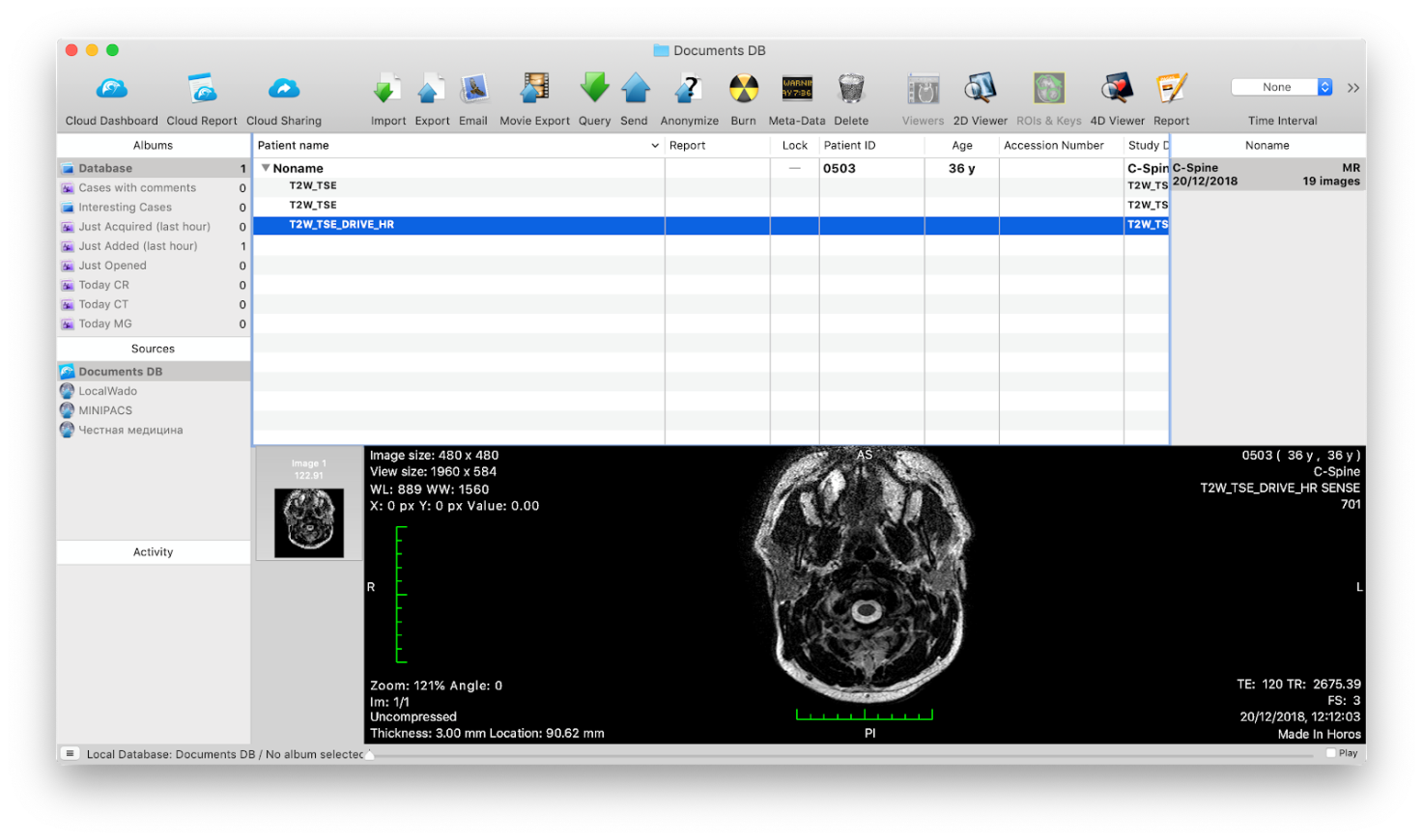
**Техническое задание на разработку модуля загрузки рентген-файлов с предварительной анонимизацией на стороне клиента.**

**Общая задача**: Необходимо разработать JS-модуль (с использованием чистого JS, либо JQuery) для реализации функционала массовой загрузки рентген-снимков с предварительной анонимизацией (стиранием данных из файла) на стороне браузера (это требование является самым главным, и продиктовано требованиям закона о персональных данных в медицине).

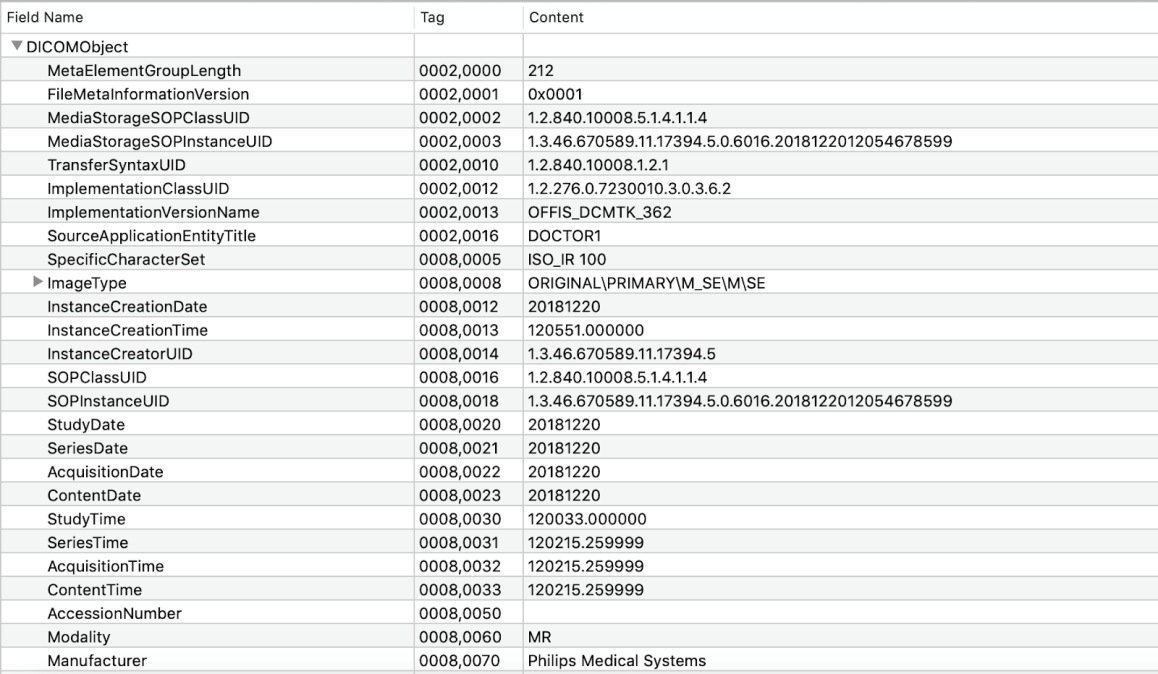
**Что такое DICOM-файлы**

DICOM (*Digital Imaging and Communications in Medicine*) – это международный стандарт для обмена медицинскими данными.

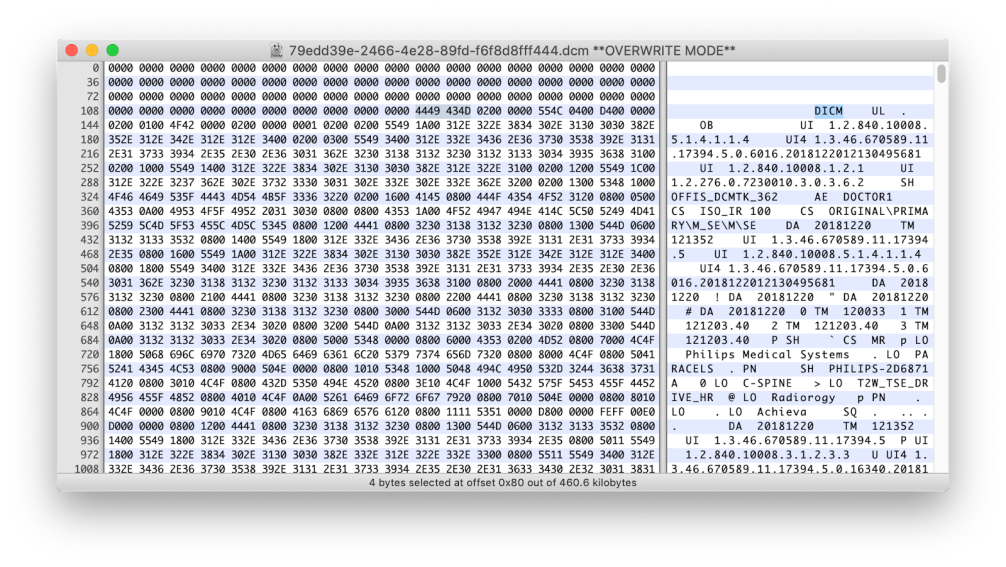
*DICOM-файл, открытый в программе Horos*

**Важные детали, касательно DICOM файлов, которые нужно знать исполнителю:**

* DICOM-файлы генерируются аппаратами МРТ, КТ, ПЭТ, рентгенами, маммографами, УЗИ, ЭКГ и некоторыми другими типами медицинских приборов.
* DICOM-файл – это бинарный файл, содержащий в себе структурированную текстовую информацию об исследовании и JPEG изображение (обычно это рентгеновский снимок)
* Информация об исследовании передается в виде таблицы "Ключ-Значение", где в качестве ключей выступают теги. Тег - это 4 байта, Например «2001,105f ⇢ Philips Imaging DD 001». Каждый тег - стандартизирован протоколом. Полное описание всех тегов можно посмотреть например тут: <https://dicom.innolitics.com/ciods>
* На удивление, разные аппараты пишут теги в разных регистрах. Может быть «**2001,105f**» а может быть «**2001,105F**»
* Открыть DICOM-файл можно в программах Horos, Osirix, E-Film, Radiant и еще десятке других.
* DICOM-файлы обычно имеют расширение .dcm, .dicom либо передаются БЕЗ расширения вовсе.
* Если открыть DICOM-файл в HEX-редакторе, то **128-131 байты** в файле - **всегда** «**4449 434D**» – что читается как «DICM». По этому признаку - можно определить является ли файл DICOM или нет.
* Наиболее популярная OpenSource библиотека для работы с DICOM-файлами - DCMTK (<https://dicom.offis.de/dcmtk.php.en>). Она разработана для UNIX- и Windows- подобных систем, однако есть порты например на JavaScript (NodeJS). Помимо основных задач по работе с тегами, извлечением и записью изображения в файл, DCMTK выполняет рад сервисных задач (реализует сервер, поиск по файлам, получение сервисных данных и многое другое). Но есть и другие OpenSource проекты для работы с DICOM.
* 1 исследование может быть представлено в виде одного DICOM-файла, а может в виде нескольких тысяч DICOM-файлов. Например, рентген-исследование легких - это 1 файл, а ПЭТ/КТ (позитронно-эмиссионная томография) может представлять из себя несколько тысяч DICOM файлов.
* Обычно DICOM-файлы записывают в клиниках на компакт-диск и заодно добавляют туда программу-просмотрщик и сервисные файлы.
* DICOM файлы могут быть объединены в в исследование согласно тегу StudyInstanceUID (0020,000d), который по требованию протокола должен быть единый у всех файлов одного исследования (например, «0020,000d ⇢ 1.3.46.670589.11.17394.5.0.16340.2018122011550039144»). Так же есть тег SeriesInstanceUID (0020,000e) который определяет серию снимков в пределах одного исследования и SeriesNumber, который определяет номер снимка внутри серии.
* DICOM файл не обязательно содержит все доступные теги. Может быть всего несколько обязательных. Технически может быть что часть обязательных тегов (например StudyInstanceUID может оказаться пустым). Очень редко, но бывает.



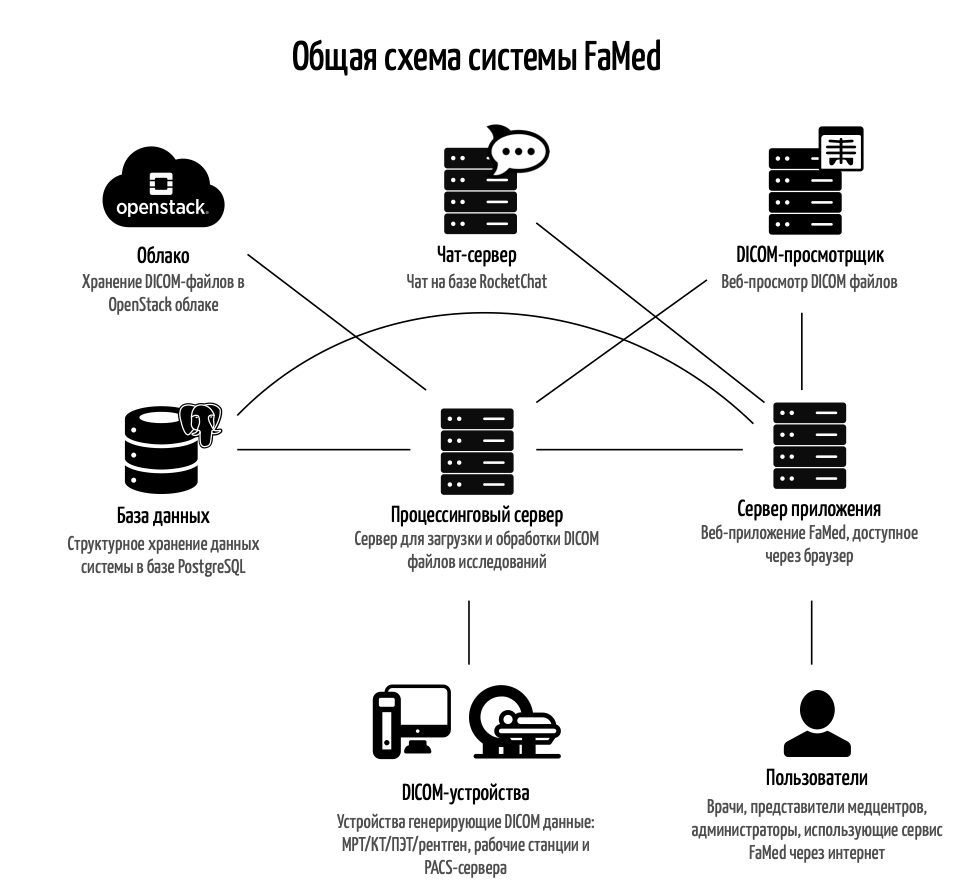
*Таблица тегов DICOM-файла, открытая в Horos (не полная, на экран влезла 1/10 часть)*



DICOM-файл, открытый в HEX-редакторе iHEX

**Что такое платформа FaMed**

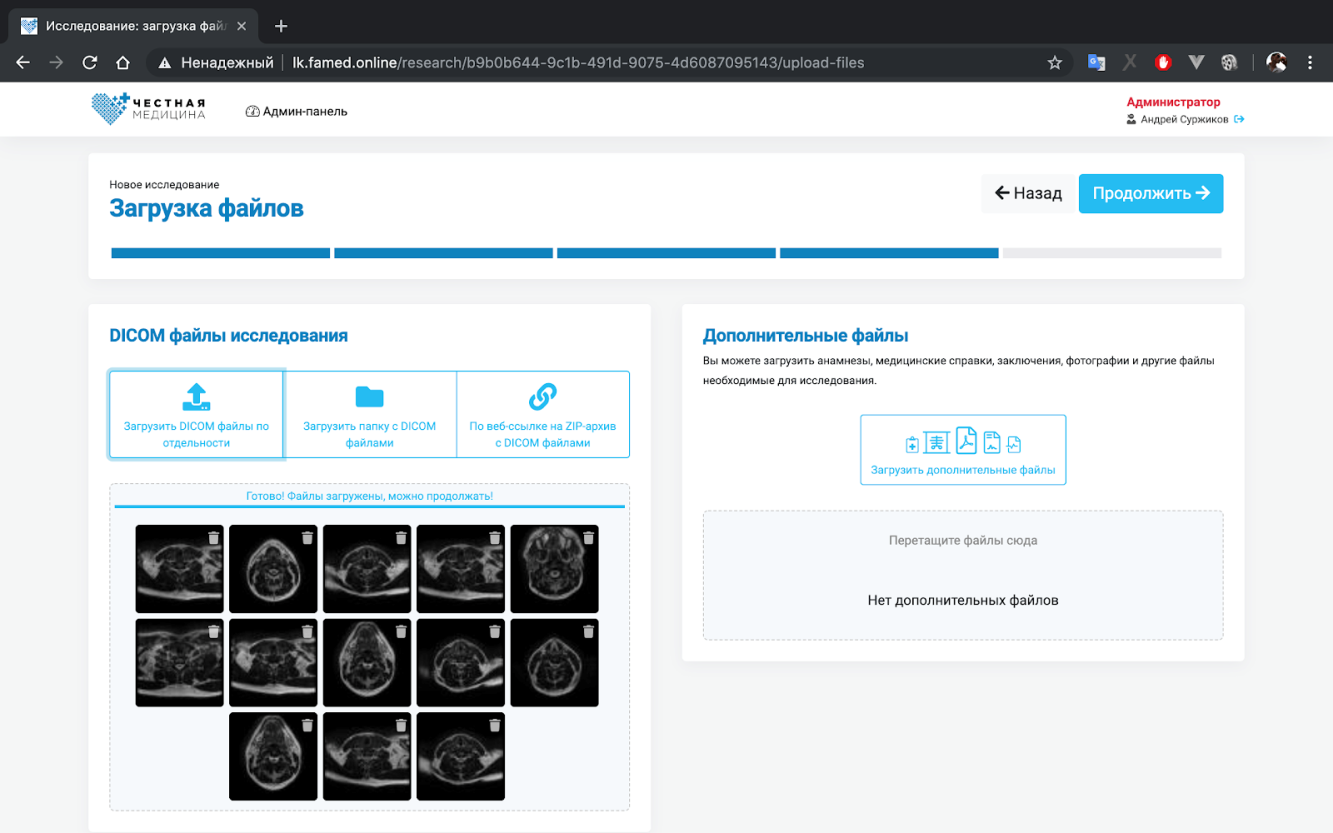
Платформа FaMed – это веб-сайт, реализующий функционал для описания рентген-снимков дистанционными врачами-рентгенологами.



*Общая схема работы*

**Принцип работы:**

1. Клиника (клиент) загружает DICOM-файлы исследования в платформу по одному из каналов передачи данных (по API, через протокол DICOM-push, через наше приложение для Windows, либо через браузер, на странице загрузки в платформе)
2. Создается исследование, к нему привязываются все эти DICOM файлы
3. Врач видит новое исследование и берет его в работу
4. Врач скачивается DICOM файлы к себе (сервер группирует их в zip архив), открывает в программе для просмотра и проводит исследование и изучение снимков.
5. Врач описывает исследование на странице создания протокола-отчета и завершает исследование
6. Клиника скачивает протокол-отчет из платформу и передает пациенту.



*Страница загрузки DICOM-файлов в платформу FaMed*

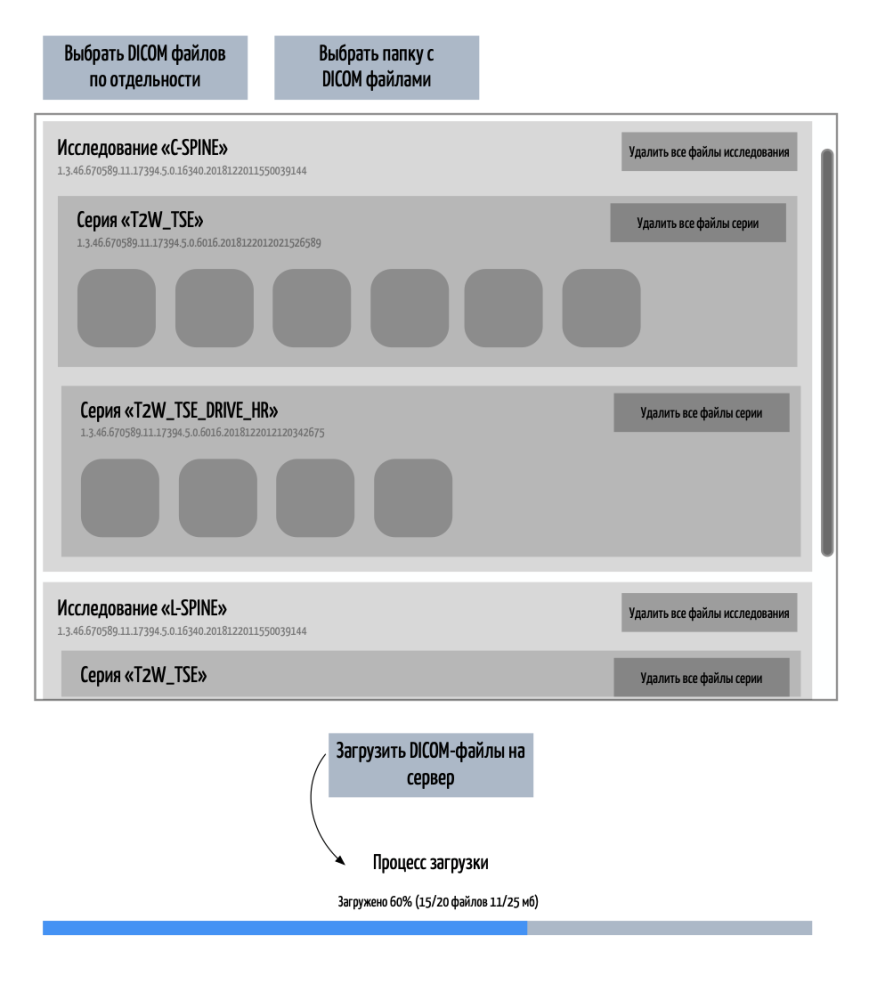
**Суть проблемы:**

В данный момент, при загрузке через браузер – мы получаем DICOM-файлы с данными пациентов, эти файлы загружаются на наш сервер, далее мы затираем необходимые поля с персональными данными (например в поле 0010,0010 – PatientsName мы записываем «NoName»). Согласно закону о персональных данных (152-ФЗ) мы не можем принимать и обрабатывать DICOM-файлы, содержащие в себе персональные данные (ФИО и др) пациента, поэтому персональные данные нужно удалить ДО попадания файлов на сервер.

**Подробная задача:**

Необходимо разработать JS-модуль для массового загрузчика DICOM-файлов на файл-сервер платформы FaMed. Сложность модуля заключается в необходимости работать с данными файла непосредственно в браузере (придется работать с функциями FileReader-а: readAsBinaryString или readAsArrayBuffer).

1. Финальная production-версия модуля должна подключаться в виде одного JS файла и одного CSS файл со стилями.
2. **Модуль должен принимать файлы от "<input type=file"** элемента HTML. Как по одному, так и группой и в виде webkitdirectory (GoogleChrome позволяет загружать "все файлы из папки", что для наших целей очень удобно).
3. **Группировка файлов по исследованиям** (по StudyInstanceUID) и по сериям внутри исследований (по SeriesInstanceUID). Сортировка внутри серии по SeriesNumber.
4. **Возможность до начала загрузки удалить отдельный файл**, всю серию, или все исследование целиком. При наведении на файл - показать иконку урны, и при клике - спросить "точно ли удалить файл?".
5. **Генерирование превьюшек** (миниатюрное изображение DICOM файла, ) первых N-файлов. Превьюшки остальных файлов генерировать и показывать при клике на изображение. По-умолчанию использовать изображение "Нажмите для показа миниатюры".
6. **Анонимизация указанных тегов в каждом DICOM файле**. В параметры модуля необходимо передавать объект вида TAG=>NEW\_VALUE (например {"0010,0010":"NoName", "2001,105f":"Hidden"}). Данные теги каждого дайком файла перед отправкой на сервер должны замениться на
7. **Модуль должен принимать URL для загрузки файлов**. Загружаем в виде formData POST запросом, сразу по несколько файлов (количество – вынести в параметр модуля) для уменьшения количества запросов на сервер (в час пик, десятки клиентов могу грузить по несколько тысяч файлов одновременно).
8. **Верстка – Bootstrap 4**.
9. **Прогресс-бар загрузки** файлов (10% (15/150 файлов, 15мб/150мб) на сервер.
10. **Модуль должен иметь возможность вывода уже загруженных ранее файлов**. Например, загрузили 100 файлов, закрыли вкладу, а через час решили загрузить еще 50 файлов. Модуль должен принять в качестве параметра массив объектов, каждый из которых репрезентуют DICOM-файл (сами DICOM-файл передавать с сервера в браузер нет смысла, будут переданы только UUID файла, URL превьюшки, UID исследования, UID серии и номер в серии). Также необходимо поддержать удаление уже загруженных ранее файлов или исследований/серий целиком (передать DELETE запрос с UUID файлов для удаления)
11. **Использовать AXIOS для XHR**.
12. Во время процесса выбора и загрузки файлов должен быть активен **слушатель события ухода со страницы** и надо выдавать стандартное предупреждение, что мол вы можете потерять несохраненные данные, если уйдете с веб-страницы сейчас.



*Схематичный прототип*

**Варианты решения задачи**

Наиболее амбициозной задачей в данном модуле – является часть, касающаяся работы с бинарными DICOM файлом. Однако OpenSource сообществом написано много различных модулей, среди которых есть те, которые выполняют наши задачи, например:

* <https://github.com/cornerstonejs/dicomParser> (извлечение тегов)
* <https://github.com/ivmartel/dwv> (генерация превьюшек)
* <https://github.com/rii-mango/Daikon> (извлечение тегов)

В качестве базиса для мультизагрузки файлов можно использовать, например, <https://www.dropzonejs.com/>, а можно написать загрузчик с чистого листа.

Написание серверной части - от вас не требуется!