

**«Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И.Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)**

| | |
|--------------------|---|
| Направление | 01.03.02 - Прикладная математика и информатика |
| Профиль | Математическое обеспечение программно-информационных систем |
| Факультет | Факультет компьютерных технологий и информатики |
| Кафедра | МОЭВМ |

К защите допустить

И.о. зав. кафедрой

А.А. Лисс

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
БАКАЛАВРА**

Тема: МОДЕРНИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ ПРОВЕДЕНИЯ ПРИЕМА В МЕДИЦИНСКОЙ ОРГАНИЗАЦИИ

| | | |
|-----------------------|----------------|-------------------|
| Студент | <hr/> | К.И. Пращутинский |
| | <i>подпись</i> | |
| Руководитель | <hr/> | А.А. Лисс |
| <i>к.т.н., доцент</i> | <i>подпись</i> | |
| Консультанты | <hr/> | Д.А. Марков |
| | <i>подпись</i> | |

Санкт-Петербург

2023

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Утверждаю

И.о. зав. кафедрой МОЭВМ

_____ А.А. Лисс

«___» _____ 20__ г.

Студент Прашутинский К.И.

Группа 9381

Тема работы: Модернизация веб-приложения проведения приема в медицинской организации

Место выполнения ВКР: ООО «ИТВОРКС»

Исходные данные (технические требования):

Повысить скорость выполнения запросов сервиса записи на прием к врачу.

Создать веб-приложение для записи на прием к врачу с просмотром предстоящих записей пациентом и врачом, просмотр расписаний врачей.

Содержание ВКР:

Введение; Формулировка требований к решению и постановка задач; Архитектура программной реализации; Исследование механизмов оптимизации механизмов.

Перечень отчетных материалов: пояснительная записка, иллюстративный материал

Дополнительные разделы: Безопасность жизнедеятельности

Дата выдачи задания

«04» апреля 2023 г.

Дата представления ВКР к защите

«19» июня 2023 г.

Студент

_____ К.И. Прашутинский

Руководитель к.т.н., доцент

_____ А.А. Лисс

Консультант

_____ Д.А. Марков

КАЛЕНДАРНЫЙ ПЛАН ВЫПОЛНЕНИЯ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Утверждаю

И.о. зав. кафедрой МОЭВМ

_____ А.А. Лисс

«___» _____ 20__ г.

Студент К.И. Пращутинский

Группа 9381

Тема работы: Модернизация веб-приложения проведения приема в медицинской организации

| № п/п | Наименование работ | Срок выполнения |
|----------|--|------------------|
| 1 | Обзор литературы по теме работы | 04.04 – 15.04 |
| 2 | Обзор аналогов | 04.04 – 06.04 |
| 3 | Обзор предметной области | 04.04 – 27.04 |
| 4 | Постановка задачи и формализация функциональных требований | 24.04 – 27.04 |
| 5 | Исследование механизмов оптимизации кода программ и запросов | 24.04 – 30.04 |
| 6 | Описание решения | 24.04 – 10.05 |
| 7 | Исследование разработанного решения | 10.05 – 13.05 |
| 9 | Оформление пояснительной записки | 01.05 – 14.05 |
| 10 | Оформление иллюстративного материала | 04.04 – 14.05 |

Студент _____ К.И. Пращутинский

Руководитель _____ А.А. Лисс
к.т.н., доцент

Консультант _____ Д.А. Марков

РЕФЕРАТ

Пояснительная записка 56 стр., 28 рис., 4 табл., 13 ист.

КЛЮЧЕВЫЕ СЛОВА, ВАШЕГО, ДИПЛОМА.

ЗДРАВООХРАНЕНИЕ, ОНЛАЙН-ЗАПИСЬ, РАСПИСАНИЕ РАБОТЫ, ПРИЕМ К ВРАЧУ, ОПТИМИЗАЦИЯ ЗАПРОСОВ, 1С: МЕДИЦИНА, МЕДИЦИНСКИЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Объектом исследования является системы организации медицинской помощи и существующие системы записи на прием к врачу.

Предметом исследования является разработка онлайн-сервиса для записи на прием к врачу на основе алгоритма, который упростит процесс записи на прием и сократит время ожидания для пациентов.

Цель работы: разработка онлайн-сервиса для записи на приём к врачу с использованием веб-технологий.

Данный сервис должен облегчить процесс записи на приём для пациентов, уменьшить нагрузку на медицинские учреждения и исключить возможности отказа в записи из-за временной недоступности государственных ресурсов в связи с проведением внеплановых технических работ.

Дипломная работа посвящена разработке онлайн-сервиса для записи на приём к врачу, улучшению процесса онлайн-записи и повышению эффективности медицинских учреждений. В работе была поставлена задача создать удобный и доступный веб-сервис, который позволит пользователям быстро записаться на прием, просмотреть расписание работы специалистов и улучшить процесс планирования приёмов.

Для достижения поставленных целей были выполнены задачи по анализу существующих методов онлайн-записи на приём к врачу, разработке и оптимизации алгоритмов, созданию функционала просмотра расписания работы специалистов и врачей, а также разработке онлайн-сервиса с удобным пользовательским интерфейсом.

Использовались различные методы и инструменты, включая анализ литературных источников, моделирование и тестирование алгоритмов, методологию разработки ПО.

Результаты работы включают создание онлайн-сервиса для записи на приём к врачу и просмотра расписания работы специалистов, а также оптимизацию процесса онлайн-записи на приём. Эти изменения могут значительно улучшить качество и эффективность медицинской помощи и упростить процесс планирования и проведения приёма.

Таким образом, данная исследовательская работа имеет практическую значимость и может быть полезной для медицинских учреждений, которые стремятся улучшить качество предоставляемой медицинской помощи и повысить удобство и доступность для пациентов. Созданный онлайн-сервис с функционалом просмотра расписания работы специалистов и врачей, а также оптимизированные алгоритмы онлайн-записи могут быть использованы не только в медицинских учреждениях, но и в других сферах, которые требуют эффективной и удобной записи на приём, таких как государственные учреждения, образовательные учреждения и т.п.

Дополнительным преимуществом данной работы является использование современных технологий, таких как веб-технологии и методология разработки программного обеспечения. Было показано, что эти технологии могут быть успешно использованы для решения задач, связанных с оптимизацией процесса записи на приём и разработкой удобного и понятного интерфейса для пользователей.

В рамках будущих исследований можно рассмотреть вопрос о возможности расширения и адаптации данного сервиса для различных типов медицинских учреждений в зависимости от их особенностей и потребностей.

В целом, данная дипломная работа имеет практическую значимость и может быть полезной для медицинских учреждений в улучшении процесса онлайн-записи на приём, а также для программистов и разработчиков, занимающихся созданием веб-сервисов.

ABSTRACT

This service should facilitate the process of making an appointment for patients, reduce the burden on medical institutions and exclude the possibility of refusal to make an appointment due to the temporary unavailability of state resources due to unscheduled technical work.

The thesis is devoted to the development of an online service for making an appointment with a doctor, improving the online recording process and improving the efficiency of medical institutions. In the work, the task was set to create a convenient and accessible web service that will allow users to quickly make an appointment, view the work schedule of specialists and improve the process of scheduling appointments.

In order to achieve these goals, tasks were performed to analyze existing methods of online appointment to a doctor, develop and optimize algorithms, create functionality for viewing the work of specialists and doctors, as well as develop an online service with a user-friendly interface.

Various methods and tools were used, including analysis of literary sources, modeling and testing of algorithms, methodology of software development.

The results of the work include the creation of an online service for making an appointment with a doctor and viewing the work schedule of specialists, as well as the optimization of the online appointment process. These changes can significantly improve the quality and effectiveness of medical care and simplify the process of planning and conducting an appointment.

Thus, this research work has practical significance and can be useful for medical institutions that seek to improve the quality of medical care provided and increase convenience and accessibility for patients. The created online service with the functionality of viewing the work schedules of specialists and doctors, as well as optimized algorithms for online recording can be used not only in medical institutions, but also in other areas that require effective and convenient appointment, such as government agencies, educational institutions, etc.

An additional advantage of this work is the use of modern technologies, such as web technologies and software development methodology. It was shown that these technologies can be successfully used to solve problems related to optimizing the appointment process and developing a user-friendly and understandable interface for users.

As part of future research, it is possible to consider the possibility of expanding and adapting this service for various types of medical institutions, depending on their characteristics and needs.

In general, this thesis has practical significance and can be useful for medical institutions in improving the process of online appointment, as well as for programmers and developers involved in the creation of web services.

СОДЕРЖАНИЕ

| | |
|---|----|
| ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ..... | 12 |
| ВВЕДЕНИЕ | 14 |
| 1 Обзор литературы по теме работы..... | 16 |
| 1.1 Выводы по разделу | 16 |
| 2 Обзор аналогов..... | 17 |
| 2.1 Принцип отбора аналогов | 17 |
| 2.2 Отобранные аналоги | 17 |
| 2.3 Таблица сравнения аналогов | 18 |
| 2.2 Выводы по итогам сравнения..... | 18 |
| 3 Обзор предметной области..... | 20 |
| 3.1 Принцип взаимодействия МО и ЕГИСЗ..... | 20 |
| 3.2 Принцип работы СЗПВ в ЕГИСЗ | 20 |
| 3.3 Принцип взаимодействия регионального узла и МО | 22 |
| 3.4 Обзор методов для СЗПВ | 24 |
| 3.5 Принцип передачи данных между узлами СЗПВ | 40 |
| 3.6 Выбор прикладного решения для хранения и подготовки данных для СЗПВ | 41 |
| 3.7 Выбор прикладного решения для взаимодействия с пользователем СЗПВ | 41 |
| 3.8 Выводы по разделу | 42 |
| 4 Постановка задачи и формализация функциональных требований..... | 43 |
| 4.1 Выводы по разделу. | 44 |
| 5 Исследование механизмов оптимизации..... | 45 |
| 5.1 Соединения с подзапросами..... | 45 |
| 5.2 Соединения с виртуальными таблицами..... | 45 |
| 5.3 Несоответствие индексов и условий запроса | 45 |
| 5.4 Использование логического ИЛИ в условиях | 46 |
| 5.5 Использование подзапросов в условии соединения | 46 |
| 5.6 Получение данных через точку от полей составного типа..... | 47 |
| 5.7 Фильтрация виртуальных таблиц без использования параметров..... | 47 |
| 5.8 Выводы по разделу | 47 |
| 6 Описание решения | 49 |
| 6.1 Создание веб-приложения..... | 49 |
| 6.3 Модернизация запросов 1С..... | 52 |

| | |
|---|----|
| 6.3 Сценарии использования..... | 52 |
| 6.4 Выводы по разделу | 53 |
| 7 Исследование разработанного решения | 54 |
| ЗАКЛЮЧЕНИЕ..... | 56 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 57 |
| ПРИЛОЖЕНИЕ А..... | 58 |
| ПРИЛОЖЕНИЕ Б | 59 |
| ПРИЛОЖЕНИЕ В..... | 60 |

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ЯП – язык программирования;

ЕГИСЗ – Единая государственная информационная система в сфере здравоохранения

API_KEY – Секретный ключ

API – (Application Programming Interface) это набор протоколов и инструментов, которые позволяют разным приложениям взаимодействовать друг с другом.

CALL-центр – это централизованная служба обработки входящих и исходящих телефонных звонков, электронных писем, чатов и других форм коммуникации с клиентами компании или организации.

МИС – Медицинская информационная система

SOAP – Simple Object Access Protocol – простой протокол доступа к объектам

БД – База данных

МО – Медицинская организация

Пользователь – Гражданин, обратившийся за получением Услуги к ИС Пользователя

ИС Пользователя – Информационная система, предоставляющая гражданам возможность записи на прием к врачу или вызов врача на дом посредством взаимодействия с ЕГИСЗ

СНИЛС – Страховой номер индивидуального лицевого счёта

XML – Extensible Markup Language - язык разметки, используемый для хранения и передачи данных.

СЗПВ – Сервис записи на прием к врачу

НСИ – Нормативно-справочная информация

Узел СЗПВ – это компонент системы, который предоставляет функциональность записи пациентов на прием к врачу.

1С: Предприятие – это платформа для автоматизации управленческой деятельности предприятий и организаций различных отраслей.

1С: Медицина – это прикладное решение на базе платформы 1С: Предприятие, предназначенное для автоматизации работы медицинских учреждений.

JavaScript – высокоуровневый интерпретируемый язык программирования, который часто используется в веб-разработке.

React – JavaScript библиотека для разработки пользовательских интерфейсов.

ЛПУ – Лечебно-профилактическое учреждение

ФИО – Фамилия, имя, отчество

мс – Миллисекунда

ВВЕДЕНИЕ

Основной текст разделов, параметры — отступ от начала каждого абзаца 1,25 см, 14 шрифт TimesNewRoman, полуторный интервал, выравнивание по ширине, интервал перед и после абзаца нулевой. Эти настройки уже включены в стиль «Основной текст данного документа».

Во Введении необходимо последовательно (и в разных абзацах) описать Актуальность, Цель работы, Задачи работы, Объект исследования, Предмет исследования, Практическую ценность работы, перечислить публикации по теме работы (при наличии).

В современном мире быстро развивающихся технологий и научных достижений, медицина не является исключением. Одним из важных аспектов, в которых наука и технологии могут помочь медицине, является оптимизация процесса записи на приём. Сложности в организации процесса записи на приём в медицинских учреждениях актуальны и всегда привлекают внимание исследователей и разработчиков.

Цель данной работы заключается в разработке онлайн-сервиса для записи на приём к врачу с использованием веб-технологий. Данный сервис должен облегчить процесс записи на приём для пациентов, уменьшить нагрузку на медицинские учреждения и исключить возможности отказа в записи из-за временной недоступности государственных ресурсов в связи с проведением внеплановых технических работ.

Для достижения поставленной цели в работе были поставлены следующие задачи:

1. Изучить существующие системы записи на прием в медицинских учреждениях и выявить их недостатки.
2. Разработать концепцию онлайн-сервиса для записи на прием, учитывающую потребности как пациентов, так и медицинских учреждений.
3. Разработать прототип онлайн-сервиса для записи на прием, основанный на выбранной концепции.

4. Оценить эффективность и удобство использования прототипа в сравнении с существующими системами записи на прием.
5. Провести анализ полученных результатов и сформулировать рекомендации для дальнейшего улучшения прототипа.

Объектом исследования является системы организации медицинской помощи и существующие системы записи на прием к врачу.

Предметом исследования является разработка онлайн-сервиса для записи на прием к врачу на основе алгоритма, который упростит процесс записи на прием и сократит время ожидания для пациентов.

Практическая ценность данной работы заключается в том, что разработанный онлайн-сервис может быть использован медицинскими учреждениями для улучшения процесса записи на приём и снижения нагрузки на персонал. Кроме того, работа может быть полезной для программистов и разработчиков, занимающихся созданием веб-сервисов.

1 Обзор литературы по теме работы

Онлайн-сервисы для записи на прием к врачу - актуальны в современном мире, где все больше людей стремятся получать медицинскую помощь быстро и удобно. Существует множество публикаций, посвященных анализу и оптимизации процесса записи на прием в медицинских учреждениях.

Одним из способов оптимизации являются CALL-центры, предоставляющие услугу записи на прием к врачу, обладают некоторыми недостатками, которые могут быть уменьшены с помощью внедрения автоматизированных систем ответов на запросы пользователей. Однако, даже в случае балансировки нагрузки, операторов могут не хватать, что приводит к задержкам в обслуживании [1].

По данным исследования, проведенным по 9 МО Москвы было выяснено, что процент использования онлайн-записи на прием к врачу значительно увеличился за последние годы - с 24% в 2019 году до 61% в начале 2022 года, что говорит о высоком удобстве и скорости онлайн-записи. Однако, люди старшего возраста, пользуются онлайн-записью на прием к врачу меньше, чем другие группы населения. Тем не менее, с увеличением количества людей, использующих онлайн-сервисы в разных сферах жизни, можно заметить тенденцию того, что процент использования онлайн-записи на прием к врачу постепенно растет, включая и среди людей старшего возраста [2].

1.1 Выводы по разделу

Из вышеописанного следует, что онлайн-сервисы для записи на прием у врача являются актуальными и востребованными в современном мире. Они позволяют людям получать медицинскую помощь быстро и удобно.

2 Обзор аналогов

2.1 Принцип отбора аналогов

Аналоги были выбраны по следующим критериям:

- 1) **Доступность.** Данный критерий означает то, что человек, имеющий страховой полис, имеет право воспользоваться медицинской услугой и должен быть открытым для использования любым лицом.
- 2) **Декоммерциализация.** Данный критерий означает то, что человек, имеющий страховой полис, имеет право на бесплатное медицинское обслуживание и запись на консультацию с врачом должна быть бесплатной.
- 3) **Запись на очную консультацию.** Данный критерий означает то, что должна быть возможность онлайн-записи на очную консультацию.
- 4) **Зависимость от сторонних сервисов.** Данный критерий означает то, что сервис онлайн-записи использует другие сервисы для предоставления услуги.

2.2 Отобранные аналоги

gostelemed.ru [9]

Веб-сервис записи на бесплатную онлайн консультацию с регистрацией по страховому полису с сотрудниками «ГБУЗ ЛО "Кировская МБ" (Кировская межрайонная больница)».

health.yandex.ru [11]

Веб-сервис записи на платную онлайн и офлайн консультацию к врачу с регистрацией по номеру телефона с выбором специальности, врача и его стажа.

spb.docdoc.ru [13]

Веб-сервис записи на платную очную консультацию, онлайн консультацию или вызов врача с регистрацией по номеру телефона.

gosuslugi.ru [12]

Веб-сервис записи на бесплатную очную консультацию с регистрацией по номеру телефона или СНИЛС.

gorzdrav.spb.ru [10]

Веб-сервис записи на бесплатную телемедицинскую консультацию с регистрацией через портал gosuslugi.ru.

2.3 Таблица сравнения аналогов

В таблице 1 представлено сравнение разных решений СЗПВ. Таким образом «+» означает наличие свойства, а «-» означает отсутствие свойства.

Таблица 1 – Сравнение аналогов веб-сервисов онлайн-записи на прием к врачу.

| | Доступность | Декоммерциализация | Запись на очную консультацию | Зависимость |
|------------------|-------------|--------------------|------------------------------|-------------|
| gosteamed.ru | + | + | - | - |
| health.yandex.ru | + | - | - | - |
| spb.docdoc.ru | + | - | + | - |
| gosuslugi.ru | + | + | + | + |
| gorzdrav.spb.ru | - | + | + | + |

2.2 Выводы по итогам сравнения

В процессе сравнения аналогов были выявлены их преимущества и недостатки. Среди найденных вариантов лучшим оказался веб-сервис gosuslugi.ru, который отличается доступностью, декоммерциализацией и возможностью записаться на очную консультацию. Однако этот сервис зависит от сторонних сервисов, что может привести к его сбоям при онлайн-записи как на рис. 27 и рис. 28 в приложении Б. Поэтому необходим альтернативный сервис для уменьшения количества неудачных обращений. Таким образом, основными критериями требуемого веб-сервиса являются: доступность, декоммерциализация, возможность записаться на очную консультацию и независимость от сторонних сервисов.

3 Обзор предметной области

3.1 Принцип взаимодействия МО и ЕГИСЗ

Ядром здравоохранения России является ЕГИСЗ. Аббревиатура расшифровывается как Единая государственная информационная система в сфере здравоохранения. Ее главная цель – обеспечить связь ключевых игроков системы здравоохранения и их непрерывное взаимодействие. Система здравоохранения ЕГИСЗ разделена на разные уровни [3]. Федеральный уровень включает в себя региональные узлы, а региональные узлы включают в себя уже отдельные государственные медучреждения и частные клиники. Подробнее на рис. 1.

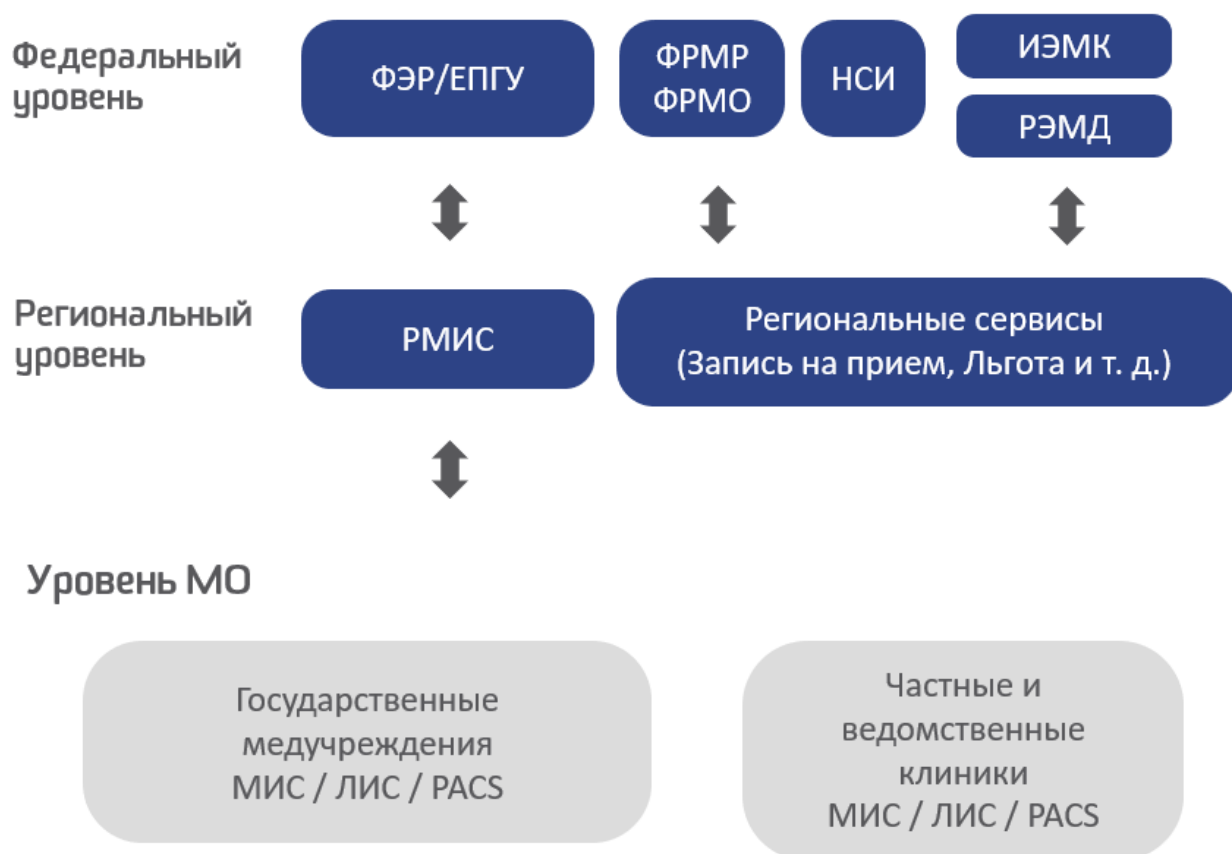


Рисунок 1 – Уровни системы ЕГИСЗ

3.2 Принцип работы СЗПВ в ЕГИСЗ

Во время записи пациента на прием через ЕПГУ или региональный портал немного отличаются, но, в целом, действуют по одному принципу. Подробнее на рис. 2.



Рисунок 2 – Общий принцип записи на прием к врачу

После предварительной записи на прием должна отработать обратная связь с пациентом, которая действует по принципу, изображенному на рис.3.

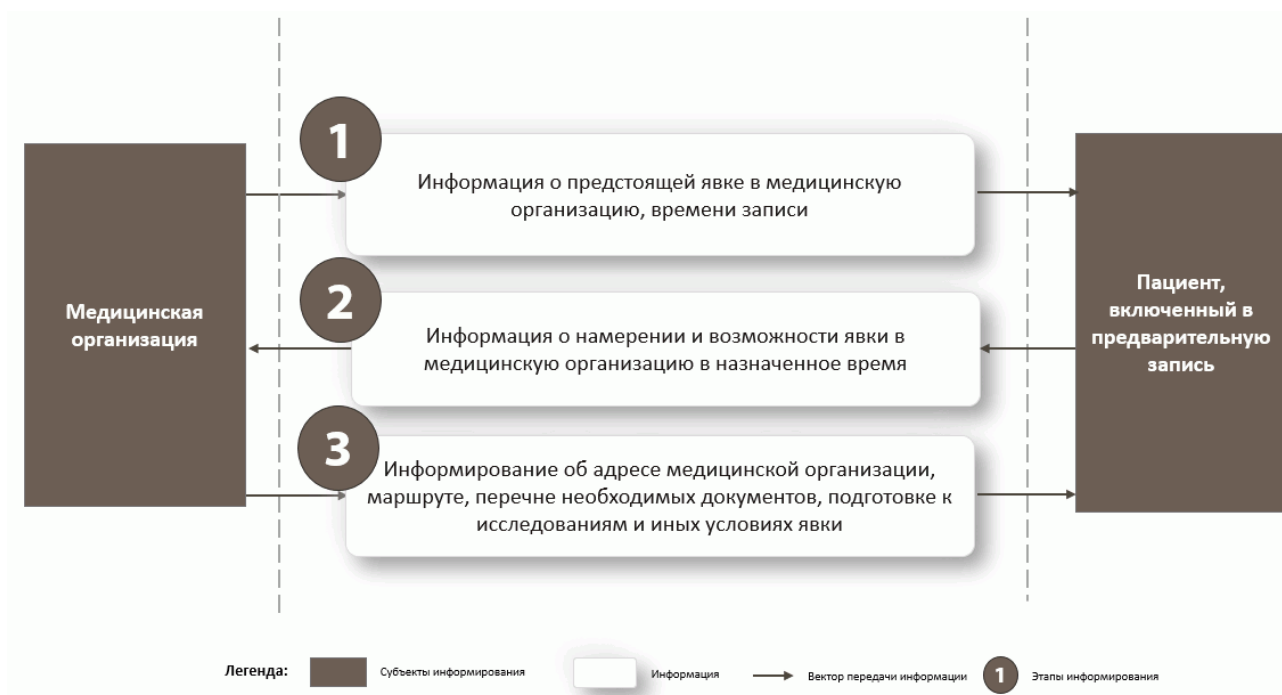


Рисунок 3 – Общая схема информирования пациента с целью оценки наполнения и актуализации расписания

Схема на рис. 3 описывает основной принцип записи на прием к врачу, но для успешного подключения МО к ЕГИСЗ необходимо ознакомиться с API Нетрики [3], который является региональным сервисом записи на прием. Использование стандартных методов взаимодействия с МО посредством изучения этого API поможет реализовать типовой механизм прямого взаимодействия без использования сторонних сервисов. Более подробно описано в методических рекомендациях N 12-22 [4].

3.3 Принцип взаимодействия регионального узла и МО

Для подключения МО к СЗПВ ЕГИСЗ требуется использовать методы API Нетрики [3], которые между собой тем, что одни методы используются для взаимодействия с федеральными сервисами записи, а другие методы используются для взаимодействия с региональными сервисами записи.

Список методов, используемых в СЗПВ:

- Выбор района (GetDistrictList).
- Выбор ЛПУ (GetLPUList).
- Идентификация пациента (CheckPatient).
- Выбор специальности (GetSpecialityList).
- Выбор врача (GetDoctorList).
- Выбор даты приема (GetAvailableDates).
- Выбор времени приема (GetAvaibleAppointments).
- Подтверждение записи (SetAppointment).
- Поиск пациента (SearchTop10patient).
- Добавление пациента в стороннюю МИС (AddNewPatient).
- Уточнение телефона пациента (UpdatePhoneByIdPat).
- Просмотр предстоящих записей пациента (GetPatientHistory).
- Подача заявки на отмену записи (CreateClaimForRefusal).
- Просмотр расписаний (без записи) (GetWorkingTime).

- Получение списка врачей, сгруппированных по специальности (GetDocListFullTree).
- Определение пользователя по его GUID (GetHubUserByGuid).
- Уведомление о записи на прием (SendNotificationAboutAppointment).
- Уведомление о изменении статуса записи на прием (SendNotificationAboutAppointmentStatus).
- Получение доступных врачей для записи пациента на прием (GetAvailableDoctors).
- Получение списка должностей (GetPositionList).
- Получение списка врачей по должности (GetDoctorList2).
- Получение списка услуг (GetServiceList).

Данные методы позволяют совершить полный цикл записи на прием к врачу как на федеральном, так и на региональном уровне. Схема работы указана на рис. 26 приложения А. В таблице 2 представлены методы, предоставляющие схожие данные, но имеющие разные условия отборы. Так, для федерального уровня используется отбор по справочнику «ФРМР. Должности медицинского персонала» (Код: 1.2.643.5.1.13.11.1102) [5], а для регионального отбор происходит по справочнику «Номенклатура специальностей специалистов с высшим и послевузовским медицинским и фармацевтическим образованием в сфере здравоохранения» (Код: 1.2.643.5.1.13.2.1.1.181) [5].

Таблица 2 – Сопоставление методов федерального и регионального уровней СЗПВ ЕГИСЗ.

| Пояснение метода | Федеральный уровень | Региональный уровень |
|---|---------------------|----------------------|
| Получить список доступных должностей/специальностей | GetPositionList | GetSpecialityList |
| Получить список доступных врачей по должности/специальности | GetDoctorList2 | GetDoctorList |

3.4 Обзор методов для СЗПВ

Для реализации полного цикла процесса записи и получения информации для дальнейшей записи были отобраны основные методы: CheckPatient, GetSpecialityList, GetDoctorList, GetAvaibleAppointments, SetAppointment, CreateClaimForRefusal, GetWorkingTime, GetPatientHistory. Данные методы будут рассмотрены далее. [3]

3.4.1 CheckPatient

Данный метод используется для идентификации пациента в МИС ЛПУ.

Для реализации запроса CheckPatient (рис. 4) СЗПВ необходимы следующие параметры:

- Авторизационный токен (guid)
- Идентификатор ЛПУ из справочника «ЛПУ» Интеграционной платформы (idLpu)
- СНИЛС пациента (Snils)
- Дата рождения (Birthday)

Идентификация пациента будет происходить по связке СНИЛС + день рождения.

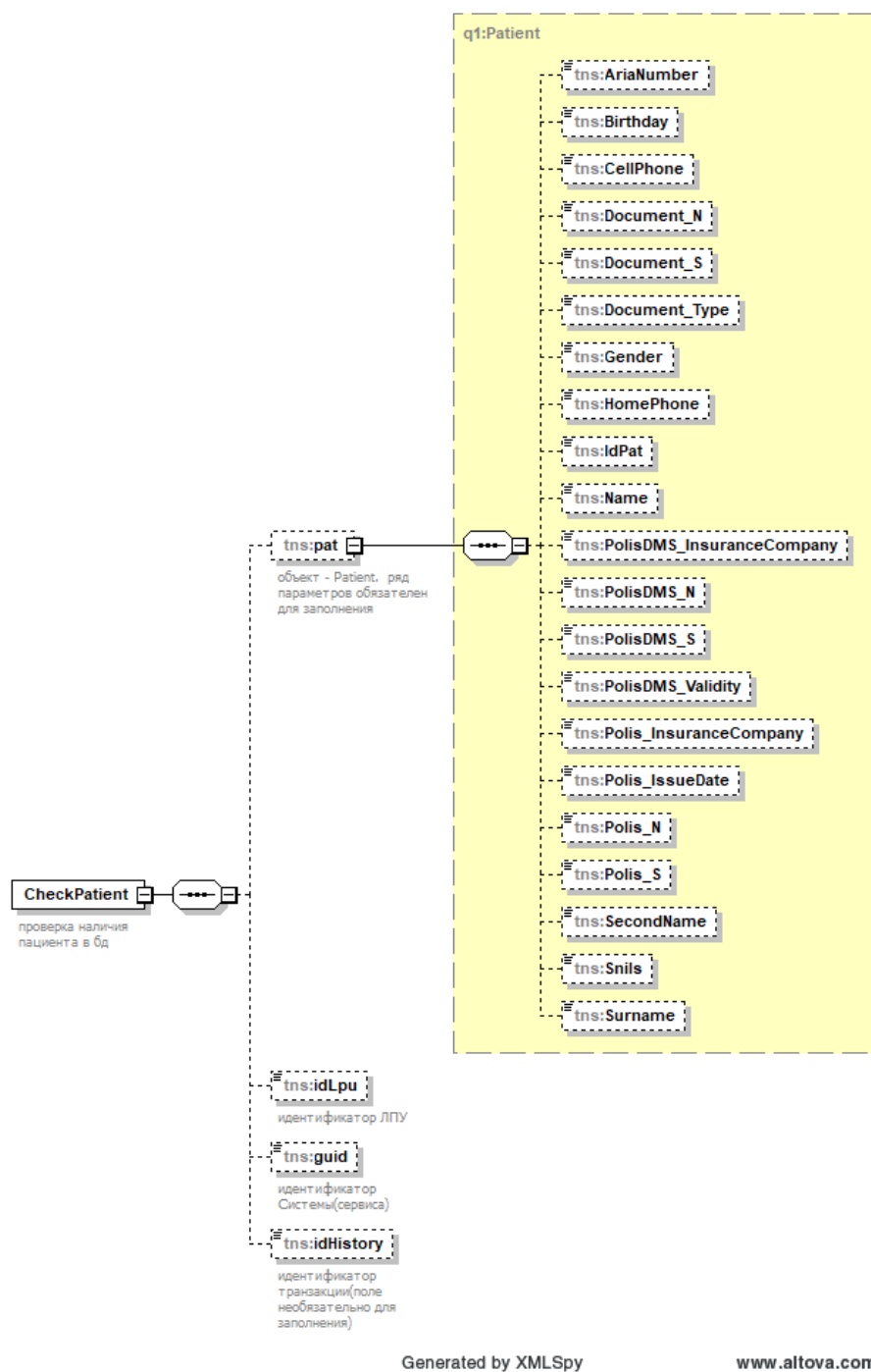


Рисунок 4 – Структура запроса CheckPatient

В ответе CheckPatient (рис. 5) СЗПВ необходимо получить следующие параметры:

- Идентификатор пациента (idPat)
- Идентификатор ошибки в справочнике (IdError)
- Результат выполнения запроса (Success)

В ответе содержится идентификатор пациента или код ошибки.

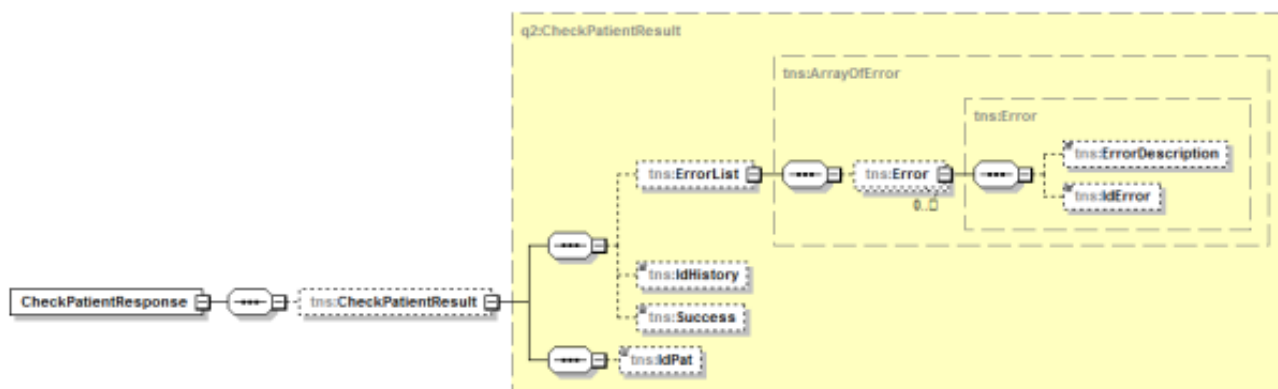


Рисунок 5 – Структура ответа CheckPatient

3.4.2 GetSpesialityList

Данный метод используется для получения списка врачебных специальностей в ЛПУ, запись к врачам которых доступна для пациента, идентификатор которого вводится в запрос метода. Список специальностей, к врачам которой для пациента доступна запись, определяется на стороне МИС ЛПУ.

Для реализации запроса GetSpesialityList (рис. 6) СЗПВ необходимы следующие параметры:

- Авторизационный токен (guid)
- Идентификатор ЛПУ из справочника «ЛПУ» Интеграционной платформы (idLpu)
- Идентификатор пациента (idPat)

Получение списка специальностей происходит по идентификатору ЛПУ.



Рисунок 6 – Структура запроса GetSpesialityList

В ответе GetSpesialityList (рис. 7) СЗПВ необходимо получить следующие параметры:

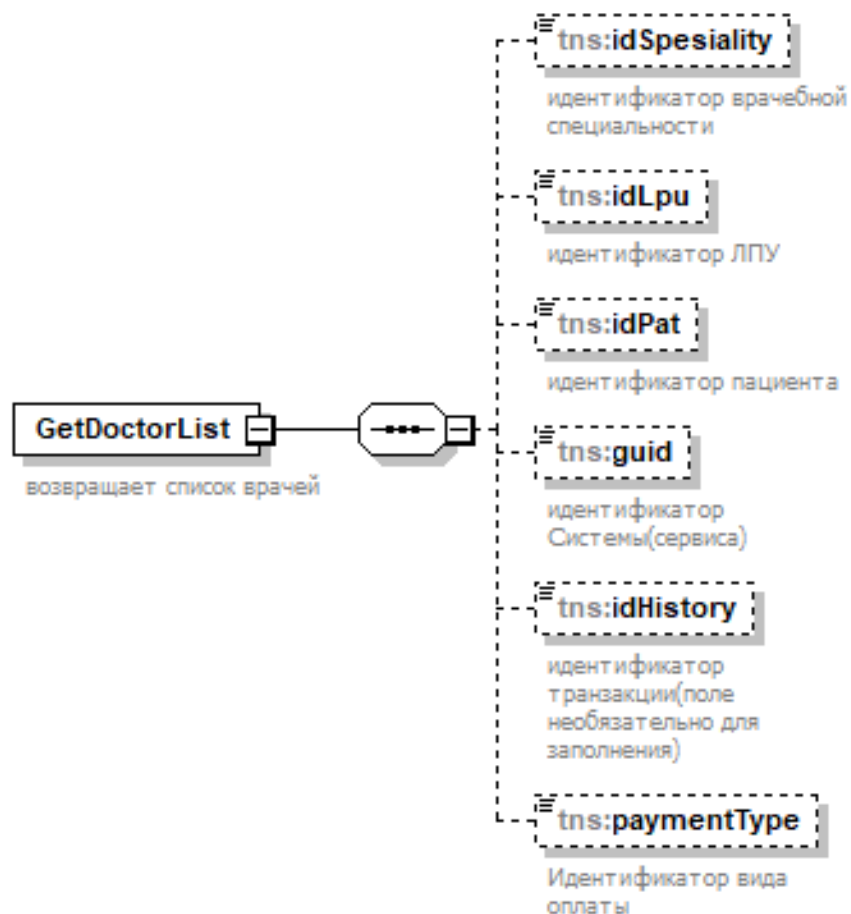
- Идентификатор ошибки в справочнике (IdError)
- Наименование врачебной специальности (NameSpesiality)
- Идентификатор врачебной специальности (IdSpesiality)
- Количество доступных участнику информационного обмена для записи талонов по врачебной специальности (CountFreeParticipantIE)
- Общее количество свободных талонов по врачебной специальности (CountFreeTicket)
- Дата приема по последнему свободному талону врачебной специальности (LastDate)
- Дата приема по ближайшему свободному талону врачебной специальности (NearestDate)
- Результат выполнения запроса (Success)

В ответе содержится список специальностей с количеством талонов, датой первого и последнего талона или код ошибки.

Рисунок 7 – Структура ответа GetSpesialityList

Данный метод используется для получения списка врачей указанной специальности, запись к которым доступна для пациента, идентификатор которого вводится в запрос метода. Список врачей, запись к которым доступна для пациента, определяется на стороне МИС ЛПУ.

- Авторизационный токен (guid)
- Идентификатор ЛПУ из справочника «ЛПУ» Интеграционной платформы (idLpu)
- Идентификатор пациента (idPat)
- Идентификатор врачебной специальности в справочнике целевой МИС (idSpesiality)



Generated by XMLSpy

www.altova.com

Рисунок 8 – Структура ответа GetDoctorList

Для ответа GetDoctorList (рис. 9) СЗПВ необходимо получить следующие параметры:

- Идентификатор ошибки в справочнике (IdError)
- Идентификатор врача в справочнике (IdDoc)
- ФИО врача (Name)
- СНИЛС врача (Snils)
- Количество доступных участнику информационного обмена для записи талонов по врачебной специальности (CountFreeParticipantIE)
- Общее количество свободных талонов по врачебной специальности (CountFreeTicket)
- Дата приема по последнему свободному талону врачебной специальности (LastDate)

- Дата приема по ближайшему свободному талону врачебной специальности (NearestDate)
- Результат выполнения запроса (Success)

В ответе содержится список врачей с количеством талонов, датой первого и последнего талона или код ошибки.

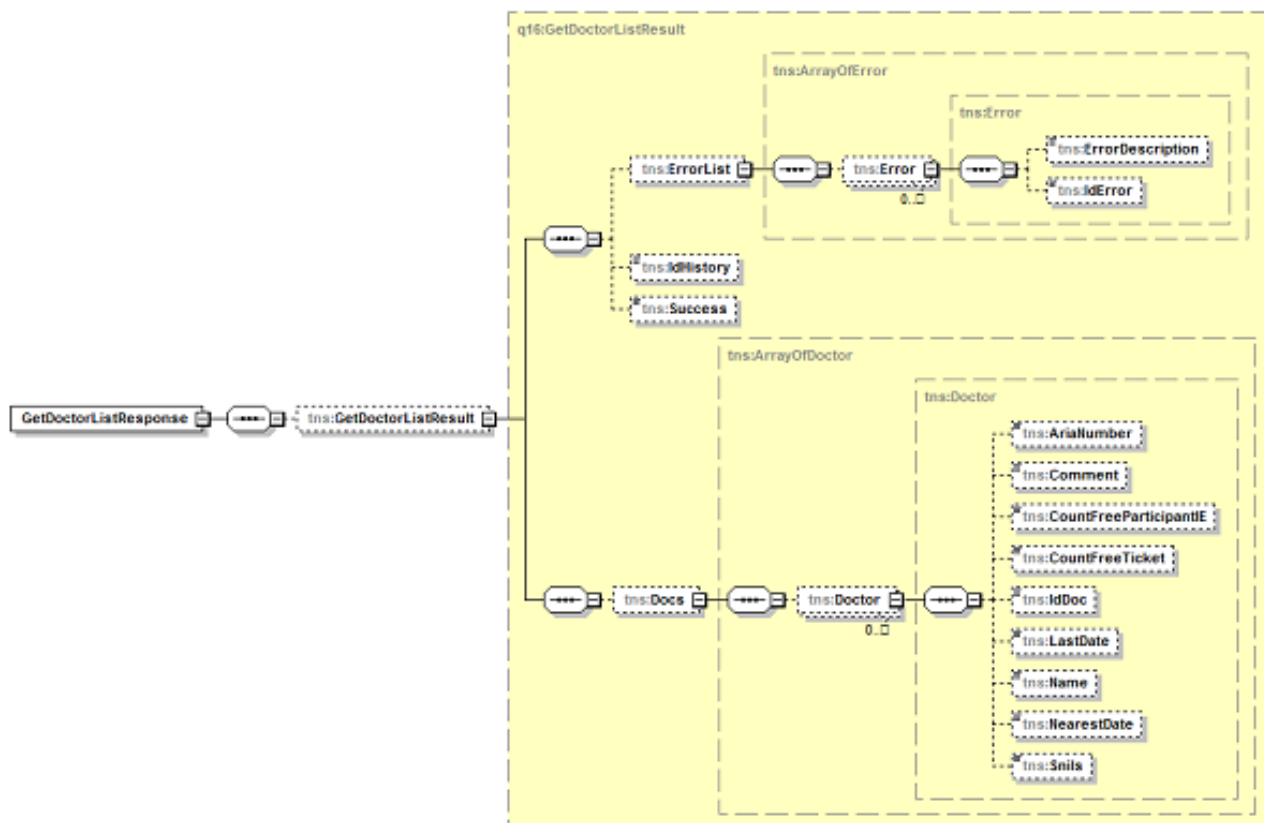


Рисунок 9 – Структура ответа GetDoctorList

3.4.4 GetAvaibleAppointments

Данный метод используется для получения списка талонов (свободных временных интервалов) указанного врача, запись на которые доступна для указанного в запросе пациента. Список талонов, доступных для выбора пациентом, определяется на стороне МИС ЛПУ.

Для реализации запроса GetAvaibleAppointments (рис. 10) СЗПВ необходимы следующие параметры:

- Авторизационный токен (guid)
- Идентификатор ЛПУ из справочника «ЛПУ» Интеграционной платформы (idLpu)

- Идентификатор пациента (idPat)
- Идентификатор врача в справочнике целевой МИС (idDoc)
- Дата начала диапазона поиска свободных талонов (visitStart)
- Дата окончания диапазона поиска свободных талонов (visitEnd)

Получение списка талонов происходит по идентификатору ЛПУ, времени начала и конца диапазона поиска и идентификатору врача.

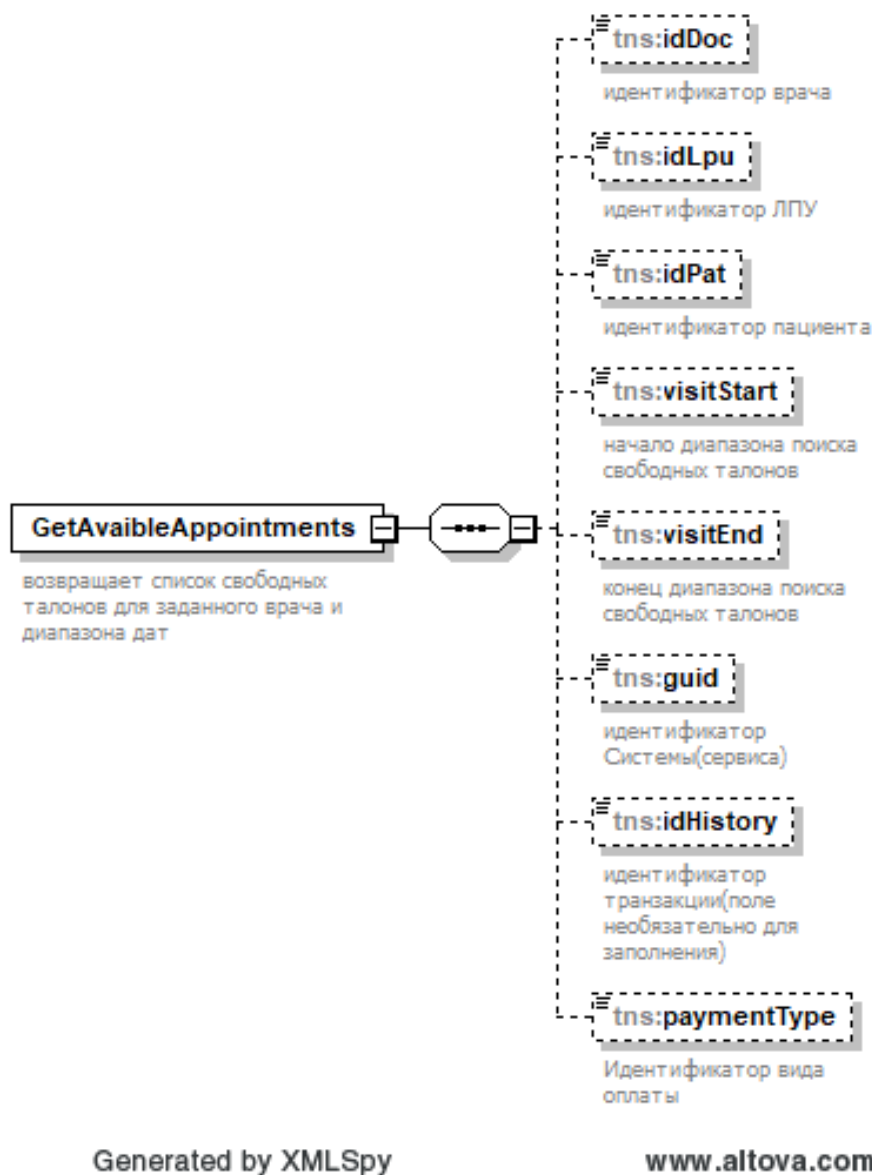


Рисунок 10 – Структура запроса GetAvaibleAppointments

В ответе GetAvaibleAppointments (рис. 11) СЗПВ необходимо получить следующие параметры:

- Идентификатор ошибки в справочнике (IdError)
- Идентификатор талона для записи (IdAppointment)

- Номер талона в очереди (Num)
- Номер кабинета (Room)
- Дата и время начала приема (VisitStart)
- Дата и время окончания приема (VisitEnd)
- Результат выполнения запроса (Success)

В ответе содержится список талонов с номером талона в очереди, номером кабинета, датой первого и последнего талона или код ошибки.

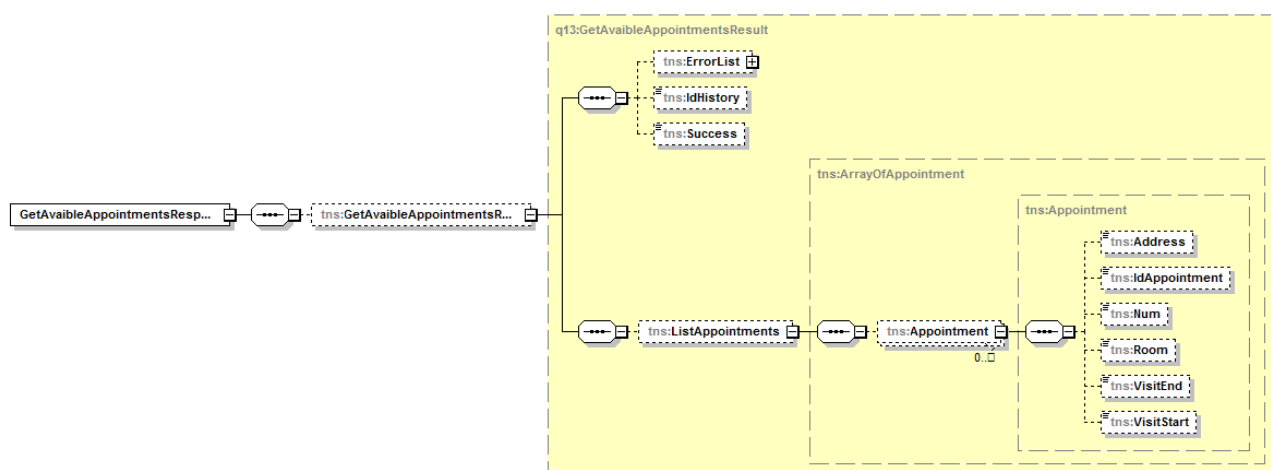


Рисунок 11 – Структура ответа GetAvaliableAppointments

3.4.5 SetAppointment

Данный метод используется для записи пациента на прием в выбранный пациентом временной интервал рабочего времени врача.

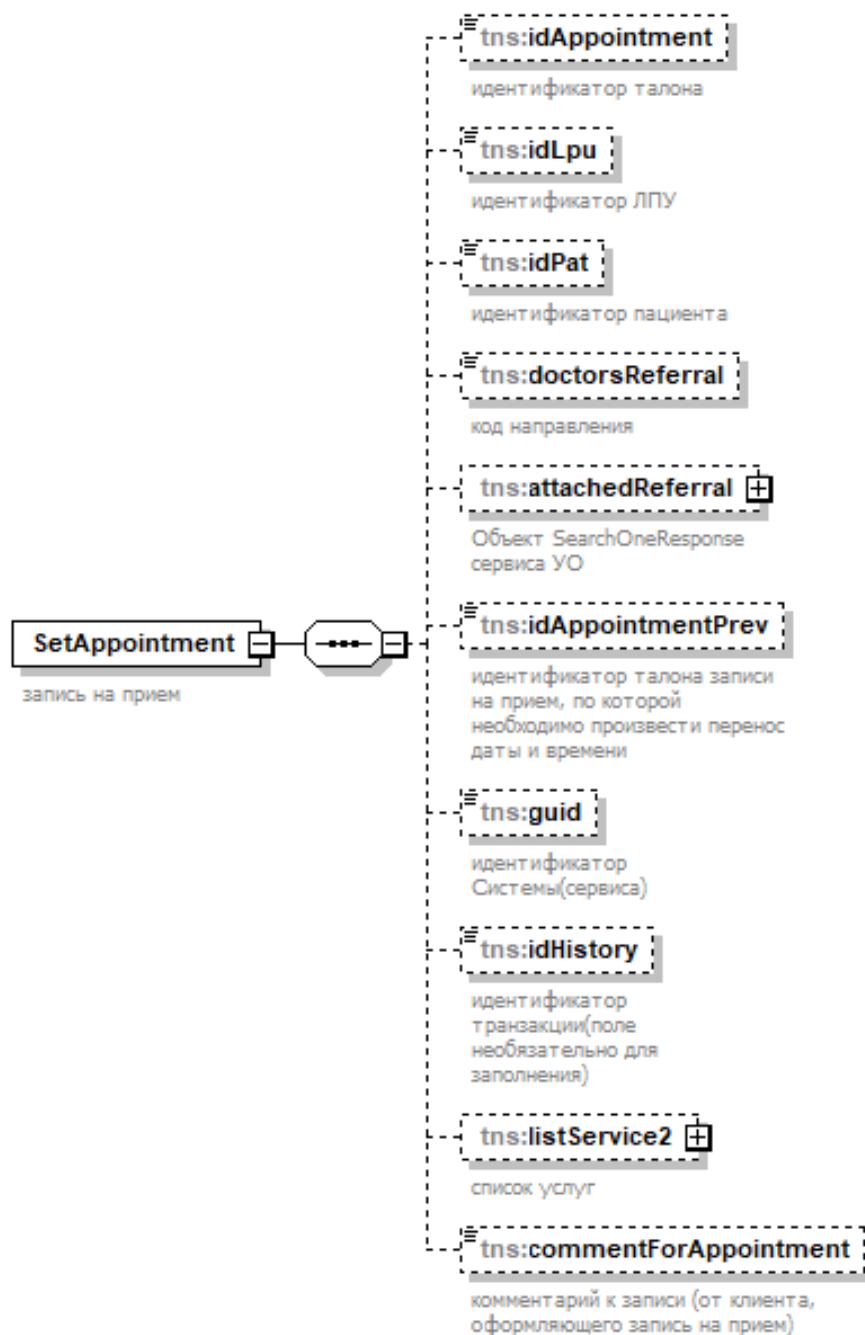
Если в запросе метода заполнен параметр idAppointmentPrev, то осуществляется перенос даты и времени приема по ранее оформленной записи.

Для реализации запроса SetAppointment (рис. 12) СЗПВ необходимы следующие параметры:

- Авторизационный токен (guid)
- Идентификатор ЛПУ из справочника «ЛПУ» Интеграционной платформы (idLpu)
- Идентификатор пациента (idPat)
- Идентификатор талона для записи из соответствующего справочника целевой МИС (idAppointment)

- Идентификатор талона записи на прием, по которой необходимо произвести перенос даты и времени (указывается идентификатор талона по существующей записи на прием) (idAppointmentPrev)

Запись на прием к врачу происходит по идентификатору ЛПУ, идентификатору пациента и идентификатору талона.



Generated by XMLSpy

www.altova.com

Рисунок 12 – Структура запроса SetAppointment

В ответе SetAppointment (рис. 13) СЗПВ необходимо получить следующие параметры:

- Идентификатор ошибки в справочнике (IdError)
- Тип специалиста, к которому записался пациент (Type)
- Результат выполнения запроса (Success)

В ответе содержится тип специалиста, к которому записался пациент, или код ошибки.

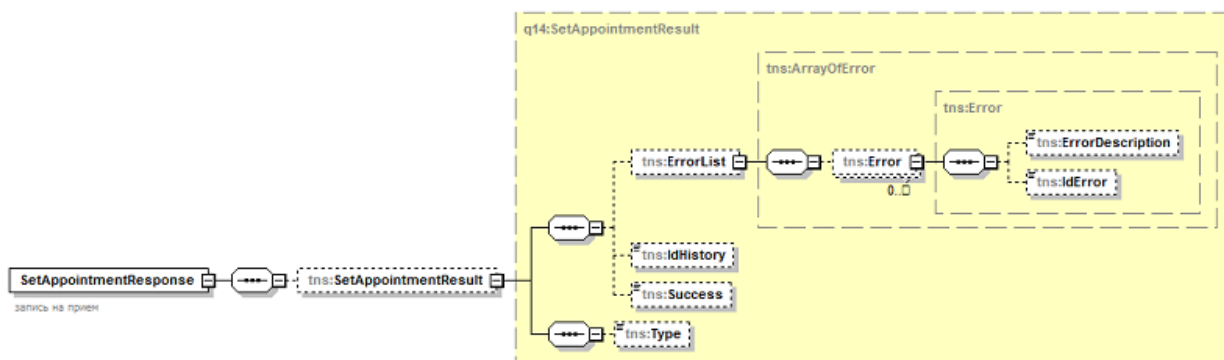


Рисунок 13 – Структура ответа SetAppointment

3.4.6 CreateClaimForRefusal

Данный метод позволяет создавать заявки на отмену записи на прием. Используется в типовом решении для записи на прием через ЦТО.

Для реализации запроса CreateClaimForRefusal (рис. 14) СЗПВ необходимы следующие параметры:

- Авторизационный токен (guid)
- Идентификатор ЛПУ из справочника «ЛПУ» Интеграционной платформы (idLpu)
- Идентификатор пациента (idPat)
- Идентификатор талона для записи из соответствующего справочника МИС (IdAppointment)

Отмена записи на прием происходит по идентификатору ЛПУ, идентификатору пациента и идентификатору талона. Перенос записи на прием происходит по идентификатору ЛПУ, идентификатору пациента, идентификатору талона и идентификатору талона, с которого будет произведён перенос.

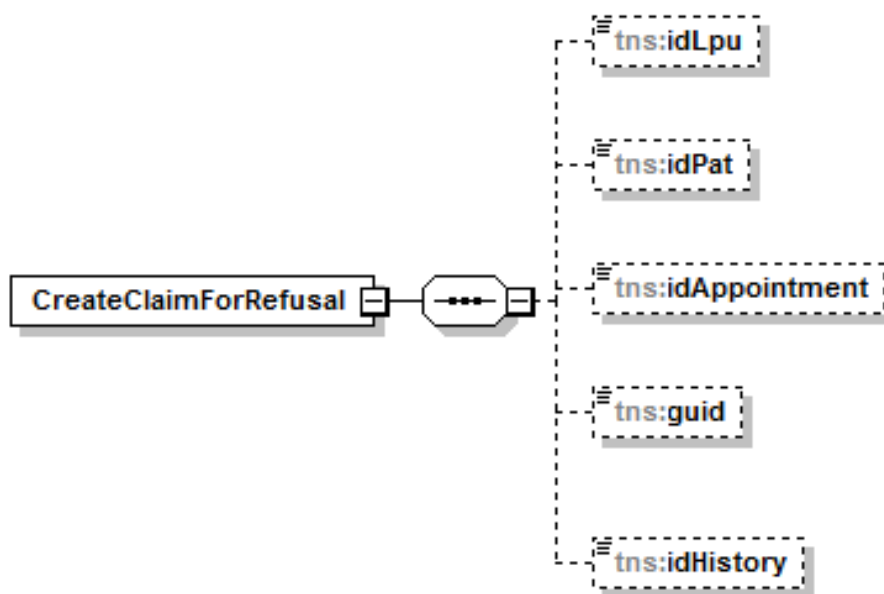


Рисунок 14 – Структура запроса CreateClaimForRefusal

В ответе CreateClaimForRefusal (рис. 15) СЗПВ необходимо получить следующие параметры:

- Идентификатор ошибки в справочнике (IdError)
- Результат выполнения запроса (Success)

В ответе содержится статус выполнения запроса или код ошибки.

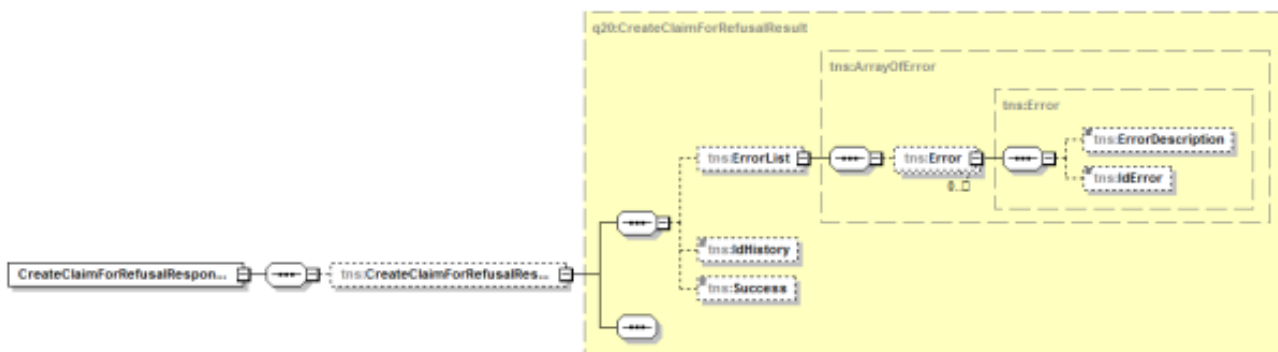


Рисунок 15 – Структура ответа CreateClaimForRefusal

3.4.7 GetWorkingTime

Данный метод используется для получения информации о времени работы врача (возвращает часы приема врача), идентификатор которого указан в запросе.

Для реализации запроса GetWorkingTime (рис. 16) СЗПВ необходимы следующие параметры:

- Авторизационный токен (guid)
- Идентификатор ЛПУ из справочника «ЛПУ» Интеграционной платформы (idLpu)
- Идентификатор врача в справочнике целевой МИС (idDoc)
- Дата начала диапазона поиска для получения информации о расписании (visitStart)
- Дата окончания диапазона поиска для получения информации о расписании (visitEnd)

Получение расписания врача происходит по идентификатору ЛПУ, идентификатору врача и времени начала и конца диапазона поиска.

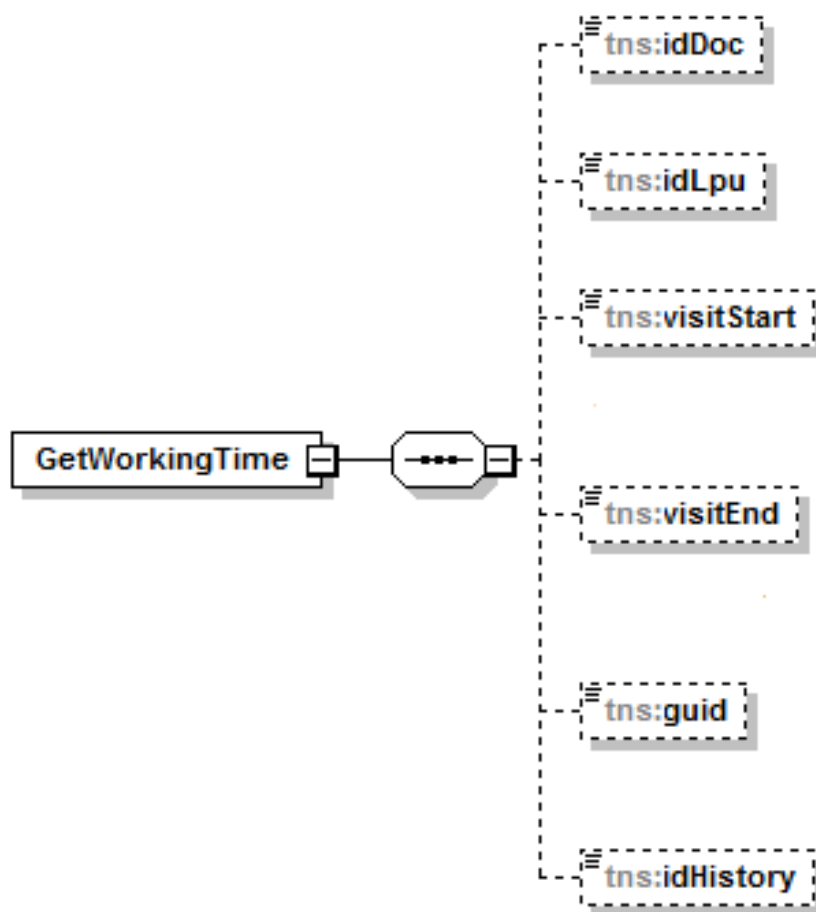


Рисунок 16 – Структура запроса GetWorkingTime

В ответе GetWorkingTime (рис. 17) СЗПВ необходимо получить следующие параметры:

- Идентификатор ошибки в справочнике (IdError)
- Дата и время начала приема (VisitStart)
- Дата и время окончания приема (VisitEnd)
- Возможна ли запись на прием (RecordableDay)
- Описание причины отсутствия записи (DenyCause)
- Результат выполнения запроса (Success)

В ответе содержится список дней с началом и концом времени приема, параметр возможности записи и описание причины отсутствия записи или код ошибки.

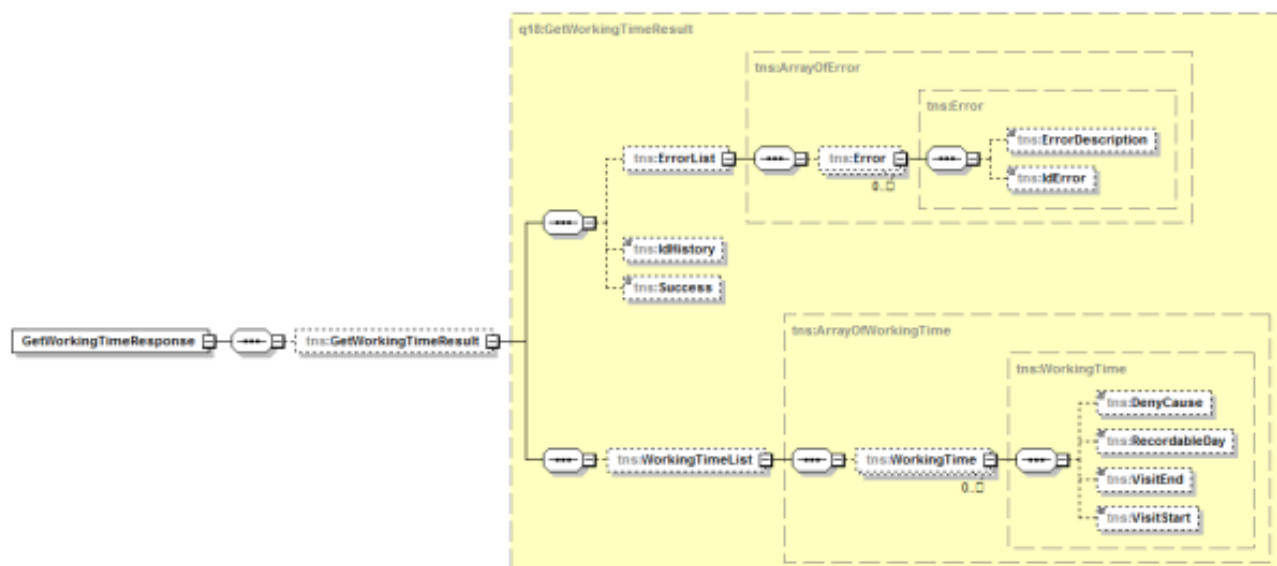


Рисунок 17 – Структура ответа GetWorkingTime

3.4.8 GetPatientHistory

Данный метод используется для получения списка записей пациента на прием и отказов от записи на прием в указанном в запросе ЛПУ, даты по которым еще не наступили.

Для реализации запроса GetPatientHistory (рис. 18) СЗПВ необходимы следующие параметры:

- Авторизационный токен (guid)
- Идентификатор ЛПУ из справочника «ЛПУ» Интеграционной платформы (idLpu)
- Идентификатор пациента (idPat)

Получение расписания врача происходит по идентификатору ЛПУ и идентификатору пациента.

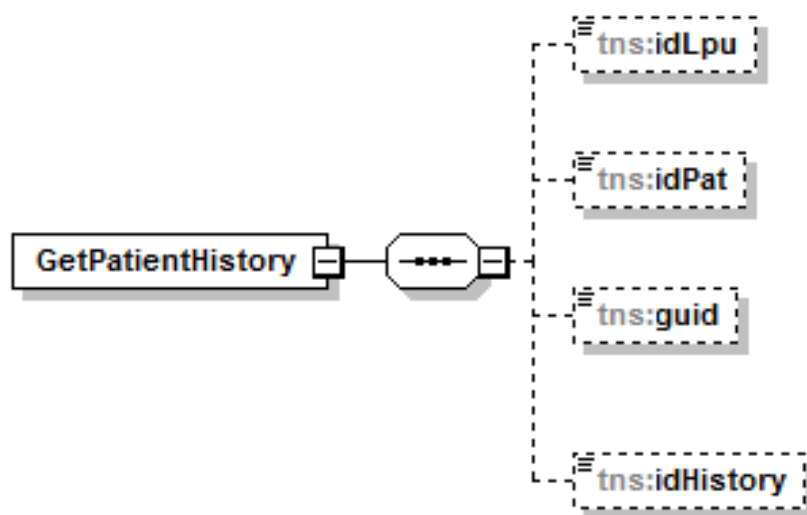


Рисунок 18 – Структура запроса GetPatientHistory

В ответе GetPatientHistory (рис. 19) СЗПВ необходимо получить следующие параметры:

- Идентификатор ошибки в справочнике (IdError)
- Дата создания записи на прием (DateCreatedAppointment)
- Идентификатор талона на запись (IdAppointment)
- Дата начала приема (VisitStart)
- Идентификатор врача (IdDoc)
- ФИО врача (Name)
- Идентификатор врачебной специальности в справочнике (IdSpesiality)
- Наименование врачебной специальности (NameSpesiality)

В ответе содержится список записей пациента с началом времени приема, ФИО врача и специальность врача или код ошибки.

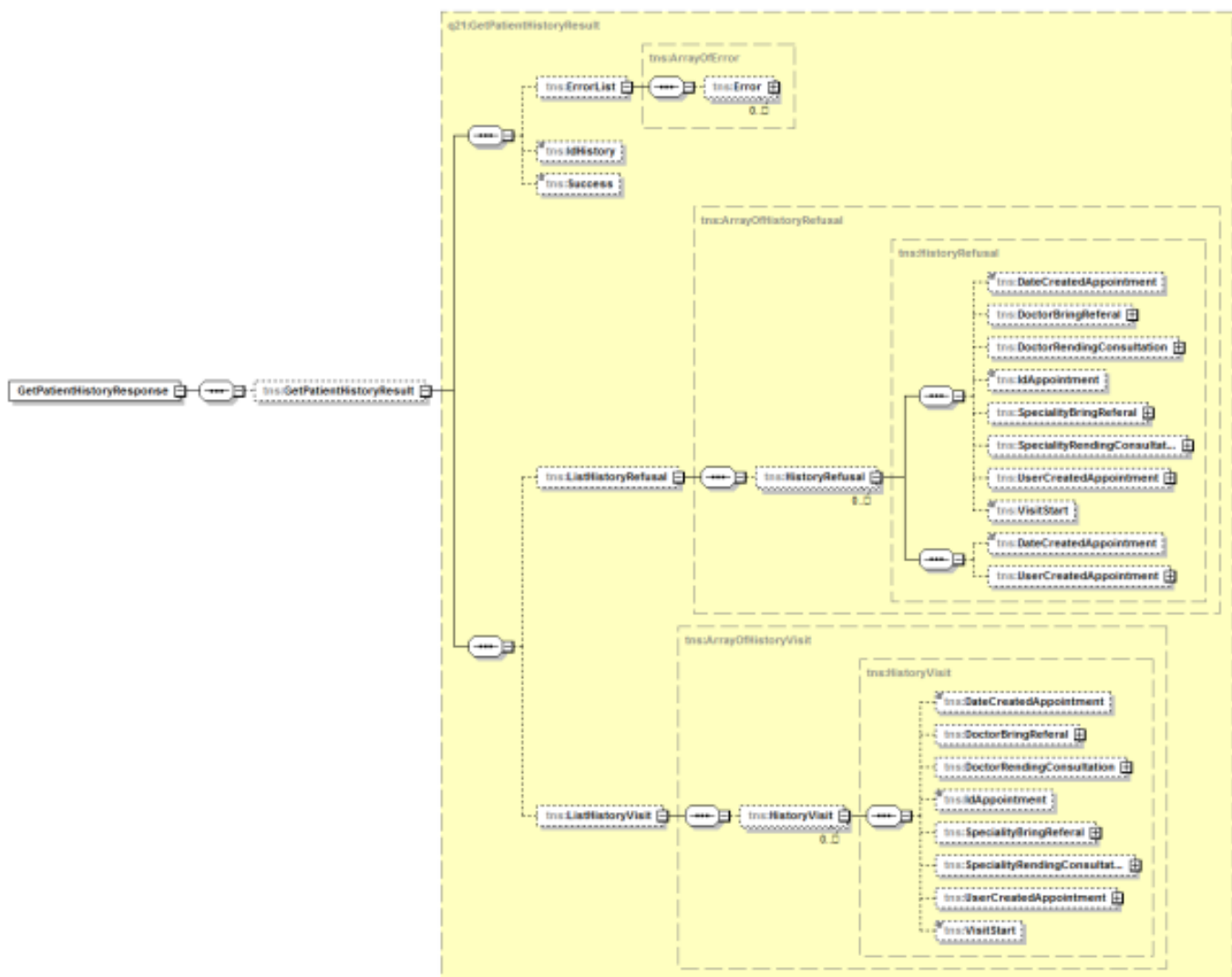


Рисунок 19 – Структура ответа GetPatientHistory

3.5 Принцип передачи данных между узлами СЗПВ

Для передачи данных между узлами СЗПВ используется протокол SOAP и формат XML. Для этого данные упаковываются в виде XML-документа (рис. 20) и отправляются через сообщение SOAP. Затем получатель распаковывает сообщение и извлекает данные из XML-документа. Протокол SOAP и формат XML используются для обеспечения стандартизации и унификации передачи данных между узлами в СЗПВ.


```

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:tem="http://tempuri.org">
  <soap:Header/>
  <soap:Body>
    <tem:GetSpecialityList>
      <!--Optional:-->
      <tem:idLpu>15</tem:idLpu>
      <!--Optional:-->
      <tem:idPat>d4085cc7-6d9f-4c87-944c-8dbb08bd2760</tem:idPat>
      <!--Optional:-->
      <tem:guid>b5de20c2-8aa6-4667-b2d0-d972c892f79b</tem:guid>
      <!--Optional:-->
      <tem:idHistory>2b177aad-eb83-4a8c-81fc-4a6215878ad2</tem:idHistory>
    </tem:GetSpecialityList>
  </soap:Body>
</soap:Envelope>

```

Рисунок 20 – Пример XML для обмена по протоколу SOAP

3.6 Выбор прикладного решения для хранения и подготовки данных для СЗПВ

Реализация хранения и подготовки данных для ЗПВ будет осуществлена на базе прикладного решения 1С: Медицина. Данный выбор обоснован основным стеком технологий компании ООО «ИТВОРКС», в которой уже есть реализация методов взаимодействия с узлами СЗПВ ЕГИСЗ.

3.7 Выбор прикладного решения для взаимодействия с пользователем СЗПВ

Для создания оболочки СЗПВ был выбран ЯП JavaScript с библиотекой React. JavaScript был выбран для написания прикладного решения для взаимодействия с пользователем СЗПВ из-за его широкого распространения в веб-разработке и его возможностей для создания динамических веб-приложений. JavaScript является одним из самых популярных языков программирования в мире, поэтому его знание является необходимым для многих разработчиков.

React, с другой стороны, был выбран как библиотека для разработки пользовательского интерфейса из-за ее компонентного подхода и высокой производительности. React позволяет легко создавать динамические и масштабируемые веб-приложения с использованием компонентного подхода, что облегчает создание и поддержку кода. Он также использует виртуальную

DOM, что позволяет более эффективно обновлять пользовательский интерфейс и повышает производительность веб-приложения.

Вместе JavaScript и React обеспечивают мощный инструментарий для создания прикладного решения для взаимодействия с пользователем СЗПВ. JavaScript позволяет создавать динамические и интерактивные элементы на веб-страницах, а React облегчает создание и поддержку пользовательского интерфейса, делая его более легким и управляемым. Кроме того, их популярность обеспечивает широкую поддержку сообщества разработчиков и обширную документацию, что делает разработку и поддержку приложения более простой и удобной.

3.8 Выводы по разделу

Исходя из этих выводов, можно сделать заключение, что взаимодействие медицинских организаций с ЕГИСЗ через СЗПВ является сложным и многоступенчатым процессом. Он основывается на использовании стандартизированных методов и протоколов, которые обеспечивают безопасность и качество передачи информации между системами. Выбор прикладных решений для хранения и подготовки данных, а также взаимодействия с пользователем, имеет важное значение для эффективной работы системы и удобства использования МО.

4 Постановка задачи и формализация функциональных требований

Для достижения поставленной цели в работе были поставлены следующие задачи:

1. Изучить существующие системы записи на прием в медицинских учреждениях и выявить их недостатки.
2. Разработать концепцию онлайн-сервиса для записи на прием, учитывающую потребности как пациентов, так и медицинских учреждений.
3. Разработать прототип онлайн-сервиса для записи на прием, основанный на выбранной концепции.
4. Оценить эффективность и удобство использования прототипа в сравнении с существующими системами записи на прием.
5. Провести анализ полученных результатов и сформулировать рекомендации для дальнейшего улучшения прототипа.

Функциональные требования:

1. Регистрация и авторизация пользователей.
2. Возможность просмотра информации о медицинском учреждении и медицинском персонале, в том числе доступных дат и времени приема.
3. Возможность записаться на прием в медицинский организации через предоставленную форму.
4. Оптимизация запросов, которая позволит пользователю получить медицинскую помощь быстрее и эффективнее.
5. Отправка напоминаний о предстоящем приеме.
6. Возможность отмены или переноса приема.

Таким образом, сервис должен обеспечивать удобную и быструю запись на прием и оптимизировать запросы, позволяя пользователям получать необходимую медицинскую помощь с меньшими временными затратами. Помимо функциональных требований, к разработке сервиса записи на прием и оптимизации запросов следует учесть его надежность и устойчивость к от-

казам. Важно иметь возможность альтернативной записи на прием в случае, если основной сервис, включающий все медицинские организации, не будет работать. Пример таких ситуаций в Приложении Б.

4.1 Выводы по разделу.

Из названных в тексте задач и функциональных требований можно сделать выводы о том, что разработка онлайн-сервиса для записи на прием в медицинских учреждениях является важной задачей, которая должна решаться с учетом потребностей как пациентов, так и медицинских организаций. Онлайн-сервис должен быть удобен и эффективен для пользователей, позволяя им легко записываться на прием и получать необходимые услуги. Оптимизация запросов является ключевой функциональной требованием, которое позволит сократить время ожидания и улучшить качество медицинского обслуживания. Кроме того, сервис должен обладать высокой надежностью и устойчивостью к отказам, чтобы обеспечить пользователей альтернативными вариантами записи на прием в случае технических проблем.

5 Исследование механизмов оптимизации

Самым долгим процессом в выполнении метода СЗВП является запрос в БД. Это связано с тем, что база данных содержит большой объем информации, и выполнение запросов может занимать значительное время. Для оптимизации запросов 1С рекомендуются избегать неоптимальных ситуаций [6].

5.1 Соединения с подзапросами

Если запрос содержит соединения с подзапросами, то это может привести к следующим негативным последствиям:

- Крайне медленное выполнение запроса при слабой загрузке серверного оборудования. Замедление запроса может быть очень значительным (до нескольких порядков).
- Нестабильная работа запроса. При некоторых условиях запрос может работать достаточно быстро, при других - очень медленно.
- Значительная разница по времени выполнения запроса на разных СУБД.

5.2 Соединения с виртуальными таблицами

Виртуальные таблицы, используемые в языке запросов 1С:Предприятия, могут разворачиваться в подзапросы при трансляции в язык SQL. Это связано с тем, что виртуальная таблица часто (но не всегда) получает данные из нескольких физических таблиц СУБД. Если вы используете соединение с виртуальной таблицей, то на уровне SQL оно может быть в некоторых случаях реализовано, как соединение с подзапросом. В этом случае оптимизатор СУБД может точно так же выбрать неоптимальный план, как при работе с подзапросом, использованным в языке 1С:Предприятия в явном виде.

5.3 Несоответствие индексов и условий запроса

Если в структуре базы данных отсутствует индекс, удовлетворяющий всем перечисленным условиям, то для получения результата СУБД будет

вынуждена сканировать таблицу или один из ее индексов. Это приведет к увеличению времени выполнения запроса, а так же к возможному снижению параллельности системы, поскольку возрастет количество установленных блокировок.

Требования к индексу, перечисленные в рекомендациях, связаны с физической структурой индекса в СУБД. Эта структура представляет собой дерево значений проиндексированных полей. На первом уровне дерева находятся значения первого поля индекса, на втором - второго и так далее. Такая структура позволяет достичь высокой эффективности при поиске по индексу. Кроме того, она гарантирует отсутствие деградации производительности индекса с ростом количества данных.

Однако, индекс такой структуры, очевидно, может быть использован только строго определенным образом. Сначала необходимо провести поиск по значению первого поля индекса, затем - второго и так далее. Если, например, условие по первому полю индекса не указано, то индекс уже не сможет обеспечить быстрый поиск. Если указано условие по нескольким первым полям индекса, а затем одно или несколько полей индекса не задано, то индекс может быть использован только частично.

5.4 Использование логического ИЛИ в условиях

Не следует использовать ИЛИ в секции ГДЕ запроса. Это может привести к тому, что СУБД не сможет использовать индексы таблиц и будет выполнять сканирование, что увеличит время работы запроса и вероятность возникновения блокировок. Вместо этого следует разбить один запрос на несколько и объединить результаты.

5.5 Использование подзапросов в условии соединения

Не следует использовать подзапросы в условии соединения. Это может привести к значительному замедлению запроса и (в отдельных случаях) к его полной неработоспособности на некоторых СУБД.

5.6 Получение данных через точку от полей составного типа

Если в запросе используется получение значения через точку от поля составного ссылочного типа, то при выполнении этого запроса будет выполняться соединение со всеми таблицами объектов, входящими в этот составной тип. В результате SQL текст запроса чрезвычайно усложняется, и при его выполнении оптимизатор СУБД может выбрать неоптимальный план. Это может привести к серьезным проблемам производительности и даже к неработоспособности запроса в отдельных случаях.

5.7 Фильтрация виртуальных таблиц без использования параметров

При использовании виртуальных таблиц в запросах, следует передавать в параметры таблиц все условия, относящиеся к данной виртуальной таблице. Не рекомендуется фильтровать виртуальные таблицы при помощи условий в секции ГДЕ и т.п. Такой запрос будет возвращать правильный (с точки зрения функциональности) результат, но СУБД будет намного сложнее выбрать оптимальный план для его выполнения. В некоторых случаях это может привести к ошибкам оптимизатора СУБД и значительному замедлению работы запроса.

5.8 Выводы по разделу

В данном разделе были рассмотрены неоптимальные ситуации, которые могут возникнуть при выполнении запросов в 1С. Одной из основных причин замедления запросов является использование соединений с подзапросами и виртуальными таблицами. Это может привести к значительному замедлению выполнения запроса, а также к его нестабильной работе на различных СУБД. Также следует обратить внимание на соответствие индексов и условий запроса, чтобы избежать сканирования таблицы или индекса при отсутствии нужного индекса. Использование логического ИЛИ в условиях так-

же может вызывать негативные последствия для производительности системы. Наконец, получение данных через точку от полей составного типа может быть очень трудоемкой операцией и вызывать серьезные проблемы производительности системы.

6 Описание решения

6.1 Создание веб-приложения

Веб-приложение было сделано на JavaScript с библиотекой React [8].

Для функционирования были созданы следующие модули:

6.1.1 Модуль авторизации.

Для авторизации пользователей была создана форма авторизации (рис. 21) с выбором роли. Для каждой роли нужны свои данные входа. Так, для пациента нужно ввести свои СНИЛС и дату рождения, для врача нужно ввести СНИЛС, а для медицинской организации API_KEY.

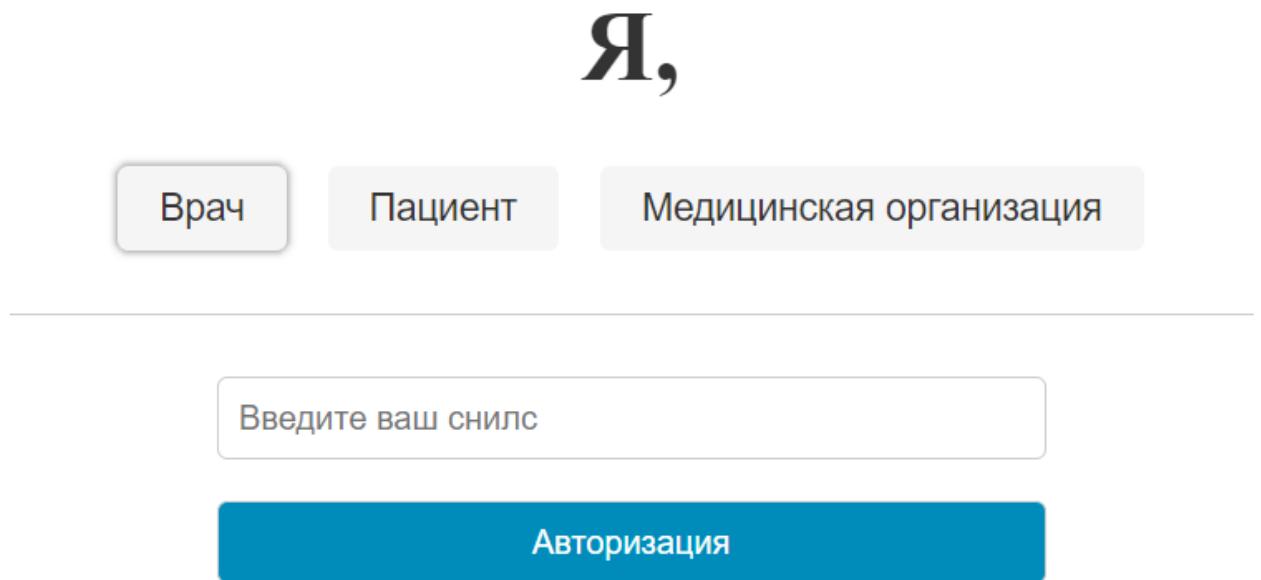


Рисунок 21 – Страница авторизации пользователя

После авторизации пользователю разрешаются действия, зависящие от его роли. Так, для пациента разрешено записываться на прием, смотреть историю записи и смотреть контакты МО. Для врача разрешено смотреть свое расписание, а также записи пациентов на него. Для МО разрешено смотреть расписание врачей.

6.1.2 Модуль выбора сеанса.

Предполагается, что пациент может выбрать специальность врача, а также дату и время записи на прием до или после авторизации (рис. 22). Если

выбор сделан и подтвержден, пользователь будет автоматически записан на прием, если он уже авторизован. Если пользователь не авторизован, ему предложат авторизоваться, и, если он откажется, запись на прием не будет произведена.

Рисунок 22 – Форма записи на прием

После успешной записи пациента на прием, он может отменить запись в любое время до назначенной даты и времени приема. Для этого необходимо авторизоваться на сайте и перейти в раздел “Мои записи”. Там будет доступен список всех назначенных приемов (рис. 23) с возможностью их отмены. В случае отмены записи, свободное время врача будет автоматически освобождено, и другие пользователи смогут записаться на это время.

Рисунок 23 – Блок сделанной записи с возможностью отмены

6.1.3 Модуль профиля.

Для пациента (рис. 25) и врача (рис. 24) сделаны страницы профиля для отображения данных авторизованного пользователя.



Тест Тестовый (Хирург)

Рисунок 24 – Пример профиля врача



Бобров Бобер

Рисунок 25 – Пример профиля пациента

6.3.4 SOAP Модуль

Для получения данных с БД 1С был написан SOAP модуль «SoapRequest». В нем был описан стандартная функция «GetHttpSoapRequest». Она отправляет POST-запрос на заданный API-URL с текстом XML в теле запроса. Она использует метод fetch для запроса и задает нужные заголовки, такие как хост и длину содержимого.

Затем функция обрабатывает ответ с помощью метода “then”, который преобразует тело ответа в текст с помощью метода “text()” и возвращает результат. Если во время запроса произойдет ошибка, функция перехватит эту ошибку и выведет сообщение об ошибке в консоль, а затем вернет пустой результат.

6.3 Модернизация запросов 1С

Были проведены работы по оптимизации запросов из 1С в базу данных, которые привели к оптимизации связанных процессов и значительному ускорению работы базы данных. Также были соблюдены общие рекомендации по написанию кода на 1С [7].

После выполнения работ по оптимизации, запросы стали длиннее, но значительно быстрее. Были проведены исследования оптимизации запросов, улучшены индексы и оптимизированы запросы базы данных. Пример оптимизации запроса «GetSpesialityList» в Приложении В.

Полезно отметить, что такая оптимизация способствует эффективному и более быстрому исполнению запросов приложения и улучшению работы системы в целом.

6.3 Сценарии использования

Конкретные шаги, которые пользователь должен выполнить при использовании сервиса записи на прием, могут выглядеть примерно так:

1. Открыть сайт сервиса записи на прием
2. Выбрать специальность врача, к которому пользователь хочет записаться
3. Ознакомиться со списком доступных врачей в выбранной специальности
4. Выбрать нужного врача из списка
5. Увидеть свободные даты и время приема
6. Выбрать наиболее удобное время и подтвердить свою запись на прием

7. Ввести свой номер телефона и электронную почту для получения уведомления о записи на прием
8. Получить уведомление о том, что запись на прием прошла успешно, а также дополнительную информацию и подсказки для успешной консультации.

Такой ряд действий позволяет пользователям легко записаться на прием не выходя из дома. А сам сервис обеспечивает быстроту и удобство приема, снижает загруженность медицинских учреждений и повышает эффективность предоставляемых услуг.

6.4 Выводы по разделу

В данном разделе было описано создание веб-приложения на JavaScript с использованием библиотеки React. Были созданы модули авторизации, выбора сеанса и профиля для пациента и врача. Также был написан SOAP модуль для получения данных из базы данных 1С. Была проведена работа по оптимизации запросов из 1С, что привело к ускорению работы базы данных и повышению эффективности приложения в целом. В результате выполненных работ удалось решить задачу по записи пациентов на прием к врачам через онлайн-сервис, что значительно облегчит процесс записи на медосмотр как для пациентов, так и для работников МО.

7 Исследование разработанного решения

Результаты оптимизации запросов показали изменение времени выполнения метода до и после проведения оптимизации. Соответствующие значения были представлены в таблице 3, для визуализации различий во времени выполнения.

Проверка оптимальности запросов происходит на следующих объемах данных:

Количество пациентов: 937 686

Количество физических лиц: 937 223

Количество врачей: 138

Количество талонов: 384 008

Количество занятых талонов: 172 756

Таблица 3 – Время выполнения методов до и после оптимизации запросов 1С

| Метод | Время неоптимизированного запроса, мс | Время оптимизированного запроса, мс |
|------------------------|---------------------------------------|-------------------------------------|
| CheckPatient | 467 | 278 |
| GetSpesialityList | 1901 | 1047 |
| GetDoctorList | 1461 | 915 |
| GetAvaibleAppointments | 1333 | 869 |
| SetAppointment | 97 | 89 |
| CreateClaimForRefusal | 104 | 107 |
| GetWorkingTime | 3515 | 851 |
| GetPatientHistory | 464 | 370 |

Из результатов оптимизации запросов для каждого метода следует, что способы оптимизации функционирования значительно снизили время выполнения методов. Наибольший выигрыш во времени выполнения был получен после оптимизации метода GetSpesialityList, что позволило уменьшить время выполнения до 1047 мс с 1901 мс и GetAvaibleAppointments изменением времени выполнения с 3515 мс до 851 мс. Также были заметны значительные улучшения в методах GetDoctorList, GetAvaibleAppointments и

GetWorkingTime, соответственно до 915 мс, 869 мс и 851 мс после оптимизации. В остальных методах выигрыш во времени был менее заметен, но также демонстрируется улучшение во времени выполнения после оптимизации.

Таким образом, оптимизация запросов является эффективным способом улучшения производительности системы. Она может значительно повысить производительность, особенно для методов, которые требуют длительной обработки входных данных. Это позволяет пользователям быстро и эффективно получать доступ к необходимой информации и повышать качество обслуживания.

ЗАКЛЮЧЕНИЕ

Исходя из проделанной работы и результатов исследования, можно сделать вывод, что оптимизация производительности системы является ключевым фактором для улучшения онлайн-сервиса записи на прием в медицинских учреждениях. Оптимизация запросов и тестирование производительности позволили сократить время выполнения методов и значительно повысить качество обслуживания пользователей.

Важно постоянно учитывать потребности и обратную связь пользователей и следить за новыми технологиями и тенденциями в области онлайн-сервисов здравоохранения, чтобы развивать и улучшать систему. Безопасность данных и систематическое обновление и обслуживание системы также очень важны для успешного функционирования онлайн-системы записи на прием.

Таким образом, оптимизация производительности, учет обратной связи пользователей, следование новым технологиям и обеспечение безопасности данных являются ключевыми составляющими современной системы онлайн-сервиса записи на прием в медицинских учреждениях.

Кроме того, в планах внедрения в систему онлайн-сервиса записи на прием в медицинских учреждениях для пользователя “Медицинская организация” - подключение сервиса управления очередями. Это позволит пользователям более эффективно оценивать временные затраты ожидания в очереди и избежать неприятных ситуаций в случае путаницы. Также, в рамках дальнейшего развития СЗПВ, будет добавлена возможность просмотра справок и документов, полученных во время консультации. Это улучшит доступность и полезность системы для пользователей, которые смогут в любое время получить необходимые документы и справки онлайн.

Все вышеупомянутые пункты будут учитываться в дальнейшем развитии и улучшении системы онлайн-сервиса записи на прием в медицинских учреждениях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Оптимизация работы медицинского CALL-центра / М. А. Алинкина, Ю.В. Петриченко // Вестник магистратуры. – 2017. – №7. – С. 27–29.
2. Эльбек Ю.В., Нефедов Р.Д. Приверженность граждан при осуществлении записи к врачу в условиях информатизации здравоохранения // ОРГЗДРАВ: новости, мнения, обучение. Вестник ВШОУЗ. 2022. Т. 8, № 2. С. 146–153. DOI: <https://doi.org/10.33029/2411-8621-2022-8-2-146-153>.
3. API Netrika [Электронный ресурс]. URL: <https://api.n3health.ru/szpv/#solution> (дата обращения: 28.04.2021).
4. Методические рекомендации N 12-22 "Организация записи на прием к врачу, в том числе через единый портал государственных и муниципальных услуг и единые региональные колл-центры" [Электронный ресурс]. URL: <https://www.garant.ru/products/ipo/prime/doc/405772431/#61> (дата обращения: 28.04.2021).
5. Реестр справочников НСИ [Электронный ресурс]. <https://nsi.rosminzdrav.ru/>
6. Причины неоптимальной работы запросов и методы оптимизации [Электронный ресурс]. <https://its.1c.ru/db/metod8dev/content/5842/hdoc>
7. Инструкции по разработке на 1С [Электронный ресурс]. <https://its.1c.ru>
8. Документация React JS [Электронный ресурс]. <https://ru.react.js.org>
9. Сервис телемедицинских консультаций [Электронный ресурс]. <https://gosteamed.ru/>
10. СЗПВ Здоровье Петербуржца [Электронный ресурс]. <https://gorzdrav.spb.ru/>
11. СЗПВ Яндекс.Здоровье [Электронный ресурс]. <https://health.yandex.ru/>
12. СЗПВ ГосУслуги [Электронный ресурс]. <https://gosuslugi.ru/>
13. СЗПВ СберЗдоровье [Электронный ресурс]. <https://spb.docdoc.ru/>

ПРИЛОЖЕНИЕ А

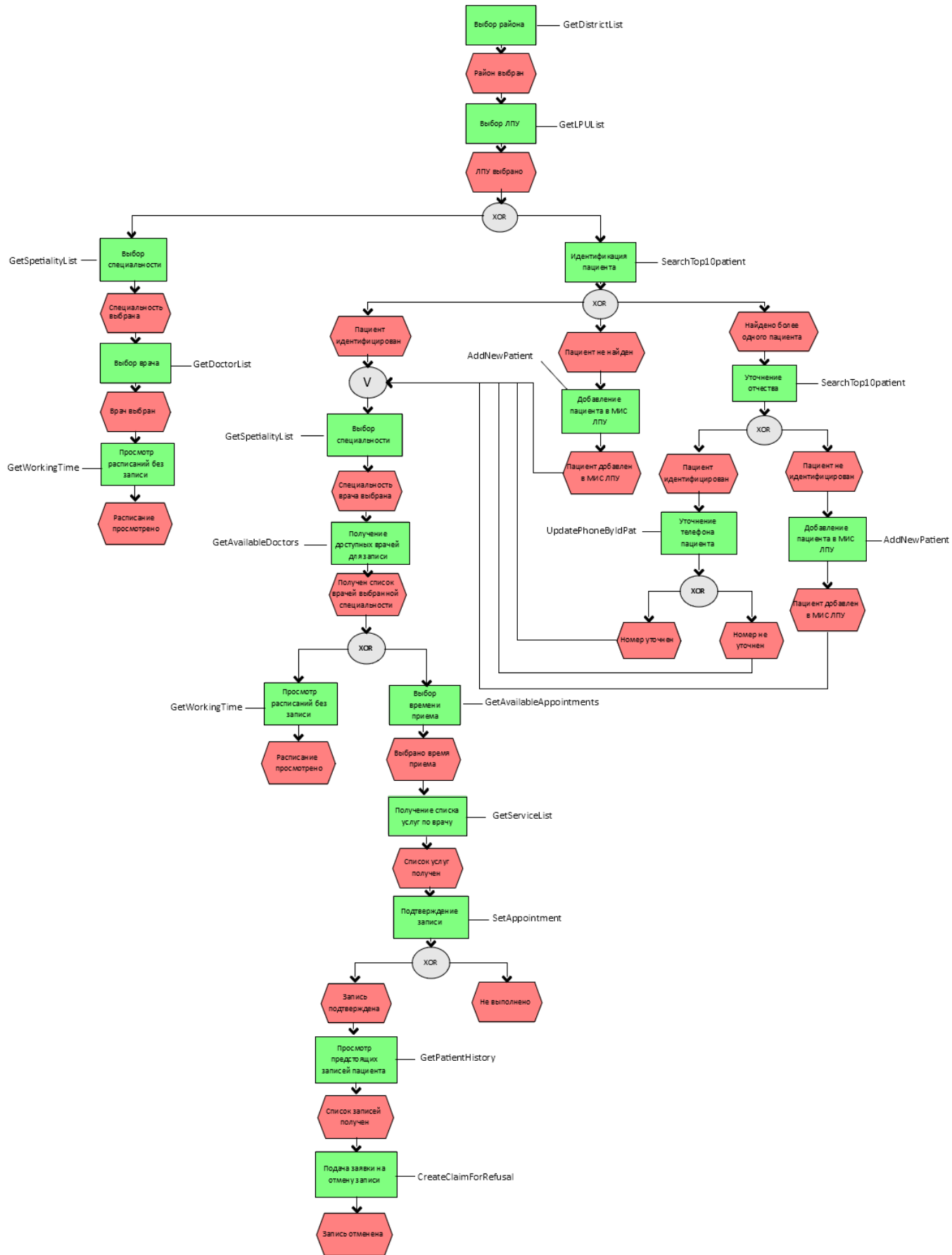


Рисунок 26 – Схема СЗПВ

ПРИЛОЖЕНИЕ Б

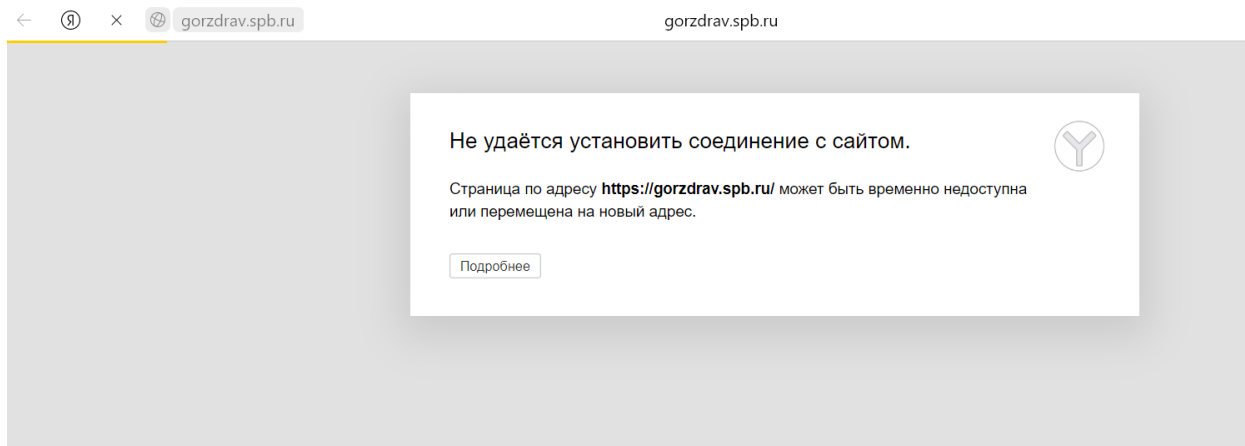


Рисунок 27 – Пример проблем СЗПВ на веб-сервисе **gorzdrav.spb.ru** на 12.05.2023

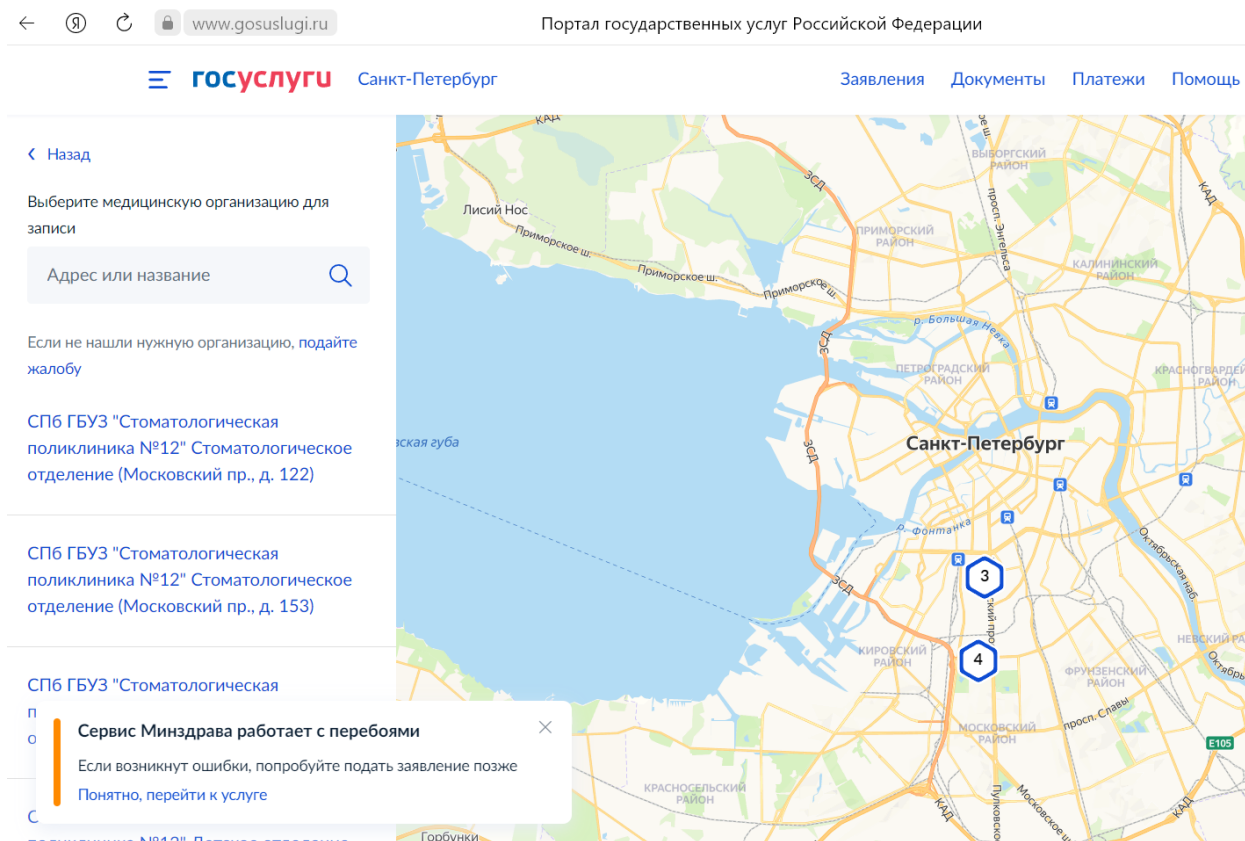


Рисунок 28 – Пример проблем СЗПВ на веб-сервисе **gosuslugi.ru** на 12.05.2023

ПРИЛОЖЕНИЕ В

Пример старого запроса данных для метода GetSpecialityList:

ВЫБРАТЬ

ЖурналЗаписиНаПриемСрезПоследних.СеансЗаписи КАК СеансЗаписи,
ЖурналЗаписиНаПриемСрезПоследних.Записан КАК Записан

ПОМЕСТИТЬ ВтЖурналЗаписиСрезПоследних

ИЗ

РегистрСведений.ЖурналЗаписиНаПрием.СрезПоследних (, СеансЗапи-
си.Начало >= &Начало) КАК ЖурналЗаписиНаПриемСрезПоследних
;

////////////////////////////////////
////////

ВЫБРАТЬ

Сотрудники.Специальность КАК Специальность,
КОЛИЧЕСТВО (РАЗЛИЧНЫЕ СеансыОказанияУслуг.Ссылка) КАК ОбщееКоличе-
ствоТалонов,

МАКСИМУМ (СеансыОказанияУслуг.Начало) КАК МаксимальнаяДата,
МИНИМУМ (СеансыОказанияУслуг.Начало) КАК МинимальнаяДата

ПОМЕСТИТЬ ВтСвободныеТалоны

ИЗ

Справочник.СеансыОказанияУслуг КАК СеансыОказанияУслуг
ВНУТРЕННЕЕ СОЕДИНЕНИЕ Справоч-
ник.СеансыОказанияУслуг.ДополнительныеРеквизиты КАК СеансыОказа-
нияУслугДополнительныеРеквизиты

ПО СеансыОказанияУслуг.Ссылка = СеансыОказанияУслугДополни-
тельныеРеквизиты.Ссылка

И (СеансыОказанияУслугДополнительныеРеквизи-
ты.Свойство.Имя = &СеансВидимостьВЗПВ)

И (НЕ СеансыОказанияУслугДополнительныеРеквизи-
ты.Свойство.ПометкаУдаления)

ВНУТРЕННЕЕ СОЕДИНЕНИЕ Справочник.Сотрудники КАК Сотрудники
ВНУТРЕННЕЕ СОЕДИНЕНИЕ Справоч-
ник.Сотрудники.ДополнительныеРеквизиты КАК СотрудникиДополнительные-
Реквизиты

ПО Сотрудники.Ссылка = СотрудникиДополнительныеРекви-
зиты.Ссылка

И (СотрудникиДополнительныеРеквизиты.Свойство.Имя
 = &СотрудникВидимостьВЗПВ)
 И (НЕ СотрудникиДополнительныеРеквизи-
 ты.Свойство.ПометкаУдаления)
 ВНУТРЕННЕЕ СОЕДИНЕНИЕ РегистрСведе-
 ний.РЕГИЗ_НСИ_Трансляция КАК Трансляция_Специальность
 ПО Сотрудники.Специальность = Трансля-
 ция_Специальность.ЭлементМИС
 И (Трансляция_Специальность.Справочник =
 &РегизСпециальность)
 И (НЕ Трансляция_Специальность.ЭлементРЕГИЗ =
 ЗНАЧЕНИЕ (Справочник.Региз_НСИ_ЭлементыСправочников.ПустаяСсылка))
 ВНУТРЕННЕЕ СОЕДИНЕНИЕ РегистрСведе-
 ний.РЕГИЗ_НСИ_Трансляция КАК Трансляция_Должность
 ПО Сотрудники.Должность = Трансля-
 ция_Должность.ЭлементМИС
 И (Трансляция_Должность.Справочник =
 &РегизДолжность)
 И (НЕ Трансляция_Должность.ЭлементРЕГИЗ = ЗНАЧЕ-
 НИЕ (Справочник.Региз_НСИ_ЭлементыСправочников.ПустаяСсылка))
 ВНУТРЕННЕЕ СОЕДИНЕНИЕ РегистрСведе-
 ний.РЕГИЗ_ЗПВ_КодыПодразделений КАК РЕГИЗ_ЗПВ_КодыПодразделений
 ПО Сотрудники.Подразделение = РЕ-
 ГИЗ_ЗПВ_КодыПодразделений.Подразделение
 ПО СеансыОказанияУслуг.Врач = Сотрудники.Ссылка
 ЛЕВОЕ СОЕДИНЕНИЕ ВтЖурналЗаписиСрезПоследних КАК ВтЖурналЗа-
 писиСрезПоследних
 ПО СеансыОказанияУслуг.Ссылка = ВтЖурналЗаписиСрезПослед-
 них.СеансЗаписи
 ГДЕ
 НЕ Сотрудники.Должность = ЗНАЧЕ-
 НИЕ (Справочник.Должности.ПустаяСсылка)
 И НЕ Сотрудники.Подразделение = ЗНАЧЕ-
 НИЕ (Справочник.Подразделения.ПустаяСсылка)
 И РЕГИЗ_ЗПВ_КодыПодразделений.ВходнойКод = &ИдОрганизации
 И СеансыОказанияУслугДополнительныеРеквизиты.Значение = ИСТИНА
 И СотрудникиДополнительныеРеквизиты.Значение = ИСТИНА
 И СеансыОказанияУслуг.ПометкаУдаления = ЛОЖЬ
 И Сотрудники.ПометкаУдаления = ЛОЖЬ

```

И НЕ Сотрудники.ФизическоеЛицо.СтраховойНомерПФР = ""
И СеансыОказанияУслуг.Начало > &Начало
И (ВтЖурналЗаписиСрезПоследних.Записан = ЛОЖЬ
      ИЛИ ВтЖурналЗаписиСрезПоследних.Записан ЕСТЬ NULL)
И НЕ Сотрудники.Подразделение = ЗНАЧЕ-
НИЕ (Справочник.Подразделения.ПустаяСсылка)
{ГДЕ
      СеансыОказанияУслуг.Начало КАК ДатаПо}

СТРУППИРОВАТЬ ПО
      Сотрудники.Специальность
;

////////////////////////////////////
/////////

ВЫБРАТЬ
      Сотрудники.Специальность КАК Специальность,
      РЕГИЗ_НСИ_ЭлементыСпециальность.РЕГИЗ_Код КАК КодФед
ПОМЕСТИТЬ ВтСпециальности
ИЗ
      Справочник.Сотрудники КАК Сотрудники
      ВНУТРЕННЕЕ СОЕДИНЕНИЕ Справоч-
ник.Сотрудники.ДополнительныеРеквизиты КАК СотрудникиДополнительные-
Реквизиты
      ПО Сотрудники.Ссылка = СотрудникиДополнительныеРеквизи-
ты.Ссылка
      И (СотрудникиДополнительныеРеквизиты.Свойство.Имя =
&СотрудникВидимостьВЗПВ)
      И (НЕ СотрудникиДополнительныеРеквизи-
ты.Свойство.ПометкаУдаления)
      ВНУТРЕННЕЕ СОЕДИНЕНИЕ РегистрСведений.РЕГИЗ_НСИ_Трансляция
КАК ТрансляцияДолжность
      ВНУТРЕННЕЕ СОЕДИНЕНИЕ Справоч-
ник.РЕГИЗ_НСИ_ЭлементыСправочников КАК РЕГИЗ_НСИ_ЭлементДолжность
      ПО ТрансляцияДолжность.ЭлементРЕГИЗ = РЕ-
ГИЗ_НСИ_ЭлементДолжность.Ссылка
      ПО Сотрудники.Должность = ТрансляцияДолжность.ЭлементМИС
      И (ТрансляцияДолжность.Справочник = &РегизДолжность)

```

И (НЕ ТрансляцияДолжность.ЭлементРЕГИЗ = ЗНАЧЕНИЕ (Справочник.Региз_НСИ_ЭлементыСправочников.ПустаяСсылка))

ВНУТРЕННЕЕ СОЕДИНЕНИЕ РегистрСведений.РЕГИЗ_НСИ_Трансляция
КАК ТрансляцияСпециальность

ВНУТРЕННЕЕ СОЕДИНЕНИЕ Справочник.РЕГИЗ_НСИ_ЭлементыСправочников КАК РЕГИЗ_НСИ_ЭлементыСпециальность

ПО ТрансляцияСпециальность.ЭлементРЕГИЗ = РЕГИЗ_НСИ_ЭлементыСпециальность.Ссылка

ПО Сотрудники.Специальность = ТрансляцияСпециальность.ЭлементМИС

И (ТрансляцияСпециальность.Справочник = &РегизСпециальность)

И (НЕ ТрансляцияСпециальность.ЭлементРЕГИЗ = ЗНАЧЕНИЕ (Справочник.Региз_НСИ_ЭлементыСправочников.ПустаяСсылка))

ВНУТРЕННЕЕ СОЕДИНЕНИЕ РегистрСведений.РЕГИЗ_ЗПВ_КодыПодразделений КАК РЕГИЗ_ЗПВ_КодыПодразделений

ПО Сотрудники.Подразделение = РЕГИЗ_ЗПВ_КодыПодразделений.Подразделение

ГДЕ

НЕ Сотрудники.Должность = ЗНАЧЕНИЕ (Справочник.Должности.ПустаяСсылка)

И РЕГИЗ_ЗПВ_КодыПодразделений.ВходнойКод = &ИдОрганизации

И НЕ Сотрудники.Специальность = ЗНАЧЕНИЕ (Справочник.Специальности.ПустаяСсылка)

И Сотрудники.ПометкаУдаления = ЛОЖЬ

И НЕ Сотрудники.ФизическоеЛицо.СтраховойНомерПФР = ""

И НЕ Сотрудники.Подразделение = ЗНАЧЕНИЕ (Справочник.Подразделения.ПустаяСсылка)

И СотрудникиДополнительныеРеквизиты.Значение = ИСТИНА

СТРУППИРОВАТЬ ПО

Сотрудники.Специальность,

РЕГИЗ_НСИ_ЭлементыСпециальность.РЕГИЗ_Код

;

////////////////////////////////////
////////

ВЫБРАТЬ

ЕСТЬNULL (ВтСвободныеТалоны.МаксимальнаяДата, НЕОПРЕДЕЛЕНО) КАК
МаксимальнаяДата,
ЕСТЬNULL (ВтСвободныеТалоны.МинимальнаяДата, НЕОПРЕДЕЛЕНО) КАК Ми-
нимальнаяДата,
ВтСпециальности.Специальность КАК Специальность,
ЕСТЬNULL (ВтСвободныеТалоны.ОбщееКоличествоТалонов, 0) КАК Количе-
ствоСвободныхТалоновВсего,
ВтСпециальности.КодФед КАК КодФед
ИЗ
ВтСпециальности КАК ВтСпециальности
ЛЕВОЕ СОЕДИНЕНИЕ ВтСвободныеТалоны КАК ВтСвободныеТалоны
ПО ВтСпециальности.Специальность = ВтСвободныеТало-
ны.Специальность

Пример нового запроса данных для метода GetSpecialityList:

ВЫБРАТЬ

Сотрудники.Ссылка КАК Врач,

Сотрудники.ФизическоеЛицо КАК ФизическоеЛицо,

ТрансляцияДолжность.ЭлементРЕГИЗ.Код КАК ДолжностьРЕГИЗ,

ТрансляцияСпециальность.ЭлементРЕГИЗ.Код КАК Специа-
льностьРЕГИЗ,

Сотрудники.ФизическоеЛицо.СтраховойНомерПФР КАК СНИЛС,

РЕГИЗ_ЗПВ_КодыПодразделений.ВходнойКод КАК ВходнойКод,

РЕГИЗ_ЗПВ_КодыПодразделений.ВыходнойКод КАК ВыходнойКод,

РЕГИЗ_ЗПВ_КодыПодразделений.ИдентификаторУчреждения КАК
ИдентификаторУчреждения,

Сотрудники.Должность КАК Должность,

Сотрудники.Подразделение КАК Подразделение,

Сотрудники.Специальность КАК Специальность,

ЕСТЬNULL (СотрудникиДопРеквОтображение.Значение, НЕОПРЕДЕЛЕ-
НО) КАК ПредставлениеВЗПВ,

СотрудникиДопРеквКомментарий.Значение КАК КомментарийВЗПВ

ПОМЕСТИТЬ ВОтборСотрудники

ИЗ

Справочник.Сотрудники КАК Сотрудники

ВНУТРЕННЕЕ СОЕДИНЕНИЕ Справоч-
ник.Сотрудники.ДополнительныеРеквизиты КАК СотрудникиДопРеквВидимость

ПО Сотрудники.Ссылка = СотрудникиДопРеквВиди-
мость.Ссылка

И (СотрудникиДопРеквВидимость.Свойство =
&СотрудникВидимостьВЗПВ)

И (НЕ СотрудникиДопРеквВиди-
мость.Свойство.ПометкаУдаления)

ВНУТРЕННЕЕ СОЕДИНЕНИЕ РегистрСведе-
ний.РЕГИЗ_НСИ_Трансляция КАК ТрансляцияДолжность

ПО Сотрудники.Должность = ТрансляцияДолж-
ность.ЭлементМИС

И (ТрансляцияДолжность.Справочник =
&РегизДолжность)

И (НЕ ТрансляцияДолжность.ЭлементРЕГИЗ = ЗНАЧЕ-
НИЕ (Справочник.Региз_НСИ_ЭлементыСправочников.ПустаяСсылка))

ВНУТРЕННЕЕ СОЕДИНЕНИЕ РегистрСведе-
ний.РЕГИЗ_НСИ_Трансляция КАК ТрансляцияСпециальность

ПО Сотрудники.Специальность = ТрансляцияСпециаль-
ность.ЭлементМИС

И (ТрансляцияСпециальность.Справочник =
&РегизСпециальность)

И (НЕ ТрансляцияСпециальность.ЭлементРЕГИЗ = ЗНАЧЕНИЕ (Справочник.Региз_НСИ_ЭлементыСправочников.ПустаяСсылка))

ВНУТРЕННЕЕ СОЕДИНЕНИЕ РегистрСведений.РЕГИЗ_ЗПВ_КодыПодразделений КАК РЕГИЗ_ЗПВ_КодыПодразделений

ПО Сотрудники.Подразделение = РЕГИЗ_ЗПВ_КодыПодразделений.Подразделение

ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Сотрудники.ДополнительныеРеквизиты КАК СотрудникиДопРеквОтображение

ПО Сотрудники.Ссылка = СотрудникиДопРеквОтображение.Ссылка

И (НЕ СотрудникиДопРеквОтображение.Свойство.ПометкаУдаления)

И (СотрудникиДопРеквОтображение.Свойство &ПредставлениеВрачаЗПВ) =

ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Сотрудники.ДополнительныеРеквизиты КАК СотрудникиДопРеквКомментарий

ПО Сотрудники.Ссылка = СотрудникиДопРеквКомментарий.Ссылка

И (НЕ СотрудникиДопРеквКомментарий.Свойство.ПометкаУдаления)

И (СотрудникиДопРеквКомментарий.Свойство &ВрачКомментарийЗПВ) =

ГДЕ

НЕ Сотрудники.Должность = ЗНАЧЕНИЕ (Справочник.Должности.ПустаяСсылка)

И НЕ Сотрудники.Специальность = ЗНАЧЕНИЕ (Справочник.Специальности.ПустаяСсылка)

И Сотрудники.ПометкаУдаления = ЛОЖЬ

И НЕ Сотрудники.ФизическоеЛицо.СтраховойНомерПФР = ""

И НЕ Сотрудники.Подразделение = ЗНАЧЕНИЕ (Справочник.Подразделения.ПустаяСсылка)

И СотрудникиДопРеквВидимость.Значение = ИСТИНА

И НЕ ТрансляцияСпециальность.ЭлементРЕГИЗ = ЗНАЧЕНИЕ (Справочник.РЕГИЗ_НСИ_ЭлементыСправочников.ПустаяСсылка)

И НЕ ТрансляцияДолжность.ЭлементРЕГИЗ = ЗНАЧЕНИЕ (Справочник.РЕГИЗ_НСИ_ЭлементыСправочников.ПустаяСсылка)

{ ГДЕ

Сотрудники.Должность.*,

Сотрудники.Организация.*,

Сотрудники.Подразделение.*,

Сотрудники.Специальность.*,

РЕГИЗ_ЗПВ_КодыПодразделений.ВходнойКод,

РЕГИЗ_ЗПВ_КодыПодразделений.ВыходнойКод,

РЕГИЗ_ЗПВ_КодыПодразделений.ИдентификаторУчреждения}

ИНДЕКСИРОВАТЬ ПО

Врач,

ФизическоеЛицо,

Специальность

;

////////////////////////////////////

////////////////////////////////////

ВЫБРАТЬ

ФИОФизическихЛицСрезПоследних.ФизическоеЛицо КАК Физическое-
Лицо,

ФИОФизическихЛицСрезПоследних.Фамилия КАК Фамилия,

ФИОФизическихЛицСрезПоследних.Имя КАК Имя,

ФИОФизическихЛицСрезПоследних.Отчество КАК Отчество

ПОМЕСТИТЬ ВФИОФизическихЛиц

ИЗ

РегистрСведений.ФИОФизическихЛиц.СрезПоследних (

,

ФизическоеЛицо В

(ВЫБРАТЬ

ВТОтборСотрудники.ФизическоеЛицо КАК
ФизическоеЛицо

ИЗ

ВТОтборСотрудники КАК ВТОтборСотрудни-
ки)) КАК ФИОФизическихЛицСрезПоследних

ИНДЕКСИРОВАТЬ ПО

ФизическоеЛицо

;

////////////////////////////////////
////////////////////////////////////

ВЫБРАТЬ

ВТОтборСотрудники.Врач КАК Врач,
 ВТОтборСотрудники.ФизическоеЛицо КАК ФизическоеЛицо,
 ВТОтборСотрудники.ДолжностьРЕГИЗ КАК ДолжностьРЕГИЗ,
 ВТОтборСотрудники.СпециальностьРЕГИЗ КАК СпециальностьРЕГИЗ,
 ВТОтборСотрудники.СНИЛС КАК СНИЛС,
 ВТОтборСотрудники.ВходнойКод КАК ВходнойКод,
 ВТОтборСотрудники.ВыходнойКод КАК ВыходнойКод,
 ВТОтборСотрудники.ИдентификаторУчреждения КАК ИдентификаторУчреждения,
 ВТОтборСотрудники.Должность КАК Должность,
 ВТОтборСотрудники.Подразделение КАК Подразделение,
 ВТОтборСотрудники.Специальность КАК Специальность,
 ВТФИОФизическихЛиц.Фамилия КАК Фамилия,
 ВТФИОФизическихЛиц.Имя КАК Имя,
 ВТФИОФизическихЛиц.Отчество КАК Отчество

ПОМЕСТИТЬ ВтСотрудники

ИЗ

ВТОтборСотрудники КАК ВТОтборСотрудники

ВНУТРЕННЕЕ СОЕДИНЕНИЕ ВТФИОФизическихЛиц КАК ВТФИОФизическихЛиц

ПО ВТОтборСотрудники.ФизическоеЛицо = ВТФИОФизическихЛиц.ФизическоеЛицо

ИНДЕКСИРОВАТЬ ПО

ФизическоеЛицо,

Врач,

Специальность

;

////////////////////////////////////
////////////////////////////////////

ВЫБРАТЬ

ЖурналЗаписиНаПриемСрезПоследних.СеансЗаписи КАК СеансЗапи-
си,

ЖурналЗаписиНаПриемСрезПоследних.Записан КАК Записан

ПОМЕСТИТЬ ВЖурналЗаписиСрезПоследних

ИЗ

РегистрСведений.ЖурналЗаписиНаПрием.СрезПоследних (

,

СеансЗаписи.Начало >= &Начало

И (Записан = ЛОЖЬ

ИЛИ Записан ЕСТЬ NULL)) КАК ЖурналЗа-
писиНаПриемСрезПоследних

ИНДЕКСИРОВАТЬ ПО

СеансЗаписи

;

////////////////////////////////////
////////////////////////////////////

ВЫБРАТЬ

ВтСотрудники.Специальность КАК Специальность,

ВтСотрудники.СпециальностьРЕГИЗ КАК КодФед

ПОМЕСТИТЬ ВтСпециальности

ИЗ

ВтСотрудники КАК ВтСотрудники

СГРУППИРОВАТЬ ПО

ВтСотрудники.Специальность,

ВтСотрудники.СпециальностьРЕГИЗ

ИНДЕКСИРОВАТЬ ПО

Специальность

;

////////////////////////////////////
////////////////////////////////////

ВЫБРАТЬ

ВтСотрудники.Специальность КАК Специальность,

КОЛИЧЕСТВО (РАЗЛИЧНЫЕ СеансыОказанияУслуг.Ссылка) КАК Общее-
КоличествоТалонов,

МАКСИМУМ (СеансыОказанияУслуг.Начало) КАК МаксимальнаяДата,

МИНИМУМ (СеансыОказанияУслуг.Начало) КАК МинимальнаяДата

ПОМЕСТИТЬ ВтСвободныеТалоны

ИЗ

Справочник.СеансыОказанияУслуг КАК СеансыОказанияУслуг

ВНУТРЕННЕЕ СОЕДИНЕНИЕ Справоч-
ник.СеансыОказанияУслуг.ДополнительныеРеквизиты КАК СеансыОказа-
нияУслугДополнительныеРеквизиты

ПО СеансыОказанияУслуг.Ссылка = СеансыОказанияУслугДо-
полнительныеРеквизиты.Ссылка

И (СеансыОказанияУслугДополнительныеРеквизи-
ты.Свойство = &СеансВидимостьВЗПВ)

И (НЕ СеансыОказанияУслугДополнительныеРеквизи-
ты.Свойство.ПометкаУдаления)

ЛЕВОЕ СОЕДИНЕНИЕ ВтЖурналЗаписиСрезПоследних КАК
ВтЖурналЗаписиСрезПоследних

ПО СеансыОказанияУслуг.Ссылка = ВтЖурналЗаписиСрезПо-
следних.СеансЗаписи

ВНУТРЕННЕЕ СОЕДИНЕНИЕ ВтСотрудники КАК ВтСотрудники

ПО СеансыОказанияУслуг.Врач = ВтСотрудники.Врач

ГДЕ

СеансыОказанияУслугДополнительныеРеквизиты.Значение = ИСТИНА

И СеансыОказанияУслуг.ПометкаУдаления = ЛОЖЬ

И СеансыОказанияУслуг.Начало > &Начало

{ГДЕ

СеансыОказанияУслуг.Начало КАК ДатаПо}

СГРУППИРОВАТЬ ПО

ВтСотрудники.Специальность

ИНДЕКСИРОВАТЬ ПО


```

        Специальность

;

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

        ВЫБРАТЬ

                ЕСТЬNULL (ВтСвободныеТалоны.МаксимальнаяДата,      НЕОПРЕДЕЛЕНО)
КАК МаксимальнаяДата,

                ЕСТЬNULL (ВтСвободныеТалоны.МинимальнаяДата,      НЕОПРЕДЕЛЕНО)
КАК МинимальнаяДата,

                ВтСпециальности.Специальность КАК Специальность,

                ЕСТЬNULL (ВтСвободныеТалоны.ОбщееКоличествоТалонов,    0)    КАК
КоличествоСвободныхТалоновВсего,

                ВтСпециальности.КодФед КАК КодФед

        ИЗ

                ВтСпециальности КАК ВтСпециальности

                ЛЕВОЕ СОЕДИНЕНИЕ ВтСвободныеТалоны КАК ВтСвободныеТа-
ЛОНЫ

                ПО ВтСпециальности.Специальность = ВтСвободныеТало-
ны.Специальность

```