

ООО "РТ МИС"

ЕДИНАЯ ЦИФРОВАЯ ПЛАТФОРМА.МИС 3.0

(ЕЦП.МИС 3.0)

Руководство администратора

Подсистема "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122" 3.0.3_1

Сервис "Подтверждение записи на прием к врачу с возможностью отмены и переноса (исходящий обзвон)" 3.0.3_1

Содержание

1	Введение	4
1.1	Область применения	4
1.2	Краткое описание возможностей	4
1.3	Уровень подготовки администратора.....	4
1.4	Перечень эксплуатационной документации, с которым необходимо ознакомиться администратору	4
2	Общие сведения	5
2.1	Описание функциональных характеристик	5
2.2	Описание информационного взаимодействия.....	6
2.3	Схема взаимодействия	6
2.4	Кэширование запросов.....	6
2.5	HTTP ошибки.....	6
2.5.1	Ошибки на стороне клиента.....	6
2.5.2	Ошибки на стороне сервера	7
2.5.3	Детализация ошибок	7
3	Авторизация	8
3.1	Авторизация в PHP API. Авторизация в системе.....	8
3.2	Авторизация в SWAN API. Метод авторизации пользователя СМП.....	9
4	Описание сценариев сервиса "Подтверждение записи на прием к врачу с возможностью отмены и переноса (исходящий обзвон)"	10
5	Описание методов.....	12
5.1	Общие параметры для всех методов	12
5.2	Описание параметров метода.....	12
5.3	Описание общих кодов ошибок.....	13
5.4	Методы работы с записью пациента на прием	14
5.4.1	Получение списка мест работы по специальности и МО, или по должности и МО GET api/MedStaffFact/MedStaffFactByМО	14
5.4.2	Получение свободных дат приема GET api/TimeTableGraf/TimeTableGrafFreeDate.....	16
5.4.3	Получение свободного времени приема GET api/TimeTableGraf/TimeTableGrafFreeTime.....	17
5.4.4	Запись пациента на прием POST api/TimeTableGraf/TimeTableGrafWrite.....	18
5.5	Методы сервиса "Подтверждение записи на прием к врачу с возможностью отмены и переноса (исходящий обзвон)"	20

5.5.1	Получение списка прошедших и предстоящих занятых бирок в МО за заданный временной промежуток GET api/TimeTableList	20
5.5.2	Получение списка предстоящих приемов пациента в МО GET api/TimeTableListbyPatient	22
5.5.3	Изменение статуса записи на прием PUT api/TimeTableGraf/TimeTableGrafStatus	23
6	Эксплуатация сервиса.....	24
7	Аварийные ситуации	25
7.1	Описание аварийных ситуаций.....	25
7.2	Действия в случае несоблюдения условий выполнения технологического процесса ..	26

1 Введение

1.1 Область применения

Настоящий документ описывает порядок работы с сервисом "Подтверждение записи на прием к врачу с возможностью отмены и переноса (исходящий обзвон)" 3.0.3_1 подсистемы "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122" 3.0.3_1 (далее – Подсистема, подсистема), являющейся частью Единой информационной системы здравоохранения (далее–Система, система).

1.2 Краткое описание возможностей

Подсистема "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122" 3.0.3_1 предназначена для реализации взаимодействия с внешними голосовыми ботами в целях обеспечения функционирования типовых сценариев голосовой коммуникации на номере 122 сторонних решений в части интеграции с ЕЦП.МИС.

Сервис "Подтверждение записи на прием к врачу с возможностью отмены и переноса (исходящий обзвон)" 3.0.3_1 предназначен для подтверждения записи на приём к врачу с возможностью отмены и переноса путём исходящего обзвона с помощью голосового помощника.

1.3 Уровень подготовки администратора

К администраторам Подсистемы предъявляются следующие требования:

- глубокое понимание Подсистемы на уровне технологий работы;
- знание основ администрирования;
- знание основ администрирования реляционных баз данных, поддерживающих клиент-серверный режим;
- навыки реализации различных режимов работы операционных систем;
- навыки администрирования учетных записей пользователей Системы.

1.4 Перечень эксплуатационной документации, с которым необходимо ознакомиться администратору

Перед началом работы администраторам рекомендуется ознакомиться с положениями данного Руководства администратора в части своих функциональных обязанностей.

2 Общие сведения

2.1 Описание функциональных характеристик

Сервис "Подтверждение записи на прием к врачу с возможностью отмены и переноса (исходящий обзвон)" 3.0.3_1 предназначен для подтверждения записи на приём к врачу с возможностью отмены и переноса путём исходящего обзвона с помощью голосового помощника:

- участники процесса:
 - Система:
 - Подсистема "Регистратура";
 - Подсистема "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122";
 - Голосовой помощник (ГП).
- критерий успешности выполнения процесса: данные о записи на прием сохранены в соответствии с действиями пациента (подтверждение записи, отменена или перенос записи);
- базовый сценарий:
 - ГП направляет в Систему запрос на получение списка прошедших и предстоящих занятых бирок в МО за заданный временной промежуток для формирования списка для обзвона;
 - Система возвращает бирки МО;
 - ГП звонит пациенту и уточняет, сможет ли он посетить врача в указанное время;
 - при подтверждении пациентом записи, данные бирки остаются неизменными;
 - при отмене пациентом приёма:
 - ГП направляет в Систему запрос на изменение статуса записи на прием;
 - Система освобождает бирку;
 - при переносе пациентом приема:
 - ГП направляет в Систему запрос на получение списка мест работы по специальности и МО, или по должности и МО;
 - Система возвращает список бирок врача;
 - ГП направляет в Систему запрос на получение свободных дат приема;
 - Система возвращает свободные даты приёма;
 - ГП направляет в Систему запрос GET api/TimeTableGraf/TimeTableGrafFreeTime. на получение свободного времени приема;

- Система возвращает свободные бирки приёма;
- ГП направляет в Систему запрос на запись пациента на прием;
- Система создаёт запись на приём;
- Система возвращает данные о приёме;
- ГП направляет в Систему запрос на изменение статуса записи на прием;
- Система освобождает первоначальную бирку.

2.2 Описание информационного взаимодействия

Для корректного сохранения данных в подсистеме и последующего обращения к ним дополнительно реализуется обмен SOAP и REST-методами.

API построен по архитектуре REST с возвратом результатов в формате JSON.

2.3 Схема взаимодействия

- а) Запрос из сторонней МИС приходит в сервис.
- б) Сервис обрабатывает запрос и вызывает соответствующие методы REST API.
- в) Методы REST API производят необходимые манипуляции с данными (поиск, сохранение, удаление).

Принцип авторизации: авторизация через сервис с последующей авторизацией в Системе.

2.4 Кэширование запросов

При выполнении GET запросов сервер может в HTTP заголовке Last-Modified передать дату последнего изменения получаемых данных. Клиент может передать эту дату в последующих аналогичных запросах в HTTP заголовке If-Modified-Since, если данные не были изменены, то сервер вернет пустую страницу с HTTP кодом 304 Not Modified.

2.5 HTTP ошибки

На каждый запрос к сервису выдается ответ в виде HTTP кода состояния. В случаях ошибок, возникших при выполнении запроса, в теле ответа возвращается код состояния, прикладной код ошибки, сообщение об ошибке.

Общие HTTP ошибки возвращаются HTTP кодами.

- 401 – не передан или неверный apiKey
- 404 – обращение к несуществующему методу

2.5.1 Ошибки на стороне клиента

Ошибки на стороне клиента представлены в таблице 1.

Таблица 1 – Ошибки на стороне клиента

HTTP-код состояния	Прикладной код ошибки	Описание ошибки/Сообщение об ошибке	Примечание
400 Bad Request		Не заполнены обязательные параметры запроса	http-ответ сервиса в случае, если в формат запроса не соответствует требованиям или однако имеется какая-то логическая ошибка, из-за которой невозможно произвести операцию над ресурсом.
401 Unauthorized		Требуется аутентификация	http-ответ сервиса в случае, если пользователь не прошел аутентификацию
405 Method Not Allowed		Метод не поддерживается	http-ответ сервиса в случае, если запрашиваемый метод не поддерживается сервисом
408 Request Timeout			http-ответ в случае, если истекло время ожидания сервером передачи от клиента истекло.

2.5.2 Ошибки на стороне сервера

В случае возникновения ошибки на стороне сервера, при исполнении прикладного метода интеграции, в ответе метода возвращается ошибка с кодом 500.

Имеется возможность получения детальной информации по таким ошибкам.

Для получения детальной информации по ошибкам на стороне сервера, в настройках Web-сервиса должна быть включена отладка:

```
api.yaml
apiErrors:
  debugSave: true
```

Для просмотра подробностей используется метод GET /app/errors (Метод получения детальной информации об ошибке) со значением id ошибки, которую вернул метод, породивший ошибку 500.

2.5.3 Детализация ошибок

Для ошибок на стороне клиента (400) предусмотрена детализация с помощью прикладных кода и текста ошибки.

Для ошибок на стороне сервера (500) предусмотрена детализация с помощью прикладных кода и текста ошибки, а, также, возможность получения подробностей ошибки отдельным методом GET /app/errors (Метод получения детальной информации об ошибке).

3 Авторизация

3.1 Авторизация в PHP API. Авторизация в системе

При авторизации пользователя в системе выполняется проверка наличия группы "Пользователь API". Если у пользователя нет группы, то отобразится ошибка "Нет прав доступа".

Авторизация зависит от используемых механизмов:

- при работе с Rest API применяется базовая (Basic) авторизация:
 - с помощью метода GET /api/user/login определяется идентификатор сессии (sess_id);
 - в последующих вызовах обязательным входящим параметром является идентификатор сессии.
- при использовании SOAP запросов применяется метод WS-security:
 - в секцию <wsse:UsernameToken> всегда передается пара логина и пароля.

При авторизации используется метод GET /api/user/login.

Таблица 2 – Сценарий взаимодействия с ИС "Авторизация в PHP API"

Интегрируемые системы	<ul style="list-style-type: none">– Система:<ul style="list-style-type: none">– модуль АРМ администратора МО– Внешняя ИС
Инициатор сценария:	Внешняя ИС
Предварительные условия:	Внешней ИС выданы учетные данные для авторизации в Системе
Иницирующее событие:	Вызов метода со стороны Внешней ИС
Критерий успешности выполнения:	Авторизация выполнена успешно, в результате работы метода получен идентификатор сессии
Базовые шаги сценария:	<ol style="list-style-type: none">1. Внешняя ИС направляет запрос на авторизацию в Систему – GET api/user/login (Авторизация пользователя в системе).2. Система принимает запрос и осуществляет поиск пользователя по переданным логину и паролю.3. Система направляет во Внешнюю ИС ответ с результатом поиска логина и пароля.
Альтернативные шаги сценария:	<p>3А. При отсутствии логина или пароля, Система направляет информирующий ответ о том, что неверный логин или пароль:</p> <ul style="list-style-type: none">– Ошибка;– Код ошибки –100;– Сообщение об ошибке "Неверный логин или пароль"

3.2 Авторизация в SWAN API. Метод авторизации пользователя СМП

При работе с Rest API применяется базовая (Basic) авторизация:

- с помощью метода GET /login определяется идентификатор сессии (session);
- в последующих вызовах прикладных методов обязательным входящим параметром является идентификатор сессии.

Время жизни сессии задается в настройках сервиса API, по умолчанию равно 1800 секунд.

Используемый метод – **GET /login**.

Таблица 3 – Сценарий взаимодействия с ИС "Авторизация в SWAN API"

Интегрируемые системы	<ul style="list-style-type: none">– Система– Внешняя ИС
Инициатор сценария:	Внешняя ИС
Предварительные условия:	Сервис развернут. Интеграция включена.
Иницирующее событие:	Вызов прикладного интеграционного метода со стороны внешней ИС
Критерий успешности выполнения:	Авторизация пользователя выполнена. Последующие вызовы интеграционных методов успешны. В случае возникновения ошибок выполнения методов - ошибки возвращаются в описанном формате
Базовые шаги сценария:	<ol style="list-style-type: none">1. Внешняя ИС формирует параметры и направляет в ЕЦП запрос GET /rest-api/login (Метод авторизации пользователя) на авторизацию пользователя Системы.2. При возникновении ошибок на стороне сервера, при выполнении методов, внешняя ИС формирует параметры и направляет в Систему запрос GET /app/errors (Метод получения детальной информации об ошибке) на получение детальной информации об ошибке

4 Описание сценариев сервиса "Подтверждение записи на прием к врачу с возможностью отмены и переноса (исходящий обзвон)"

Таблица 4 – Описание сценария "Подтверждение записи на прием к врачу (исходящий обзвон)"

Интегрируемые системы	<ul style="list-style-type: none"> – Система: <ul style="list-style-type: none"> – Подсистема "Регистратура"; – Подсистема "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122". – Голосовой помощник (ГП)
Инициатор сценария:	Голосовой помощник (ГП)
Предварительные условия:	<ul style="list-style-type: none"> – ГП авторизован в РНР API Системы; – в БД создана запись к врачу со следующими данными: <ul style="list-style-type: none"> – идентификатор МО; – наименование МО; – ИД пациента; – фамилия пациента; – имя пациента; – отчество пациента; – номер телефона пациента; – идентификатор бирки
Иницирующее событие:	ГП направляет в Систему запрос GET api/TimeTableList (Получение списка прошедших и предстоящих занятых бирок в ЛПУ за заданный временной промежуток) для формирования списка для обзвона
Критерий успешности выполнения:	<p>Данные бирки сохранены в соответствии с действиями пациента</p> <p>Варианты:</p> <ul style="list-style-type: none"> – если пациент подтверждает запись на приём, данные остаются прежними; – если пациент отменяет запись, бирка освобождается; – если пациент отменяет запись или переносит её: <ul style="list-style-type: none"> – создаётся новая запись на приём; – старая бирка освобождается
Базовые шаги сценария:	<ol style="list-style-type: none"> 1. ГП направляет в Систему запрос GET api/TimeTableList (Получение списка прошедших и предстоящих занятых бирок в ЛПУ за заданный временной промежуток) для формирования списка для обзвона. 2. Система возвращает бирки МО 3. ГП звонит пациенту и уточняет, сможет ли он посетить врача в указанное время. 4. Данные бирки остаются неизменными

<p>Альтернативные шаги сценария:</p>	<p>Пациент отменяет приём 4А. ГП направляет в Систему запрос PUT api/TimeTableGraf/TimeTableGrafStatus (Изменение статуса записи на прием) с параметром EvnStatus_id=12. 5А. Система освобождает бирку</p>
	<p>Пациент переносит приём 4Б. ГП направляет в Систему запрос GET api/MedStaffFact/MedStaffFactByMO (Получение списка мест работы по специальности и МО, или по должности и МО). 5Б. Система возвращает список бирок врача. 6Б. ГП направляет в Систему запрос GET api/TimeTableGraf/TimeTableGrafFreeDate (Получение свободных дат приема). 7Б. Система возвращает свободные даты приёма. 8Б. ГП направляет в Систему запрос GET api/TimeTableGraf/TimeTableGrafFreeTime (Получение свободного времени приема). 9Б. Система возвращает свободные бирки приёма. 10Б. ГП направляет в Систему запрос POST api/TimeTableGraf/TimeTableGrafWrite (Запись пациента на прием). 11Б. Система создаёт запись на приём. 12Б. Система возвращает данные о приёме. 13Б. ГП направляет в Систему запрос PUT api/TimeTableGraf/TimeTableGrafStatus (Изменение статуса записи на прием) с параметром EvnStatus_id=12. 14Б. Система освобождает бирку</p>

5 Описание методов

5.1 Общие параметры для всех методов

Входящие параметры передаются или через строку запроса (query string), или непосредственно записываются в путь(path) URL, если это указано в названии метода:

- apiKey – ключ API;
- sess_id – идентификатор сессии. Не обязателен для методов, не требующих авторизации пользователя;
- offset – смещение для порции записей.

Выходные данные, в случае если возвращен код HTTP 200 представляют JSON объект со следующими полями:

- error_code – код ошибки;
- error_msg – сообщение с подробностями ошибки;
- count – количество записей в ответе, в случае если ответ неполон;
- offset – смещение текущей возвращаемой порции записей, если оно ненулевое;
- data – массив объектов, зависящих от запроса, может содержать 1 и больше элементов.

5.2 Описание параметров метода

- title, содержащий текстовое название параметра;
- type, указывающий тип значения параметра; указывается символ формата, а вслед за ним в квадратных скобках максимальная длина атрибута; символы формата соответствуют вышеописанным обозначениям:
 - T – <текст>;
 - N – <число>;
 - D – <дата> в формате ГГГГ–ММ–ДД;
 - DT – <дата и время> в формате ГГГГ–ММ–ДД чч:мм:сс;
 - V – <время> в формате чч:мм:сс;
 - S – <элемент>; составной элемент, (список/массив), описывается отдельно.
- description, содержащий текстовое описание параметра;
- required, указывающий на то, что данный параметр является обязательным (допускается указать несколько символов, например OM);
 - O – обязательный реквизит, который должен обязательно присутствовать в элементе;

- Н – необязательный реквизит, который может, как присутствовать, так и отсутствовать в элементе. При отсутствии, не передается;
- У – условно–обязательный реквизит. При отсутствии, не передается;
- М – реквизит, определяющий множественность данных, может добавляться к указанным выше символам.

Используются русские символы

- default, содержащий значение параметра по умолчанию.

Пример описания параметра

- title (type, required, default) – description
- email (T[60],OM,1@mail.ru) – электронная почта, уникальная

5.3 Описание общих кодов ошибок

- 0 – успешный запрос, отсутствие ошибки;
- 1 – метод требует авторизации, а сессия не передана;
- 2 – неверный идентификатор сессии, просроченная сессия;
- 3 – не передан какой–то из обязательных параметров, в error_msg – расшифровка;
- 4 – ошибка во входящих параметрах, в error_msg – расшифровка;
- 5 – попытка изменения данных в БД под демо–пользователем.

5.4 Методы работы с записью пациента на прием

5.4.1 Получение списка мест работы по специальности и МО, или по должности и МО GET api/MedStaffFact/MedStaffFactByMO

Правила метода:

- для параметров offset, limit, page – стандартные правила постраничного просмотра;
- если указан параметр offset, то параметр page игнорируется;
- при использовании page необходимо учесть, что limit имеет значение по умолчанию, поэтому с &page=3 будет искать третью страницу с числом записей по 100 на каждой;
- возвращаемый методом список упорядочен по значению MedStaffFact_id;
- если не указан ни один параметр: offset, limit, page, то метод будет работать без учета правил.

Таблица 5 – Параметры метода GET api/MedStaffFact/MedStaffFactByMO

Входящие параметры	<ul style="list-style-type: none"> – MedSpecOms_id (N, O) – идентификатор специальности; – Lpu_id (N, O) – идентификатор МО; – Post_id (N, O) – идентификатор должности; – MedStaffFact_IsPaidRec (N, H) – признак наличия платного приёма; – TimeTableType_id (N, H) – тип бирки; – offset (N, H) – номер начальной возвращаемой записи; – limit (N, H) – максимальное число возвращаемых записей; – page (N, H) – номер страницы
Ответ	<p>Успешный ответ – массив объектов:</p> <ul style="list-style-type: none"> – MedStaffFact_id (N, O) – место работы врача; – Person_id (N, O) – идентификатор человека по месту работы; – PersonSurName_SurName (S[50], H) – фамилия; – PersonFirName_FirName (S[50], H) – имя; – PersonSecName_SecName (S[50], H) – отчество. – RecType_id (N, H) – тип записи; – LpuUnitType_id (N, O) – тип подразделения МО; – MedStaffFact_setDate (D, O) – дата начала работы сотрудника; – MedStaffFact_disDate (D, H) – дата окончания работы сотрудника; – LpuBuilding_id (N, O) – идентификатор; – Post_id (N, H) – идентификатор должности;

	<ul style="list-style-type: none"> – Lpu_id (N, H) – идентификатор МО; – LpuSection_id (N, H) – идентификатор отделения; – TimetableGraf_Count (N, H) – количество доступных (свободных) бирок с указанным типом для определенного врача. Если параметр TimeTableType_id имеет пустое значение или значение в параметре не указано, то в параметре TimetableGraf_Count будет указано количество свободных бирок с любым типом
Пример	<pre> { "error_code": 0, "data": [{ "MedStaffFact_id": "140101000000013", "Person_id": "101010000000060", "PersonSurName_SurName": "БАНАЛЬНЫЙ", "PersonFirName_FirName": "ФЕДОР", "PersonSecName_SecName": "ИВАНОВИЧ", "RecType_id": null, "LpuUnitType_id": "1", "MedStaffFact_setDate": "01.03.2020", "MedStaffFact_disDate": null, "LpuBuilding_id": "10101000000194" }, { "MedStaffFact_id": "140101000000014", "Person_id": "101010000000060", "PersonSurName_SurName": "БАНАЛЬНЫЙ", "PersonFirName_FirName": "ФЕДОР", "PersonSecName_SecName": "ИВАНОВИЧ", "RecType_id": null, "LpuUnitType_id": "2", "MedStaffFact_setDate": "01.02.2020", "MedStaffFact_disDate": null, "LpuBuilding_id": "10101000000177" }] } </pre>

	<pre> "MedStaffFact_id": "140101000001993", "Person_id": "101010000000002", "PersonSurName_SurName": "ВРАЧ", "PersonFirName_FirName": "ТЕСТ", "PersonSecName_SecName": "", "RecType_id": null, "LpuUnitType_id": "9", "MedStaffFact_setDate": "01.01.2021", "MedStaffFact_disDate": null, "LpuBuilding_id": "101010000000074" }] } </pre>
--	--

5.4.2 Получение свободных дат приема GET api/TimeTableGraf/TimeTableGrafFreeDate

Таблица 6 – Параметры метода GET api/TimeTableGraf/TimeTableGrafFreeDate

Входящие параметры	<ul style="list-style-type: none"> – MedStaffFact_id (N, O) – место работы врача – TimeTableGraf_beg (D, O) – дата начала диапазона – TimeTableGraf_end (D, O) – дата окончания диапазона – TimeTableGrafFreeDate (N, H) – параметр учета прошедшего времени (1 – да, 0 – нет); – TimeTableType_id (N, H) – тип бирки: <ul style="list-style-type: none"> – если указано значение входного параметра – ответ со списком свободных дат, на которых есть бирки с указанным типом; – если пустое значение или не указан – метод работает по старым правилам
Ответ	<p>Успешный ответ – список свободных дат</p> <ul style="list-style-type: none"> – TimeTableGraf_begTime (D, O) – даты, где есть бирки со статусом "свободная". Если TimeTableGrafFreeDate=1 и свободные бирки на текущую дату отсутствуют, дата не возвращается.

5.4.3 Получение свободного времени приема GET api/TimeTableGraf/TimeTableGrafFreeTime

Таблица 7 – Параметры метода GET api/TimeTableGraf/TimeTableGrafFreeTime

<p>Входящие параметры</p>	<ul style="list-style-type: none"> – MedStaffFact_id (N, O) – место работы врача; – TimeTableGraf_begTime (D, O) – свободная дата приема; – Person_id (N, H) – идентификатор пациента. Правила: <ul style="list-style-type: none"> – если Person_id не указан, то возвращаются все виды бирок, кроме вида "Диспансерный учёт"; – если Person_id указан, и у пациента нет открытой карты диспансерного наблюдения, то возвращаются все виды бирок, кроме вида "Диспансерный учёт"; – если Person_id указан и у пациента есть открытая карта диспансерного наблюдения, то возвращаются все виды бирок, в том числе вида "Диспансерный учёт"; – TimeTableType_id (N, H) – тип бирки; – TimeTableGrafFreeDate (N, H) – параметр учета прошедшего времени ("1" – да, "0" – нет). Если значение не указано или указано "0", то метод возвращает бирки со временем начала, большим текущего времени. Если в параметре передано значение "1", то метод возвращает все бирки за указанную дату; – offset (N, H) – номер начальной возвращаемой записи; – limit (N, H) – максимальное число возвращаемых записей. Значение по умолчанию – 10000; – page (N, H) – номер страницы. Значение по умолчанию – 1
<p>Ответ</p>	<p>Успешный ответ – массив свободных бирок на прием к врачу: {</p> <ul style="list-style-type: none"> – TimeTableGraf_id (N, H) – идентификатор свободной бирки. Если по пациенту (Person_id) найдена открытая контрольная карта диспансерного наблюдения в отделении МО места работы врача и специальность врача совпадает со специальностью ответственного врача, указанного в карте дисп. наблюдения, то дополнительно передаются идентификаторы свободных бирок с типом "Диспансерный учёт"; – TimeTableGraf_begTime (DT, O) – дата и время начала приема; – TimeTableGraf_Time (N,O) – длительность приема.

	<p>} Выходной параметр для случаев, когда бирка, на которую пытаются записаться, заблокирована:</p> <ul style="list-style-type: none"> - TimeTableStatusLockComment (Т, Н) – сообщение о блокировке бирки. Сообщение содержит в себе возможное время разблокировки бирки. <p>Проверка на наличие переданной во входных параметрах бирки в таблице БД по TimeTableMedService_id:</p> <ul style="list-style-type: none"> - если TimeTableMedService_id есть в БД, записывать на бирку нельзя. Производится подсчет минут от даты и времени, указанных в БД исходя из логики, что бирка может быть заблокирована оператором 15 минут. Соответственно, если TimeTableLock_lockTime=2017-12-23 21:59:34.260000, то бирка станет доступной для записи или занятой в 2017-12-23 22:14:34.260000; - в связи с данной проверкой для бирок, которые нашлись в таблице БД выводится в ответе параметр: - TimeTableStatusLockComment, в котором будет содержаться следующая информация: "На данный момент бирка редактируется оператором и может быть занята. Время возможного высвобождения бирки и доступности записи на нее 2017-12-23 21:59:34.260000" (т.е. в конце комментария выводится время из TimeTableLock_lockTime + 15 мин)
--	---

5.4.4 Запись пациента на прием POST api/TimeTableGraf/TimeTableGrafWrite

Таблица 8 – Параметры метода POST api/TimeTableGraf/TimeTableGrafWrite

Входящие параметры	<ul style="list-style-type: none"> - Person_id (N, O) – идентификатор пациента; - TimeTableGraf_id (N, O) – идентификатор свободной бирки; - EvnQueue_id (N, H) – идентификатор постановки в очередь
Ответ	<ul style="list-style-type: none"> - если по переданному значению Person_id не найден пациент, то ошибка "Пациент не найден в системе". - если по переданному значению TimeTableGraf_id не найдена свободная бирка, то ошибка "Не найдена свободная бирка". - если по переданному значению EvnQueue_id не найдена запись в БД, то ошибка "Не найдена запись о постановке в очередь". - если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например: <ul style="list-style-type: none"> - ожидается числовое значение, а передано текстовое значение; - ожидается параметр, состоящий из 11 символов, а

	<p>передан параметр, состоящий из большего или меньшего количества символов то, ошибка: "Неверный формат поля <Наименования полей через запятую>", запись не производится.</p> <ul style="list-style-type: none"> - если запись на переданную в методе бирку для данного пациента найдена в БД, то ошибка "Человек уже записан на данную бирку", запись не производится. <p>Выходной параметр для случаев, когда бирка, на которую пытаются записаться, заблокирована:</p> <ul style="list-style-type: none"> - TimeTableStatusLockComment (Т, Н) – сообщение о блокировке бирки. Сообщение содержит в себе возможное время разблокировки бирки. <p>Проверка на наличие переданной во входных параметрах бирки в таблице БД по TimeTableMedService_id:</p> <ul style="list-style-type: none"> - если TimeTableMedService_id есть в БД, записывать на бирку нельзя. Производится подсчет минут от даты и времени, указанных в БД исходя из логики, что бирка может быть заблокирована оператором 15 минут. Соответственно, если TimeTableLock_lockTime=2017-12-23 21:59:34.260000, то бирка станет доступной для записи или занятой в 2017-12-23 22:14:34.260000; - в связи с данной проверкой для бирок, которые нашлись в таблице БД выводится в ответе параметр: <ul style="list-style-type: none"> - TimeTableStatusLockComment, в котором будет содержаться следующая информация: "На данный момент бирка редактируется оператором и может быть занята. Время возможного высвобождения бирки и доступности записи на нее 2017-12-23 21:59:34.260000" (т.е. в конце комментария выводится время из TimeTableLock_lockTime + 15 мин). <p>Успешный ответ – составной идентификатор записи на прием:</p> <ul style="list-style-type: none"> - Person_id (N, O), TimeTableGraf_id (N, O) – составной идентификатор записи на прием. <p>Если во входящих параметрах указан идентификатор постановки в очередь, то в направление передается TimeTableGraf_id, а Evn_Queue_id удаляется.</p>
--	---

5.5 Методы сервиса "Подтверждение записи на прием к врачу с возможностью отмены и переноса (исходящий обзвон)"

5.5.1 Получение списка прошедших и предстоящих занятых бирок в МО за заданный временной промежуток GET api/TimeTableList

Метод предназначен для получения списка прошедших и предстоящих занятых бирок в МО за заданный временной промежуток. Поиск визитов производится в рамках МО вызова метода.

Метод разработан для получения списка занятых бирок с информацией о том, кто записан на бирку, контакт пациента с целью обзвона для подтверждения записи/визита пациента в МО.

Таблица 9 – Параметры метода GET api/TimeTableList

<p>Входящие параметры</p>	<ul style="list-style-type: none"> – Lpu_id (N, H) – идентификатор медицинской организации; – TimeTable_beg (DT, H) – дата и время начала диапазона поиска (если не указано, то считать текущие время и дату за дату начала диапазона); – MedStaffFact_id (N, H) – идентификатор места работы. Правила передачи: <ul style="list-style-type: none"> – если передано значение и в параметре TimeTableSource передано значение "MedService", то метод вернёт бирки в соответствии с указанным типом (на службу), переданные значения параметра MedStaffFact_id будут проигнорированы; – если передано значение и при этом параметр TimeTableSource не передан, то метод вернёт все бирки с типом "на службу" и те бирки с типом "к врачу", которые соответствуют переданным значениям параметра MedStaffFact_id; – Post_id (N, H) – идентификатор должности. Правила передачи: <ul style="list-style-type: none"> – если передано значение и в параметре TimeTableSource передано значение "MedService", то метод вернёт бирки в соответствии с указанным типом (на службу), переданные значения параметра Post_id будут проигнорированы; – если передано значение и при этом TimeTableSource не передан, то метод вернёт все бирки с типом "на службу" и те бирки с типом "к врачу", которые соответствуют переданным в параметре Post_id. – TimeTable_end (DT,H) – дата и время окончания диапазона; – TimeTableSource (T, H) – тип бирки. Возможные значения: Graf/MedService для получения бирок на прием к врачу/на услугу службы соответственно; – offset (N,H,0) – номер начальной возвращаемой записи; – limit (N,H,100) – максимальное число возвращаемых записей;
----------------------------------	---

	<p style="text-align: center;">– Page (N,H,1) – номер страницы.</p> <p>Ответ</p> <p>Успешный ответ – список бирок, на которые записаны пациенты, в заданной МО за заданный диапазон дат.</p> <p>Если у УЗ пользователя есть группа "Вендор API" И в запросе была указана МО (параметр Lpu_id), то в ответе будут переданы записи только по выбранной МО.</p> <p>Если у УЗ пользователя есть группа "Вендор API" И в запросе не была указана МО (параметр Lpu_id), то в параметре ответа "data" будут переданы записи по всему региону.</p> <p>Если у УЗ пользователя нет группы "Вендор API", то в ответе будут переданы записи только по МО пользователя.</p> <p>Бирки в ответе выводить по степени близости приема к TimeTable_beg.</p> <p>Возвращаемые параметры:</p> <ul style="list-style-type: none"> – Lpu_id (N, O) – идентификатор МО; – Org_Name (T, O) – Наименование МО; – TimeTables(S, O) – объект; – Person_id (N, O) – идентификатор пациента; – Person_SurName (T, O) – фамилия пациента; – Person_FirName (T, O) – имя пациента; – Person_SecName (T, O) – отчество пациента; – Person_Phone (T, O) – номер телефона пациента; – TimeTableSource (T, O) – тип бирки:Graf или MedService; – TimeTable_id (N, O) – идентификатор бирки; – TimeTableType_id (N, O) – тип бирки; – TimeTable_factTime (DT, H) – фактические дата и время приема для обслуженных бирок (если у бирки есть Фактическое время приема TimeTableGraf_factTime/TimeTableMedService_factTime, то пациент приходил); – TimeTable_begTime (DT, O) – дата и время начала приёма; – MedPersonal_id (N) – идентификатор сотрудника; – MedPersonal_SurName (T) – фамилия сотрудника; – MedPersonal_FirName (T) – имя сотрудника; – MedPersonal_SecName (T) – отчество сотрудника; – MedPersonal_FIO (T) – Ф. И. О. сотрудника в нормализованном виде. – MedService_id (N, H) – идентификатор службы (выводить только
--	---

	<p>если тип бирки MedService);</p> <ul style="list-style-type: none"> – MedService_Name (Т, Н) – наименование службы (выводить только если тип бирки MedService); – MedStaffFact_id (N, Н) – идентификатор врача (выводить только если бирка поликлиническая Graf); – Post_id (N, Н) – идентификатор должности врача (выводить только если бирка поликлиническая Graf); – Post_name (Т, Н) – наименование должности врача (выводить только если бирка поликлиническая Graf); – LpuSectionProfile_id (N, О) – идентификатор профиля отделения; – Usluga_Code (Т, Н) – код услуги; – Usluga_Name (Т, Н) – наименование услуги.
Возможные ошибки	<ul style="list-style-type: none"> – код "1", прикладной код "6", текст ошибки: "Неверное значение TimeTableSource. Должно быть Graf или MedService"; – код "2", прикладной код "6", текст ошибки: "Значение MedStaffFact_id или Post_id не найдено в системе"

5.5.2 Получение списка предстоящих приемов пациента в МО GET api/TimeTableListbyPatient

Метод предназначен для получения списка прошедших и предстоящих визитов пациента в МО за заданный временной промежуток. Поиск визитов производится в рамках МО вызова метода.

Таблица 10 – Параметры метода **GET api/TimeTableListbyPatient**

Входящие параметры	<ul style="list-style-type: none"> – Person_id (N,O) – идентификатор пациента; – TimeTable_beg (DT,H) – дата и время начала диапазона поиска (если не указано, то считать текущие время и дату за дату начала диапазона); – TimeTable_end (DT,H) – дата и время окончания диапазона; – Lpu_id (N,H) – идентификатор медицинской организации (доступно только для вендора)
Ответ	<p>Правила:</p> <ul style="list-style-type: none"> – значение параметра TimeTable_beg не может быть ранее текущих даты и времени. Иначе метод вернёт ошибку; – если параметр TimeTable_beg не задан, то устанавливается равным текущим дате и времени;

	<ul style="list-style-type: none"> – значение параметра TimeTable_end не может быть меньше параметра TimeTable_beg. Иначе метод вернёт ошибку; – если идентификатор человека не найден в Системе, то метод вернёт ошибку. <p>Возвращаемые параметры:</p> <ul style="list-style-type: none"> – Person_id (N, O) – идентификатор пациента; – Person_SurName (T, O) – фамилия пациента; – Person_FirName (T, O) – имя пациента; – Person_SecName (T, O) – отчество пациента; – Person_Phone (T, O) – номер телефона пациента; – TimeTable (S, O) – список ячеек расписания, на которые записан человек: <ul style="list-style-type: none"> – Lpu_id (N, O) – идентификатор медицинской организации (доступно только для вендора); – TimeTableSource (T, O) – метка источника данных; – TimeTable_id (N, O) – идентификатор ячейки расписания; – TimeTable_factTime (DT, H) – фактические дата и время приема для обслуженных ячеек расписания (если у ячейки есть фактическое время приема TimeTableGraf_factTime/TimeTableMedService_factTime, то пациент приходил); – TimeTable_begTime (T, O) – дата и время начала приёма; – MedService_id (N, H) – идентификатор службы; – MedService_Name (T, H) – наименование службы; – MedStaffFact_id (N, H) – идентификатор места работы врача; – Post_id (N, O) – идентификатор должности врача; – Post_name (T, O) – наименование должности врача; – LpuSectionProfile_id (N, O) – идентификатор профиля отделения; – Usluga_Code (T, H) – код услуги; – Usluga_Name (T, H) – наименование услуги.
--	--

5.5.3 Изменение статуса записи на прием PUT api/TimeTableGraf/TimeTableGrafStatus

Таблица 11 – Параметры метода PUT api/TimeTableGraf/TimeTableGrafStatus

Входящие параметры	<ul style="list-style-type: none"> – Person_id (N, O), TimeTableGraf_id (N, O) – составной идентификатор записи на прием. – EvnStatus_id (N,O) – идентификатор статуса направления – TimeTableGraf_id (N, O) – ссылка на бирку поликлиники
Ответ	<p>Успешный ответ – пустой ответ с кодом ошибки "0"</p> <p>Если EvnStatus_id меняется на 12 или 13, то в таблице dbo.TimeTableGraf значения полей RecClass_id, Person_id, EvnDirection_id меняется на NULL.</p>

6 Эксплуатация сервиса

Система предназначена для функционирования 24 часа в сутки 7 дней в неделю. Обеспечивается возможность взаимодействия с пользователями в круглосуточном режиме без перерывов, в том числе при доступе пользователей из других по отношению к серверной части временных зон.

Для программного обеспечения Системы определены следующие режимы функционирования:

- штатный режим (режим, обеспечивающий выполнение функций Системы);
- предаварийный режим (режим, предшествующий переходу в аварийный режим);
- аварийный режим (характеризуется отказом одного или нескольких компонентов программного и/или аппаратного обеспечения. В данном режиме функционируют ресурсы, которые в штатном режиме находятся в режиме горячего резерва)
- сервисный режим (режим для проведения реконфигурирования, обновления и профилактического обслуживания).

Информационный обмен со стороны Системы построен через:

- интеграционную шину Системы с соблюдением правил информационной безопасности;
- Сервисы интеграции.

Подробное описание приведено в документе "Регламент эксплуатации".

7 Аварийные ситуации

7.1 Описание аварийных ситуаций

Надежность Системы обеспечивается при следующих аварийных ситуациях:

- отказ Системы;
- сбой Системы.

Отказом Системы следует считать событие, состоящее в утрате работоспособности Системы и приводящее к невыполнению или неправильному выполнению контрольных примеров или задач функциональных модулей.

Сбоем Системы следует считать событие, состоящее во временной утрате работоспособности Системы и характеризующееся возникновением ошибки при выполнении контрольных примеров или задач функциональных модулей.

В Системе предусмотрено автоматическое восстановление обрабатываемой информации в следующих аварийных ситуациях:

- программный сбой при операциях записи–чтения;
- разрыв связи с клиентской программой (терминальным устройством) в ходе редактирования/обновления информации.

В Системе предусмотрена возможность ручного восстановления обрабатываемой информации из резервной копии в следующих аварийных ситуациях:

- физический выход из строя дисковых накопителей;
- ошибочные действия обслуживающего персонала.

В Системе предусмотрено автоматическое восстановление работоспособности серверной части Системы в следующих ситуациях:

- штатное и аварийное отключение электропитания серверной части;
- штатная перезагрузка Системы и загрузка после отключения;
- программный сбой общесистемного программного обеспечения, приведший к перезагрузке Системы.

В Системе предусмотрено полуавтоматическое восстановление работоспособности серверной части Системы в следующих аварийных ситуациях:

- физический выход из строя любого аппаратного компонента, кроме дисковых накопителей – после замены компонента и восстановления конфигурации общесистемного программного обеспечения;

- аварийная перезагрузка системы, приведшая к нефатальному нарушению целостности файловой системы – после восстановления файловой системы.

Для восстановления Системы после отказа или сбоя, необходимо сначала устранить причину отказа/сбоя (заменить неисправное оборудование, устранить системные ошибки и др.), а затем предпринять следующие действия:

- установить операционную систему, а затем – соответствующий пакет обновления; проверить правильность работы домена.
- установить СУБД, а затем – соответствующий пакет обновления.
- восстановить базу данных из резервной копии; перезагрузить сервер после восстановления базы данных.
- проверить доступность Системы; чтобы убедиться в правильности работы, запустите сценарий проверки основных функций.
- активировать возможность работы пользователей в штатном режиме.

В случае отказа или сбоя Системы, связанного с неисправностью оборудования, работы проводит Администратор Заказчика.

В случае отказа или сбоя Системы, связанного с системной ошибкой, работы проводит Администратор Исполнителя.

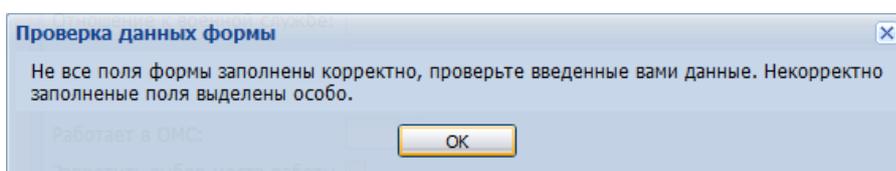
7.2 Действия в случае несоблюдения условий выполнения технологического процесса

При работе с Системой пользователю могут отображаться сообщения нескольких типов:

- сообщение об успешном завершении действия;
- сообщение об ошибке;
- предупреждение;
- сообщение о неисправности системы.

Сообщение об успешном завершении действия содержит краткое резюме операции. Для закрытия сообщения нажмите кнопку "ОК".

Сообщение об ошибке отображается в случае, когда дальнейшее выполнение действия в Системе невозможно. Как правило, в таком сообщении содержится краткое описание причины возникновения ошибки. Для закрытия сообщения об ошибке нажмите кнопку "ОК".



Предупреждение отображается в том случае, если действия, совершенные оператором, могут повлечь за собой какие-либо особенности в выполнении операции, но не приведут к ошибке. Например, если оператор укажет у сотрудника ставку менее 0,1, то отобразится сообщение, что такая ставка не будет учитываться при выгрузке. Для того чтобы продолжить выполнение действия, нажмите кнопку "Да"/"Продолжить". Для того чтобы прекратить действие, нажмите кнопку "Нет"/"Отмена".

В случае возникновения ошибки о неисправности системы, пользователю системы следует обратиться к администратору системы.

Администратор системы для решения проблем обращается к эксплуатационной документации, настоящему руководству, онлайн справочной системе.

В случае невозможности разрешения ситуации следует обратиться в техническую поддержку.