

Тестовое задание для кандидатов на позицию системного аналитика в MedTech

Требования:

1. Бизнес-требования:

- Обеспечение удобного процесса записи для пациентов.
- Повышение заполняемости расписания врачей.
- Минимизация пропущенных приёмов через систему напоминаний.
- Интеграция с внешними календарями для удобства врачей.
- Гибкость управления записями (изменение, отмена, уведомления).

2. Функциональные требования:

- Возможность записи, изменения и отмены приёма.
- Отправка уведомлений пациентам и врачам.
- Отображение доступных слотов в режиме реального времени.
- Синхронизация с календарями врачей.
- Защита персональных данных пациентов.

3. Нефункциональные требования:

- Высокая надёжность и отказоустойчивость системы.
- Интуитивно понятный пользовательский интерфейс.
- Быстрая обработка запросов.
- Масштабируемость системы для работы с большим количеством пользователей.
- Соответствие требованиям законодательства о защите персональных данных.

Заинтересованные стороны

Врачи.

Администраторы клиники

Руководство клиники

Пациенты

Администраторы системы

Интеграция модуля управления записями пациентов с другими системами

Интеграция с CRM

1. Создание записи о приеме в CRM

POST /crm/appointments

Host: api.crm-system.com

Authorization: Bearer YOUR_ACCESS_TOKEN

Content-Type: application/json

```
{  
  "patient_id": "12345",  
  "doctor_id": "67890",  
  "date": "2025-04-10",  
  "time": "14:00",  
  "status": "confirmed"  
}
```

2. Обновление записи

PUT /crm/appointments/12345

Host: api.crm-system.com

Authorization: Bearer YOUR_ACCESS_TOKEN

Content-Type: application/json

```
{  
  "date": "2025-04-11",  
  "time": "15:00",  
  "status": "rescheduled"  
}
```

3. Удаление записи

DELETE /crm/appointments/12345

Host: api.crm-system.com

Authorization: Bearer YOUR_ACCESS_TOKEN

Интеграция с Google Calendar врача

Создание события в Google Calendar

Ссылка на документацию

<https://developers.google.com/workspace/calendar/api/v3/reference/events?hl=ru#resource>

Параметры:

- **summary:** *"Прием у врача"* – Краткое название события, которое будет отображаться в календаре.
- **location:** *"Клиника №1, ул. Ленина, 10"* – Адрес или место встречи.
- **description:** *"Запись пациента Иванов И.И. к доктору Петрову П.П."*
- **start:** Определяет начало события.
- **dateTime:** *"2025-04-10T14:00:00"* – Дата и время начала в формате ISO 8601 (YYYY-MM-DDTHH:MM:SS).
- **timeZone:** *"Europe/Moscow"* – Часовой пояс, чтобы время отображалось правильно.
- **end:** Определяет окончание события.
- **dateTime:** *"2025-04-10T14:30:00"* – Дата и время окончания.
- **timeZone:** *"Europe/Moscow"* – Часовой пояс.
- **attendees:** Список email-адресов участников, которым отправится приглашение:
 - **{"email": "doctor@example.com"}** – Email врача.
 - **{"email": "patient@example.com"}** – Email пациента.
- **reminders:** Настройки уведомлений перед событием.
- **useDefault:** **false** – Отключает стандартные напоминания Google Calendar.
- **overrides** – Указывает собственные напоминания:
 - **{"method": "email", "minutes": 1440}** – Отправить email за 1440 минут (24 часа).
 - **{"method": "popup", "minutes": 60}** – Показывать всплывающее уведомление за 60 минут.

POST <https://www.googleapis.com/calendar/v3/calendars/primary/events>

Authorization: Bearer YOUR_ACCESS_TOKEN

Content-Type: application/json

```
{
  "summary": "Прием у врача",
  "location": "Клиника №1, ул. Ленина, 10",
  "description": "Запись пациента Иванов И.И. к доктору Петрову П.П.",
  "start": {
    "dateTime": "2025-04-10T14:00:00",
    "timeZone": "Moscow"
  },
  "end": {
    "dateTime": "2025-04-10T14:30:00",
    "timeZone": "Europe/Moscow"
  },
  "attendees": [
    {"email": "doctor@example.com"},
    {"email": "patient@example.com"}
  ],
  "reminders": {
    "useDefault": false,
    "overrides": [
      {"method": "email", "minutes": 1440},
      {"method": "popup", "minutes": 60}
    ]
  }
}
```

При тестировании модуля управления записями пациентов важно учитывать его функциональность, безопасность, производительность и удобство использования. Подход к тестированию может включать следующие методы:

1. Функциональное тестирование

Проверка соответствия требованиям:

- Создание, чтение, обновление и удаление (CRUD) записей пациентов.
- Валидация данных (например, корректность форматов дат, номеров телефонов, электронной почты).
- Проверка интеграции с другими модулями (например, системой назначений или лабораторных исследований).

2. Тестирование производительности

Нагрузочное тестирование (например, как система справляется с массовым внесением и запросом данных).

1. Юзабилити-тестирование

Проверка удобства интерфейса для пациентов

5. Интеграционное тестирование

Проверка работы API

Диаграмма последовательности вместо BPMN

