологий, способствует увеличению эффективности лечения и снижению врачебных ошибок, а также оптимизации расходов на лечение, что приведет к понижению процента постановки неверных диагнозов, а это приведет к снижению заболеваемости и летальных случаев в ходе лечения.

Применение электронного документооборота в управленческой деятельности представляет собой ключевой фактор современной «офисной революции». При этом не просто осуществляется переход от традиционного бумажного к электронному документообороту, но и значительно изменяются процессы управленческого взаимодействия [1].

Существует проблема, которую не удалось решить путем внедрения информационных медицинских систем. Такой проблемой является анализ карты пациента. С введением электронной медицинской карты пациента, доктору приходится тратить больше времени на просмотр электронной медицинской карты пациента в поисках нужных анализов и осмотров, чем при использовании его бумажного эквивалента. Данную проблему пытались решить путем оптимизации интерфейса, но данная методика не показала себя эффективно.

Хоть в современной медицине и используется различное множество информационных систем, но эффективность работы врачей зависит от лечебных назначений и рекомендаций, которые предоставляет лечащий врач. Чтобы доктор чаще принимал верные решения, ему необходимо использовать системы оказания информационной поддержки, которые будут способствовать увеличению эффективности лечения.

При принятии какого-либо решения врачом ему приходится просмотреть множество записей, анализов и рекомендации от других врачей, у которых наблюдался данный пациент, что занимает большое количество времени. Для упрощения процесса принятия решений врачом, было принято решение создать приложения, которое бы упрощало все вышесказанные процессы.

Выбор платформы разработки

Для реализации данного программного продукта были рассмотрены следующие варианты:

1. Написание Desktop-приложения;
2. Написание Сайта;
3. Написание расширения для Google Chrome.

Ни для кого не секрет, что большинство врачей являются люди в возрасте, которые умеют выполнять только минимальный функционал работы с компьютером. Поэтому данный программный продукт должен был представлять из себя что-то простое в использовании, иметь интуитивно понятный интерфейс и этому можно было легко обучаться.

Архитектура веб-приложения включает в себя множество особенностей. Основные из них, на которых стоит акцентировать внимание – это схема разделения данных и выбор типа взаимодействия сервера и клиента. Все они важны, и играют определенную роль в работе веб-приложения [2].

В моем случае, моделью данных является база данных, связь с которой я поддерживаю через сервер, с которым поддерживается связь через API. Контроллером является сам сервер, а view есть изображение данных на экране.

Принцип работы программы «Помощник врачу»

Работа приложения выглядит следующим образом. Сначала пользователь выделяет необходимый параметр на веб-странице. Затем данный параметр обрабатывается клиентом и формирует запрос на основе API, который предоставляет сервер. После этого запрос уходит на сервер, сервер проверяет уровень доступа и решает предоставить информацию или нет, а также проверяет наличие этого параметра в БД. Если условия проверки были соблюдены, то из базы берутся необходимые данные, которые обрабатываются сервером и там принимают вид JSON объекта и возвращаются ответом на запрос от клиента. Далее JSON обрабатывается клиентом и принимает нужный для отображения вид. Далее на основании этих данных, формируются различные графики, которые пользователь хотел отобразить.

JavaScript - мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript [3].

HTML - стандартизированный язык разметки документов во Всемирной паутине. Теговый язык разметки документов. Любой документ на языке HTML представляет собой набор элементов, причём начало и конец каждого элемента обозначается специальными пометками — тегами. Элементы могут быть пустыми, то есть не содержащими никакого текста и других данных [4].

Так как функционал приложения подразумевает собой реализацию визуализации данных, а отрисовка линейных диаграмм, гистограмм и круговых диаграмм реализуется путем использования библиотеки ECharts. Чтобы визуализация корректно работала, нужно преобразовать данные в нужный для библиотеки формат. Для корректной обработки данных библиотекой, данные должны подаваться в виде массива. Ниже представлена реализации построение гистограммы, в поле «data» подается массив «value» с данными, которые нужно визуализировать.

Чтобы превратить полученный JSON объект в данные пригодные для обработки, в моем случае это массив, нужно реализовать функцию парсинга данных. Парсинг осуществляется путем разложения JSON объекта по полям и значениям, и дальнейшее занесение данных в переменные.

Программный продукт, который был реализован в ходе выполнения научно-исследовательской работы был отправлен на тестирование в одно из медицинских учреждений.

В ходе работы ставились следующие задачи:

- Анализ средств веб-разработки;
- Проектирование архитектуры веб-сайта:

- Создание макета и его верстка для главной страницы проектов;
- Реализовано получение данных с веб-страницы;
- Реализована отправка данных на сервер с помощью API;
- Реализована визуализация данных;

В итоге поставленный комплекс задач выполнен в полном объеме.

Список литературы

1. Развитие принципов документооборота при переходе от бумажного к электронному взаимодействию [Электронный ресурс] // www.top-personal.ru
URL: <http://www.top-personal.ru/officeworkissue.html?224> (дата обращения: 01.06.2018).
2. Принципы работы и структура Web-приложений на основе ASP.NET [Электронный ресурс] // www.intuit.ru URL: <http://www.intuit.ru/studies/courses/1139/250/lecture/6422> (дата обращения: 06.06.2018).4
3. Alexei White. Major JavaScript Engines // JavaScript Programmer's Reference. — Indianapolis, IN 46256: Wiley Publishing, Inc., 2009. — P. 12 — 13. — (Programmer's Reference).
4. Э. Фримен, Э. Фримен. Изучаем HTML, XHTML и CSS = Head First HTML with CSS & XHTML. — П.: «Питер», 2010. — 656 с.

РАЗРАБОТКА ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ ИССЛЕДОВАНИЙ ПРЕДПРИНИМАТЕЛЬСКИХ СЕТЕЙ: СИСТЕМНО-ДИНАМИЧЕСКАЯ МОДЕЛЬ И ЛОКАЛЬНАЯ ПЛАТЕЖНАЯ СИСТЕМА

Апанасенко А.В., Овчинникова Т.А., Медведева М.А., Чепуров Е.Г

Уральский федеральный университет, Екатеринбург, Россия
stacy-chan@yandex.ru ovchinnicova_t@mail.ru marmed55@yandex.ru
chepurov.evgeny@gmail.com

Аннотация. В статье описан комплекс приложений, позволяющий исследовать деятельность предпринимательской сети, состоящий из двух частей: высокофункционального прототипа приложения для оценки экономической эффективности функционирования предпринимательской сети и веб-