

Informatyka Medyczna

Projekt

Dokumentacja

Wiktor Pawłowski

1. Temat projektu

A3. Generacja raportów konsultowanych danych obrazowych

Celem projektu jest stworzenie aplikacji generującej raporty odnośnie danych obrazowych w formacie DICOM. Na potrzeby realizacji projektu można założyć, że pliki pojawiają się w dedykowanym katalogu, zadaniem aplikacji jest więc okresowe śledzenie jego zawartości i wydobycie *metadanych*, takich jak modalność (CT, MR, RTG, ...), identyfikator badania i serii, identyfikator pacjenta i lekarza, dawka promieniowania, natężenie pola magnetycznego (stosownie do modalności). Pozyskane dane powinny być umieszczane w repozytorium pozwalającym na łatwe przetwarzanie. Dodatkowo powinna być dostarczona aplikacja z interfejsem graficznym pozwalająca tworzyć raporty na podstawie zadanych kryteriów.

2. Ogólny opis stworzonego projektu

Została stworzona dwuczęściowa aplikacja w języku Python. Wykorzystywana baza danych to baza MySQL.

Pierwszą częścią jest „Repository checker”. Okresowo przegląda on katalog, w którym pojawiają się pliki DICOM. W momencie, gdy pojawi się nowy plik, próbuje on wydobyć zdefiniowane wcześniej dane z pliku a następnie, zapisuje informacje w bazie danych. Przeglądane pliki z katalogu są sortowane malejąco ze względu na datę ich modyfikacji, a dodatkowo, zapisywana jest data najnowszego wcześniej przejrzanego pliku.

Drugą częścią jest aplikacja GUI służąca do przeglądania przetworzonych plików, zarządzania nimi, tworzenia raportów, dokonywania zmian w bazie danych oraz dodawania komentarzy do poszczególnych plików.

W ramach projektu powstała także maszyna wirtualna w celu łatwiejszego rozwoju projektu.

Repozytorium z projektem znajduje się na stronie https://github.com/wiktor145/im_projekt.

3. Instrukcja uruchomienia

Aby uruchomić „repository checker”, należy uruchomić plik `repository_checker.py` z katalogu `repository_checker`.

Aby uruchomić aplikację GUI, należy uruchomić plik `dicom_repository.py` z katalogu `client_gui`.

4. Wymagania techniczne

Preferowany system Linux

Baza MySQL

Python 3.8

Pakiety Pythona:

- Babel

- mysql-connector-python

- protobuf

- pydicom

- pytz

- setuptools

- six

- tkalendar

5. Krótki opis implementacji

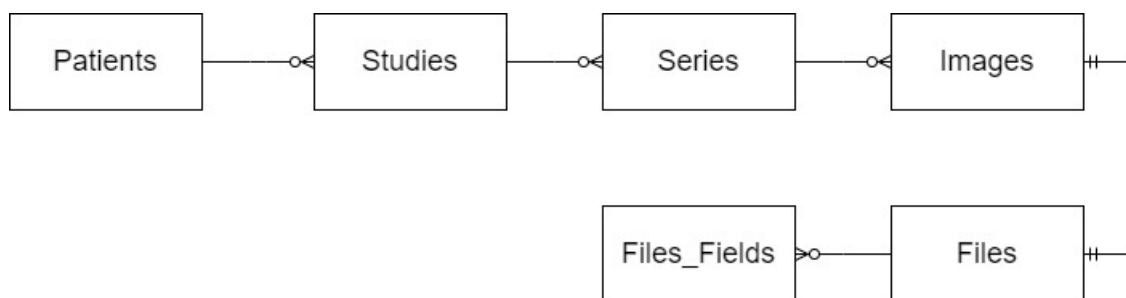
Wszystkie pliki dotyczące aplikacji GUI (jej działania i wyglądu) znajdują się w katalogu `client_gui`. GUI zostało wygenerowane z pomocą narzędzia Page.

Domyślnie, katalogiem z plikami DICOM jest katalog `data`, a wyjściem dla raportów katalog `reports`.

W katalogu `database` znajduje się plik z klasą służącą do komunikacji z bazą danych (wstawianie nowych danych wyciąganie aktualnych itp.), klasy pythonowe odpowiadające strukturze bazy danych oraz plik `db.sql` pozwalającym stworzyć potrzebną bazę danych od zera.

Wszystkie dane konfiguracyjne oraz używane stałe znajdują się w pliku `constants.py` w katalogu `other_classes`. Znajdują się tam również zmienne z zamiarami na bazę danych.

6. Schemat bazy danych



Patients – pacjenci

Studies – badania

Series – serie

Images – zdjęcia

Files – pliki

Files_fields – tagi DICOM plików

7. Opis aplikacji „repository checker” i jej możliwości

Jego działanie rozpoczyna się od połączenia z bazą danych oraz pobrania aktualnej konfiguracji dotyczącej tego, jakie dane powinny być wyciągane z plików DICOM (przy zmianie konfiguracji należy zrestartować repository checker).

Następnie w pętli, co 60 sekund (wartość konfigurowalna, dla zmiany potrzeby restart) odczytuje z pliku czas modyfikacji ostatnio przetworzonego pliku (jeśli istnieje).

Jeśli czasu nie ma – został usunięty przez odpowiednią flagę w pliku z czasem, wtedy przeglądane są jeszcze raz wszystkie pliki z katalogu.

Następnie, pobiera on listę plików z katalogu, wraz z ich systemowym czasem modyfikacji, sortuje malejąco względem czasu modyfikacji i dla każdego z plików z listy, sprawdza, czy został już wcześniej przetworzony (informacja w bazie danych) i jeśli nie, próbuje wyciągnąć z niego informacje DICOM-owe.

Jeśli się to nie powiedzie zupełnie (np. nieprawidłowy format, plik uszkodzony itp.), zapisuje on na bazie informację o tym, że jest on nieprawidłowy (nie jest on później przetwarzany).

Jeśli się powiedzie, na bazie zapisywane są wszystkie informacje zgodnie z wyciągniętymi informacjami.

Informacje są wyciągane dla pacjenta, serii, badania i samego pliku. Jeśli na bazie już istnieje np. podany pacjent, zostaje on „reuzyty”.

W momencie, gdy dojdziemy do pliku, którego data modyfikacji jest mniejsza, niż ostatnio zapisany czas, przerywamy sprawdzanie i zapisujemy nowy czas, będący najnowszym przetworzonym plikiem.

Repository checker można zatrzymać „miętko” poprzez sygnał SIGINT lub SIGTERM. Wtedy, po jednokrotnym przejściu, jego działanie się zakończy.

8. Opis aplikacji gui i jej możliwości

Ekran główny:

1. Pole, w którym pojawiają się pacjenci znajdujący się w bazie – pacjenci, którzy posiadają co najmniej jeden plik DICOM

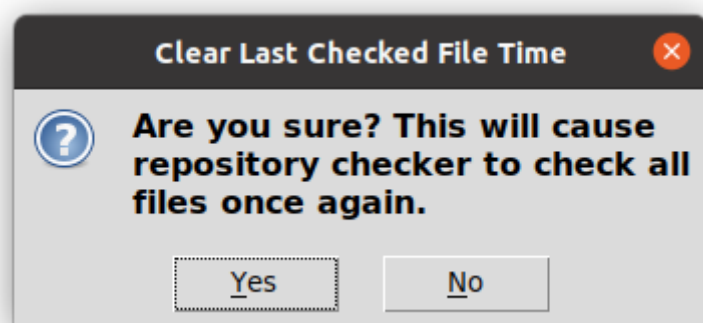
2. Nazwa pacjenta – PatientId oraz PatientName

3. Przycisk służący to przejścia na ekran pacjenta – nowe okno

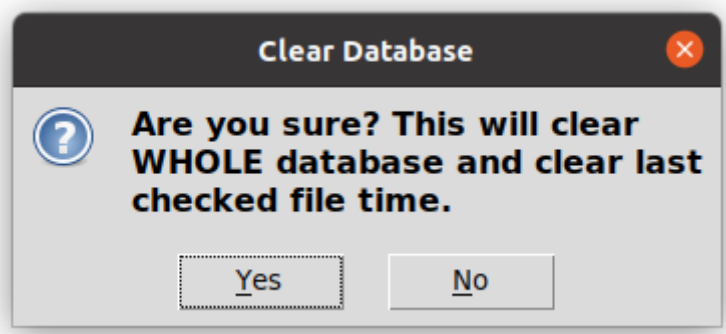
4. i 6. Filtry służące do ograniczenia widoczności pacjentów poprzez ograniczenia do tych, którzy posiadają jakiś plik z datą modyfikacji pomiędzy podanymi datami. Możliwe jest ograniczenie z dwóch stron, z dowolnej jednej, bądź z żadnej. Aby włączyć filtrowanie, należy wybrać odpowiednie daty/datę w 4 i/lub 6, następnie włączyć odpowiedni filtr za pomocą checkboxów 5 i/lub 7 oraz nacisnąć przycisk 8

8. Przycisk służący do odświeżenia listy pacjentów – zarówno, gdy w bazie pojawią się nowe wpisy, jak i przy aplikowaniu filtrów

9. Przycisk służący do wyczyszczenia daty ostatnio sprawdzonego pliku. Po jego naciśnięciu (i potwierdzeniu), przy następnym przejściu repository checker przejrzy wszystkie pliki z odpowiedniego katalogu.

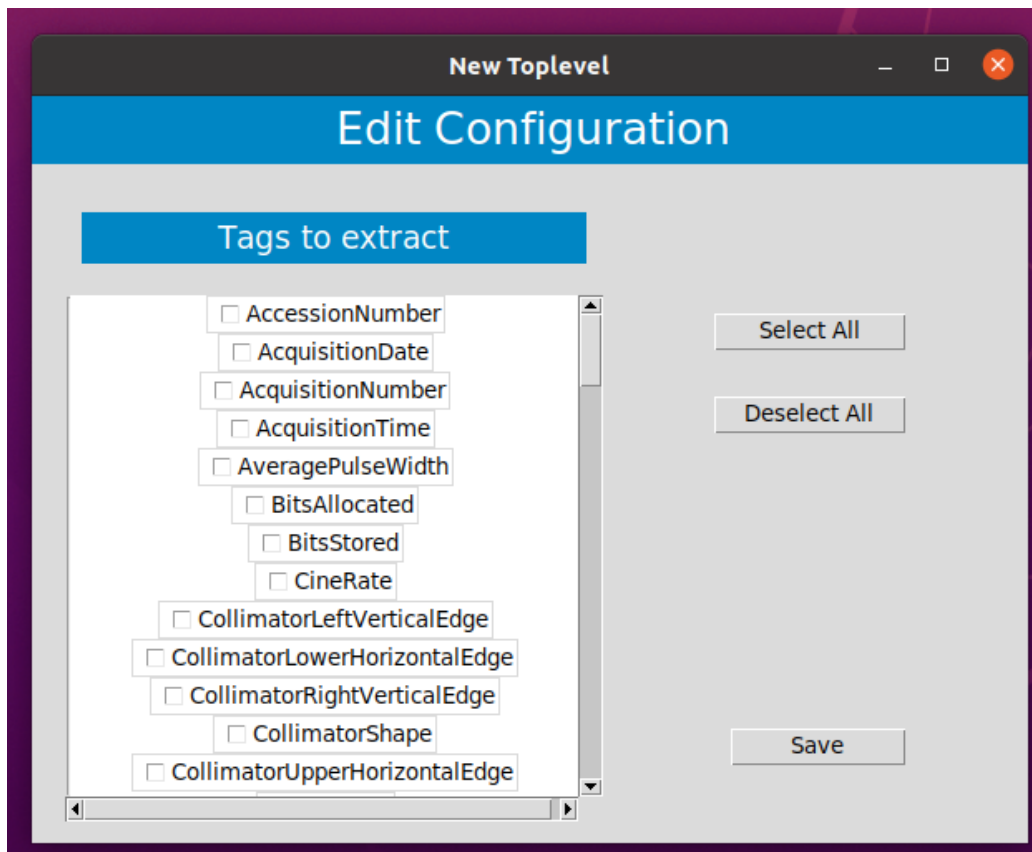


10. Przycisk służący do wyczyszczenia całej bazy danych oraz wyczyszczenia daty ostatnio sprawdzonego pliku. Spowoduje to powtórne przejście po wszystkich plikach z katalogu oraz wstawienie wyników do bazy danych.



11. Przycisk otwierający okno do edycji konfiguracji – po edycji należy zrestartować repository checker.

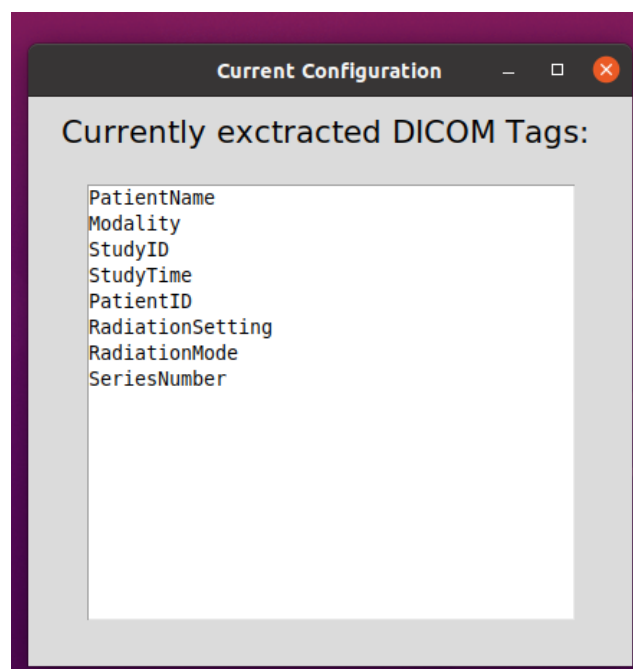
Okno edycji konfiguracji:



Działanie poszczególnych przycisków jest raczej oczywiste. Widoczne do wyboru tagi są zapisane w pliku configuration.py (katalog other_classes) w zmiennej small_list_of_fields (w tej samej klasie jest również zmienna all_keywords zawierająca wszystkie dostępne tagi DICOM).

12. Przycisk służący do wyświetlenia aktualnej konfiguracji – pól DICOM, które oprócz zdefiniowanych dla pacjenta, serii, badania i obrazu, są dodatkowo wyciągane z pliku i zapisywane na bazie.

Okno podglądu aktualnej konfiguracji:



Ekran pacjenta

patient_id 26 2 Generate Report

PatientID 123123123

PatientAge 28 1

PatientBirthDate 00000000

PatientSex 00

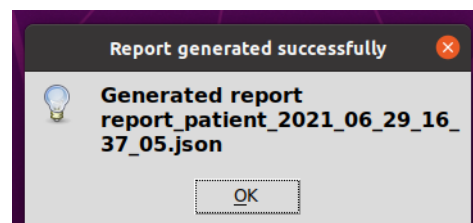
PatientName John Doe

3 Patients Studies

1.3.12.2.1107.5.99.1.22358.30000016091514024250000000016	Open 4
1.3.12.2.1107.5.99.1.21350.300000141102014922953000000004	Open
1.2.840.113619.6.353.298268352034214040993086555746091083988	Open
1.2.826.0.1.3680043.2.1041.4.5.11237098	Open
1.2.840.113619.2.98.4183.1286525519.0.2	Open
1.2.840.113680.1.103.51111.1012999089.60438	Open
1.3.12.2.1107.5.4.3.13452335544320.20030707.42232.312	Open

1. Lista podstawowych informacji o pacjencie
2. Przycisk służący do generowania raportu. Raport to plik json, sygnowany datą stworzenia, zawierający widoczne w 1. informacje o pacjencie, oraz ilość jego badań.

```
{
  "PatientID": "123123123",
  "PatientAge": "28",
  "PatientBirthDate": "00000000",
  "PatientName": "John Doe",
  "PatientSex": "00",
  "patient_studies": 7,
  "db_patient_id": "26"
}
```



3. Lista badań pacjenta wraz z przyciskami do ich otwierania (4, nowe okno)

Ekran badania:

The screenshot shows a web application window titled "Study". It contains a form with the following fields:

Field	Value
study_id	35
StudyInstanceUID	1.3.12.2.1107.5.99.1.22358.30000016091514024250000000016
patient_id	26
StudyDate	20160915
StudyID	123456789
StudyDescription	
StudyIDIssuer	16.09.15-20:16:59-DST-IVS
StudyTime	201658.968000

There is a "Generate Report" button to the right of the form fields.

Below the form is a section titled "Study Series" with a yellow background. It contains a list of two study series:

Study Series	Action
1.3.12.2.1107.5.4.5.137056.300000160915143448828000000499	Open
1.3.12.2.1107.5.4.5.137056.300000160915143448828000000463	Open

Jest on analogiczny do ekranu pacjenta.

Przykładowy generowany raport (dodatkowo mamy informacje na temat pacjenta):

```
{
  "StudyInstanceUID": "1.3.12.2.1107.5.99.1.22358.30000016091514024250000000016",
  "StudyDate": "20160915",
  "StudyDescription": "",
  "StudyID": "123456789",
  "StudyIDIssuer": "16.09.15-20:16:59-DST-IVS",
  "StudyTime": "201658.968000",
  "study_series": 2,
  "db_study_id": "35",
  "db_patient_id": "26",
  "PatientID": "123123"
}
```


Ekran serii

The screenshot shows a web application window titled 'Series'. It features a blue header bar with the word 'Series' in white. Below the header, there is a form with several input fields and a button. The fields are labeled as follows:

Field Label	Value
series_id	41
SeriesInstanceUID	1.3.12.2.1107.5.4.5.137056.30000016091514344882800000499
study_id	35
SeriesDate	20160915
SeriesNumber	8
SeriesDescription	
SeriesTime	205215.000000

To the right of these fields is a button labeled 'Generate Report'. Below the form is a yellow section titled 'Series Images'. Inside this section, there is a text input field containing a long string of numbers and a button labeled 'Open'.

Jest on analogiczny do ekranu pacjenta.

Przy generowaniu raportu, oprócz ilości zdjęć dla serii, dodawana jest także informacja o ich modalności, jak również informacje o badaniu i pacjencie

```
{
  "SeriesInstanceUID": "1.3.12.2.1107.5.4.5.137056.30000016091514344882800000499",
  "SeriesDate": "20160915",
  "SeriesDescription": "",
  "SeriesNumber": "8",
  "SeriesTime": "205215.000000",
  "series_images": 1,
  "db_series_id": "41",
  "db_study_id": "35",
  "Modality": "XA",
  "StudyInstanceUID": "1.3.12.2.1107.5.99.1.22358.30000016091514024250000000016",
  "patient_id": 26,
  "PatientID": "123123"
}
```

Ekran zdjęcia/pliku

Image

Image

image_id	33	Modification time	2019-04-01 10:12:26
series_id	41	Modality	XA
file_id	52		
file_name	233-205-83-75-162-50-186-15-72-119-90-17-255-207-202-228-94-134-225-225		
ImageType	['DERIVED', 'PRIMARY', 'SINGLE PLANE', 'SINGLE A']		
processed_date	2021-07-07 19:25:50		
PixelData	(7fe0, 0010) Pixel Data OW: Array of 12845056		

Image Tags

Modality	XA
PatientID	123123
PatientName	John Doe
RadiationMode	PULSED
RadiationSetting	GR
SeriesNumber	8
StudyID	123456789
StudyTime	201658.968000

5

Comment

Image comment 123456

6

7 Save Comment

1. Podstawowe informacje o zdjęciu/pliku.
2. Czas przeprocesowania pliku przez repository checker.
3. Informacje na temat PixelData obrazu (tylko rozmiar, brak dokładnych danych obrazowych).
4. Czas systemowej modyfikacji pliku.
5. Lista dodatkowo wyciągniętych z pliku tagów DICOM zdefiniowanych w konfiguracji.
6. Miejsce do wpisywania komentarza dotyczącego zdjęcia/pliku. Można go zapisać poprzez przycisk 7.