DOCUMENTATION

Anbindung einer alternativen Quelldatenbank für Krankheits-Entitäten für die Biomarker-Datenbank BIONDA

Projektdokumentation für das
Studienprojekt im Studienfach
M.Sc. Angewandte Informatik

Prüfer:
Dr. Michael Turewicz

Saif Al-Dilaimi (108014211768)
Arlind Avdullahu (108014222746)
Dejan Babic (108014235782)
RUHR-UNIVERSITÄT BOCHUM
Institut für Neuroinformatik
Ruhr-Universität Bochum
Universitätsstraße 150
44801 Bochum

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AIDS	Acquired Immune Deficiency Syndrome
API	Application Programming Interface
BIONDA	Biomarker and biomarker candidate database
DNA	Deoxyribonucleic acid
EMBL-EBI	European Bioinformatics Institute
GARD	Genetic and Rare Diseases Information Center
HIV	Human Immunodeficiency Virus
JRE	Java Runtime Environment
PIR	Protein Information Resource
PMID	PubMed identification number
RegExp	Regular Expression
SIB	Swiss Institute of Bioinformatics
MeSH	Medical Subject Headings
miRNA	micro Ribonucleic Acid
OBO	Open Biological and Biomedical Ontology
OMIM	Online Mendelian Inheritance in Man
UniProt	The Universal Protein Resource

1. Introduction

In bioinformatics, especially in bioinformatics for proteomics, databases are becoming increasingly important. One research topic of Medical Bioinformatics of the Ruhr-University Bochum, located at Medizinisches Proteom-Center, is the investigation of relationships between biomarkers and diseases. In other words, which biomarkers are related to a specific disease. To answer this question the database Biomarker and biomarker candidate database (BIONDA) was developed. Using this application, it is possible to search for disease-biomarker-relations in scientific publications. The target group of BIONDA are not only scientists worldwide but also the pharmaceutical industry and affected persons worldwide (Frericks-Zipper, 2019). Unfortunately, the problem is that the current source database for disease entities, namely The Universal Protein Resource (UniProt), does not contain enough disease terms. Furthermore, another problem is that there are too few synonyms of a disease. Beside diseases as an entity, the database also contains genes, micro Ribonucleic Acid (miRNA) and proteins as biomarker entities. Therefore, this work deals with the integration of an independent alternative disease database into the existing structure of BIONDA. In a first step, suitable disease databases needs to be identified, then compared under defined criteria and a suitable database needs to be selected. In a second step, the new source database will be integrated into BIONDA, which includes, among other things, adapting the tables of the database. These two steps are explained in more detail in chapter 2. At first, basics are presented, which are necessary for further understanding of this work. First of all, the term *disease* will be explained. Since there is no unique definition, this term is described by disease subcategories. As mentioned, the goal of the research is the investigation of relationships between diseases and so-called *biomarkers*. Thus, section 1.2 defines the term biomarker. Finally, an overview of the application BIONDA will be drawn. The following chapter will discuss the available disease databases and the process of defining criteria, which will make sure a suitable replacement for the current disease database. In chapter 2 all technical aspects will be explained in detail and the necessary steps to build the new database will be discussed. Finally, the results will be presented and a discussion to this work will be given.

1.1. Diseases

When searching the term *disease*, it quickly becomes clear that this is not easy to define. The term is used synonymously with other terms, which makes it difficult to formalize a definition. If one visits the website of Human Disease Ontology (disease ontology.org, 2019) it becomes clear that a disease can be categorized into multiple

subcategories. Thus, this section describes the subcategories of the highest level of disease classification. Syndromes or physical disorders are defined as diseases, too. Instead of a general definition, parts of the subcategories of the superordinate category disease are presented. Figure 1 provides an overview of the subcategories. Disease by infectious agents is a group of diseases "that is caused by the invasion of a host by agents whose activities harm the hosts tissues (that is, they cause *disease*) and can be transmitted to other individuals (that is, they are *infectious*)" (NIH, 2007). The agents can be categorized into five types, such as: viruses, bacteria, fungi, protozoa and helminths (Charles A Janeway et al., 2001). The most known diseases of this subcategory include Human Immunodeficiency Virus (HIV) and Acquired Immune Deficiency Syndrome (AIDS).

Continuing with the next category, disease of anatomical entity, Parkinson is one of the most researched disease in this category. Parkinson is described as "a chronic progressive neurological disease chiefly of later life that is linked to decreased dopamine production in substantia nigra and is marked especially by tremor of resting muscles, rigidity, slowness of movement, impaired balance" (merriam webster.com, 2020). Thus this category of diseases refers to ones that develop over time.

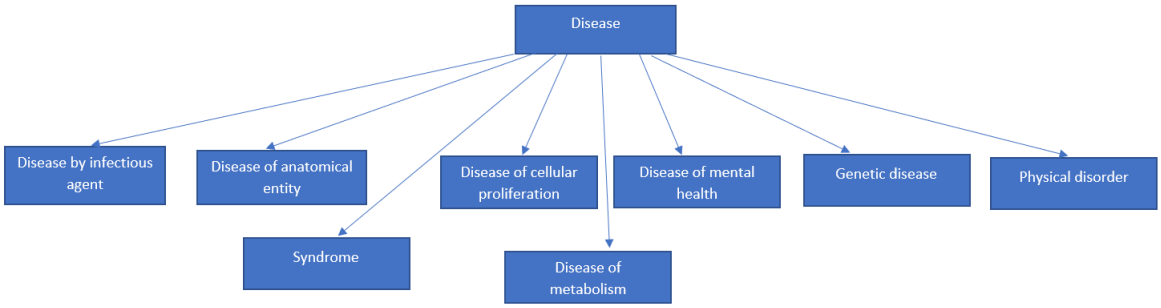


Figure 1.: General classification of diseases according to the Disease Ontology
Source: (disease ontology.org, 2019)

Cancer is one of the most known diseases, which falls into the category disease of cellular proliferation. Especially it is a disease "in which abnormal cells divide without control and can invade nearby tissue. Cancer cells can also spread to other parts of the body" (cancer.gov, 2011). Hereinafter, the subcategory describes diseases in which cells proliferate abnormally. On one hand, mental illnesses "are conditions that affect your thinking, feeling, mood, and behavior" (medlineplus.gov, 2020), which include diseases such as dementia or depression and on the other hand, down syndrome is an example of a genetic disease. A genetic disease "is a disease caused in

whole or in part by a change in the Deoxyribonucleic acid (DNA) sequence away from the normal sequence" and "can be caused by a mutation in one gene mutations, in multiple gene combination of gene mutations or damage to chromosomes" (Genome.gov, 2020).

The cleft palate-lateral synechia syndrome is a physical disorder and "is a congential malformation syndrome characterized by the association of cleft palate and intra-oral lateral synechiae connecting the free borders of the palate and the floor of the mouth" (orpha.net, 2020).

Understanding the multiple categories of a disease one can now break it further down to elements, which may be already in the human system or emerge due a disease. This quantity will be discussed in detail in the next section.

1.2. Biomarkers

Biomarkers are processes, structures or any substances that can be measured in any living body. Especially, elements which influence or predict the result of a disease are called biomarkers, too. (inchem.org, 2019). Biomarkers are used, among other purposes, to provide information on the health status of a patient or to determine the progress of a disease (Atkinson et al., 2001; Mayeux, 2004). Biomarkers can be used, for instance, to determine the stage of a disease. This is done, for example, by measuring specific antigen in the blood concentration. To determine the progress of a disease, for example prostate cancer, a specific antigen (antigen-125) (Atkinson et al., 2001) is examined to determine the growth of a tumor.

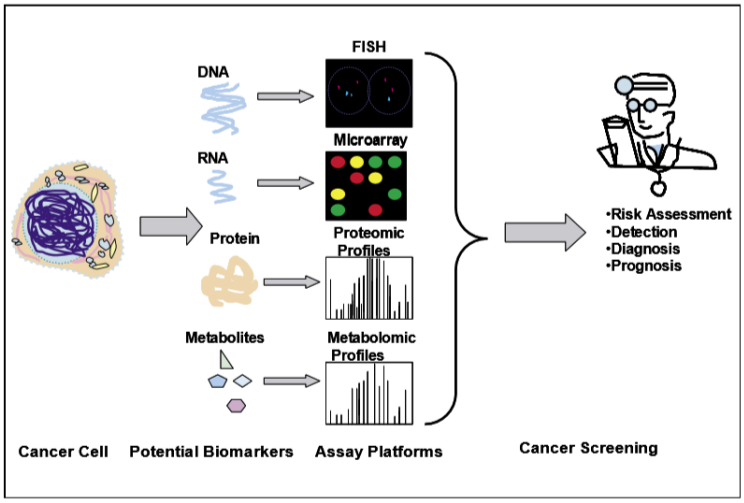


Figure 2.: Relationship between diseases and biomarkers
Source: (Maruvada et al., 2005)

Figure 2 further illustrates the relationship between biomarkers and diseases. There

are different types of biomarkers. Prognostic biomarkers are used to calculate the probability of a reoccurring disease. They are used also to measure the progress of a specific medical condition, which can be of specific value to the scientists. An example of such a biomarker is the Gleason-Score (Group, 2016c).

Predictive biomarkers are to identify which and how individuals could react to an exposure of a medical product or a specific agent (Group, 2016b). The mutations of cystic fibrosis transmembrane conductance regulator are an example of such a biomarker (Group, 2016b). This can also be said to pharmacodynamic biomarkers which are used to show that an individual has positive or negative reaction to a medical product or an environmental agent. Hemoglobin A1c is an example of such a biomarker (Group, 2016a). The last type, namely surrogate endpoints, is "an indicator or sign used in place of another to tell if a treatment works." (cancer.gov, 2011). Blood pressure is an example of a surrogate endpoint (eupati.eu, 2015). Since BIONDA deals with the relationship between biomarkers and diseases it is crucial to understand, which role biomarkers have.

In this case one can note that biomarkers can be crucial for the diagnosis of a disease. Analyzing for example the blood or urine of a patient the resulting biomarkers are used as biological factors, which for example represent the stage of disorder. These biomarkers can also be used to warn patients of possible diseases in advance (Mayeux, 2004).

1.3. BIONDA

The database BIONDA, which is accessible via a web application, was developed by the Medical Bioinformatics Department of the Ruhr-University Bochum. The motivation of it was to investigate the relationships between diseases and biomarkers. The target group are scientists worldwide, who can use this database for their research. Further target groups should not only be the pharmaceutical industry but also affected persons. Figure 3 shows the general architecture of BIONDA.

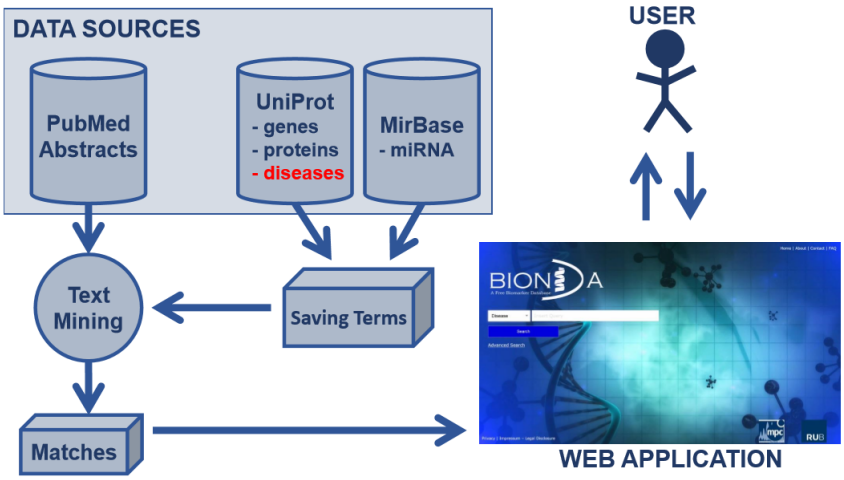


Figure 3.: Overview of BIONDA
Source: (Frericks-Zipper, 2019)

BIONDA allows to search for relations between specific diseases and biomarkers in scientific publications. This is done by submitting a disease or biomarker name as a search term in the BIONDA web application. This is achieved by parsing abstracts of publications and perform text mining to match papers with specific search terms. The search can also be restricted to a sentence-wise or abstract-wise biomarker match. Depending on the search category, the search returns the disease metadata, but the relationship between the metadata is interpreted differently. The search is by default using the abstract-wise option, which makes sure that the desired biomarker and the associated disease were mentioned in the abstracts of the publications. When searching with abstract-wise, the evidence is not directly displayed, because the abstracts are indexed via the PubMed identification number (PMID). However, using the sentence-wise option BIONDA makes sure that the term is mentioned in the same sentence of the abstract. This allows the user a more refined search. However, both option make it possible to draw relations between biomarkers and diseases. One of the most important metadata in BIONDA is the co-occurence-based p-value (Wasserstein and Lazar, 2016). This resembles a score to the user, which is calculated using the χ^2 test (Pearson, 1900) and is based on the co-occurence of markers and diseases. This test requires the true positive, true negative, false positive, and false negative of a specific biomarker and disease pair (bionda.mpc.rub.de, 2019).

- True positives are defined as the number of a matched pair of biomarker X and a disease Y in the same abstract or sentence of an abstract,
- False negatives are defined as the number of search hits from a biomarker X but without disease Y, in the same abstract or sentence of an abstract,
- False positives are defined as the number of hits from a disease Y but without the biomarker X, in the same abstract or sentence of an abstract,

- True negatives are defined as the number of search hits from all other pairs (bionda.mpc.rub.de, 2019).

In general the more matches are achieved of a biomarker and disease pair, the better the p-value and thus the score.

The current database which is used by BIONDA is UniProt. UniProt is a data resource for protein sequences, whose data is freely accessible. It gets updated monthly and is provided by the European Bioinformatics Institute (EMBL-EBI), the Swiss Institute of Bioinformatics (SIB) and the Protein Information Resource (PIR). Over 100 people work on different aspects of this specific database. Originally, each institute managed its own database. Yet, in 2002 they decided to pool them together, which made UniProt become a collection of them all (uniprot.org, 2020). The main aspect which is of importance for this project are the 5,466 disease entries in the UniProt database. This total number of disease entries need to be increased by the new database.

1.4. Aim of this Project

In the first part of this project, the goal is to determine a suitable database for BIONDA, which contains a large number of diseases. This new disease database should make it possible to search for a larger amount of diseases within BIONDA. BIONDA already uses a database but the total number of disease entries needs to be increased. Thus, the main target of this project is to find a larger disease database than UniProt, which is the currently used disease resource (Frericks-Zipper, 2019). In order to find a suitable database, a rough overview of existing databases will be made first, in that way it is possible to choose a suitable option. After that the integration process needs to be explained in detail. The following chapters will discuss which database are available and the requirements of these.

2. Methods

The target of this work is to implement an alternative solution to the existing source database of BIONDA. However, before one can choose an alternative database or a development environment it is crucial to get a deeper understanding of the available data structure of Disease Ontology. Disease Ontology, as outlined in chapter 3, is a well maintained and frequently updated disease database. This is a necessary point to mention as the authenticity of such sensitive information requires monitoring by a public institution. All of the requirements (see table 4) are fulfilled, which allows one to take a deeper look into the available data structure. Disease Ontology references a public repository at Github. This specific repository is monthly updated, which can be regarded as quite reasonable, taking into consideration that the possibility of new diseases being discovered is rather low. Each monthly update consists of an updated file (HumanDO.obo) containing all diseases and each previous versions of this file. As mentioned before it is relevant to analyze the structure of this file to find a repeating pattern. This is especially important to extract just the diseases and ignore everything else. This process will be further explained in detail in the next sections.

2.1. Requirements of a Disease Database

In order to determine which of the mentioned databases is best to be integrated into BIONDA, specific criteria need to be defined. Later, these criteria will be weighted according to the importance to the project lead. Beginning with the first criteria it is necessary to increase the search results while searching with BIONDA. To find a specific disease it is significant to have trivial names of a specific biomarker or disease. This could solve the problem of a disease being named differently by other groups of people. A medicine researcher would name a disease by its medical name which would restrict the usability of the database to other fields. Anyhow, BIONDA should be usable for every person in general, which makes the use of trivial names the number one criteria. Furthermore the database should be updated frequently. This would ensure that the latest research of publications would be immediately displayed on BIONDA, as the selected database is always up to date and thus new diseases are frequently added to the database. This would ensure that BIONDA could become a more easy to use disease database for many people. In addition, the selected database should contain a large number of entries. To ensure that the database is accessible to everyone it is essential for it to be freely accessible. It is crucial to have an Application Programming Interface (API) or a database dump in any format which allows the development team to download and parse the disease entities. In summary the criteria for the UniProt replacement are the following ones.

- Trivial names available
- Update frequency guaranteed
- API available
- High number of entries
- Freely accessible
- From public organization

In order to create an evaluation matrix the established criteria need to be weighted and ranked. The most important criterion will be free and open accessibility. In the best case, the database should be accessible to everyone. Closely followed by the use of trivial names. This is especially important to non-scientists, as the medical terms of the diseases are often only used in research fields. Furthermore, the number of entries in the database as well as the update frequency are also very important. A high number of disease terms is essential to make an accurate search possible. The update frequency is as important as the number of entries. A database that receives frequent updates and always stays up-to-date can provide an overview of many new publications through the many entries and thus shed light on areas of medicine that have already been researched intensely. Finally, the last criterion is a working or existing API. A working API, is essential to automatically download and parse the database. However, this criterion is the least weighted, since it was clear from a previous search that all databases have a limit and thus the entire database could not be downloaded at once. It is important to note, that the Disease Ontology database does not offer an API but a dump of the complete disease database. The file format of the database is purely text based, which allows to choose a development environment that suits the knowledge of the development team. The weights of the criteria can be seen in the following table.

Criterion	**Weight**
Number of entries	x4
Synonyms	x5
Freely accessible	x6
Public organization	x6
API (unlimited)	x3
Update frequency	x4

Table 1.: Weights of Database Criteria

2.2. Software, Hardware and Programming Language

When developing a software component it is important to decide which programming language should be used. As argued in the requirements specification of this project (see appendix D, figure 6) the decision depends on the current setup of

BIONDA. BIONDA has been developed with typical web based technologies but the backend has been developed in Java (Liguori and Liguori, 2018) with a MySQL (DuBois, 2014) database which holds more than just the diseases (see section 2.4). After discussing this question with the project lead it was clear to use Java for this project. However, this project is not entirely attached to the infrastructure of BIONDA. The hardware used in this project is purely personalized. The developed component runs on a Java Runtime Environment (JRE) which can be used on every personal computer. Therefore no special hardware is required to run the component. The main task of the project is to feed new diseases into the BIONDA database, which makes it independent and not linked to the backend neither frontend.

Organizing this project accordingly to the development software cycle it is essential to split the tasks into smaller ones. Overall one can say that the development part of the project comprises a text parsing and a database modeling component, which are both crucial for the application. In the following sections both components and the used technologies are explained in detail.

2.3. Regular Expression

Beginning with the first component "text parsing" it is significant to get an overview of the source and the format of the file. The file does not follow a widely known format like XML or Markdown. However, it does follow a format specially created by the Open Biological and Biomedical Ontology (OBO) Foundry. The mission of OBO is to develop "a family of interoperable ontologies that are both logically well-formed and scientifically accurate" (Schriml et al., 2019). To achieve this collaborative work, it follows these four principles, such as: open use, collaborative development, common syntax. More importantly, the foundry is overseen by an operation committee and work groups consisting of researchers located worldwide. This is especially significant in order to assure that the content of the delivered data is verified.

```
format-version: 1.2
data-version: doid/releases/2019-05-13/doid-non-classified.obo
date: 13:05:2019 09:41
saved-by: lschriml
auto-generated-by: OBO-Edit 2.3.1
subsetdef: DO_AGR_slim "DO_AGR_slim"
subsetdef: DO_cancer_slim "DO_cancer_slim"
...
default-namespace: disease_ontology
remark: The Disease Ontology content is available via the Creative Commons
Public Domain Dedication CC0 1.0 Universal license
(https://creativecommons.org/publicdomain/zero/1.0/).
ontology: doid/doid-non-classified.obo
property_value: http://purl.org/dc/elements/1.1/title "Human Disease Ontology" xsd:string
```

Figure 4.: First Lines of the Human.obo File
Source: (Schriml et al., 2019)

Now focusing on the content of the delivered file one can observe that the first lines define properties like ontology type, creation date, created by and other information which are quite irrelevant for the project. Nonetheless, after those first introductory lines the main content of the file begins, the diseases. These are defined inside a specific sequence of characters named "[Term]". This sequence precedes the properties of the disease. However, one can see the possible properties of each disease. It shall also be noted that some properties are multi-value properties, meaning a disease may contain zero to many values of these very properties.

Property	Type	Example
ID	single value	id: DOID:0001816
Name	single value	name: angiosarcoma
Alternative DOID	multi value	alt_id: DOID:267
		alt_id: DOID:4508
Definition	single value	def: "A vascular cancer that derives_from the cells that line the walls of blood vessels or lymphatic vessels." [url:http://en.wikipedia.org/wiki/Hemangiosarcoma, ...]
Belongs to subsets	multi value	subset: DO_AGR_slim
		subset: NCIthesaurus
Synonyms	multi value	synonym: "acetylsalicylic acid allergy" EXACT []
		synonym: "ASA allergy" EXACT []
Ontology references	multi value	xref: ICD10CM:E88.9
		xref: ICD9CM:277.9
Parent diseases	multi value	is_a: DOID:712 ! refractory hematologic cancer
		is_a: DOID:9538 ! multiple myeloma
Created by	single value	created_by: snadendla
Creation date	single value	creation_date: 2011-06-15T02:48:20Z
Obsolete	single value	is_obsolete: true

Table 2.: Overview of Disease Properties
Source: (Schriml et al., 2019)

Some diseases may also miss some of these properties. To get a better overview of the raw input that needs to be processed and extracted see appendix I. These disease properties are mostly self-explanatory but some of them need to be explained in detail. The alternative disease ontology ids reference ids of diseases that may link to a similar one. However, some of them may be obsolete and do not reference any diseases. A crucial property is e.g. the synonyms. Those are one of the key features that led to the decision to pick disease ontology as a source database. Because of that the users of BIONDA get a higher probability to find a disease with a search term that may not be a professional research term. Finally, the parent diseases are essential for building a connection between all diseases. Those connections allow us to build a graph[1] which meets the requirement to build a hierarchical structure of the diseases.

Knowing what type of data is delivered and what needs to be extracted one can decide which technology is right to extract the data. Of course there are a lot of

[1]In graph theory, a graph is a structure consisting of a set of objects in which some pairs of the objects are in some sense related. The objects are mostly called vertices and each connection of vertices is called an edge (Trudeau, 1993).

possibilities but when working with data extraction or any kind of pattern matching[2] the most used technology is called Regular Expression (RegExp). RegExp is a subfield of theoretical informatics and describes a specific pattern by means of a string. This string follows a specific syntax and rules to build a matching algorithm. The most common use of RegExp is a functionality in text editors. The search and replace functionality in each text editor uses RegExp to find a specific pattern. In detail RegExp consists of constants, resembling sets of strings and operator symbols, which define operations over these sets. The following definition of RegExp can be found in many literature about formal language theory (Hopcroft et al., 2007).

Given a finite alphabet Σ the following constants are defined as a regular expressions.

- $\varnothing$ is an empty set
- ϵ is an empty string
- A in Σ defines a set containing only the character A

Now, given the regular expression B and C it is possible to produce different matching patterns by using the following operations.

- Concatenating: expression BC takes every string in B and C and produces a matching pattern matching every combination of the elements in B and C.
- Alternation: expression $B \mid C$ matches any element in B or C but ignores elements that are already in B.
- Kleene star: expression B^* matches the smallest superset[3] of B, which contains an empty string, too.

However, those basics are only a tiny portion of the functionality of RegExp. The full explanation would exceed the scope of this work. For this very reason it is important to focus on the regular expression that extracts the diseases from the delivered file.

```
\Q[Term]\E((?:(?!\Q[Term]\E|\Q[Typedef]\E)[\s\S])*)
```

Figure 5.: Regular Expression for Diseases

In principle, this regular expression searches for every string containing the tokens *[Term]* until another sequence of these characters is found. In this case, every string between those sequences will be matched and extracted. This assures that some tokens, defining the properties and begin with *[Typedef]*, are not matched (see appendix J). This is secured by creating groups inside the expression, which have to fulfill both

[2] Pattern matching is the process of checking if a given sequence of characters exist. However, the match usually has to be exact: "either it will or will not be a match." (Hopcroft et al., 2007)

[3] In mathematics, a set A is a subset of a set B, or equivalently B is a superset of A, if A is contained in B. (Hopcroft et al., 2007)

criteria. In detail the expression applies this matching algorithm, which is a naive representation of RegExp.

Algorithm 1 Matching Algorithm

Require: $string\_of\_diseases := S$

```
Initialize matched_groups
for each line ∈ S do
    Initialize group
    if [Term] = line then
        if [Term] ≠ line + 1 then
            group ← line + 1
    if line + 1 empty then
        if [Term] = line + 2 then
            matched_groups ← group
return matched_groups
```

At this point all diseases are extracted and ready to be inserted into the database. In the following section the disease insertion will be discussed in detail.

2.4. Database Implementation

When building an application it is significant to choose a database architecture where all data is saved. However, BIONDA already has an existing infrastructure (see figure 6) and a database where all relevant data is stored. This makes the decision, which database architecture to choose, obsolete. This work focuses on the modification of the current database scheme, which is built on top of a MySQL server. In appendix F the entity relationship model describes which new tables and relationships are added to the database. Especially the modifications on the existing disease table are displayed.

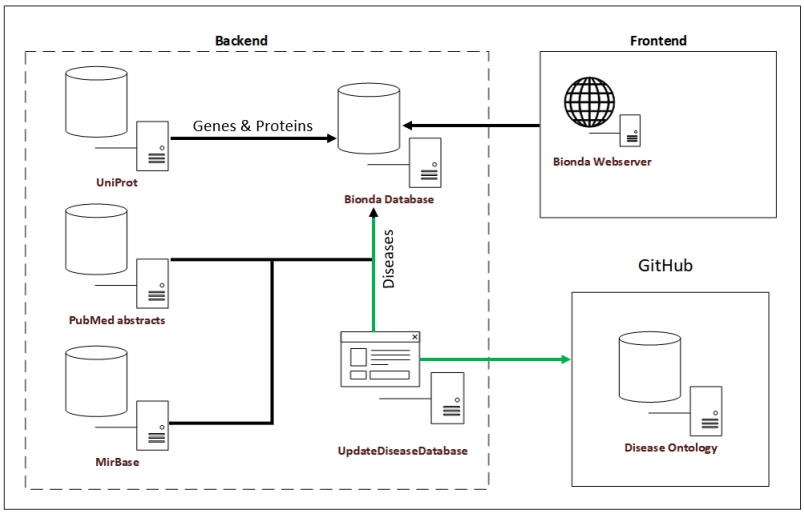


Figure 6.: BIONDA Infrastructure

When inserting the diseases into the database it is of great importance to consider the following two requirements: Building relationships between the diseases and its multi-valued properties and creating the hierarchical data structure of the parent-child-relationships of these diseases. The relationships are crucial to guarantee the consistency of the database. In order to achieve this, the concept of foreign keys is applied on all newly-added and modified tables of the database. This makes sure that all database records are accordingly updated, in case of a modification or deletion. In order to solve the secondly-mentioned case, a typical relational database is not the best choice while working with graph-based data. Instead, the use of a graph-based database would be a better choice in this case. However, as explained in chapter 3 the overall amount of diseases is 11,684 with 9,357 connections (edges) between those diseases. This number is more than feasible with a relational database. In order to build a graph from these connections, indexes and optimized queries make it possible to get a result in milliseconds.

This update mechanism is directly tied to the database. The whole workflow is developed in Java and a Java-MySQL-Connector to establish, add and update records of BIONDAs database. In this very case a Java class has been implemented to offer all methods that are needed to handle the required operations (see appendix H).

2.5. Maintenance

Now that there is an understanding of both components, it is of special interest to outline the maintenance of this application. Maintaining an application can be solved in several ways. Yet in the case of this specific one, it will consist of logging error messages, handling the case if the update process has been interrupted, modifying the applications behavior without source code changes and the implementation of additional methods, which output statistics of the parsed data of the disease ontology. All of this would happen during the update process of the database. Logging is an essential part of an application, especially with applications that are mostly working as a background process. As this is the case, logging is responsible for writing any errors or debug messages to a file on the current working directory. This includes any connection errors to the database or insertion of diseases. Downloading the disease ontology archive and possible permission errors are also written to the file. Each error is documented and corresponds to an error code, which can be referred to by the attached list of errors in appendix K.

Another case that can occur is an unexpected interruption of the update process. This could have several reasons. The best way to avoid any of these possible error states is to firstly, repeat the download process of the disease ontology archive if it gets

interrupted. This may increase the overall runtime by several minutes due to the size of the archive. However, because time is not a constraint, it is reasonable to take this overhead if required. Secondly, the insertion of diseases into the databases can get interrupted, too. This could as well have many reasons, starting from a simple insertion error to a mysql timeout. A timeout could occur due to the fact that the insertion of nearly 12,000 diseases may take a while. Depending on the configuration of the mysql server a timeout may occur. To avoid this issue, the application fetches a list of pre-available diseases from the database. This list is then used to compare which diseases are missing and which need to be inserted. This routine solves any case of missing diseases in the BIONDA database, even if all are missing.

Another significant feature of an application is to allow the user to modify the behavior of the application without any source code changes. This is accomplished by adding a configuration file, which contains all properties that influence the application. The properties change the behavior of all major tasks of the application. As an example, it is possible to change the update interval to a more suitable value. This increases the possibilities of this application. In the following table all possible properties are listed.

Property	**Value Type**	**Example**
db.user.name	String	db.user.name=root
db.user.password	String	db.user.password=xxxx
db.url	String	db.url=jdbc:mysql://localhost:3306/bionda? autoReconnect=true &useSSL=false &allowPublicKeyRetrieval=true
repository.download.buffer_size	Integer	repository.download.buffer_size=8192
repository.download.timeout	Integer	repository.download.timeout=1500000
repository.ontology.file_name	String	repository.ontology.file_name=HumanDO.obo
repository.save_path	String	repository.save_path=user.home
repository.url	String	repository.url=https://github.com/Disease Ontology/HumanDiseaseOntology/archive/master.zip
disease.update.interval	Integer	disease.update.interval=30

Table 3.: Configuration Properties

Lastly, a few methods are implemented as part of the utility class (see appendix H), offering a look inside the data. Especially, the diseases are further inspected in order to have a look at how many diseases have parents, which diseases have synonyms and how many ontology references they contain. It is also important to note that a function is implemented to get all obsolete diseases, which will be important for data science[4] purposes. In chapter 3 the mentioned statistics will be further explained in detail.

[4]"Data science is a multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data."(Dhar, 2013)

3. Results

The main aim of this work was to increase the amount of disease entities by replacing the UniProt database. To find a suitable database, a research must be carried out to find databases that meet the requirements. As already mentioned BIONDA is using the database UniProt, which does not contain a large amount of disease entries. In addition it does not have any or very few synonyms, which is a problem as they are not linked to each other and therefore it is harder for scientists to recognize relations between diseases. The alternative database needs to fulfill those two main features. This research process has identified a few candidates that fulfill these requirements. However, all candidates and other requirements will be presented in the following section.

3.1. Overview of Available Disease Databases

The first candidate is Disease Ontology. This database contains 11,924 disease entries, which is more than double the number of UniProt. It is provided by the Institute of Genome Sciences University of Maryland School of Medicine and is updated frequently (disease ontology.org, 2019). Similar, to the UniProt database it is also freely accessible. A second candidate is the Medical Subject Headings (MeSH) database, which is also freely accessible. The exact number of disease entries is unfortunately not apparent, since different fields of medicine are integrated in this database. These include types of injuries, proteins and also other areas that are irrelevant for this project. This makes it necessary to extract the diseases from any other field, which complicates the automation of this process. Identical to the Disease Ontology, this database is provided by a public organization, too, namely the U.S. Department of Health & Human Services (ncbi.nlm.nih.gov/mesh, 2020).

The database Malacards has about 22,000 disease entries, which has the most entries of all considered database. It is also provided by a public organization, the Weizmann Institute of Science. However, it is not freely accessible, which is essential for the integration into BIONDA (malacards.org, 2020). One other database is Open Targets. This database is freely accessible as well and offers 12,422 disease entries, which is again more than twice of UniProt. Open Targets is provided by GSK, EMBL-EBI, Sanger, Snnofi, Biogen, Takeda and Celgene (opentargets.org, 2020). All of these companies are big players in the biomedical scene. Based on these properties the following databases will be taken into consideration.

- UniProt
- Open Targets

- MeSH
- Disease Ontology

All of these databases could be a suitable replacement of UniProt. The main target of getting more disease entries will be fulfilled with each of these.

3.2. Comparison of Available Disease Databases

Based on these criteria and their weights these databases can now be evaluated. It is immediately noticeable that Malacards is not freely accessible and therefore can be excluded from the comparison of the given databases. The other databases fulfill most of the established criteria and can now be tested.

Criteria	**Weight**	**Databases**			
		Open Targets	**Disease Ontology**	**MeSH**	**UniProt**
Number of entries	x4	8	10	4	5
Synonyms	x5	2	10	4	8
Freely accessible	x6	10	10	10	10
Public organization	x6	10	10	10	10
API (unlimited)	x3	5	0	3	5
Update frequency	x4	6	10	9	6
Points scored		**201/270**	**240/270**	**201/270**	**219/270**

Table 4.: Evaluation Matrix of Compared Databases

However, it is important to check raw data of these databases in order to test how significant the properties are. After a check of the given data it is clear that Disease Ontology (see appendix J) gives the best result. The Disease Ontology is a standardized community driven database for biomedical data dealing with human diseases. Its aim is to provide consistent, reusable and sustainable descriptions of human disease terms, phenotype characteristics and related medical vocabulary disease concepts for clinical and medical research (Schriml et al., 2019). Disease Ontology is updated frequently through cross-referencing of terms to other databases like MeSH, Online Mendelian Inheritance in Man (OMIM), Genetic and Rare Diseases Information Center (GARD), Orphanet and a lot of other vocabularies. This cross-referencing allows frequent updates of diseases and other medical terms. However, all releases can be found on GitHub, too. For each disease the database provides an DOID, a Disease Ontology unique ID, a disease term name, whether or not the disease is obsolete and a definition. Further it provides Xrefs which are cross-references from other clinical vocabularies, alternative ids which are merged DOIDs, synonyms of each disease and a parent-child-relationship, which connects diseases with each other. The amount of diseases is more than twice of the UniProt database. Furthermore, the relationships between diseases are given in Disease

Ontology, which would help to visualize their relations, e.g. as a tree. All of these attributes are very important for the database later on, i.e. could improve and expand the functions of BIONDA in a future release.

3.3. Comparison UniProt vs. Disease Ontology

However, one needs to discuss how effective the replacement is. As a matter of fact due to the new source database the available disease entities have increased by 113.73% from 5,466 to 11,683. As already mentioned the fact that Disease Ontology contains synonyms was one of the key features to consider it as a suitable disease database. At the beginning of this project the UniProt database did not contain any synonyms, making the decision to choose Disease Ontology unworthy to discuss. However, during the staging phase of this project UniProt released a new version of its human disease database. This fact makes it necessary to compare random diseases and their properties from both databases with each other. Especially one needs to compare how naively their synonyms are. This is truly significant as BIONDA, whose target groups are not only scientists but also people without any scientific background. Table 5 compares three diseases each from the UniProt and Disease Ontology database. One can observe that the Disease Ontology database contains more synonyms on each disease compared to the UniProt database. They also seem to sound more intuitively, which makes it possible for non scientists to get search results. An important point to mention is that the raw data of the database contains multiple disease entries of the same disease. At first, those may seem as duplicates. However, on second sight, they resemble the same disease, yet in different stages. A good example for this case is Alzheimer, which is extremely dependent from which gene or chromosome it originates.

Property/Disease Database	Disease 1	Disease 2	Disease 3
UniProt			
Name	breast cancer	Alzheimer	malaria
Number of Synonyms	4	1	-
Synonyms	"Breast cancer familial", "Breast cancer familial male", "Breast carcinoma", "Mammary carcinoma"	"Presenile and senile dementia"	-
Disease Ontology			
Name	breast cancer	Alzheimer	malaria
Number of Synonyms	6	2	1
Synonyms	"breast tumor", "malignant neoplasm of breast", "malignant tumor of the breast", "mammary cancer", "mammary neoplasm", "mammary tumor", "primary breast cancer"	"Alzheimer disease", "Alzheimers dementia"	"induced malaria"

Table 5.: Comparision of synonyms for three diseases between UniProt and Disease Ontology
Source: (Schriml et al., 2019; The-UniProt-Consortium, 2018)

However, these results are good indications that the decision to choose Disease Ontology was the right one. Nonetheless, it is still necessary to verify the success rate of this new data. To achieve this one needs to validate the new diseases with the help of the text mining component of BIONDA. This component is part of the infrastructure of BIONDA and cannot be accessed by the project team. Nonetheless, the project lead decided to take the extracted disease data and allow the text mining component to process them. The process will consider only abstracts of publications which were published in 2019. Including earlier years would increase the overall runtime without any statistical benefit. The returned results from the project lead did not include any disease synonyms. Even when excluded, the process did find 7,751 disease-biomarker pairs in abstracts, which is a 102.80% increase. Another factor which gets evaluated is the unique number of diseases found through the process. UniProt did find 5,986, which is not comparable to the 14,159 diseases found by Disease Ontology as a source database.

Overall, one can say that the replacement of the database is beneficial in all cases. Including all previous years will increase the search results even more. In the next chapter these results will be discussed.

4. Discussion

The aims of this project were to compare the available source databases for diseases and to implement a mechanism which adds diseases to the existing BIONDA database. The first part of this work compared several databases based on specific requirements, which are of specific value to the BIONDA project team. The research and evaluation process of these databases was done in close cooperation with the project team. Especially, BIONDAs core values also needed to be represented by this new database. These discussions resulted in choosing the Disease Ontology database, which fulfilled all needed requirements (see figure 4) and matched BIONDAs core values. After this question is answered, the development of the update process was implemented in java as a background process. Many use cases were handled to make the process more robust. An example for that was to save the update progress and continue at the same state in case of an interruption.

This work was based on the premise to increase the amount of results by replacing the source database. This is successfully done by this project and one can observe how much more beneficial this change is, as detailed in chapter 3. The next step for this application will be the integration into the infrastructure of BIONDA. As mentioned, this process will take care of updating the BIONDA database with new diseases published by Disease Ontology. This process can be started as a daily cron job to populate the database. As part of this process the text mining component will parse all abstracts and build all scores for each new disease. This is an important step to determine how good the results over all years are. Now it is also possible to visualize the parent-child-relationships between diseases in BIONDA. Due to the representation of all disease relationships in the new database it is possible to track each relationship to the root or leaf disease. This, however, is not part of this work and will be part of the future development process of BIONDA.

References

[Atkinson et al. 2001] ATKINSON, Arthur ; COLBURN, Wayne ; DEGRUTTOLA, Victor ; DEMETS, David ; DOWNING, Gregory ; HOTH, Daniel ; OATES, John ; PECK, Carl ; SCHOOLEY, Robert ; SPILKER, Bert ; WOODCOCK, Janet ; ZEGER, Scott: Biomarkers and surrogate endpoints: Preferred definitions and conceptual framework*. In: *Clin Pharmacol Ther* 69 (2001), März, S. 89–95

[bionda.mpc.rub.de 2019] BIONDA.MPC.RUB.DE: *BIONDA - a free biomarker database*. `http://bionda.mpc.rub.de`. [Accessed on: 2020-01-09]. Version: 2019

[cancer.gov 2011] CANCER.GOV: *NCI Dictionary of Cancer Terms*. Version: Februar 2011. `https://www.cancer.gov/publications/dictionaries/cancer-terms`. [Accessed on: 2020-01-01]

[Charles A Janeway et al. 2001] CHARLES A JANEWAY, Jr ; TRAVERS, Paul ; WALPORT, Mark ; SHLOMCHIK, Mark J.: Infectious agents and how they cause disease. In: *Immunobiology: The Immune System in Health and Disease. 5th edition* (2001). `https://www.ncbi.nlm.nih.gov/books/NBK27114/`. [Accessed on: 2020-01-01]

[Dhar 2013] DHAR, Vasant: Data Science and Prediction. In: *Commun. ACM* 56 (2013), Dezember, Nr. 12, 64–73. `http://dx.doi.org/10.1145/2500499`. – DOI 10.1145/2500499. – ISSN 0001–0782

[DuBois 2014] DUBOIS, Paul: *MySQL Cookbook - Solutions for Database Developers and Administrators*. Sebastopol : "O'Reilly Media, Inc.", 2014. – ISBN 978—1–4–49–37–4

[eupati.eu 2015] EUPATI.EU: *Surrogate endpoint*. `https://www.eupati.eu/glossary/surrogate-endpoint/`. [Accessed on: 2020-01-29]. Version: 2015

[Frericks-Zipper 2019] FRERICKS-ZIPPER, Anika: *BIONDA: A free database for a fast information on published biomarkers and biomarker candidates*. February 2019

[Genome.gov 2020] GENOME.GOV: *Genetic Disorders*. `https://www.genome.gov/For-Patients-and-Families/Genetic-Disorders`. [Accessed on: 2020-01-01]. Version: 2020

[Group 2016a] GROUP, FDA-NIH Biomarker W.: *Pharmacodynamic/Response Biomarker*. Food and Drug Administration (US), 2016 `https://www.ncbi.nlm.nih.gov/books/NBK402286/`. [Accessed on: 2020-01-29]

[Group 2016b] GROUP, FDA-NIH Biomarker W.: *Predictive Biomarker*. Food and Drug Administration (US), 2016 `https://www.ncbi.nlm.nih.gov/books/NBK402283/`. [Accessed on: 2020-01-29]

[Group 2016c] GROUP, FDA-NIH Biomarker W.: *Prognostic Biomarker*. Food and Drug Administration (US), 2016 `https://www.ncbi.nlm.nih.gov/books/NBK402289/`. [Accessed on: 2020-01-29]

[Hopcroft et al. 2007] HOPCROFT, John E. ; MOTWANI, Rajeev ; ULLMAN, Jeffrey D.:

Introduction to Automata Theory, Languages, and Computation. 3. Aufl. Boston : Pearson/Addison Wesley, 2007. – ISBN 978–0–321–46225–1

[inchem.org 2019] INCHEM.ORG: *Biomarkers In Risk Assessment: Validity And Validation (EHC 222, 2001)*. http://www.inchem.org/documents/ehc/ehc/ehc222.htm. [Accessed on: 2019-12-29]. Version: 2019

[Liguori and Liguori 2018] LIGUORI, Robert ; LIGUORI, Patricia: *Java - kurz und gut*. O'reilly, 2018. – ISBN 9783960090519

[malacards.org 2020] MALACARDS.ORG: *Malacards*. https://www.malacards.org/. [Accessed on: 2020-01-02]. Version: 2020

[Maruvada et al. 2005] MARUVADA, Padma ; WANG, Wendy ; WAGNER, Paul ; SRIVASTAVA, Sudhir: Biomarkers in molecular medicine: cancer detection and diagnosis. In: *BioTechniques* Suppl (2005), Mai, S. 9–15. http://dx.doi.org/10.2144/05384SU04. – DOI 10.2144/05384SU04

[Mayeux 2004] MAYEUX, Richard: Biomarkers: potential uses and limitations. In: *NeuroRx* 1 (2004), Nr. 2, S. 182–188

[medlineplus.gov 2020] MEDLINEPLUS.GOV: *Mental Disorders*. Version: 2020. https://medlineplus.gov/mentaldisorders.html. [Accessed on: 2020-01-01]

[ncbi.nlm.nih.gov/mesh 2020] NCBI.NLM.NIH.GOV/MESH: *MeSH*. https://www.ncbi.nlm.nih.gov/mesh. [Accessed on: 2020-01-02]. Version: 2020

[NIH 2007] NIH: *Understanding Emerging and Re-emerging Infectious Diseases*. National Institutes of Health (US), 2007 https://www.ncbi.nlm.nih.gov/books/NBK20370/. [Accessed on: 2020-01-01]

[disease ontology.org 2019] ONTOLOGY.ORG disease: *Disease Ontology - Institute for Genome Sciences @ University of Maryland*. http://disease-ontology.org/. [Accessed on: 2019-12-30]. Version: 2019

[opentargets.org 2020] OPENTARGETS.ORG: *Open-Targets*. https://www.opentargets.org/. [Accessed on: 2020-01-02]. Version: 2020

[orpha.net 2020] ORPHA.NET: *Orphanet: Cleft palate lateral synechia syndrome*. https://www.orpha.net/consor/cgi-bin/OC_Exp.php?lng=en&Expert=2016. [Accessed on: 2020-01-01]. Version: 2020

[Pearson 1900] PEARSON, Karl: X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50 (1900), Nr. 302, 157-175. http://dx.doi.org/10.1080/14786440009463897. – DOI 10.1080/14786440009463897

[Schriml et al. 2019] SCHRIML, L. M. ; MITRAKA, E. ; MUNRO, J. ; TAUBER, B. ; SCHOR, M. ; NICKLE, L. ; FELIX, V. ; JENG, L. ; BEARER, C. ; LICHENSTEIN, R. ; BISORDI, K. ; CAMPION, N. ; HYMAN, B. ; KURLAND, D. ; OATES, C. P. ; KIBBEY, S. ; SREEKUMAR,

P. ; LE, C. ; GIGLIO, M. ; GREENE, C.: Human Disease Ontology 2018 update: classification, content and workflow expansion. In: *Nucleic Acids Res.* 47 (2019), Jan, Nr. D1, S. D955–D962

[The-UniProt-Consortium 2018] THE-UNIPROT-CONSORTIUM: UniProt: a worldwide hub of protein knowledge. In: *Nucleic Acids Research* 47 (2018), 11, Nr. D1, D506-D515. http://dx.doi.org/10.1093/nar/gky1049. – DOI 10.1093/nar/gky1049. – ISSN 0305–1048

[Trudeau 1993] TRUDEAU, Richard J.: *Introduction to Graph Theory*. 2nd Revised ed. New York : Courier Corporation, 1993. – ISBN 978–0–486–67870–2

[uniprot.org 2020] UNIPROT.ORG: *UniProt*. https://www.uniprot.org/help/about. [Accessed on: 2020-01-02]. Version: 2020

[Wasserstein and Lazar 2016] WASSERSTEIN, Ronald L. ; LAZAR, Nicole A.: The ASA Statement on p-Values: Context, Process, and Purpose. In: *The American Statistician* 70 (2016), Nr. 2, 129-133. http://dx.doi.org/10.1080/00031305.2016.1154108. – DOI 10.1080/00031305.2016.1154108

[merriam webster.com 2020] WEBSTER.COM merriam: *Definition of PARKINSON'S DISEASE*. https://www.merriam-webster.com/dictionary/Parkinson%27s+disease. [Accessed on: 2020-01-01]. Version: 2020

A. Reflexion - Saif Al-Dilaimi

Hat sich die Aufgabenstellung oder Ihr Verständnis der Aufgabenstellung im Laufe der Zeit verändert? Wenn ja, wie?
Das Ergebnis der Anwendung war klar definiert, das Verständnis und das resultierende Vorgehen zur Bewältigung der Aufgabe war nach dem Erstellen des Pflichtenheftes definiert. Zunächst war es notwendig sich in den Format der HumanDO.obo Datei einzuarbeiten. Daraufhin wurden alle Arbeitspakete nach Zeitplan absolviert.

Wie passend war die Planung des Projektverlaufs, insbesondere der Termine? Wie häufig musste die Planung angepasst werden? Warum? Was war Ihre Rolle dabei?
Die Planung und Aufgabenteilung wurde bereits früh im Pflichtenheft festgelegt. Dadurch konnten alle Teammitglieder selbstständig die einzelnen Arbeitspakete durchführen. Mit unserem Betreuer wurden 1-2 mal im Monat Besprechungen eingeplant und aktuelle Themen diskutiert. Als Teamleiter war ich für die Abstimmung verschiedener Aspekte des Projekts zuständig.

Welche Methoden haben Sie zur Unterstützung der gemeinsamen Planung genutzt und wie haben sie sich bewährt?
Zur gemeinsamen Planung haben wir online Medien genutzt. Über Github konnten wir die kollaborative Zusammenarbeit am Quellcode durchführen. Des Weiteren haben wir auch darüber die Reviews der einzelnen Komponenten durchgeführt. Für die bessere Planung wurden unter anderem Arbeitspakete und Meilensteine im Pflichtenheft erstellt.

Wie hat die Koordination bei der Projektarbeit funktioniert? Welche Probleme gab es? Wie wurden sie angegangen? Was war Ihre Rolle dabei?
Die Koordination bei der Projektarbeit ging anhand von wöchentlich gesetzten Zielen und deren Aufteilung problemlos vorwärts. Durch Github und persönlichen Austausch konnten alle Komponenten reibungslos zusammengeführt werden.

In wieweit waren Sie ausreichend kompetent für die Mitarbeit? Welche Kompetenzen mussten Sie sich im Projektverlauf aneignen? Was haben Sie durch das Studienprojekt gelernt?
Aufgrund der Größe des Softwareprojektes war es notwendig, auf strukturierte Planung und Aufgabenteilung innerhalb des Teams zurückzugreifen. Erlernt und vertieft wurde vor allem die Sensibilität und Notwendigkeit von Planung und die

Anwendung von Design-Patterns. Insbesondere wurde hier das theoretische Wissen über das Gebiet der Bioinformatik vertieft.

Was haben Sie unternommen, um möglichst effizient und verlässlich im Projekt mitzuarbeiten?
Für die effiziente und verlässliche Umsetzung wurde auf strukturierte Planung der einzelnen Programmablaufschritte und zielgerichtete Absprachen unter den Projektmitgliedern zurückgegriffen. Es war wichtig offene Fragen so früh wie möglich zu identifizieren und mit den Projektleitern zu besprechen. Die, zu jeder Projektzeit, aktuelle technische Dokumentation ermöglichte es, den Programmcode einfach, erweiterbar und wartbar zu halten.

Was würden Sie anders machen, wenn Sie mit der zum Projektende erworbenen Erfahrung die Aufgabe nochmals bearbeiten müssten?
Da bereits früh das Pflichtenheft erstellt wurde und die Anforderungen definiert wurden, besteht nicht der Wunsch es anders zu wiederholen.

Welche Herausforderungen ergaben sich hinsichtlich der Gruppendynamik im Team und wie sind Sie damit umgegangen?
In der Durchführung eines Projekts ist die Kommunikation sehr wichtig. Aus diesem Grund haben wir bereits bei der Verkündung der Projektmitglieder zueinander Kontakt aufgenommen und eine Whatsapp-Gruppe gegründet für eine bessere Kommunikation. Daher sah ich keine Herausforderung in diesem Aspekt.

B. Reflexion - Dejan Babic

Die Aufgabenstellung wurde an den ersten Treffen mit unserem Studienprojektbetreuer festgelegt. Um das Projekt grob planen zu können haben wir ein Pflichtenheft erstellt, in welchem die wichtigsten Projektschritte aufgeführt wurden.

Wie passend war die Planung des Projektverlaufs, insbesondere der Termine? Wie häufig musste die Planung angepasst werden? Warum? Was war Ihre Rolle dabei?
Dadurch, dass wir uns schon relativ früh Gedanken gemacht haben wie das Projekt ablaufen soll haben wir uns dafür entschieden ein Pflichtenheft zu erstellen. Die Erstellung des Pflichtenhefts war zu Anfang des Projekts ein wichtiger Teil, welcher auch von unserem Betreuer abgesegnet wurde.

Welche Methoden haben Sie zur Unterstützung der gemeinsamen Planung genutzt und wie haben sie sich bewährt?
Zu Beginn des Projekts haben wir festgelegt, dass wir uns die meiste Zeit über Whatsapp oder Teamspeak(Sprachchat) unterhalten werden. Zudem kamen auch einige Treffen in der Uni hinzu.

Wie hat die Koordination bei der Projektarbeit funktioniert? Welche Probleme gab es? Wie wurden sie angegangen? Was war Ihre Rolle dabei?
Die Koordination war sehr gut. Durch das frühe Beginnen wollten wir verhindern, dass wir in Zeitdruck geraten. Das Pflichtenheft war dabei eine sehr große Hilfe. Probleme gab es eigentlich keine. Das größte Problem war es gemeinsame Termine zu finden um sich miteinander zu treffen.

In wieweit waren Sie ausreichend kompetent für die Mitarbeit? Welche Kompetenzen mussten Sie sich im Projektverlauf aneignen? Was haben Sie durch das Studienprojekt gelernt?
Es ist das zweite mal, dass ich an so einem großen Projekt mitgewirkt habe. Durch das erste Studienprojekt im Bachelorstudium wusste ich wie ein Projekt dieser Größe abläuft und konnte vieles aus dem ersten Studienprojekt nun im zweiten wiederverwenden. Vor Allem die Planung lief dieses mal noch koordinierter ab. Durch das Projekt habe ich erstmals einen praktischen Bezug zu Datenbanken gemacht. Jedoch konnte ich mich mithilfe der grundlegenden Kenntnisse aus der Bachelorvorlesung Datenbanksysteme sehr gut in die Thematik einarbeiten. Das Projekt war eine gute praxisbezogene Arbeit, bei welcher ich weitere Erfahrung

gesammelt habe im Team zu arbeiten.

Was haben Sie unternommen, um möglichst effizient und verlässlich im Projekt mitzuarbeiten?
Nach der Erstellung des Pflichtenhefts wurden die Aufgabenteile an die verschiedenen Gruppenmitglieder verteilt und es wurde angefangen zu arbeiten. Ich habe versucht meinen Teil des Projekts so gut es geht zu bearbeiten. Als Ziel habe ich mir gesetzt früher als gewollt fertig zu werden. Wenn Probleme oder Unklarheiten gab habe ich so schnell wie möglich nach Problemlösungen zu suchen oder meine Teammitglieder um Rat gebeten.

Was würden Sie anders machen, wenn Sie mit der zum Projektende erworbenen Erfahrung die Aufgabe nochmals bearbeiten müssten?
Dadurch, dass alle Teammitglieder schon einmal ein Projekt bearbeitet hatten, legten wir von Anfang an eine sehr geordnete und koordinierte Vorgehensweise an den Tag. Die Erstellung des Pflichtenhefts zu Anfang hat enorm viel zum Verständnis der Planung des Projekts beigetragen.

Welche Herausforderungen ergaben sich hin sichtlich der Gruppendynamik im Team und wie sind Sie damit umgegangen?
Das größte Problem waren unsere unterschiedlichen Stundenpläne. Dadurch, dass wir uns recht früh am Anfang des Projektes schon Gedanken darüber gemacht haben wann wir uns für das Studienprojekt treffen wollen, waren immer schnell passende Termine gefunden. Diese frühe Planung hat uns enorm geholfen keinen Zeitdruck zu bekommen. Zudem haben wir uns bei Problemen immer über Whatsapp oder Mail erreichen können.

C. Reflexion - Arlind Avdullahu

Hat sich die Aufgabenstellung oder Ihr Verständnis der Aufgabenstellung im Laufe der Zeit verändert? Wenn ja, wie?
Zu Beginn wurde ein Pflichtenheft angefertigt, welches den Zweck hatte die Aufgabenstellung für beide Seiten klar zu definieren. Nachdem das Pflichtenheft abgenommen wurde, war für beide Seiten klar, was das Ziel dieses Studienprojektes ist. Daher änderte sich das Verständnis nicht.

Wie passend war die Planung des Projektverlaufs, insbesondere der Termine? Wie häufig musste die Planung angepasst werden? Warum? Was war Ihre Rolle dabei?
Mithilfe dss Pflichtenheftes wurde das Projekt sehr früh geplant, womit sichergestellt wurde, dass es zu keinen Terminverzögerungen kommt. Da dieses Projekt nach dem Klassischen Vorgehensmodell durchgeführt wurde, haben wir die Meilensteintermine und Arbeitspakete geplant. Diese Arbeitspakete wurden danach unter den Teammitgliedern aufgeteilt.

Welche Methoden haben Sie zur Unterstützung der gemeinsamen Planung genutzt und wie haben sie sich bewährt?
Neben Github, welches zur Zusammenarbeit am Quellcode genutzt wurde, wurden zur Planung des Projektes Meilensteine und Arbeitspakte, mit der jeweiligen Dauer dieser, erstellt.

Wie hat die Koordination bei der Projektarbeit funktioniert? Welche Probleme gab es? Wie wurden sie angegangen? Was war Ihre Rolle dabei?
Wir haben für jede Woche Ziele definiert und uns einmal die Woche über Teamspeak getroffen, um den Fortschritt zu besprechen. Durch Nutzung von Github konnten alle Teilkomponenten zusammengebracht werden.

In wieweit waren Sie ausreichend kompetent für die Mitarbeit? Welche Kompetenzen mussten Sie sich im Projektverlauf aneignen? Was haben Sie durch das Studienprojekt gelernt?
Durch tiefes Verständnis des Projektmanagements und der Kenntnisse der geforderten Programmierkenntnisse, waren die Rahmenbedingungen erfüllt. Auch wenn Biologiekenntnisse vorhanden waren, musste ein Verständnis der Bioinformatik der Proteomik angeeignet werden. Durch das Studienprojekt habe ich somit ein Verständnis der Bioinformatik der Proteomik erworben.

Was haben Sie unternommen, um möglichst effizient und verlässlich im Projekt mitzuarbeiten?
Um die Effiziente und verlässliche Arbeit im Projekt zu unterstützen, wurde auf eine strukturierte Planung gesetzt. Jeder hat seinen Recherche- und Programmteil durchgeführt. Bei Fragen haben wir uns gegenseitig unterstützt.

Was würden Sie anders machen, wenn Sie mit der zum Projektende erworbenen Erfahrung die Aufgabe nochmals bearbeiten müssten?
Durch die Erstellung des Pflichtenheftes haben wir gemerkt, dass dies sehr wichtig war, da wir dadurch alles in der definierter Zeit erledigen konnten. Ich würde dies für jedes Projekt genauso wieder machen und nichts anders machen.

Welche Herausforderungen ergaben sich hinsichtlich der Gruppendynamik im Team und wie sind Sie damit umgegangen?
Die Gruppendynamik war sehr gut, da wir ständig miteinander kommuniziert haben. Für mich gab es keine Herausforderungen in dieser Hinsicht.

D. Requirements Specification (German)

Zielbestimmung

Es soll eine alternative Krankheits-Entitäten Datenbank anstatt der bestehenden UniProt-Anbindung recherchiert und in die bestehende Web-Applikation "BIONDA" eingebunden werden. Das Projekt was unter dem Arbeitstitel "Anbindung einer alternativen Quelldatenbank für Krankheits-Entitäten für die Biomarker-Datenbank" realisiert wird, soll in der Lage sein folgende Aufgaben zu erfüllen.

- Alternative Quelldatenbank für Krankheiten und evtl. Krankheitshierarchien mit BIONDA ansprechen.
- Quelldatenbank automatisch mit neue Krankheiten aktualisieren.
- Quelldatenbank liefert Daten wie Krankheit, Marker, Art des Markers, Genauigkeit, Suchquelle und mehr.
- Abbildung der Krankheiten in der Quelldatenbank als Hierarchie oder Auflistung.

Das Projekt ist in zwei Aufgabenbereiche aufgeteilt. Zum einen die Einbindung einer alternativen Quelldatenbank (Datenbank-Routine) und zum anderen die Anpassung der Web-Applikation (Client-Anwendung) "BIONDA". In den nächsten Abschnitten werden die genauen Anforderungen für das Projekt festgelegt.

Muss-Kriterien

- Der Aufbau der Anwendung muss modular sein, so dass ein einfaches Ersetzen der einzelnen Komponenten möglich ist.
- Die Anwendung muss über zwei Komponenten verfügen: Datenbank-Routine und Client-Anwendung.
- Die Client-Anwendung muss mithilfe von PHP angepasst werden.
- Die Datenbank-Routine muss eine MySQL-Datenbank ansprechen können.
- Es muss eine geeignete Quelldatenbank für die Datenbank-Routine recherchiert werden.
- Es müssen Kriterien für die Auswahl einer Quelldatenbank definiert werden.
- Es muss eine Entscheidung getroffen werden, ob die Krankheiten der Quelldatenbank als Hierarchie oder Auflistung gespeichert werden.

- Die Datenbank-Routine muss eine Parsing-Funktionalität besitzen, um den Inhalt der Quelldatenbank auszulesen (API-Zugriff) und zu verarbeiten.
- Die Parsing-Funktionalität muss automatisiert angestoßen werden können, um Wartungsfrei zu sein.
- Die Parsing-Funktionalität muss neue Inhalte in der Quelldatenbank erkennen und an die Datenbank-Routine weiterleiten.

Soll-Kriterien

- Die Datenbank-Routine soll ggf. neue Datenbankrelationen definieren oder bestehende anpassen.
- Die Krankheiten der Quelldatenbank sollen als Hierarchie gespeichert werden.
- Die Parsing-Funktionalität soll mit Perl implementiert werden.

Kann-Kriterien

- Die Parsing-Funktionalität kann, wenn die API-Guidelines es vorbestimmen, mit einer anderen Sprache implementiert werden.

Produkteinsatz

Zielgruppe

Die Anwendung wird von Mitarbeitern des Forschungsbereichs *Medizinische Bioinformatik*, ansässig am *Medizinischen Proteom-Center*, verwendet. Zukünftige Zielgruppen sollen sowohl die Pharmaindustrie, Wissenschaftler weltweit als auch Betroffene sein.

Betriebsbedingung

Die Anwendung ist in zwei Teilkomponenten aufgeteilt: Web-Applikation und Datenbank-Routine. Die Web-Applikation ist webbasiert und soll, bei noch vorhandenen zeitlichen Ressourcen, angepasst werden. Die Datenbank-Routine soll dahingehend angepasst werden, in dem eine alternative Krankheits-Datenbank als Quell-Datenbank integriert wird. Bedeutet eine Integration in die BIONDA-MySQL-Datenbank durch hinzufügen neuer Tabellen und Einträge in die Datenbank.

Folgende Programmiersprachen sollen hierbei zum Einsatz kommen:

- Java
  - Vorbereitung der Daten für die Datenbank, gegebenenfalls Perl, da zukünftig die Daten mit dieser Sprache vorbereitet werden sollen
- PHP
- Python
- SQL

Produktfunktionen

Da sich der Projektumfang hauptsächlich mit der Recherche und Anbindung einer alternativen Quelldatenbank in die bestehende Applikation "BIONDA" beschäftigt, müssen Anpassungen bzw. Erneuerungen an der Suche in "BIONDA" implementiert werden. Aus diesem Grund werden im folgenden die angepassten Produktfunktionalitäten näher erläutert.
Mit der Applikation "BIONDA" ist es möglich kostenfrei mithilfe von Biomarkern[5] oder einer Krankheits-Bezeichnung nach übereinstimmenden Treffern in wissenschaftlichen Veröffentlichungen zu suchen. Speziell hierfür werden die Abstracts jeder Veröffentlichung mithilfe von Text-Mining untersucht und inhaltlich für die Suche indexiert. Zusätzlich zu der Möglichkeit nach Krankheiten und Biomarkern zu suchen können die Treffer nach Wunsch auch auf "Sentence-wise", "Abstract-wise" oder beides eingeschränkt werden. Die Suche liefert abhängig von der Suchkategorie zwar die selben Metadaten allerdings ist ihre Bedeutung anders zu deuten. In den folgenden Abschnitten werden die Suchtreffer und ihre Metadaten erläutert.

- Krankheit
- ID
- Marker
- Art des Markers (miRNA, Gene, Protein)
- Jahr der Veröffentlichung
- Co-Occurence-based pValue
- Anzahl der Co-Occurence
- Evidenz (Sentence oder Abstract)

Abstracts-wise

Die Suche wird mithilfe der Option "Abstract-wise" reguliert, sodass der gewünschte Biomarker oder die Krankheit in den Abstracts der Veröffentlichungen erwähnt wurde. Dadurch ist es möglich Relationen zwischen den Veröffentlichungen zu ziehen, um einen Wert zu erhalten, welcher die Zuverlässigkeit des Treffers widerspiegelt. Bei einer Suche mit "Abstract-wise" werden die Evidenzen, im vgl. zu "Sentence-wise" nicht direkt angezeigt, weil die Abstracts im ganzen indexiert werden.

[5] Biomarker sind biologische Merkmale, die durch gewisse Prozesse gemessen werden können, um biologische Eigenschaften aufzuzeigen. Dabei kann es sich um Zellen, Gene, Proteine oder bestimmte Moleküle handeln.

Sentence-wise

Die Suche wird mithilfe der Option "Sentence-wise" reguliert, sodass der gewünschte Biomarker oder die Krankheit in Sätzen des Abstracts erwähnt wurde. Dadurch ist es möglich Relationen zwischen den Sätzen zu ziehen, um einen Wert zu erhalten, welches die Zuverlässigkeit des Treffers widerspiegelt. Bei einer Suche mit "Sentence-wise" wird als Evidenz der Satz zurückgegeben, welches den Biomarker bzw. die Krankheit enthielt. Dadurch hat der Nutzer die Möglichkeit den Kontext der Veröffentlichung zu verstehen.

Co-Occurence-based pValue

Eines der wichtigsten Metadaten in "BIONDA" ist der Co-Occurence-based pValue. Dieser gibt dem Nutzer eine Art Score zurück, um eine Zuverlässigkeit zu gewährleisten. Der Score wird mithilfe des χ^2-Test berechnet und basiert auf den Übereinstimmungen (Co-Occurence) von Markern und Krankheiten. Dieser Test benötigt für die Berechnung die true positives, true negatives, false positives, und false negatives eines spezifischen Biomarker und Krankheits Paares.

- True positives sind definiert als die Anzahl von Übereinstimmungen von einem Paar Biomarker X und einer Krankheit Y.
- False negatives sind definiert als die Anzahl von Suchtreffern von einem Biomarker X, jedoch ohne Krankheit Y.
- False positives sind definiert als die Anzahl von Suchtreffern von einer Krankheit Y, jedoch ohne den Biomarker X.
- True negatives sind definiert als die Anzahl von Suchtreffern von allen anderen Paare.

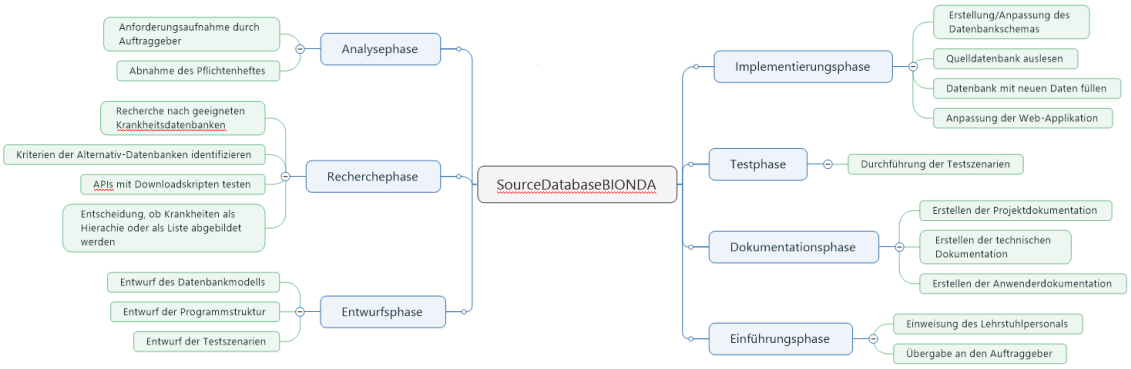
Generell kann gesagt werden, dass je mehr Übereinstimmungen in den Abstracts erreicht werden, desto besser ist der pValue und somit der Score. Die Aufgabe besteht nun darin die Suchfunktionen dahingehend anzupassen, sodass die neue Quelldatenbank einbezogen wird. Des Weiteren soll die Option existieren Krankheiten in einer Hierarchie-Ansicht zu veranschaulichen.

Projektphasen, Zeitplanung und Meilensteine

In den folgenden Abschnitten gehen wir auf die Projektphasen, Zeitplanung und Meilensteine (inkl. Arbeitspakete) ein.

Projektphasen

Das Projekt teilt sich in 7 Phasen auf, beginnend mit der Analysephase, und wird mit der Übergabe des Projekts beendet.



Projektphasen im Überblick

1. Analysephase
   1.1. Anforderungsaufnahme durch Auftraggeber
   1.2. Erstellung und Abnahme des Pflichtenheftes
2. Recherchephase
   2.1. Recherche nach geeigneten Krankheitsdatenbanken
   2.2. Kriterien der Alternativ-Datenbanken identifizieren
   2.3. APIs mit Downloadskripten testen
   2.4. Entscheidung, ob Krankheiten als Hierarchie oder als Liste abgebildet werden
3. Entwurfsphase
   3.1. Entwurf des Datenbankmodells
   3.2. Entwurf der Programmstruktur
   3.3. Entwurf der Testszenarien
4. Implementierungsphase
   4.1. Erstellung/Anpassung des Datenbankschemas

4.2. Quelldatenbank auslesen

4.3. Datenbank mit neuen Daten füllen

4.4. Anpassung der Web-Applikation

5. Testphase

5.1. Durchführung der Testszenarien

6. Dokumentationsphase

6.1. Erstellen der Projektdokumentation

6.2. Erstellen der technischen Dokumentation

6.3. Erstellen der Anwenderdokumentation

7. Einführungsphase

7.1. Einweisung des Lehrstuhlpersonals

7.2. Übergabe an den Auftraggeber

Arbeitspakete

Jede Phase des Projekts wurde auch als Arbeitspaket zusammengefasst. Ein Arbeitspaket besteht aus Startdatum und Enddatum, sowie dem Name des Teammitgliedes, das für das Arbeitspaket zuständig ist.

Analysephase

Arbeitspaket 1.1: Anforderungsaufnahme durch Auftraggeber

- Startdatum: 07.10.19
- Enddatum: 29.10.19
- Voraussetzung: -
- Ergebnis: Anforderungen an die Software
- Verantwortung: Projektteam

Arbeitspaket 1.2: Erstellung und Abnahme des Pflichtenheftes

- Startdatum: 07.10.19
- Enddatum: 29.10.19
- Voraussetzung: -
- Ergebnis: Pflichtenheft
- Verantwortung: Projektteam

Recherchephase

Arbeitspaket 2.1: Recherche nach geeigneten Krankheitsdatenbanken

- Startdatum: 19.10.19
- Enddatum: 25.10.19
- Voraussetzung: -
- Ergebnis: Krankheitsdatenbank-Kandidaten liegen vor
- Verantwortung: Projektteam

Arbeitspaket 2.2: Kriterien der Alternativ-Datenbanken identifizieren

- Startdatum: 19.10.19
- Enddatum: 25.10.19
- Voraussetzung: -
- Ergebnis: Kriterien für den Vergleich liegen vor
- Verantwortung: Projektteam

Arbeitspaket 2.3: APIs mit Download-Skripten testen

- Startdatum: 30.10.19
- Enddatum: 08.11.19
- Voraussetzung: AP 2.1
- Ergebnis: Alternativ-DB für BIONDA
- Verantwortung: Projektteam

Arbeitspaket 2.4: Entscheidung, ob Krankheiten als Hierarchie oder als Liste abgebildet werden

- Startdatum:08.11.19
- Enddatum: 17.11.19
- Voraussetzung: AP 2.1, AP 2.3
- Ergebnis: Plan für die Entwurfsphase
- Verantwortung: Projektteam

Entwurfsphase

Arbeitspaket 3.1: Entwurf des Datenbankmodells

- Startdatum: 18.11.19

- Enddatum: 24.11.19
- Voraussetzung: AP 2.4
- Ergebnis: Datenbankmodell
- Verantwortung: P1

Arbeitspaket 3.2: Entwurf der Programmstruktur

- Startdatum: 18.11.19
- Enddatum: 24.11.19
- Voraussetzung: AP 2.4
- Ergebnis: Programmstruktur
- Verantwortung: P2

Arbeitspaket 3.3: Entwurf der Testszenarien

- Startdatum: 18.11.19
- Enddatum: 24.11.19
- Voraussetzung: AP 2.4
- Ergebnis: Testszenarien
- Verantwortung: P3

Implementierungsphase

Arbeitspaket 4.1: Erstellung/Anpassung des Datenbankschemas

- Startdatum: 25.11.19
- Enddatum: 09.12.19
- Voraussetzung: AP 2.1-2.4, AP 3.1-3.3
- Ergebnis: Angepasstes Datenbankschema
- Verantwortung: Projektteam

Arbeitspaket 4.2: Quelldatenbank auslesen

- Startdatum: 25.11.19
- Enddatum: 09.12.19
- Voraussetzung: AP 2.1-2.4, AP 3.1-3.3
- Ergebnis: Geparste Dateien
- Verantwortung: Projektteam

Arbeitspaket 4.3: Datenbank mit neuen Daten füllen

- Startdatum: 09.12.20
- Enddatum: 19.12.20
- Voraussetzung: AP 4.1
- Ergebnis: Integration der Alternativ-Datenbank
- Verantwortung: Projektteam

Arbeitspaket 4.4: Anpassung der Web-Applikation

- Startdatum: 06.01.20
- Enddatum: 10.01.20
- Voraussetzung: AP 4.3
- Ergebnis: Angepasste Webseite
- Verantwortung: Projektteam

Testphase

Arbeitspaket 5.1: Durchführung der Testszenarien

- Startdatum: 11.01.20
- Enddatum: 11.01.20
- Voraussetzung: Testszenarien
- Ergebnis: Es ist bekannt, ob die Software alle Tests erfolgreich durchläuft
- Verantwortung: Projektteam

Dokumentationsphase

Arbeitspaket 6.1: Erstellen der Projektdokumentation

- Startdatum: 12.01.20
- Enddatum: 24.01.20
- Voraussetzung: fertige Software
- Ergebnis: Dokumentation über das Projekt
- Verantwortung: Projektteam

Arbeitspaket 6.2: Erstellen der technischen Dokumentation

- Startdatum: 12.01.20
- Enddatum: 24.01.20
- Voraussetzung: fertige Software
- Ergebnis: Dokumentation über die technischen Hintergründe der Software
- Verantwortung: Projektteam

Arbeitspaket 6.3: Erstellen der Anwenderdokumentation

- Startdatum: 12.01.20
- Enddatum: 24.01.20
- Voraussetzung: fertige Software
- Ergebnis: Dokumentation für den Anwender
- Verantwortung: Projektteam

Arbeitspaket 6.4: Pufferzeit

- Startdatum: 24.01.20
- Enddatum: 31.01.20
- Voraussetzung: unfertige Arbeitspakete
- Ergebnis: Vollendung der fehlenden Arbeitspakete
- Verantwortung: Projektteam

Einführungsphase

Arbeitspaket 7.1: Einweisung des Lehrstuhlpersonals

- Startdatum: 02.02.20
- Enddatum: 02.02.20
- Voraussetzung: fertige Software
- Ergebnis: Lehrstuhlpersonal ist in die Software eingewiesen
- Verantwortung: Projektteam

Arbeitspaket 7.2: Übergabe an den Auftraggeber

- Startdatum: 07.02.20
- Enddatum: 07.02.20
- Voraussetzung: fertige Software

- Ergebnis: Software an den Lehrstuhl übergeben
- Verantwortung: Projektteam

Meilensteine

Jede Phase des Projekts endet mit einem Meilenstein. Beim Erreichen des Meilensteins besteht die Möglichkeit die beendete Phase zu überprüfen und einen dieser 3 Wege zu wählen:

1. Alle bisherigen Aktivitäten befinden sich im Plan, die Phase kann abgeschlossen werden, das Projekt kann wie geplant fortgesetzt werden.
2. Einige Aktivitäten, die eigentlich laut Planung bereits abgeschlossen sein sollten, weisen in den relevanten Größen (Kosten, Termine, Ergebnisse) signifikante Abweichungen auf. Es muss nachgearbeitet werden, um die Phase abschließen zu können.
3. Es sind Ereignisse eingetreten, die eine sinnvolle Projektfortsetzung unmöglich erscheinen lassen, das Projekt wird gestoppt und ggfs. ganz eingestellt oder unter neuen Rahmenbedingungen völlig neu aufgestellt.

Des Weiteren wird jeder Meilenstein eine Anzahl von Arbeitspaketen zugewiesen. Diese Arbeitspakete sollten zur Deadline des Meilensteins erreicht sein, um eine Verzögerung des Projekts zu vermeiden. Nachfolgend werden die Meilensteine (MS) und Arbeitspakete aufgelistet:

MS 1: Analysephase beendet

Deadline: 29.10.19

- Arbeitspaket 1.1: Anforderungsaufnahme durch Auftraggeber
- Arbeitspaket 1.2: Erstellung und Abnahme des Pflichtenheftes

MS 2: Recherchephase beendet

Deadline: 17.11.19

- Arbeitspaket 2.1: Recherche nach geeigneten Krankheitsdatenbanken
- Arbeitspaket 2.2: Kriterien der Alternativ-Datenbank identifizieren
- Arbeitspaket 2.3: APIs mit Download-Skripten testen
- Arbeitsoaket 2.4: Entscheidung, ob Krankenheiten als Hierarchie oder Liste abgebildet werden

MS 3: Entwurfsphase beendet

Deadline: 24.11.19

- Arbeitspaket 3.1: Entwurf des Datenbankmodells
- Arbeitspaket 3.2: Entwurf der Programmstruktur
- Arbeitspaket 3.3: Entwurf der Testszenarien

MS 4: Implementierungsphase beendet

Deadline: 10.01.20

- Arbeitspaket 4.1: Erstellung/Anpassung des Datenbankschemas
- Arbeitspaket 4.1: Quelldatenbank auslesen
- Arbeitspaket 4.2: Datenbank mit neuen Daten füllen
- Arbeitspaket 4.3: Anpassung der Web-Applikation

MS 5: Testphase beendet

Deadline: 11.01.20

- Arbeitspaket 5.1: Durchführung der Testszenarien

MS 6: Dokumentationsphase beendet

Deadline: 24.01.20

- Arbeitspaket 6.1: Erstellen der Projektdokumentation
- Arbeitspaket 6.2: Erstellen der technischen Dokumentation
- Arbeitspaket 6.3: Erstellen der Anwenderdokumentation

MS 7: Pufferzeit ausgeschöpft

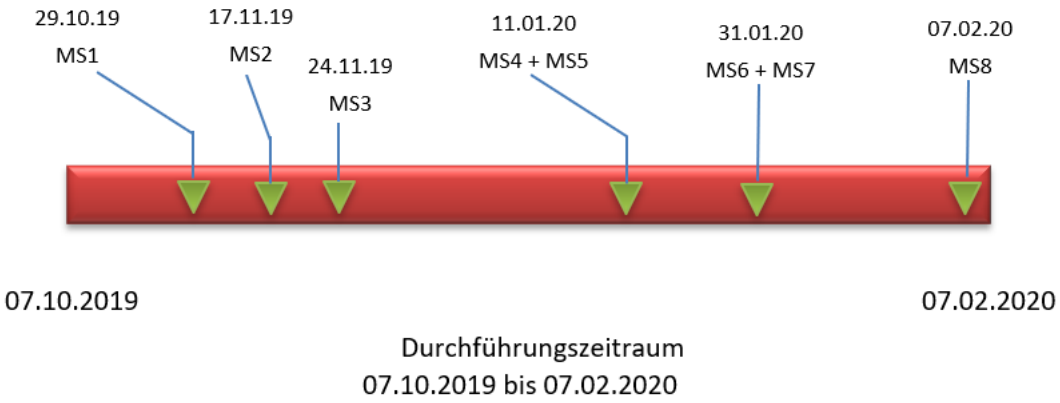
Deadline: 31.01.20

- Arbeitspaket 6.4: Pufferzeit

MS 8: Projekt abgeschlossen

Deadline: 07.02.20

- Arbeitspaket 7.1: Einweisung des Lehrstuhlpersonals
- Arbeitspaket 7.2: Übergabe an den Auftraggeber



Meilensteine im Überblick

Qualitätsanforderung

Im Folgenden werden die wesentlichen Qualitätsanforderungen erläutert.

Erweiterbarkeit

Die zu entwickelnde Softwarestruktur muss so gewählt werden, dass eine einfache Implementierung der angedachten Erweiterungen möglich ist. Mögliche Erweiterungen oder Änderungen bereits implementierter Funktionen sollen mit geringem zeitlichem Aufwand durchgeführt werden können, ohne dass eine tief greifende Umstrukturierung des bestehenden Quellcodes vorgenommen werden muss.

Fehlerrobustheit

Alle während des Betriebes auftretende Fehler und Warnungen müssen abgefangen und zur Anzeige gebracht werden. Im Fehlerfall muss der laufende Prozess kontrolliert abgebrochen werden. Zusätzlich muss entschieden werden, ob die Schwere des Fehlers einen Programmabbruch zur Folge haben soll. Angefallene Fehler sollen in einer lokal gespeicherten Datei protokolliert werden.

Wartbarkeit

Um eine langfristige Nutzung der Software zu ermöglichen, muss das Projekt so aufgebaut und dokumentiert sein, dass die Administration, Wartung und Weiterentwicklung ohne großen Einarbeitungsaufwand von einem anderen Entwickler oder Administrator durchgeführt werden kann. Um dies zu gewährleisten, muss die Anwendung modular programmiert und exakt dokumentiert sein. Dazu gehören eine Projektdokumentation, eine Anwenderdokumentation und eine technische Dokumentation.

Testszenarien

Die folgenden Tabellen stellen die Testfälle dar, die sich aus den Qualitätsanforderungen ableiten. Diese müssen nach der Implementierung des Projektes durchgeführt und erfolgreich durchlaufen werden. Jedes der folgenden Testszenarien muss auf allen unter Punkt 5.3 genannten Client-Betriebssystemen und Server-Betriebssystemen mit dem Ergebnis „bestanden“ durchlaufen werden.

QS-Ziele

QS-Ziel	Prüfung	Bestanden
Erweiterbarkeit	Integrierung von weiteren Funktionen	
Fehlerrobustheit	s. Punkt 5.2	
Wartbarkeit	Test durch zweiten Entwickler	

QS-Ziele

Des Weiteren muss eine Auswahl einer neuen alternativen Datenbank erfolgen. Da es sehr viele mögliche Datenbanken gibt, die der Aufgabenstellung entsprechen, muss eine Entscheidung anhand von verschiedenen Kriterien getroffen werden. Die folgende Tabelle stellt fünf Kriterien dar, welche bei der Wahl der neuen alternativen Datenbank berücksichtigt werden müssen. Um eine Entscheidung treffen zu können müssen die verschiedenen Kriterien gewichtet werden, um besser abwägen zu können, welche der Möglichkeiten die bestmögliche ist. Die Gewichtung der verschiedenen Kriterien ist ebenfalls in der Tabelle abgebildet.

Kriterium	Faktor	Rang
Updatehäufigkeit	$\times 2$	4
Anzahl an Einträgen	$\times 5$	1
Kostenlos und frei zugänglich	$\times 3$	3
Von öffentlicher Organisation	$\times 4$	2
Suche anhand Trivialnamen	$\times 1$	5

Kriterien der Quelldatenbank

Fehlerrobustheit

Testszenarien können in den verschiedenen Projektphasen ermittelt bzw. erstellt werden. Die wichtigeten Testszenarien sind jedoch die, die am Ende für den Test der neu angebundenen Datenbank erfolgen. Durch diese wird sichergestellt, dass BIONDA durch die neu angebundene Datenbank dazu in der Lage ist, deutlich mehr Suchtreffer für diverse Krankheiten zu liefern.

Fehler	**Erwartete Reaktion**	**Bestanden**
Suche nach Krankheit mithilfe eines Unterbegriffs zeigt nicht korrekte Hierarchie an	Gesamthierarchie anzeigen, Fehler protokolliere	
Suche nach Krankheit mithilfe des Oberbegriffs zeigt nicht korrekte Hierarchie an	Gesamthierarchie anzeigen, Fehler protokollieren	
Suche nach Biomarker, welcher einer Krankheit gehört, gibt falsche Krankheit an	Alternative Krankheiten anzeigen, Fehler protokollieren	
Downloadskript konnte Quelldatenbank nicht ansprechen	Fehler protokollieren, Wiederholung in einer Stunde	

Fehlerrobustheit

Produktumgebung

Anforderungen an die Produktumgebung

Für das Produkt wird ein Linuxserver benötigt, um die Webapplikation “BIONDA” bereitzustellen. Des Weiteren wird ein MySQL-Server benötigt, um alle nötigen Informationen für “BIONDA” zu speichern. Beides wird vom Auftraggeber bereitgestellt.

Anforderungen an die Entwicklungsumgebung

Hinsichtlich der Auswahl der zu benutzenden Entwicklungsumgebung gibt es keine Einschränkungen. Es muss aber gewährleistet sein, dass sämtliche Anforderungen erfüllt werden können. Insbesondere muss die Ausführung von Python, Java und Perl Programmen möglich sein. Des Weiteren wird vom Auftraggeber eine Testinfrastruktur zur Verfügung gestellt, um die nötigen Entwicklungsschritte zu testen.

E. Activity Diagram of Application

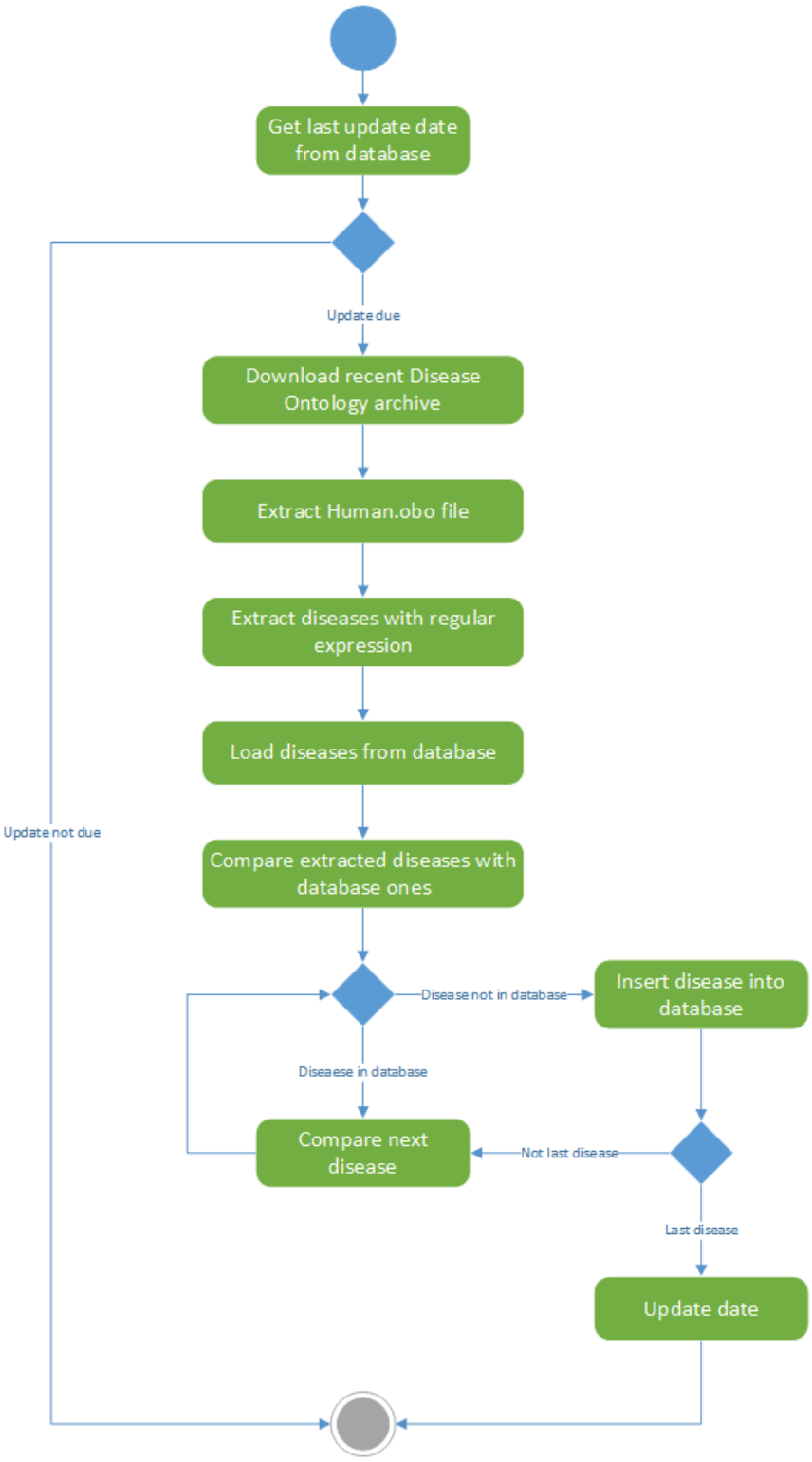


Figure 7.: Activity Diagram of Application

F. Entity Relationship Model

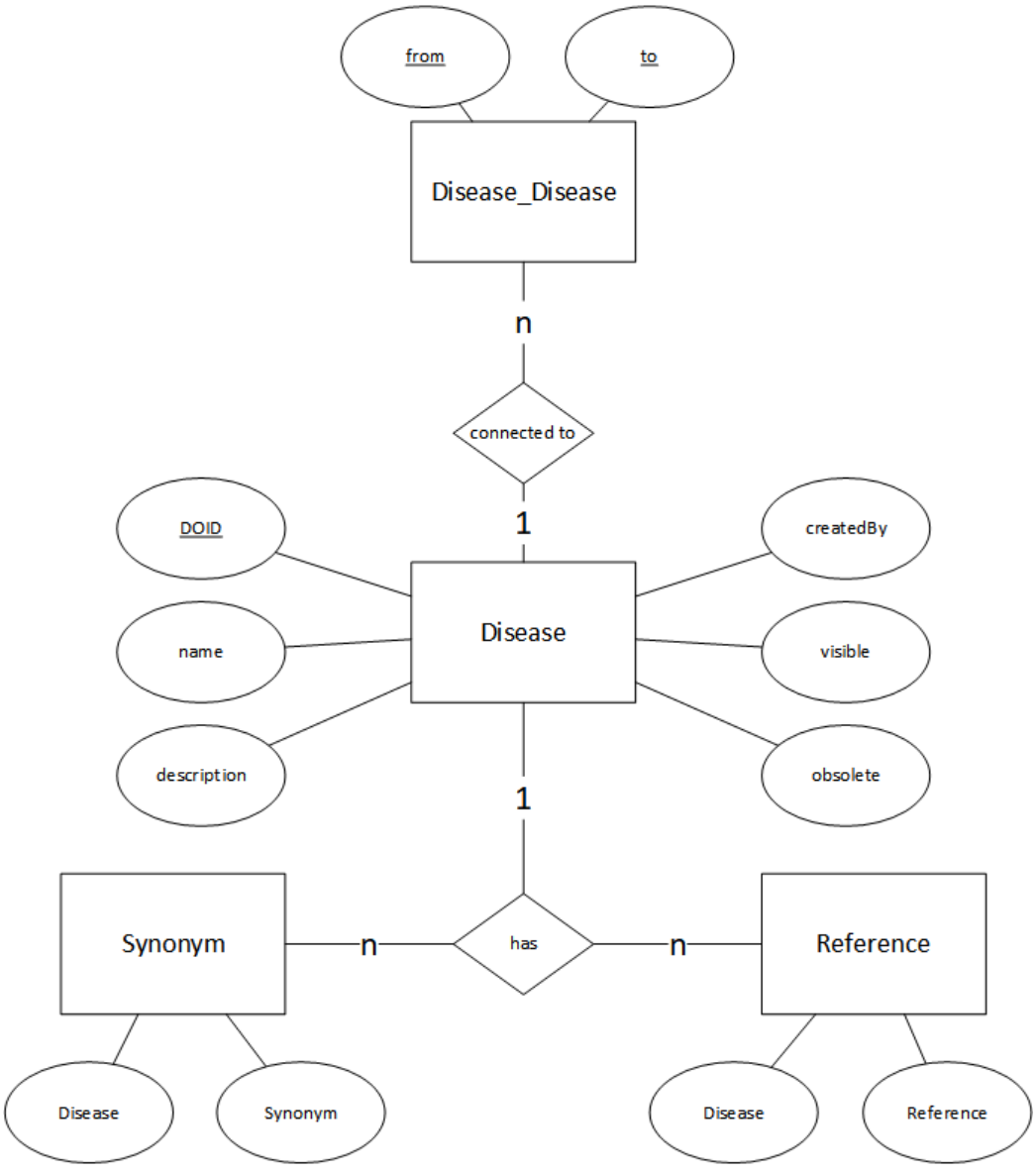


Figure 8.: Entity Relationship Model

G. Database Schema

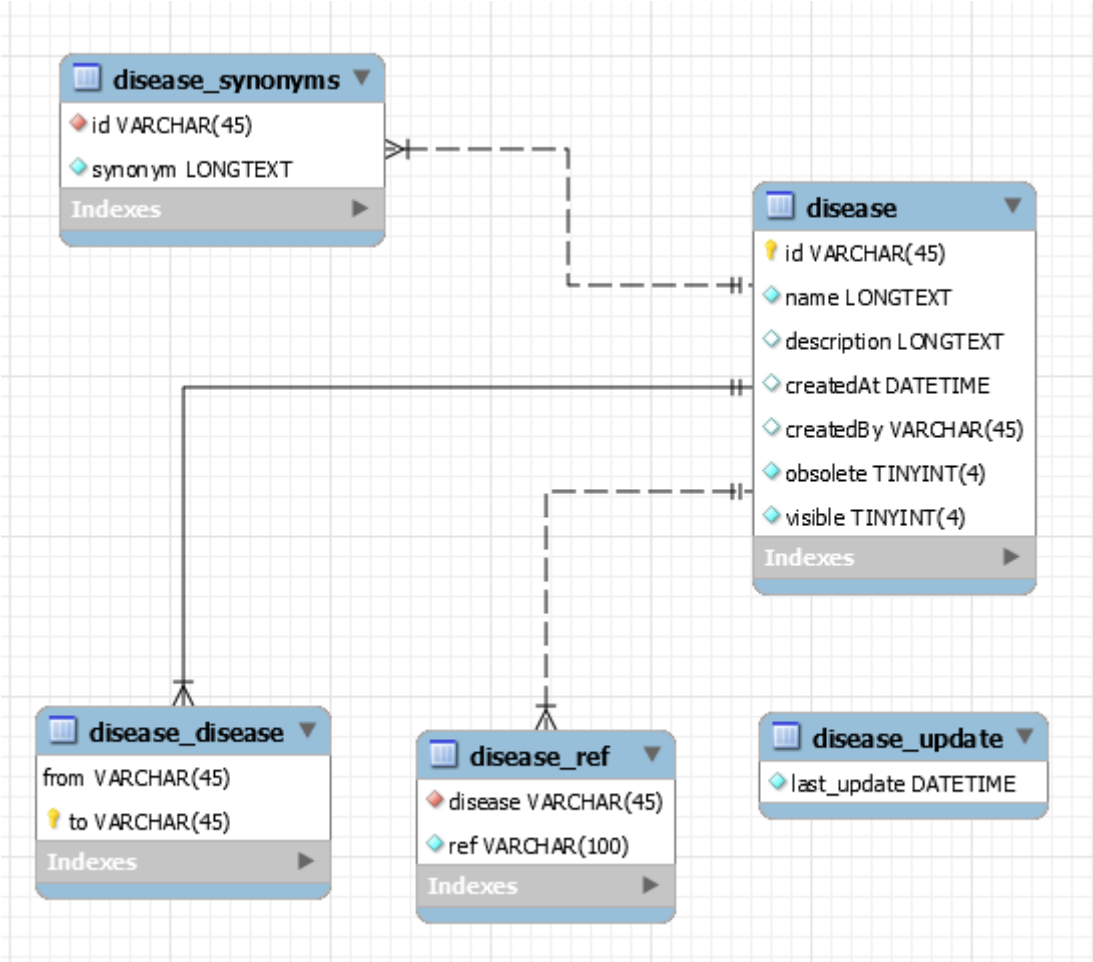


Figure 9.: Database Schema Modifications

SQL-Statements

Following are the SQL statements that are necessary for adding and modify the current database of Bionda.

```
--
-- Table structure for table `disease`
--

DROP TABLE IF EXISTS `disease`;
CREATE TABLE `disease` (
  `id` varchar(45) NOT NULL,
  `name` longtext CHARACTER SET utf8mb4
  COLLATE utf8mb4_0900_ai_ci NOT NULL,
```

```
  `description` longtext CHARACTER SET utf8mb4
  COLLATE utf8mb4_0900_ai_ci,
  `createdAt` datetime DEFAULT NULL,
  `createdBy` varchar(45) CHARACTER SET latin1
  COLLATE latin1_swedish_ci DEFAULT NULL,
  `obsolete` tinyint(4) NOT NULL DEFAULT '0',
  `visible` tinyint(4) NOT NULL DEFAULT '1',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

--
-- Table structure for table `disease_disease`
--

DROP TABLE IF EXISTS `disease_disease`;
CREATE TABLE `disease_disease` (
  `from` varchar(45) NOT NULL,
  `to` varchar(45) NOT NULL,
  PRIMARY KEY (`from`,`to`),
  CONSTRAINT `from` FOREIGN KEY (`from`)
  REFERENCES `disease` (`id`)
  ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

--
-- Table structure for table `disease_ref`
--

DROP TABLE IF EXISTS `disease_ref`;
CREATE TABLE `disease_ref` (
  `disease` varchar(45) NOT NULL,
  `ref` varchar(100) NOT NULL,
  KEY `disease_idx` (`disease`),
  CONSTRAINT `disease` FOREIGN KEY (`disease`)
  REFERENCES `disease` (`id`)
  ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
```

```
COLLATE=utf8mb4_0900_ai_ci;

--
-- Table structure for table `disease_synonyms`
--

DROP TABLE IF EXISTS `disease_synonyms`;
CREATE TABLE `disease_synonyms` (
  `id` varchar(45) NOT NULL,
  `synonym` longtext NOT NULL,
  KEY `disease_idx` (`id`),
  KEY `disease_syn_idx` (`id`),
  CONSTRAINT `disease_syn` FOREIGN KEY (`id`)
  REFERENCES `disease` (`id`)
  ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

--
-- Table structure for table `disease_update`
--

DROP TABLE IF EXISTS `disease_update`;
CREATE TABLE `disease_update` (
  `last_update` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

H. Class Diagram

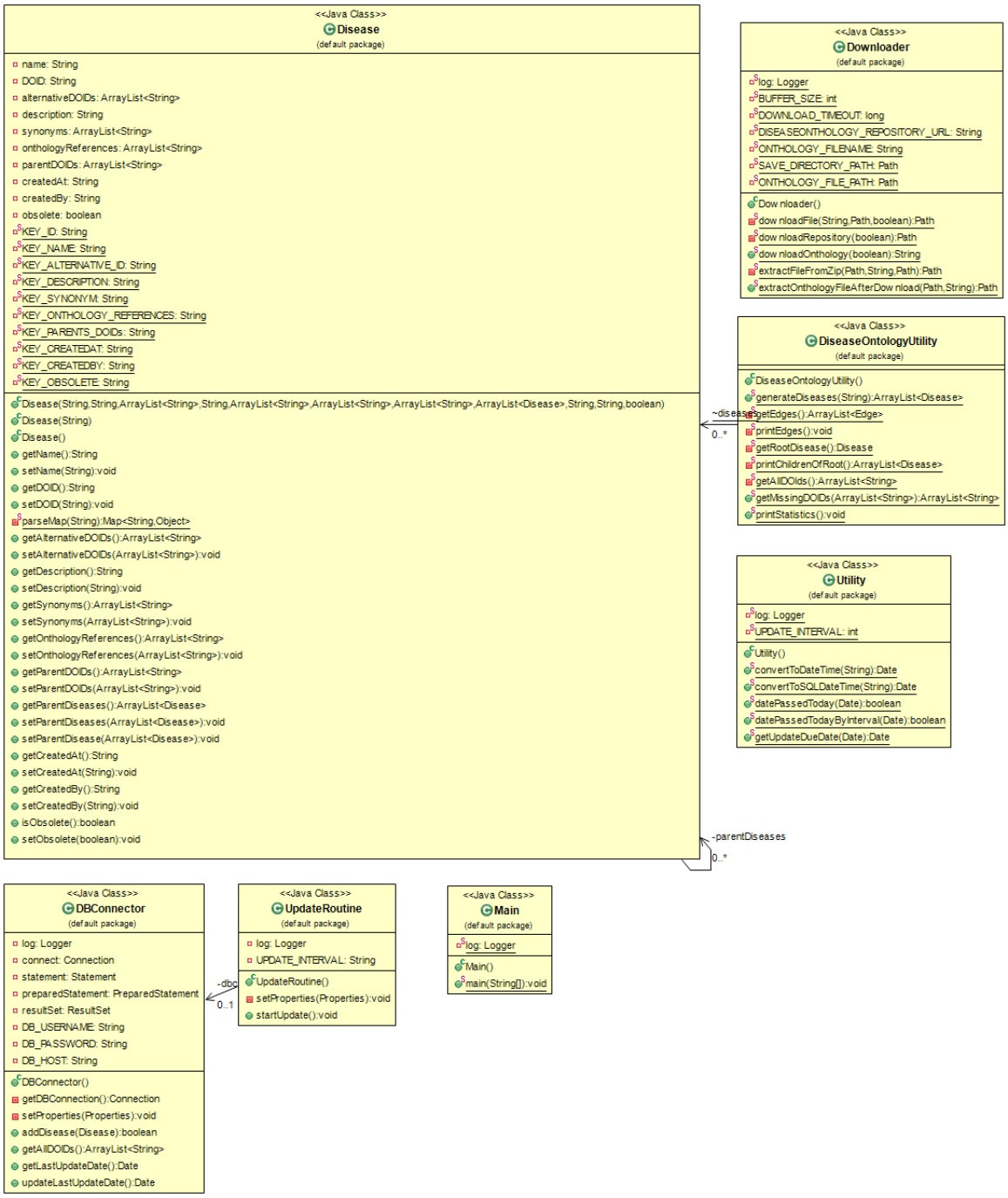


Figure 10.: Class Diagram of Application

I. Snippet of Diseases in HumanDO.obo

```
[Term]
id: DOID:0001816
name: angiosarcoma
alt_id: DOID:267
alt_id: DOID:4508
def: "A vascular cancer that derives from the cells that line the walls of blood vessels or
lymphatic vessels." [url:http\://emedicine.medscape.com/article/276512-overview,
url:http\://en.wikipedia.org/wiki/Hemangiosarcoma,
url:https\://en.wikipedia.org/wiki/Angiosarcoma,
url:https\://www.ncbi.nlm.nih.gov/pubmed/23327728]
subset: NCIthesaurus
synonym: "hemangiosarcoma" EXACT []
xref: MESH:D006394
xref: NCI:C3088
xref: NCI:C9275
xref: SNOMEDCT_US_2018_03_01:33176006
xref: SNOMEDCT_US_2018_03_01:39000009
xref: UMLS_CUI:C0018923
xref: UMLS_CUI:C0854893
is_a: DOID:175 ! vascular cancer

[Term]
id: DOID:0002116
name: pterygium
def: "A corneal disease that is characterized by a triangular tissue growth located_in cornea
of the eye that is the result of collagen degeneration and fibrovascular proliferation."
[url:https\://en.wikipedia.org/wiki/Pterygium_(conjunctiva)]
synonym: "surfer's eye" EXACT []
xref: MESH:D011625
xref: UMLS_CUI:C0033999
is_a: DOID:10124 ! corneal disease
created_by: laronhughes
creation_date: 2010-06-30T02:44:30Z

[Term]
id: DOID:0014667
name: disease of metabolism
def: "A disease that involving errors in metabolic processes of building or degradation of
molecules." [url:http\://www.ncbi.nlm.nih.gov/books/NBK22259/]
subset: DO_AGR_slim
subset: NCIthesaurus
synonym: "metabolic disease" EXACT [SNOMEDCT_2005_07_31:75934005]
xref: ICD10CM:E88.9
xref: ICD9CM:277.9
xref: MESH:D008659
xref: NCI:C3235
xref: SNOMEDCT_US_2018_03_01:30390004
xref: SNOMEDCT_US_2018_03_01:75934005
xref: UMLS_CUI:C0025517
is_a: DOID:4 ! disease
```

Figure 11.: Snippet of Diseases in Human.obo File
Source: (Schriml et al., 2019)

J. Example of Pattern Matching of Diseases

```
[Term]
id: DOID:999
name: hypereosinophilic syndrome
synonym: "eosinophilia" EXACT []
synonym: "Eosinophilic leukocytosis" EXACT [MTHICD9_2006:288.3]
xref: ICD10CM:D72.1
xref: ICD9CM:288.3
xref: MESH:D004802
xref: MTHICD9_2006:288.3
xref: SNOMEDCT_US_2018_03_01:27955006
xref: UMLS_CUI:C0014457
is_a: DOID:9500 ! leukocyte disease

[Term]
id: DOID:9993
name: hypoglycemia
def: "A glucose metabolism disease that is characterized by abnormally low
levels of blood glucose." [url:https\://www.niddk.nih.gov/health-
information/diabetes/overview/preventing-problems/low-blood-glucose-
hypoglycemia]
subset: NCIthesaurus
synonym: "Hypoglycaemia" EXACT []
xref: ICD10CM:E16.2
xref: ICD9CM:251.2
xref: MESH:D007003
xref: NCI:C3126
xref: SNOMEDCT_US_2018_03_01:66694000
xref: UMLS_CUI:C0020615
is_a: DOID:4194 ! glucose metabolism disease

[Term]
id: DOID:9997
name: peripartum cardiomyopathy
alt_id: DOID:11697
alt_id: DOID:11980
synonym: "antepartum peripartum cardiomyopathy" EXACT []
synonym: "postpartum peripartum cardiomyopathy" EXACT []
xref: GARD:220
xref: ICD10CM:O90.3
xref: ICD9CM:674.5
xref: SNOMEDCT_US_2018_03_01:16253001
xref: UMLS_CUI:C0877208
is_a: DOID:12930 ! dilated cardiomyopathy

[Typedef]
id: complicated_by
name: complicated_by

[Typedef]
id: composed_of
name: composed_of
def: "Component parts of anatomy of tissue made up of certain cells or other
body area/system or tissue types." [DO:lh]
```

Figure 12.: Example Pattern Matching of Diseases
(Green: Group Match, Blue+Green: Full Match)

K. Error List

Error Code	Error	Class
100	Couldn't read Application Properties	Downloader, UpdateRoutine, DBConnector, Utility
110	Couldn't set Application Properties	Downloader, UpdateRoutine, DBConnector, Utility
120	Couldn't set update interval. Wrong number? Please set a number as string	Utility
130	Couldn't set buffer_size/timeout. Wrong number/long? Please set a number/long as string	Downloader
200	Connection to the database couldnt be established. Wrong credentials?	DBConnector
210	Disease couldn't be inserted into the database due to an SQL Error	DBConnector
220	Disease couldn't be inserted into the database due to a SQL Date Parse Error	DBConnector
230	Diseases DO ID couldnt be fetched from database.	DBConnector
240	Update record couldnt be fetched. Maybe no first default entry?	DBConnector
250	Last update date couldn't be updated!	DBConnector
000	Database connection couldnt be closed!	DBConnector
300	Couldn't read HumanDO.obo file. Does it exist in the home directory?	DiseaseOntologyUtility
310	Couldn't process HumanDO.obo file. Did it changed its size (>2GB)?	DiseaseOntologyUtility
320	Couldn't process HumanDO.obo file. Something is wrong with the format of the file or the regular expression!	DiseaseOntologyUtility
400	Couldn't compare update date with today. Is update interval set right?	Main
410	Couldn't download DO archive. Check stack trace for reason.	Main
420	Something interrupted the download process. Check stack trace for reason.	Main
010	Unexpected error occurred. Check stack trace for reason.	Main

Table 6.: Documentation of Errors

Erklärung

Wir erklären, dass das Thema dieser Arbeit nicht identisch ist mit dem Thema einer von uns bereits für eine andere Prüfung eingereichten Arbeit ist.
Wir erklären weiterhin, dass wir die Arbeit nicht bereits an einer anderen Hochschule zur Erlangung einer Studienleistung eingereicht haben.

Wir versichern, dass wir die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen verwendet haben. Die Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen sind, sind unter Angabe der Quellen der Entlehnung kenntlich gemacht. Dies gilt sinngemäß auch für gelieferte Zeichnungen, Skizzen, bildliche Darstellungen und dergleichen.

We declare that the topic of this thesis is not identical to the topic of another thesis authored by us for another examination. Furthermore we declare that we did not submit this thesis to another university for the purpose of obtaining an academic degree.

We assure that we composed this thesis on our own without using other sources than the denoted ones. Those parts of this thesis, which are taken literally or in meaning from other works, are indicated by citing their origin. This also applies analogously to the provided drawings, sketches, illustrations and suchlike.

Datum	Unterschrift
Datum	Unterschrift
Datum	Unterschrift