**РЕФЕРАТ**

ВКР бакалавра «Система для онлайн-консультирования с врачами. Разработка клиентской части».

Пояснительная записка: 62 стр., 32 рис., 6 табл., 28 источн., 4 прилож.

Ключевые слова: онлайн-консультации с врачами, web-приложение, JavaScript, информационная система (далее ИС), программный продукт (далее ПП).

Объектом исследования в данной работе являются системы для онлайн-консультирования с врачами.

Цель работы – разработка клиентской части web-приложения для поиска врача и проведения онлайн-консультации.

Web-приложение предназначено для пациентов желающих получить медицинскую консультацию, а так же для врачей, осуществляющих эту консультацию за оплату.

Для пациента приложение позволяет удобно выполнять поиск врача по заданным фильтрам, записываться на консультацию с выбором удобного для пациента времени, вести медицинскую карту, хранить анализы и снимки. Для врача сервис предоставляет выбирать удобный рабочий график и проводить онлайн-консультации за назначенную врачом оплату.

В системе предусмотрены мероприятия защиты персональных данных и разграничения доступа. Приложение имеет интуитивно понятный дружественный интерфейс.

Пояснительная записка состоит из введения, 4 разделов и заключения, в которых проводится анализ информационных процессов в задаче, а именно, исследование истории вопроса и его состояния на сегодняшний день, рассмотрены существующие на сегодняшний день аналоги, а также выполнена разработка и тестирование получившегося программного модуля.

**СОДЕРЖАНИЕ**

СОДЕРЖАНИЕ 5

ВВЕДЕНИЕ 6

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ, СУЩЕСТВУЮЩИХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, СИСТЕМ ИЛИ МЕТОДОВ И АЛГОРИТМОВ, КОТОРЫЕ РЕШАЮТ АНАЛОГИЧНЫЕ ЗАДАЧИ 7

1.1 Анализ предметной области и её информационные характеристики 7

1.2 Обзор существующих аналогов 7

1.3 Обоснование выбора инструментальных средств 7

1.4 Постановка задачи 7

Выводы по разделу 1 7

2 СИСТЕМОТЕХНИЧЕСКИЙ АНАЛИЗ И ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «КЛИЕНТСКАЯ ЧАСТЬ ДЛЯ СИСТЕМЫ ОНЛАЙН-КОНСУЛЬТИРОВАНИЯ С ВРАЧАМИ» 14

2.1 Построение диаграмм потоков данных (DFD) в проектируемой системе для описания процессов документооборота и обработки информации 14

2.2 Разработка функциональных и информационных моделей IDEF0 проектируемой системы 14

2.3 Разработка структуры данных 14

2.4 Разработка математической (имитационной) модели 14

Выводы по разделу 2 14

3 РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ «КЛИЕНТСКАЯ ЧАСТЬ ДЛЯ СИСТЕМЫ ОНЛАЙН-КОНСУЛЬТИРОВАНИЯ С ВРАЧАМИ» 14

3.1 Разработка компонентов программного модуля 14

3.2 Разработка интерфейса программного модуля 14

Выводы по разделу 3 14

4 ТЕСТИРОВАНИЕ РАЗРАБОТАННОГО ПРОГРАММНОГО МОДУЛЯ «КЛИЕНТСКАЯ ЧАСТЬ ДЛЯ СИСТЕМЫ ОНЛАЙН-КОНСУЛЬТИРОВАНИЯ С ВРАЧАМИ» 20

4.1 Сценарии тестирования программного модуля 20

Выводы по разделу 4 20

ЗАКЛЮЧЕНИЕ 69

**ВВЕДЕНИЕ**

**Актуальность темы**. В связи с нынешней эпидемиологической ситуацией в мире остро стоит проблема записи на прием к врачу и получения как минимум первичной медицинской консультации, по причине того, что государственные больницы не справляются с большим количеством пациентов. Кроме того, посещение поликлиник увеличивает количество потенциально опасных контактов. Так же часто встречаются ситуации, связанные с ограничениями по месту жительства и отсутствия возможности своевременно получить консультацию специалиста.

Существуем большое количество готовых программных продуктов, однако среди них отсутствуют системы где врачи имеют возможность зарегистрироваться, пройти верификацию и предоставлять услуги консультирования пациентам.

Целью выпускной квалификационной работы является разработка web-приложения для онлайн-консультирования с врачами. В процессе достижения поставленной цели решались следующие задачи:

1) проанализировать функциональные обязанности врача, пациента и типовых вариантов реализации web-приложений;

2) разработать алгоритмы функционирования приложения, выбор платформы и языка программирования для создания web-приложения;

3) разработать структурную схему приложения, дизайн и реализовать его в виде web-приложения;

4) протестировать web-приложение.

**Объектом исследования** являются системы для онлайн-консультирования с врачами.

**Предметом исследования** настоящей работы является web-приложение для проведения онлайн-консультаций.

**Практическое значение работы.** Результаты данной работы могут представлять интерес для лечебно-профилактических учреждений общего профиля и специализированных клиник.

**Научная и практическая новизна.** Разработано оригинальное web-приложение, отличающееся широкой функциональностью, информационной безопасностью и простотой использования.

**Структура работы.** Данная работа состоит из пояснительной записки, включающей в себя введение, четыре раздела, выводы, список использованных источников и приложения.

Во введении приведено обоснование актуальности решаемой задачи и сформирована цель и задачи работы.

В первом разделе выполнен анализ информационных процессов в задаче разработки web-приложения для онлайн-консультирования с врачами. Проведен сравнительный анализ существующих web-приложений, в которых решаются схожие задачи. Выполнена постановка задачи на разработку web-приложения.

Во втором разделе разработаны алгоритмы функционирования приложения, выбрана платформа и язык программирования для создания web-приложения.

В третьем разделе разработана структурная схема приложения, пользовательский интерфейс, реализованы модули web-приложения.

В четвертом разделе выбрана методология тестирования и протестировано web-приложение. В заключении сделаны выводы во всей работе.

**1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ, СУЩЕСТВУЮЩИХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, СИСТЕМ ИЛИ МЕТОДОВ И АЛГОРИТМОВ, КОТОРЫЕ РЕШАЮТ АНАЛОГИЧНЫЕ ЗАДАЧИ**

**1.1 Анализ предметной области и её информационные характеристики**

**1.2 Обзор существующих аналогов**

**1.3 Обоснование выбора инструментальных средств**

В ходе работы на стороне клиента было решено использовать язык TypeScript и библиотеку для разработки пользовательских интерфейсов React. Для управления состоянием приложения была выбрана библиотека MobX. В качестве CSS препроцессора был взят SASS. Для организации соединения в режиме реального времени была использована библиотека Socket.IO.

TypeScript – это язык программирования, представленный Microsoft в 2012 году и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript. TypeScript отличается от JavaScript возможностью явного статического назначения типов, поддержкой использования полноценных классов, а также поддержкой подключения модулей, что призвано повысить скорость разработки, облегчить читаемость, рефакторинг и повторное использование кода, помочь осуществлять поиск ошибок на этапе разработки и компиляции, и, возможно, ускорить выполнение программ.

React – JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. React может использоваться для разработки одностраничных и мобильных приложений. Его цель — предоставить высокую скорость, простоту и масштабируемость. Из особенностей можно выделить следующее:

* Virtual DOM - легковесная копия DOM дерева, в которую вносятся изменения, после чего происходит сравнение DOM дерева с его виртуальной копией, определяется разница и происходит перерисовка того, что было изменено.
* JSX – расширение синтаксиса JavaScript, которое позволяет использовать HTML синтаксис для описания структуры интерфейса.
* Методы жизненного цикла, при помощи которых разработчик может описывать поведение компонента на каждом этапе его жизни (при монтировании, обновлении данных, либо его удалении).

MobX – это автономная библиотека, для управления фронтенд-состоянием приложения. MobX обеспечивает консистентность и согласованность внутреннего состояния фронтенд-приложения, предоставляя удобные инструменты для его изменения.

MobX имеет следующие преимущества по сравнению с его аналогом Redux:

* Эффективен сразу после установки, отсутствие “Многословности” по сравнению с Redux;
* Объектно-ориентированных подход;
* Упрощение работы с асинхронными действиями;
* Производительность и скорость разработки.

Sass (Syntactically Awesome Stylesheets) — это метаязык на основе CSS, предназначенный для увеличения уровня абстракции CSS-кода и упрощения файлов каскадных таблиц стилей.

Язык Sass имеет два синтаксиса:

* sass — отличается отсутствием фигурных скобок, в нём вложенные элементы реализованы с помощью отступов;
* scss (Sassy CSS) — использует фигурные скобки, как и сам CSS.

Socket.IO — это библиотека JavaScript для веб-приложений реального времени. Он обеспечивает двустороннюю связь в реальном времени между веб-клиентами и серверами. Он состоит из двух частей: клиентской библиотеки, которая запускается в браузере, и серверной библиотеки для node.js. Оба компонента имеют идентичный API.

Написание приложения для реального времени с использованием популярных стеков веб-приложений, таких как LAMP (PHP), традиционно было очень трудным. Он включает в себя опрос сервера на наличие изменений, отслеживание временных меток, и это намного медленнее, чем должно быть.

Сокеты традиционно были решением, вокруг которого строится большинство систем реального времени, обеспечивая двунаправленный канал связи между клиентом и сервером. Это означает, что сервер может отправлять сообщения клиентам. Всякий раз, когда происходит событие, идея заключается в том, что сервер получит его и отправит заинтересованным подключенным клиентам.

Socket.IO довольно популярен, его используют Microsoft Office, Yammer, Zendesk, Trello и многие другие организации для создания надежных систем реального времени. Это одна из самых мощных JavaScript-фреймворков на GitHub и наиболее зависимая от модуля NPM (Node Package Manager). Socket.IO также имеет огромное сообщество, что означает, что найти помощь довольно легко.

**2 СИСТЕМОТЕХНИЧЕСКИЙ АНАЛИЗ И ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «КЛИЕНТСКАЯ ЧАСТЬ ДЛЯ СИСТЕМЫ ОНЛАЙН-КОНСУЛЬТИРОВАНИЯ С ВРАЧАМИ»**

**2.1 Построение диаграмм потоков данных (DFD) в проектируемой системе для описания процессов документооборота и обработки информации**

В системе онлайн консультаций с врачами основным процессом является регистрация врачей/пациентов, поиск врачей, запись на консультацию и ее последующее проведение. Для своей работы система использует внешние сущности: врач и пациент.

Пациент определяет свою проблему сам находит врача с помощью фильтров или поиска, далее оплачивает услугу и записывается на консультацию. После прохождения консультации пациент получает результат в виде рекомендации по лечению.

При регистрации врача происходит проверка подлинности документов о квалификации врача модератором, если ответ положительный, то врач может перейти к процессу консультаций пациентов, которые записались к нему на прием.

Основной процесс включает в себя следующие процессы: регистрация, запись на консультацию, поиск врача пациентом, консультация, составление отзыва после консультации, перевод денег врачу.

На рисунках 2.1-2.4 представлены DFD-диаграммы полученные в ходе проектирования системы.

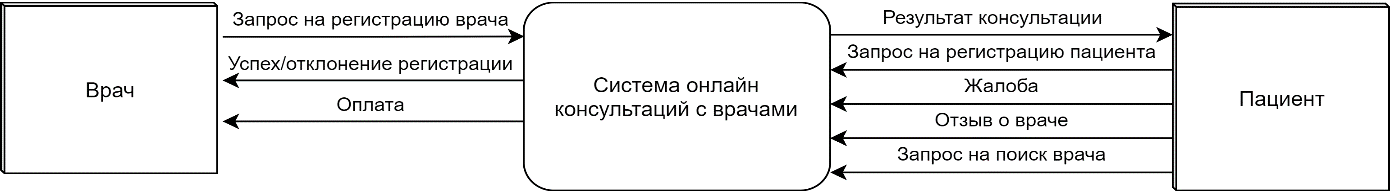
****

Рисунок 2.1 – DFD-диаграмма основного процесса

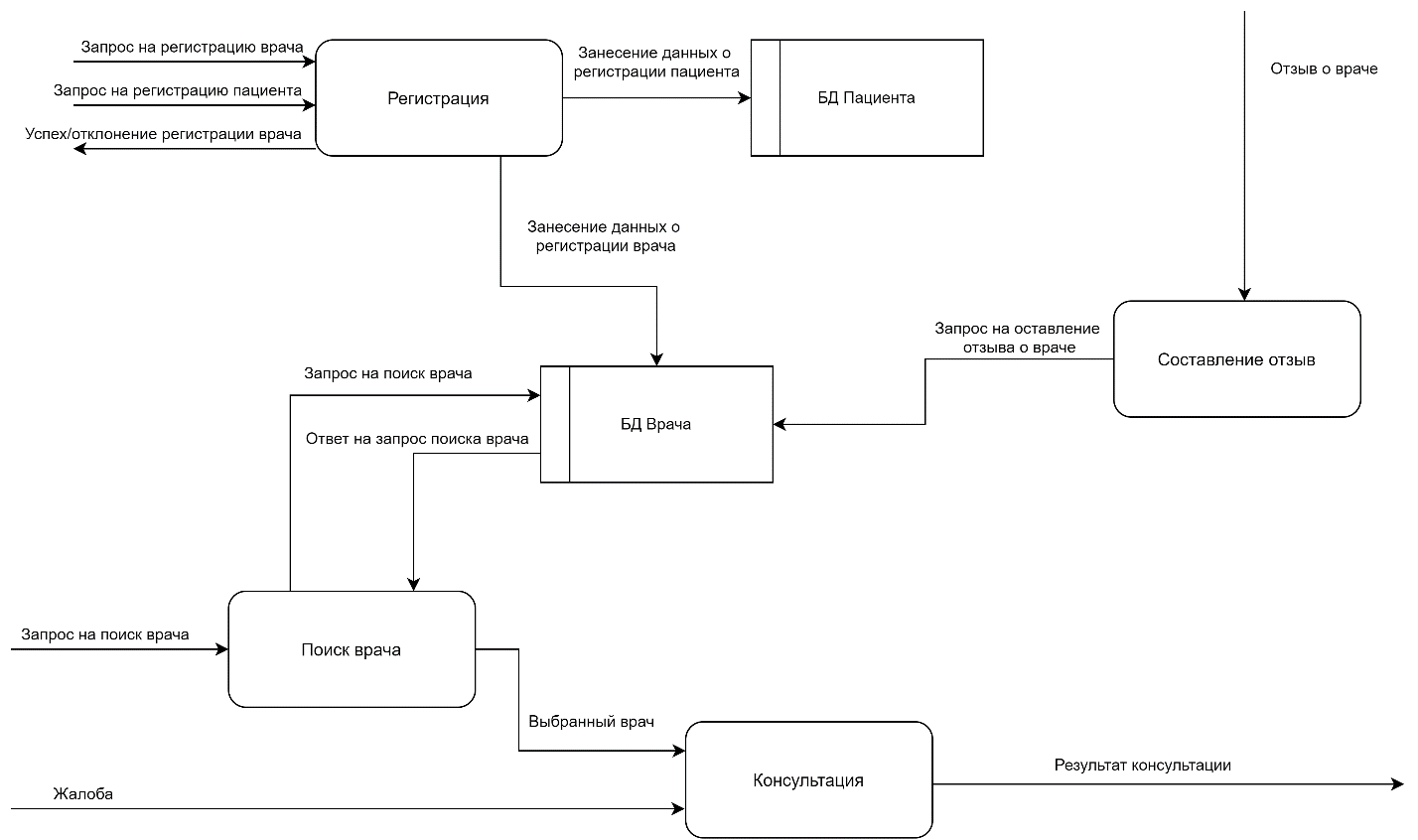
****

Рисунок 2.2 – DFD-диаграмма декомпозиции основного процесса

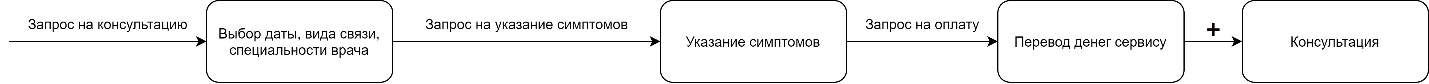
****

Рисунок 2.3 – DFD-диаграмма декомпозиции процесса консультации

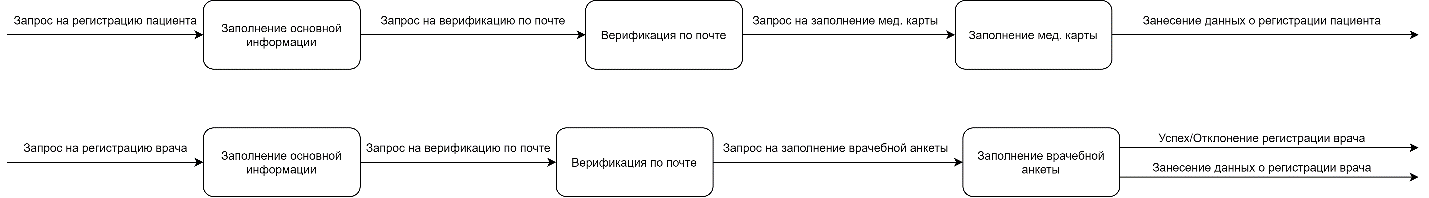
****

Рисунок 2.4 – DFD-диаграмма декомпозиции процесса регистрации

**2.2 Разработка функциональных и информационных моделей IDEF0-IDEF1 проектируемой системы**

Таблица 2.1 – Процессы диаграммы A1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Шифр | Название процесса | Входные данные | Управляющие данные | Механизм | Результат процесса |
| A1 | Предоставить услуги консультации | Запрос на регистрацию врача и пациента, | Прайс лист, установленное время, форма регистрации и тестирования | Врач, пациент, модератор | Результат консультации, оплата, отклонение оплаты, отклонение регистрации врача |

Таблица 2.2 – Декомпозиция процесса A1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Шифр | Название процесса | Входные данные | Управляющие данные | Механизм | Результат процесса |
| A11 | Регистрация врача | Запрос на регистрацию врача | Форма регистрации | Врач, модератор | Отклонение регистрации врача |
| A12 | Регистрация пациента | Запрос на регистрацию пациента | Форма регистрации | Пациент, модератор | Жалоба |
| A13 | Поиск врача | Жалоба | Форма тестирования | Пациент, врач | Выбранный врач |
| A14 | Оплачивать | Выбранный врач | Прайс лист | Пациент | Чек об оплате |
| A15 | Консультация | Чек об оплате | Установленное время | Врач, пациент | Результат консультации |
| A16 | Составить отзыв | Результат консультации | Результат консультации | Пациент | Отклонение оплаты |
| A17 | Перевод врачу | Положительный отзыв | Положительны отзыв | Модератор | Оплата |

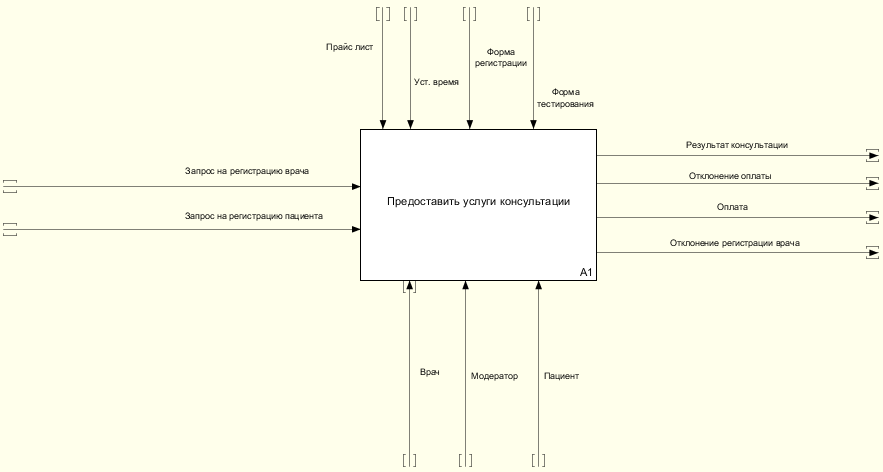


Рисунок 2.5 – IDEF0-диаграмма основного процесса

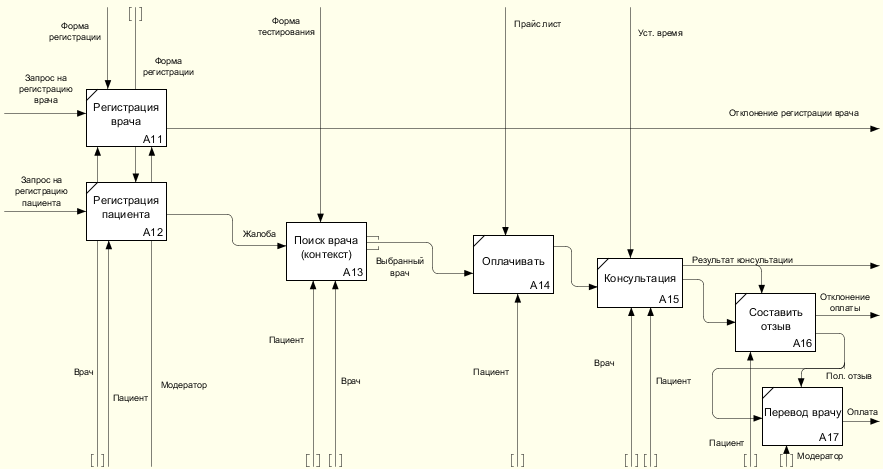


Рисунок 2.6 – IDEF0-диаграмма декомпозиции основного процесса

**2.3 Разработка структуры данных**

Написать про сторы