New Open Access Interactive Multifunctional Database Management System for Research of Biological Terminology: Technical Solutions

Karina Šķirmante[1](✉), Gints Jasmonts[1], Roberts Ervīns Ziediņš[1], Silga Sviķe[1], and Arturs Stalažs[2]

[1] Ventspils University of Applied Sciences, Inženieru iela 101a, Ventspils LV-3601, Latvia
karina.krinkele@venta.lv

[2] Institute of Horticulture, Graudu iela 1, Cerini, Krimunu pagasts,
Dobeles novads LV-3701, Latvia

Abstract. Terminology work requires a lot of manual processing, especially extracting data from materials that have not been digitized, so more efficient solutions and tools are necessary to improve the efficiency of the research. Since 2021, a team of terminologists, translators, researchers, information system developers worked together, during which the new open-access interactive multifunctional information management system was designed and developed for data storage and a wide range of statistical and search options especially for language research purposes and comparative multilingual studies in linguistics. The information system consists of multiple modules, which are successfully designed and developed and have been effectively used by researchers during the period of August 2021—July 2023 for entering, collection and retrieval data for special lexis research.

Keywords: Design and development of information system · terminology · data collection

1 Introduction

1.1 State of Art

In some research areas, the lack of appropriate wide usage electronic tools and databases still slows down the practical research in contrastive linguistics, especially when it comes to large groups of special lexis with many synonyms, such as the names of plants or animals. It should be noted that special lexis and terminology is often an issue not only for lexicographers and translators, but also for people in other professions whose work involves this terminology, for instance journalists, content creators, teachers and students [1]. Many scholars usually create their own glossaries and term bases using Computer Assisted Translation tools (e.g. Memsource, SDL Trados Studio) or simply in Word files or Excel worksheets to collect linguistic information for their studies. Incidentally, most of the scientific data remains in personal computers and is not available to a wider public, especially for international researchers, linguists, terminologists and translators,

T. Guarda et al. (Eds.): ARTIIS 2023, CCIS 1935, pp. 282–296, 2024.
https://doi.org/10.1007/978-3-031-48858-0_23

therefore the results of initial studies are not verifiable, research is not repeatable, and it cannot be compared to more recent studies, as information sources from previous research are usually not known.

There are databases which are dedicated to scientific names of organisms, but the coverage of the local names of organisms is minor. For example, the World's Flora Online [2] as 'an open-access web-based compendium of the world's 400,000 species of vascular plants and mosses' or all animal names e.g. Additionally, the Plants of the World Online database [3] offers a wealth of accurate information, making it highly valuable for scientific research. This database includes an extensive collection of synonyms and comprehensive species coverage, encompassing an impressive 1,423,000 scientific names of plants worldwide. All Animals A-Z List [4] which allows to search animal names by starting letter (in English), by scientific name, by class, location etc., but such electronic database would also be needed with the local names of organisms, which would be useful for terminology research and translators and experts in the field. There are several IT solutions that collect local names of organisms in Latvian, for example, Skosmos [5], however, this resource is more suitable for researchers in library science because researchers need the full set of designations for a given taxon to obtain data such as frequency of use, track changes or to clarify the accepted scientific or local name. There are different term bases and encyclopedias for Latvian as well but they are either not renewed, are small or contain only the names of the local flora [6].

This study is part of a larger project and the aim of the particular paper is to describe development process, characterize problems and solutions for creation of the open-access interactive multifunctional database management system (hereinafter—IMDS) which provides data storage and a wide range of statistical and search options especially for language research purposes and comparative multilingual studies in linguistics and terminology. The system will be published at the end of the year 2023 as Biolexipedia under the domain Bioleksipedija.lv. Biolexipedia is a blend word that consists of three words merged into one. It comes from the parts of words from Greek *bios* meaning 'life', from English *lexis* meaning 'the vocabulary of a particular field' and 'pedia' (from a back-formation of encyclopedia) meaning a specialized encyclopedia of biology vocabulary. Biolexipedia is planned as an universal repository of biological vocabulary, especially names of organisms.

It should be noted that the developed system may later be used for collection and research of other systems. The objectives of the IMDS development process are: 1) creation of a novel solution with a wide range of statistical and search possibilities suitable for language research and public use worldwide; 2) description of the initial linguistic and terminological research possibilities by using organism names as the lexical model field.

The IMDS solution provides statistical data for the usage of organism names, as well as automatically provides the different meanings of homonyms. From the viewpoint of language heritage, a vast number of organism name records will allow recognition of which names have already disappeared from daily usage, and which are close to disappearing because they are not included in specialized sources. A large amount of lexical background information from the IMDS gives the possibility to track back the timeline and the earliest publications to when particular new names were established.

These diachronic studies will allow researchers to compare how the old and the new names are used; and whether and when the new names possibly start dominating over the old ones. Section 3 gives an insight into some research possibilities for which the system may be used for.

1.2 Implementation of the Project

Previous mentioned activities are implemented during project ‘Smart complex of information systems of specialized biology lexis for the research and preservation of linguistic diversity’ which objectives are: 1) Creation of a novel system encompassing previously mentioned requirements; 2) Characterisation of a new IMDS by using organism names as the lexical model field. To verify the possibilities of newly created research database, more than 149,000 entries of names of different organisms (mainly in Latvian, additionally also in English, German, Russian, as well as in Estonian, Lithuanian, Polish etc.) are collected on the basis of excerpts from different publications (more than 8,470 bibliography units are used). Organism names cover all organism kingdoms, with current focus on plants, although animal and disease names are also included. Names of organisms are excerpted from different printed and electronic publications (scientific and popular scientific sources, dictionaries, specialized textbooks etc.).

2 Design of the IMDS

2.1 Overview of Used Technologies and IMDS Architecture

A team of researchers and system developers have worked together to design and develop the open-source web tool IMDS which is an effective solution for the research on special lexis of biology and related fields. In order to design the IMDS it was important to develop an application programming interface (API) driven information system to provide a successful web information system where the front-end can consist of multiple solutions and be used simultaneously. In this case, IMDS was designed as a two-part system with 1) the back-end which is primarily developed using Java programming language and a popular framework called Spring Framework [7]; 2) the front-end as single-page application (SPA) using React JavaScript libraries. This back-end and front-end design requires delivering effective communication between back-end servers and front-end application and for this action, Representational State Transfer (REST) was used. REST technology provides transmission of JavaScript Object Notation (JSON) requests and responses, and it allows to easily create an API for server client communication, by creating constant endpoints using Spring Framework controllers’ classes.

The main objective of using Java as the back-end programming language—is that it is a high-performance, object-oriented, threaded, open source, platform-independent general purpose programming language. Although it is considered interpreted, it is also compiled, and uses a technique called Just-In-Time, which compiles and makes optimizations during run time. Spring framework is based on Java, and it simplifies the development process, as it comes with built-in annotations that can map functions as necessary. IMDS includes multiple sub-frameworks of Spring framework, for example, predefined framework for security—Spring Security and predefined framework for data usage—Spring Data Java Persistence Application Programming Interface (JPA). IMDS was developed using Spring Model-View-Controller (MVC) [8] pattern with additional service implementation for the system business functionality. The front-end of the IMDS is developed using SPA technology, which helps to make the application mobile friendly and generally usable on any device. SPA is also typically faster than multiple page applications, as their scripts are only loaded once during the lifespan of the application. ReactJS's technology was selected as front-end main technology because of ReactJS effectiveness, performance and popularity in the last three years [9] and research of the possible used technologies showed that there is effective synergy between Spring framework and ReactJS in multiple large scale IT projects. For data management, three database systems were used 1) MySQL database system was used for the storage of the IMDS data collection and non- hierarchical linkages. MySQL [10] is one of the top database engines, as it is efficient and easy to use. It guarantees constant uptime, which is critical for a web-based system. MySQL is also free and open-source, and it is possible to connect using Spring Data JPA framework using interfaces of repositories. 2) Mongo DB database system was used to obtain hierarchical linkages between scientific names of organisms. MongoDB [11] is a source-available cross-platform document-oriented database program and it allows us to organize data using tree data structure. 3) Redis database was used in large data caching of IMDS. Redis [12] is an open source (BSD licensed), in-memory data structure store, used as a database, cache, and message broker. An architecture of the IMDS is shown in Fig. 1.

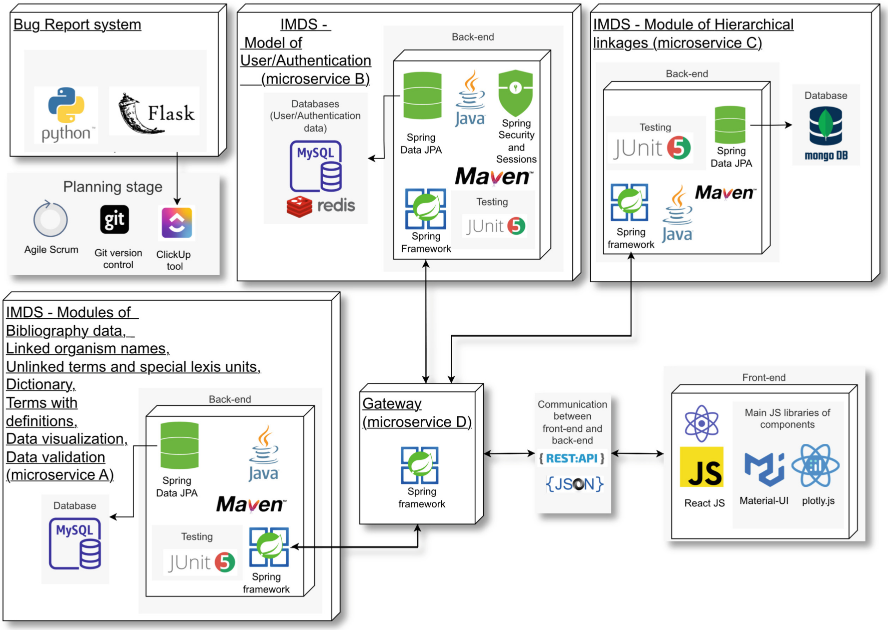

Bug Report system
python
Flask
Planning stage
Agile Scrum
git
Git version control
ClickUp tool
IMDS - Model of User/Authentication (microservice B)
Databases (User/Authentication data)
MySQL
redis
Back-end
Spring Data JPA
Java
Spring Security and Sessions
Maven
Testing
JUnit 5
Spring Framework
IMDS - Module of Hierarchical linkages (microservice C)
Back-end
Testing
JUnit 5
Spring Data JPA
Database
mongo DB
Maven
Spring framework
Java
IMDS - Modules of Bibliography data, Linked organism names, Unlinked terms and special lexis units, Dictionary, Terms with definitions, Data visualization, Data validation (microservice A)
Back-end
Spring Data JPA
Java
Maven
Database
MySQL
Testing
JUnit 5
Spring framework
Gateway (microservice D)
Spring framework
Communication between front-end and back-end
{ REST:API }
{JSON}
Front-end
Main JS libraries of components
JS
React JS
Material-UI
plotly.js

Fig. 1. Technology used in the IMDS design and development process.

IMDS design and development process was organized using Agile method Scrum, where software design and development activities were split into 2 weeks exercises called Sprints. The ClickUp tool was used for developing task management. GitHub was used for the version control and management of the IMDS code and GitHub Actions—as a continuous deployment pipeline—was used for automatization multiple processes of development workflow, for example testing and finding the vulnerabilities of the IMDS. IMDS testing was organized using the Junit framework to test model classes, validations, services, repositories, and Jest to test front-end components.

As an IMDS sub-system the new 'Bug reporting web system' was developed using the Flask framework and Python programming language. Bug reporting system is integrated with the ClickUp project of the IMDS where all development tasks are managed. IMDS users can create bug reports in the developed Bug system and the reports automatically are integrated as new tickets in the ClickUp project of the IMDS and programmers are informed about identified bugs.

At the first phase, the IMDS has been published on the internal server of Ventspils University of Applied Sciences, but the final version of IMDS will be developed as public open-access web system.

2.2 Overview of IMDS Modules and Data Layer

The developed IMDS consists of multiple modules which are mutually connected (see Fig. 2):

1. User module. All data entry is stored with information about the user who entered data. This module consists of user registration, authentication, and authorization. Additionally, the system implements a security level system based on five user roles:
   a. Unregistered users, who can perform searches and access statistics but do not have export privileges.

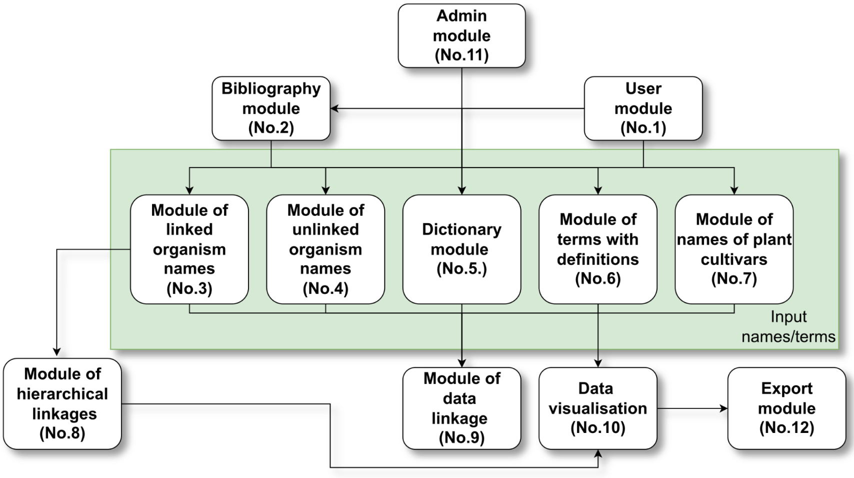

Admin module (No.11)
Bibliography module (No.2)
User module (No.1)
Module of linked organism names (No.3)
Module of unlinked organism names (No.4)
Dictionary module (No.5.)
Module of terms with definitions (No.6)
Module of names of plant cultivars (No.7)
Input names/terms
Module of hierarchical linkages (No.8)
Module of data linkage (No.9)
Data visualisation (No.10)
Export module (No.12)

Fig. 2. Final version of modules for the IMDS

b. First-level registered users, who have all the capabilities of the previous role and can also export data.
c. Second-level registered users, who possess all the capabilities of the previous roles and can input data into the system for specific publications but are restricted from editing or deleting data entered by others.
d. Third-level registered users, who are qualified professionals with expertise and credentials to ensure IMDS's integrity through data validation.
e Admin user which is responsible for system administration tasks.

2. Bibliography module. All lexeme information is linked to the bibliographical information. This module provides the ability to store all publication data: monographs, journals, proceedings and papers with linked additional information, for example, publisher, ISBN, alternative title, place, Digital Object Identifier (DOI), authors, etc. In this case, it was important: 1) that the bibliographic source is correctly linked with the specific excerpted unit; 2) to organize the collection of such publication data, which will allow further provision of an appropriate reference to the source of information in the database. In this case, it was important to ensure that only papers, monographs, series of monographs and their parts can be used as a bibliography source for the organism name, species or term linkage. It was done using Java inheritance and polymorphism possibilities. (see Fig. 3).

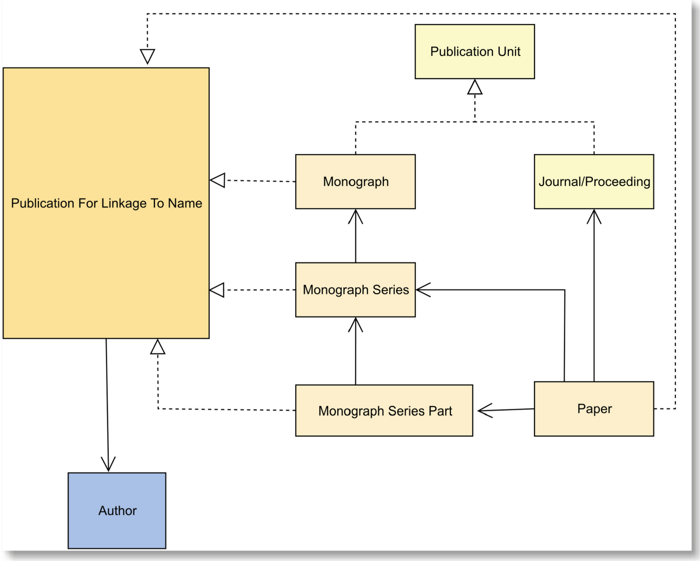


Fig. 3. The simplified Unified Modeling Language (UML) class diagram of the module of bibliography

3. Module of linked organism names. There are three types of organism names entered in this module—scientific name, local name, and names of diseases caused by organisms (since the names of organisms can also be the names of diseases). In this module, the linkage between the organism name and bibliography unit is stored with additional

information, for example, language mark, page number on the source bibliography, system user data, date and time, user comments. Also, the linkage between the root element and other names of organisms in the same linkage group is carried out in this module. In this case, multiple many-to-many tables were developed.

4. Module of unlinked terms and special lexis units. In this module, the linkage between a specific organism name (not linked to other languages) and bibliography unit is stored with additional information, for example, language mark, page number on the source bibliography, system user data, date and time, user comments. There are no linkages between other organism names.
5. Dictionary module. It is for lexemes of different languages that are not related to the scientific name of organisms and where entries are linked to specific bibliography units with additional information, for example, part of speech, gender, number, language mark, page number on the bibliographic source, system user data, date and time of the entry, user comments. Dictionary entries are linked together if there is a terminological or linguistic connection between them. This module also provides links to synonyms of the same language, if the scientific name of the organisms is not used in the original publication used for data collection in the database.
6. Module of terms and definitions. In this module, terms with their corresponding definitions are linked to specific bibliography units with definitions, for example, identified subdomains for term and for definition, related scientific term (e.g. scientific or Latin designation) and umbrella term, language mark, page number on the source bibliography, system user data, date and time of the entry, user comments, etc. The entered terms are stored with linkage to term equivalents in other languages and their corresponding definitions.
7. Module of names of plant cultivars. In this module, cultivar names are linked to bibliography units with additional information, for example, country of origin, cultivar breeder's rights owner, breeder's rights protection date, system user data, language, related organism taxon name, time and date of the entry, user comments, cultivar's description, species epithet, group affiliation, etc.

A simplified representation of the data layer of 3–7 modules shown in Fig. 4.

8. Module of hierarchical linkages. This module establishes linkages between scientific organism names at various taxonomic category levels, encompassing a comprehensive range of 33 taxonomic levels. It organizes the data into a tree-like structure, facilitating a hierarchical representation. The module consists of five distinct categories (kingdoms): plants, animals, fungi, bacteria, and viruses, enabling the creation of multiple hierarchical trees for organism names. Moreover, the module allows for linking data from the scientific name table in a MySQL database, ensuring synchronization between the two databases.
9. Module of data linkage. In this module, the linking of scientific names entered in Module 8 with the excerpt from different publications will be ensured and controlled. This will ensure that regardless of spelling differences, scientific names will be linked to the appropriate and correctly spelled names.

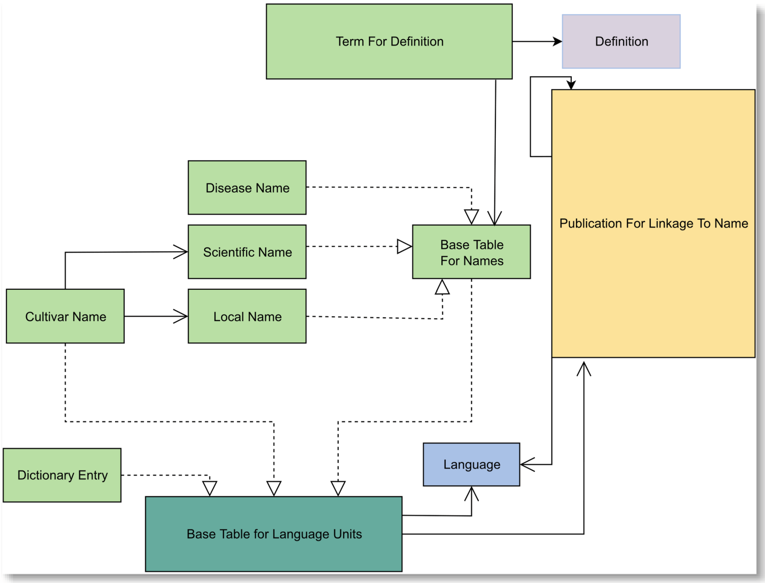


Fig. 4. The simplified UML class diagram of the names of organisms, plant cultivars names, dictionary words and terms with definitions

10. Data visualization. This is a module for information filtering and retrieving from the databases of the developed IMDS. Results are reproduced in multiple ways, for example, in graphs, plots of time series, word clouds, tables, etc. (see Fig. 6, Fig. 7, Fig. 8 and Fig. 9).
11. Admin Module. In this module, users with higher-level privileges have the ability to edit and delete associated organism names, scientific names, author first names and last names, etc. Additionally, they can approve user requests for input at a higher level.
12. Export Module. This is an additional module to complement the data visualization (No.10) module. With the help of this module, information about the searched name can be easily exported to an Excel spreadsheet for further research. The exported data is presented in a tabular format, organized into multiple sheets, each providing valuable insights. For example:
    a. Basic Information: This sheet contains data with the global comment for the name, the searched name, all found name linkages, including their name, language, group, and the number of publications. It also includes statistics with publication years and a histogram of publications within a specific time period. Additionally, users have the option to view detailed information about the publications.
    b. Data from Module No. 3 to Module No. 7: These sheets include information retrieved from various modules, together with bibliography details, language, comments, and corresponding page numbers.

2.3 Limitations and Challenges of IMDS

When developing IMDS, we encountered various challenges and limitations, such as:

1. Time Constraints: Due to the project's three-year timeline, the IMDS development process was time-limited. To address this, we implemented a strategy of publishing system modules on an internal server, where project researchers could test the functionality and report errors through the Bug Report System.
2. Complexity of Requirements: One of the most challenging tasks was designing the database schema to efficiently handle data retrievals, despite having numerous records with many-to-many relationships.
3. Compatibility: We chose JAVA programming language and Spring Framework as the core technology, as JAVA is platform-independent, ensuring the execution of code on various operating systems.
4. Scalability: The system was developed as microservices, enabling scalability with the addition of new modules.
5. Security Concerns: To safeguard user data and prevent vulnerabilities, we employed Spring Security and Session framework in conjunction with JWT tokens for secure user authentication. User data is stored in a separate database to enhance security.
6. Software Testing: We incorporated tests using JUnit and Jest frameworks during the IMDS development process to ensure software reliability.
7. User Acceptance: The system was developed in close collaboration with terminologists, translators and linguists, considering their requirements and customizing the user interface for specific researchers in the field.
8. Technical limitations (a few examples):
   a. the complexity of the data model arises due to intricate linkages between organism names, both horizontally through publications and vertically through a hierarchical tree (see optimization in Sect. 2.3);
   b. the system currently supports internationalization only for Latvian and English languages, considering the development scope and time constraints;
   c. there is a need to synchronize scientific names between two databases - MongoDB, where they are stored hierarchically, and MySQL, where they are associated with specific publications and other names. This is ensured by developing an additional synchronization Python script that runs once a day;
   d. now, the system imposes a simultaneous 30-connection limit with MySQL;
   e. data is extracted from the frontend to the backend using sessions, commonly employing batch sizes ranging from 100 to 1000 records. If necessary, this batch size can be further increased to accommodate larger data sets, up to a maximum of 10,000 records per batch.

By addressing these challenges and limitations, we aimed to create a robust and user-friendly IMDS system that meets the needs of its users effectively.

2.4 Optimization of the IMDS Performance of Big Data Retrieval

The main challenging factor was to design the opportunity to save organism name linkage to other organism names in one specific bibliography and to retrieve all linked objects as fast as possible. The IMDS system has been used for almost two years and in the most advanced case scenario so far, there are multiple cases where 49 names of organisms are linked together. Overall, a histogram of linkages counts is shown in Fig. 5.

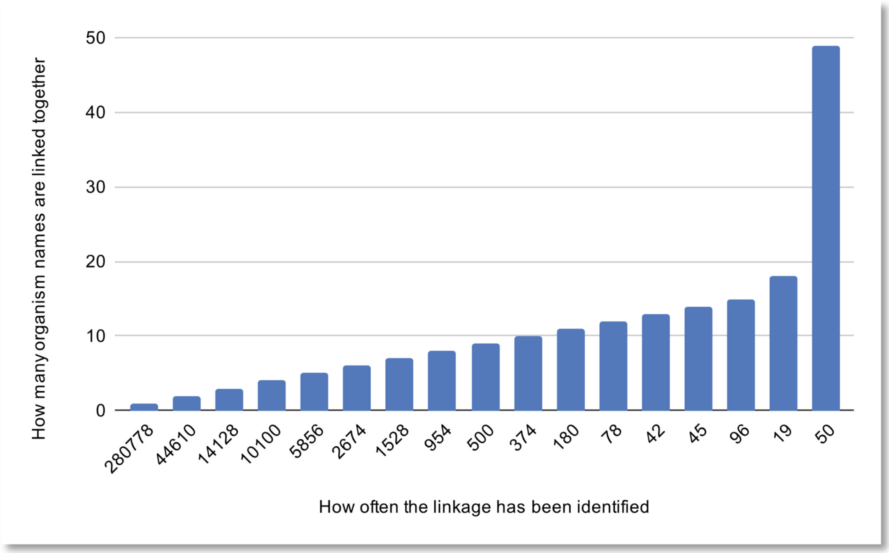


Fig. 5. The histogram depicts the linkage between organism names, showcasing the variability in how frequently particular organism names are linked with other names within one group.

The @ManyToMany Spring annotation was used for linkage storage of the same type of object, but Spring Data JPA retrieval process was very slow, because the Spring Data JPA usage not only filters and retrieves data from the database, but also loads processed data into the collection using many-to-many linkage and organizes them. Currently, the number of the linkages per bibliography unit is in range from 1 to 10,006 (avg). In the advanced-case scenario, when 10,006 names of organisms are linked to one specific publication, there were problems retrieving data from the database in optimal time using Spring Data JPA. The chosen solution was to manage data filtering and retrieving with MySQL database procedures instead of usage of the Spring Data JPA framework. More than thirty database procedures in the database side and relevant interfaces in the back-end side were created for data filter functionality, thus improving the retrieval of linked names of organisms more than ten times. To optimize data transfer from the back-end to the front-end, the data was split per page, where five linkage groups were organized in each page. Each linkage group may contain a different number of linkages. Although, in the usage of database procedures, it was necessary to use a self-developed paginator, instead of Spring Pagination. To optimize data retrieval from the database, advanced

Spring Data JPA methods were also used. In this case, specifications with builder criteria were applied to process results of the API requests dynamically. Specifications allow the usage of pre-stored query parts that can be combined at runtime in different combinations. To ensure the speed of data transfer between front-end and back-end of the IMDS, the Data Transfer Object (DTO) classes were created including necessary data for the specific controller endpoint, thus ensuring more optimized bandwidth throughput and IMDS performance. Also, the IMDS was designed not to use multiple requests, which may be required to gather all the data to render a view on the front-end side. In this case, paginator was used. Optimization of requesting unimportant data at back-end side was done, to decrease computation time and unnecessary load across the back-end to the database.

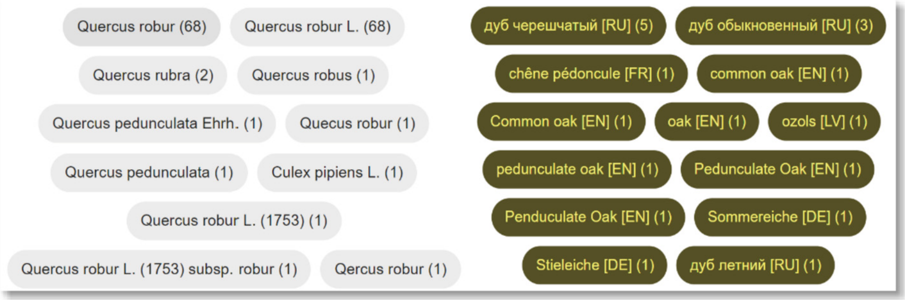
3 Application of IMDS

The objective of this case study is to demonstrate how IMDS operates in terms of functionalities, its user interface, and data visualization, showcasing its effectiveness in facilitating language research and comparative linguistic studies.

IMDS offers a range of functionalities tailored to the specific needs of the previously mentioned requirements, for example:

1. Data Storage: On 12.07.2023, IMDS provides a secure and efficient data storage solution including 149,000 entries of names of different organisms from more than 8,470 publications (statistics in August 2023).
2. Statistical Analysis: The system incorporates a statistical tool to perform quantitative analyses, including frequency counts, lexeme similarity measures and linguistic pattern identification (see Fig. 7).
3. Multilingual Search: IMDS supports advanced search options, enabling researchers to retrieve specific entries across multiple languages, facilitating cross-linguistic comparisons (see Fig. 6 on the right side).
4. Comparative Studies: IMDS allows researchers to conduct in-depth comparative analyses, identifying linguistic patterns and variations among different languages (see Fig. 6 and Fig. 8). It is possible to compare multiple definitions of terms for linguistic and translation studies (see Fig. 9) [13].

The next example shows the *Quercus robur* species name in Latvian ‘parastais ozols’, ‘common oak’ in English. This name is referenced in 140 publications from 1950 to 2023.



Quercus robur (68)
Quercus robur L. (68)
Quercus rubra (2)
Quercus robus (1)
Quercus pedunculata Ehrh. (1)
Quecus robur (1)
Quercus pedunculata (1)
Culex pipiens L. (1)
Quercus robur L. (1753) (1)
Quercus robur L. (1753) subsp. robur (1)
Qercus robur (1)

дуб черешчатый [RU] (5)
дуб обыкновенный [RU] (3)
chêne pédoncule [FR] (1)
common oak [EN] (1)
Common oak [EN] (1)
oak [EN] (1)
ozols [LV] (1)
pedunculate oak [EN] (1)
Pedunculate Oak [EN] (1)
Penduculate Oak [EN] (1)
Sommereiche [DE] (1)
Stieleiche [DE] (1)
дуб летний [RU] (1)

Fig. 6. On the left, there are all linked scientific names of ‘parastais ozols’, along with the corresponding frequency of mentions in IMDS. On the right, there are linked local names of ‘parastais ozols’ with the option to view the frequency of mentions in IMDS for each linked name. Additionally, language filters can be applied to refine the search and analysis.

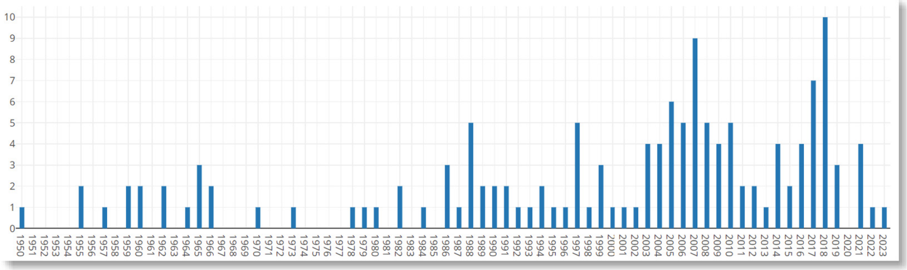


Fig. 7. Timelines depicting the usage of the searched species name ‘parastais ozols’ within publications covered by IMDS

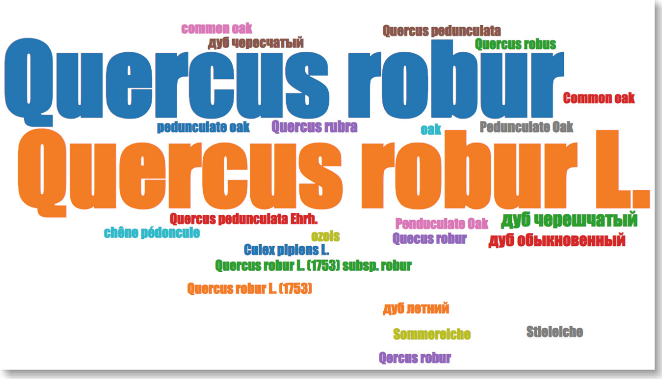
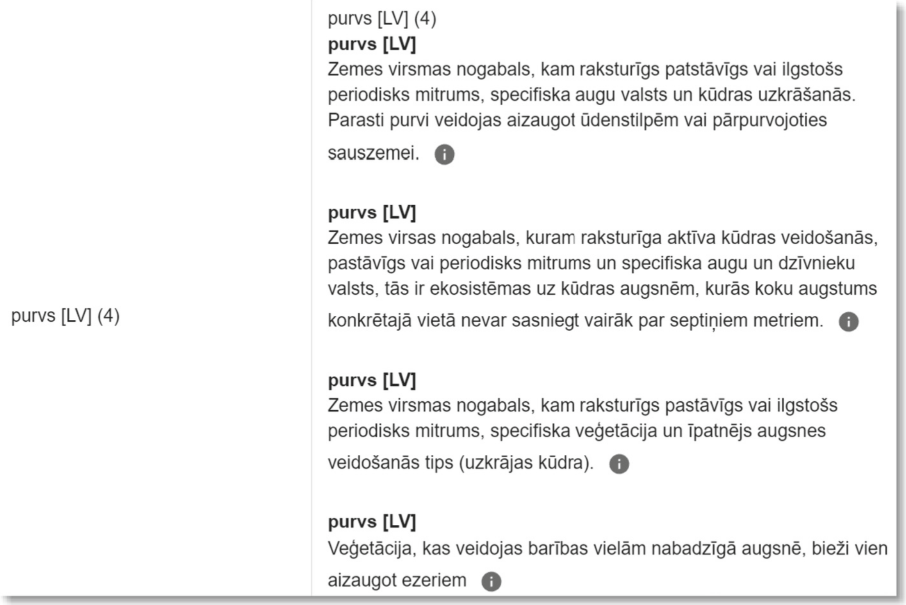


Fig. 8. Word cloud depicting the searched species name ‘parastais ozols’ within publications covered by IMDS.



purvs [LV] (4)

purvs [LV] (4)

purvs [LV]
Zemes virsmas nogabals, kam raksturīgs patstāvīgs vai ilgstošs periodisks mitrums, specifiska augu valsts un kūdras uzkrāšanās. Parasti purvi veidojas aizaugot ūdenstilpēm vai pārpurvojoties sauszemei.

purvs [LV]
Zemes virsas nogabals, kuram raksturīga aktīva kūdras veidošanās, pastāvīgs vai periodisks mitrums un specifiska augu un dzīvnieku valsts, tās ir ekosistēmas uz kūdras augsnēm, kurās koku augstums konkrētajā vietā nevar sasniegt vairāk par septiņiem metriem.

purvs [LV]
Zemes virsmas nogabals, kam raksturīgs pastāvīgs vai ilgstošs periodisks mitrums, specifiska veģetācija un īpatnējs augsnes veidošanās tips (uzkrājas kūdra).

purvs [LV]
Veģetācija, kas veidojas barības vielām nabadzīgā augsnē, bieži vien aizaugot ezeriem

Fig. 9. Definitions extracted from the IMDS system for the term ‘purvs’ (in English ‘bog’).

4 Conclusion and Future Work

Since 2021, linguists, terminologists, translators and IT specialists have been working together to design the IMDS during the implementation of the Project ‘Smart complex of information systems of specialized biology lexis for the research and preservation of linguistic diversity’. After almost twenty four months, members of the grant have used developed models of the IMDS without significant problems. On 12.07.2023, overall 62,836 scientific and 81,137 local names of organisms, 1,657 names of diseases caused by organisms, 2,996 dictionary words, 403 terms and 361,598 linkages in 8,471 bibliography units (656 monographs and 7,815 papers) are stored during this time not only manually by users, but also using our developed tool, where scanned books are processed automatically with Optical Character Recognition (OCR) and language recognition algorithms [14].

Design of the IMDS was a challenging task to ensure effective performance in all IMDS operations because of the data linkages in multiple directions, for example, organisms’ name linkage to bibliography and linkage to other names of organisms). Optimization of IMDS performance was mainly achieved by creating MySQL database procedures and DTO classes, ensuring optimal data transfer from the front-end to the back-end and database. The good practice to increase web system performance is to split functionality in two parts—back-end and front-end and run each system on individual hardware. In this case, IMDS back-end was developed using Java and Spring Framework and front-end—using ReactJS.

As part of system enhancements, search optimization can be implemented using a machine learning algorithm, for example, Word2vec [15] which is a technique for natural language processing (NLP) that utilizes a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence. Currently, the Levenshtein algorithm [16] is used for searching similar lexemes, but its performance could be optimized.

The use of old printed books remains crucial for studying organism names in the past and their transformation through ages. This data collection could increase the importance of IMDS data repository. We developed a methodology [17] for digitizing Old Latvian Orthography using the Tesseract machine learning algorithm, with the main focus on G. H. Kawall's book 'Dieva radījumi pasaulē' (God's Creatures in the World), published in 1860 and translated from German into Latvian. Our digitization process achieved an approximate accuracy of 83%. However, manual verification is still required before integrating this data collection into IMDS. The optimization of model training can be continued. In the future, it would be necessary to involve volunteers to continue entering data into the system. Data input from printed and not digitized books in the Old Orthography is definitely needed in order to implement historical studies and to trace the very first publication of different organism names.

Acknowledgements. This research has been funded by the Latvian Council of Science, project 'Smart complex of information systems of specialized biology lexis for the research and preservation of linguistic diversity', No. Lzp-2020/1-0179.

References

1. Rudziša, V.: Ekvivalences problēmas juridisko tekstu tulkojumos. [Equivalence Problems in Translations of Legal Texts]. In: Aktuālas tendences terminoloģijas teorijā un praksē. Rakstu krājums. Sast. I. Pūtele. [Topical Trends in the Theory and Practice of the Terminology, Collection of the Articles, Pūtele, I. (ed.)] Rīga: LU Latviešu valodas institūts, pp. 17–27 (2013)
2. World Flora Online. http://www.worldfloraonline.org. Accessed 06 Aug 2023
3. Plants of the World Online. https://powo.science.kew.org/. Accessed 06 Aug 2023
4. All Animals A-Z List. https://a-z-animals.com/animals/. Accessed 06 Aug 2023
5. Skosmos, NLL subject authority ontology ('LNB tematisko priekšmetu ontoloģija' in Latvian). https://dati.lnb.lv/onto/nllsh/en/page/LNC10-000047992. Accessed 06 Aug 2023
6. Stalažs, A., Šķirmante, K., Sviķe, S., Jasmonts, G., Ziediņš, R.E.: Experience of design and development of a new open access interactive multifunctional database management system for special lexis of biology. Studies about Languages/Kalbų studijos **42**, 52–67 (2023). https://doi.org/10.5755/j01.sal.1.42.33107
7. Lui, M., Gray, M., Chan, A., Long, J.: Introduction to core spring framework. In: Pro Spring Integration. Apress, Berkeley, CA (2011). https://doi.org/10.1007/978-1-4302-3346-6_3
8. Singh, A., Piyush, C., Singh, K., Singh, A.: Formulating an MVC framework for web development in JAVA, pp. 926–929 (2018). https://doi.org/10.1109/ICOEI.2018.8553746
9. Mukthapuram, R.: Analysis of Component Libraries for ReactJS. IARJSET. **8**, 43–46 (2021). https://doi.org/10.17148/IARJSET.2021.8607
10. Győrödi, C., Dumşe-Burescu, D., Zmaranda, D., Győrödi, R., Gabor, G., Pecherle, G.: Performance analysis of NoSQL and relational databases with CouchDB and MySQL for application's data storage. Appl. Sci. (2020). https://doi.org/10.3390/app10238524

11. Győrödi, C.A., Dumşe-Burescu, D.V., Zmaranda, D.R., Győrödi, R.Ş: A comparative study of MongoDB and document-based MySQL for big data application data management. Big Data Cogn. Comput. **6**, 49 (2022). https://doi.org/10.3390/bdcc6020049
12. Chen, S., Tang, X., Wang, H., Zhao, H., Guo, M.: Towards scalable and reliable in-memory storage system: a case study with Redis, 1660–1667 (2016). https://doi.org/10.1109/TrustCom.2016.0255
13. Ozola-Ozoliņa, L., Sviķe, S.: Latvian environmental term *purvs*: problems and solutions in Latvian-English-Latvian translations of helonyms. Studies about Languages/Kalbų studijos, **42**, 80–92 (2023). https://doi.org/10.5755/j01.sal.1.42.33113
14. Jasmonts, G., Sviķe, S., Šķirmante, K.: New information extracting and analysis methdology for the terminology research purposes: the field of biology. In: CEUR Workshop Proceedings, vol. 3160. IRCDL 2022: 18th Italian Research Conference on Digital Libraries (2022). https://ceur-ws.org/Vol-3160/paper2.pdf
15. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of Workshop at ICLR (2013)
16. Haldar, R., Mukhopadhyay, D.: Levenshtein distance technique in dictionary lookup methods: an improved approach. In: Computing Research Repository – CORR https://doi.org/10.48550/arXiv.1101.1232 (2011)
17. Šķirmante, K., Jasmonts, G., Sviķe, S.: Digitisation of Printed Books in Old Latvian Orthography for the Preservation and Sustainability of Cultural Heritage: Workflow and Methodology Digital Humanities in the Nordic & Baltic Countries (DHNB2023): Sustainability, Environment, Community, Data March 8–10 (2023)