Техническое задания

DocFlow

1. Общая информация о проекте

Название проекта: Docflow

Рынок: Россия

Сфера: Документооборот в медицинской сфере

Цель: Автоматизация процесса подписания и обмена документами между медицинскими учреждениями и пациентами, используя ЕСИА (госуслуги) и альтернативные методы аутентификации.

2. Описание MVP

Основные функции, реализуемые в MVP:

Регистрация пациентов через два варианта:

Через портал Госуслуг (ЕСИА).

Через номер телефона (без использования ЕСИА).

Подписание документов пациентом через Госуслуги.

Отправка уведомлений о новых документах через СМС.

Личный кабинет пациента с доступом ко всем отправленным и подписанным документам.

Личный кабинет клиники с возможностью управления документами и отправки их пациентам на подписание.

3. Функциональные требования

3.1. Регистрация пациентов

Через ЕСИА (госуслуги):

Пользователь переходит на страницу регистрации.

Выбор варианта "Войти через Госуслуги".

Перенаправление на страницу авторизации ЕСИА.

Аутентификация через ЕСИА.

Получение данных пользователя (ФИО, номер телефона).

Создание личного кабинета на основе данных, полученных через ЕСИА.

Возможность сразу подписывать документы, отправленные клиниками.

Через номер телефона:

Пользователь переходит на страницу регистрации.

Ввод номера телефона.

Получение СМС с кодом подтверждения.

Подтверждение номера телефона.

Ввод данных: ФИО, другие необходимые данные.

Создание личного кабинета.

Возможность просматривать документы, но без возможности подписания через систему (подпись возможна только физически в клинике).

3.2. Авторизация

Возможность авторизоваться в системе через:

Госуслуги (ЕСИА).

Номер телефона (OTP код).

3.3. Подписание документов

Через Госуслуги (ЕСИА):

Клиника отправляет запрос на подписание документа пациенту.

Пациент получает СМС с информацией о новом документе (название клиники и ссылка на документ). Пример: “Здравствуйте! [Название клиники] отправила вам запрос на подписание документа. Ссылка: [Ссылка]”

При переходе по ссылке, если пациент авторизован через ЕСИА, ему предоставляется возможность подписать документ с использованием данных Госуслуг.

При успешной подписи документ передаётся обратно в систему клиники как подписанный.

Без использования Госуслуг:

Если пациент зарегистрирован по номеру телефона без ЕСИА, ему будет предложено заполнить поля в документе (личные данные такие как:ФИО, ИНН, паспортные данные и тд.)

Сотрудник клиники сможет скачать документ с предварительно заполненными данными для дальнейшей распечатки.

3.4. СМС-уведомления

Пациенту:

При отправке документа клиникой, пациенту отправляется СМС с информацией о документе и ссылкой для его просмотра/подписания.

3.5. Личный кабинет пациента

Просмотр всех документов, которые были отправлены клиниками.

Документы разбиваются по клиникам.

При выборе конкретной клиники отображаются все связанные с ней документы.

Возможность просмотреть подписанные и отправленные документы.

3.6. Личный кабинет клиники

Возможность создания нового документа.

Отправка документов пациентам для подписания с указанием ФИО и номера телефона.

Просмотр всех отправленных документов и их статусов (подписан/ожидает подписи).

Просмотр подписанных документов.

4. Требования к интерфейсу

4.1. Интерфейс пациента

Главная страница:

Выбор способа авторизации: через Госуслуги или по номеру телефона.

Уведомления о новых документах.

Раздел "Мои документы":

Список клиник, от которых были отправлены документы.

Просмотр документов по выбранной клинике.

4.2. Интерфейс клиники

Главная страница:

Возможность создания нового документа.

Отправка документа на подписание пациенту.

Просмотр статусов всех документов.

5. Интеграции

5.1. Интеграция с ЕСИА (Госуслуги)

Осуществляется через официальный API для аутентификации пользователей и подписания документов.

5.2. СМС-шлюз

Необходим для отправки уведомлений пациентам о поступлении новых документов.

Требуется интеграция с провайдером СМС-услуг.

6. Безопасность

Защита данных пользователей (персональные данные) в соответствии с законодательством РФ, в том числе ФЗ-152 "О персональных данных".

Шифрование всех взаимодействий с ЕСИА.

Двухфакторная аутентификация при входе по номеру телефона.

Защита документов на всех этапах передачи и хранения.

7. Технические требования

7.1. База данных

Хранение данных пользователей и документов должно быть организовано с учётом безопасности и возможности масштабирования.

7.2. Архитектура

Микросервисная архитектура для разделения логики приложения (регистрация, авторизация, документы, уведомления).

7.3. Производительность

Система должна поддерживать возможность одновременной работы большого числа пользователей, особенно в моменты массовой отправки документов.

8. План разработки

8.1. Этапы разработки

Проектирование системы и базы данных.

Разработка интеграции с ЕСИА.

Реализация регистрации и авторизации пользователей.

Создание личных кабинетов (клиника и пациент).

Подписание документов.

СМС-уведомления.

Тестирование системы.

Подготовка к запуску.

8.2. Сроки

Предполагаемый срок выполнения MVP — 1-2 месяца.

9. Тестирование

Тестирование всех сценариев взаимодействия пользователей (пациенты и клиники).

Проверка корректности работы интеграций (ЕСИА, СМС).

Тестирование производительности системы при нагрузке.

10. Запуск и поддержка

После завершения разработки MVP необходимо организовать:

Пилотный запуск с ограниченным числом клиник.

Сбор обратной связи от пользователей.

Оперативное устранение ошибок.

Бэкенд архитектура

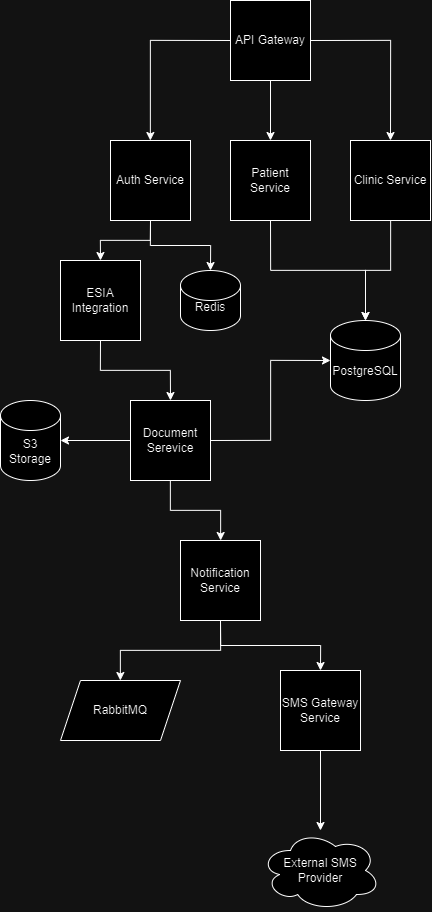
Каждый сервис будет отвечать за отдельную часть функционала, что облегчит разработку, поддержку и масштабирование системы.

**1. Общая архитектура**

Мы разобьём систему на несколько микросервисов, которые будут взаимодействовать через REST API и/или очереди сообщений (например, RabbitMQ). Также будет использована база данных, система авторизации и другие ключевые компоненты.

Основные микросервисы:

* **Auth Service (Сервис авторизации и аутентификации)**
* **Patient Service (Сервис управления пациентами)**
* **Clinic Service (Сервис управления клиниками)**
* **Document Service (Сервис управления документами)**
* **Notification Service (Сервис уведомлений)**
* **SMS Gateway Service (Сервис отправки СМС)**
* **ESIA Integration Service (Интеграция с Госуслугами)**
* **Logging & Monitoring Service (Мониторинг и логирование)**
* **API Gateway (шлюз для всех API-запросов)**

****

Изображение 1. Диаграма архитектуры проекта

**2. Детальная разбивка микросервисов**

**2.1. Auth Service (Сервис авторизации и аутентификации)**

Этот сервис отвечает за:

* Аутентификацию через ЕСИА (госуслуги).
* Аутентификацию по OTP (СМС-код для подтверждения номера телефона).
* Управление сессиями пользователей.
* Обработку JWT-токенов для аутентификации между микросервисами.

**Технологии:** OAuth 2.0, JWT, OpenID Connect, ЕСИА API.  
**Хранилище:** Redis для сессий пользователей и черного списка токенов.

**2.2. Patient Service (Сервис управления пациентами)**

Отвечает за:

* Регистрацию и хранение данных пациентов.
* Обновление профилей пациентов.
* Получение списка документов для пациентов.
* Хранение метаданных о пациентах (ФИО, номер телефона, клиники и прочее).

**Технологии:** REST API, gRPC.  
**Хранилище:** PostgreSQL или MySQL для хранения данных пациентов.

**2.3. Clinic Service (Сервис управления клиниками)**

Этот сервис отвечает за:

* Управление данными о клиниках (название, контактные данные, зарегистрированные врачи).
* Сопоставление пациентов с клиниками.
* Управление документами, отправленными от клиники к пациенту.

**Технологии:** REST API, gRPC.  
**Хранилище:** PostgreSQL или MySQL.

**2.4. Document Service (Сервис управления документами)**

Ключевой сервис, отвечающий за:

* Генерацию и хранение документов.
* Интеграцию с механизмами подписания документов через ЕСИА (госуслуги).
* Хранение метаданных о подписанных и неподписанных документах.
* Создание и отправку документов пациентам для подписания.
* Подготовка документов для печати, если пациент не может подписать онлайн.

**Технологии:** REST API, JSON/XML для хранения документов, интеграция с Госуслугами.  
**Хранилище:** PostgreSQL для метаданных и S3 (или аналог) для хранения документов.

**2.5. Notification Service (Сервис уведомлений)**

Отвечает за:

* Управление уведомлениями для пользователей (пациентов и клиник).
* Отправку уведомлений через веб (интерфейс) или через email.
* Интеграцию с СМС-шлюзом для отправки СМС-уведомлений пациентам о новых документах.

**Технологии:** RabbitMQ для асинхронной отправки уведомлений, REST API для запросов от других сервисов.  
**Хранилище:** Redis для очередей уведомлений.

**2.6. SMS Gateway Service (Сервис отправки СМС)**

Этот сервис обрабатывает отправку всех СМС-сообщений через интеграцию с провайдером СМС-услуг:

* Отправка OTP-кодов для авторизации через телефон.
* Отправка уведомлений о новых документах пациентам.
* Логирование отправленных СМС и обработка ошибок (например, в случае сбоя).

**Технологии:** REST API для интеграции с провайдерами (например, Twilio, SMS.ru), RabbitMQ для очередей сообщений.  
**Хранилище:** MongoDB для хранения логов и статусов отправленных СМС.

**2.7. ESIA Integration Service (Интеграция с Госуслугами)**

Интеграция с ЕСИА через официальный API:

* Аутентификация пациентов через ЕСИА.
* Подпись документов через Госуслуги.
* Получение данных о пользователе из ЕСИА (ФИО, телефон).

**Технологии:** OAuth 2.0 для аутентификации, SOAP/REST API для интеграции с ЕСИА.

**2.8. Logging & Monitoring Service (Мониторинг и логирование)**

Отвечает за:

* Сбор логов со всех микросервисов.
* Мониторинг производительности системы (метрики, ошибки, задержки).
* Алерты и уведомления при сбоях или отказах компонентов.

**Технологии:** ELK Stack (Elasticsearch, Logstash, Kibana) для логов, Prometheus и Grafana для мониторинга.  
**Хранилище:** Elasticsearch для хранения логов и метрик.

**2.9. API Gateway (шлюз для всех API-запросов)**

Единая точка входа для всех клиентских запросов (и от пациентов, и от клиник):

* Маршрутизация запросов к нужным микросервисам.
* Авторизация и аутентификация запросов (через JWT).
* Балансировка нагрузки и ограничение скорости запросов.
* Безопасность: защита от атак (например, DDoS) и контроль доступа.

**Технологии:** NGINX или Kong для реализации API-шлюза, OAuth 2.0 и JWT для безопасности.

**3. Хранилища данных**

1. **PostgreSQL/MySQL (Реляционные БД)**: Для хранения данных о пациентах, клиниках и метаданных о документах.
2. **Redis**: Для хранения сессий пользователей, кеширования и работы с очередями задач.
3. **MongoDB (или другое NoSQL-хранилище)**: Для логов о СМС и статусах отправки уведомлений.
4. **S3 (или аналоги)**: Для хранения подписанных и неподписанных документов в формате PDF/JSON/XML.
5. **Elasticsearch**: Для хранения логов, метрик и поиска по ним.

**4. Взаимодействие микросервисов**

**Пример сценария: Отправка документа на подпись**

1. **Клиника** через веб-интерфейс отправляет запрос на создание и отправку документа пациенту (через **API Gateway**).
2. **Clinic Service** обрабатывает запрос, проверяет данные пациента, клиники, и отправляет задачу на создание документа в **Document Service**.
3. **Document Service** создаёт документ, сохраняет его в хранилище (S3), и отправляет задачу в **Notification Service** для уведомления пациента.
4. **Notification Service** создаёт запись о задаче уведомления и передаёт запрос в **SMS Gateway Service** для отправки СМС пациенту.
5. **SMS Gateway Service** отправляет СМС с ссылкой на документ, используя провайдера СМС.
6. Пациент, получив СМС, переходит по ссылке на документ. Запрос направляется в **Document Service**, который проверяет статус документа и доступ.
7. Если пациент авторизован через ЕСИА, он может подписать документ. Для этого **Document Service** инициирует запрос в **ESIA Integration Service**.
8. **ESIA Integration Service** выполняет аутентификацию и подписывает документ через Госуслуги, после чего обновляет статус документа в **Document Service**.

**5. Безопасность**

* **Авторизация**: Использование OAuth 2.0 и JWT для авторизации всех запросов.
* **Шифрование данных**: TLS/SSL для передачи данных между клиентом и сервером.
* **Хранение данных**: Все персональные данные пользователей шифруются как в процессе передачи, так и при хранении, в соответствии с требованиями ФЗ-152 "О персональных данных".
* **Логирование и мониторинг**: Полное логирование всех действий пользователей и событий в системе для анализа и безопасности.

**6. Инструменты для DevOps и CI/CD**

1. **Docker**: Для контейнеризации микросервисов.
2. **Kubernetes**: Для оркестрации контейнеров и управления масштабированием.
3. **Jenkins/GitLab CI**: Для автоматизации CI/CD процессов.
4. **Helm**: Для управления конфигурациями Kubernetes.
5. **Prometheus и Grafana**: Для мониторинга и алертинга.

Эта архитектура обеспечивает гибкость, безопасность и масштабируемость, что особенно важно для системы документооборота в медицинской сфере, где необходимо учитывать строгие требования к защите данных и высокой доступности.