

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
Кафедра программного обеспечения

РЕКОМЕНДОВАНО К ЗАЩИТЕ В ГЭК

Заведующий кафедрой, к.т.н, доцент

М. С. Воробьева

02.07. 2021 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
магистерская диссертация

**ПОДСИСТЕМА АВТОМАТИЗИРОВАННОЙ КОРРЕКТИРОВКИ
ГРАФИКА ПРИЕМА ПАЦИЕНТОВ С ИСПОЛЬЗОВАНИЕМ ИМИТАЦИИ
ГОЛОСОВОГО ВЗАИМОДЕЙСТВИЯ**

02.04.03 Математическое обеспечение и администрирование
информационных систем

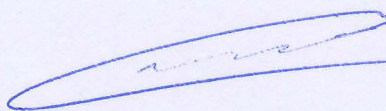
Магистерская программа «Разработка технологий Интернета вещей и
больших данных»

Выполнил работу
студент 2 курса
очной формы обучения



Дзябенко Егор
Анатольевич

Научный руководитель
доцент



Сальников Дмитрий
Евгеньевич

Рецензент
д.т.н., профессор, заведующий кафедрой
информационных систем



Глухих Игорь
Николаевич

Тюмень
2021

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛАВА 1. ИЗУЧЕНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.....	7
1.1. ПРОЦЕСС ЗАПИСИ НА ПРИЕМ И ОТМЕНЫ ПРИЕМА	7
1.2. АЛЬТЕРНАТИВНЫЕ СРЕДСТВА ВЗАИМОДЕЙСТВИЯ.....	11
1.3. ПРИМЕНЕНИЕ ГОЛОСОВЫХ РОБОТОВ	13
1.4. СУЩЕСТВУЮЩИЕ РЕШЕНИЯ.....	16
ГЛАВА 2. ПОДХОД К РЕШЕНИЮ	19
2.1. АНАЛИЗ ДАННЫХ.....	19
2.2. ВЗАИМОСВЯЗЬ ПРОГРАММНЫХ СИСТЕМ.....	28
2.3. ИНФОРМАЦИОННАЯ СИСТЕМА 1С.....	31
2.4. IP-ТЕЛЕФОНИЯ ASTERISK.....	32
2.5. ТЕХНОЛОГИЯ YANDEX SPEECHKIT	34
ГЛАВА 3. РЕАЛИЗАЦИЯ СИСТЕМЫ	35
3.1. ОЧЕРЕДЬ ЗАПЛАНИРОВАННЫХ ЗВОНКОВ	35
3.2. ФОРМИРОВАНИЕ ОЧЕРЕДИ ЗВОНКОВ.....	36
3.3. ИНИЦИАЦИЯ ЗАПЛАНИРОВАННЫХ ЗВОНКОВ	38
3.4. СОЗДАНИЕ АРІ ДЛЯ ИНИЦИАЦИИ ЗВОНКА ИЗ 1С.....	40
3.5. РЕАЛИЗАЦИЯ ГОЛОСОВОГО ВЗАИМОДЕЙСТВИЯ	41
3.6. ОБРАБОТКА РЕЗУЛЬТАТА ЗВОНКА.....	46
ЗАКЛЮЧЕНИЕ	52
СПИСОК ЛИТЕРАТУРЫ	54
ПРИЛОЖЕНИЕ 1-5.....	56

ВВЕДЕНИЕ

В настоящей квалификационной работе рассматривается неявка пациента, как причина неэффективного использования трудовых ресурсов врачей и необходимости оптимизации деятельности call-центра медицинского учреждения (далее Предприятие).

На Предприятии посещение врача происходит либо в порядке живой очереди, либо по записи. В последнем случае возможна ситуация, когда пациент по какой-либо причине не может посетить врача, но не сообщает об этом, и при наступлении времени приема, оказывается, что пациент не явился. Это приводит к неэффективному использованию врачей, поскольку на освободившееся место можно было записать другого пациента, если бы удалось об этом узнать заранее, а также, в целом сократить очередь на запись к врачу.

В результате анализа данных удалось установить следующее. Статистика об оказанных и отмененных услугах на Предприятии за первую половину 2021 года показала, что каждый день, в среднем, по 43 приема отменяется, в том числе 35 отменяются из-за отказа пациента от записи, 5 переносятся на другое время и 3 отменяется из-за неявки пациента. Зная, что каждый день в среднем выполняется 273 приема (Рис. 2), можно сказать, что приблизительно по 14% приемов не выполняются из-за того, что не удалось вовремя узнать придет пациент на прием или нет.

На Предприятии решение этой задачи осуществляется силами call-центра, чья обязанность – уточнять у пациентов за день до приема, придут они или нет. Если пациент отказывается, тогда его запись освобождается, следовательно, можно записать на это время другого пациента (Рис. 1).

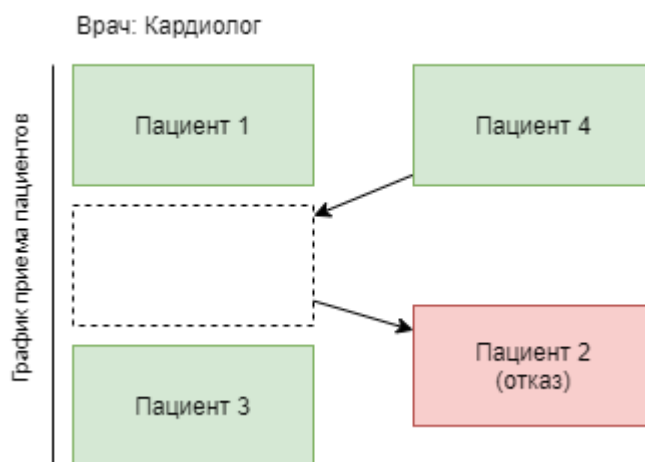


Рис. 1. Принцип актуализации графика приема пациентов

На первый взгляд проблема решена, но при наличии 316 записей на прием и средней продолжительности звонка в 2 минуты, для обзвона всех пациентов необходимо более 10 часов, что выходит за рамки восьмичасового рабочего дня. Этот факт является причиной того, что сотрудник call-центра не может выполнить обзвон всех пациентов в течение рабочего дня и сформировать уточненный график приема врачей. Для решения проблемы Предприятию предложена разработка и дальнейшее внедрение вспомогательной системы автоматической корректировки.

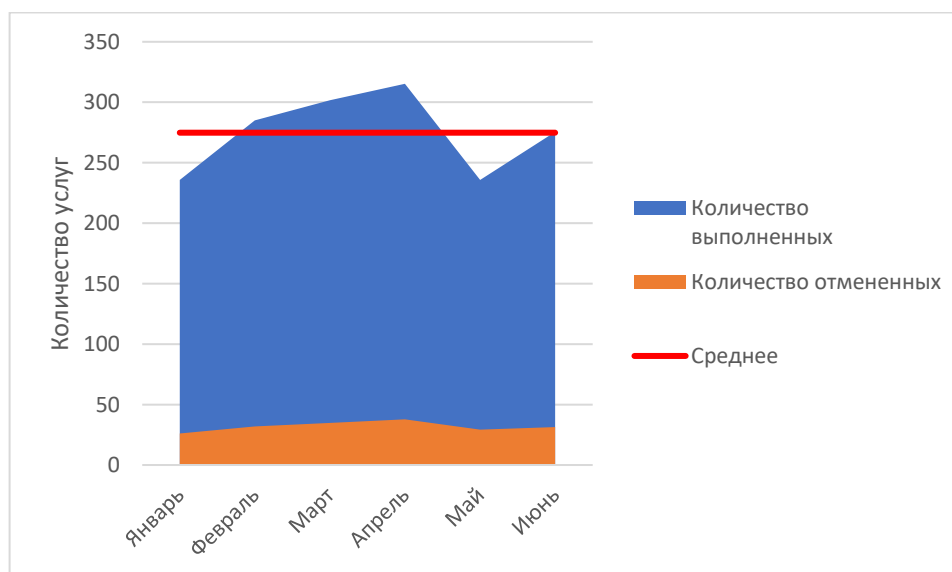


Рис. 2. Количество услуг за первую половину 2021 г.

Для автоматизации рутинной работы сотрудников call-центра в последнее время широко применяются средства автоматизированного голосового взаимодействия, в которых робот имитирует речь человека, может реагировать на ответы человека и действовать в соответствии с ними. В некоторых случаях удается достичь хороших результатов: «в торговой сети «Пятёрочка» уже до 50% всех обращений по вопросам лояльности решает робот. Конверсия в целевое действие — более 60%, а затраты на бота — в 5-7 раз ниже, чем расходы на зарплату операторов» [3].

Цель работы – разработать систему для автоматического определения и освобождения неиспользуемых записей на прием к врачу в медицинском учреждении.

Задачи:

- выполнить анализ данных;
- на основании анализа данных определить приоритет и правила обзвона пациентов;
- спроектировать систему голосового взаимодействия;
- реализовать систему автоматического голосового взаимодействия и корректировки графика приемов.

Решение поставленных задач будет выполняться с использованием следующих программных средств:

- программа 1С:Предприятие (далее 1С);
- IP-телефония Asterisk;
- сервис синтеза и распознавания речи – Yandex SpeechKit.

Для подготовки и защиты выпускной квалификационной работы использовались: поиск, анализ информации, системный подход для решения поставленных задач, приемы критического анализа проблемных ситуаций, а также средства и методы саморазвития и самореализации; методики

межкультурного взаимодействия, умение расставлять приоритеты собственной деятельности.

Формулирование выводов по итогам проведенной работы осуществлялись с учетом применения современных коммуникативных технологий (в том числе на иностранном языке) для представления результатов на академических, профессиональных, экспертных ИТ-мероприятиях.

ГЛАВА 1. ИЗУЧЕНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. ПРОЦЕСС ЗАПИСИ НА ПРИЕМ И ОТМЕНЫ ПРИЕМА

Как в большинстве медицинских учреждений, на Предприятии приемы пациентов выполняются как в порядке живой очереди, так и в порядке предварительной записи.

Основной канал для записи на прием – входящий телефонный звонок от пациента. В процессе разговора определяется к какому врачу и на какое время требуется запись. После получения этой информации необходимо проверить ведется ли прием врача в указанное время и есть ли в наличии свободное место для записи.

Для просмотра расписания приемов врача в учетной системе Предприятия есть необходимые инструменты, позволяющие увидеть загруженность врача в виде таблицы с ячейками, окрашенными в различные цвета, отображающие загруженность расписания врача и доступные места для записи (Рис. 3). Оранжевым цветом отмечены свободные места, черным и розовым – занятые. Различие цветов для занятых мест обосновано дополнительной классификацией для удобства пользователя.

В случае, если на выбранное пациентом время нет свободных мест, оператор изменяет период отображения расписания (вплоть до одного месяца вперед) и предоставляет пациенту на выбор ближайшие свободные даты приема.

Когда дата и время приема согласованы, сотрудник call-центра совершает двойной клик по соответствующей ячейке, чтобы занять выбранное место для записи. Для последующей идентификации пациента при его посещении, сотрудник call-центра получает контактную информацию пациента и его ФИО. После завершения заполнения контактной информации оператор нажимает «Записать и закрыть», чтобы завершить запись на прием.

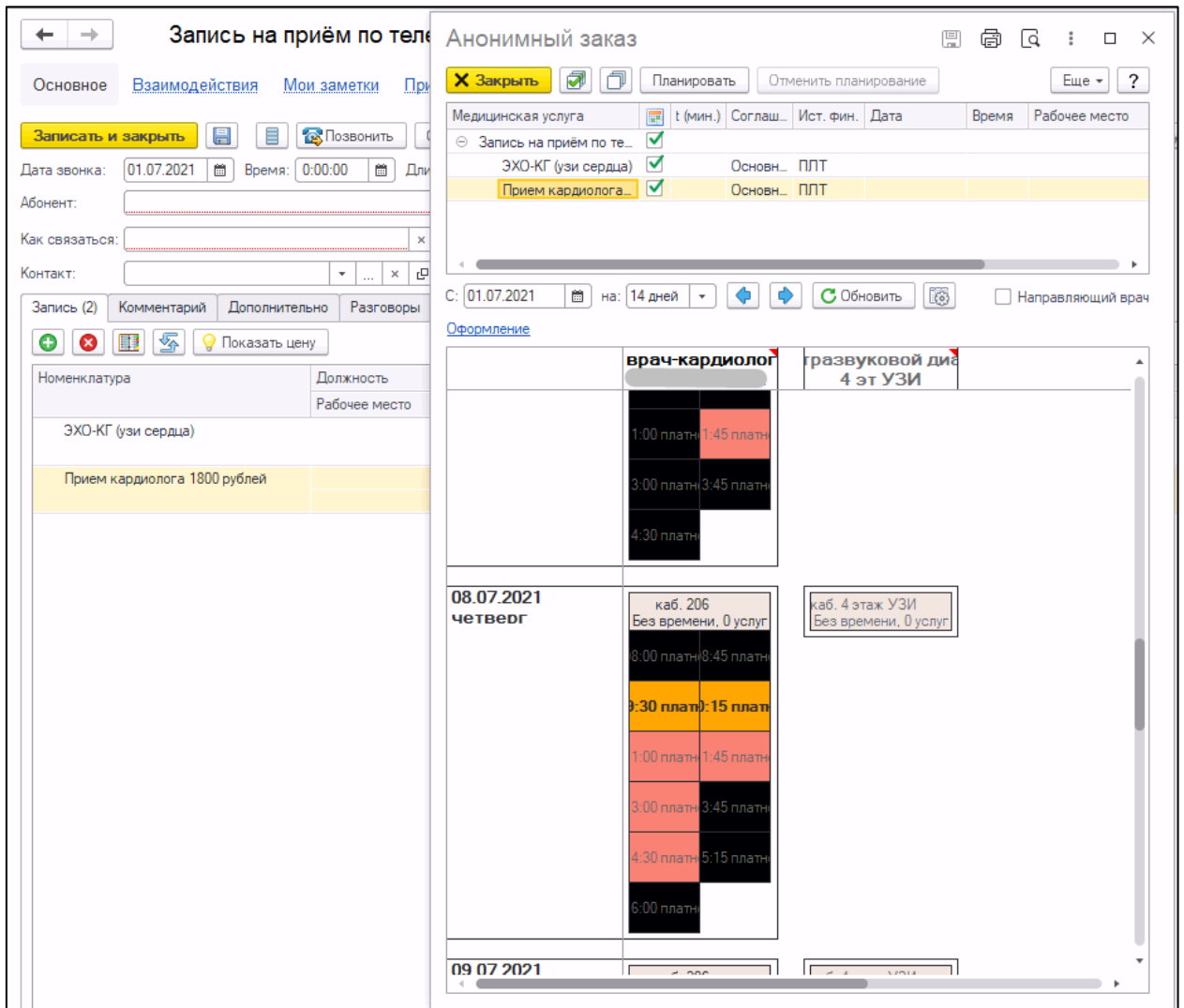


Рис. 3. Регистрация записи на прием в учетной системе

Запись на прием к врачу не обязывает пациента к посещению, поэтому возникают ситуации, когда пациент не является на прием, по какой-то причине и не сообщает об этом на Предприятие. Такие случаи важно выявлять заранее, так как, если запись была выполнена к врачу, у которого большая загруженность и очередь на прием может достигать нескольких недель, это позволит сократить размер очереди и записать на освобожденные места других пациентов.

Для обеспечения корректировки графика приемов в данный момент существующим решением является ручной обзвон пациентов сотрудниками call-центра. Недостатком существующего решения является то, что

первоочередной задачей является прием входящих звонков для записи новых пациентов, и только на втором плане корректировка графика приема уже записанных. В связи с этой возможностью обзвона ограничена и выполняется только для нескольких медицинских направлений с большой загруженностью и только при появлении свободного времени от приема входящих звонков.

Для того, чтобы выполнить контроль посещения, сотрудник контакт-центра открывает расписание врача на день, по которому будут выполняться звонки (Рис. 4). При двойном нажатии на приеме, откроется окно управления заказанными услугами (Рис. 5), в котором пользователь открывает документ записи на прием для просмотра контактной информации пациента.

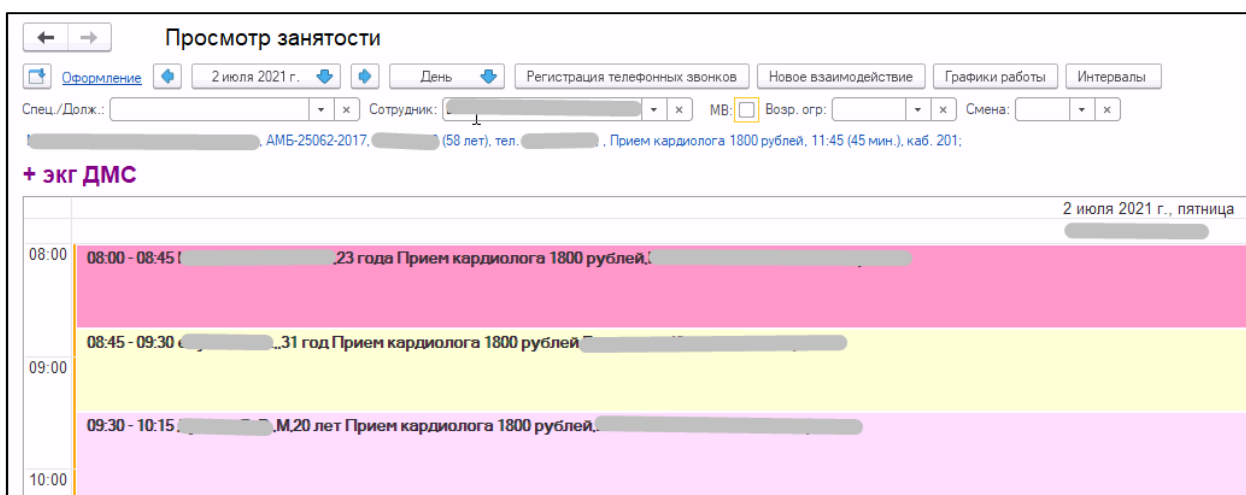


Рис. 4. Просмотр записанных пациентов

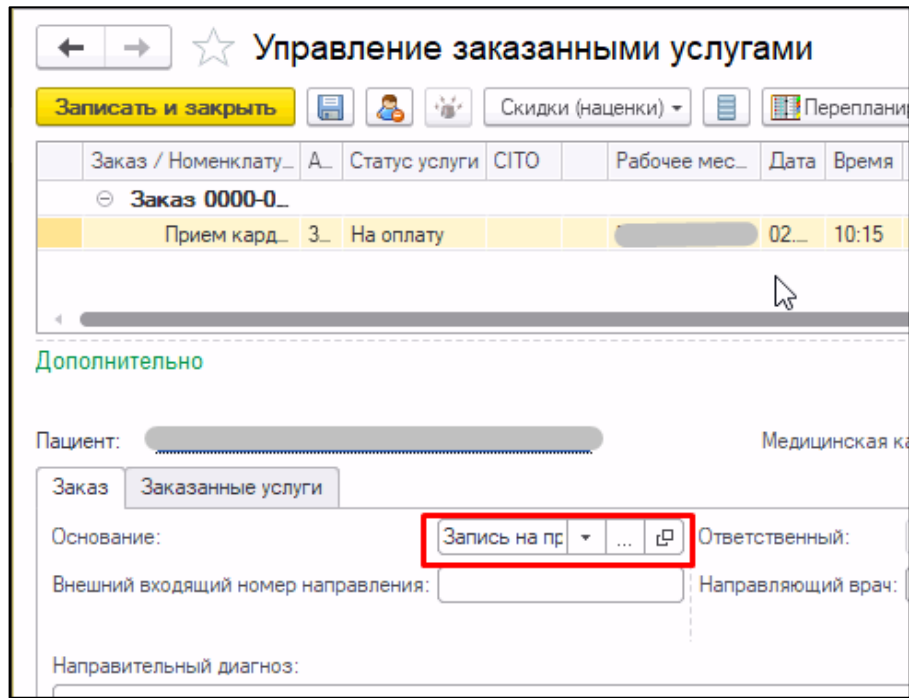


Рис. 5. Окно управления заказанными услугами

После открытия окна сотрудник call-центра набирает номер, указанный в поле «Как связаться» и далее взаимодействует с пациентом (Рис. 6).

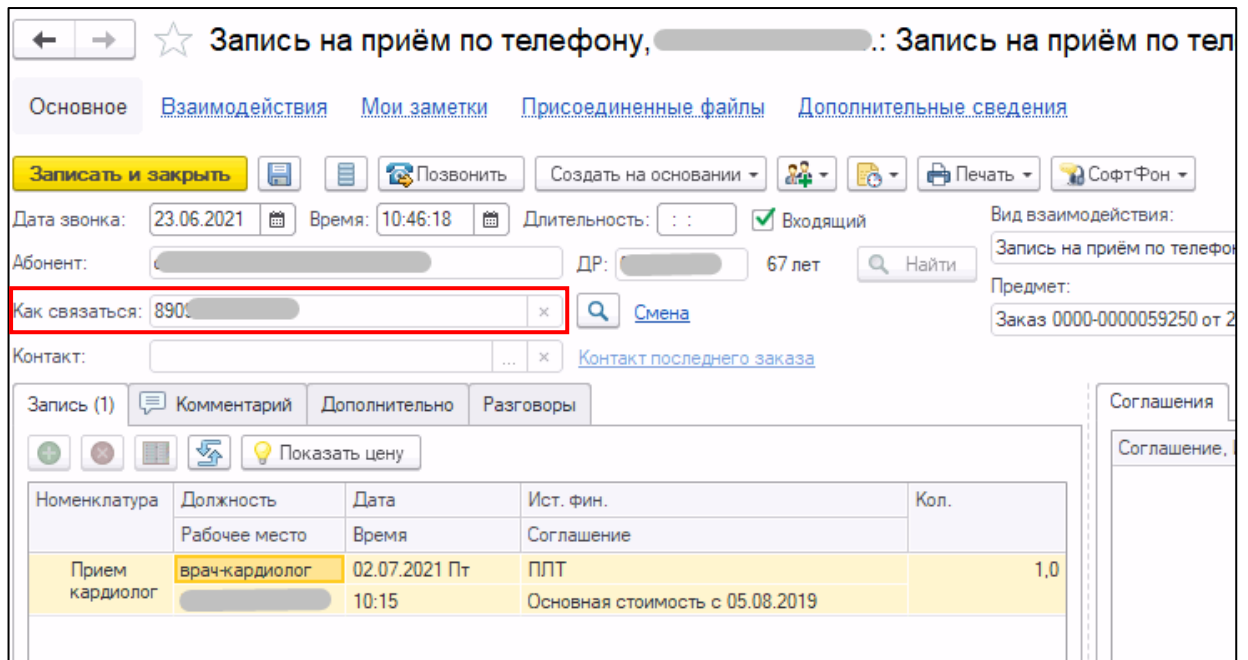


Рис. 6. Документ записи пациента на прием

В случае если пациент отменяет запись на прием, пользователь возвращается к окну управления заказанными услугами и нажимает кнопку

«отменить услугу». Уточнив у пациента причину отказа от приема, выбирает ее из выпадающего списка (Рис. 7).

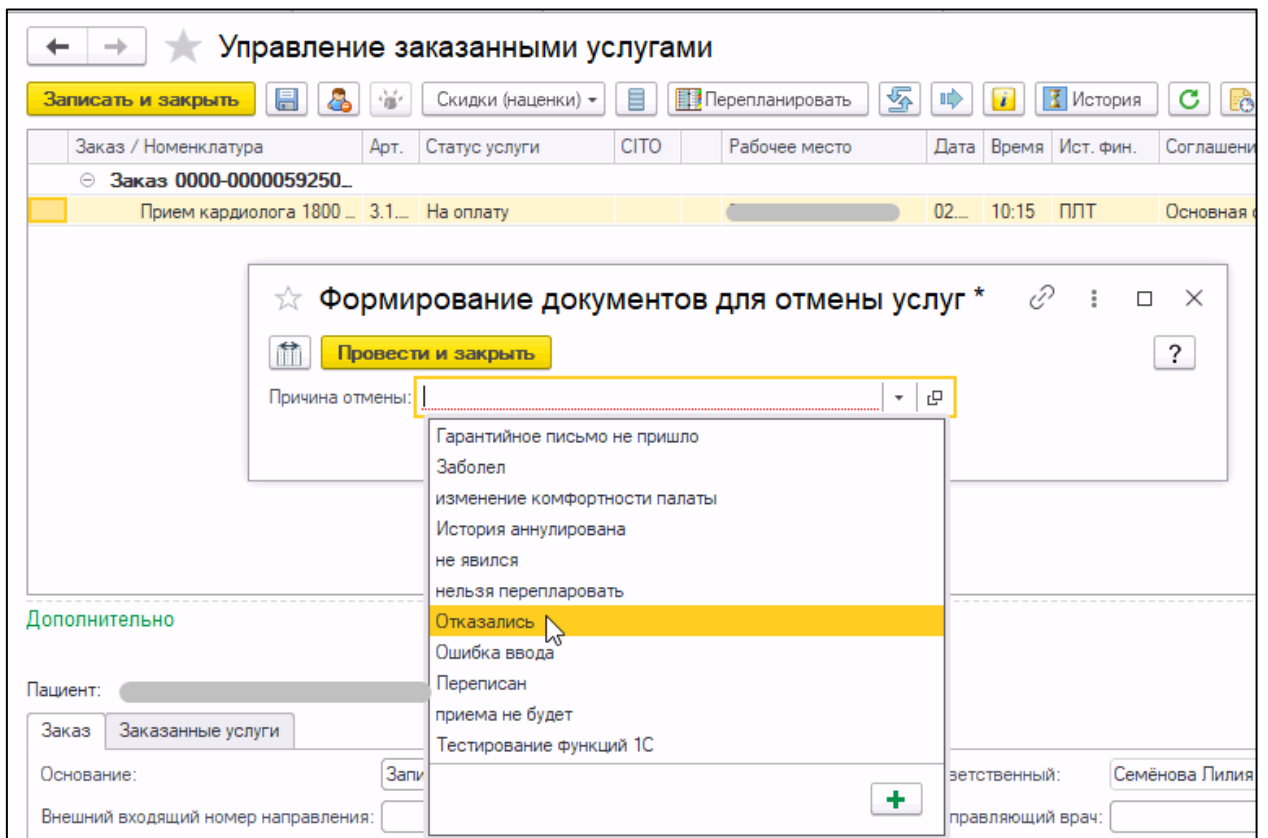


Рис. 7. Отмена записи на прием

По завершении, оператор нажимает кнопку «Провести и закрыть», чтобы подтвердить отмену приема. После нажатия этой кнопки запись на запланированное ранее время освободится для записи других пациентов.

1.2. АЛЬТЕРНАТИВНЫЕ СРЕДСТВА ВЗАИМОДЕЙСТВИЯ

Для взаимодействия с клиентами компании используют современные средства коммуникации. Некоторые из них, такие как СМС уже устаревают и популярность набирают мессенджеры и чат-боты как бесплатная альтернатива.

Из определения компании Oracle: «в самом общем смысле чат-бот — это компьютерная программа, которая интерактивно имитирует человеческую речь (устную или письменную) и позволяет общаться с цифровыми устройствами так, как если бы они были живыми людьми. Чат-боты могут быть очень простыми, как элементарные программы, отвечающие на простой запрос однострочным ответом, или сложными, как цифровые помощники, которые обучаются и развиваются по мере сбора и обработки информации, тем самым повышая свой уровень персонализации» [2].

Видя преимущество чат-бота перед СМС-рассылкой, может сложиться впечатление, что это идеальное решение для взаимодействия с клиентами. Но использование современных технологий может негативно сказаться на качестве взаимодействия с клиентами, если они относятся к старшей возрастной категории, которая с трудом может использовать новые методы коммуникации. Трудность заключается в том, что большая часть пациентов Предприятия – люди в возрасте от 50 до 80 лет. Для использования чат-бота пользователь должен иметь смартфон и уметь пользоваться одним из популярных мессенджеров, чтобы с ним можно было взаимодействовать таким способом. Кроме того, текстовое сообщение не обязывает человека к ответу, а ответ о намерении посетить прием должен быть получен как можно раньше.

В качестве дополнительного инструмента для уменьшения количества неявок на прием к врачу на Предприятии используется СМС-рассылка, которая содержит в себе информацию о записи на прием, а также в текст

сообщения включена просьба позвонить и уведомить при отказе от посещения врача. Этот подход имеет эффект, но так как отклик не обязателен, не все пациенты удовлетворяют просьбу. Но, когда оператор звонит пациенту, чтобы определить придет он на прием или нет, ответ оказывается получен в реальном времени, что дает дополнительное время для поиска нового пациента на освободившееся место.

Исходя из вышесказанного, по причине преобладающего количества пациентов старшей возрастной группы, было принято решение использовать голосовое взаимодействие вместо чат-бота. Это обосновано тем, что выбранный способ более прост для понимания и позволяет получить ответ в момент звонка.

1.3. ПРИМЕНЕНИЕ ГОЛОСОВЫХ РОБОТОВ

Как было описано выше, для осуществления звонков силами сотрудников потребовалось бы большое количество времени, которое сотрудники не могут выделить из-за большого количества входящих звонков, на которые необходимо отвечать. Поэтому было предложено решение, для рутинной операции контроля посещения приема, использовать голосового робота (бота).

В целом, для большинства операций в call-центре, когда необходимо получить однотипную информацию от клиента, сложные (для программы) навыки человеческого общения не требуются. Поэтому благодаря развитию технологий распознавания и синтеза речи, голосовые боты получили широкое распространение.

Современные технологии позволяют обеспечить качество синтезированной речи на таком уровне, что ее сложно отличить от обычной речи человека, если к этому добавить быструю скорость ответа, тогда создается ощущение разговора с реальным человеком [4].

Голосовые помощники применяются в различных сферах, таких как HR (найм сотрудников), службы доставки, продажи и других. Их основная задача снимать с человека выполнение рутинных задач вроде обзвона кандидатов на вакантные места или уточнение деталей заказа на доставку. Робот может собирать и обрабатывать обратную связь от пользователей, информировать о статусе заказа, напоминать о платежах, запрашивать и актуализировать данные.

Один из тривиальных случаев применения голосовых роботов – это получение простых ответов от абонента и предоставление ему информации. К примеру, робот может позвонить клиенту и уточнить актуальность заявки на подключение интернета и при получении утвердительного ответа, продолжить диалог вопросом об удобном времени для ожидания мастера. Ответ клиента фиксируется в базе данных и отправляется мастеру в работу [6].

Проблема обзвона базы клиентов часто встречается в сфере оказания услуг. Основная причина использования голосовых роботов – это использование их для осуществления рекламной кампании, поскольку этот инструмент позволяет выполнять массовые звонки гораздо быстрее человека. Для того, чтобы провести обзвон большой базы клиентов также требуется значительный штат сотрудников, что подразумевает также большие расходы на зарплату. Голосовой робот обходится значительно дешевле, когда цель – произнести заранее записанную фразу.

Однако, когда речь заходит о голосовом взаимодействии робота и человека, в котором робот должен правильно взаимодействовать с человеком, механизм усложняется. От робота требуется не только произнести речь, но и понять, что говорит человек в ответ.

Так как у звонков есть цель, существует некоторая последовательность этапов, через которые робот должен пройти, чтобы ее достичь. Таким образом, в алгоритме предусматривается ограниченное количество стадий разговора с

определенными вариантами ответа и соответствующими им переходами к следующему этапу. Данный механизм может быть реализован в двух вариантах: использование NLP (Natural Language Processing) или ограничение несколькими вариантами ответа. В первом случае после преобразования ответа человека в речь выполняется обработка естественного языка, которая позволяет определить к какому из заранее определенных вариантов более близок ответ. Второй же вариант более прост в реализации, но не позволяет скрыть, что взаимодействие выполняется с роботом. Суть заключается в использовании заранее ограниченного набора ответов, которые человеку предлагается выбрать. После получения ответа происходит проверка полученного из речи текста на соответствие одному из предусмотренных вариантов, что позволяет направить взаимодействие далее по соответствующему направлению диалога.

В сокрытии факта взаимодействия с роботом есть преимущество – человек не будет думать о том, что теряет время и не будет пытаться прекратить разговор досрочно или требовать переключения на оператора. Однако при использовании более сложных методов распознавания речи риск ошибок робота выше. При использовании готовых фраз процесс распознавания и условного перехода по ветвям диалога для робота упрощается.

Для реализации автоматического голосового взаимодействия на Предприятии было решено использовать подход с использованием заранее определенного набора ответов, которые робот будет предлагать пациенту для выбора. Это обосновано тем, что каждый этап диалога содержит малое количество вариантов ответа и система должна быть максимально проста для понимания.

1.4. СУЩЕСТВУЮЩИЕ РЕШЕНИЯ

Актуальность проблемы и широкая область применения голосовых роботов привела к тому, что многие компании стали предоставлять бизнесу готовые сервисы для решения задачи по автоматическому обзвону клиентов.

Предлагаемые решения:

- проведение рассылки рекламных сообщений;
- проведение опросов;
- осуществление записи на оказание услуги;
- робот для автоматического ответа на часто задаваемые вопросы.

Рекламная рассылка осуществляется с использованием заранее записанной диктором речи и дальнейшим ее воспроизведением при поднятии трубки потенциальным клиентом. Возможно использование дополнительной проверки на желание абонента прослушать рекламную запись: «удобно ли вам выслушать наше предложение?». Такой тип решений также позволяет после воспроизведения рекламного сообщения предложить отправить абоненту СМС с контактной информацией для дальнейшего оформления заявки.

Автоматическое проведение опросов выполняется с использованием заранее сформированного набора вопросов, которые озвучиваются абоненту. Позволяют собрать требуемую информацию по имеющемуся набору телефонных номеров. Часто используется для оценки качества оказанных услуг, например, уровня удовлетворенности домашним интернетом или оценкой качества оказанных услуг. Результаты опроса могут быть отправлены в административную панель сервиса или в учетную систему для анализа.

Следующий тип решения - запись на оказание услуг. Для реализации такого механизма, помимо подготовленного сценария диалога, требуется интеграция с базой данных, которая обеспечивает хранение данных о занятых и свободных окнах для записи. Это нужно для того, чтобы голосовой робот мог в ответ на просьбу о записи на оказание услуги (автомойка, медицинские

услуги и др.) проверить доступность мест для записи и предложить свободные. При выборе пользователем одного из вариантов робот осуществляет запись на выбранное время.

Робот в качестве автоответчика на часто задаваемые вопросы используется в организациях, предоставляющих услуги большому количеству людей и имеющих горячую линию для консультации по вопросам, связанным с оказываемыми услугами. Большое количество однотипных вопросов загружает горячую линию вынуждая ожидать освобождения оператора. Система позволяет предусмотреть ответы на наиболее распространенные вопросы, привлекая оператора только в случае, когда система не может предложить решение вопроса самостоятельно. Для определения сущности вопроса используются средства обработки естественной речи. После преобразования речи в текст, содержимое вопроса классифицируется предварительно обученной моделью к одной из predeterminedных категорий, например «оформление паспорта», «регистрация на рейс». После классификации вопроса взаимодействие продолжается по ходу сценария диалога.

Большинство решений позволяют, используя минимальное количество программного кода, сформировать список номеров для обзвона, настроить произношения вопроса с изменяемыми частями такими как имя абонента или размер скидки на бонусной карте, обработать ответы и отправить результат в другую систему для анализа и обработки.

Существующие решения на рынке предлагают быстрый обзвон базы клиентов, сбор результатов. Например, компания МТТ заявляет, что использование их сервиса VoiceVox, позволит совершать более 300 звонков в минуту [5]. При этом возможна отправка результатов ответа в учетную систему предприятия для дальнейшей обработки и анализа. Подобным функционалом обладают такие продукты как Robovoice и Voximplant Kit, но дополнительно имеют графический интерфейс для построения скрипта звонка

без необходимости программирования. Сервисы также предоставляют API для инициации звонков из программного кода.

Достоинство существующих механизмов заключается в том, что они позволяют быстро решить задачу типового обзвона по списку номеров. Готовые решения включают скрипт, с изменяемыми, который можно настроить под требования заказчика. Недостаток готовых решений заключается в наличии ограничений по настройке, фиксированной абонентской плате за использование сервиса и отдельной платы за телефонную линию.

При использовании собственной разработки затраты на использование телефонной линии, распознавания и синтеза речи остаются, но будут зависеть только от интенсивности использования. Кроме того, собственное решение позволит в дальнейшем использовать дополнительные возможности интеграции с IP-телефонией, например перевод звонка на оператора по просьбе пациента. Поэтому для того, чтобы инструмент голосового взаимодействия был более гибким и экономичным для Предприятия, было принято решение реализовать его самостоятельно.

ГЛАВА 2. ПОДХОД К РЕШЕНИЮ

2.1. АНАЛИЗ ДАННЫХ

Данные необходимые для оценки полезности разработки – это статистика по оказанным услугам: количество выполненных и отмененных услуг. Эти данные позволяют оценить загруженность врачей и на основании полученных результатов определить целесообразность обзвона по каждому медицинскому направлению. Также для анализа будут использованы данные по статистике оказанных услуг в разрезе возрастных групп пациентов, которые позволят определить правила, определяющие способ взаимодействия с пациентом: с использованием робота или через оператора. Данные были получены с использованием запросов 1С. «Запросы – это один из базовых механизмов «1С:Предприятия» наряду со встроенным языком, который позволяет читать и обрабатывать данные, хранящиеся в базе. Для составления запросов «1С:Предприятие» использует собственный язык, основанный на SQL» [1].

Цель изучения данных – определить наиболее критичные медицинские направления, на которых самая большая плотность записи из-за чего пациентам сложно туда записаться. Под плотностью записи подразумевается отношение количества записанных пациентов к количеству доступных окон для приема. Количество доступных окон для записи определяется графиком работы врача, который может быть разным на каждый день недели (рис. 8). Данные по записанным пациентам хранятся в таблице информационной базы и выводятся врачам в интерфейсе на выбранный день (рис. 9).

← → ☆ **_____ _01.08.20_X суббота (График работы)**

Записать и закрыть | Записать | Извлечь из архива | Поместить в архив

Наименование: _____ _01.08.20_X суббота

Период заполнения графика с: 01.08.2020 по: ... Вид графика: Основной

Владелец графика: Бердинских С.Г. | Время выполнения услуги: 45

Шаблон заполнения:

Понедельник	<input checked="" type="checkbox"/>	6,8 ч. (09:00-12:00, 12:30-16:15)
Вторник	<input type="checkbox"/>	Заполнить расписание
Среда	<input type="checkbox"/>	Заполнить расписание
Четверг	<input type="checkbox"/>	Заполнить расписание
Пятница	<input type="checkbox"/>	Заполнить расписание
Суббота	<input checked="" type="checkbox"/>	6,8 ч. (09:00-12:00, 12:30-16:15)
Воскресенье	<input type="checkbox"/>	Заполнить расписание

Отмеченные дни будут включены в график работы.

Расписание работы:

Добавить | ↑ ↓ | Сохранить

Время начала	Время окончания	Количество окон	Отметка периода	Условие функционирования
9:00:00	12:00:00	4	платно	Амбулаторно-поликлинические
12:30:00	16:15:00	5	платно	Амбулаторно-поликлинические

Рис. 8. Количество окон для приема в графике работы сотрудника

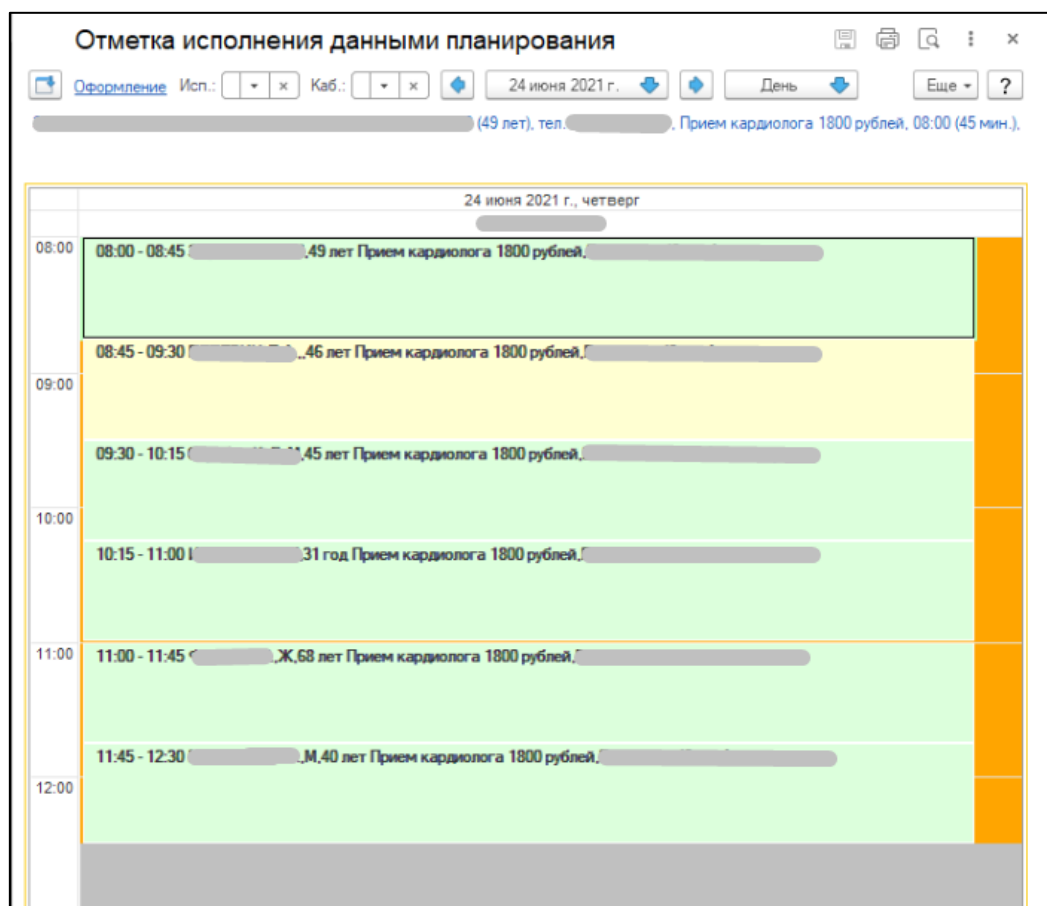


Рис. 9. Расписание приемов врача на текущий день

Для достижения цели был выполнен анализ данных по загруженности врачей (Рис. 10), которые показали, что загрузка по врачам неравномерна, что подтверждает необходимость определения приоритета звонков. Доля загруженности более единицы означает, что к врачу записывают пациентов сверх максимального количества окон, что допустимо, и выполняется с согласия врача.

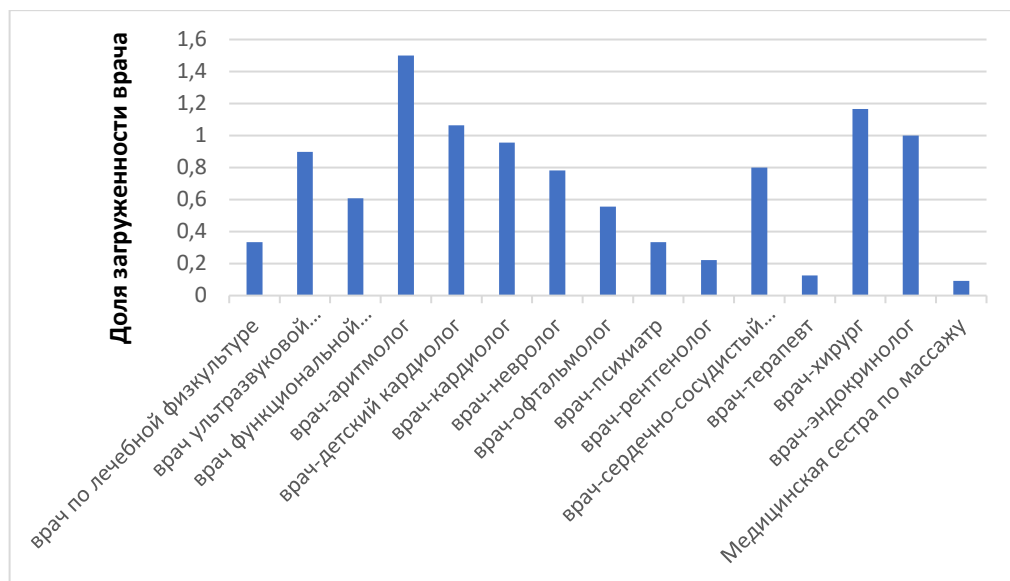


Рис. 10. Данные по загруженности врачей на 22.06.2021

Для определения приоритета звонков по отношению к врачам, запланированные услуги ранжируются по загруженности врачей. Принято решение выполнять обзвон по медицинским направлениям с загрузкой более 80%. Для формирования выборки был использован встроенный в 1С механизм запросов. Для формирования результата были использованы данные по запланированным услугам, графики работы врачей и данные о нормах времени для приемов. Услуги сгруппированы по запланированному дню и плотность записи рассчитана как отношение количества записанных пациентов к количеству доступных окон для записи. Для определения количества доступного количества окон для записи было использовано отношение количества выделенного рабочего времени к установленной норме времени приема.

Другим возможным критерием для выбора пациентов для обзвона может быть использован возраст, поскольку есть предположение, что для общения с пожилыми пациентами лучше не использовать средства автоматического голосового взаимодействия. Исследование, проведенное в 2019 году Пермским национальным исследовательским политехническим университетом [15] показывает, что оценка качества общения с голосовыми роботами отличается у разных категорий людей, поэтому для упрощения взаимодействия возникает необходимость разделить пациентов на группы по виду взаимодействия на традиционный способ общения и автоматический. В исследовании был проведен эксперимент взаимодействия людей с голосовыми роботами и на основании опросов были выявлены зависимости оценок качества взаимодействия от нескольких факторов. Важным фактором в оценке качества взаимодействия была степень погружения человека в область технологий и его опыта взаимодействия с голосовыми роботами. Выявлена зависимость от возраста участвующих в опросе. Наиболее положительно отзывались люди в возрасте от 18 до 30 лет.

Исследование показало, что возраст имеет важное значение в оценке качества взаимодействия, поэтому этот фактор необходимо учитывать, чтобы исключить автоматическое общение с людьми, которые к нему не привыкли. Но учитывая, что основная доля пациентов на Предприятии в возрасте от 30 до 80 лет (рис. 11), то возникла необходимость расширения границ ограничений по возрасту до 70 лет.

Для того, чтобы определить стратегию совершения звонков была собрана статистика за первую половину 2021г., в которой было рассмотрено отношение оказываемых услуг к возрасту пациента (рис. 11).

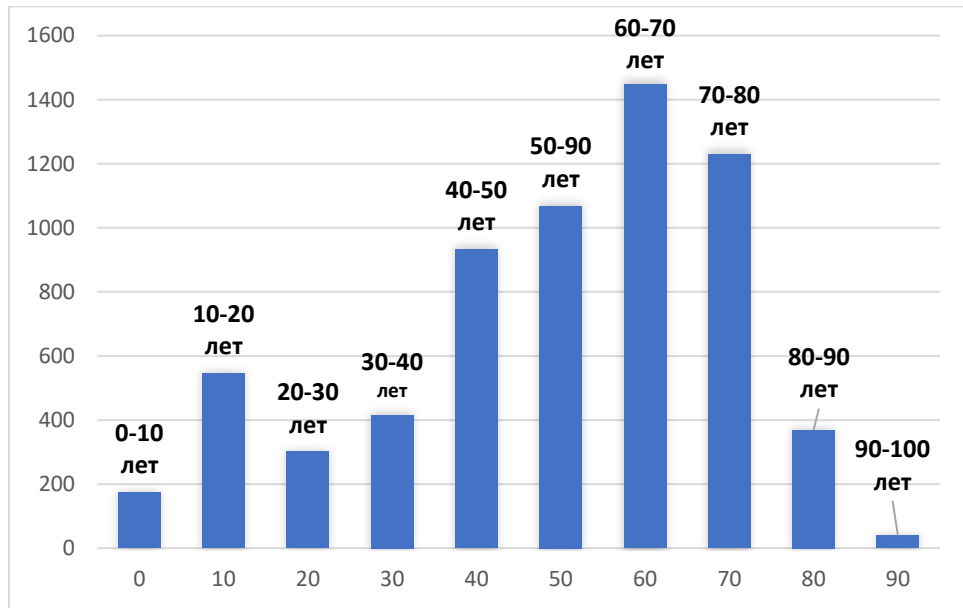


Рис. 11. Количество оказанных услуг по возрастам пациентов

На графике видно преобладание пациентов в возрасте от 30 до 80. Округлив возраст до десятков лет и разделив пациентов на возрастные группы, можно определить соотношение количества оказанных услуг по возрастным группам (см. рис. 12).

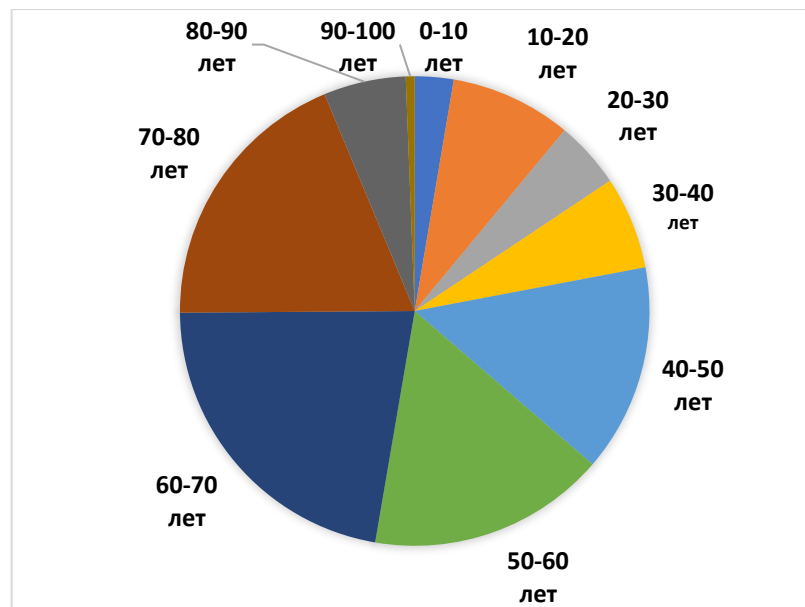


Рис. 12. Отношение количества оказанных услуг по возрастным группам пациентов

Принимая во внимание полученные данные, принято решение автоматический обзвон делать для пациентов в возрасте от 16 до 70 лет (в возрасте до 15 лет, взаимодействие через законного представителя), поскольку это позволит охватить большую часть пациентов и при этом не причинять неудобств старшей категории пациентов.

Для того, чтобы выявить дополнительные предпосылки для сокращения количества звонков были получены данные за полугодие с группировкой по месяцам (Рис. 13). На графике заметно снижение количества приемов в январе и мае, предположительно большое влияние оказывает большое количество праздничных дней. Для проверки гипотезы проверим данные за эти периоды. После получения данных за январь 2021 г. (Рис. 14) подтвердилось предположение о том, что праздники влияют на количество оказываемых услуг. Это связано с тем, что не все врачи выходят на работу в указанный период, снижая пропускную способность. Количество оказываемых услуг снова возрастает по завершении праздников (11 января).

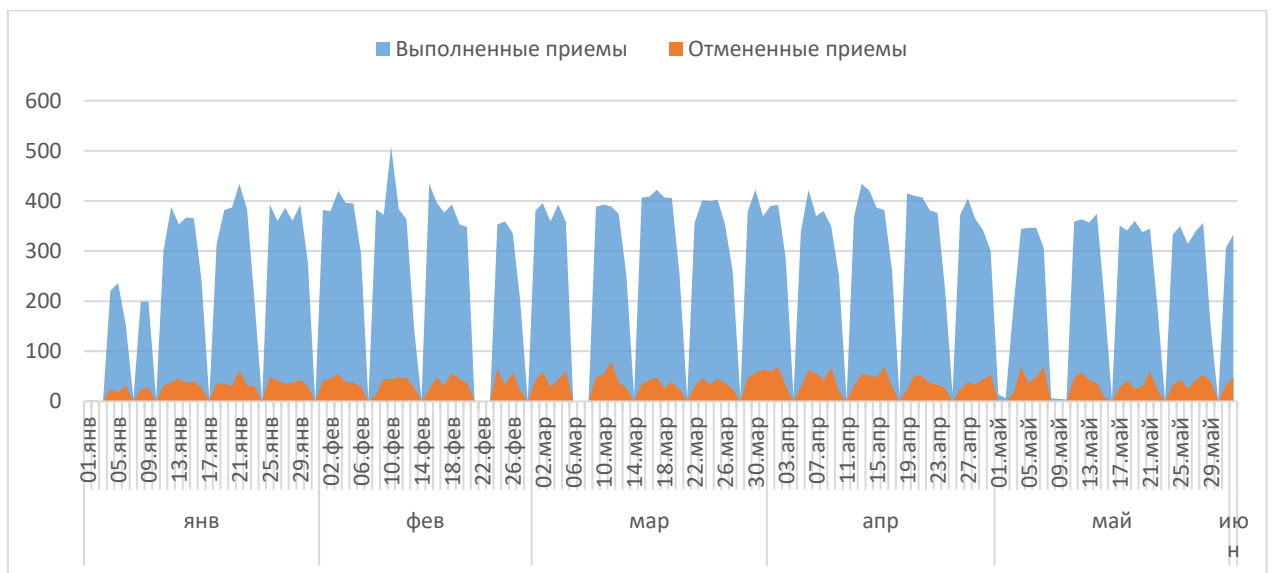


Рис. 13. Соотношение оказанных услуг за первое полугодие 2021 г.

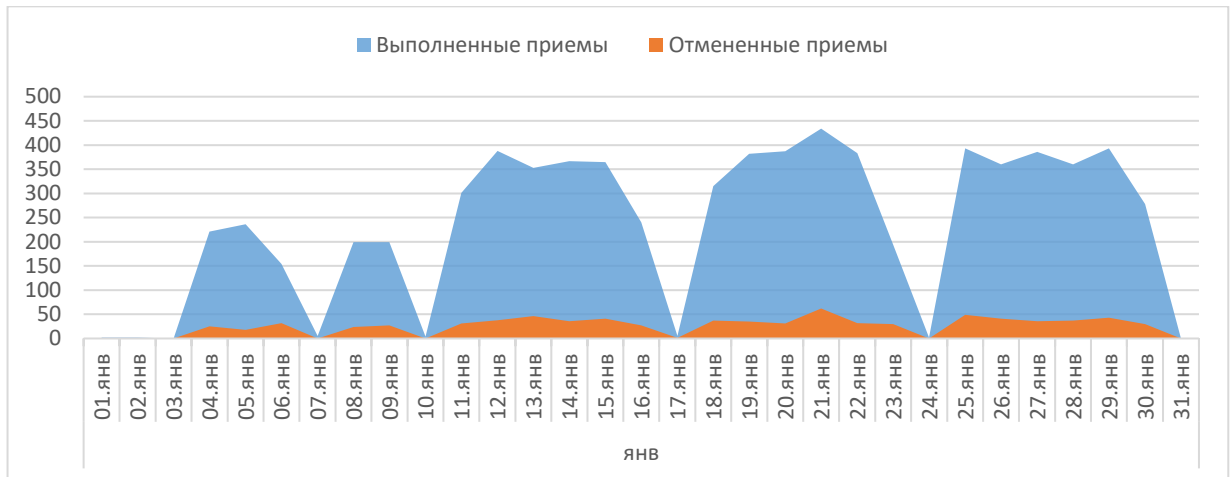


Рис. 14. Соотношение оказанных услуг за январь 2021 г.

Проанализировав данные за май 2021 г., где также было замечено уменьшение среднего количества оказанных услуг, выявлено равное снижение среднего количества оказанных услуг (Рис. 15). Предположительно, такая ситуация вызвана началом летнего сезона отпусков. Анализ данных в разрезе дней показывает, что количество оказываемых услуг не связано с днем недели.

Полученные данные по праздничным дням не позволяют сформулировать дополнительное ограничение для обзвона, поэтому изучение данных было продолжено.

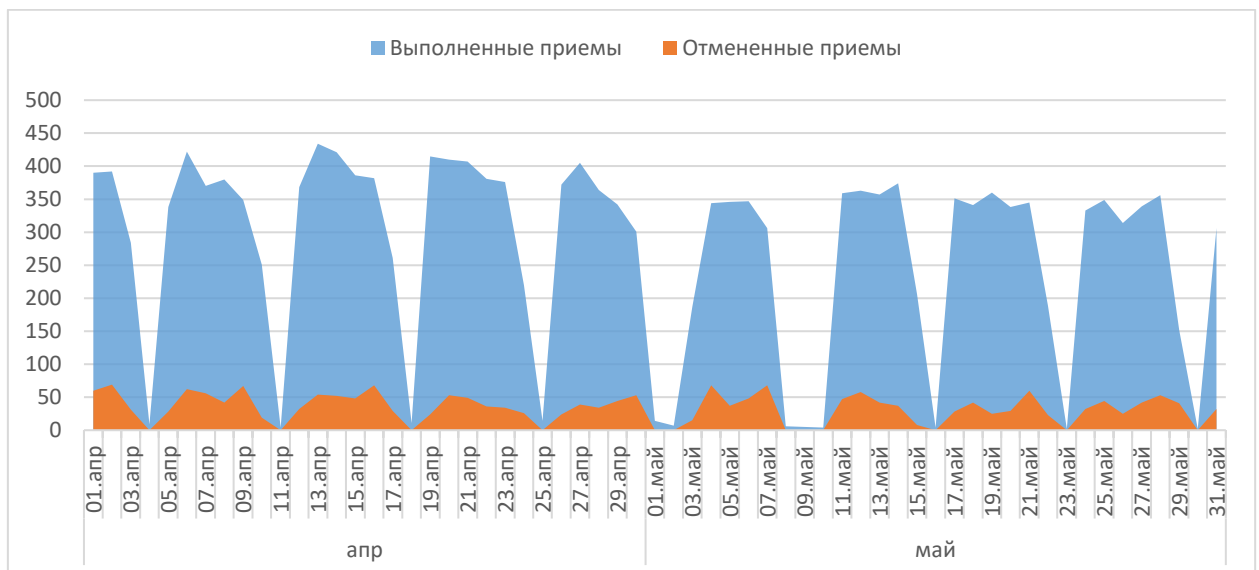


Рис. 15. Соотношение количества оказанных услуг за апрель, май 2021 г.

Следующее предположение заключается в том, что количество отказов услуг может зависеть от времени дня. Возможно человеку проще выделить время, чтобы посетить врача в утреннее и вечернее время, а не в середине дня. Для проверки этой гипотезы были взяты данные за февраль 2021г. Запланированные приемы распределены по времени по половинам дня (Рис. 16 и Рис. 17).

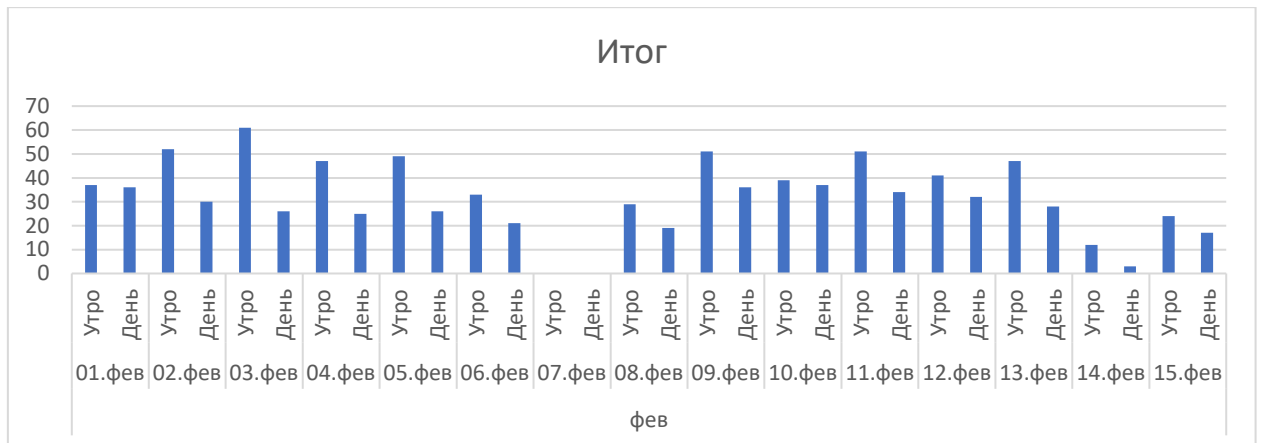


Рис. 16. Зависимость количества отмен услуг от части дня (первая половина месяца)

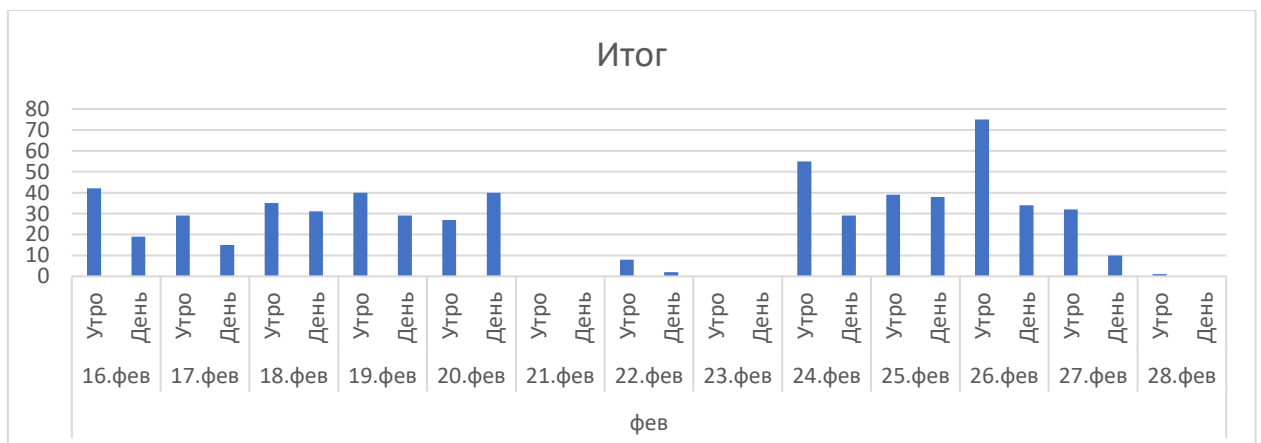


Рис. 17. Зависимость количества отмен услуг от части дня (вторая половина месяца)

На графиках можно заметить, что количество отмененных услуг меньше во второй половине дня, но при этом четко сформулировать правило, которое бы

позволило сократить количество вечерних звонков, не представляется возможным.

В результате произведенного анализа данных удалось выявить несколько закономерностей, связанных с количеством отменяемых и выполненных услуг, но только часть из них можно применить для уменьшения количества совершаемых звонков. Так, полученные данные по загруженности врачей и отношении количества оказанных услуг к возрасту пациента позволяют сформулировать условия, по которым будет выполняться автоматический обзвон пациентов. Звонки будут выполняться только по тем медицинским направлениям, где загрузка сотрудника более 80% и возраст пациента попадает в промежуток от 16 до 70 лет. Эти правила будут использованы при формировании задач на совершение автоматического звонка.

2.2. ВЗАИМОСВЯЗЬ ПРОГРАММНЫХ СИСТЕМ

Для разработки механизма автоматической оптимизации графика приемов требуется объединение нескольких программных средств в единую подсистему, которая будет решать следующие задачи:

- формирование очереди обзвона пациентов;
- автоматическая инициация звонков на основании очереди;
- обработка ответов пациента и отправка результатов в информационной базе Предприятия;
- отмена или перенос записей пациентов на основании результатов звонков.

Для решения указанных задач будут использованы следующие программные модули:

- подсистема 1С для постановки задач на корректировку запланированного приема;
- сервис, предоставляющий API для управления звонками через Asterisk Management Interface (AMI);
- приложение EAGI для осуществления голосового взаимодействия;
- сторонний сервис для синтеза и распознавания речи - Yandex SpeechKit [9].

Программа 1С содержит данные о приемах пациентов и в ней они корректируются, поэтому планирование звонков выполняется в этой системе. Для формирования очереди используется расписание приемов, на основании которого заполняется очередь звонков (рис. 18). Формирование очереди выполняется по расписанию каждую ночь.

Очередь звонков представляет собой очередь FIFO, то есть первый запланированный звонок будет выполнен первым. На стороне 1С происходит

обращение к сервису, который напрямую взаимодействует с Asterisk и предоставляет возможность управления звонками.

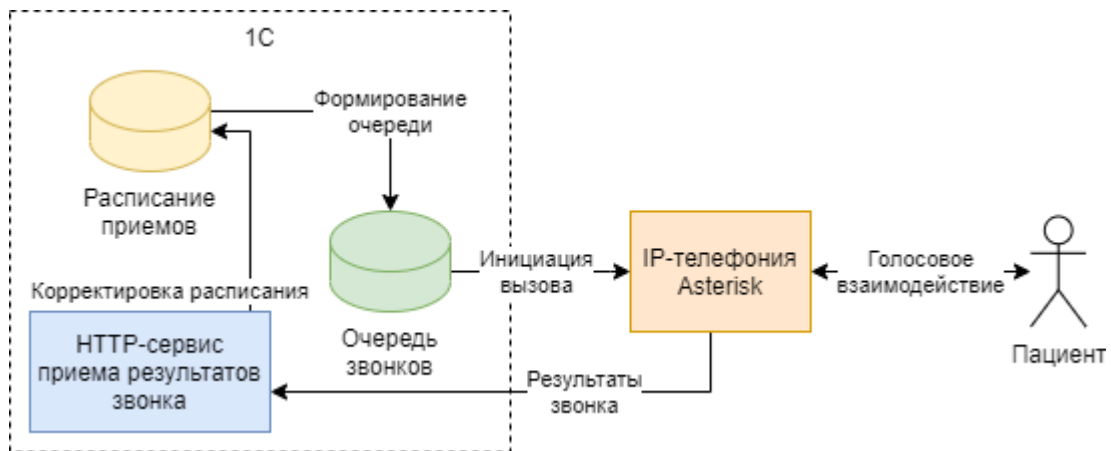


Рис. 18. Детализация системы 1С

На стороне Asterisk будет развернут дополнительный сервис для инициации звонков через АМІ. АМІ – это программный интерфейс, позволяющий сторонним программам осуществлять контроль и мониторинг системы Asterisk. Этот интерфейс часто используется для интеграции Asterisk с существующими бизнес-процессами и системами. Также может применяться для реализации автоматического обзвона и систем звонка по одной кнопке [8, с 227].

В диалплане (настройках маршрутизации вызовов) выполнена настройка для запуска EAGI – приложения (Enhanced Asterisk Gateway Interface), с передачей в него аудиоканала и работы с распознаванием и синтезом речи [8, с. 209]. Это приложение также будет выполнять отправку результатов голосового взаимодействия (общения) с пациентом в информационную систему Предприятия посредством HTTP-запросов. Схема работы компонентов с детализацией голосового взаимодействия представлена на рисунке 19.

Распознавание и синтез речи будет выполняться готовым инструментом, предоставляемым компанией «Яндекс» - Yandex SpeechKit. Это позволит

сконцентрировать внимание на разработке архитектуры и не касаться разработки инструмента для обработки речи. Для формирования исходящего аудио будет использована комбинация заранее заготовленных аудио фрагментов и генерируемых во время звонка, например, для того чтобы озвучить время приема и специальность врача. Распознавание ответов пациента должно выполняться в реальном времени, чтобы система могла реагировать на то, что говорит пациент. Для этого исходящий аудиоканал отправляется на распознавание не цельной записью разговора, а фрагментами, и результаты распознавания будут определять поведение робота.

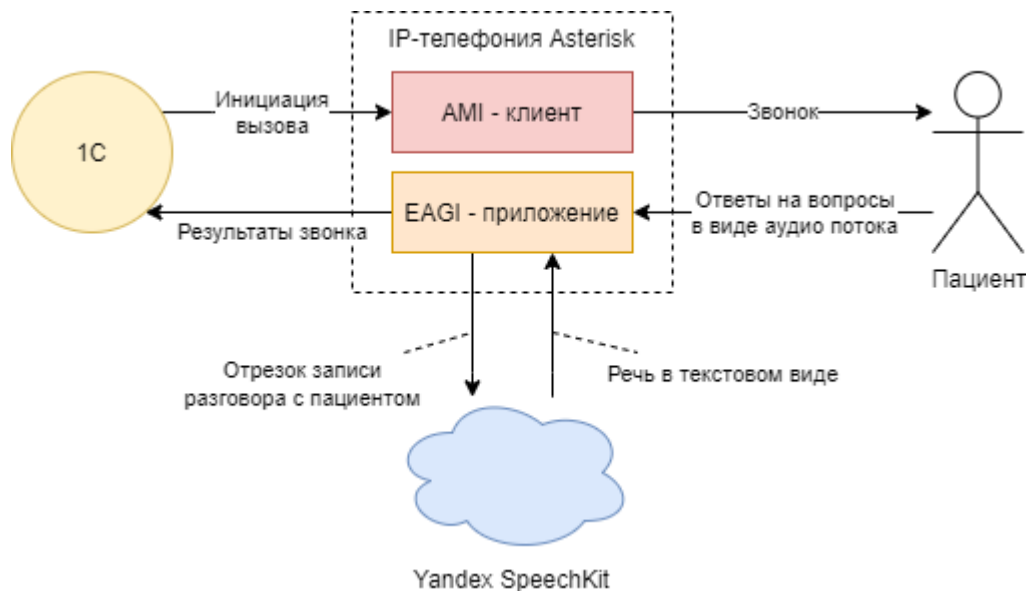


Рис. 19. Детализация подсистемы IP-телефонии

Для приема результатов взаимодействия с пациентом в 1С добавляется HTTP-сервис, который выполняет корректировку расписания приемов врачей.

2.3. ИНФОРМАЦИОННАЯ СИСТЕМА 1С

Большая часть результатов деятельности Предприятия сконцентрирована в одной учетной системе, в 1С. В ней осуществляется работа врачей, кассиров, сотрудников call-центра, регистратуры.

Программа 1С: Медицина. Больница была разработана для хранения и управления, связанными с оказанием медицинских услуг. Часть этих данных составляет информация о запланированных приемах пациентов. Она является отправной точкой для реализации автоматической корректировки, потому что будет использована для подтверждения и отмены записи пациента по результатам звонка. Для совершения звонка и осуществления голосового взаимодействия необходимы следующие данные:

- номер телефона пациента;
- специальность врача;
- запланированное время приема.

Вышеуказанная информация заполняется сотрудниками call-центра в момент планирования приема. Данные записываются в программу в интерактивном режиме. В случаях, когда требуется изменить данные записи, они корректируются пользователем. В виду того, что система содержит все данные, связанные с записью пациента, это означает, что она должна быть мастер-системой, управляющей процессом автоматической корректировки.

Исходный программный код конфигураций 1С является открытым для доработки тем самым позволяя свободно расширять функционал программы и осуществлять интеграцию со сторонними системами. 1С позволяет «легко» создавать решения с нуля — благодаря встроенным объектам, реализующим базовый функционал учетных систем. Именно продуманная система встроенных объектов (а не язык, который в общем-то обычный скриптовый) делает «1С:Предприятие» мощным инструментом создания бизнес-приложений. Разработчику не нужно писать слой доступа к данным, базовый

UI и т.п. – можно сразу сосредоточиться на решении бизнес-задачи. Для решения бизнес-задач также многое уже реализовано во встроенных объектах (читай – базовых библиотеках) – например, поддержка иерархических справочников, учетные машины для реализации бухгалтерского и товарного учета, механизмы для сложных периодических расчетов (например, расчета зарплаты) и многое другое» [7]. Для обеспечения взаимодействия с внешними системами в 1С предусмотрен механизм HTTP-запросов (для отправки запросов) и HTTP-сервисов (для получения запросов). Эти механизмы позволяют реализовать интеграцию спроектированных подсистем. Детали реализации описаны далее.

2.4. IP-ТЕЛЕФОНИЯ ASTERISK

Для решения задачи осуществления звонков и голосового взаимодействия будет использовано открытое программное обеспечение для создания IP-телефонии – Asterisk (далее Asterisk). Данное ПО позволяет разворачивать сеть IP-телефонии любой сложности, отличается широкими возможностями доработки и интеграции с другими системами. Программное обеспечение позволяет получить полный контроль над прохождением звонка от его начала до завершения. Asterisk предоставляет такие возможности как создание очереди звонков, автоматическое распределение звонков по свободным операторам, возможность настройки автоответчика на основании графика работы Предприятия, взаимодействие автоответчика с абонентом путем тонального набора (набора цифр на телефоне), а также множество других функций. Данная система может быть усовершенствована за счет интеграции с другими подсистемами [8, с. 15].

То, что невозможно сделать предусмотренными инструментами системы, можно сделать интеграцией с другими сервисами. Кроме того, также существуют встроенные инструменты, с помощью которых, во время ожидания ответа пользователю может быть озвучена погода или передан звук с интернет-радио вместо гудков.

Примером большого количества возможностей, может служить интеграция IP-телефонии и учетной системы. Путем настройки Asterisk, использования его программных интерфейсов реализуется такая задача как отображение карточки входящего звонка в учетной системе. Там же могут быть отображены связанные с абонентом заказы и последние записи разговоров. Возможна реализация и обратной задачи – из карточки контактного лица реализовать выполнение звонка нажатием одной кнопки без набора номера. Для решения подобных задач уже существует множество готовых решений, но ПО Asterisk позволяет реализовать это самостоятельно при необходимости.

Описанное программное обеспечение уже установлено и настроено на Предприятии. Его задача – обеспечение телефонной связи по всему зданию и между филиалами. Для решения выявленной проблемы и вытекающих из нее задач, система IP-телефонии будет модифицирована и объединена с другими системами.

2.5. ТЕХНОЛОГИЯ YANDEX SPEECHKIT

Для решения задачи распознавания ответов пациента было решено использовать отечественный продукт от компании Яндекс. Преимущество этого сервиса перед аналогичными от Amazon или Google заключается в более качественном синтезе русской речи, что является предпочтительным, потому что все взаимодействие с пациентами происходит на русском языке.

Две ключевые функции, которые требуются от этого сервиса и которые он предоставляет, это синтез и распознавание речи. Синтез речи будет необходим при формировании изменяемых частей вопросов, которые будут задаваться пациенту:

- время записи на прием;
- специальность врача.

Распознавание речи будет использовано при обработке ответов:

- подтверждение записи;
- отмена записи;
- перезапись.

Так как голосовое взаимодействие будет происходить в реальном времени, то время реакции системы на ответы пациента должно быть минимальным. Для соответствия этому требованию на распознавание будут отправляться небольшие фрагменты из разговора с пациентом, записываемые сразу после озвучивания вопроса.

ГЛАВА 3. РЕАЛИЗАЦИЯ СИСТЕМЫ

3.1. ОЧЕРЕДЬ ЗАПЛАНИРОВАННЫХ ЗВОНКОВ

Для обеспечения возможности хранения и управления очередью звонков необходимо организовать хранение этой информации. Информация должна храниться в виде, готовом для последующего использования, то есть подготовлена для непосредственной отправки в сервис для инициации звонка.

Так как инициация вызовов будет выполняться из 1С, требуется добавить в её конфигурацию таблицу для обеспечения функции очереди звонков. Конфигурация 1С – это представление модели предметной области в виде отдельных структурных блоков (регистров, справочников, документов и др.). Она представляет из себя совокупность этих блоков и программного кода, связывающего их логикой для обеспечения функционирования бизнес-процессов.

При корректировке графика приема врачей, алгоритм в первую очередь проверяет наиболее загруженные медицинские направления, и звонки должны быть выполнены в том же порядке. Поэтому таблица должна быть организована для обеспечения хронологической последовательности исполнения задач и будет имитировать очередь типа FIFO (первый пришел, первый ушел). Структура таблицы представлена на рисунке 20. В терминах 1С, измерение – это ключевое поле, которое контролирует уникальность записи в таблице, таких полей может быть несколько, и вместе они организуют составной уникальный ключ. Реквизиты – данные, которые хранятся в таблице.

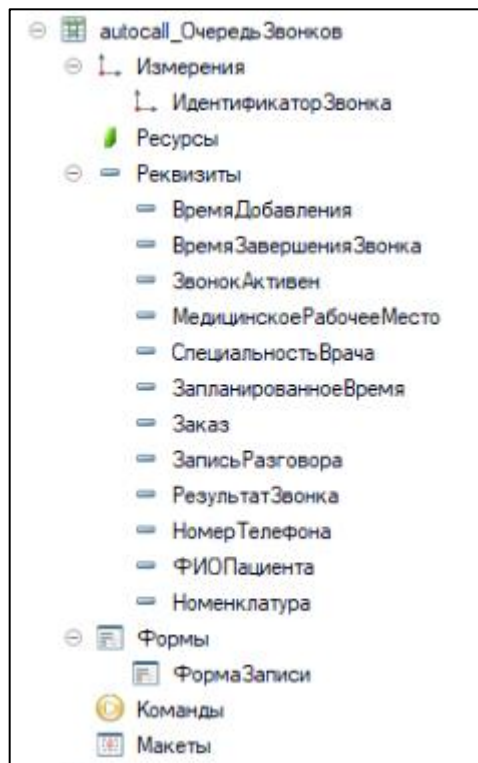


Рис. 20. Структура хранения очереди звонков

3.2. ФОРМИРОВАНИЕ ОЧЕРЕДИ ЗВОНКОВ

«Регламентные задания предназначены для выполнения прикладных задач по расписанию. Регламентные задания хранятся в информационной базе и создаются на основе метаданных, определяемых в конфигурации» [16]. Этот механизм используется для автоматического выполнения таких задач как актуализации системных данных, выполнения конвертаций данных после обновлений структуры данных, очистка таблиц от битых ссылок и др.

Алгоритм выполняет запрос к базе данных и получает из нее информацию по запланированным услугам пациентов и рабочим местам врачей. Группировка данных по рабочему месту и запланированному дню позволяет узнать количество запланированных на день услуг, а присоединение данных о графике работы врача, позволяет определить количество доступных для записи окон.

В случае освобождения места для записи пациента у сотрудника call-центра должно быть достаточно времени, чтобы записать туда другого

пациента, поэтому для формирования очереди будут использованы данные о приемах на четыре дня вперед. Таким образом предоставляя больше возможностей для корректировки расписания.

На основании полученных данных появляется возможность отсортировать приемы по возрастанию доли загрузки рабочих мест. Из результата выборки исключены данные по записям на лабораторные анализы и запланированные приемы стационарных пациентов, потому что в звонках нет необходимости. Программный код формирования очереди и запрос для получения данных представлен в приложении 3.

Для того, чтобы разработанный алгоритм работал автоматически необходимо добавить регламентное задание в конфигурации 1С [16]. Алгоритм формирует список приемов, запланированных через четыре дня от текущей даты. Звонок будет запланирован только если средняя загруженность врача на этот день будет больше 80% и возраст пациента от 16 до 70 лет. Полученные данные помещаются в таблицу запланированных звонков. Схема работы регламентного задания представлена на рисунке 21.



Рис. 21. Процесс формирования плана звонков

3.3. ИНИЦИАЦИЯ ЗАПЛАНИРОВАННЫХ ЗВОНКОВ

Так как на 1С возлагается контроль над инициацией вызовов, для этого должен быть реализован соответствующий функционал. Он будет состоять из регламентного задания и алгоритма, все вместе позволит алгоритму выполняться по расписанию с настраиваемой периодичностью.

Запускаемый по расписанию алгоритм выполняет проверку наличия активных звонков в таблице очереди звонков. Допустимое количество может меняться для того, чтобы выполнять обзвон по нескольким номерам одновременно. После получения очередной задачи на звонок выполняется отправка команды на инициацию вызова и статус звонка в очереди меняется на «выполняется». Позже при получении ответа от приложения голосового взаимодействия звонку будет установлен результат звонка, дата завершения и активность звонка будет установлена в «выполнен» (реквизит «ЗвонокАктивен»). Схема механизма инициации звонков представлена на рисунке 22.

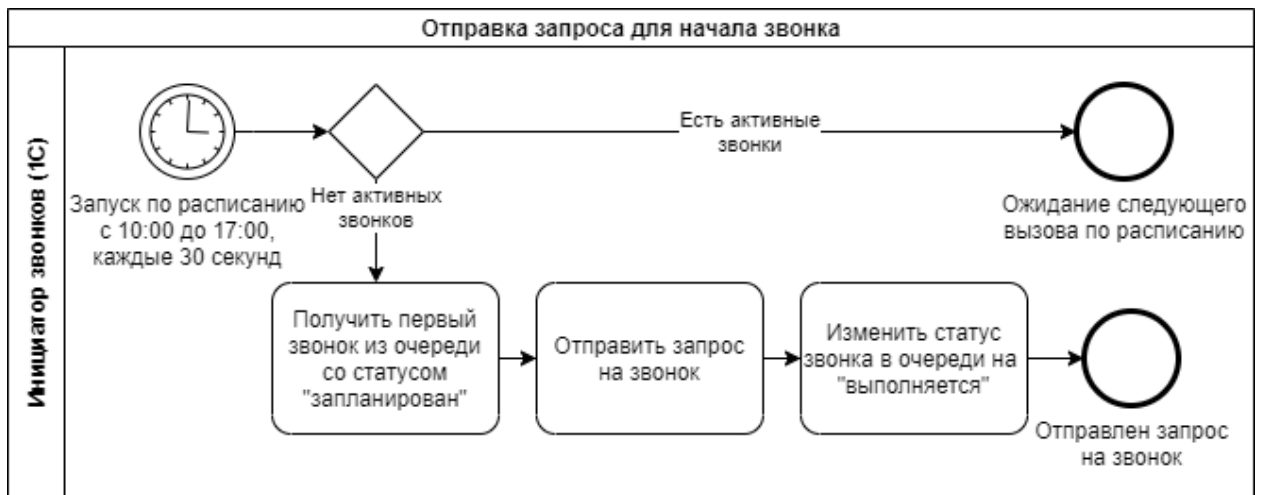


Рис. 22. Отправка запроса на осуществление звонка

Для отправки команды используется встроенный в 1С механизм HTTP-запросов. Он позволяет связать учетную систему с промежуточным API для создания вызовов (описан далее). Для создания вызова помимо номера

телефона пациента должны быть переданы параметры, которые используются при озвучивании параметров записи на прием:

- фамилия имя пациента для подтверждения личности;
- специальность врача;
- запланированное время приема;
- идентификатор звонка для отправки результата в 1С.

Данные передаются в запрос в виде строки в формате JSON (Листинг 1)

```
{
  "doctor_speciality": "Кардиолог",
  "patient_name": "Иванов Иван Иванович",
  "sheduled_time": "15 июня 17:30",
  "dial_number": "687539",
  "call_id": "ad2b87fd-1178-4a28-8738-ee099e9ff1b8"
}
```

Листинг 1. Структура запроса для инициации звонка пациенту

Для того, чтобы связать поставленную задачу на звонок и результат звонка используется уникальный идентификатор, который передается вместе с параметрами в сервис инициации вызова. При получении результата разговора по этому идентификатору выполняется поиск связанной с ним задачи и установка результата в таблице очереди звонков. Идентификатор соответствует уникальному коду запланированной услуги и используется при корректировке расписания. Программный код процесса инициации вызова представлен в приложении 4.

3.4. СОЗДАНИЕ API ДЛЯ ИНИЦИАЦИИ ЗВОНКА ИЗ 1С

Являясь управляющей системой 1С, должна инициировать звонок, но так как этот функционал в ней изначально не предусмотрен, требуется отправить команду той системе, которая сможет. Для этого необходим интерфейс для взаимодействия двух программных систем - API.

В данном случае API будет реализован посредством HTTP-сервиса, являющегося оберткой для функции создания звонка. В листинге 2 представлена команда, которая используется для инициации звонка из CLI (командного интерфейса) Asterisk. Интерфейс будет предоставлять возможность вызова этой команды для других систем.

```
sip-server*CLI> originate SIP/102 application EAGI voice_interaction.py
[дополнительные параметры]
```

Листинг 2. Команда для инициации звонка в CLI Asterisk

Указанная команда содержит следующие параметры:

- `originate` - команда на инициацию вызова;
- `102` – номер, с которым будет установлено соединение (для тестирования выбран внутренний номер);
- `EAGI` – команда Asterisk для запуска приложения;
- `voice_interaction.py` – имя файла с кодом программы, которая будет запущена (см. приложение 1);
- дополнительные параметры – параметры звонка, необходимые для голосового взаимодействия (ФИО пациента, специальность врача и запланированное время приема).

Команда на отправку выполняется из интерфейса командной строки, чтобы вызывать команды из программы, была использована библиотека `Pyst2` [14]. Она позволяет отправлять эти команды из программ на языке Python (Листинг 3).

```
mng.originate(
```



```
channel=f'SIP/{data["dial_number"]}',
exten=None, application='EAGI',
data=f'voice_interaction.py,\
    {data["call_id"]},\
    {data["patient_name"]},\
    {data["doctor_speciality"]},\
    {data["sheduled_time"]}')
```

Листинг 3. Программная инициация вызова

В указанном блоке программы: `data` – структура с параметрами звонка, `mng` – экземпляр класса `Manager`, класса библиотеки `Pyst2` для работы с интерфейсом АМІ.

Для обеспечения возможности вызова программного кода из другой системы необходим API. Он был разработан с использованием библиотеки `Flask` [13] и сделал возможным передачу команд от сторонних по отношению к телефонии систем. Программный код сервиса представлен в приложении 1.

3.5. РЕАЛИЗАЦИЯ ГОЛОСОВОГО ВЗАИМОДЕЙСТВИЯ

Для обеспечения голосового взаимодействия с пациентом, системе телефонии необходимо стартовое событие для начала выполнения алгоритма. Таким стартовым событием является команда на начало звонка, отправляемая из 1С.

Для того чтобы осуществить взаимодействие с абонентом, было реализовано приложение EAGI (Приложение 2). Оно вызывается при инициации получения команды от 1С и позволяет выполнять управление текущим вызовом на любом языке программирования. Данное расширение позволяет из скрипта получить полный контроль над звонком для решения задачи распознавания речи, этот программный модуль позволит получить аудио-поток, который можно использовать для преобразования речи пациента в текст и управлять поведением системы на основании результатов распознавания.

Принцип голосового взаимодействия заключается в синтезе predetermined набора фраз, в которых есть изменяемые части. Например, при проверке кто ответил на звонок робот произносит фразу: «Вы Иванов Иван Иванович?», ФИО пациента в этой фразе динамическое и отправляется из 1С вместе с параметрами записи на прием, которые тоже являются изменяемыми частями во фразе: «вы записаны на прием к *кардиологу* на 17 июня 17:00». Использование сервиса Yandex SpeechKit позволяет синтезировать речь близкую по произношению и интонации к речи человека, но это требует использования «премиальных голосов», которые предоставляются по более дорогому тарифу [18].

При распознавании ответов может возникнуть сложность в распознавании. Даже на простой вопрос, подразумевающий ответы «да» или «нет» может быть получен схожий по смыслу, но отличающийся по написанию ответ. При проведении социологического исследования с использованием автоматического опроса по телефону [20] столкнулись с тем, что вместо «да», могли быть использованы другие слова: хорошо, ага, угу, о'кей, правильно, допустим, предположим да, было дело, возможно, вроде, конечно, может быть, наверно, очень может быть, так точно, таки да.

Также большое количество аналогов для слова «нет». Поэтому для снижения вероятности некорректного распознавания в разговоре робот предлагает пациенту заранее определенные варианты ответа, например, после произнесения вопроса добавляет фразу «ответьте да или нет». Таким образом, человеку значительно проще понять, чего от него ожидает робот и сможет быстрее скорректировать свои ответы при некорректном распознавании.

Для описания хода диалога системы и пациента была построена схема (Рис. 23). Логика обработки ответов, которые не удалось распознать представлена на рисунке 24. В начале разговора с пациентом, голосовой робот предупреждает, что говорит не человек. Это позволяет заранее настроить собеседника на то, чтобы быть готовым к общению с роботом, как это

происходит при общении взрослых с детьми или иностранцами [19]. Таким образом, человек заранее будет говорить более медленно, громко и разборчиво, принимая к выбору только предложенные роботом варианты ответа.

Результаты ответов отправляются в формате JSON. Пример результата звонка представлен в листинге 4.

```
{  
  "call_id": "83f1b8fb-7844-431e-85ef-4988c03d13e2",  
  "call_result": "other_person_answered",  
  "voice_record_link":  
  "http://sip.server/spool/monitor/21062021/21062021093212_102.wav"  
}
```

Листинг 4. Результат звонка

В структуре результата звонка `call_id` – идентификатор звонка, `call_result` – результат звонка, `voice_record_link` – ссылка на запись разговора для прослушивания оператором.

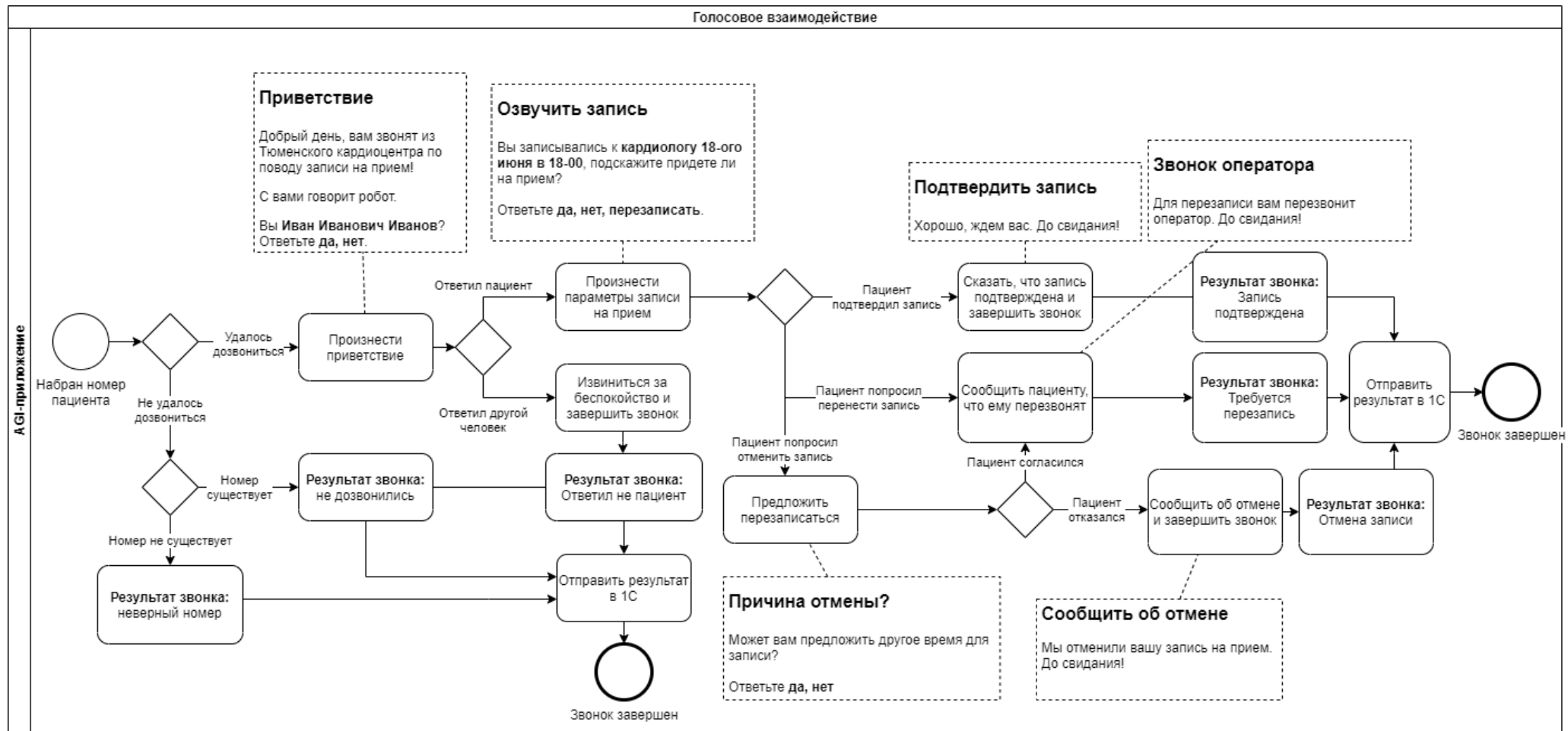


Рис. 23. Схема голосового взаимодействия

Использование упрощенного механизма взаимодействия с использованием predetermined набора ответов упрощает распознавание ответов пациента, но не исключает ошибок. Поэтому для того, чтобы обрабатывать такие ситуации используется дополнительный алгоритм (Рис. 24). После озвучивания вопроса пациенту и получения от него некорректного ответа (отличающийся от предложенных роботом) или не распознанным, выполняется повторная попытка восстановить ход диалога. Голосовой робот сообщает, что не смог распознать ответ человека и просит его повторить, снова предлагая варианты ответа на выбор. При повторении ситуации робот останавливает попытки получить ответ и прощается с пациентом, завершая звонок. Для данной ситуации предусмотрен отдельный вид результата звонка – «серия неразборчивых ответов», который отправляется в 1С вместе с записью разговора для того, чтобы оператор call-центра мог прослушать ее и принять решение о том, нужно ли перезвонить, чтобы уточнить какое решение принял пациент по поводу посещения приема к врачу.

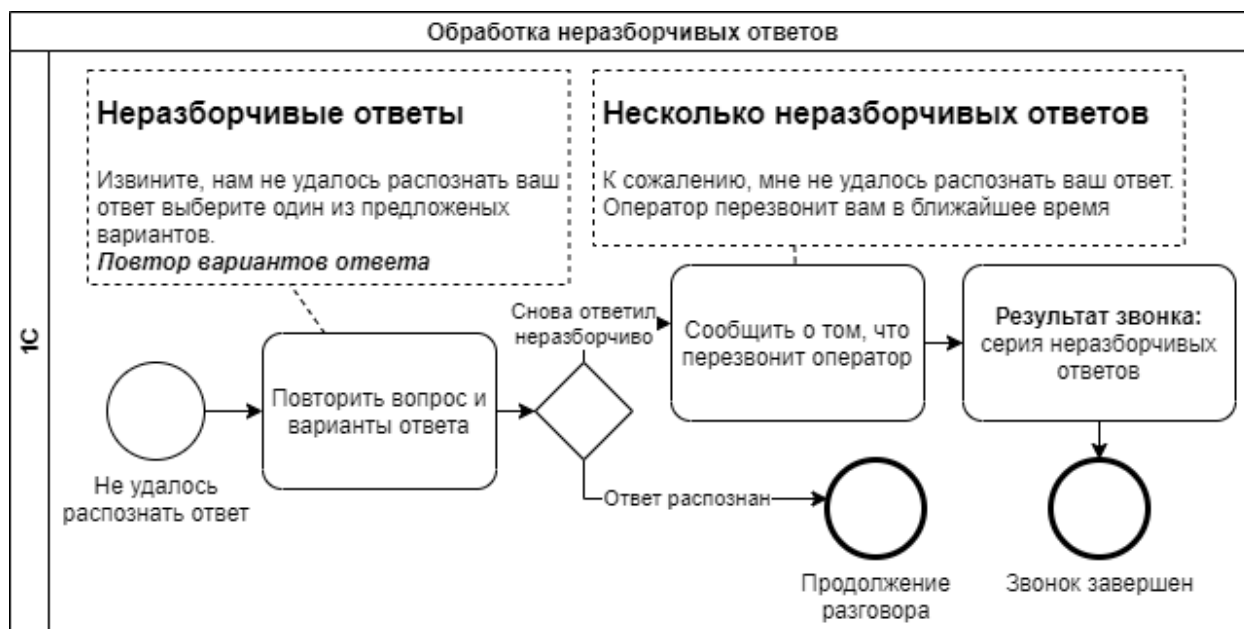


Рис. 24. Обработка неразборчивых ответов

3.6. ОБРАБОТКА РЕЗУЛЬТАТА ЗВОНКА

Как было указано ранее, для сохранения и обработки результатов звонка у системы должна быть возможность получать запросы извне. Для этого 1С будет предоставлять HTTP-сервис для приема HTTP-запросов от AGI – приложения с результатами распознавания ответов пациентов. Схема взаимодействия модулей представлена на рисунке 25. После приема запроса сервис отмечает результат звонка в таблице выполненных звонков и, в случае отказа пациента от приема, отменяет запись в информационной базе. Схема процесса обработки представлена на рисунке 26. После осуществления голосового взаимодействия с пациентом результат отправляется в 1С. На основании результата звонка алгоритм передает дальнейшие действия оператору или выполняет действия самостоятельно (подтверждает или отменяет запись).

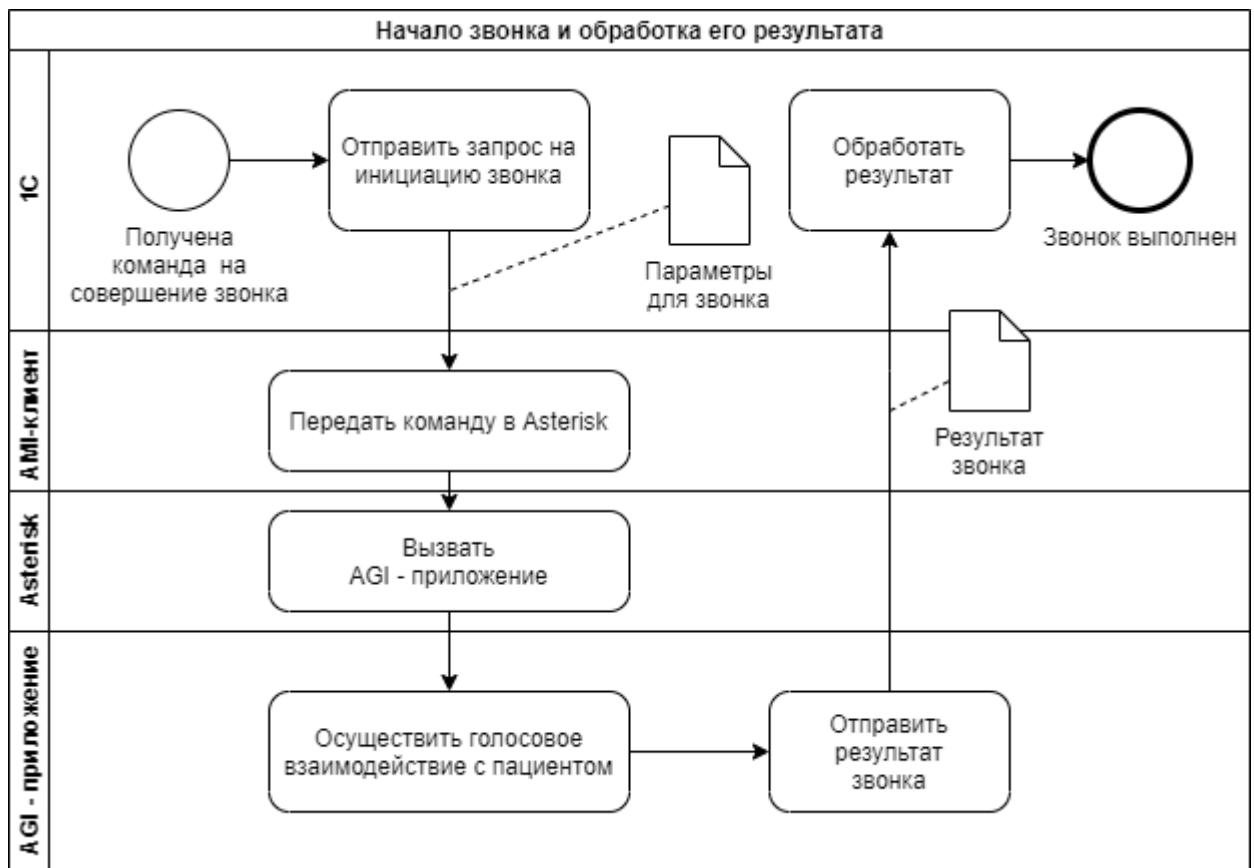


Рис. 25. Начало звонка и его обработка

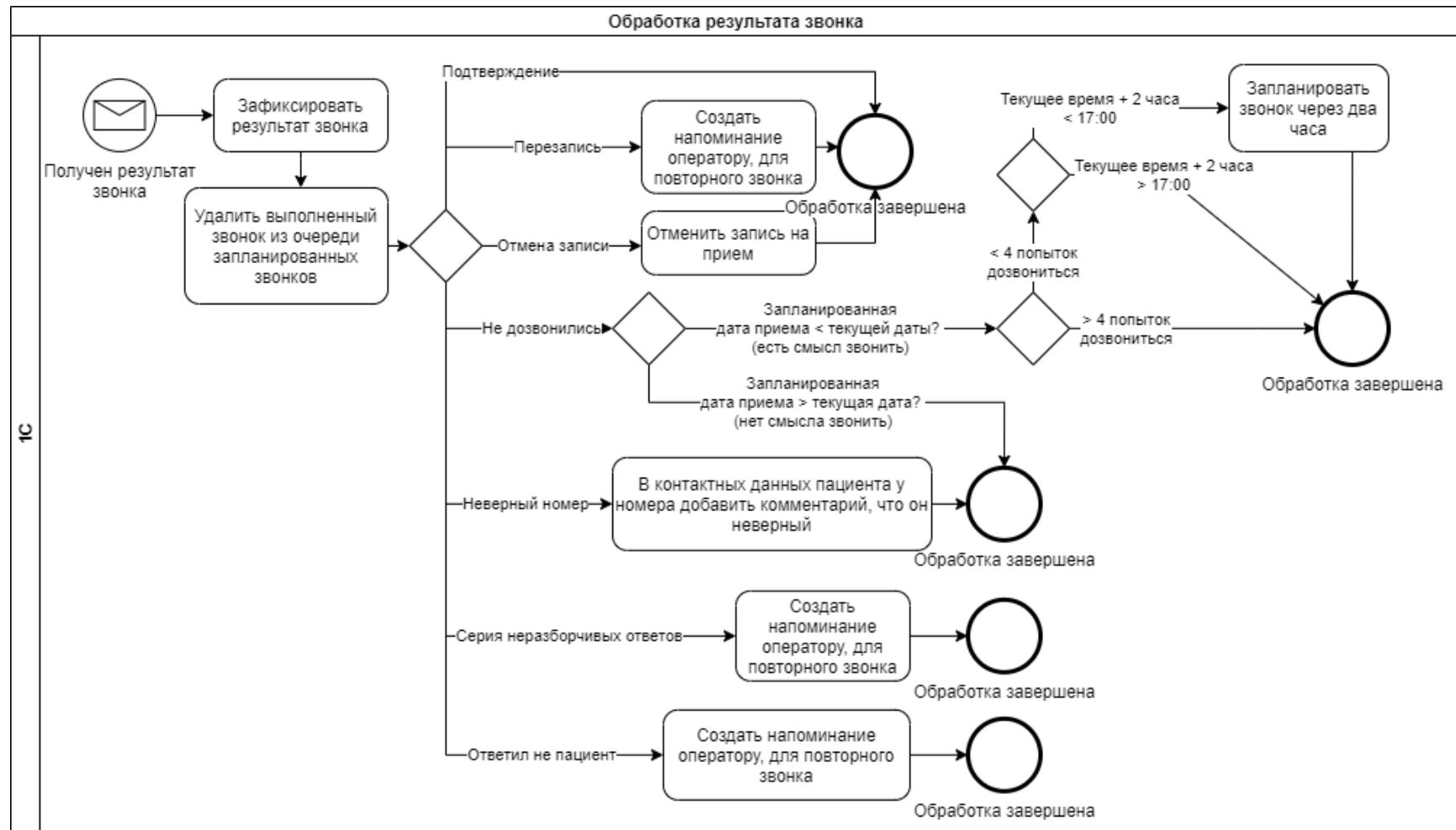


Рис. 26. Обработка результата звонка

Прием результатов осуществляется с помощью публикации HTTP – сервиса, реализованного средствами 1С (Рис. 27). Программный код обработки запроса приведен в приложении 5. При получении входящего запроса выполняется запись в таблицу очереди звонков информации об успешном завершении звонка и его результате:

- подтверждение;
- отмена;
- перезапись;
- ошибки распознавания;
- ответил другой человек.

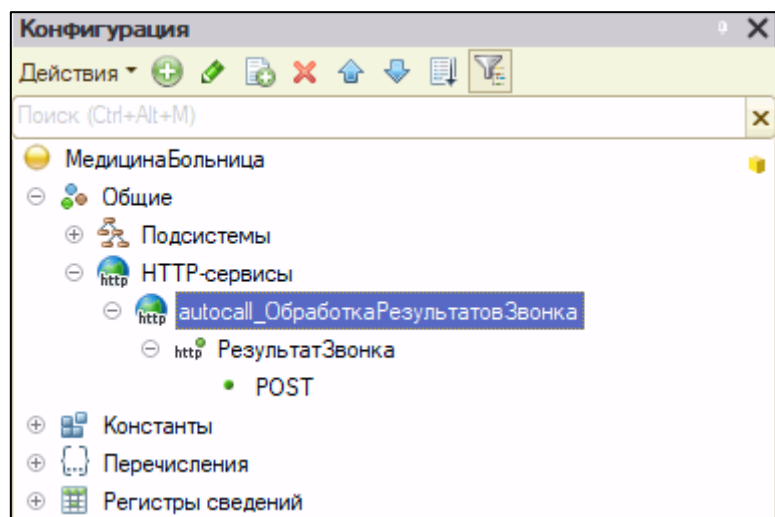


Рис. 27. HTTP-сервис 1С

При добавлении HTTP сервиса в конфигурацию 1С указывается адрес, по которому он будет доступен. Для целей работы добавлена группа методов “auto_call”, а метод для обработки получения результата звонка – “call_result” (Рис. 28, Рис. 29).

Если бы отправка результата звонка выполнялась с использованием утилиты CURL, запрос имел бы вид, аналогичный запросу в листинге 5.

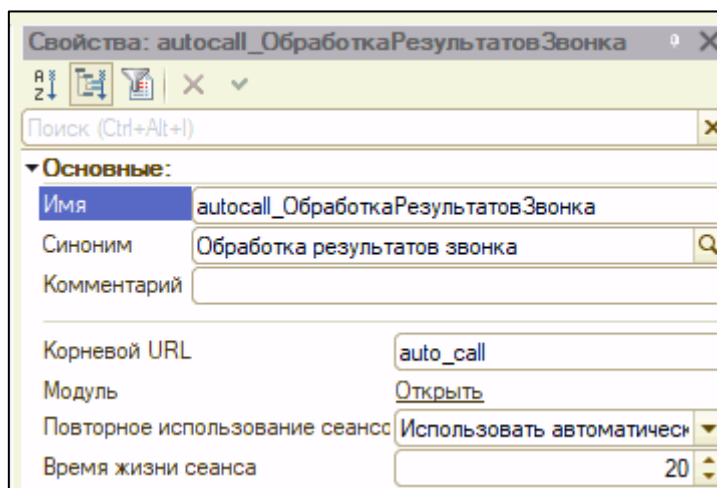


Рис. 28. Корневой URL

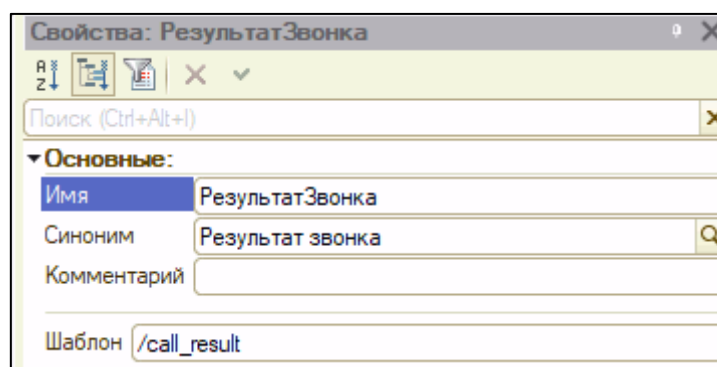


Рис. 29. HTTP-метод для получения результата звонка

```
curl -X POST localhost:8080/hs/hospital/auto_call/call_result -d \
{ \
  "call_id": "83f1b8fb-7844-431e-85ef-4988c03d13e2", \
  "call_result": "other_person_answered", \
  "voice_record_link": \
  http://sip.server/spool/monitor/21062021/21062021093212_102.wav" \
}
```

Листинг 5. Запрос к HTTP-сервису с передачей результатов звонка

Автоматическая корректировка расписания выполняется в случае, если результат звонка – подтверждение или отмена. В случае других вариантов, требуется вмешательство оператора. Сопоставление результата звонка и корректируемого приема выполняется по уникальному идентификатору звонка, который в точности повторяет идентификатор приема в таблице запланированных услуг. Во время обработки запроса с результатом звонка

выполняется поиск услуги в таблице по идентификатору и в зависимости от ответа пациента выполняется подтверждение или отмена. При получении просьбы отменить услугу формируется документ отмены услуг, с комментарием, что отмена выполнена автоматически (рис. 30).

N	A.	Номенклату...	Причина отмены услуги заказа пациента	Назначение услуг	Дата и время	Медицинское рабочее место
1	3...	ЭКГ	Отказались		21.06.2021 11:00:00	ЭКГ, 213 амбулаторные
2	3...	Прием кард.	Отказались		21.06.2021 11:20:00	

Рис. 30. Сформированный документ отмены услуг

Для того чтобы оператор мог своевременно отреагировать на результат звонка, для привлечения его внимания используется механизм всплывающих напоминаний 1С (рис. 31). После появления уведомления сотрудник вручную перезванивает пациенту для выполнения работы, которую роботу выполнить не удалось.

Просмотреть все запланированные и выполненные звонки оператор может в специальном окне (рис. 32). В случаях, когда роботу не удалось распознать ответы пациента, оператор может прослушать запись разговора и самостоятельно определить необходимость повторного звонка пациенту.

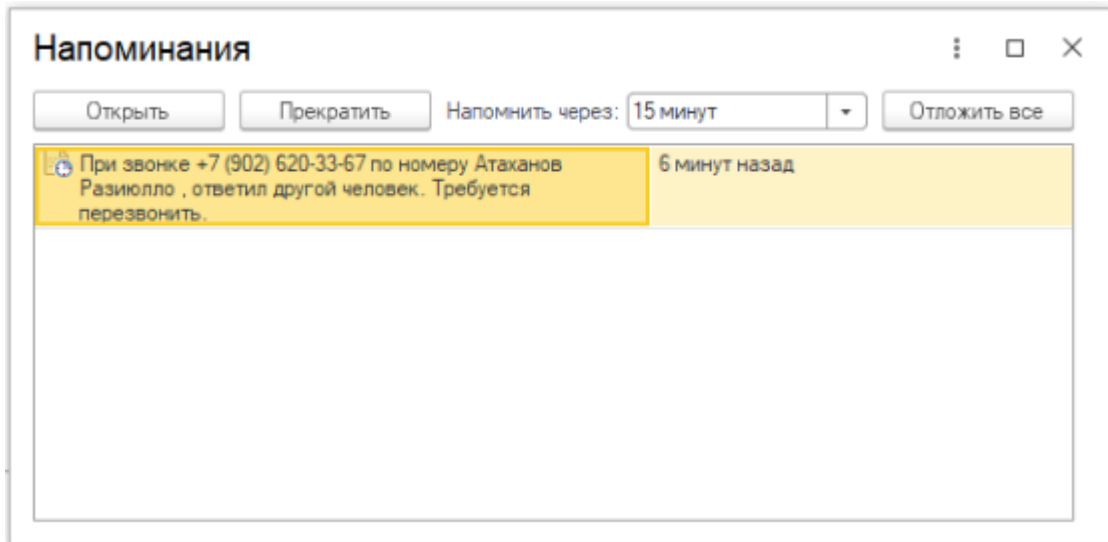


Рис. 31. Уведомление о необходимости вмешательства оператора

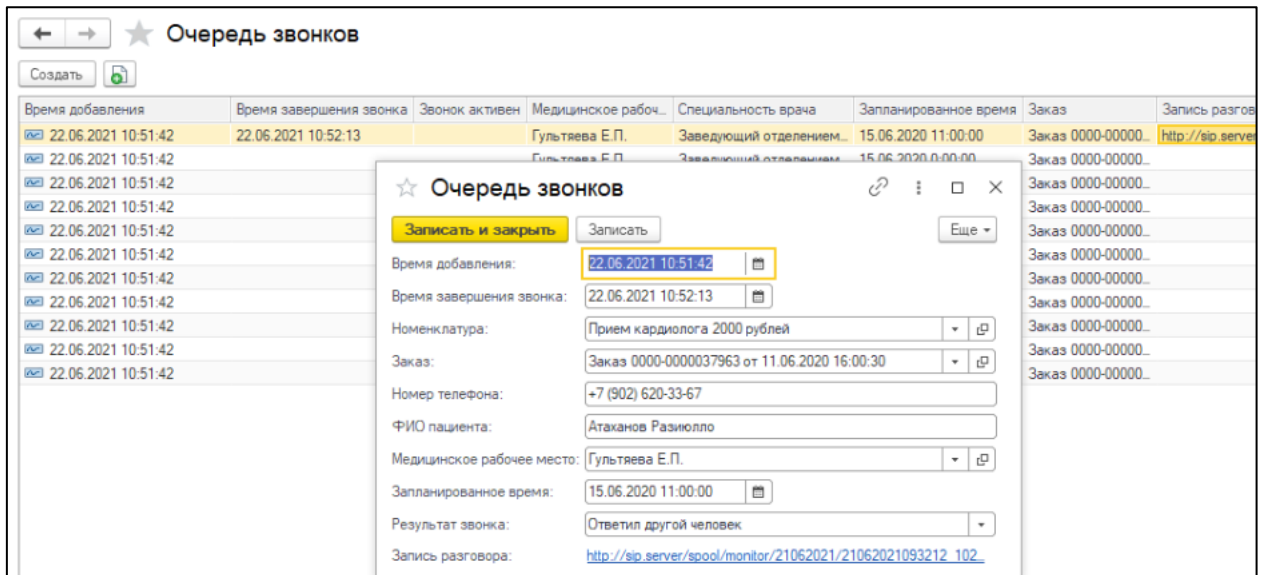


Рис. 32. Просмотр результата звонка

ЗАКЛЮЧЕНИЕ

В результате выполненной работы была определена проблема значительного количества неявок пациентов на приемы. Выявлены недостатки существующего решения:

- большие трудозатраты для совершения обзвона;
- однообразность действий сотрудников.

В качестве возможного альтернативного решения была рассмотрена технология чат-ботов. После проведенного анализа данных Предприятия, данная технология уступила голосовому роботу по причине того, что основная группа пациентов, люди в возрасте от 50 до 80 лет, и голосовой интерфейс взаимодействия для них проще для понимания.

Для выбора подходящего способа коммуникации голосового робота и пациента были изучены возможные варианты взаимодействия, их преимущества и недостатки. В результате было принято решение использовать простой вариант построения диалога, в котором робот одновременно с вопросом предлагает пациенту ограниченный набор вариантов ответа. Это позволяет сделать взаимодействие более простым для понимания и повысить точность распознавания ответов.

Полученные из информационной базы статистические данные об оказанных услугах на Предприятии позволили сформулировать несколько гипотез для оптимизации обзвона пациентов. Целью анализа было выявить закономерности, которые помогли бы ограничить количество совершаемых звонков, оставив только те, что имеют больший приоритет.

Для выполнения этой задачи были изучены данные распределения оказанных услуг по месяцам, дням недели, часам в течение дня. Полученные периодические данные имели закономерности, но не позволили сформулировать критерий для выделения приоритетных для проверки приемов.

В качестве основного параметра отбора был определен коэффициент загрузки врача. Этот коэффициент был использован при получении статистики оказанных услуг по медицинским направлениям, что позволило упорядочить их по степени загруженности. Полученные данные привели к определению порогового значения загрузки равным 80%, являющегося критерием для включения медицинского направления в план автоматической корректировки расписания приемов.

Для формирования второго критерия формирования очереди обзвона было проанализировано распределение оказываемых услуг по возрастам пациентов в результате чего для максимального охвата и снижения недовольства старших возрастных групп пациентов было принято решение выполнять звонки пациентам в возрасте от 16 до 70 лет.

Полученные в результате анализа данных параметры отбора, были использованы при реализации алгоритма формирования очереди звонков для контроля явки на прием.

На основании данных, полученных на предыдущих этапах, была спроектирована и разработана система автоматического голосового взаимодействия, позволяющая заблаговременно выполнять автоматическое выявление и освобождение записей на приемы врачей, которые не будут использованы пациентами по причине их отказа от посещения.

Разработанная для Предприятия система передана в тестирование, полученные результаты будут использованы для ее дальнейшей доработки и совершенствования.

СПИСОК ЛИТЕРАТУРЫ

1. Хрусталева Е. Ю., Язык запросов «1С:Предприятия 8», 2013, ООО «1С-Публишинг»
2. Oracle. Что такое чат-бот? URL: <https://www.oracle.com/ru/chatbots/what-is-a-chatbot/>
3. Как снизить затраты на операторов колл-центров в пять раз: опыт X5 Retail Group и Yandex.Cloud. URL: <https://cloud.yandex.ru/cases/x5-retail-group> (дата обращения: 20.06.2021)
4. Автоматизация бизнеса с применением голосовых и чат-ботов. URL: <https://softline.ru/about/blog/avtomatizatsiya-biznesa-s-primeneniem-golosovyih-i-chat-botov> (дата обращения: 25.06.2021)
5. МТТ. Решения. Опрос/анкетирование.
URL: https://www.mtt.ru/solutions_new/interview/ (дата обращения: 25.06.2021)
6. Анатомия бота: как работают и из чего состоят современные голосовые технологии. URL: <https://rb.ru/opinion/anatomiya-bota/> (дата обращения: 25.06.2021)
7. И снова про 1С. URL: <https://habr.com/ru/company/1c/blog/267321/>
8. Jim Van Meggelen, Jared Smith, Leif Madsen, Asterisk: The Future of Telephony - 2005 O'Reilly Media, Inc.- 408 с.
9. Yandex SpeechKit. Описание. URL: <https://cloud.yandex.ru/docs/speechkit/> (дата обращения: 02.05.2021)
10. Документация Yandex SpeechKit. Синтез речи.
URL: <https://cloud.yandex.ru/docs/speechkit/tts/> (дата обращения: 02.05.2021)
11. Документация Yandex SpeechKit. Распознавание речи.
URL: <https://cloud.yandex.ru/docs/speechkit/stt/> (дата обращения: 02.05.2021)
12. Документация SoX. SoX – Sound eXchange, the Swiss Army knife of audio manipulation URL: <http://sox.sourceforge.net/sox.html> (дата обращения: 02.05.2021)

13. Документация Flask. Быстрый старт. URL: <https://flask.palletsprojects.com/en/2.0.x/quickstart/#a-minimal-application> (дата обращения: 10.05.2021)
14. Документация Pyst2. URL: <https://pyst2.readthedocs.io/en/latest/> (дата обращения: 10.05.2021)
15. Роботы и люди. Что говорит наука о готовности людей получать услуги у роботов. URL: <https://habr.com/ru/post/489120/> (дата обращения: 05.06.2021)
16. Руководство разработчика 1С. Регламентные задания URL: <https://its.1c.ru/db/v837doc#bookmark:dev:TI000000792> (дата обращения: 07.06.2021)
17. Руководство разработчика 1С Концепция системы. URL: <https://its.1c.ru/db/v8317doc#bookmark:dev:TI000000016> (дата обращения: 07.06.2021)
18. Калькулятор стоимости сервисов Yandex Cloud. URL: <https://cloud.yandex.ru/prices> (дата обращения: 01.05.2021)
19. Корбут А.М. «Простите, я никак не могу понять»: способы реагирования на непонимание во взаимодействии человека и робота // Laboratorium: Журнал социальных исследований. 2018. Т. 10. № 3. С. 57–78. DOI: 10.25285/2078-1938-2018-10-3-57-78
20. Рогозин Д. М., Вырская М. С. Коммуникативные сбои и способы их преодоления в роботизированном телефонном опросе // Социологический журнал. 2019. Том 25. № 2. С. 33-59. DOI: <https://doi.org/10.19181/socjour.2019.25.2.6385>

Программный код сервиса для отправки команд в IP-телефонию

```
#!/usr/bin/python3.9

import flask
from asterisk import manager

app = flask.Flask(__name__)

def ami_originate_call(data):
    mng = manager.Manager()
    mng.connect('localhost')
    mng.login('ami', 'ami')
    mng.originate(
        channel=f'SIP/{data["dial_number"]}',
        exten=None, application='EAGI',
        data=f'voice_interaction.py,\
            {data["call_id"]},\
            {data["patient_name"]},\
            {data["doctor_speciality"]},\
            {data["sheduled_time"]}')
    mng.close()

@app.route('/originate', methods=['POST'])
def originate():
    data = flask.request.get_json(force=True)
    if data is None:
        return flask.Response(status=500)
    else:
        ami_originate_call(data)
        return flask.Response(status=200)

if __name__ == '__main__':
    app.run(host='0.0.0.0')
```


Программный код приложения для голосового взаимодействия

```
#!/usr/bin/python3.9

import os
from signal import raise_signal
import sys
from asterisk.agi import AGI, AGIException
import time
import requests
import urllib.request
import json

FOLDER_ID='b1g88t02i0upbnfd9rq7'
IAM_TOKEN='t1.9euelZrIm5XPxszMzs_IyI6ZkJ2Ni-3rnpWakM-
OnJCOj52Sk8_Jkoy0x8b18_dGYT95-
e89FV1C_N3z9wYQPXn57z0VXUL8.2oKcQmXRljVZeYdlZKOZNUji-
74Za9AgP1GSPSq9CQ8zzY8QKWY0oGQSFKB50F_RtxE1k83BPv-02qRAI70Aw'
ONEC_ENDPOINT = 'localhost:8080/auto_call/call_result'
DIR = '/tmp/ivr'

CHUNK_SIZE = 4000

audio_stream = None

def synthesize(folder_id, iam_token, text):
    url = 'https://tts.api.cloud.yandex.net/speech/v1/tts:synthesize'
    headers = {
        'Authorization': 'Bearer ' + iam_token,
    }
    data = {
        'text': text,
        'lang': 'ru-RU',
        'folderId': folder_id,
        'format': 'lpcm',
        'sampleRateHertz': 48000,
        'voice': 'alena'
    }
    with requests.post(url, headers=headers, data=data, stream=True) as resp:
        if resp.status_code != 200:
            raise RuntimeError("Invalid response received: code: %d, message:
%s" % (resp.status_code, resp.text))

        for chunk in resp.iter_content(chunk_size=None):
            yield chunk

app = AGI()
app.verbose('***Launched***')
```

```

if not os.path.exists(DIR):
    os.mkdir(DIR)

def speak(text):
    # Синтез речи
    with open(f'{DIR}/speech.raw', "wb") as f:
        for audio_content in synthesize(FOLDER_ID, IAM_TOKEN, text):
            f.write(audio_content)

    # Кодирование в формат для asterisk
    os.system(f'sox -r 48000 -b 16 -e signed-integer -c 1 {DIR}/speech.raw
{DIR}/speech.sln')
    app.verbose(f'Озвучиваем: {text}')
    app.stream_file(f'{DIR}/speech')

def decode() -> str:
    with open(f"{DIR}/voice_record.ogg", "rb") as fs:
        data = fs.read()

        params = "&".join([
            "topic=general",
            "folderId=%s" % FOLDER_ID,
            "lang=ru-RU"
        ])

        url =
urllib.request.Request("https://stt.api.cloud.yandex.net/speech/v1/stt:recogn
ize?%s" % params, data=data)
        url.add_header("Authorization", "Bearer %s" % IAM_TOKEN)

        responseData = urllib.request.urlopen(url).read().decode('UTF-8')
        decodedData = json.loads(responseData)

        if decodedData.get("error_code") is None:
            return decodedData['result']
        else:
            raise AGIException

def listen(audio_stream) -> str:
    try:
        app.verbose('*** READ ***')
        data = audio_stream.read(CHUNK_SIZE)

        with open(f'{DIR}/rec.sln', 'wb') as record_file:
            start = time.perf_counter()
            while audio_stream.readable():
                end = time.perf_counter()
                record_file.write(data)
                data = audio_stream.read(CHUNK_SIZE)
                time_diff = end - start
                app.verbose(f'time diff: {time_diff}')

            if time_diff > 3:
                break

```

```

os.system(f'sox {DIR}/rec.sln {DIR}/voice_record.ogg')
app.verbose('*** CONVERTED TO OGG ***')

response = decode()
app.verbose(f'Ответ пациента: {response}')
return response.lower()

except Exception as e:
    app.verbose(str(e))
    return ''

if __name__ == '__main__':
    try:
        call_id = sys.argv[1]
        patient_name = sys.argv[2]
        doctor_speciality = sys.argv[3]
        sheduled_time = sys.argv[4]

        call_result = None

        # Открыть аудиопоток
        with os.fdopen(3, 'rb') as audio_stream:
            time.sleep(1)
            speak(f'Здравствуйте. Вам позвонили из тюменского кардиоцентра.
В+ы {patient_name}? Ответьте: - да - или - нет')
            result = listen(audio_stream)

            # Ответил другой человек - кладем трубку
            if 'нет' in result:
                speak('Извините за беспокойство. Хорошего дня')
                time.sleep(1)
                call_result = 'other_man_answered'

            # Ответил пациент - проверим запись
            elif 'да' in result:
                speak(f'Вы записаны к {doctor_speciality} на {sheduled_time}.
Придете ли на прием? Ответьте: -да, - нет или перезаписать')
                result = listen(audio_stream)

                if 'да' in result:
                    speak('Ваша запись подтверждена. Хорошего дня')
                    time.sleep(2)
                    call_result = 'other_man_answered'

                elif 'нет' in result:
                    speak('Может вы хотите записаться на другое время?
Ответьте: - да - или - нет')
                    result = listen(audio_stream)
                    if 'нет' in result:
                        speak('Ваша запись отменена. Хорошего дня')
                        time.sleep(2)
                        call_result = 'cancelled'

                    elif 'да':

```

```
        speak('Хорошо. Для перезаписи вам перезвонит
оператор. Хорошего дня')
        time.sleep(2)
        call_result = 'reassign_needed'

    elif 'перезаписать' in result:
        speak('Хорошо. Для перезаписи вам перезвонит оператор.
Хорошего дня')
        time.sleep(2)
        call_result = 'reassign_needed'

    requests.post(ONEC_ENDPOINT, json={"call_id":call_id,
"call_result":call_result})
    app.verbose('Результат звонка отправлен в 1С')

except Exception as e:
    app.verbose(str(e))
```

Программный код формирования очереди звонков

```

// Формирует очередь звонков на основании данных о записях на прием
//
Процедура СформироватьОчередьЗвонков() Экспорт

    // ПОЛУЧИТЬ СТАТИСТИКУ ПО ЗАНЯТОСТИ СОТРУДНИКОВ
    Запрос = Новый Запрос();
    Запрос.Текст =
        "ВЫБРАТЬ ПЕРВЫЕ 5000
        |     СтатусыУслуг.УникальныйИдентификаторУслуги КАК
УникальныйИдентификаторУслуги,
        |     СтатусыУслуг.Номенклатура КАК Номенклатура,
        |     СтатусыУслуг.СтатусУслуги КАК СтатусУслуги,
        |     СтатусыУслуг.ЗапланированноеВремя КАК ЗапланированноеВремя,
        |     НАЧАЛОПЕРИОДА(СтатусыУслуг.ЗапланированноеВремя, ДЕНЬ) КАК
ЗапланированныйДень,
        |     СтатусыУслуг.Пациент КАК Пациент,
        |     СтатусыУслуг.Контакт КАК Контакт,
        |     СменныеЗадания.МедицинскоеРабочееМесто КАК
МедицинскоеРабочееМесто,
        |     СтатусыУслуг.Заказ КАК Заказ
        | ПОМЕСТИТЬ ВТ_ЗапланированныеПациенты
        | ИЗ
        |     РегистрСведений.СтатусыУслуг КАК СтатусыУслуг
        |     ВНУТРЕННЕЕ СОЕДИНЕНИЕ РегистрСведений.СменныеЗадания КАК
СменныеЗадания
        |     ПО СтатусыУслуг.УникальныйИдентификаторУслуги =
СменныеЗадания.УникальныйИдентификаторУслуги
        | ГДЕ
        |     НАЧАЛОПЕРИОДА(СменныеЗадания.ЗапланированноеВремя, ДЕНЬ) =
&ЗапланированноеВремя
        |     И СменныеЗадания.Номенклатура.ВидНоменклатуры <> &ВидНоменклатуры
        |     И НЕ СменныеЗадания.Номенклатура В ИЕРАРХИИ
        | (&ИсклГруппыНоменклатуры)
        |     И СтатусыУслуг.МедицинскаяКарта.ТипМедицинскойКарты =
&ТипМедицинскойКарты
        | УПОРЯДОЧИТЬ ПО
        |     ЗапланированноеВремя УБЫВ
        | ;
        |
        | //////////////////////////////////////
        | //////////////////////////////////////
        | ВЫБРАТЬ
        |     КОЛИЧЕСТВО(ВТ_ЗапланированныеПациенты.УникальныйИдентификаторУслуги)
КАК КоличествоУслуг,
        |     ВТ_ЗапланированныеПациенты.ЗапланированныйДень КАК
ЗапланированныйДень,
        |     ВТ_ЗапланированныеПациенты.МедицинскоеРабочееМесто КАК
МедицинскоеРабочееМесто

```

```

|ПОМЕСТИТЬ ВТ_ЗагруженностьВрачей
|ИЗ
|      ВТ_ЗапланированныеПациенты КАК ВТ_ЗапланированныеПациенты
|
|СГРУППИРОВАТЬ ПО
|      ВТ_ЗапланированныеПациенты.МедицинскоеРабочееМесто,
|      ВТ_ЗапланированныеПациенты.ЗапланированныйДень
|
|;
|
|////////////////////////////////////
|////////////////////////////////////
|
|ВЫБРАТЬ
|      ГрафикиРабочихМест.МедицинскоеРабочееМесто КАК
МедицинскоеРабочееМесто,
|      ГрафикиРабочихМест.Дата КАК Дата,
|      СУММА(РАЗНОСТЬДАТ(КалендариРасписаниеРаботы.ВремяНачала,
КалендариРасписаниеРаботы.ВремяОкончания, СЕКУНДА) / РАЗНОСТЬДАТ(ДАТАВРЕМЯ(1,
1, 1), ГрафикиРабочихМест.МедицинскоеРабочееМесто.ВремяВыполнения, СЕКУНДА))
КАК КоличествоСлотов
|ПОМЕСТИТЬ ВТ_КоличествоСлотов
|ИЗ
|      РегистрСведений.ГрафикиРабочихМест КАК ГрафикиРабочихМест
|      ВНУТРЕННЕЕ СОЕДИНЕНИЕ Справочник.Календари.РасписаниеРаботы
КАК КалендариРасписаниеРаботы
|      ПО ГрафикиРабочихМест.ГрафикРаботы =
КалендариРасписаниеРаботы.Ссылка
|      И (ДЕНЬНЕДЕЛИ(ГрафикиРабочихМест.Дата) =
КалендариРасписаниеРаботы.НомерДня)
|ГДЕ
|      ГрафикиРабочихМест.Дата = &ЗапланированноеВремя
|      И (КалендариРасписаниеРаботы.УсловиеФункционирования = &АМБ
|      ИЛИ КалендариРасписаниеРаботы.УсловиеФункционирования
= ЗНАЧЕНИЕ(Перечисление.УсловияФункционированияРабочегоМеста.ПустаяСсылка))
|
|СГРУППИРОВАТЬ ПО
|      ГрафикиРабочихМест.МедицинскоеРабочееМесто,
|      ГрафикиРабочихМест.Дата
|
|;
|
|////////////////////////////////////
|////////////////////////////////////
|
|ВЫБРАТЬ
|      ВТ_ЗагруженностьВрачей.МедицинскоеРабочееМесто КАК
МедицинскоеРабочееМесто,
|      ВТ_ЗагруженностьВрачей.ЗапланированныйДень КАК
ЗапланированныйДень,
|      ВТ_ЗагруженностьВрачей.КоличествоУслуг КАК КоличествоУслуг,
|      ВТ_КоличествоСлотов.КоличествоСлотов КАК КоличествоСлотов,
|      ВТ_ЗагруженностьВрачей.КоличествоУслуг /
ВТ_КоличествоСлотов.КоличествоСлотов КАК Загруженность
|ПОМЕСТИТЬ ВТ_ЗагруженностьРабочихМест
|ИЗ
|      ВТ_ЗагруженностьВрачей КАК ВТ_ЗагруженностьВрачей
|      ВНУТРЕННЕЕ СОЕДИНЕНИЕ ВТ_КоличествоСлотов КАК
ВТ_КоличествоСлотов

```

```

|          ПО ВТ_ЗагруженностьВрачей.МедицинскоеРабочееМесто =
ВТ_КоличествоСлотов.МедицинскоеРабочееМесто
|          И ВТ_ЗагруженностьВрачей.ЗапланированныйДень =
ВТ_КоличествоСлотов.Дата
|;
|
|////////////////////////////////////
|////////////////////////////////////
|
|ВЫБРАТЬ
|          ВТ_ЗапланированныеПациенты.Пациент КАК Пациент,
|          ВТ_ЗапланированныеПациенты.МедицинскоеРабочееМесто КАК
МедицинскоеРабочееМесто,
|          ВТ_ЗапланированныеПациенты.УникальныйИдентификаторУслуги КАК
УникальныйИдентификаторУслуги,
|          ВТ_ЗапланированныеПациенты.Номенклатура КАК Номенклатура,
|          ВТ_ЗапланированныеПациенты.СтатусУслуги КАК СтатусУслуги,
|          ВТ_ЗапланированныеПациенты.ЗапланированноеВремя КАК
ЗапланированноеВремя,
|          ВТ_ЗапланированныеПациенты.Контакт КАК Контакт,
|          ВТ_ЗагруженностьРабочихМест.Загруженность КАК Загруженность,
|          КонтактнаяИнформацияПациентаСрезПоследних.Представление КАК
НомерТелефона,
|          МедицинскиеРабочиеМестаИсполнители.Сотрудник.Должность КАК
Должность,
|          ВЫРАЗИТЬ(РАЗНОСТЬДАТ(ДанныеПациентовСрезПоследних.ДатаРождения,
&Сегодня, ДЕНЬ) / 365 КАК ЧИСЛО(3, 0)) КАК Возраст,
|          ВТ_ЗапланированныеПациенты.Заказ КАК Заказ
|ИЗ
|          РегистрСведений.КонтактнаяИнформацияПациента.СрезПоследних КАК
КонтактнаяИнформацияПациентаСрезПоследних
|          ВНУТРЕННЕЕ СОЕДИНЕНИЕ ВТ_ЗапланированныеПациенты КАК
ВТ_ЗапланированныеПациенты
|          ВНУТРЕННЕЕ СОЕДИНЕНИЕ ВТ_ЗагруженностьРабочихМест КАК
ВТ_ЗагруженностьРабочихМест
|          ВНУТРЕННЕЕ СОЕДИНЕНИЕ
Справочник.МедицинскиеРабочиеМеста.Исполнители КАК
МедицинскиеРабочиеМестаИсполнители
|          ПО
ВТ_ЗагруженностьРабочихМест.МедицинскоеРабочееМесто =
МедицинскиеРабочиеМестаИсполнители.Ссылка
|          И
(МедицинскиеРабочиеМестаИсполнители.ОтветственныйЗаНазначение)
|          ПО ВТ_ЗапланированныеПациенты.МедицинскоеРабочееМесто
= ВТ_ЗагруженностьРабочихМест.МедицинскоеРабочееМесто
|          И ВТ_ЗапланированныеПациенты.ЗапланированныйДень
= ВТ_ЗагруженностьРабочихМест.ЗапланированныйДень
|          ВНУТРЕННЕЕ СОЕДИНЕНИЕ
РегистрСведений.ДанныеПациентов.СрезПоследних КАК
ДанныеПациентовСрезПоследних
|          ПО ВТ_ЗапланированныеПациенты.Пациент =
ДанныеПациентовСрезПоследних.Пациент
|          ПО КонтактнаяИнформацияПациентаСрезПоследних.Пациент =
ВТ_ЗапланированныеПациенты.Пациент
|          И (КонтактнаяИнформацияПациентаСрезПоследних.Вид =
&ВидТелефон)

```

```

| ГДЕ
|     VT_ЗагруженностьРабочихМест.Загруженность > &ДоляЗагруженности И
|     VT_ЗагруженностьРабочихМест.Возраст >= &ВозрастНиз
|     И VT_ЗагруженностьРабочихМест.Возраст <= &ВозрастВерх
| УПОРЯДОЧИТЬ ПО
|     Загруженность УБЫВ";
Запрос.УстановитьПараметр("ДоляЗагруженности",
Константы.autocall_ДоляЗагруженностиДляОбзвона.Получить());
Запрос.УстановитьПараметр("ВозрастНиз", 16);
Запрос.УстановитьПараметр("ВозрастВерх", 70);
ЗапланированноеВремя = ТекущаяДатаСеанса() +
МагическиеКонстанты.ОдниСутки() * 4;
Запрос.Параметры.Вставить("ЗапланированноеВремя",
ЗапланированноеВремя);
Запрос.Параметры.Вставить("ТипМедицинскойКарты",
Справочники.ТипыМедицинскихКарт.АМБ); // <Типы медицинских карт>[Медицинская
карта амбулаторного больного], Справочники.ТипыМедицинскихКарт.АМБ;
Запрос.Параметры.Вставить("ВидНоменклатуры",
Справочники.ВидыНоменклатуры.ЛабораторноеИсследованиеАлтэй); // <Вид
номенклатуры >[Лабораторное исследование Алтэй],
Справочники.ВидыНоменклатуры.ЛабораторноеИсследованиеАлтэй;
ИсклГруппыНоменклатуры = Новый СписокЗначений;
ИсклГруппыНоменклатуры.Добавить(Справочники.Номенклатура.НайтиПоНаимено
ванию("Часть 5. Процедурный кабинет")); // <Позиция номенклатуры>[Часть 5.
Процедурный кабинет], Справочники.Номенклатура.НайтиПоНаименованию("Часть 5.
Процедурный кабинет");
ИсклГруппыНоменклатуры.Добавить(Справочники.Номенклатура.НайтиПоНаимено
ванию("Часть 6. Клинико-диагностическая лаборатория ")); // <Позиция
номенклатуры>[Часть 6. Клинико-диагностическая лаборатория ],
Справочники.Номенклатура.НайтиПоНаименованию("Часть 6. Клинико-
диагностическая лаборатория ");
Запрос.Параметры.Вставить("ИсклГруппыНоменклатуры",
ИсклГруппыНоменклатуры); // <Список значений>[Часть 5. Процедурный каби...;
Часть 6. Клинико-диагност...]
Запрос.Параметры.Вставить("АМБ",
Перечисления.УсловияФункционированияРабочегоМеста.Поликлиника); // <Условия
функционирования рабочего места>[Амбулаторно-поликлинические]
Запрос.УстановитьПараметр("ВидТелефон",
Справочники.ВидыКонтактнойИнформации.МобТелефонПациента);
Запрос.УстановитьПараметр("Сегодня", НачалоДня(ТекущаяДата()));
Выборка = Запрос.Выполнить().Выбрать();

// ДОБАВИТЬ ДАННЫЕ В ТАБЛИЦУ
Пока Выборка.Следующий() Цикл
    НовыйЭлементОчереди =
РегистрыСведений.autocall_ОчередьЗвонков.СоздатьМенеджерЗаписи();
    НовыйЭлементОчереди.ВремяДобавления = ТекущаяДатаСеанса();
    НовыйЭлементОчереди.ИдентификаторЗвонка =
Выборка.УникальныйИдентификаторУслуги;
    НовыйЭлементОчереди.ФИОПациента = Строка(Выборка.Пациент);
    НовыйЭлементОчереди.НомерТелефона =
Строка(Выборка.НомерТелефона);
    НовыйЭлементОчереди.СпециальностьВрача = Выборка.Должность;
    НовыйЭлементОчереди.МедицинскоеРабочееМесто =
Выборка.МедицинскоеРабочееМесто;

```



```

        НовыйЭлементОчереди.ЗапланированноеВремя =
Выборка.ЗапланированноеВремя;
        НовыйЭлементОчереди.Номенклатура = Выборка.Номенклатура;
        НовыйЭлементОчереди.Заказ = Выборка.Заказ;
        НовыйЭлементОчереди.Записать();
    КонечЦикла;

КонецПроцедуры // СформироватьОчередьЗвонков()

```

Приложение 4.

Программный код инициации вызова из 1С

```

// Выполняет вызов первому в очереди звонков
//
Процедура ИнициироватьПервыйЗвонокВОчереди() Экспорт

    // Если есть активные звонки, обработка прерывается
    // до очередного старта по расписанию
    КоличествоАктивныхЗвонков = ПолучитьКоличествоНезавершенныхЗвонков();
    Если КоличествоАктивныхЗвонков >= 1 Тогда
        Возврат;
    КонечЕсли;

    ПараметрыЗвонка = ПолучитьПервыйЗвонокВОчереди();

    Соединение = Новый HTTPСоединение("ws0991101", 5000);
    Запрос = Новый HTTPЗапрос("originate");

    ПараметрыЗапроса = Новый Соответствие;
    ПараметрыЗапроса.Вставить("call_id",
Строка(ПараметрыЗвонка.ИдентификаторЗвонка));
    ПараметрыЗапроса.Вставить("dial_number",
ПараметрыЗвонка.НомерТелефона);
    ПараметрыЗапроса.Вставить("patient_name", ПараметрыЗвонка.ФИОПациента);
    ПараметрыЗапроса.Вставить("doctor_speciality",
ПараметрыЗвонка.СпециальностьВрача);
    // Формат даты: 1 июня 17:30
    ПараметрыЗапроса.Вставить("sheduled_time",
        Формат(ПараметрыЗвонка.ЗапланированноеВремя, "ДФ='d ММММ
НН:mm'"));

    СтрокаЗапроса = ПолучитьJSON(ПараметрыЗапроса);
    Запрос.УстановитьТелоИзСтроки(СтрокаЗапроса);
    Ответ = Соединение.ОтправитьДляОбработки(Запрос);

КонецПроцедуры

```

Программный код обработки звонка в 1С

```

// Записывает результат звонка и уведомляет пользователя
// при необходимости его участия
//
Процедура ОбработатьРезультатЗвонка(Запрос) Экспорт

    ТелоЗапроса = Запрос.ПолучитьТелоКакСтроку();
    ПараметрыЗапроса = РазобратьJSON(ТелоЗапроса);

    Запрос = Новый Запрос();
    Запрос.Текст =
        "ВЫБРАТЬ
        |     autocal1_ОчередьЗвонков.ВремяДобавления КАК ВремяДобавления,
        |     autocal1_ОчередьЗвонков.ВремяЗавершенияЗвонка КАК
ВремяЗавершенияЗвонка,
        |     autocal1_ОчередьЗвонков.Заказ КАК Заказ,
        |     autocal1_ОчередьЗвонков.ЗаписьРазговора КАК ЗаписьРазговора,
        |     autocal1_ОчередьЗвонков.ЗапланированноеВремя КАК
ЗапланированноеВремя,
        |     autocal1_ОчередьЗвонков.ЗвонокАктивен КАК ЗвонокАктивен,
        |     autocal1_ОчередьЗвонков.ИдентификаторЗвонка КАК
ИдентификаторЗвонка,
        |     autocal1_ОчередьЗвонков.МедицинскоеРабочееМесто КАК
МедицинскоеРабочееМесто,
        |     autocal1_ОчередьЗвонков.Номенклатура КАК Номенклатура,
        |     autocal1_ОчередьЗвонков.НомерТелефона КАК НомерТелефона,
        |     autocal1_ОчередьЗвонков.РезультатЗвонка КАК РезультатЗвонка,
        |     autocal1_ОчередьЗвонков.СпециальностьВрача КАК
СпециальностьВрача,
        |     autocal1_ОчередьЗвонков.ФИОПациента КАК ФИОПациента
        |ИЗ
        |     РегистрСведений.autocal1_ОчередьЗвонков КАК
autocal1_ОчередьЗвонков
        |ГДЕ
        |     autocal1_ОчередьЗвонков.ИдентификаторЗвонка =
&ИдентификаторЗвонка
        |     И autocal1_ОчередьЗвонков.РезультатЗвонка =
ЗНАЧЕНИЕ(Перечисление.autocal1_РезультатыЗвонка.ПустаяСсылка)";
    Запрос.УстановитьПараметр("ИдентификаторЗвонка", Новый
УникальныйИдентификатор(ПараметрыЗапроса.call_id));
    Выборка = Запрос.Выполнить().Выбрать();

    // Установить результат звонка
    Если Выборка.Следующий() Тогда
        ЗаписьРегистра =
РегистрыСведений.autocal1_ОчередьЗвонков.СоздатьМенеджерЗаписи();
        ЗаполнитьЗначенияСвойств(ЗаписьРегистра, Выборка);
        ЗаписьРегистра.РезультатЗвонка =
Перечисления.autocal1_РезультатыЗвонка[ПараметрыЗапроса.call_result];
        ЗаписьРегистра.ВремяЗавершенияЗвонка = ТекущаяДатаСеанса();

```

```

        ЗаписьРегистра.ЗаписьРазговора =
        ПараметрыЗапроса.voice_record_link;
        ЗаписьРегистра.Записать();
        КонецЕсли;

        // Уведомить менеджера пациента, об изменении в записи
        Если ПараметрыЗапроса.call_result = "other_person_answered" Тогда
            НапоминанияПользователя.УстановитьНапоминание(
                СтрШаблон("При звонке %1 по номеру %2, ответил другой
                человек. Требуется перезвонить.", Выборка.НомерТелефона, Выборка.ФИОПациента)
                ,,, Выборка.Заказ);
            ИначеЕсли ПараметрыЗапроса.call_result = "llegible_replies" Тогда
                НапоминанияПользователя.УстановитьНапоминание(
                    СтрШаблон("При звонке %1 по номеру %2, получены
                    неразборчивые ответы. Требуется перезвонить.", Выборка.НомерТелефона,
                    Выборка.ФИОПациента)
                    ,,, Выборка.Заказ);
            ИначеЕсли ПараметрыЗапроса.call_result = "reassign_needed" Тогда
                НапоминанияПользователя.УстановитьНапоминание(
                    СтрШаблон("При звонке %1 по номеру %2, пациент захотел
                    перезаписаться. Требуется перезвонить.", Выборка.НомерТелефона,
                    Выборка.ФИОПациента)
                    ,,, Выборка.Заказ);
            ИначеЕсли ПараметрыЗапроса.call_result = "cancelled" Тогда
                НапоминанияПользователя.УстановитьНапоминание(
                    СтрШаблон("При звонке %1 по номеру %2, пациент отменил
                    запись.", Выборка.НомерТелефона, Выборка.ФИОПациента)
                    ,,, Выборка.Заказ);
            ИначеЕсли ПараметрыЗапроса.call_result = "accepted" Тогда
                НапоминанияПользователя.УстановитьНапоминание(
                    СтрШаблон("При звонке %1 по номеру %2, пациент подтвердил
                    запись.", Выборка.НомерТелефона, Выборка.ФИОПациента)
                    ,,, Выборка.Заказ);
        КонецЕсли;

        КонецПроцедуры

```