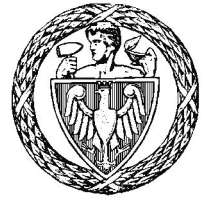


Warsaw University of Technology

FACULTY OF
ELECTRONICS AND INFORMATION TECHNOLOGY



Institute of Computer Science

Bachelor's diploma thesis

In the field of study Computer Science
and specialization Computer Systems and Networks

Structured radiological reporting system

Paweł Paczuski

student record book number 271082

thesis supervisor

Jan J. Mulawka PhD, DSc

WARSAW 2018

Abstract

Structured radiological reporting system.

Design and implementation of a system that can be used by radiologists to create structured radiological reports. The system uses sets of standardized, frequently used phrases to: describe state of patient's body captured by other medical diagnostics methods, provide set of tools that minimize risk of mistake and increase productivity.

Streszczenie

streszczenie po polsku

Contents

1	Introduction	4
1.1	The need for medical diagnostics	4
1.2	Existing solutions	4
1.3	Typical work-flow of a radiologist in Poland	5
1.4	Discussion about presented work-flow	6
1.5	Other ways to create radiological reports	7
1.6	Problem definition	7
1.7	Incentives for the thesis	8
2	Description of the proposed solution	9
2.1	General idea	9
2.2	Assumptions	9
2.3	Goals	10
2.4	Proposed reporting ontology based on ideas from DICOM SR and HL7 CDA	10
2.5	Work-flow of a radiologist who uses the proposed system	12
2.6	Examination template	15
2.7	Productivity improvements	17
2.8	Reporting quality improvements	18
2.9	Integration with existing systems	20
3	Implementation	21
3.1	Technological stack	21
3.2	Internationalization and localization	23
3.3	Templates	24
3.4	Report creator	25
3.5	Report history	28
3.6	Compare reports	29
3.7	Deployment strategy	30
4	Testing and verification	32
5	Validation of the proposed solution and examples of places where it was used	33
5.1	User satisfaction measurements	33
5.2	Teleradiologists	33
5.3	Hospital	34

<i>CONTENTS</i>	3
5.4 Large network of clinics	34
6 Conclusion and ideas for the future	35

Chapter 1

Introduction

1.1 The need for medical diagnostics

Everyday millions of physicians treat injuries and illnesses of different kinds. Before a doctor can plan an individual treatment for a patient, they have to diagnose which organs are in pathological states [1]. This sometimes can be achieved by simply glancing at the body, however, there are many illnesses that require specialized set of tools and methods in order to observe which parts of patient's body are in an unwanted state. Through years of research, many different techniques were established and a separate specialization emerged – radiology. Radiologists focus mainly on analyzing and interpreting diagnostic imagery and as a result of their work they create a document called radiological report which contains description of what can be observed in the image of patient body. Reports may contain description of state of particular organs, measurements (e.g. radius, volume, concentration of certain substances in the blood), comparison of medical condition of a patient observed at different times and description of overall state of the patient.

1.2 Existing solutions

Currently, the research is focused on finding new ways of diagnosing diseases by the use of more advanced equipment or brilliant algorithms that try to automate image analysis [2].

On the other hand, there exist initiatives that try to improve quality of the radiological reports themselves. There are groups consisting of both computer scientists and physicians that try to standardize reports, prepare checklists that require doctors to describe patients' state in particular order and create a set of phrases that will be understood in the same way by all physicians [3]. A lot of work has been done to provide both common medical nomenclature for medical conditions and theoretical framework to describe relations between causes and effects of patients' condition. As there are more and more methods used to diagnose, the amount of information captured increases, so the reporting methodology has to be kept up to date with the state of art. This is why a very specific field – Structural Reporting (SR) emerged. The basic idea is to provide a way to create radiological reports

that conveys as much semantics as possible in a way that is easy to understand. One can find great ideas implemented in such standards as SNOMED SR [4] and also HL7 version 3 Clinical Document Architecture (HL7 V3 CDA). By using these standards, one can encode relations between organs and diseases (causality) in a very regular format. After encoding structure in the report, one can use algorithms to e.g. highlight what changed since last visit, look for diseases that were diagnosed in the specified time range etc. This is very difficult to achieve when reports are stored in plain text. Structural reporting focuses mainly on encoding meaning – the visual representation of resulting reports is a separate matter that is treated as an implementation detail [4]. In spite of the existence of these standards, it is almost impossible to find software that implements structural reporting techniques. One of the most important reasons is the fact that in order to understand benefits of SR, one has to acquire certain level of understanding of the typical work-flow of a radiologist. As there is huge demand for software that is much easier to understand (e.g. RIS and HIS software), most of the effort is made to implement simpler concepts.

1.3 Typical work-flow of a radiologist in Poland

In order to find places where optimization of productivity could be applied, one has to get to know how a radiologist works and what are activities that waste significant amounts of time.

In a medium-sized clinic, medical imagery is captured by a radiologic technologist who then uploads the data to the Picture Archiving and Communication System (PACS) and attaches identification information to the images. Next to the PACS system in most cases exists Radiology Information System (RIS) that is used by radiological staff to keep track of patients treated in the clinic. These systems are used to distribute imagery to the team of radiologists.

Imagery can be distributed in one of the following manners:

- Particular patient is always serviced by the same radiologist.
- RIS system acts as an accumulator of service requests and radiologist decides which patient they should focus on now.
- Certain types of medical examinations are always assigned to a radiologist that is specialized in describing them.

After receiving diagnostic imagery, a radiologist uses special software called Viewer[8] to navigate through images, make measurements by using visual tools like virtual ruler and examine what is the state of patient's body. Figure 1.1 shows an example of the Viewer software. In parallel to this, the doctor uses text editor and describes what he or she sees in the images. Some RIS systems are equipped with a simple text editors that allow doctors to write the report inside the system without using external editors.

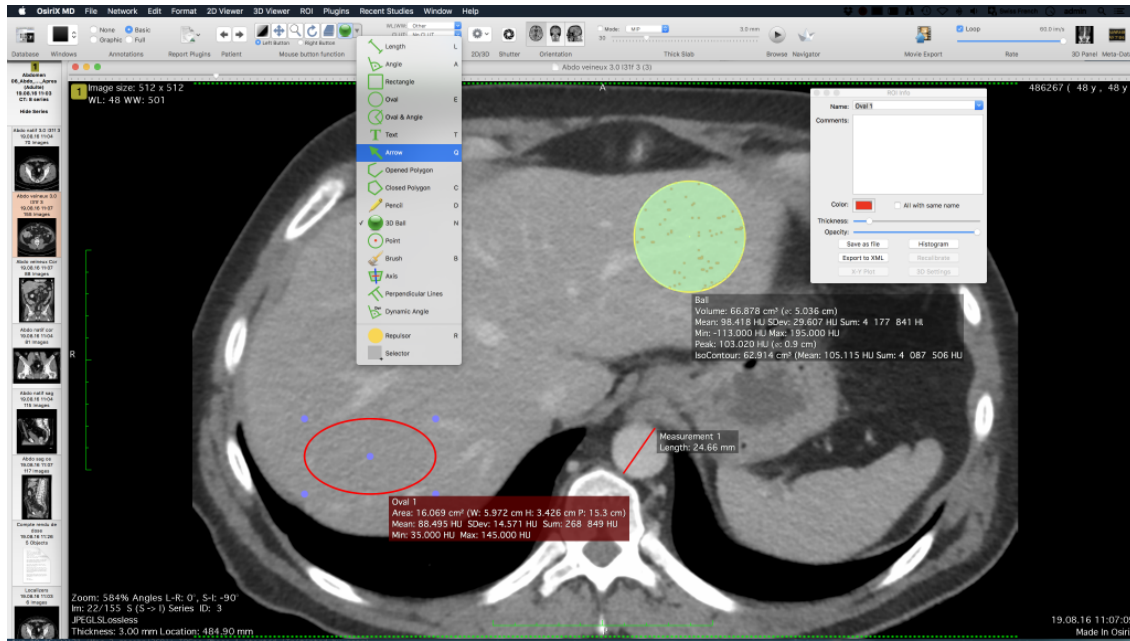


Figure 1.1: OsiriX is one of the most popular image viewers used by the radiologists

1.4 Discussion about presented work-flow

1.4.1 The good parts

Viewer software provides expected functionality

Viewer software has a very stable position on the market and is perfectly tailored to the needs of a radiologist. It often uses advanced techniques of computer graphics to present patient's body as accurately as possible.

PACS systems provide centralized place for storing images

They implement functionalities that allow for archiving medical imagery. Lifetime of medical images is controlled by the government and these systems must take this into account. PACS also make it easy to distribute images not only withing local networks but also to any doctor who has the remote access granted.

RIS systems make it easy to exchange data between physicians

There is no need need for the radiologist to deliver the radiological report to the other physician personally as it is done automatically. Also, RIS systems provide good means of presenting medical records from different medical specializations allowing for interdisciplinary diagnostics.

1.4.2 The bad parts

Focus on text rather than semantics

It is expected that the radiologist will produce a consistent, ordered report by means of text editors. Some RIS systems provide basic text formatting functionality like italicization, underlining but they are limited to the textual presentation of the report. There are no dedicated tools to encode relations in this representation.

Each radiologist has their own style of writing

Usually there are no structural expectations about the resulting report. Each radiologist can have their own style of writing, order of organs observation, text formatting. This leads to the waste of time as people who read the report have to make some effort to infer the meaning from plaintext.

Selective description

It is very frequent that radiologists include in the report only things that they consider bad for the patient. This makes the report more goal-oriented but it means that it is useless to get to know the overall state of patient body.

Copy-paste

Radiologists try to solve the problem of typing on their own by creating templates that contain certain pathologies listed and what they do is execution of the commonly called 'copy-paste' method to create report content. Sometimes they do not notice parts of copied text that are different from the actual state of the body, so the reports may contain observations that are false.

1.5 Other ways to create radiological reports

In many English-speaking countries the work-flow of a radiologist differs in the way the radiological report is generated. A radiologist may record their voice while describing the imagery vocally. The recordings are then transcribed using either speech recognition algorithms or manually by technologists. Having a good skill of typing by a radiologist is not required, only knowledge to interpret imagery is used. This approach, however, has some architectural disadvantages. More personnel are needed – technologists, who transcribe the recorded voice [9]. Also, it is difficult to make changes to what has been said. In some cases, a mistake can be made while transcribing the text [10].

1.6 Problem definition

After analyzing bad parts of the presented work-flow and general situation on the market – demand for radiological services increases but the number of radiologists appears to

be constant, the author decided to design and implement a system that would be used by radiologists to encode semantics of diagnostic imagery in a form that can be easily transformed to the human readable form, analyzed by algorithms.

1.7 Incentives for the thesis

A primary goal is to find a solution to the problem of not satisfactory productivity of radiologists by implementing a program used to create structured radiological reports. The system should apply sets of standardized, frequently used phrases to: describe state of patient's body captured by other medical diagnostics methods and should also provide set of tools that minimize risk of mistake and allows radiologists to create reports faster.

Chapter 2

Description of the proposed solution

2.1 General idea

2.1.1 Area of interest

The system that is proposed in this thesis focuses entirely on the part of the typical radiological work-flow in which a radiologist focuses on the textual description of what can be seen in the diagnostic imagery.

2.1.2 Contextual suggestions

The author has analyzed several hundreds of anonymized radiological reports and observed that a lot of time could be saved, if a radiologist would at any time select text fragments from a predefined set of possibilities that are appropriate to the current context. This approach can be similar to the way Integrated Development Environment (IDE) for statically typed languages (e.g. Visual Studio supporting C#) are suggesting what the programmer can write based on the namespace, class, scope they currently edit.

In the case of programming languages, the problem is defined in a stricter way as languages are formally defined using grammars [11]. Radiological reports, however, are written using natural language, so there always exist some exceptions. Ideally, a radiologist would always use phrases that are well-defined in the standards, but the the act of scanning long list of possibilities by a radiologist could take a lot of time making it less appealing to the practitioners.

2.2 Assumptions

After observation of actions taken by radiologists while working in environments like hospital, medium-sized clinic, independent teleradiological practice the following assumptions are suggested:

- Radiologists prefer using mouse to keyboard.
- The user interface of the program should be simple.

- Reports must be rendered in a way that allows for copying using clipboard as radiologists developed custom ways to save drafts of reports.
- The system should allow for exporting reports into formatted text and pdf.
- Text formatting should be based on semantics that is encoded by a radiologist.
- The system should favor reports that describe the whole state of body, not only organs in bad condition.
- The system should allow to automatically include in the report segments of text that are (almost) always used (sometimes due to some legal regulations).

2.3 Goals

- Minimize the time radiologist uses keyboard.
- Split report into structural parts that allow to express description of patient's state.
- Any fragment of template text should be editable.
- Maximize number of reports that can be generated by a radiologist in a unit of time.
- At any time allow to include custom phrases that are not defined in the template.

2.4 Proposed reporting ontology based on ideas from DICOM SR and HL7 CDA

In order to provide context for the radiologist at any given moment, it is proposed to split report into the following nested structures:

1. Examinations
2. Organs
3. Properties

Figure 2.1 shows how report structures are related by semantic relations.

By using this ontology, a radiologist can create a report that is similar in structure to a tree (as it is presented in figure 2.2). At any given moment, the doctor modifies the tree at a single level, which allows for suggesting what are the items that can be included. The idea was taken from statically typed languages which, thanks to their strictness, allow for coding with smaller number of mistakes at the lexical level [13].

¹For simplicity, only names of nodes are presented (associated meta-data were omitted)

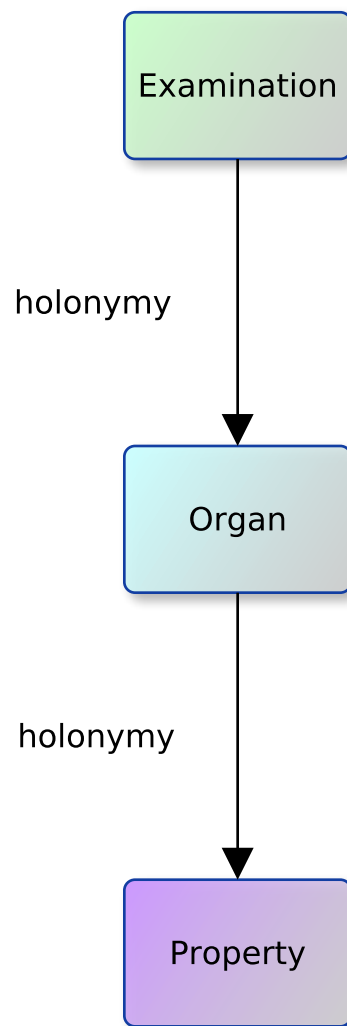


Figure 2.1: Simple ontology representing entities and relationships between them encoded in reports generated using the proposed system

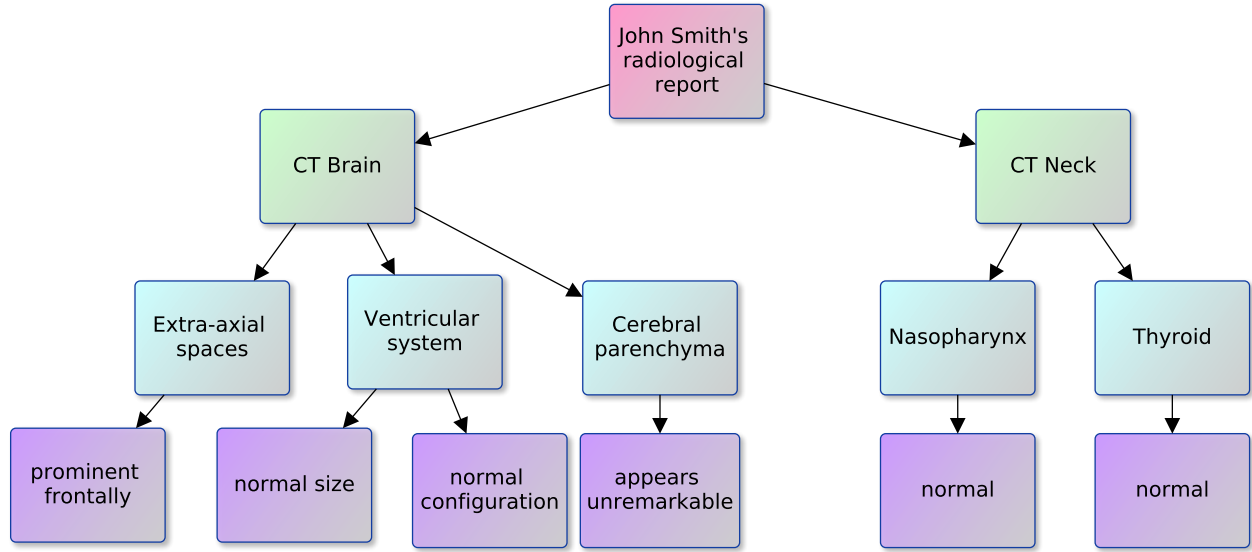


Figure 2.2: Typical structure of a radiological report created using the proposed system¹

2.5 Work-flow of a radiologist who uses the proposed system

In figure 2.3 a typical work-flow of a radiologist is represented in the form of a flowchart.

2.5.1 Connotation

Besides having text, each property can have a semantical attribute called connotation. Connotation is used to highlight the impact of this property on the result of the organ and examination. This attribute can have one of three values: positive, negative and neutral. Figure 2.4 presents connotation selection box that is present next to the property text editor.

2.5.2 Meta-data

Report may also contain some meta-data which can be used to identify the patient whose body is described in it. Shape of these pieces of information strongly depends on the RIS system used in a clinic. Text segments present in meta-data have no special influence on the semantics of report so it has no detailed description in this work. Figure 2.5 presents user interface used to input commonly used meta-data.

Names for ontological structures shown in Figure 2.1 were chosen so it is easier for radiologists to imagine what they describe, however, they can be used not strictly (e.g. in a CT scan it is possible that a radiologist adds an organ that contains details about intracranial hemorrhage which is a kind of bleeding within the skull [12] but not an organ in its very meaning). This is one of the known weak-points of the proposed ontology.

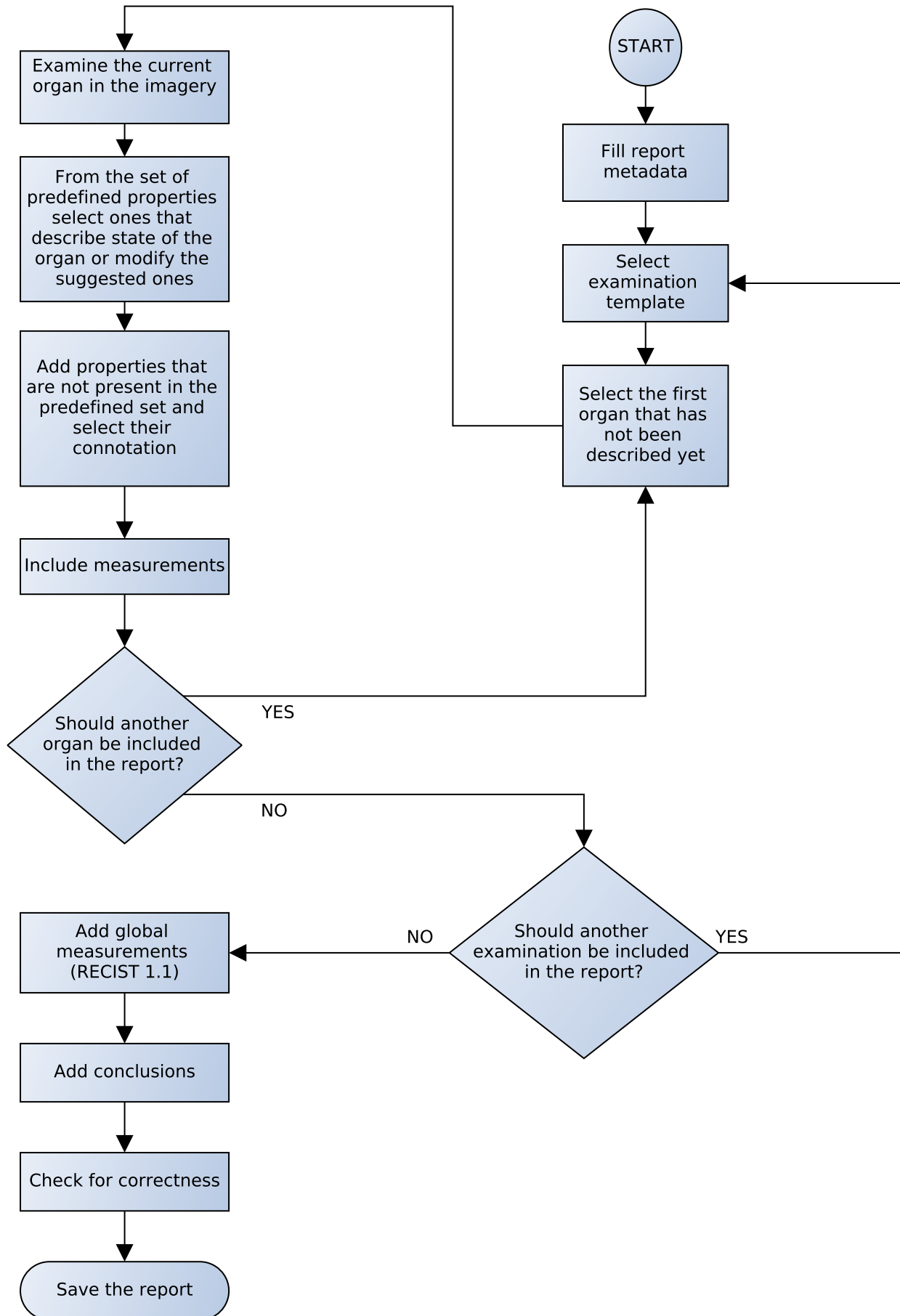
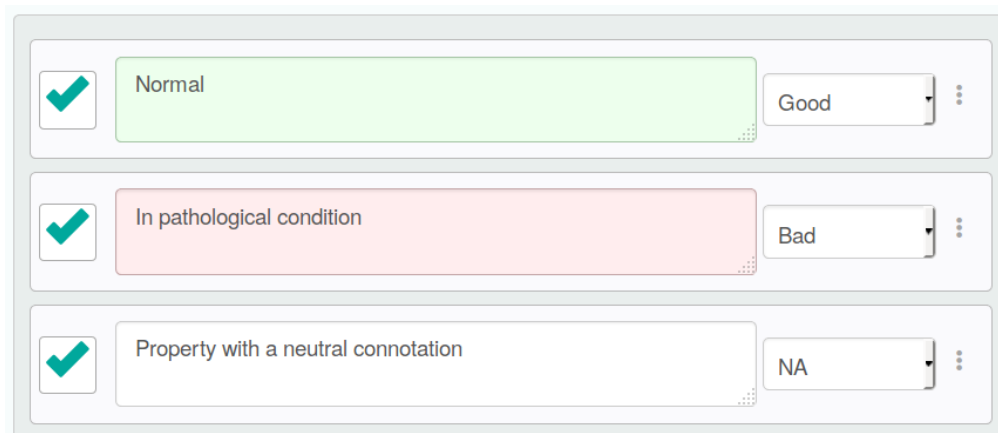
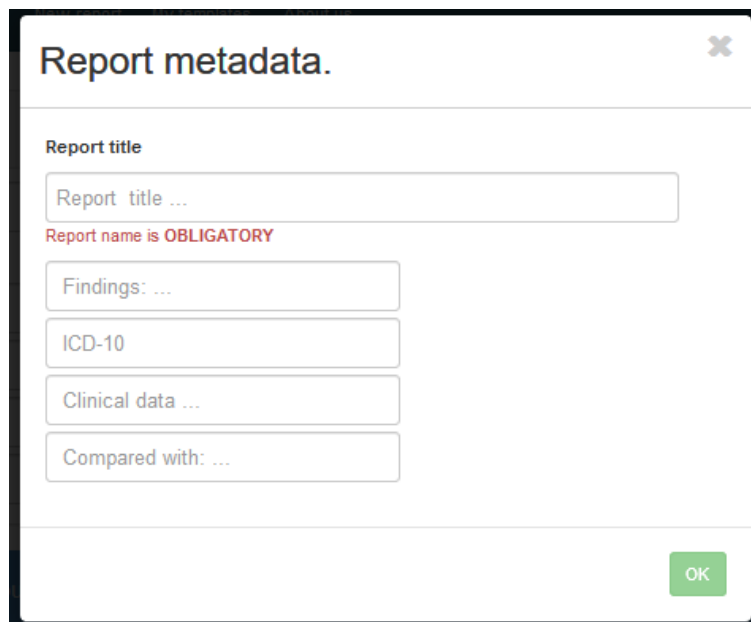


Figure 2.3: Work-flow of a radiologist who uses the proposed system to create contents of a radiological report



<input checked="" type="checkbox"/>	Normal	Good
<input checked="" type="checkbox"/>	In pathological condition	Bad
<input checked="" type="checkbox"/>	Property with a neutral connotation	NA

Figure 2.4: Connotation can have one of three values: positive, negative and neutral



Report metadata. ✕

Report title

Report title ...

Report name is **OBLIGATORY**

Findings: ...

ICD-10

Clinical data ...

Compared with: ...

OK

Figure 2.5: Report meta-data allow user to specify general information about new report

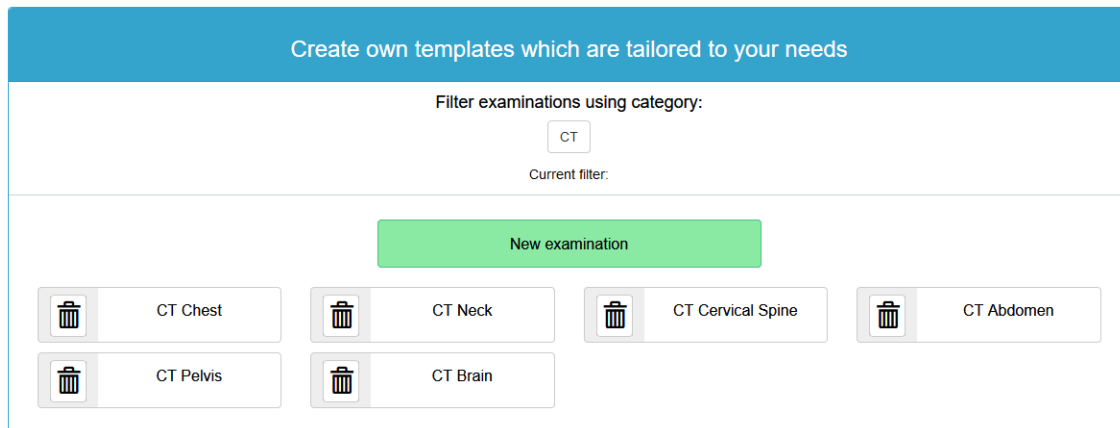


Figure 2.6: List of templates that can be edited by the user

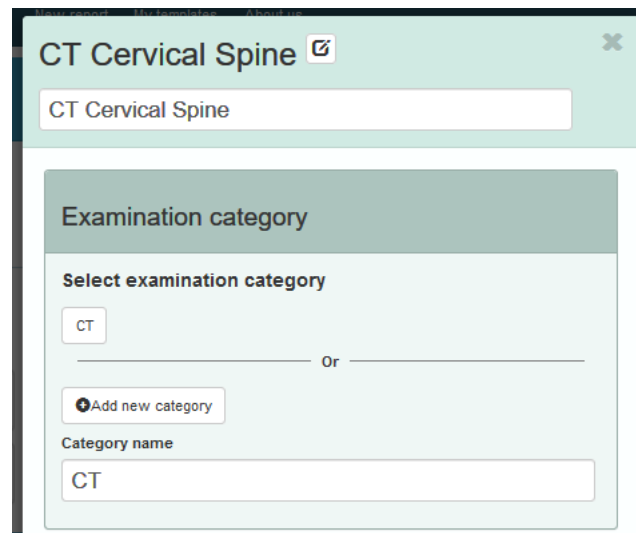


Figure 2.7: GUI used to edit template meta-data

2.6 Examination template

Examination template (or simply a 'template') consists of an ordered list of all organs that could be described when creating a report. It also has some portion of meta-data used for filtering like category, discipline. Templates use the same ontology as the reports as templates can be imagined as reports with all properties, organs included.

Examination templates can be prepared by the leader of radiologists team, so the team can use common language to describe diagnostic imagery. Templates prepared by the leader constitute the so called set of public examinations.

If a given template is not sufficient for a particular radiologist, they can create their own template from scratch or extend the existing one (an action that for people familiar with git version control system can resemble forking a repo [14]).

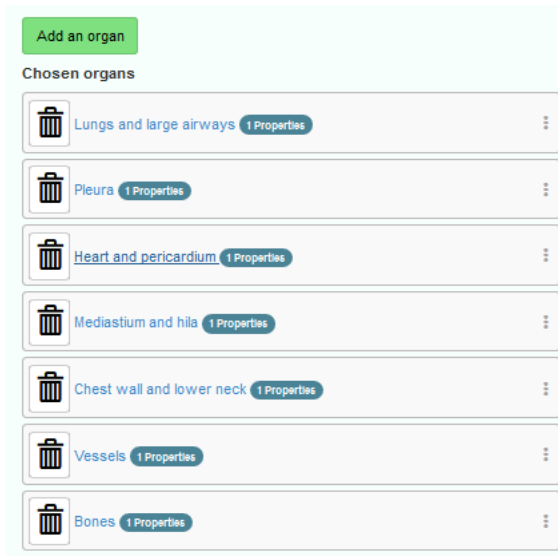


Figure 2.8: GUI used to edit organs

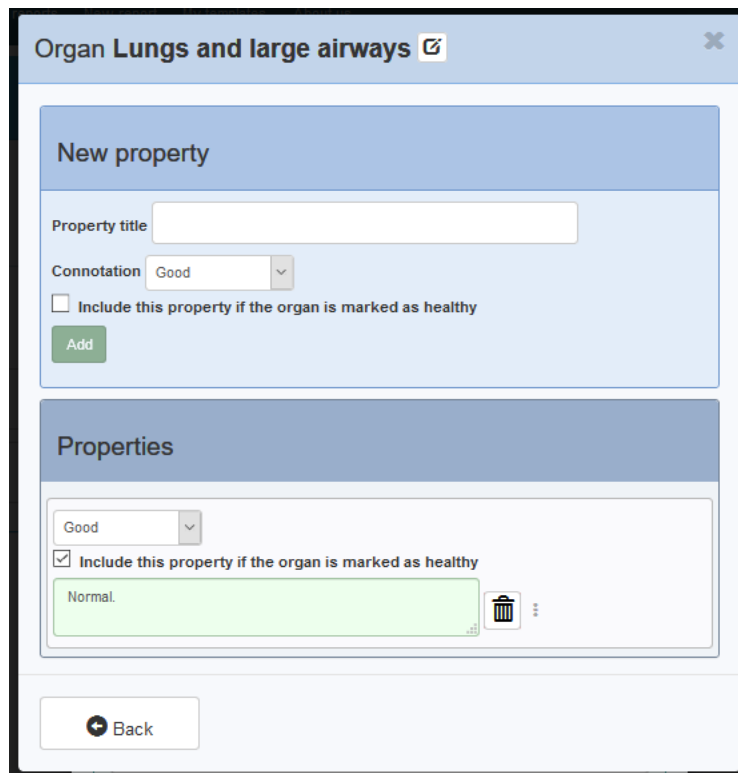


Figure 2.9: GUI used to edit properties of an organ. User can predefine connotation and tell the program that this property can be automatically included if the organ is marked as healthy

2.7 Productivity improvements

2.7.1 Using templates

The goal to increase productivity of a radiologist is achieved mostly by decreasing the average time a doctor types on keyboard. Thanks to the usage of examination templates, the time spent to create the schema of the report is shifted from the doctor to the leader of the radiologists team. It can take a lot of time to create a template that is both universal and correct but it is a one-time-only investment. The more templates are created using a template, the more it pays off in the long run.

2.7.2 Mark organ as healthy

In the proposed system, each property has additional attribute (it can be set in template editor) named "include if organ healthy". When a radiologist creates a report, an organ can be marked as healthy, which means that all properties that have this attribute set to true are automatically included in the report. This allows to quickly add properties that are very commonly used. In order to minimize the possibility of inclusion a property that does not correspond to the actual state of patient's body, only properties with positive connotation can have this attribute set.

2.7.3 Calculators

There exist many standardized numerical methods used to assess whether values of certain parameters are in the range suggesting poor health condition, e.g. one of the most popular parameter is the body-mass-index (BMI) which is calculated using formula:

$$BMI = \frac{mass}{height^2} \quad (2.1)$$

After calculating this index, a doctor has to look through tables to check whether the value is usually attached to people with obesity or not. BMI is only one formula out of hundreds of techniques used by doctors. It is not integrated, however, into the proposed system as it was decided that it is not used frequently by end-users.

RECIST 1.1

During the design of the proposed systems several radiologists who specialize in oncological medicine reporting suggested that a method called Response Evaluation Criteria In Solid Tumors (RECIST 1.1) is of special importance to them. This method is used to assess whether tumors in cancer patients improve, stay the same, or worsen during treatment [15]. As there are hundreds of similar methods used by radiologists, it would be impossible to implement them in the proposed system all at once, so it was decided to use RECIST 1.1 as a proof of the concept of calculators that can be used in structured reports. A radiologist, while creating a report, can measure size of lesions and write down the results of measurement in the report. Proposed program should recognize that the numerical value

Figure 2.10: Measurements data are automatically parsed when user types text into the property textbox

represents a measurement (as shown in figure 2.10) and should suggest whether it should be included in the calculation of RECIST 1.1 or not (as shown in figure 2.11).

2.8 Reporting quality improvements

2.8.1 Standardized nomenclature

There exist several attempts to standardize the language doctors use to describe precisely medical imagery. Templates created for the proposed structured reporting system can be created with the help of terminologists who specialize in systematized nomenclatures like SNOMED and LOINC to create reports what can lead to improved reporting quality as a radiologist does not have to explore huge volumes of precise textual definitions of terms used.

2.8.2 No repetitions

Templates can be interpreted as TODO-lists containing tasks which have to be done while describing medical imagery, so a radiologist can be sure that nothing was omitted or no item appears twice in the report as a result of a mistake.

2.8.3 Reports have ordered list of elements

Reports created using the proposed system can be represented as trees, nested lists of items, so it is easy to spot relationships between properties and organs and also differences between two reports describing one patient.

2.8.4 Highlighting the most important pieces of information

The most important pieces of information can be highlighted using proper value of connotation. When one of properties of an organ should be interpreted as a main cause of

Select which measurements should be included in RECIST 1.1 calculation

CT Chest

Lungs and large airways

Text:

3mm[4mm]

Extracted values:

3, [4]

☒ Include ☒ Node

Text:

1mm[3mm]

Extracted values:

1, [3]

☒ Include ☒ Node

Calculate RECIST 1.1

Sum of previous measurements: 7
Sum of current measurements: 4
Difference (absolute): -3
Difference (relatively): -42.857%
Status: Complete response

Figure 2.11: Just before confirming the completeness of the report the user is asked to decide which measurements should be included into RECIST 1.1 calculation¹

Normal ; In pathological condition .

Figure 2.12: Exemplary representation of connotation in the rendered version of the report: positive connotation is represented as green background of the text, negative — red, neutral — transparent.

health problems it is marked as a pathology. When the report is then rendered as a document, this property can have e.g. different font color, some text-decoration element or different background color (Figure 2.12).

2.8.5 Shift from difference reporting to the current state reporting

There exist practices quite popular among radiologists mostly from smaller clinics that are considered as bad among professionals. One is to include in the radiological report only properties of body which they consider as bad for patient's health. Due to the subjective nature of the report there were cases when a radiologist skipped a finding that was important stating the patient was healthy leading to more difficult and expensive treatment later. There were also cases when a radiologist was unable to compare current state with descriptions of several images that were taken at different times, only with the last one. The doctor stated that there is a progress in treatment, but in comparison to the second from the last description, the health condition worsened. [16] Templates created using the proposed system can favor radiological reports that are more self-contained, it means they

describe both bad and good (with respect to connotation) properties of patient's state.

2.8.6 Update of the report

Having a previous report, a radiologist can create another one that will be an update to the former. This procedure is often applied during chemotherapy to see whether tumors react to the treatment. A software functionality that allows the radiologist to introduce small changes to the report without rewriting it, would increase the quality of reporting as the resulting report would be still self-contained.

2.9 Integration with existing systems

As there are many ways to store radiological reports. Many radiologists work remotely for several companies that have different RIS systems. From the practical point of view, it is very difficult to integrate the system with all of the systems as it requires cooperation of both developer of the proposed system and the developers of the RIS system.

Current approach to integration is very robust, works for all of the systems used nowadays but is not very elegant from the point of view of software engineering. After generating the report using the proposed system, it is the radiologist's duty to manually copy the resulting report to the RIS system. It is simplified in a way that the report is automatically copied to the clipboard after clicking on its content.

On the other hand, the proposed system was designed in a way that, at least theoretically, makes it easy to create communication channels that can be used to integrate this system with existing RIS systems.

Chapter 3

Implementation

The proposed system was implemented as a web application. It is split into two parts: front-end and back-end.

3.1 Technological stack

3.1.1 Back-end

C#

As a main technology that is used to structure data that is displayed to the user or received from them, ASP.NET C# framework was used. C# is an imperative, strongly-typed, object-oriented language that is used on the market to create enterprise software. There is a high availability of good quality tooling that makes development in this language faster and less error-prone.

ASP.NET

The ASP.NET application is created using MVC pattern (Model-View-Controller) that is widely known for helping to create applications that consists of loosely coupled, plugable components. Some interactions with user interface result in HTTP requests that are then mapped by the server to routes that are forwarded to Controllers. Controllers are used to react to user action, instantiate proper classes which implement application logic (they are part of the Model), feed them with sanitized and validated user input and invoke proper methods. When the results of execution are present, Controller finds a View that is expected to be used to present results to the user. View can be treated as a template for the data that is returned from Controller. It can be an ordinary HTML file with placeholders for certain pieces of information (it can also include formatting) or even JSON or XML documents that are only concerned with shape of the data, not the visual representation (e.g. formatting).

Entity Framework

Entity Framework is a set of libraries that is used to create high-level abstract model using C# that are then converted to sets of relations and attributes in the underlying DBMS. It can also be used to transform LINQ constructions to SQL queries taking into the account vendor-specific features of many DBMS. The approach of first creating C# classes that represent Model and then automatically converting it to the database relations and attributes (scaffolding) is called Code First Approach. It can be especially useful when the schema of database changes very often. When changes are introduced to the code representing Model, Entity Framework can be used create a migration – set of transformations for DB schema that after applying to the DB will keep code and schema synchronized.

3.1.2 Database

MSSQL server is used as a DBMS because it is a battle-tested, well-documented solution that can be easily paired with ASP.NET and Entity Framework's code generation methods are best optimized for this particular DBMS.

3.1.3 Front-end

Not all interactions with the application result in HTTP requests sent to the server, many of them are about shaping data that has already been received from the server. In order to make use of this fact main part of the application – tabs: New Report and Templates were developed as a Single Page Applications (SPA) in order to minimize the time that user has to wait between actions they take. Technologies used in these parts are JavaScript and AngularJS framework.

JavaScript

It is a very popular scripting language that can be executed in any popular web browser. Version of the language used in this project is ECMAScript 5.1 compliant.

AngularJS

AngularJS is a JavaScript framework that helps with building MVC SPA applications. It has several mechanisms like two-way-binding that reduce size of the code needed to implement Create, Read, Update, Delete (CRUD) functionality.

Razor views

The remaining front-end subpages are generated using view engine present in ASP.NET called Razor. These views are responsible for interactions like logging in , registration (backed with ASP.NET identity mechanism), listing reports, searching for particular report etc. They are generated on the server side and sent to the user as HTML and rendered by the browser.

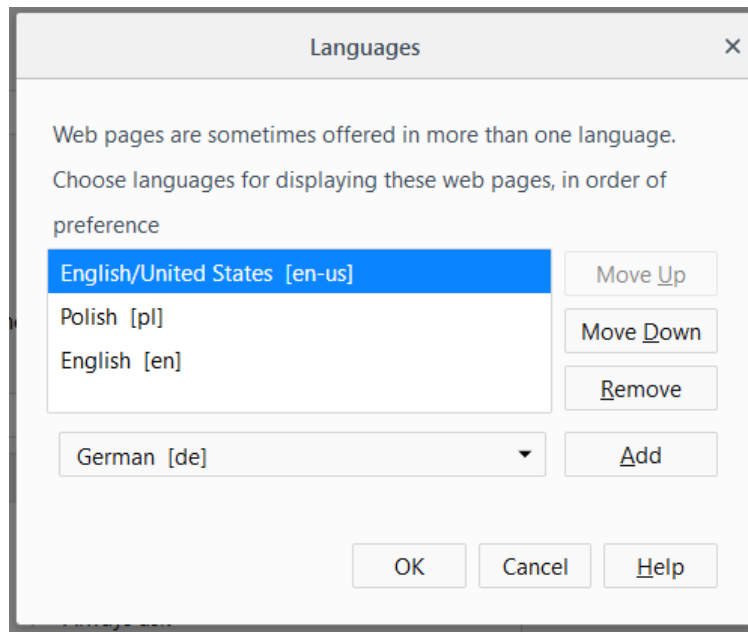


Figure 3.1: Language preferences list present in Firefox 54

3.2 Internationalization and localization

The application was designed to be accessible by radiologists from many countries, so support for many languages is one of its core functionalities. The application automatically detects user preferred language by looking into user browser settings. Each of the most popular Internet browsers allows users to specify list of preferred languages used by web-pages. The presented solution scans this list and chooses first language it supports, if there are no languages in the list that are supported, English language is presented to the user by default. Figure 3.1 shows how this list can be manipulated in one of most popular web browsers.

At the back-end there are dictionaries for each supported languages that contain mapping name -> value. In Razor views names are used to specify where particular piece of text should be placed. During HTML generation, the application resolves user language and refers to proper resource. Figure 3.2 presents contents of a dictionary for ReportWizard for Polish language. Adding support for another language is simply providing values for existing names in the dictionary.

For SPA parts of the program the situation is slightly different as DOM elements representing most of their content are generated at the user side with the help of AngularJS. In order to have single source of truth about translations, it was decided to serialize contents of dictionaries to JSON format, and send it to the user with page template. At the front-end the translation dictionary entries are deserialized and made available to JavaScript functions. This solution makes it faster to edit and version translations as they are kept in a single place. One of drawbacks can be the performance of this solution, but it was measured that for number of dictionary entries used in this application, the decrease in performance is negligible.

Name	Value
NameOfTheContrastAgent	Nazwa środka
NextStep	Dalej
Node	Węzeł
Obligatory	Obowiązkowy
OrganName	Nazwa organu
Organs	Organy
OrganSelection	Wybór organu do opisu
OrganTip	Kliknij, aby otworzyć okno wyboru organów
Preview	Podgląd opisu
PreviousReportContent	Treść wykonanego wcześniej opisu
PreviousStep	Wstecz
Properties	Właściwości organu
PropSelection	Wybór właściwości organu, które zostaną zawarte w opisie. Możliwa jest zmiana treści właściwości.
Quantity	w ilości
RecistTip	Należy wpisać wartość w nawiasie kwadratowym np. [12mm]

Figure 3.2: Dictionary for Report creator for Polish language

3.3 Templates

Templates can be manipulated by the user at the examination level. Users can create them, edit, remove. In order to keep things simple and less error-prone, they are treated as immutable in the database. If user edits a template, the previous version of the template is deleted and template version with edits is inserted into database. The only thing that is not changed in this process is its primary key. This makes reasoning about database queries much easier and has satisfactory performance for templates used by most radiologists. Thanks to the use of database transactions, one can be sure that removing contents of an existing template and inserting contents of a template with modifications is performed atomically.

3.3.1 Feeding front-end with data from database

After browser finishes loading static files (scripts, fonts, etc.), an additional HTTP request is created to download report templates attached to the current account. Server handles the request by querying database, instantiating model classes that correspond to the data received from database. Finally, server serializes reports data to the JSON format and sends it to the front-end. When front-end receives JSON data, a success callback is invoked and sets the data as current view model. From now on a user can edit any existing template: add properties, organs, remove them, change connotations, reorder elements. Thanks to the two-way-binding mechanism, user interface can be directly connected with references to the fields of an object so the developer can focus on the visual representation of the editor. In order to make editor more robust, length of text contained in it is measured on each onChange event and its height is adjusted, so all its contents are visible to the user.

3.3.2 New Examination

User can create new template by clicking on New examination button. Then they are asked to input title of new examination. One of requested features was the ability to copy contents of existing examinations what is similar to forking in git. In order to achieve this, when modal window with new examination's title is opened, front-end loads from server list of user-owned examinations concatenated with list of publicly-available examinations. The latter allows users to quickly customize templates prepared by the administrator.

3.3.3 Edit examinations

On page load all templates that are owned by the user are downloaded. Any modification can be applied: renaming, reordering of elements, setting connotation, etc. When user clicks Save button, actions related to the immutability of templates described in section 3.3.1 are performed.

3.4 Report creator

3.4.1 Data loaded lazily

Report creator is the most optimized part of the application. It is the place where user spends most of their time and it has to respond to any action immediately. It is not easy to predict which examinations will be used in a particular report, so all of available examinations must be ready to be loaded. At the page load, only list of available examinations' titles (user-owned and public examinations) is loaded from the server. When the user clicks on examination's title, the rest of template's content is loaded via HTTP call and ready to be modified and included in the report. Figure 3.3 shows the initial screen of a template editor with titles of templates listed.

3.4.2 Caching at the user side

When radiologists are specialized in a particular field, they have subset of examinations that they describe most often. Loading examination templates separately each time they create a report, would be a waste of time. In order to solve this issue, a caching mechanism at the user side was designed. After loading, JSON data describing examination elements are saved to the storage managed by the browser called `localStorage`. It is a dictionary-like storage that can be used from JavaScript to save string values. When user tries to open an examination, the `localStorage` cache is checked whether this examination is present. If the examination is in the cache, it is immediately shown to the user. Otherwise, an HTTP request to the API is issued to download data.

¹Initially only a list of names of available templates is presented to the user. Contents of the template are downloaded after clicking on the pill representing a template

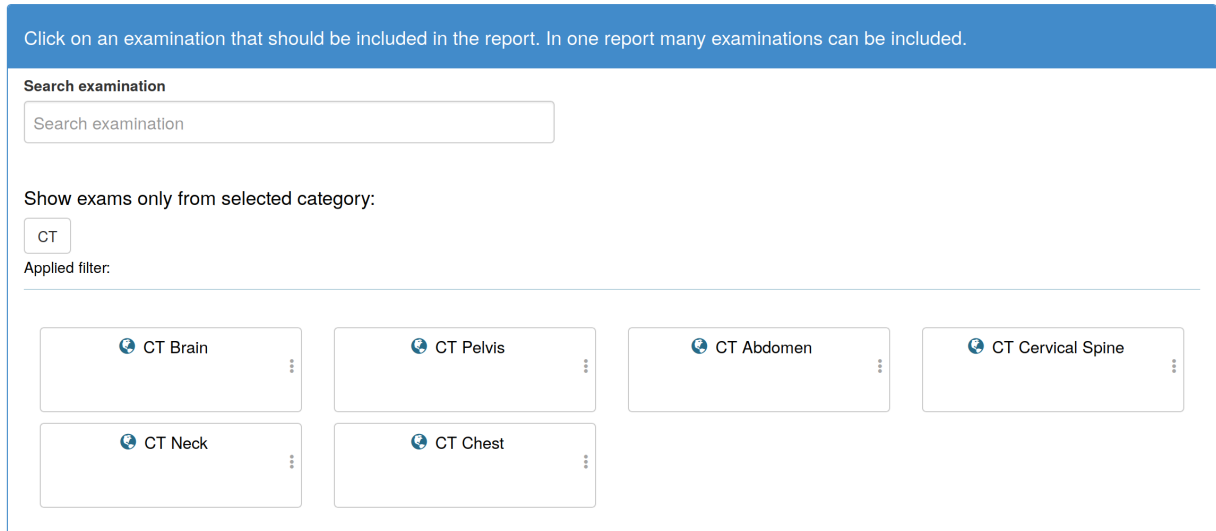


Figure 3.3: A window demonstrating list of examinations to the user¹

Cache invalidation

As templates contain no sensitive information, they can be stored in `localStorage` without restrictions regarding synchronization of cache invalidation with session expiry time. However, the invalidation has to happen when user updates an examination in the Templates tab. In order to keep things simple, the whole cache gets purged anytime user navigates to the Templates tab.

3.4.3 Editing functionality

As user navigates through the levels of the report structure, the program signifies this by opening modals windows one on top of another. This results in the following behavior: when user makes decisions that have the biggest impact on the shape of the report – edits report meta-data, selects examinations that will be used – the program is at the 1st level of depth, no modals are present. When user decides to include an examination, a modal is opened hiding all information that is not necessary at the moment and the user is shown list of organs with preview of the report (As shown in figure 3.4). Work-flow is designed to favor selecting Examinations/Organs/Properties from predefined sets. It was measured that with suggested use of the templates on average 82% of report's content can be generated from template contents. As most of the content can be added to the report by simply clicking on a check-boxes, a lot of thought went into designing interaction model that allows radiologists to look at sequence of suggested elements and make a binary decision 'yes' or 'no' whether an element should be included in the report or not. As the program can be used on any device, it can be interacted using mouse or using touch. When the program is executed on a device with mouse, the font-size and check-boxes are smaller as user uses mouse to precisely click on them. When the program is executed on a smaller device like tablet or smartphone, some elements are shrunk but check-boxes

Examination name

CT Brain

☐ Contrast agent was used

OK

Preview

Select an organ

Organs

Mark all organs as healthy

Check all properties with good connotation in all organs

Search organs

Extra-axial spaces	N	Intracranial hemorrhage	N	Ventricular system	N	Basal cisterns	N
Cerebral parenchyma	N	Midline shift	N	Cerebellum	N	Brainstem	N
Calvarium	N	Vascular system	N	Paranasal sinuses and mastoid air cells	N	Visualized orbits	N
Visualized upper cervical spine	N	Sella	N	Skull base	N	Impression	N

Add new organ

OK

Figure 3.4: List of organs defined in a CT Brain examination template

become bigger, so it is easier to tap them.

3.4.4 Predefined but modifiable

In some situations edits must be made to the predefined pieces of text. This was solved by putting any report text in a slightly customized text-area that is attached to AngularJS context. The text-area is aware of the length of text it contains and automatically fits its height to show the whole text to the user at any time.

3.4.5 Live preview

It is crucial for the user to always be aware of the impact of edits they make to the report contents. Live preview mechanism was designed specifically to solve this. Any significant change of context or report contents sends an event to the function that filters elements that were selected and generates report's textual representation. As this function can take a lot of time to execute, it cannot be left to AngularJS to decide when it should be called (infamous `$digest` cycle [17]) as it would have a severe impact on performance. AngularJS internally observes arrays of data that are attached to its scope to look for changes, so it was decided to call this function only in several situations, e.g. when report-context is switched, a property is checked etc. but not on every single text-change of the report. This allows the radiologists to have a close-to-live preview of resulting report while focusing on selecting predefined pieces of text.

3.4.6 Storing a report in the database

This is a very important decision from the architectural perspective. Format of the saved report can impact the performance of retrieving saved report, interoperability with other systems, etc. The format of stored report could be a topic for a very broad discussion. In order to maintain the simplicity of the whole project, it was decided to store JSON representing the report in the database. Many of the DMBS have special functionalities that are based on ideas used in NoSQL databases, especially document databases to allow for storing JSON objects, validating and querying them as if they were part of the relational data. This makes it easy to store data that can have numerous optional fields. Unfortunately, MSSQL Server does not support storing JSON objects natively (Microsoft calls their limited way of supporting this format as *built-in JSON support*[?]), however, it is possible to store JSON string representation in a column of type NVARCHAR and then validate it on SQL INSERT by the use of ISJSON function.

3.5 Report history

Report history is a part of the program that is used to list saved reports, search for particular ones, request pdf version of reports, sending report content via email. Users can invoke on existing reports such important actions as editing them, deleting them, comparing them. As one user may have created thousands of reports using this program, this

Previous report content:

John Smith

CT Brain
Extra-axial spaces: Normal .
Intracranial hemorrhage: None .
Basal cisterns: Normal .

CT Pelvis
Reproductive organs: No pelvic masses .
Ureters: Normal .
Bladder: Normal .
Bowel: Normal caliber .
Peritoneum: No ascites or free air; no fluid collection .
Vessels: Atherosclerotic changes .
Bones: Normal .

CT Chest
Lungs and large airways: 3mm[4mm] ; 1mm[3mm] .

Conclusions: Further observation recommended

Sum of previous measurements: 7
Sum of current measurements: 4
Difference (absolute): -3
Difference (relatively): -42.857%
Status: Complete response

Report content:

John Smith

CT Brain
Extra-axial spaces: Normal .
Intracranial hemorrhage: None .
Basal cisterns: Normal .

CT Pelvis
Reproductive organs: No pelvic masses .
Ureters: Normal .
Bladder: Normal .
Bowel: Normal caliber .
Peritoneum: No ascites or free air; no fluid collection .
Vessels: Atherosclerotic changes .
Bones: Normal .

CT Abdomen
Bile ducts: Normal .

CT Chest
Lungs and large airways: 3mm[4mm] ; 1mm[3mm] .

In order to copy the report, it has to be confirmed

Conclusions

RECIST 1.1

Figure 3.5: Compare reports mode presents simultaneously contents of the currently edited report and the old one

part of the program paginates reports. There are 10 reports per page and user can increment index of the page they are currently viewing or decrement it. One useful feature is grouping reports by title. This is useful when user has the convention of naming that report's title is always name of the patient (most of the current users of the program do it in this way) as they can see and count reports for any user organized in groups.

3.6 Compare reports

This feature strongly depends on the mechanisms included in the Report creator. It copies meta-data of an existing report and creates a new report using it. Content of the old report is presented to the radiologist as a reference and it is also copied as a basis of a new report. When the doctor makes some changes to the contents of the report, only the new version is modified. This allows for fast creation of a report that has only small portion of fragments modified when compared to the previous one. Figure 3.5 presents interface that makes it easier for the radiologist to compare two reports.

3.7 Deployment strategy

Figure 3.6 presents the sequence of actions and set of subjects that are required to deploy the application.

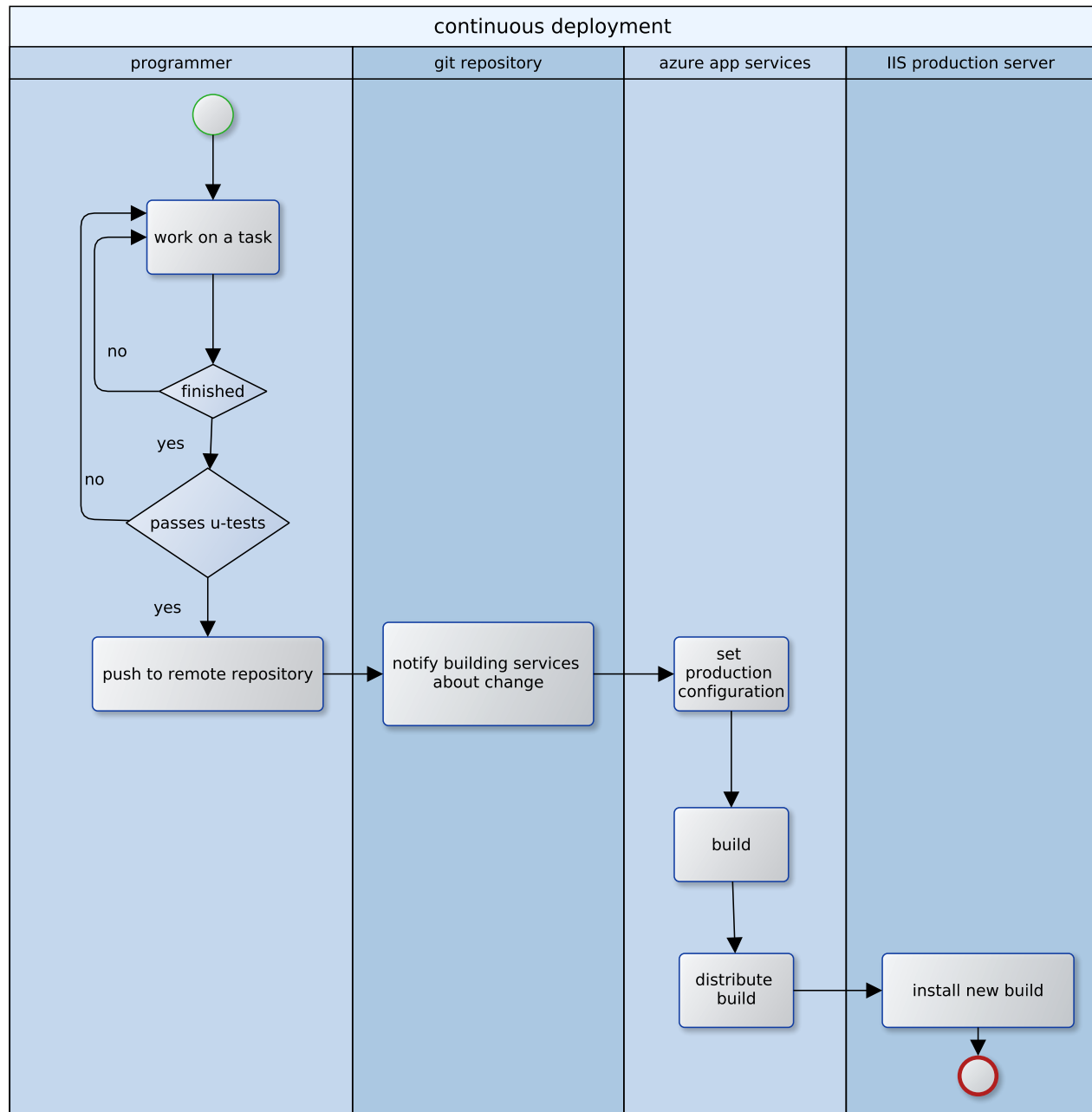


Figure 3.6: Continuous deployment setup makes it faster to distribute new builds to the production servers

Chapter 4

Testing and verification

Most of the testing and verification was performed as user-acceptance testing as the program is heavily focused on interaction with user, however, security-critical parts of the program were unit-tested. The following scenarios were taken into account while testing:

- entity ownership tests
 - user sees only their reports
 - user can edit only their reports
 - while creating a report, user sees public and custom templates names
 - user can download contents of public and custom templates
- database consistency tests
 - deleting a user causes removal of all their reports, templates from the database
 - proper handling of many-to-one relationships between examinations, organs and properties
 - report json representation received from front-end
- report structure tests
 - report tree should be correctly transformed into plain text
 - report tree should be correctly transformed into JSON format
- report editor functionality
 - live preview refreshes on proper stimuli
 - connotation changes in properties should propagate up the tree to organs and examinations
 - customization of template phrases

Chapter 5

Validation of the proposed solution and examples of places where it was used

5.1 Time savings

5.2 User satisfaction surveys

Over the time of first few months after the release of the software to the users, several surveys were conducted. The users were asked open-ended questions such as: "What is the most important feature that you would like to use everyday?", "What is your overall experience you have when you are interacting with the program?", with addition to scale questions like "How likely would you recommend the software to your colleague? 1 – very unlikely, 5 – very likely", "Using the software as my main text-editing tool makes me more productive. 1 – strongly disagree, 5 – strongly agree".

It could be observed that users who initially assigned good marks to user experience, were keener to suggest further improvements. Moreover, quick reactions to the suggestions, tended to result in better marks in the subsequent surveys. On the other hand, users that were not interested in using additional tools to be more productive, were very difficult to encourage to try using the software for a longer period of time. The best results were achieved when a director of radiology observed benefits of using the program and taught their team how to use the system.

5.3 Teleradiologists

Structured reporting system is being used by several independent teleradiologists in Poland as their main text editing tool. They receive reporting requests with images they are asked to describe. They use several RIS systems from different vendors to store reports they generate. Most of them simply copy and paste reports from the system to the RIS systems they use. As majority of the teleradiologists who have used this software are paid on the per-report basis and the average time to create a report using this program is on average three times shorter, most of the users observed significant increases in their salary.

5.4 Hospital

Proposed system was used in hospital in Wieliszew by a group of radiologists who specialize in oncological reporting. They provided precious feedback and many of their suggestions were included in the system.

5.5 Large network of clinics

Structured reporting system is used by a large network of private clinics in Łódź. Integration with their RIS system is being developed, so the systems will cooperate with existing systems to optimize the work-flow of radiologists.

Chapter 6

Conclusion and ideas for the future

6.0.1 Realization of the thesis objectives

As follows from the previous chapters, the main goals of this work have been achieved. Ideas implemented in this structured reporting system are very simple from the theoretical point of view. In fact, template-based text generation existed for years and can be found in most of the programs developed hitherto. However, even such simple methods, supported with well-thought user experience design can give satisfying results. Although the software is not designed for wide spectrum of types of users, several dozens of radiologists from different working environments have used the software as their main tool to create radiological reports. The overall satisfaction

6.0.2 Possible further extensions

In the future, one could think about a way to generalize the reporting ontology or implement a mechanism to create custom ontologies to provide tools that would allow to encode a hierarchy of an arbitrary depth. After obtaining a well-formed tree representing relations between causes and effects, one could experiment with applying recurrent neural networks (RNNs) to automatically generate report's text from structured data source [19]. Another direction for improvement would be a natural language processing mechanism that would learn typical co-occurrences of report items and would ask the radiologist whether an unusual report content is caused by a mistake made while creating the report or a truly unusual pathology.

Bibliography

- [1] <https://www.bls.gov/ooh/healthcare/physicians-and-surgeons.htm>, [08.10.2017]
- [2] Artificial Intelligence: Threat or Boon to Radiologists?, Recht, Michael et al, Journal of the American College of Radiology, Volume 14 , Issue 11 , 1476 - 1480.
- [3] T. Benson, Principles of health interoperability HL7 and SNOMED, Springer-Verlag, London, 2010.
- [4] DICOM Structured Reporting, D. A. Clunie, 1st Edition, 2000, .
- [5] <http://www.hl7.org/Special/committees/structure/index.cfm>
- [6] <https://www.asrt.org/main/careers/careers-in-radiologic-technology/who-are-radiologic-technologists>
- [7] <https://radiologystories.com/2013/10/31/the-typical-radiologist-work-day/>
- [8] <http://www.osirix-viewer.com/osirix/overview/>
- [9] S. Langer, Impact of Speech Recognition on Radiologist Productivity.
- [10] J. du Toit, R. Hattingh and R. Pitcher, The accuracy of radiology speech recognition reports in a multilingual South African teaching hospital.
- [11] <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/lexical-structure>
- [12] <http://emedicine.medscape.com/article/1163977-overview>
- [13] S. Hanenberg, et. al., An empirical study on the impact of static typing on software maintainability.
- [14] <https://help.github.com/articles/fork-a-repo/>
- [15] https://en.wikipedia.org/wiki/Response_Evaluation_Criteria_in_Solid_Tumors
- [16] <https://www.ncbi.nlm.nih.gov/pubmed/3497558>
- [17] <https://www.sitepoint.com/understanding-angulars-apply-digest/>

- [18] <https://blogs.msdn.microsoft.com/jocapc/2015/05/16/json-support-in-sql-server-2016/>
- [19] <http://www2.fiit.stuba.sk/~kvasnicka/NeuralNetworks/>

Warszawa, dnia

Oświadczenie

Oświadczam, że pracę inżynierską pod tytułem: „Structured radiological reporting system”, której promotorem jest prof. dr hab. Jan Mulawka, wykonałem samodzielnie, co poświadczam własnoręcznym podpisem.

.....