UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Ida Maria Orula

The Process of Creating a Scientific Knowledge Base for Pharmacogenetic Testing

Master's Thesis (30 ECTS)

Supervisor(s): Sulev Reisberg, PhD
Kersti Jääger, PhD

Tartu 2021

The Process of Creating a Scientific Knowledge Base for Pharmacogenetic Testing

Abstract:

Pharmacogenetics is a branch of personalised medicine that investigates the impact of genetic factors on drug response. Adequate IT solutions are the key components of applying pharmacogenetic knowledge to clinical practice. This thesis describes the process of designing and implementing a curated scientific knowledge base to be used by a software-based medical device for pharmacogenetic testing. Based on scientific literature, a prototype is built to identify main domain objects and processes for curating the knowledge base. The final implementation is created by using the Qure Data Management Platform.

Keywords:

Knowledge base, pharmacogenetics, data curation

CERCS: P170 Computer science, numerical analysis, systems, control, B110 Bioinformatics, medical informatics, biomathematics, biometrics

Farmakogeneetilise testi teadmusbaasi loomise protsess

Lühikokkuvõte:

Farmakogeneetika on personaalmeditsiini haru, mis uurib geneetiliste tegurite mõju ravimivastusele. Farmakogeneetiliste teadmiste tervishoiusüsteemis rakendamise eelduseks on sobivate IT lahenduste kättesaadavus. Selles magistritöös kirjeldatakse farmakogeneetilist testimist läbiviiva meditsiiniseadme jaoks kureeritud teadmusbaasi disainimise ja rakendamise protsessi. Teaduskirjanduse põhjal loodakse lahendusele prototüüp, mida kasutatakse teadmusbaasi domeeniobjektide ja põhiliste kureerimisprotsesside tuvastamiseks. Teadmusbaas luuakse Qure Data Management Platform tarkvara kasutades.

Võtmesõnad:

Teadmusbaas, farmakogeneetika, andmete kureerimine

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria), B110 Bioinformaatika, meditsiiniinformaatika, biomatemaatika, biomeetrika

Table of Contents

1 Introduction

According to Vogenberg et al. [1], a very common obstacle for healthcare professionals is that drugs can affect different patients in different ways. While a certain treatment might work without complication for one person, it can cause severe adverse reactions in another. Variable drug response has been affecting people for centuries, but it was not until the early 1950s that it became a subject of careful examination. Over the past few decades, it has been acknowledged that in addition to general health status, age or environmental exposure, drug response is highly dependent on genetic factors.

Personalised medicine is a development in the medical system that aims to overcome the one-size-fits-all approach in treatment plans, therefore helping doctors make better-informed decisions for each patient based on their individual medical and genetic background [2].

In order to implement personalised medicine as common practice, solutions relying on information technology are of crucial importance in making genetic data and instructions for its interpretation available for the physicians [1].

In Estonia, the GenMed project[1] is building an IT infrastructure to make genetics-based medical software devices available to doctors in their daily work. The author of this thesis is doing her practical training in the pharmacogenetics working group of the project. The working group's goal is to develop a pharmacogenetics test that the physicians can use to make adjustments to drug prescriptions. In order for the test to give reliable outputs, its decisions need to be based on a reliable set of scientific and clinically applicable information. Moreover, all research articles and scientific evidence that the test is based on have to be stored in an accessible, updatable and trackable database.

The goal of this thesis is to propose a design and implementation for such a knowledge base. The author of the thesis determines the necessary processes and functionalities of the knowledge base, validates and improves them on a prototype and introduces the final implementation. In addition, the author also curates relevant pharmacogenetic information from scientific literature and online databases in order to detect possible complications from real data and to ensure that the proposed solution is well suited for the data it is expected to hold and provide.

The first chapter provides a brief introduction to relevant fields of study. In the second chapter, the methodology of creating the knowledge base is introduced and justified. The third chapter focuses on the requirements and the implementation of the actual knowledge base using the Qure Data Management Platform[2].

[1] https://sisu.ut.ee/genmed/en?lang=en

[2] https://www.quretec.com/

2 Terms and Notations

Allele – Version of the same gene that have different DNA spelling changes compared to each other[3]. In the context of this thesis, allele refers to the allele of one specific genetic variant.

CPIC – Clinical Pharmacogenetics Implementation Consortium – provides clinical gene-drug specific guidelines.

Diplotype - A specific combination of two haplotypes (see haplotype definition below).

Effect allele – The mutation allele in a variant. It is compared against the reference or wild type allele.

Gene – Genes are made up of DNA and provide instructions for the body on how to make proteins.

Genotype – The genetic state of both copies of a genetic variant.

Haplotype - A combination of multiple spelling changes within a particular gene.

Linkage disequilibrium – Correlation between nearby alleles. Perfect linkage disequilibrium means that alleles are always inherited together.

Nucleotide – A building block of DNA that consists of a phosphate group, a 5-carbon sugar, and a nitrogenous base (adenine (A), cytosine (C), guanine (G), and thymine (T)).

Pharmacogenetics (PGx) – The study of how variation across the genome influences drug response.

Pharmacogene – A gene whose variants affect drug response.

PharmGKB – Pharmacogenomics Knowledge Base – public PGx knowledge base.

PharmVar – Pharmacogene Variation Consortium - provides classifications of pharmacogenetic variants in the standardized star allele nomenclature.

Phenotype – A physical characteristic that is determined by genetic variants in your genome. In the context of PGx, they describe how effectively drugs are metabolized.

Reference allele – The wild type allele of a variant that effect alleles are compared against.

Reference genome – A complete genome sequence that genetic variability is compared against.

Related allele – In the context of this thesis, a variant allele that belongs to a certain star allele.

Rs-number – An identification number assigned by the National Center for Biotechnology Information (NCBI) to a specific genetic variant.

Single nucleotide polymorphism (SNP) – A type of genetic variant where only one letter is changed in the DNA sequence.

[3] PharmGKB glossary: https://www.pharmgkb.org/page/glossary

Star allele – A method of labelling haplotypes in genes (e.g. *2, *3 etc.). (One or more genetic variants that are inherited together.)

Genetic variant – An individual change found in a person's genetic code. Also referred to as a variant.

3 Background

Since this thesis's topic is multidisciplinary, some general background information about the related fields of study is crucial. The first subchapter introduces the concept of personalised medicine, while the second subchapter explicitly introduces pharmacogenetics. Some of the most renowned online resources for pharmacogenetic information and the availability of genetic data are briefly described in subchapters three and four, respectively. The fifth subchapter describes the concept of a medical software device. The sixth subchapter provides more information about the GenMed project and the pharmacogenetic device that will use the knowledge base designed in this thesis.

3.1 Personalised medicine

Personalised medicine is not a new expression, but its interpretation has shifted over time. According to Pokorska-Bocci et al. [3], the term was first mentioned in literature in 1971. In this article [4], the author expressed his concerns about losing the personalised approach in the modern healthcare system and seeing patients as conditions to be cured rather than as fellow human beings. According to Jørgensen [5], it was almost 30 years later when the article "New Era of Personalized Medicine: Targeting Drugs for Each Unique Genetic Profile" was published in *The Wall Street Journal* and a few months later in *The Oncologist*, bringing the term in the context of individual drug prescription.

One of the goals of personalised medicine today is to move from one-size-fits-all treatment plans and trial and error based drug prescriptions to a more individual approach to patient care, in which the patients' genomic and epigenomic data is also taken into consideration when making therapeutic decisions [2]. This does not in any way imply that the new methods would override the physicians' expertise and experience - the idea is to provide additional information along with guidance for its interpretation [1].

Mathur et al. [2] state that another valuable asset that a personalised medicine approach can bring to the healthcare system is its preventive abilities. Instead of just identifying a disease by its symptoms and providing treatment accordingly, personalised medicine offers ways to assess the patients' risk of developing certain conditions. Therefore, personalised medicine can play a significant role in improving life quality in the future. In addition, it can bring financial advantages since adverse drug reactions and misdiagnosis are known to augment healthcare costs significantly.

A critical prerequisite for implementing personalised medicine is the existence of a suitable IT infrastructure - physicians must have easy access to electronic health records that encompass the patient's medical data, their genetic and molecular profile, as well as to additional resources that provide guidance on how the genomic information should be interpreted [1].

3.2 Pharmacogenetics

According to Lass et al. [6], pharmacogenetics (PGx) is a field of study that combines knowledge from pharmacology and genetics to provide better-informed prescription recommendations. Several genes across the human genome can alter the person's response to different drugs. These genes

are known as pharmacogenes. Mutations in a pharmacogene may change the functionality of the protein that it encodes, thereby compromising the protein's ability to regulate the metabolism, toxicity or transportation of certain drugs. Based on their functionality, gene variants can be of *unknown*, *uncertain*, *normal*, *increased*, *decreased*, or *no* function. If the functions of both copies of a pharmacogene (one copy from each parent) are identified, a person's pharmacogenetic phenotype, or the ability of the organism to efficiently metabolise certain drugs, can be determined.

The exact names of the phenotypes may vary across resources due to changes in naming conventions. In 2017, a term-standardisation project [7] was carried out by the Clinical Pharmacogenetics Implementation Consortium[4] (CPIC) and the terms *ultrarapid*, *rapid*, *normal*, *intermediate* and *poor* metabolizer were agreed upon. In older literature, the term *extensive metaboliser* is to be interpreted as *normal*.

There are several kinds of mutations of different magnitudes that might occur in a pharmacogene's DNA sequence. Only one DNA nucleotide has been replaced in the most straightforward cases, but more complex cases, where a much longer sequence is either missing or multiplied, are also possible [6]. The *rs* or *rsid* accession numbers assigned by the dbSNP database[5] are used internationally to document discovered variants. The human reference genomes GRCh37 and the newer GRCh38 are used[6] for documenting the location of a mutation.

Apart from the rs-number, which corresponds to one specific mutation, the star allele nomenclature is the standard way of describing genetic variance. Star alleles can be defined by the alleles (possible DNA sequences) of one or several genetic variants, and each star allele is associated with a specific function. The most common sequence of a gene that typically encodes a fully functional protein is named *1 or *wild type,* and all other mutations are compared against it [8].

It is important to note that in this thesis, the term allele strictly means the allele of one particular genetic variant. If a longer region (a combination of variants) is meant, then the term *star allele* is always used. Table 3.1 below shows a small segment of the CYP2C9 gene's allele reference table.

Table 3.3-1. CYP2C9 Allele Reference Table *(adapted from https://www.pharmgkb.org/page/cyp2c9RefMaterials)*

rsID		rs200183364	rs1799853	rs7900194
CYP2C9 Allele	**Function**			
*1	Normal	G	C	G
*2	Decreased		T	

It includes three variants (rs200183364, rs1799853, rs7900194) and two star alleles (*1, *2). *1 is the wild-type star allele that provides the reference values for all variants of that gene. Star allele *2 with decreased function is defined by the change of C to T at rs1799853.

[4] https://cpicpgx.org/

[5] https://www.ncbi.nlm.nih.gov/snp/

[6] https://www.ncbi.nlm.nih.gov/grc/human

Table 3.2 below depicts a section of the genotype-phenotype table of CYP2C9, which shows how different star allele functions result in different pharmacogenetic phenotypes.

Table 3.3-2. CYP2C9 Genotype-Phenotype Table (*adapted from* [15])

Likely phenotype	Genotypes	Examples of diplotypes
Normal metaboliser	Normal + Normal	*1/*1
Intermediate metaboliser	Normal + Decreased OR Normal + No function OR Decreased + Decreased	*1/*2 *1/*3 *2/*2
Poor metaboliser	Decreased + No function OR No function + No function	*2/*3 *3/*3

3.3 Pharmacogenetic Resources

As pharmacogenetics is a rapidly developing field of study, there are many online sources where research results can be found. This includes web pages where relevant articles can be searched by specifying the gene name and variant, curated online databases for variant-specific information and clinical guidelines that provide gene and drug-specific information and prescription recommendations. In addition, there have been several projects focused on collecting genetic data from specific populations. The resulting data of some of those projects have been made publicly available for research purposes. This subchapter introduces the resources that are relevant in the context of this thesis.

According to Gaedigk et al. [9], the Pharmacogene Variation Consortium[7] (PharmVar) is a publicly available database that provides classifications of pharmacogenetic variants in the standardized star allele nomenclature. The star allele definitions from PharmVar are widely accepted by leading pharmacogenetic experts, including the CPIC [10].

In the context of this thesis, PharmVar was mostly used to clarify different definitions of the same star allele found in literature. PharmVar includes tables that associate different suballeles with their corresponding core allele definition. Therefore, if the newly appearing allele in an article was a suballele of a known star allele, it could simply be interpreted as one star allele. The limitation is that currently, PharmVar only holds information about the CYP family genes.

Another online database that operates in close collaboration with PharmVar is the Pharmacogenomics Knowledge Base[8] (PharmGKB), further described by Whirl Carrillo et al. [11]. It is a publicly available knowledge base that contains a wide variety of carefully curated pharmacogenetic information. They present summaries of variant-drug-phenotype combinations based on a thorough literature review. Based on this information, they proceed to report relative risks of side effects among reported genotypes. Evidence level scores are used to show how confident the curators are of each association's correctness. Based on scores, evidence is then marked as high, moderate, low or unsupported[9].

[7] https://www.pharmvar.org/
[8] http://www.pharmgkb.org
[9] https://www.pharmgkb.org/page/clinAnnLevels

In the context of this thesis, the search engine on the PharmGKB web page was used for finding article references for certain genetic variants. In addition, two types of information tables were used as a starting point for determining the variants most relevant to the GenMed pharmacogenetic test. The data files that were used from PharmGKB were the following:

1) Allele definition table, which associates different genetic variants to star alleles and also provides information such as the variant's effect on protein and its position according to GRCh38.
2) Allele functionality reference, which assigns standardized function names to all star alleles.

These files also include a change log and a notes section, where additional information can be found if necessary.

According to Relling and Klein [12], CPIC intends to facilitate bringing pharmacogenetic knowledge to clinical practice by providing publicly available, evidence-based gene and drug-specific guidelines that link different genotypes to phenotypes and corresponding dosage recommendations.

Relling et al. [13], states that since its creation, CPIC has become internationally accepted as a benchmark for the clinical implementation of pharmacogenetics. As of July 2019, CPIC had published 23 guidelines for 19 genes and 46 drugs. Drugs and genes are selected for new guidelines based on emerging evidence about their clinical significance, and existing guidelines are also updated accordingly.

In this thesis, CPIC was considered the primary information source, and in case of contradiction between different sources, information from CPIC was preferred. CPIC guidelines, along with their supplementary files, were used to confirm different star alleles' functions and gather genotype to phenotype association tables.

3.4 Genetic Resources

The Estonian Biobank is a database where the genetic data of adults in the Estonian population has been gathered since 2002 [14]. In addition, the gene donors' electronic health records are also stored in the database. This allows for a thorough analysis of how different genes and health conditions are related to each other.

The amount of genetic data available about patients is growing rapidly. Within a few years, the number of gene donors in the Estonian Biobank has grown from 50 000 to 200 000 people[10]. This reduces the cost and complexity of pharmacogenetic testing since necessary data has already been gathered for many patients, and there is no need for additional blood samples and DNA analysis. In addition, such sudden growth in the number of gene donors suggests that people are indeed interested in interpreting their genetic data.

In addition to collecting and analysing the genomes, the Estonian Biobank is also giving feedback to the gene donors about how their health may be affected by their genes. However, to securely

[10] https://geenidoonor.ee/geenivaramu

and effectively provide this information to the patients and their physicians in the national health care system, a suitable IT infrastructure is required.

One of the obstacles that need to be overcome before extensively applying genetics-based recommendations and risk scores to clinical practice is that the software solutions that analyse genetic data fall under the regulations of a medical software device, which sets specific demands for their development.

3.5 Medical Software Device

According to article 2(1) of Regulation (EU) 2017/745[11], a software falls under the category of a medical device if its intended use as defined by its manufacturer serves a certain medical purpose, such as giving a diagnosis or providing information resulting from *in-vitro* examination of human body specimens. Therefore, any device intended to bring interpretations of genetic data to clinical practice qualifies as a medical device.

Each medical software device must follow specific risk management principles and a specified development life cycle (Annex I, Chapter I Section 17.2). The life cycle is regulated by the international standard IEC 62304 "Medical Device Software – Software Life Cycle Processes". The regulation aims to prevent any harm a medical device could produce for the patient.

3.6 GenMed and the Pharmacogenetic Device

The national GenMed project aims to create an IT infrastructure to enable applying personalised medicine concepts in clinical practice. The project runs from 2019 to 2022 and is conducted in collaboration between the University of Tartu, and national institutions, such as the Estonian Health Insurance Fund and the Ministry of Social Affairs. One of the key goals of this project is to enable comprehensive software-based pharmacogenetic testing using genetic data available in the Estonian Biobank. A significant milestone is to develop a medical device for PGx testing which interprets the patient's genome and identifies his/her pharmacogenetic phenotype for various pharmacogenes. This phenotype will be later used for providing drug dosage recommendations for the doctors. A detailed description of the PGx testing device can be found in the Master's thesis "Designing a Pharmacogenetic Test as a Medical Software Device" by Taavi Luik, which is in draft status today.

One of the essential components of the PGx test is a detailed compilation of evidence that forms the basis of all decisions made by the algorithm. Even though many online resources provide relevant information, they are scattered around over multiple websites, which makes them harder to access and raises uncertainty about which resources to choose. In addition, the provided tables tend to be superfluous and sometimes even contradictory - they can include large numbers of variants and star alleles with meagre clinical evidence, and some definitions can differ among resources.

Consequently, it was decided that a new knowledge base, where relevant information is clearly

[11] https://ec.europa.eu/health/sites/default/files/md_topics-interest/docs/md_mdcg_2019_11_guidance_en.pdf

separated and all peculiarities are carefully reviewed by the GenMed PGx team, is necessary to ensure a comprehensive solution and correct implementation of the pharmacogenetic device.

The interaction between currently available resources, knowledge base and the PGx device is seen in figure 3.1. The knowledge base contents are created by curating the data from various public resources. The curated data, which includes instructions on how different genotypes and phenotypes are related, is then exported to the PGx device. The device uses these instructions to assign pharmacogenetic phenotypes to the patient, based on their genetic data, which was received as an input. These phenotypes are later associated with specific prescription guidelines and made available to the doctors.

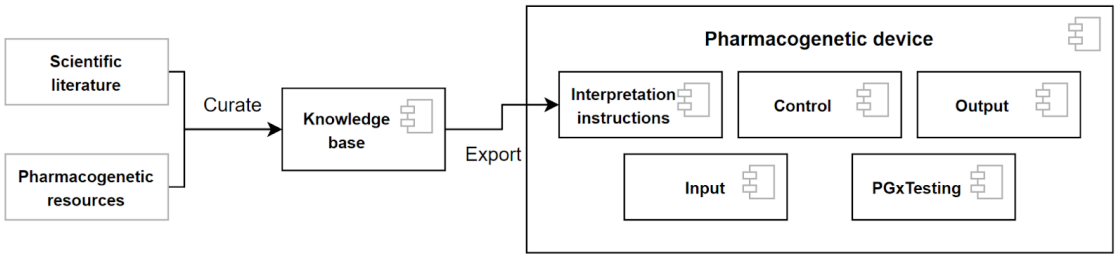

Figure 3.1. Knowledge base interaction with the pharmacogenetic device.

This thesis aims to create and describe a methodology for creating the knowledge base, determine its user groups and main processes, propose an implementation, and, based on the scientific evidence, fill it with information that is important specifically in the context of the GenMed pharmacogenetic device.

4 Methodology

In order to create a sufficient, curated, and fully traceable knowledge base for the PGx medical device, its requirements, supported processes and necessary user roles needed to be defined. To better understand the required individual data elements and data types of the knowledge base and define the curator processes, an initial prototype was created based on the reviewed literature.

The author of this thesis conducted a thorough review of available literature for five pharmacogenes of interest (CYP2C9, CYP3A5, UGT1A1, TPMT, DPYD) and learned to navigate multiple online resources. This process helped to attain an initial grasp of the available data and the key problems with using it for clinical PGx testing. Since the prototype needed to be easily accessed and modified by all members of the pharmacogenetic team in the GenMed project, Google Drive and Google Sheets were chosen as the most straightforward and most reasonable data collection platform for developing the knowledge base prototype.

The initial structure was agreed upon, and actual scientific data from the reviewed literature was added to the prototype upon careful inspection for consistency. By further analysing the data and the design of the PGx testing algorithm, the exact structure and necessary functionalities of the knowledge base were determined in more detail. The prototype was then iteratively reviewed and updated to turn it into a suitable baseline for the actual implementation. From the resulting structure, the domain objects for the knowledge base implementation were derived.

Working with the prototype was a close simulation of how the actual knowledge base should be used. Therefore, it allowed the author of this thesis to determine the general processes that the knowledge base must support (e.g. content modification, review, data export etc.). These workflows were documented using UML action diagrams and short descriptions.

Furthermore, a list of user needs from the perspective of the knowledge base's curators was gathered. As the prototype was improved over time, new and more detailed needs became apparent. These user needs were then transformed into a list of expectations. Additional requirements were also derived from the PGx device. The list of expectations was reviewed and validated by the GenMed pharmacogenetics team.

Once all necessary structural, content- and process-oriented elements of the knowledge base had been determined and stored in the prototype, and the expectations for the knowledge base were clear, a need for a more sophisticated database platform became apparent. From the existing software solutions that would support dynamic data management functions, the Qure Data Management Platform (QDMP)[12] was selected for the final knowledge base implementation. The complete data model for the knowledge base containing the necessary domain objects was implemented, and default data insertion forms were created in QDMP. Additional dynamics were then added to the data insertion forms to meet the previously determined expectations. All processes initially identified on the prototype were formulated as step-by-step instructions for the QDMP. Technical support for their execution was provided using Qure Designer's built-in functionalities and additional SQL queries and JavaScript scripts.

[12] https://www.quretec.com/

Once the baseline knowledge base version had been implemented, actual data was added to the system, and all workflows were monitored step-by-step in different scenarios. The expectations list was also looked over again, and the system was verified against each one.

5 Knowledge Base Development

The goal of this chapter is to describe the knowledge base (KB) development. The first subchapter introduces the initial prototype. The second subchapter lists the main domain objects that were identified during the prototyping phase. The third subchapter describes the processes that were designed on the prototype and that the knowledge base must offer technical support to. The fourth subchapter lists some further expectations that the knowledge base final implementation must meet.

5.1 Knowledge Base Prototype

The goal of the prototype development was to identify the knowledge base's structure, necessary processes, relevant user groups and other specifications. Actual data from public online sources were added to the prototype to simulate working with the final KB, and based on the experiences gained, the prototype was iteratively analysed and improved. Only the final version of the prototype is described here.

The initial prototype was created using Google Docs and Google Sheets, and its contents were stored in a separate folder that all members of the GenMed team had access to. Since the prototype does not include actual genetic data and encompasses only publicly available information, stricter access rules were not considered necessary. The structure of the prototype is visualized in figure 5.1 below.

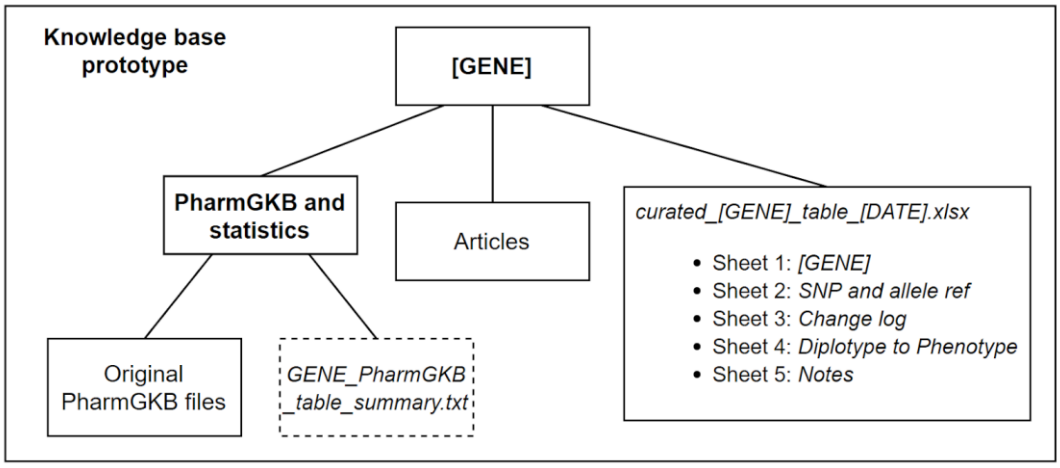

Figure 5-1. Prototype structure.

One folder was created for each pharmacogene of interest. These folders were named after the genes, so for example, *CYP3A5*. Under each gene, there is an additional subfolder with the original star allele definition table and the allele functionality reference table downloaded from PharmGKB. In addition, a script was used to create a small statistical overview of the gene, including, for example, its number of variants, number of star alleles and a list of possible functions. This summary file is not imperative in the context of the knowledge base but can be helpful for research purposes to get a quick overview of the gene. Therefore, its presence in the actual knowledge base implementation is optional.

The main component in each gene's file is the multilevel Google Sheets document, where all relevant information about the variants, star alleles and phenotypes of this gene was gathered in separate tables. The document consists of four sheets.

1) A modified PharmGKB allele definition table. A *function* column from the PharmGKB allele functionality reference table and a row with positions according to GRCh37 have been added. In addition, some rs numbers that were missing in the original table have been found online and added.
2) A table with all information gathered about the variants and star alleles of the gene, along with references to the original sources. Each row represents a variant, and each column provides a unit of information about this variant or the star allele it is related to. This table forms the core of the knowledge base content. All decisions about choosing significant variants for the pharmacogenetic algorithm are made based on the content of this table. Each decision is documented, with the date and the names of the curators who participated in the discussion.
3) A change log is kept for all modifications that are made to the original PharmGKB table.
4) The fourth sheet holds the diplotype to phenotype tables that have been collected from different CPIC guidelines for this gene.
5) The fifth sheet contains both the notes that come along with the original PharmGKB table and notes added by the GenMed pharmacogenetics team.

In addition, all articles that are referenced from the aforementioned *SNP and Allele References* table were collected in the gene's folder for easy access.

5.2 Main Domain Objects

During the prototyping phase, several domain objects were identified that need to be supported by the knowledge base. The list of the main object types together with a brief description is given below. The final data model in its full details is provided in section 6.3.

1) Gene - contains all information that is important in the context of one gene. This object type is an approximate equivalent for the gene folder in the prototype. It has the following subobject types:
    a. Variant - holds variant information, references, information about whether the variant data has been reviewed and whether the variant is important for the PGx device.
    b. Star allele - holds star allele information, references, information about whether the star allele data has been reviewed and whether the star allele is important for the PGx device.
    c. Star allele function - lists all possible star allele functions of the gene.
    d. Phenotypes - lists all possible phenotypes and their definitions of the gene.
2) Reference - represents guidelines and other publications used to gather variant- or star-allele-specific information. Since some articles can contain information about several genes, this needs to be a separate object rather than something gene-specific

3) Drug - currently only states the name of the drug, but additional information may be included in the future, such as drug-specific pharmacogenetic recommendations to the doctor. Since some medicines may be affected by several genes, the drug list cannot be gene-specific and is, therefore, a separate object.
4) External information table - contains information tables from external sources, such as PharmGKB. These tables are mainly used for getting an initial list of variants and star alleles for each gene.

5.3 Main Processes

In addition to the knowledge base specific procedures, some general recommendations for effectively gathering relevant information from the literature were also determined during prototyping. All research should start with the CPIC guidelines as they have been created through meticulous research and are most likely to present only reliable information. Articles that are referenced from CPIC should be read and cited for more detailed information. It might also be a good idea to create an overview of expired or incorrect conclusions that have been reached about the variant or star allele of interest to avoid confusions in the future when encountering incorrect information. If a CPIC guideline does not mention a variant or a star allele or states it to have low or moderate evidence, some additional literature research might be reasonable. If several articles that are newer than the latest CPIC have reached similar conclusions, their findings may be accurate and may be included in the next guideline update. Therefore, it should be inserted into the knowledge base, even though no decisions can usually be made based on them at that stage.

The following processes were identified during the prototyping stage and need to be supported by the KB.

5.3.1 Add/Edit Gene

To add a new gene, the curator must enter the gene’s name and chromosome number. Designated spaces for variants, star alleles and other gene-specific information should be created automatically and must be in the same format for each gene. When needed, the gene's name and chromosome number can be edited. Each modification must be trackable via a change history.

5.3.2 Add/Edit External Information Table

External information tables are a starting point for data curation and need to be kept close to the curated data. Such tables currently include PharmGKB star allele definition tables and allele functionality reference tables but might not be limited to them in future versions of the knowledge base. These tables are typically gene-specific, but as this might also change in the future, it should not be limited in the knowledge base. For each new table that is uploaded to the system, the curator must specify the gene that the table is related to, the type of the table and the date on which the table was downloaded from the original source. When needed, all added information can be modified, and a different file can be uploaded. Each modification must be trackable via a change history.

5.3.3 Add/Edit Star Allele Function

Each star allele is linked to a specific function. The list of possible functions is gene-specific and predefined. In order to add a new function, the correct gene must already be added to the knowledge base. The function could be written next to each star allele in free text form, but that would leave more room for error. Therefore, possible star allele functions must be added to a separate list from which they can later be selected. The curator must only enter the name of the function to add it. When needed, the name can be edited. Each modification must be trackable via a change history.

5.3.4 Add/Edit Phenotype

Phenotypes are gene-specific, and their exact names can also vary for different drugs due to changes in naming conventions. To add a new one to the knowledge base, the curator must insert the name of the phenotype, and it will be added to the gene-specific list. When needed, the name can be edited. Each modification must be trackable via a change history.

5.3.5 Add/Edit Phenotype Definition

Each phenotype definition consists of two star allele functions, and there can be one or several definitions per phenotype. In order to add a new definition, a phenotype and two star allele functions must be selected from predefined lists. Therefore, before adding a new phenotype definition, the related phenotype and star allele functions must be present in the knowledge base already. When needed, the definition can be edited. Each modification must be trackable via a change history.

5.3.6 Add/Edit Drug

To add a new drug to the knowledge base, the curator must insert the name of the drug, and it will be added to a list that is accessible from all genes. In future developments, it might be necessary to include more information about each drug, but currently, the name is sufficient and can be used to link the drug to related genes and publications. When needed, the name can be edited. Each modification must be trackable via a change history.

5.3.7 Add/Edit Reference

Figure 5.2 shows the process of adding a new reference (publication, link to web source, etc.). The curator must specify details such as the publication year and the author and, if applicable, also upload the publication file to the system. In addition, the drugs that the publication covers should be added by selecting them from a predefined list.

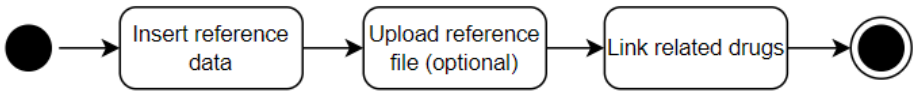

Figure 5.2. Add a reference to the knowledge base.

When needed, all data can be edited, and different files can be uploaded. Each modification must

be trackable via a change history.

5.3.8 Add/Edit Variant

In order to add a new variant to the knowledge base, the correct gene and related articles must already be inserted. The process is described in figure 5.3 below. The curator must first navigate to the variants section under the correct gene. Then the variant data, such as its rs-number and its positions according to different references, must be inserted. The international rs-numbers are not published for all variants, and sometimes the same variant is published with different labels. Therefore, to reduce ambiguity, the curator has to assign a unique variant ID for each variant in the knowledge base (if an rs-number exists, it is used as the variant ID). The possible alleles of the variant must also be specified. In addition, articles covering the variant must be linked to it by selecting them from among the references.

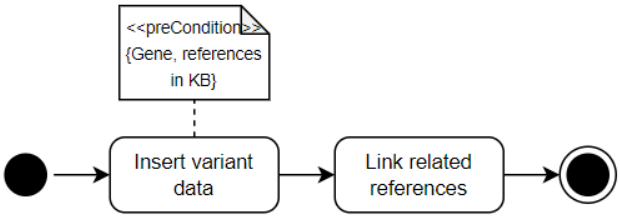

Figure 5.3. Add a variant to the knowledge base.

It should also be possible to add a comment when necessary. When needed, all data can be edited. Each modification must be trackable via a change history.

5.3.9 Add/Edit Star Allele

The precondition to adding a new star allele to the knowledge base (figure 5.4) is that the correct gene, related references and defining variants must already be inserted. Then the star allele data can be inserted, and associated references and alleles can be linked.

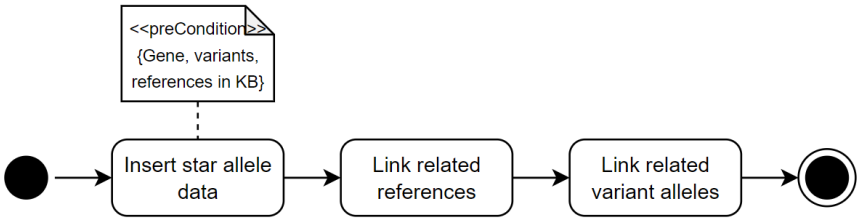

Figure 5.4. Add star allele to the knowledge base.

It should also be possible to add a comment if necessary. When needed, all data can be edited. Each modification must be trackable via a change history.

5.3.10 Choose Important Star Alleles and Variants

The goal of the KB is to provide only correct and relevant input to the pharmacogenetic algorithm - therefore, identifying which star alleles and variants have enough solid evidence to justify using them in a medical device is an essential step. The decisions must always be based on the same

criteria to ensure uniformity. Deciding whether a variant should be taken into consideration by the algorithm consists of two parts:

1) Determine the importance of the star allele that this variant's allele is part of.
2) Determine whether the variant's allele is a defining one for the related important star allele.

If the related star allele is important (*importantForAlgorithm = true*) and the variant's allele is a part of its definition, then this variant should be included in the data that will be exported to the algorithm (*usedByAlgorithm = true*). However, if the star allele is not considered important or if the variant's allele is not a defining one in its definition, there is no reason to include this variant in the list of exportable data since no reliable decisions can be made on it. This concept is visualised in figure 5.5 below.

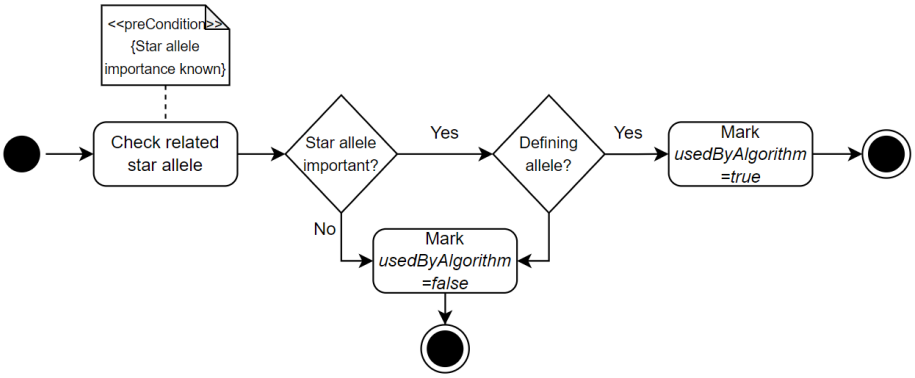

Figure 5.5. Determine the importance of a variant.

Determining whether a star allele is essential depends on multiple factors concerning its function and the evidence levels behind it. There were long discussions about the exact requirements for a star allele in order to label it as important in the GenMed project PGx team, but the final decision criteria for marking a star allele as important are the following (all must be met):

1) The function must not be uncertain or unknown, according to PharmGKB and CPIC. Unknown or uncertain function means that the star allele has been detected in some studies, but there is very little or no evidence to determine its pharmacogenetic effect.
2) The function must not be normal unless it is a wild type star allele. The algorithm is only supposed to detect deviations from normalcy, and differentiating between different star alleles with normal functions is not considered important.
3) The function must be confirmed in either the main article of a CPIC guideline or in the guideline's supplementary with high evidence. This ensures that the final clinical recommendations derived from the algorithm's output are based only on the most reliable information available at the time.

A visualisation of the decision criteria is provided below in figure 5.6.

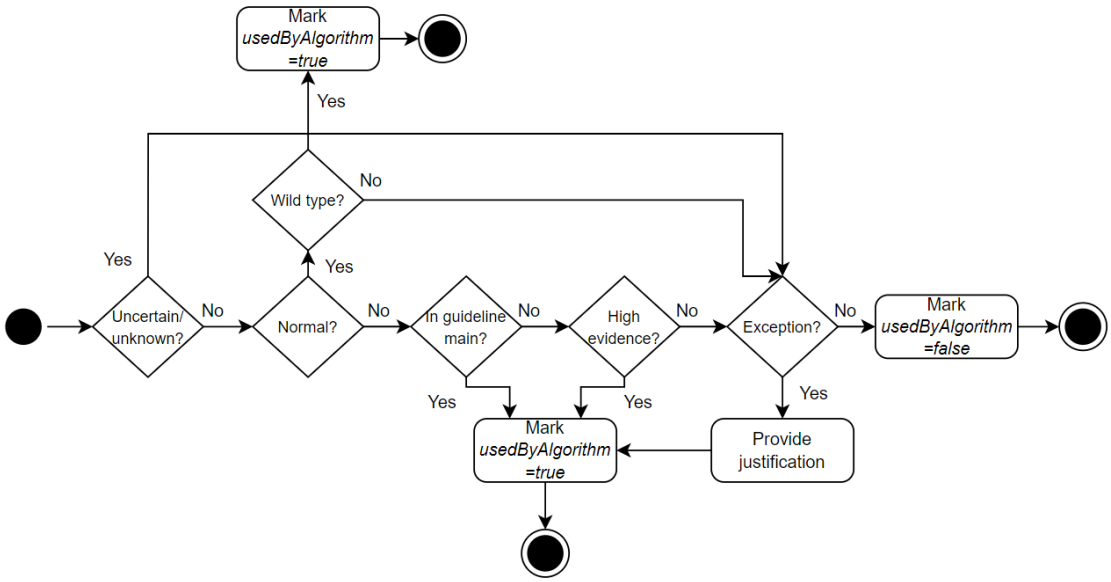

Figure 5.6. Determine the importance of a star allele.

For exceptional cases, there is also a possibility to make an exception and bypass these criteria. Still, this option must be used with great caution, and a sufficient explanation with references to reliable sources must be provided. An example of a situation where this would be justified is when a CPIC guideline for a certain gene has not been updated for a longer time, and a large number of reliable studies have since reached similar conclusions about the star allele's functionality. In such cases, it might be inferred that the star allele will be included in the next update of the CPIC guideline. However, discussing this decision with experts in pharmacogenetics is obligatory

5.3.11 Review Gene

Reviewing a gene's data consists of several steps that must be completed in the correct order to ensure that every important aspect is covered. In order to mark an object as reviewed, all of its subobjects must be reviewed. Figure 5.7 below visualises this concept.

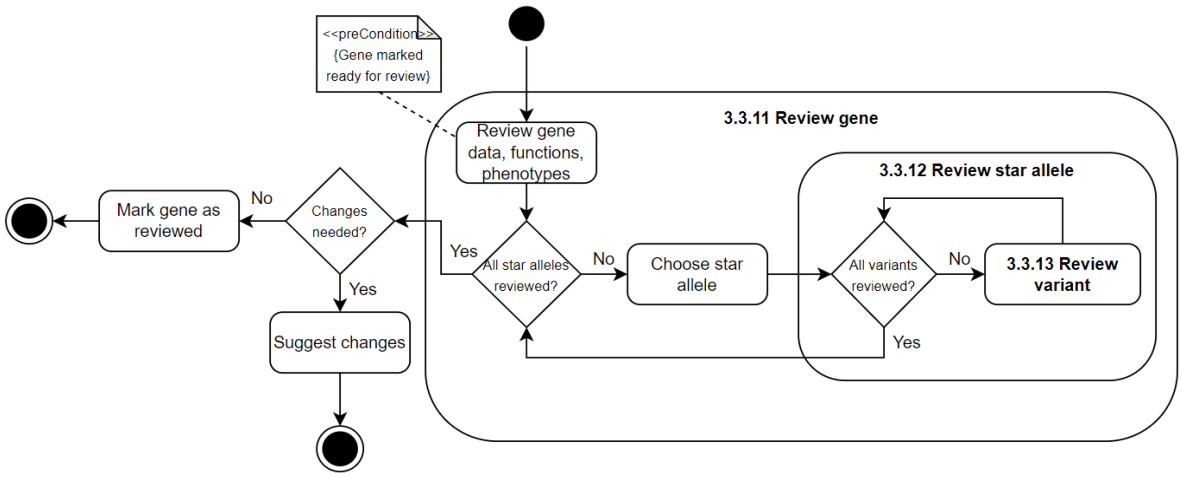

Figure 5.7. Gene review process.

During the review process, modifying or adding data must be prohibited. When reviewing a gene, the gene data, such as the name and the chromosome, must be checked first. The next step is checking whether all star alleles of this gene have been reviewed. If all star alleles have been reviewed, then the gene can also be marked as reviewed. Otherwise, the star allele review process shall be initiated. In order to mark a star allele as reviewed, all variants whose alleles are included in the star allele definition must also be examined. Therefore, the variant review process is initiated for each variant. Both these processes are described in more detail in the sections below.

5.3.12 Review Star Allele

Once a star allele is ready to be reviewed, its references and function are checked first. For each reference, it is checked if the article has been cited or interpreted correctly. Then the star allele function is checked, and the alleles related to this star allele are reviewed. It is also determined whether the variants corresponding to these alleles have been reviewed. If not, the variant review process is initialised. Once all the variants have been reviewed, the star allele can be marked as reviewed as well. The date on which the review was completed should also be included. The process is visualised in figure 5.8.

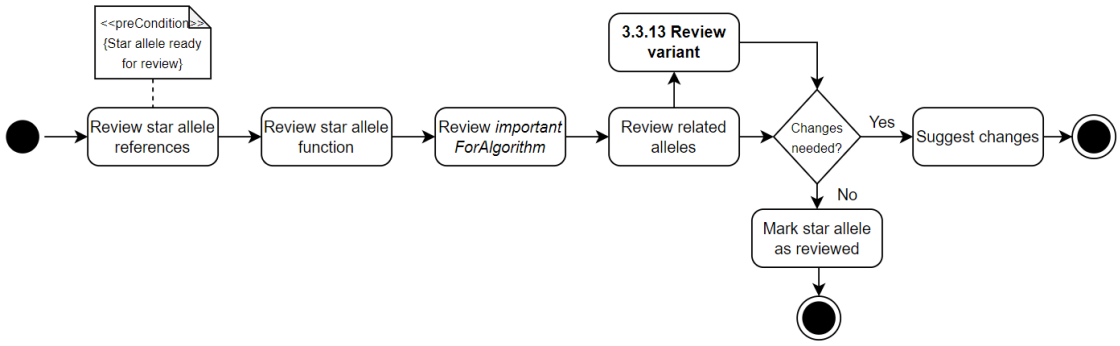

Figure 5.8. Star allele review process.

If any inconsistencies or errors are found in the data, the review status should not be confirmed, and the problems should be pointed out to other KB curators.

5.3.13 Review Variant

Figure 5.9 shows the process of reviewing a variant.

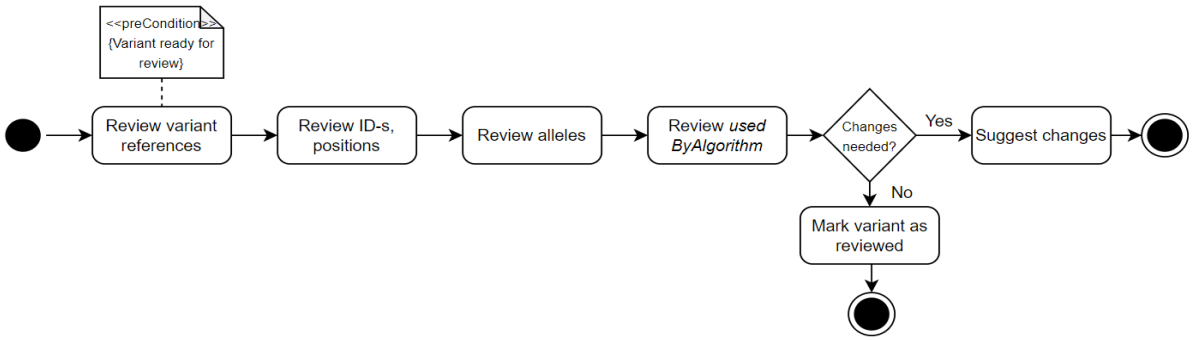

Figure 5.9. Variant review process.

For each variant, its references are reviewed with the same goal as for the star alleles. Then rs-number, name, positions according to different positions and related alleles are checked. As the final step, the decision in *usedByAgorithm* is reviewed. Once the review is complete, the variant is marked as reviewed, and the review date is included. If any mistakes are found in the data, the review status is not confirmed, and the errors are discussed with other curators.

5.3.14 Export Data for the Pharmacogenetic Device

The format of the data that needs to be exported to the algorithm was agreed upon during the prototyping phase. The required tables with their obligatory columns are shown below in figure 5.10.

Figure 5.10. Exportable tables for the pharmacogenetic device.

The variant table contains all variants that the PGx algorithm uses to determine a patient's phenotype. For each variant, its gene name, chromosome, knowledge base specific variant ID, rs-number (if exists), position according to HGCh37 and a list of its possible alleles is given. Based on this table, the algorithm can find necessary variants in the patient's genetic data.

The star allele table specifies the gene, star allele name and function of each significant star allele. In addition, two boolean values are used to state whether the star allele is a wild type or a no function star allele. These two types are treated differently from other star alleles in the pharmacogenetic algorithm and need to be easily identifiable.

The star allele definition table shows how different variants' alleles form star alleles. The attributes include the gene, star allele name, variant ID, and that variant's specific allele.

The function to phenotype table represents possible phenotype definitions for a gene. It includes the gene's name, phenotype and the two star allele functions that define this phenotype.

Some critical aspects needed to be considered when designing the process for data export. First of all, to improve the system's usability, the exportable tables should be accessible at all times in case some details need to be checked. Therefore, a script that collects the information and creates the tables only when executed is not the optimal solution.

Secondly, only those variants and star alleles that have been marked as important (as described in 5.3.10) should be included in the exported tables. Thirdly, information that has not yet been reviewed should not be exported as it may contain errors. A warning message should be displayed whenever a variant, star allele or gene has been marked as important but has not been reviewed.

5.3.15 Delete a Domain Object

On rare occasions, such as when a variant's or star allele's data has been inserted under the wrong gene, a publication entry is made twice, etc., it must be possible to delete entire data entries to avoid leaving wrong associations in the knowledge base. This process must be approached very carefully, and the access to the rights for executing this process must be limited.

5.4 Additional Expectations

The prototype was sufficient for identifying the knowledge base curation processes and provided some initial input for the PGx algorithm prototype. However, several aspects essential for the KB were not supported by Google Sheet-based data management. By analysing these constraints and considering ways in which they could be solved, the following expectations were determined for the final knowledge base solution:

1) The knowledge base must have a graphical user interface.
2) Access to the knowledge base must be limited to authenticated users only.
3) It must be possible to have different user roles with different access rights:
    a. administrator - manages the knowledge base settings, assigns user roles;
    b. curator - the default role for knowledge base users, can add and edit information in the knowledge base;
    c. reviewer - can edit the review statuses but no other data elements;
    d. delete - a role temporarily assigned to curators for deleting data elements.
4) Each knowledge base user (curator) must be able to access data with different rights depending on the task they want to complete.
5) An empty information table in a predefined format must be created automatically when a new gene, reference, database table or any of their subobject types is entered into the knowledge base.
6) Different processes must be isolated from each other.
7) It must be possible to mark attributes as unique.
8) It must be possible to mark some object attributes as mandatory.
9) If mandatory attributes are not filled in, an error must be shown.
10) Data cannot be saved when mandatory attributes are missing.
11) It must be possible to view only a certain subset of information when needed.
12) Each column of each information table must be searchable.
13) It must be possible to open and edit only one row at a time in each table.
14) Changing data directly in information tables must be prohibited.
15) Each modification must be automatically linked to the curator who made it.
16) Each modification must be automatically linked to the date on which it was made.
17) A change log with names and dates must be available for each cell in any information table.
18) It must be possible to add a comment for each change to each cell.
19) The review status of each gene, variant and star allele must be visible.
20) Marking a gene as reviewed must be prohibited if any of its variants or star alleles are not reviewed.
21) For each star allele, the review status of the variants of its related alleles must be visible.

22) If an object is modified, its review status must automatically switch to *not reviewed*.
23) If any subobject of an object is modified, the object's status must automatically switch to *not reviewed.*
24) Identifying important variants and star alleles must be supported with a script.
25) Variants and star alleles that have been marked as important must be selected for export automatically.
26) Exportable tables must be kept as separate objects.
27) Exporting data that has not been reviewed must be prohibited.
28) The exported data must be machine-readable.
29) The exportable data cannot be modified manually and should not include any data in free text form.
30) The exported data must be in a format that is suitable for the pharmacogenetic algorithm.
31) Adding duplicates must be prohibited for every gene, variant, star allele, drug and reference.
32) It must be possible to link one or several references to each variant and star allele.
33) It must be possible to link one or several drugs to each reference.
34) It must be possible to link one or several variant alleles to each star allele.
35) Gene-specific information must be isolated from other genes.
36) It must be possible to access the list of drugs in the system from under every gene.
37) It must be possible to add multiple comments for each variant and star allele.

This list of expectations is not exhaustive but was deemed to cover the most important aspects and functionalities of the KB, along with the processes described in 5.3. Some of those expectations could have been met by improving the prototype with more advanced features, but the solution would still have been clumsy. Therefore, it was determined that a more sophisticated and dynamic data management platform is required to implement a secure, sustainable, functional and user-friendly knowledge base. In the following chapter, the final KB implementation using the Qure Data Management Platform is described.

6 Knowledge Base Implementation Using Qure Data Management Platform

This chapter gives an overview of the KB implementation using the Qure Data Management Platform (QDMP) and provides illustrations/snapshots of the structure and user interface in its current state. The first subchapter introduces the QDMP and its component. The second subchapter describes how the design of the knowledge base was implemented in the Qure Designer program. The third subchapter describes the resulting object model. The fourth subchapter describes how the process support was ensured. In the fifth subchapter, the implementation is verified against the processes and expectations defined in sections 5.3 and 5.4. The sixth subchapter gives a brief overview of the data inserted into the final implementation of the knowledge base.

6.1 Qure Data Management Platform

Qure Data Management Platform (QDMP) is a suite of software components for collecting data and managing databases. Qure Designer provides the user with a graphical user interface through which the database structure can be developed and edited. In addition, several views with different functionalities and complexity levels can be constructed on each data model. These views are called questionnaires, and they can be used to insert and edit data on the server-side. It is also possible to modify the dynamics of the questionnaires by adding the system's built-in conflicts or JavaScript-based scripts to them. All data models, questionnaires and scripts are uploaded to the Qure Server as XML files. Data access rights and other project settings can also be managed from the server. Qure browser is used to create functional web applications from the data models, questionnaires and scripts defined in the Qure Designer. All database contents can be queried via a graphical user interface offered by Qure Dataview. Dataview supports queries, such as filtering and aggregation and can output resulting data in different formats, such as HTML or CSV.

Several reasons justify using QDMP for implementing the knowledge base. First of all, it has been developed specifically for working with biological and clinical data. It was initially designed for the Estonian Biobank and is currently used for storing the Estonian Genome Project database and several national health registries, such as the medical birth registry, the cancer registry and the causes of death registry. Since the system is designated to work with clinical data, it offers high security. It ensures that inserted data will not change or disappear and that only authenticated users can access it. In addition, it provides an audit trail, where each value's change history is easily traceable.

Secondly, the use of a graphical user interface makes the system very user-friendly. The Qure Designer allows for quick and easy modifications to the data model and the questionnaires dynamics whenever necessary. Thirdly, QDMP was developed in Estonia. Therefore, Estonian language support is available for the system.

The academic licence of QDMP was obtained from Quretec for this thesis. Qure DMP was deployed to High Performance Computing Centre at pgx.cs.ut.ee by the team of HPC.

6.2 Designing the Knowledge Base in Qure Designer

The first step in a Qure Designer project is creating an object model with all necessary tables (called *object types*) and attributes. QDPM requires a strict hierarchical model, where each table is either a root level table or a child table of other tables. No many-to-many relations are allowed in the system. Some tables that require gathering data from different tables (such as for exporting) are implemented as database views. For each attribute, its type must be specified (e.g. *boolean*, *timestamp*, *integer*).

Based on this object model, several questionnaires can be generated. Each table in a questionnaire represents an object type in the object model, and each question represents an attribute. Questions can have several types, such as *question* for free text answers, *check* for boolean answers, *file* for uploading files etc. For each question, the type must be in concordance with the object model's corresponding attribute. For example, if the type of the object model's attribute is *boolean*, then the corresponding question must be of type *check*, which results in either a checkbox or a *yes/no* question in the uploaded questionnaire form. A basic questionnaire can be generated automatically, but usually, a more complex structure is required. New pages can be added, and the questionnaire can be restructured by dragging its components to correct places. In addition, scripts and conflicts can be added to the questionnaires to make them dynamic. Conflicts are simple predefined functions with a limited set of possible actions and can only be used to describe the interaction between two fields. For example, it is possible to check whether an answer to a particular question is *true* or *false* and hide or show another question based on this. Scripts must be written from scratch and can be used to describe more complex interactions between several fields. Once a script is written, its output value can be assigned to a specific field in the questionnaire and the fields for getting input values can be specified. An example of working with Qure Designer is given below in figure 6.1.

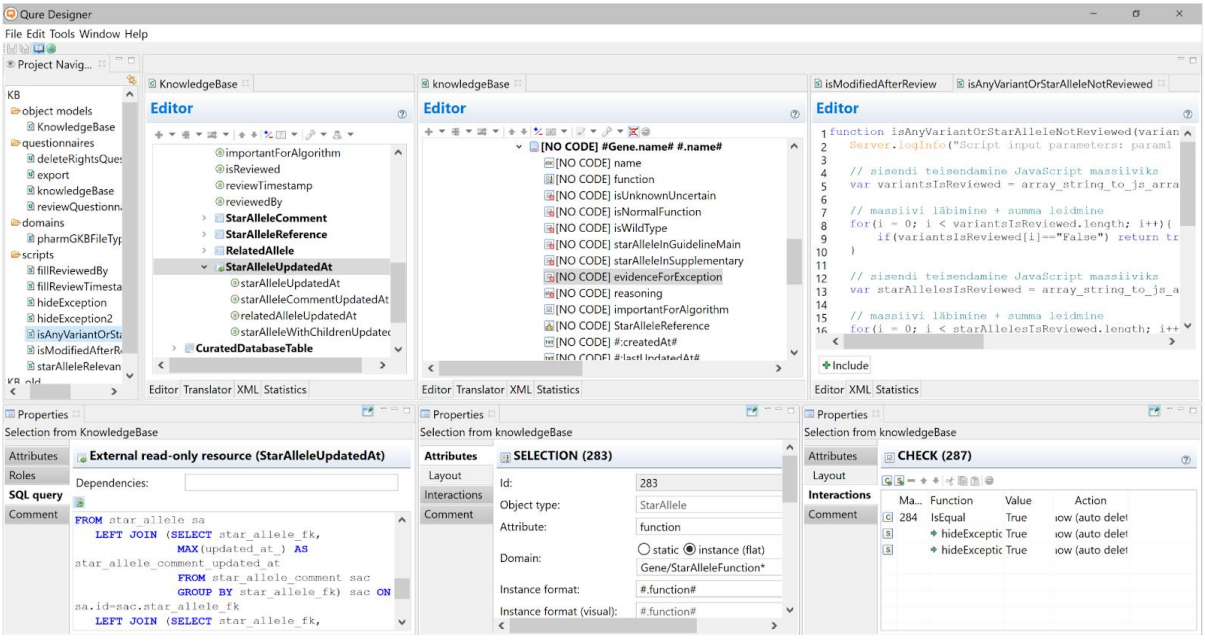

Figure 6.1. Qure Designer window with several *Editor* and *Properties* windows.

The components of the project can be seen and navigated in the *Project Navigator* window in the top left corner. All object models, questionnaires, custom domains and scripts are listed there. In addition, several *Editor* and *Properties* windows can be opened to work on several object models or questionnaires simultaneously. In figure 6.1, three *Editor* windows are opened with their corresponding *Properties* windows to best illustrate different functionalities available in Qure Designer.

The *Editor* windows are used for editing data models, questionnaires, domains and scripts. When any component (object type/attribute/question) of an object model or questionnaire is clicked, a *Properties* window opens for it, where details about this component can be specified. For example, the leftmost *Properties* window shows an SQL query used to create a read-only database view in the object model. Such views help collect information from several places in the knowledge base and represent it compactly in one place. The query in the figure collects the updating dates of all related alleles, references and comments of a star allele to facilitate checking if they have been changed after the last review.

The second *Properties* window shows the attributes of the *function* field under the *StarAllele* object type. Things like the answers domain and question's text in the knowledge base can be specified in this window. The domain for the *function* attribute is set as an *instance of Gene/StarAllele-Funtion**, which means that the answer to that question can only be selected from among the star allele functions inserted to the KB. In addition, it is possible to mark any attribute as mandatory, read-only, commentable, etc. The third *Properties* window shows the interactions of the *evidenceForException* attribute of the *StarAllele* object type. From this section, conflicts and scripts can be added.

Once the project design is complete, it can be uploaded to the server and accessed via a web browser. In the context of this thesis, one object model and four questionnaires with different rights and functionalities were created. Having multiple questionnaires helped to ensure that different processes are isolated from each other. Inside of them, 8 SQL database views and 9 Javascript scripts were created. Conflicts were applied to several fields across the questionnaires.

The resulting Qure Designer project containing the data model, questionnaires, scripts, and domains for the knowledge base is available on GitHub: https://github.com/idamaria1/GenMedKB.

6.3 Data Model Overview

All domain objects derived from the prototype were implemented in Qure Designer, and the proposed object model is shown in figure 6.2 below.

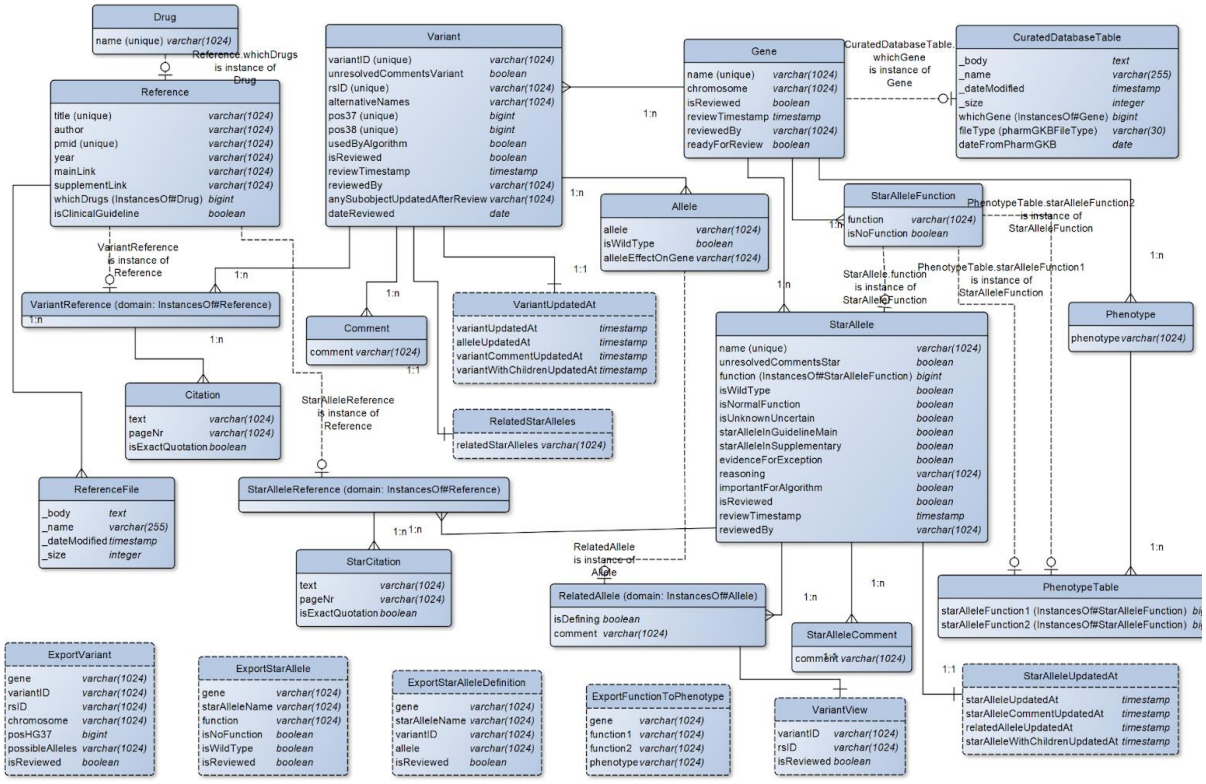

Figure 6.2. Object model designed in Qure Designer.

Below, some of the most important relations in the data model are explained and justified. Object types are marked with solid lines and database views with dashed lines. One of the root objects in the model is *Drug*, which lists the names of drugs that are affected by the pharmacogenes listed in the knowledge base.

The *Reference* object type consists of all publications and other information sources used to collect information about the genes in the knowledge base. The *Reference* object type has a one-to-many relationship with the *ReferenceFile* subobject type, meaning that several files can be uploaded for each reference. This opportunity is mainly used for CPIC guidelines that consist of the main article and a supplementary file. The references are linked to the star alleles and variants and can be accessed from every gene in the KB.

The most significant root level table corresponds to the gene domain object. The *Gene*'s one-to-many relationships show that several variants, star alleles etc., can be added to each one. In addition, the attribute *whichGene* in the *CuratedDatabaseTable* object is an instance of the *Gene* object, which allows connecting relevant external information tables to one correct gene.

The *Variant* table's one-to-many relationships represent that a variant can have several alleles and comments related to it, and several publications or other sources can be referenced. Each instance of the *Reference* object type has a one-to-many relationship with a *Citation* object, meaning that

multiple quotes can be brought out from each article.

Each *StarAllele* can include multiple alleles in its description, have several comments about it, and several references can be used as a source for information. Each reference is an instance of the *Reference* object type and can also include several citations. The *RelatedAllele* object type is an instance of *Allele* under the *Variant* table, connecting relevant star alleles and their variants. To see which variants are related to each star allele, the *VariantView* table was implemented as a database view. The function attribute in the *StarAllele* table is an instance of the *StarAlleleFunction* object type, which is used to list possible functions for a specific gene's star alleles.

The *Phenotype* object type is only used for listing the names of possible phenotypes for the gene. Each *Phenotype* table can include several tables of the *PhenotypeTable* object type, which present all possible definitions for that specific phenotype. In each new definition, the two star allele functions can be selected from a list that is defined under the *StarAlleleFunction* table.

In addition, the four exportable tables (*ExportStarAllele, ExportFunctionToPhenotyp, ExportStarAlleleDefinition* and *ExportVariant*) are implemented as database views using SQL queries. These tables collect relevant attributes from several tables and present them in a suitable format for the pharmacogenetic algorithm.

6.4 Process Support

This subchapter presents how all processes described in section 5.3 were implemented in QDMP by using different functionalities of questionnaires. Having multiple questionnaires helped to isolate different workflows from each other and simplified user role assignment. A separate *reviewer* role was no longer necessary due to a separate questionnaire for reviewing. This way, the knowledge base user will not have to be switched between different roles that often. A distinct role is still required for deleting operations since this activity is only necessary in rare cases and must not be done carelessly.

6.4.1 Questionnaire for Main Data Editing Process

The first questionnaire is used to insert and edit the data, covering the processes from 5.3.1 to 5.3.9. The questionnaire contains four separate tabs: *Genes, References, Drugs* and *Database tables*. When a tab is opened, the tabs of all of its subobject types become visible. Qure Designer offers different layouts, but this solution makes the knowledge base easily navigable since the entire path can be tracked level by level.

Here, this questionnaire's layout and main functionalities are demonstrated with the example of the *StarAllele* object type, but the description applies to other object types as well. Note that only a subset of data is shown in the following figures to fit all important features into screenshots.

Figure 6.3 below shows the table view of the *StarAllele* object type, where some star alleles of the CYP2C9 gene are listed. The data can be sorted or searched by any of the columns. All data in the table view is in read-only format to protect it from accidental modifications. The review status of each star allele is also displayed in the table, but it can only be manually modified from a different questionnaire. However, a combination of scripts has been added to the *isReviewed* checkbox that

automatically sets its value to *false* when any modifications are made to the star allele or any of its child tables. These scripts compare the modification dates of each relevant subobject type to the latest review date. Subobject types' modification dates are gathered via an SQL statement. The same logic is used for each variant and each gene as well.

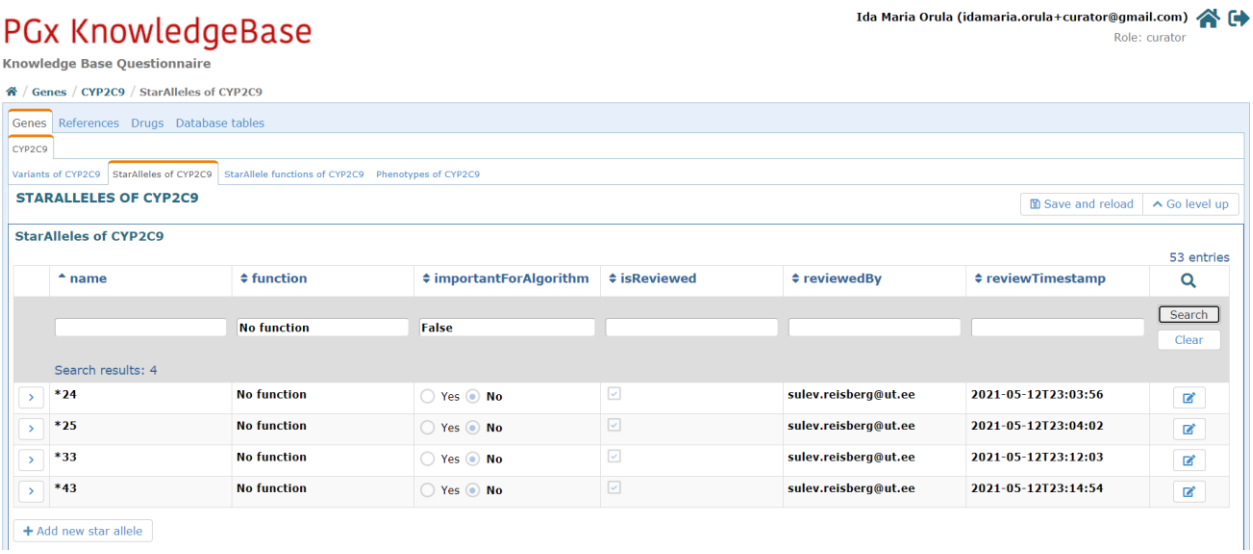

Figure 6.3. Table view of CP2C9 star alleles with the search bar activated.

To edit a star allele's information (other than the review status), a modal edit window can be accessed by clicking on the icon at the end of the star allele's row. A separate modal window must be opened by clicking the *Add new star allele* button below the table for adding a new star allele. These windows, along with the effects of several constraints applied to the questionnaire in the Qure Designer's *Properties* window, are shown in figure 6.4.

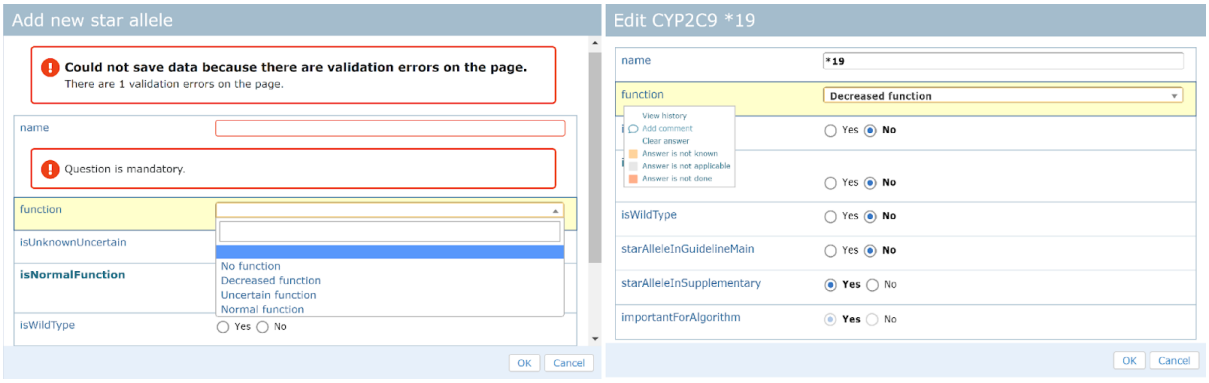

Figure 6.4. Left: Modal add window. **Right:** Modal edit window.

For each star allele, its name and function are required attributes. When trying to save a new star allele without entering its name, an error message is displayed, and the data entry cannot be saved. In addition, each star allele's name must be unique in the context of the gene. Therefore, if a duplicate star allele name is inserted, a similar error message appears, stating that the value must be unique.

When the question's domain has been limited, the answers can be selected from a predefined list. For example, the star allele's function can only be chosen from a drop-down list of possible functions, inserted under the *StarAllele functions of [GENE]* tab (visible in figure 6.3). In addition, each entered value can be separately commented on. Those comments will appear alongside the entered value in this field's change history (figure 6.5), which is accessed by double-clicking the field in the table view. The comments can be written in both the adding and the editing windows.

History: function

Date	Changer	Value	Comment	Warning
2021-05-09 19:18:46	ida.maria.orula@ut.ee	**Decreased function**	PharmGKB and CPIC 2014 stated uncertain function, but CPIC 2020 suggested decreased function with high evidence. New reliable studies had been published by 2020, which confirmed decreased function.	
2021-05-09 19:16:16	ida.maria.orula@ut.ee	**No function**		

Figure 6.5. Change history for *function* attribute of a star allele.

Process 5.3.10 (*Choose relevant star alleles and variants*) is also covered in the first questionnaire. While adding or editing a new star allele, its importance to the PGx device is also determined. A combination of conflicts and scripts added to the questionnaire prompt the curator with a set of *yes/no* questions (figure 6.4). Based on these answers, the importance of the star allele is determined automatically (*importantForAlgorithm*). The exact set of questions depends on the curator's answers. In the example in figure 6.4, five questions are asked before making the decision. In figure 6.6 below, a different subset of questions is visible due to how the first questions were answered.

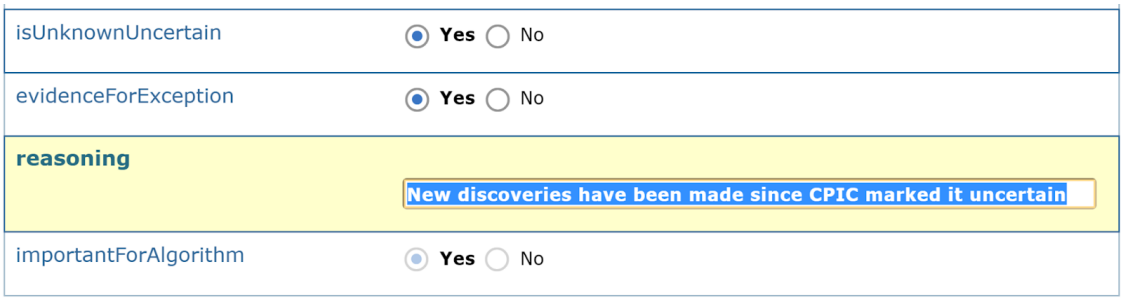

Figure 6.6. An example of determining the importance of a star allele.

Typically, a star allele with uncertain function would not be important for the algorithm, but in this case, an exception was made, and an explanation was given. The exact functioning of the added conflicts and scripts follows the logic described under process 5.3.10.

As could be seen in figure 6.3, only the final decision, *importantForAlgorithm*, is displayed in the table. To keep tables clear and concise, only a subset of information is displayed in them. Each star allele can be opened by clicking the arrow at the beginning of each row in the table view. This opens a detailed read-only information page (a small section of which is shown in figure 6.7 below) and provides access to the star allele's child tables: related alleles, comments, and references.

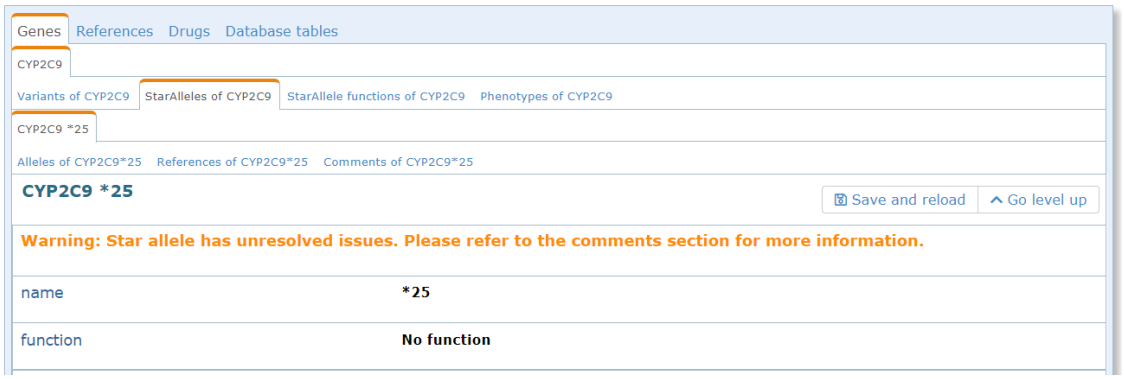

Figure 6.7. A small section if the star allele's detailed view page.

The detailed page shows the curator's names who have modified this star allele, along with corresponding dates, as well as all questions used for determining the star allele's importance. In addition, if there are any details that the curator wants to discuss or bring other curators' attention to, a boolean variable can be set to *true* in the star allele's editing window, which will display a warning in the star allele's detailed view page. The comments section can be used to provide more details. Once the issues have been resolved, the boolean variable can be set to *false*, and the warning will disappear. The same functionality has been added to the *Variant* object type as well.

Once the importance of the star allele has been determined, the next step is to mark its defining alleles' variants as important to complete the execution of process 5.3.10. For this, relevant alleles must be selected under the *Alleles of [STARALLELE]* tab (figure 6.8).

Figure 6.8. Left: Selecting related alleles. **Right:** Marking defining alleles.

The related alleles can be selected from a multiple-choice list, consisting of all alleles inserted under the variants in the *Variants of [GENE]* tab. Once all alleles relevant to this star allele have been selected, the defining ones must be marked. For each defining allele, its variant's *usedByAlgorithm* status is marked as *yes*.

6.4.2 Questionnaire for reviewing

The second questionnaire is used for reviewing the inserted data (covers processes 5.3.11-5.3.13). The general structure matches the structure of the main questionnaire, but there are some differences. Since modifying data during the review process is prohibited, all data is in read-only format and no separate adding or editing windows are used. The table view and detailed view page are the same as in the main questionnaire. The functionality of adding new rows has been disabled for each table, apart from the tables with variant and star allele comments, where modifications may be suggested when necessary. It has been decided that since the number of knowledge base curators is very small and they communicate regularly, a notification system is not required for suggested changes.

During the review process, the reviewer has to confirm the correctness of data on three different levels: *Gene*, *StarAllele* and *Variant*. Scripts have been used to ensure the correct execution of the review processes. First of all, a gene can only be marked as reviewed if all of its star alleles and variants have been reviewed. Figure 6.9 below shows the warning message that accompanies the disabled *isReviewed* checkbox in the detailed view page of a gene.

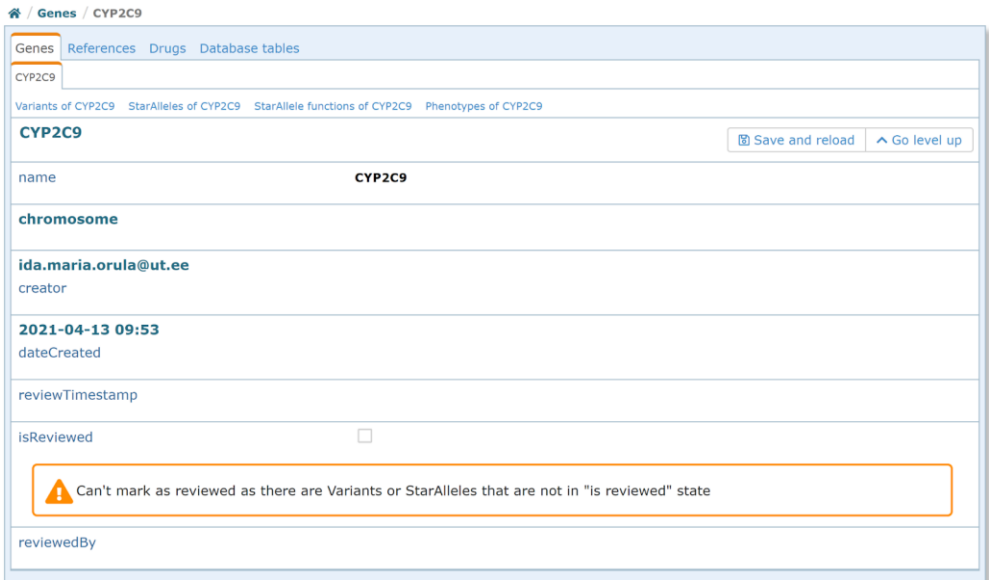

Figure 6.9. Gene detailed view page with a warning and disabled *isReviewed* checkbox.

In the star alleles table, the reviewer can see the review statuses of each star allele and focus on the ones that have not been reviewed. In addition to verifying that the star allele's data has been inserted correctly, the related alleles' variants must also be reviewed. To keep track of this, the *is variant reviewed* attribute from an SQL based view has been added to the *Alleles of [STARALLELE]* table. This can be seen in figure 6.10 below.

Figure 6.10. Related alleles of a star allele with review status.

Once all variants and star alleles of a gene are in the reviewed state and the correctness of the rest of the gene-specific data (star allele functions, phenotypes) has been checked as well, the gene can be marked as reviewed. This results in a notification (figure 6.11) on the gene's detailed page in both the main and the review questionnaire.

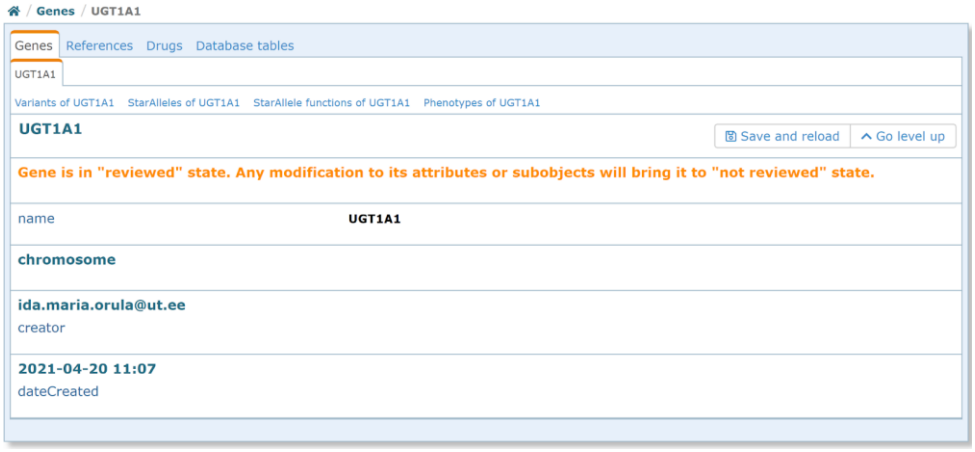

Figure 6.11. Gene's detailed view page with a notification about its review state.

Once all genes have been reviewed, the review process is complete, and the questionnaire may be closed. All information about the reviewer, the review statuses and added comments are visible both from this and the main questionnaire. If reviewers find any issues with the data, they can write it to the comments section and raise the warning shown above in section 6.4.1 (figure 6.7).

6.4.3 Questionnaire for Delete Operations

The third questionnaire allows data entries to be deleted. All other modifications are prohibited, and only the attributes required for correctly identifying the data entry are displayed. For example, for each variant, only its rs-number and knowledge base specific variant id are shown.

This questionnaire is used in rare circumstances such as incorrect data entry under the wrong gene, and a user role must be set temporarily to *deleteData* for conducting this action. Users with that role cannot access any other questionnaire to prevent keeping the role unchanged after deleting tasks.

Figure 6.12 below shows the phenotypes definition page in the deleting questionnaire. An incorrectly inserted phenotype definition must be deleted. A confirmation is asked before completing the delete operation.

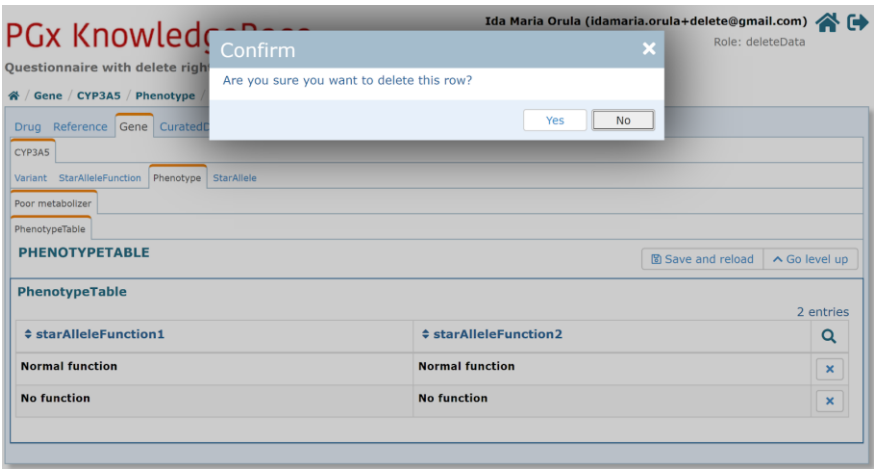

Figure 6.12. Deleting a data entry.

Important things to note here are that the *Add new definition* button is not available below the table, and the *edit* button at the end of each row has been replaced by a *delete* button. The specifically assigned user role *deleteData* is visible in the top right corner of the image. Before any data entry is deleted, the decision must be confirmed in a pop-up window.

6.4.4 Questionnaire for Exporting

The fourth questionnaire provides access to the four exportable tables (format presented in figure 5.10 and in the data model in figure 6.2). Figure 6.13 shows a few example rows of one of the exportable tables and the *Download XLSX* button at the bottom of the page.

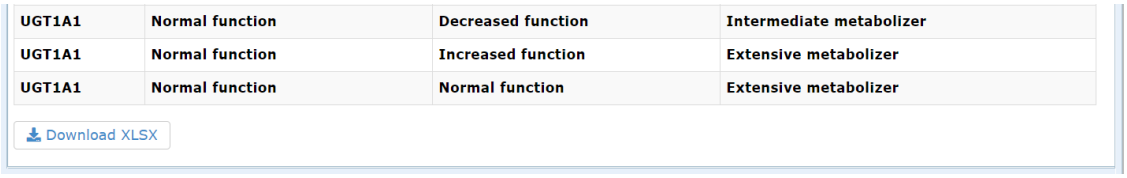

Figure 6.13. Option to download data as *XLSX* files in the export questionnaire.

All tables are implemented as SQL query-based database views and cannot be modified manually. All tables can be downloaded as Microsoft Excel (XLSX) files whenever needed.

A special first page was added to the exporting questionnaire for running different checks to ensure that the data is complete (figure 6.14).

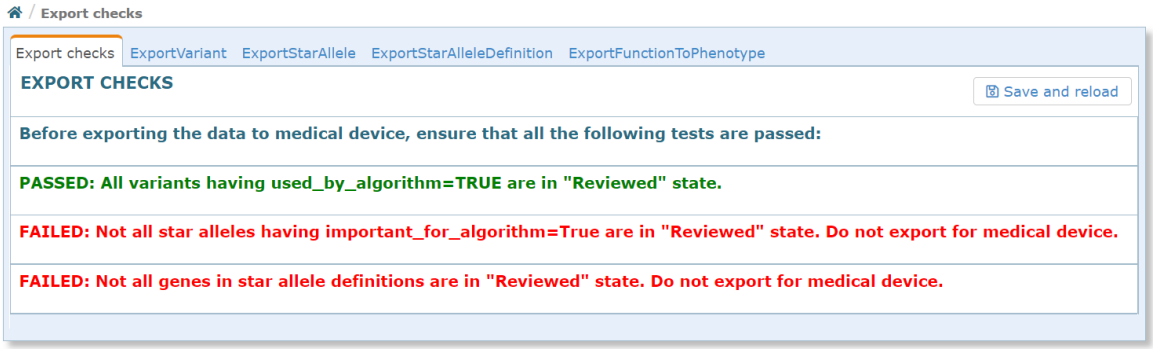

Figure 6.14. Export checks page.

Currently, it checks for the review statuses of the variants, star alleles and genes included in the exportable data, but additional checks can be added in the future.

6.5 Verification

This section shows how the proposed implementation satisfies the expectations set for it as described in section 3.4 and how it supports the processes identified in section 3.3. For verification, all processes were followed through step-by-step. The knowledge base implementation was compared against the list of expectations when applicable (e.g. expectations regarding reviewing were verified while executing the review process).

6.5.1 Verification of Process Support

Table 6.1 below lists all the processes identified in section 5.3 and explains how the processes are supported by the implementation described in the previous chapter.

Table 6-1. Verification of Process Support

Process	How supported by the proposed implementation
5.3.1 Add/edit gene	• *Add new gene* button in the main questionnaire's *Gene* table opens a modal window where required fields can be filled in • Clicking *OK* creates empty tables for variants, star alleles, star allele functions and phenotypes. • For editing, a button next to each gene opens a modal edit window.
5.3.2 Add/edit information table	• *Add new table* button in the main questionnaire's *Curated Database Table* opens a modal window where required fields can be filled in and the table file uploaded. • The corresponding gene can be selected from a drop-down list of genes in the *Gene* table. • For editing, a button next to each table opens a modal edit window.

5.3.3 Add/edit star allele function	• *Add new function* button in the main questionnaire's *StarAlleleFunction* table under each gene opens a modal window where the function name can be inserted. • For editing, a button next to each function opens a modal edit window.
5.3.4 Add/edit phenotype	• *Add new phenotype* button in the main questionnaire's *Phenotype* table opens a modal window where the phenotype name can be inserted. • Clicking *OK* creates an empty definitions table under the phenotype. • For editing, a button next to each phenotype opens a modal edit window.
5.3.5 Add/edit phenotype definition	• *Add new definition* button in the main questionnaire's *PhenotypeTable* table opens a modal window where two defining star allele functions from the *StarAlleleFunction* table can be selected via a drop-down list. • For editing, a button next to each definition opens a modal edit window.
5.3.6 Add/edit drug	• *Add new drug* button in the main questionnaire's *Drug* table opens a modal window, where the drug name can be inserted. • For editing, a button next to each drug opens a modal edit window.
5.3.7 Add/edit reference	• *Add new reference* button in the main questionnaire's *Reference* table opens a modal window, where required fields can be filled, and a publication file may be uploaded if applicable. • Related drugs can be selected from the *Drug* table via a drop-down list. • For editing, a button next to each reference opens a modal edit window.
5.3.8 Add/edit variant	• *Add new variant* button in the main questionnaire's *Variant* table opens a modal window, where required fields can be filled. • If the variant does not have an rs-number, a knowledge base specific variant ID is assigned to it by the curator. System checks that the ID is unique over the knowledge base. • Clicking *OK* creates empty tables for alleles, references and comments under each variant. • Related references can be selected from the *Reference* table via a searchable multiple-choice list. • For editing, a button next to each variant opens a modal edit window.

5.3.9 Add/edit star allele	• *Add new star allele* button in the main questionnaire's *StarAllele* table opens a modal window, where required fields can be filled. • Clicking *OK* creates empty tables for related alleles, references and comments. • References can be selected from the *Reference* table and related alleles from *Allele* under *Variant* table via a searchable multiple-choice list.
5.3.10 Choose relevant star alleles and variants	• When adding or editing a star allele in the main questionnaire, a script is used to determine the importance of the star allele according to predefined criteria by asking simple yes/no questions from the curator. • Variants whose alleles define important star alleles are marked as important for the algorithm. • Values changed by scripts cannot be changed manually.
5.3.11 Review gene	• A separate questionnaire that does not allow data modification is used for reviewing. • Gene data and review status can be seen in the *Gene* table. • A list of not reviewed star alleles and variants is shown under each gene, and only once the lists are empty, the gene can be marked as reviewed. • The name of the reviewer is saved in the review status's change history automatically.
5.3.12 Review star allele	• Reviewing questionnaire is used. • Star allele data and review status can be seen in the *StarAllele* table. • For each related allele in the *RelatedAllele* table, its variant's review status is shown. • The name of the reviewer is saved in the review status's change history automatically.
5.3.13 Review variant	• Reviewing questionnaire is used. • Variant data and review status can be seen in the *Variant* table. • After confirming that all data (including references and alleles) is correct, a variant is marked as reviewed, and the review date is added automatically. • The name of the reviewer is saved in the review status's change history automatically.
5.3.14 Export data for the pharmacogenetic device	• Four separate tables are accessible via the exporting questionnaire. • These tables are created with SQL queries and cannot be changed manually. • Only variants and star alleles that are important for the pharmacogenetic device are included in the tables, and a warning is shown if some included data have not been

	reviewed.
5.3.15 Delete domain object	• Deleting data from the knowledge base is only done on very rare occasions and must be strictly controlled. • A separate questionnaire is used for this, and only users with a temporarily assigned *delete* role can access it. The role can be assigned by the system administrator and must be changed back to the user's usual role afterwards.

All processes were executable on the QDMP solution. Over time, more processes may be defined and implemented.

6.5.2 Verification of Meeting the Expectations

In table 6.2 below, the list of additional expectations identified in section 5.4 is shown together with the description of how this requirement is met.

Table 6-2. Expectations verification

Expectation	**How supported by the proposed implementation**
1. The knowledge base must have a graphical user interface.	Yes. The knowledge base is accessible via a web browser
2. Access to the knowledge base must be limited to authenticated users only.	Yes. Standard QDMP and LDAP authentication (university login) is used.
3. It must be possible to have different user roles with different access rights: • Administrator – manages the knowledge base settings, assigns user roles and can also add and edit information in the knowledge base; • Curator – the default role for knowledge base users, can add and edit information in the knowledge base; • reviewer – can edit the review statuses but no other data elements; • delete – a role temporarily assigned to curators for deleting data elements.	Yes. The data model is role-based (*administrator, curator, delete*). Each role has a different set of rights. The system administrator can assign roles to users. The *curator* role fills the purposes of both *curator* and *reviewer*, but their set of rights for each task are separated by using two different questionnaires for adding/modifying data and reviewing it.
4. Each knowledge base user (curator) must be able to access data with different rights depending on the task	Yes. The system administrator can temporarily assign a different role to a user (e.g. *deleteRight* role for accessing the deleting questionnaire).

they want to complete.	
5. An empty information table in predefined format must be created automatically when a new gene, reference, database table or any of their subobject types is entered into the knowledge base.	Yes. When the *OK* button is clicked after adding a new gene, variant, reference, etc., empty tables are created for each of their subobject types.
6. Different processes must be isolated.	Yes. Separate questionnaires are used for reviewing, deleting and adding/editing data. Adding and editing data are separated by using different modal windows for either process.
7. It must be possible to mark attributes as unique.	Yes. Qure Designer has the choice to mark an attribute as unique in the *Attributes* section of the *Properties* window.
8. It must be possible to mark some object attributes as mandatory.	Yes. Qure Designer has the choice to mark an attribute as mandatory in the *Attributes* section of the *Properties* window.
9. If mandatory attributes are not filled in, an error must be shown.	Yes. When *OK* is clicked before filling in mandatory fields, an error message "Could not save data because there are validation errors on the page" is shown at the top of the adding page and "Question is mandatory" is displayed next to the unanswered mandatory question.
10. Data cannot be saved when mandatory attributes are missing.	Yes. When mandatory fields are not filled in, clicking *OK* will only result in displaying error messages, and data cannot be saved.
11. It must be possible to view only a certain subset of information when needed.	Yes. The Dataview section in the Qure Browser can be used to query only certain data by selecting wanted object types and fields in a multiple-choice list.
12. Each column of each information table must be searchable.	Yes. Each table in the Qure Browser has a search button at the top right corner. Once it is clicked, a search box appears for each column.
13. It must be possible to open and edit only one row at a time in each table.	Yes. Read-only detailed information pages and modal edit pages are available for each object type. While the modal page is open, no changes can be made elsewhere.
14. Changing data directly in information tables must be prohibited.	Yes. All data is set as read-only in each table. Editing is possible only through separate editing pages.
15. Each modification must be automatically linked to the curator who made it.	Yes. Each field has a separate change history by default, where the curator's name is saved automatically with each change.
16. Each modification must be automatically linked to the date on which it was made.	Yes. Each field has a separate change history by default, where the date of each change is saved automatically.

17. A changelog with names and dates must be available for each individual cell in any information table.	Yes. Each field has a separate change history by default, where the date and the curator's name is saved automatically with each change.
18. It must be possible to add a comment for each change to each individual cell.	Yes. Each attribute has a Commentable attribute in the Attributes section of the Properties window in Qure Designer. When this is enabled, double-clicking the question while inserting or editing data in the browser opens a field where the comment can be inserted.
19. The review status of each gene, variant and star allele must be visible.	Yes. Each gene, variant and star allele has a boolean checkbox that states whether or not it has been reviewed.
20. Marking a gene as reviewed must be prohibited if any of its variants or star alleles are not reviewed.	Yes. While some variants or star alleles are not reviewed, the *isReviewed* checkbox is disabled with a conflict, and a warning is displayed.
21. For each star allele, the review status of the variants of its related alleles must be visible.	Yes. An SQL based database view is used to display the variants' review statuses next to each related allele under each star allele.
22. If an object is modified, its review status must automatically switch to *not reviewed*.	Yes. A JavaScript script is used to compare the last review date with the last modification date. If the review date is older than the modification date, the status is set to *not reviewed.*
23. If any subobject of an object is modified, the object's status must automatically switch to not reviewed.	Yes. SQL based dataviews have been created to collect the review dates of all subobjects and a JavaScript script is used to compare the largest date with the latest review date. If the review date is older than the modification date, the status is set to *not reviewed.*
24. Identifying important variants and star alleles must be supported with a script.	Yes. When adding or editing a star allele, a script is used to determine the importance of the star allele according to predefined criteria by asking simple yes/no questions from the curator. Variants whose alleles define important star alleles are marked as important for the algorithm.
25. Variants and star alleles that have been marked as important must be selected for export automatically	Yes. The content for export tables is gathered by SQL queries, which filter the variants and star alleles based on their importance (*usedByAlgorithm* and *importantForAlgorithm*).
26. Exportable tables must be accessible at all times.	Yes. Each exportable table is a separate object type, and they can be accessed by a special questionnaire.
27. Exporting data that has not been reviewed must be prohibited.	Yes. A warning is displayed on a separate page in the exporting questionnaire if exportable tables include star alleles, variants or genes that have not been reviewed.

28. The exported data must be machine-readable.	Yes. Data is exported as XLSX files.
29. The exportable data cannot be modified manually and should not include any data in free text form.	Yes. All values in the exportable tables are gathered from structured fields. Exportable tables cannot be modified manually.
30. The exported data must be in a format that is suitable for the pharmacogenetic device.	Yes. The exact format and content of the exportable tables were discussed with the people in charge of designing the pharmacogenetic algorithm.
31. Adding duplicates must be prohibited for every gene, variant, star allele, drug and reference.	Yes. Each gene's, drug's and star allele's name attribute, each reference's title and each variant's variant ID and rs number are marked as unique in Qure Designer.
32. It must be possible to link one or several references to each variant and star allele.	Yes. Each *Variant* and *StarAllele* table has a separate subtable, to which references available in the knowledge base can be added via a multiple-choice list. *Reference* object type is marked as the domain of possible options for the list in the *Attributes* section of the *Properties* window in QureDesigner.
33. It must be possible to link one or several drugs to each reference.	Yes. The *Reference* table has a *whichDrug* attribute, for which the suitable drugs can be selected via a drop-down list of all drugs present in the *Drug* table. *Drug* object type is marked as the domain of possible options for *whichGene* in the *Attributes* section of the *Properties* window in QureDesigner.
34. It must be possible to link one or several variant alleles to each star allele.	Yes. The *StarAllele* table has a *RelatedAllele* child table, to which relevant alleles can be selected from a multiple list choice. *RelatedAllele* is an instance of *Allele* under *Variant*, so *Allele* object type is marked as the domain of possible options for the list in the *Attributes* section of the *Properties* window in QureDesigner.
35. Gene-specific information must be isolated from other genes	Yes. A limited gene-specific domain has been set for each multiple-choice or drop-down list (e.g. for selecting related alleles for a star allele) from the *Attributes* section of the *Properties* window in Qure Designer.
36. It must be possible to access the list of drugs in the system from under every gene.	Yes. *Drug* object type is a root level table, and its contents can be accessed from each gene.
37. It must be possible to add multiple comments for each variant and star allele.	Yes. A separate *Comment* (*StarAlleleComment*) object type has been added under *Variant* and *StarAllele*, where several comments can be added.

The KB implementation managed to meet all predetermined expectations. Additional ideas for improvement were also explored during the verification process. Some of them are briefly mentioned in the discussion chapter.

6.6 Filling the Knowledge Base for Five Pharmacogenes

In order to get the absolute confidence that the proposed implementation is well suitable for the knowledge base, the author acted as a curator and filled in the knowledge base with actual data about five pharmacogenes which altogether are responsible for metabolizing over 20 different drugs. For this purpose, a thorough and organised literature review was conducted. The data was curated and inserted into the knowledge base, following the processes defined in 5.3. The supervisor of this thesis acted as a reviewer. Table 6.3 below summarizes the information that was researched about these genes.

Table 6.3 Summary of Researched Pharmacogenes.

Gene	Related drugs	Variants studied	Variants high evidence	Star alleles studied	Star alleles high evidence
CYP2C9	Phenytoin, celecoxib, diclofenac, flurbiprofen. Indomethacin, ibuprofen, mornoxicam, eloxicam, nabumetone, naproxen, piroxicam, tenoxicam, warfarin	58	18	61 (including wild type)	19 (including wild type)
CYP3A5	Tacrolimus	8	3	9	4
UGT1A1	Atazanavir	5	3	10	7
TPMT	Azathioprine, mercatopurine, thioguanine	39	3	43	4
DPYD	Fluorouracil, capecitabine, tegafur	83	6	83	6

The differences between the number of all researched variants/star alleles and the number of those with high evidence is a clear illustration of the abundance of information available in most online databases. The resulting export tables from QDMP are given in appendix I.

7 Discussion

When designing a new IT solution, general processes and descriptions can be derived from the nature of the data and the expectations of the final users. They provide a good insight into the system's main functionalities and characteristics, but attention must be paid to their level of detail. Vague descriptions are not informative enough, while too detailed ones may unnecessarily limit the choice of suitable technology.

While implementing the knowledge base in QDMP, some aspects had more complexity to them than initially anticipated. For example, in order to automatically change the review status of a gene, variant or star allele whenever their data is modified, creating additional SQL database views was required since the hierarchical object model does not allow referencing to subobject types from a higher level and the possibilities of keeping track of the subobject's modifications were limited.

Several complications originated from the peculiarities of pharmacogenetic data. The representation of star alleles and variants can vary across different resources and datasets. Different reference genomes may be used to determine the position of a mutation, and different notations may be used for reference and effect alleles containing more than one nucleotide. For example, the alleles of variant rs3064744 in the UGT1A1 gene are described as C, CAT, CATAT, CATATAT in some sources and as delTATA, delTA, dupTA, dupTATA in others. In the current implementation of the knowledge base, one notation is selected for each variant, but the options for adding more flexibility might be considered in the future.

The star allele nomenclature is a convenient way of describing genetic variation, but it must be interpreted with caution. In some cases, star alleles can be defined incorrectly. If two variants are in high linkage disequilibrium and typically occur together, it might happen that only one of them is detected, and the star allele function is assigned to the wrong variant. This makes it harder to find variant-specific references since often only the star allele is mentioned in articles, and no guarantee is given on whether the correct definition is used. The worst-case scenario would be that these inconsistencies lead to incorrect conclusions about the patient's phenotype since incorrect variants were exported to the algorithm.

In addition, while for most pharmacogenes, the phenotype is determined by the star allele functions, there are some exceptions. For example, *80 of the UGT1A1 gene is of uncertain function but in high linkage disequilibrium with two decreased function alleles. Therefore, CPIC states that if only rs887829 (defining variant for *80) is queried for in the data, then the phenotype may be determined based on this variant as well. The phenotype definition table offers rs887829 T/T as one possible definition for the poor metabolizer phenotype. This is a unique case among the genes investigated so far by the GenMed team, and the knowledge base implementation does not currently support such definitions.

Information may be contradictory even in renowned sources that work in close collaboration. For example, CYP2C9*19 was deemed as uncertain function by PharmGKB and decreased function by a CPIC guideline. The possible reason for that is that PharmGKB had simply not updated their information yet based on new evidence. Such occurrences once again accentuate the importance

of a curated knowledge base, where only relevant information is included and has been meticulously reviewed.

Over time, there were several modifications to the methodology of creating the knowledge base. In the beginning, variant and star allele references were searched from a database, and almost all articles were skimmed over to find information. This approach was somewhat chaotic and resulted in references with different levels of reliability. Therefore, a more structured approach was adopted. All research began with the gene-specific guidelines, and only a few additional articles were read and referenced for more detailed information. In addition, if several articles had been published after the latest CPIC guideline and they all had reached the same conclusions, these articles were also included in the knowledge base.

The criteria based on which the important variants and star alleles were selected was also modified twice during the prototyping phase. At first, all decisions were based on the curators' interpretation of the read articles. This approach left a lot of room for inconsistencies since no precise criteria were defined. A very strict criterion was formulated to mitigate this risk - only variants and star alleles mentioned in the main article of the CPIC guideline can be considered important. This approach was later deemed overly restrictive since the main articles often only bring examples of star alleles for each possible function. The criteria were then formulated as it was presented in this thesis. Once the data had been inserted into the QDMP knowledge base, the decisions about star alleles' and variants' importance were compared to those made during the prototyping phase. Some prototype decisions were found that deviated from the defined decision criteria. This finding supports the claim that well-defined processes with additional IT support are needed to ensure the dataset's consistency. Even though the exact choice of decision criteria can be arguable, following one still ensures that all decisions are made on even ground and can be justified.

The current implementation in QDMP is a good baseline for the knowledge base. Still, there are some usability aspects for which the most convenient approach can be determined over time by actively using the knowledge base. For example, additional warnings and database views or different comments sections might facilitate navigating the information in the knowledge base. Currently, all questions and concerns can be written in the comments sections under each star allele and variant, and a warning can be raised on the variant' or star allele's detailed view page that notifies other curators of unresolved issues. Since the knowledge base has not been in active use by multiple curators, it is hard to say whether this solution is effective or not.

The complex and multi-layered hierarchy of the knowledge base is good for creating necessary associations between object types, but it makes the knowledge base harder to navigate. The tabs of all subobject types are always visible, but one might still forget about them. For example, one might forget to add related alleles to a star allele because it is done from a separate page. In the future, it might be reasonable to add more SQL views to higher-level objects that would show a list of subobject types that have not been added yet or simply use conflicts to show reminders when certain fields are filled.

The *usedByAlgorithm* decision is currently made for each variant, but making it separately for each allele might be justified. This would make a difference in how multiallelic variants are handled. If only one of the two effect alleles of a variant is part of an important star allele, only this allele needs to be exported to the algorithm. In the CYP2C9 gene, variant rs7900194 has three possible

alleles: G (wild-type), A (defines a star allele with decreased function) and T (defines a star allele with uncertain function). Only G and A are important for the algorithm. All three alleles would be exported with the current solution, but only G and A would be used in the algorithm code. Moving the usedByAlgorithm to the related allele would result in only G and A being exported.

Currently, the knowledge base also includes some articles with contradictory data to document that this kind of information has erroneously been provided about a variant. This helps to avoid confusion in the future when similar wrong information is reencountered. However, it might not be necessary to crowd the knowledge base with irrelevant articles - a comment explaining past misconceptions might be enough.

Since the pharmacogenetic device is also still in development, exact requirements for the knowledge base may still be specified over time. This means that a thorough testing and verification plan could not yet be carried out. However, initial verification has been done against existing processes and expectations, and major changes to those are unlikely to occur.

One of the main advantages of the GenMed knowledge base is that it does not strongly depend on any specific external variables. If some of the resources where the data is gathered from should ever shut down, alternative sources could be used for data gathering while leaving other workflows implemented in the knowledge base unchanged. In addition, the novelty of the GenMed knowledge base among similar solutions described above is that only the variants and star alleles with the highest available evidence levels are exported to the algorithm. Such an approach may limit the number of genotype interpretations that the algorithm can make, but at the same time, it helps to ensure the highest possible quality, which is crucial in a clinical setting.

8 Conclusion

The goal of this thesis was to determine the main processes, necessary user roles and important features of a reliable knowledge base for a pharmacogenetic test, and propose a solution for its implementation. This asked for a thorough review of published literature and online resources, whereby only carefully curated scientific data was incorporated into the knowledge base.

A meticulous review was conducted to analyse the advantages and disadvantages of existing resources and to detect as many idiosyncrasies about pharmacogenetic data as possible. Based on the results of the review, a detailed prototype was created, where the processes, user roles and general expectations to the knowledge base were identified and elaborated while working with real data.

The knowledge base was implemented on the Qure Data Management Platform, and the data was inserted into the final implementation while verifying that all necessary processes had been followed without complications.

The created knowledge base meets all expectations that were identified during the prototyping phase, but its development is expected to continue. Once the pharmacogenetic phenotyping device that uses the exported knowledge base will reach the implementation phase, the two essential components will have to be matched and tested for compliance. Therefore, a detailed list of requirements will be created, and additional process support might become necessary. Moreover, clinically applicable data of both existing and additional pharmacogenes will increase and change over time, and that has to be reflected in the updated knowledge base. However, no fundamental changes to the existing structure and processes are expected to become necessary.

As a result of this thesis, a knowledge base that contains carefully curated scientific data is available for use by the GenMed pharmacogenetics team. In addition, a detailed description of the knowledge base and the curation processes is described in this thesis. This can serve as a reference point for potential new curators who need to learn to use this system. The knowledge base supports the testing of the pharmacogenetic algorithm and facilitates future data curation by offering technical support for detailed workflows and decision criteria.

9 References

[1] R. F. Vogenberg, C. I. Barash and M. Pursel, "Personalized medicine," *Pharmacy and Therapeutics,* vol. 35, no. 10, pp. 560-562, 565-567, 576, 2010. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2957753/. (19.04.2021)

[2] S. Mathur and J. Sutton, "Personalized medicine could transform healthcare," *Biomedical Reports,* vol. 7, no. 1, pp. 3-5, 2017. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5492710/. (19.04.2021)

[3] A. Pokorska-Bocci, A. Stewart, S. G. Sagoo, A. Hall, M. Kroese and H. Burton, "'Personalized medicine': what's in a name?," *Personalised medicine,* vol. 11, no. 2, pp. 197-2010, 2014. https://www.futuremedicine.com/doi/pdf/10.2217/pme.13.107. (18.04.2021)

[4] W. M. Gibson, "Can Personalized Medicine Survive?," *Canadian Family Physician,* vol. 17, no. 8, pp. 29-31, 33, 88, 1971. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2370041/?page=5. (18.04.2021)

[5] J. T. Jørgensen, "Twenty Years with Personalized Medicine: Past, Present, and Future of Individualized Pharmacotherapy," *The Oncologist,* vol. 24, no. 7, pp. 432-440, 2019. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6656435/. (18.04.2021)

[6] J. Lass, K. Krebs, A. Irs and L. Milani, "Farmakogenoomika – teekond ravivastuse päriliku varieeruvuse baasteadusest kliinilisse meditsiini," *Eesti Arst,* vol. 100, no. 1, pp. 34-43, 2021.

[7] K. E. Caudle, H. M. Dunnenberger, R. R. Freimuth, J. F. Peterson, J. D. Burlison, M. Whirl-Carrillo, S. A. Scott, H. L. Rehm, M. S. Williams, T. E. Klein, M. V. Relling and J. M. Hoffman, "Standardizing terms for clinical pharmacogenetic test," *Genetics in Medicine,* vol. 19, no. 2, pp. 215-223, 2017. https://www.nature.com/articles/gim201687.pdf?origin=ppub. (09.05.2021)

[8] J. Robarge, L. Li, Z. Desta, A. Nguyen and D. Flockhart, "The Star-Allele Nomenclature:," *Clinical Pharmacology andTherapeutics,* vol. 82, no. 3, pp. 244-248, 2007. https://ascpt.onlinelibrary.wiley.com/doi/abs/10.1038/sj.clpt.6100284. (07.05.2021)

[9] A. Gaedigk, M. Ingelman-Sundberg, N. A. Miller, J. S. Leeder, M. Whirl-Carrillo, T. E. Klein and PharmVar Steering Comitee, "The Pharmacogene Variation (PharmVar) Consortium: Incorporation of the Human Cytochrome P450 (CYP) Allele Nomenclature Database," *Clinical Pharmacology & Therapeutics,* vol. 103, no. 3, pp. 399-401, 2017. https://ascpt.onlinelibrary.wiley.com/doi/full/10.1002/cpt.910. (21.04.2021)

[10] A. Gaedigk, K. Sangkuhl, M. Whirl-Carrillo, G. P. Twist, T. E. Klein and N. A. Miller, "The Evolution of PharmVar," *Clinical Pharmacology & Therapeutics,* vol. 105, no. 1, pp. 29-32, 2018. https://ascpt.onlinelibrary.wiley.com/doi/full/10.1002/cpt.1275. (18.04.2021)

[11] M. Whirl-Carrillo, E. McDonagh, J. Hebert, L. Gong, K. Sangkuhl, C. Thorn, R. Altman and T. Klein, "Pharmacogenomics Knowledge for Personalized Medicine," *Clinical Pharmacology & Therapeutics,* vol. 92, no. 4, pp. 414-417, 2012. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3660037/. (04.05.2021)

[12] M. Relling and T. Klein, “CPIC: Clinical Pharmacogenetics Implementation Consortium of the Pharmacogenomics Research Network,” *Clinical Pharmacology & Therapeutics,* vol. 89, no. 3, pp. 464-467, 2011. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3098762/. (25.04.2021)

[13] M. V. Relling, T. K. Klein, R. S. Gammal , M. Whirl-Carrillo, J. M. Hoffman and K. E. Caudle, “The Clinical Pharmacogenetics Implementation Consortium: 10 Years Later,” *Clinical Pharmacology & Therapeutics,* vol. 107, no. 1, pp. 171-175, 2020. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6925644/. (25.04.2021)

[14] L. Leitsalu, T. Haller, T. Esko, M.-L. Tammesoo, H. Alavere, H. Snieder, P. Markus, P. C. Ng, R. Mägi, L. Milani, K. Fischer and A. Metspalu, “Cohort Profile: Estonian Biobank of the Estonian Genome Center, University of Tartu,” *International Journal of Epidemiology,* vol. 44, no. 4, pp. 1137-1147, 2015. https://academic.oup.com/ije/article/44/4/1137/666872?login=true. (08.01.2021)

[15] K. N. Theken, C. R. Lee, L. Gong, K. E. Caudle, C. M. Formea, A. Gaedigk, T. E. Klein, J. A. Agundez and T. Grosser, “Clinical Pharmacogenetics Implementation,” *Clinical Pharmacology & Therapeutics,* vol. 108, no. 2, pp. 191-200, 2020. https://pubmed.ncbi.nlm.nih.gov/32189324/. (06.05.2021)

Appendix

I. Exportable tables for the PGx medical device

1. Export Variant

gene	variantID	rsID	chro-mosome	posHG37	possible alleles
CYP2C9	GenMedCYP2C9_4	-	10	96701672	G,A
CYP2C9	rs72558187	rs72558187	10	96701715	C,T
CYP2C9	rs762239445	rs762239445	10	96701739	T,G
CYP2C9	rs12414460	rs12414460	10	96701988	A,G
CYP2C9	rs72558189	rs72558189	10	96701991	A,T,G
CYP2C9	rs199523631	rs199523631	10	96702011	C,T
CYP2C9	rs1799853	rs1799853	10	96702047	C,T
CYP2C9	rs7900194	rs7900194	10	96702066	A,T,G
CYP2C9	rs72558190	rs72558190	10	96707539	A,C
CYP2C9	rs9332131	rs9332131	10	96709040	delA,A
CYP2C9	rs72558192	rs72558192	10	96731936	A,G
CYP2C9	rs988617574	rs988617574	10	96731937	G,C
CYP2C9	rs57505750	rs57505750	10	96740958	C,T
CYP2C9	rs28371685	rs28371685	10	96740981	C,T
CYP2C9	rs1057910	rs1057910	10	96741053	C,A
CYP2C9	rs56165452	rs56165452	10	96741054	C,T
CYP2C9	rs28371686	rs28371686	10	96741058	C,G
CYP2C9	rs769942899	rs769942899	10	96748674	C,G
CYP3A5	rs41303343	rs41303343	7	99250393	delA,insA
CYP3A5	rs10264272	rs10264272	7	99262835	C,T
CYP3A5	rs776746	rs776746	7	99672916	C,T
DPYD	rs67376798	rs67376798	1	97547947	A,T
DPYD	rs3918290	rs3918290	1	97915614	T,C
DPYD	rs55886062	rs55886062	1	97981343	A,C
DPYD	rs56038477	rs56038477	1	98039419	C,T
DPYD	rs75017182	rs75017182	1	98045449	C,G
DPYD	rs115232898	rs115232898	1	98165030	T,C
TPMT	rs1142345	rs1142345	6	18130918	C,G,T
TPMT	rs1800460	rs1800460	6	18139228	C,T
TPMT	rs1800462	rs1800462	6	18143955	C,G
UGT1A1	GenMedUGT1A1_1	-	2	234668879	C,CAT,CA-TAT,CATATAT
UGT1A1	rs4148323	rs4148323	2	234669144	A,G
UGT1A1	rs35350960	rs35350960	2	234669619	A,C

2. Export Star Allele

gene	starAlleleName	function	isNo-Function	isWildType
CYP2C9	*1	Normal function	false	true
CYP2C9	*11	Decreased function	false	false
CYP2C9	*13	No function	true	false
CYP2C9	*15	No function	true	false
CYP2C9	*16	Decreased function	false	false
CYP2C9	*19	Decreased function	false	false
CYP2C9	*2	Decreased function	false	false
CYP2C9	*23	Decreased function	false	false
CYP2C9	*3	No function	true	false
CYP2C9	*31	Decreased function	false	false
CYP2C9	*35	No function	true	false
CYP2C9	*39	No function	true	false
CYP2C9	*4	Decreased function	false	false
CYP2C9	*42	No function	true	false
CYP2C9	*45	No function	true	false
CYP2C9	*5	Decreased function	false	false
CYP2C9	*52	No function	true	false
CYP2C9	*6	No function	true	false
CYP2C9	*8	Decreased function	false	false
CYP3A5	*1	Normal function	false	true
CYP3A5	*3	No function	true	false
CYP3A5	*6	No function	true	false
CYP3A5	*7	No function	true	false
DPYD	c.1129-5923C>G, c.1236G>A (HapB3)	No function	true	false
DPYD	c.1679T>G (*13)	No function	true	false
DPYD	c.1905+1G>A (*2A)	No function	true	false
DPYD	c.2846A>T	Decreased function	false	false
DPYD	c.557A>G	Decreased function	false	false
DPYD	Reference	Normal function	false	true
TPMT	*1	Normal function	false	true
TPMT	*2	No function	true	false
TPMT	*3A	No function	true	false
TPMT	*3B	No function	true	false
TPMT	*3C	No function	true	false
UGT1A1	*1	Normal function	false	true
UGT1A1	*27	Decreased function	false	false
UGT1A1	*28	Decreased function	false	false
UGT1A1	*36	Increased function	false	false
UGT1A1	*37	Decreased function	false	false
UGT1A1	*6	Decreased function	false	false
UGT1A1	*80	Uncertain function	false	false

3. Export Star Allele Definition

gene	starAlleleName	variantID	allele
CYP2C9	*11	rs28371685	T
CYP2C9	*13	rs72558187	C
CYP2C9	*15	rs72558190	A
CYP2C9	*16	rs72558192	G
CYP2C9	*19	rs769942899	C
CYP2C9	*2	rs1799853	T
CYP2C9	*23	GenMedCYP2C9_4	A
CYP2C9	*3	rs1057910	C
CYP2C9	*31	rs57505750	C
CYP2C9	*35	rs72558189	T
CYP2C9	*39	rs762239445	T
CYP2C9	*4	rs56165452	C
CYP2C9	*42	rs12414460	A
CYP2C9	*45	rs199523631	T
CYP2C9	*5	rs28371686	G
CYP2C9	*52	rs988617574	G
CYP2C9	*6	rs9332131	delA
CYP2C9	*8	rs7900194	A
CYP3A5	*3	rs776746	C
CYP3A5	*6	rs10264272	T
CYP3A5	*7	rs41303343	insA
DPYD	c.1129-5923C>G, c.1236G>A (HapB3)	rs56038477	T
DPYD	c.1129-5923C>G, c.1236G>A (HapB3)	rs75017182	C
DPYD	c.1679T>G (*13)	rs55886062	C
DPYD	c.1905+1G>A (*2A)	rs3918290	T
DPYD	c.2846A>T	rs67376798	A
DPYD	c.557A>G	rs115232898	C
TPMT	*2	rs1800462	G
TPMT	*3B	rs1800460	T
TPMT	*3C	rs1142345	C
UGT1A1	*27	rs35350960	A
UGT1A1	*28	GenMedUGT1A1_1	CATAT
UGT1A1	*36	GenMedUGT1A1_1	C
UGT1A1	*37	GenMedUGT1A1_1	CATATAT
UGT1A1	*6	rs4148323	A

4. Export Function To Phenotype

gene	function1	function2	phenotype
CYP2C9	No function	No function	Slow metabolizer
CYP3A5	No function	No function	Poor metabolizer
CYP3A5	Normal function	No function	Intermediate metabolizer
CYP3A5	Normal function	Normal function	Extensive metabolizer
CYP3A5	Normal function	Uncertain function	Intermediate metabolizer
CYP3A5	Uncertain function	Uncertain function	Indeterminate
DPYD	Decreased function	Decreased function	Intermediate metabolizer
DPYD	No function	Decreased function	Poor metabolizer
DPYD	No function	No function	Poor metabolizer
DPYD	Normal function	Decreased function	Intermediate metabolizer
DPYD	Normal function	No function	Intermediate metabolizer
DPYD	Normal function	Normal function	Normal metabolizer
UGT1A1	Decreased function	Decreased function	Poor metabolizer
UGT1A1	Increased function	Decreased function	Intermediate metabolizer
UGT1A1	Increased function	Increased function	Extensive metabolizer
UGT1A1	Normal function	Decreased function	Intermediate metabolizer
UGT1A1	Normal function	Increased function	Extensive metabolizer
UGT1A1	Normal function	Normal function	Extensive metabolizer

II. License

Non-exclusive licence to reproduce thesis and make thesis public

I, Ida Maria Orula,

(*author's name*)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

The Process of Creating a Scientific Knowledge Base for Pharmacogenetic Testing,

(*title of thesis*)

supervised by Sulev Reisberg, Phd and Kersti Jääger, Phd.

(*supervisor's name*)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Ida Maria Orula

14/05/2021