

Design and Implementation of Advanced Web-based Dictionary System for Ukrainian and German Languages

Rustam Gamzayev^a, Tilmann Reuther^b, Mykola Tkachuk^a, Vitalii Filatov^a

^a V.N. Karazin Kharkiv National University, Kharkiv, Ukraine

^b University of Klagenfurt, Austria

Abstract

The actuality of the problem how to design and to develop advanced solutions in the domain of electronic dictionary systems (EDS) is motivated, in particular, with respect to the such linguistic phenomena as lexical parallels in Ukrainian and German languages. The briefly overview of some existing domestic and international EDS solutions is done, and some their specific features and shortcoming are analyzed. The own framework for EDS development is proposed, which supposes a data-centered system design approach, where the specific noSQL data model for the dictionary entry's representation with its existing lexical parallels is presented. The microservices system software architecture is elaborated in order to increase system's availability and scalability. The special attention is paid to the problem how to design and to implement the advanced UI solutions for the target Web-based EDS. For its quality assessment the enhanced version of the *Goals, Objects, Methods and Selection rules* (GOMS) model was presented and applied successfully (titled as the GOMS+), especially the final quality estimation shows the increasing about 12,35%. The obtained test results shown the acceptable functionality assessment of the elaborated Web-based EDS, and it allows to make the positive conclusions about the proposed approach, and to define the further steps to be done in this research.

Keywords 1

dictionary, lexical parallel, noSQL data model, microservice architecture, user interface, GOMS model

1. Introduction: Problem actuality and research aims

Nowadays there are a large variety of IT – solutions for electronic dictionary systems (EDS) in different application domains. The most popular today are the Web-based EDS solutions that allow translation of words and retrieval of dictionary entries directly in the browser. Even the briefly observation of some recent publications in this domain shows the main research and applied trends in EDS development and operation. In particular, in [1] an initial document classification is stated as a key functionality component that provides ability to advanced information search among EDS articles. One of the ways to provide an intelligent search is extracting of domain keywords from documents, and the authors in [1] propose the model for ontology-based keyword extraction to increase the accuracy of document classification. In [2] the intelligent EDS is presented and studied, how to design and to implement such applications according to the tasks and issues to be resolved in Internet of Things (IoT) domain, and in [3] it pointed to the fact that we already have been the Digital era where it is needed for applied linguists and interpreters to acquire the useful knowledge and skills for writing and publishing their e-dictionaries. The linguistic profession tends to be a bit updated to understand that information technologies (IT) can be helping in creating personal templates and libraries.

Basing on the abovementioned issues it can be stated, that one of special and enough important challenges in the domain of EDS development is how to provide more intelligent and sophisticated

ITTAP'2022: 2nd International Workshop on Information Technologies: Theoretical and Applied Problems, November 22–24, 2022, Ternopil, Ukraine

EMAIL: rustam.gamzayev@karazin.ua (A. 1); tilmann.reuther@aau.at (A.2); mykola.tkachuk@karazin.ua (A. 3); xa11867727@student.karazin.ua (A. 4)

ORCID: <https://orcid.org/0000-0002-2713-5664> (A. 1); <https://orcid.org/0000-0002-4475-9014> (A. 2); <https://orcid.org/0000-0003-0852-1081> (A. 3); <https://orcid.org/0000-0002-8673-216X> (A. 4)



© 2022 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

functionality of these software solutions, in particular, taking into account the phenomena of lexical parallels in different languages, and these problems are discussed intensively in such works as, e.g., [3,4]. The illustrative example of this approach is shown in Figure 1 as the dictionary item with the lexical parallels in Ukrainian and German languages which is represented in [4]. The authors [4] define a lexical parallel is a structure included at least two phonetically/graphically similar lexical items of the same part of speech in two synchronously compared languages. If the such semantic structures, i.e. all meanings of these terms agree completely, such a lexical parallel is called as a complete lexical parallel. If even one of these meanings differs, then it is an incomplete lexical parallel, and if the lexical items of phonetically/graphically similar lexical items of the same part of speech differ in all meanings, then it is a case of so-called false lexical parallel.

<p>⚠ АБИТУРІЄНТ, -а, ч: той, хто вступає до вищого або середнього спеціального навчального закладу: <i>заява абітурієнта</i> - Bewerber(in) um die Zulassung zu einem Studium. * * * * *</p>	<p>⚠ Abiturient, m (-en, -en): випускник середньої школи: <i>die Abiturienten feiern den Schulabschluss.</i></p>
<p>АБСОЛЮТН/ИЙ, -а, -е: 1. безвідносний, узятий поза зв'язком, без порівняння з чим-н.; безумовний: <i>абсолютна маячня</i>; 2. цілковитий, повний; необмежений; досконалий, довершений: <i>абсолютна монархія</i>; 3. <i>матем.</i> про величину без знаків плюс або мінус: <i>абсолютна величина</i>; 4*. цілковито чистий, без домішок: <i>абсолютний спирт</i> - rein. * * * * *</p>	<p>absolut, <i>adj</i>: 1.= абсолютний 1: <i>absolute Höhe</i>; 2.= абсолютний 2: <i>absolute Monarchie, absolutes Halteverbot</i>; 3.= абсолютний 3: <i>absoluter Wert (einer Zahl)</i>.</p>

Figure 1: The example of a Ukrainian-German dictionary article's structure with the lexical parallels

More detailed the linguistic issues of this approach are discussed in [4], and the goal of this research is to present the most important issues of the system design, data modeling and software development needed to implement the proposed Web-based EDS with lexical parallels concept, and we abbreviate this EDS as the eUGDLP (electronic Ukrainian German Dictionary for Lexical Parallels).

The rest of our paper is structured in the following way: Section 2 provides the short overview of some existing EDS; Section 3 presents the main project artifacts developed: the data model to store dictionary entries with lexical parallels, the functional requirements specifications and microservice system architecture, and the intelligent user interface layout and the quantitative metrics to evaluate its quality; Section 4 shows the method to system testing, and discusses the obtained results. The paper concludes with Section 5 giving the brief summary and outlook on future work, Section 6 includes our acknowledgements, and Section 7 provides the list of the information sources used in this research.

2. Related work: Some solutions in the domain of EDS development

As it is mentioned above, Web-based EDS already are used intensively, and in order to implement own solution with an advanced functionality and higher quality, the existing ones were investigated to define common functionality, some advantages and shortcomings, to be taken into account in their further design and implementation.

In Figure 2 the main interface of “Ukrainian academic dictionary” website is shown [5], and it is to mention that it was the digital implementation of the first explanatory dictionary of the Ukrainian language.

The user interface (UI) of this dictionary contains the following elements:

- Menu with buttons for sign in and sign up.
- Input field for a word, hints appear during entering a word.
- Checkbox to search entered text in word definitions.
- Information about dictionary.
- List of last viewed words.

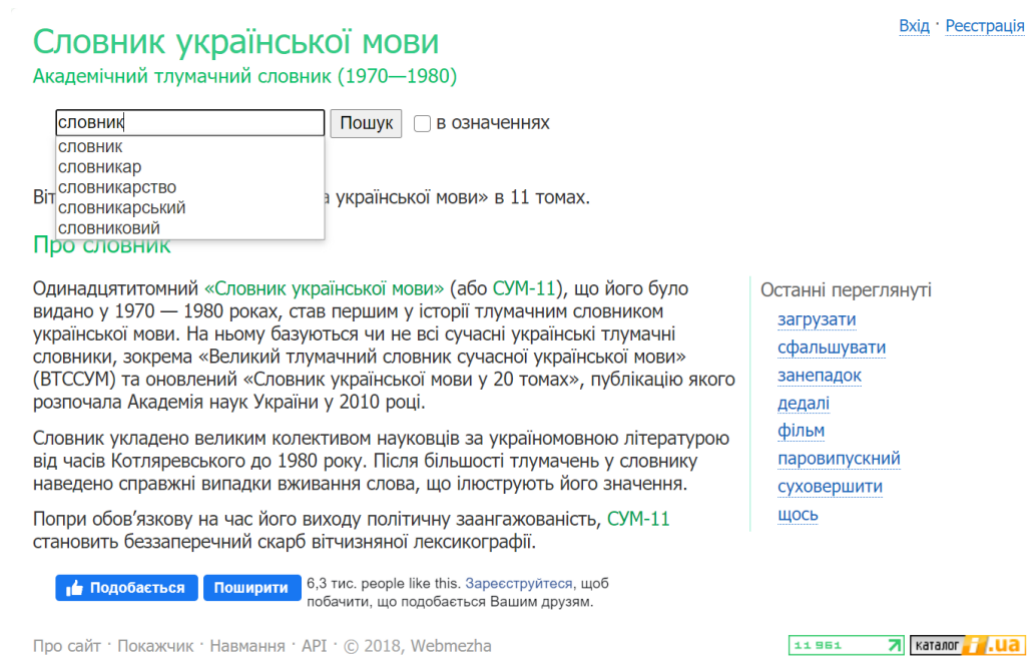


Figure 2: Web user interface of “Ukrainian academic dictionary”

The next observed dictionary “СЛОВНИК.ua” [6] is based on the same paper dictionary, but it has additionally 21000 words meanings that were added by the dictionary users, and it contains the official “passport” (CMU 2010) for online transliteration (see in Figure 3 in Ukrainian language).

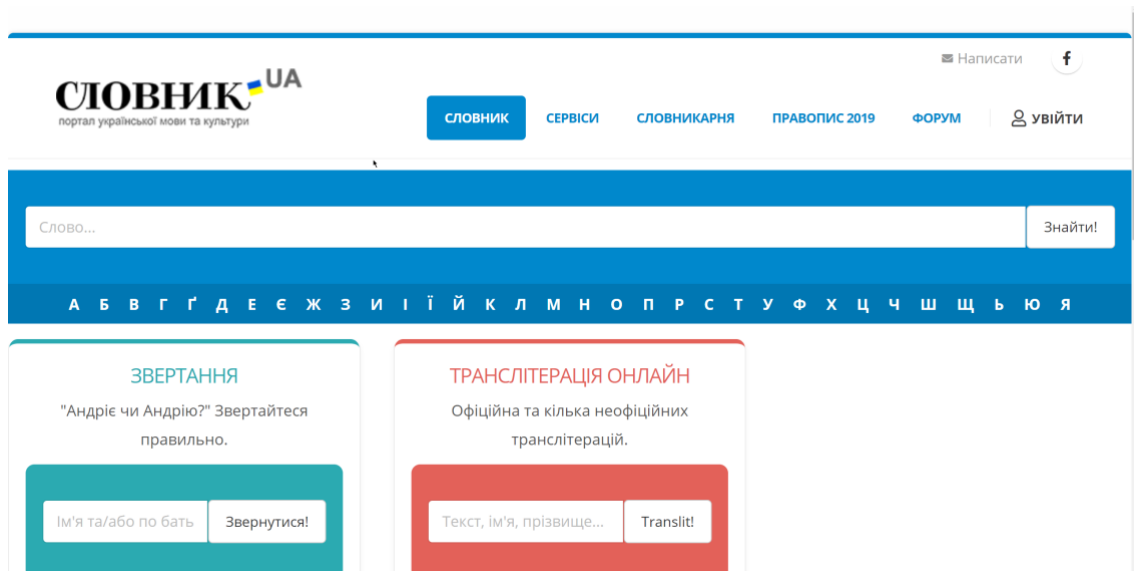


Figure 3: Web user interface of “СЛОВНИК.UA”

This UI consists of the next elements:

- Menu with links to other project services and link to sign in page.
- Input field for a word.
- Alphabet to search dictionary articles by word’s first letter.
- Other service components.

In Figure 4 the home page of “УКРЛІТ.ORG” website is shown. It includes 3 separate dictionaries: Ukrainian academic dictionary, Grinchenko’s dictionary and Zhaivoronka’s dictionary [7]. All their UI include the similar elements: menu with links to another project services, input field for a word, dropdown with dictionaries, alphabet to search word articles by first two letters, dictionary switching.

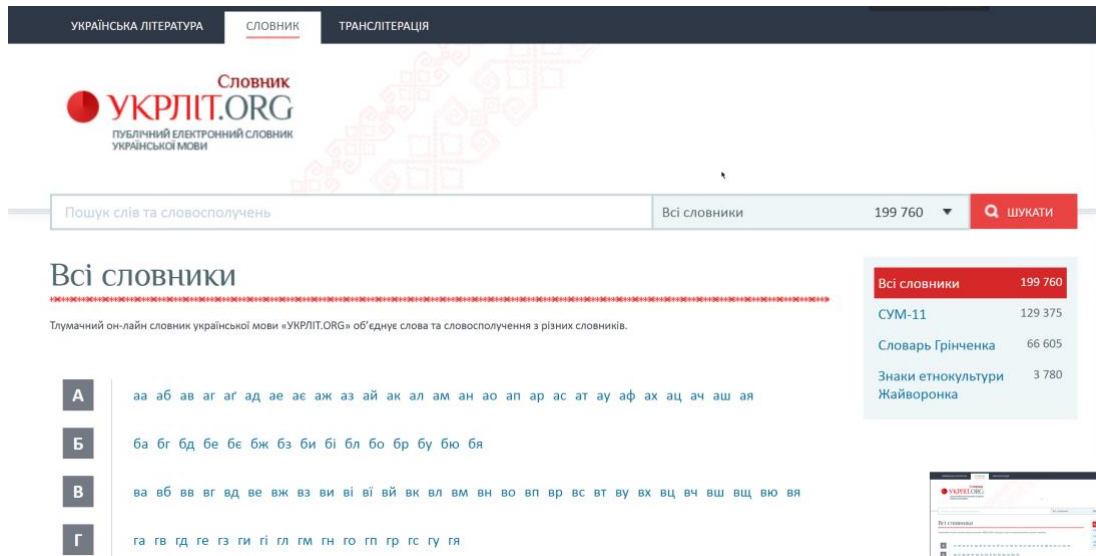


Figure 4: Web user interface of “УКРЛІТ.ORG”

Additionally, three recognized international EDS were analyzed briefly. The first one is the “DICTIONARY.COM” [8], and its UI consists of the next elements (see in Figure 5):

- Menu with links to other project services and link for authentication.
- Dropdown with dictionaries.
- Input field for a word, hints appear during entering a word.
- Trended words.
- Word of the day.
- Other services components.

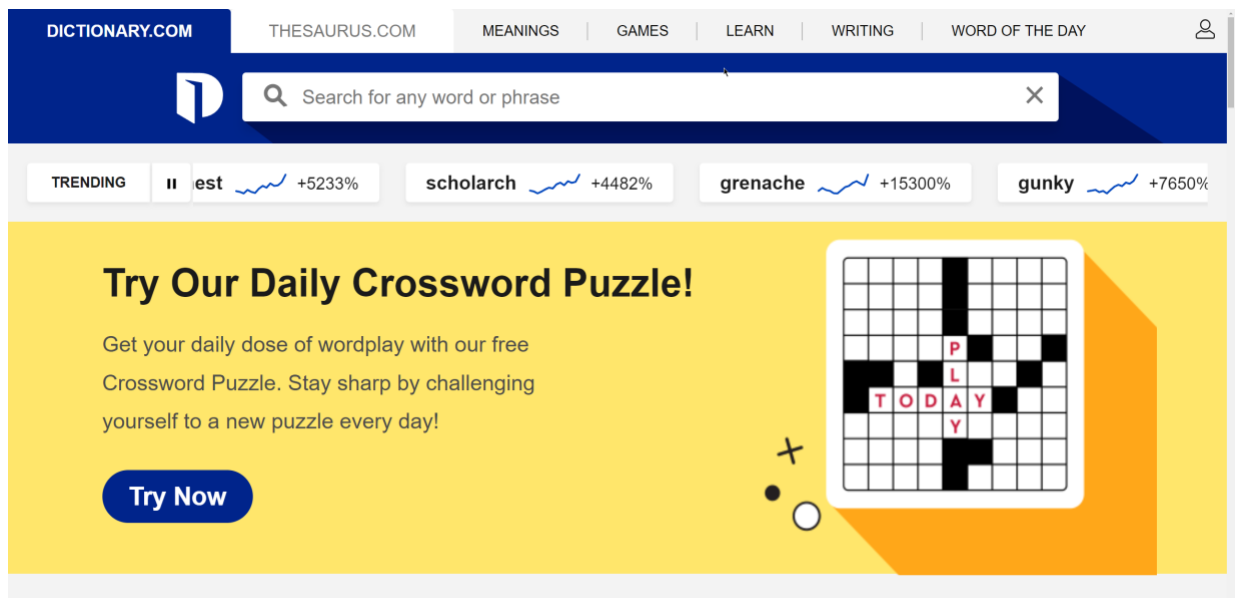


Figure 5: Web user interface of “DICTIONARY.COM”

In Figure 6 the home page of “Cambridge Dictionary” website is shown [9]. It has meaning dictionary for English language and translation dictionary that allows translation to multiple languages. The UI

consists of the following elements: menu with the links to other project services, the links to the selected social networks, the links for authentication option, the dropdown menu with languages, the input field for a word, hints appear during entering a word, dictionaries select button, and the list of dictionaries.

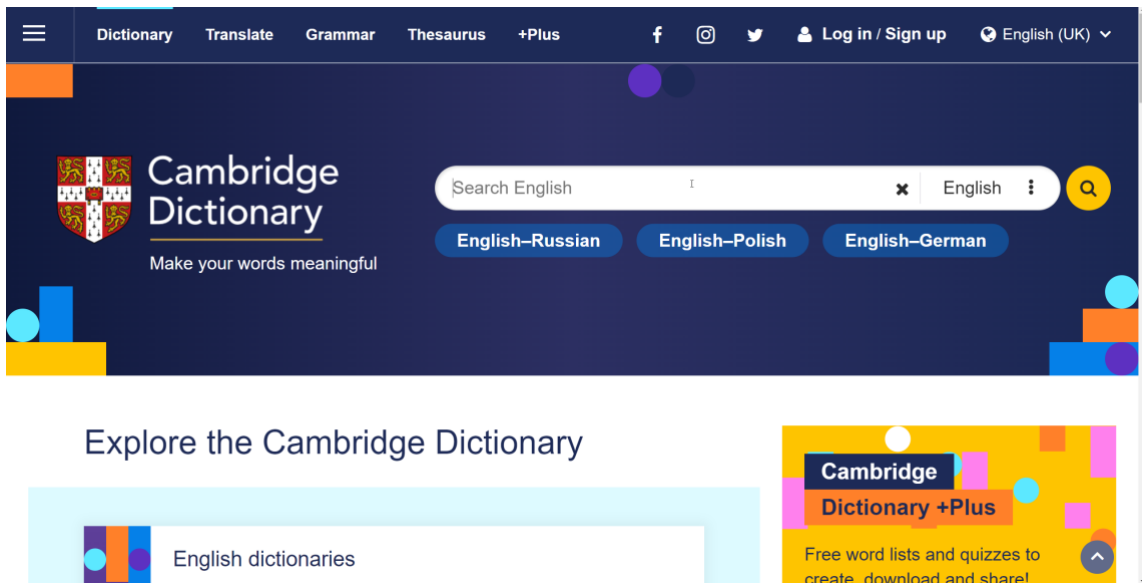


Figure 6: Web user interface of “Cambridge Dictionary”

The “Macmillan Dictionary” website [10] is shown in Figure 7. Its UI consist of the next elements: menu with links to another project services, input field for a word, hints appear during entering a word, links to other project services.

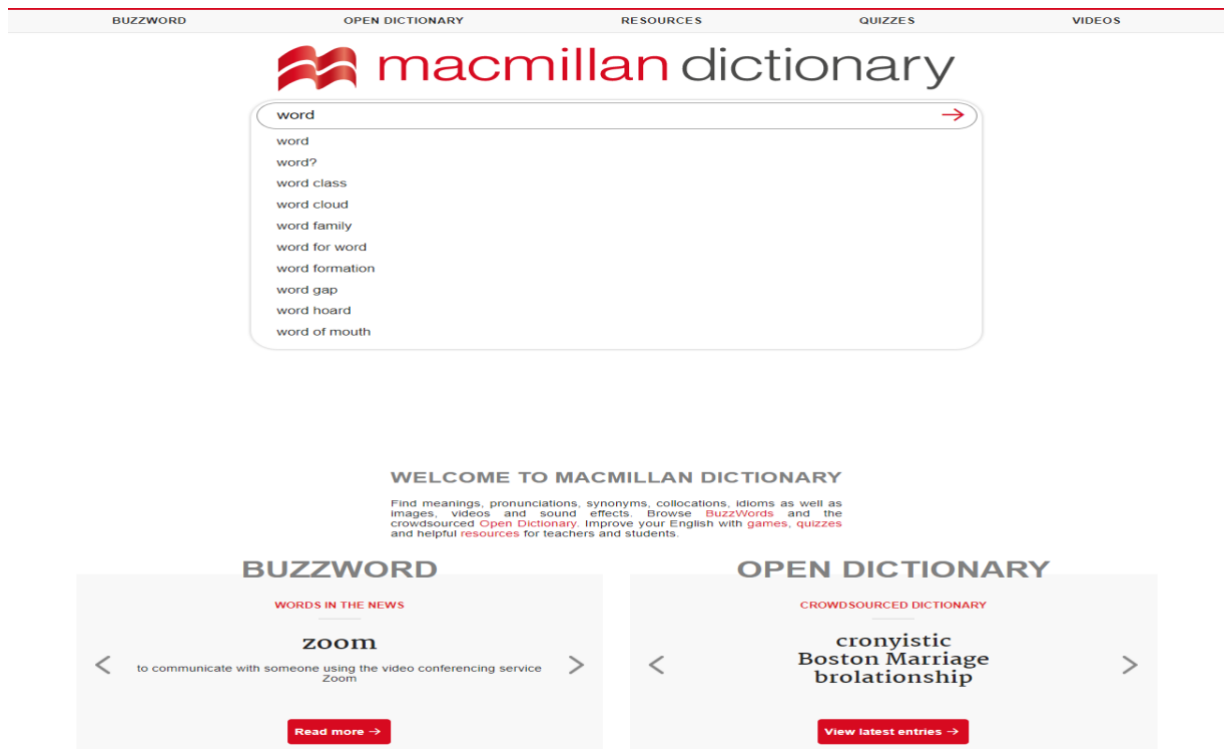


Figure 7: Web user interface of “Macmillan Dictionary”

Based on performed investigate we can make conclusion about the set of typical UI - options that present the main functionality of the observed Web-based EDS, namely:

- menu bar that contains some controls for user’s authentication and navigation,

- input fields to customizing search for dictionary articles,
- another services components (if there are any), e.g.controls for dictionary selection, alphabet to search articles by first letter(s) of word.

At the same time as some common shortcomings of the EDS / UI presented above the following points can be mentioned:

1. almost all reviewed solutions provide ability to search articles by entering full word or its part into a search input fields, but only some of them provide ability to list articles by word's first letter. This functionality should allow users to explore dictionary.
2. all reviewed solutions provide simple search by exact match of substring. It means that if word has alternation of letters, like in Ukrainian language, or different form of word formatted by changing it parts (prefixes, suffixes, endings, etc.) then it would be harder to find a target article.

The proposed EDS solution with the lexical parallels (see the example in Figure 1) is supposed to provide the search functionality that takes into account the different forms of an appropriate word.

3. System Development Issues: Data-Centered Design, Service-oriented Architecture, and Intelligent User-Interface Layout

3.1. Data modeling for dictionary entry with lexical parallels

It is necessary to define the appropriate data model, that will be processed by software solution, before start to design graphical UI and implement its functionality. The dictionary article's structure with linguistic parallel discussed in [4] was analyzed to build a target data model, and this result is provided as UML (Unified Modeling Language) class diagram [11] in Figure 8.

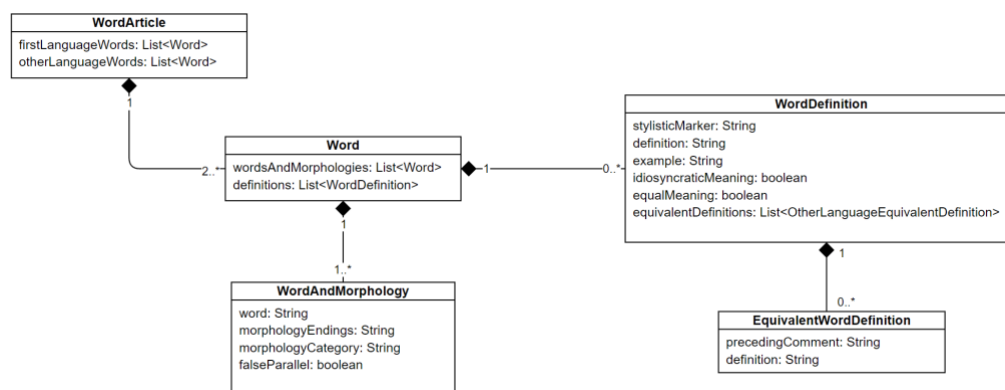


Figure 8: Dictionary article data model

This data model contains the next entities: *WordArticle*, *Word*, *WordAndMorphology*, *WordDefinition* и *EquivalentWordDefinition*. *WordArticle* entity represents whole dictionary article and consist of two fields: *firstLanguageWords* – list of first language words and field *otherLanguageWords* – list of other language words. Article should contain at least one word from first language and one from another.

Information about word, or words when similar spelled synonyms are provided, and definitions represented by *Word* entity. Class *Word* contains of two fields: *wordsAndMorphologies* – represent list of words and its morphology parts and field *definitions* – list of definitions for current word(s).

Class *WordAndMorphology* contains data about word (field *word*), word endings (field *morphologyEndings*), word category (field *morphologyCategory*) and false parallel flag (field *falseParallel*).

Class *WordDefinition* represents definition of the word(s) and contains information about: stylistic marker (field *stylisticMarker*), word(s) definition (field *definition*), example of word usage (field *example*), flag of semantic difference from other language word definitions (field *idiosyncraticMeaning*), flag of equality with definition from first language word(s) (field *equalMeaning*), list of equivalent word or definitions from second language (field *equivalentDefinitions*).

The equivalent words and definitions are represented by class *EquivalentWordDefinitions* that contains information about comment to word or definition (field *precedingComment*), and word or definition itself (field *definition*).

3.2. Functional requirements

Basing on the UI analysis in Section 2, the functional requirements for a target EDS was elaborated, and they are represented as the UML use case diagram [11] in Figure 9.

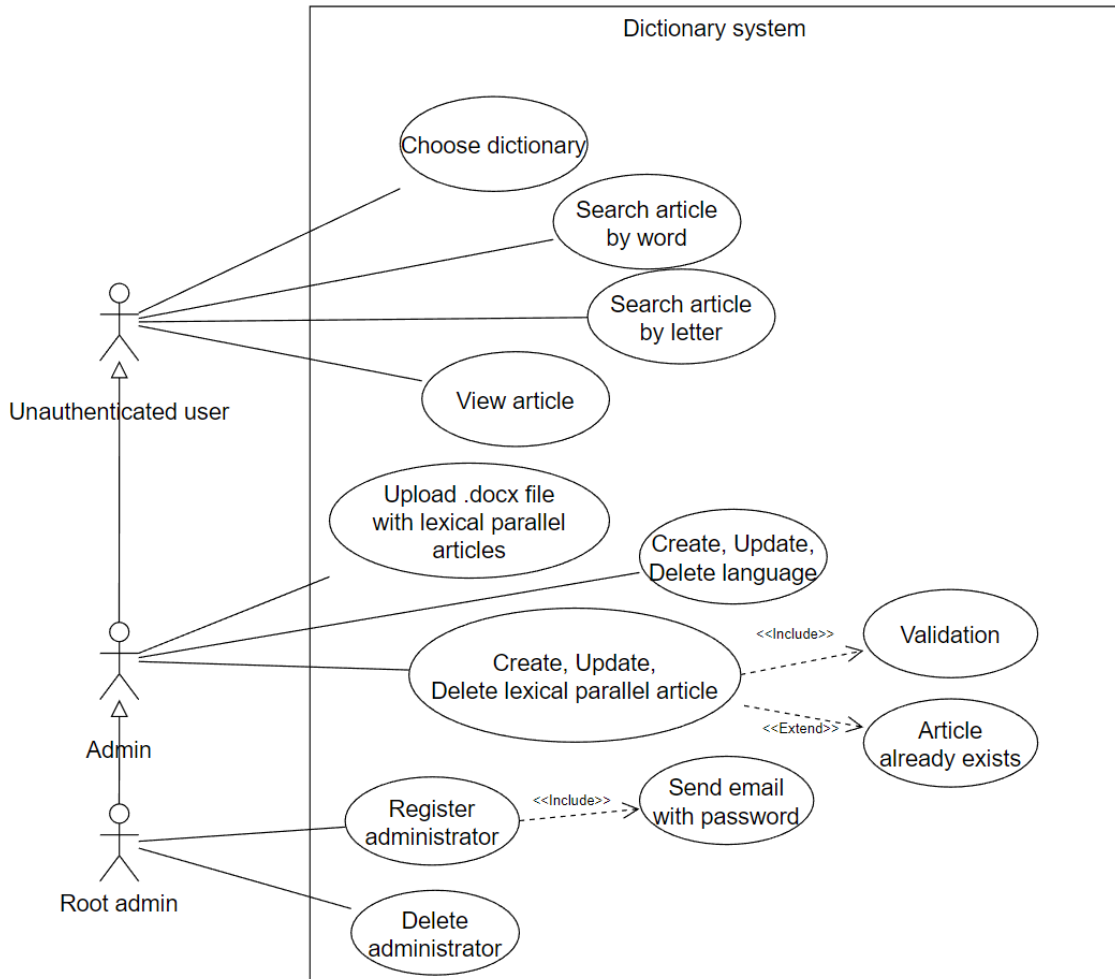


Figure 9: A usage scenario for a eUGDLP in the use cases notation

The designed EDS has three types of users: unauthenticated users, administrators and root administrators. Depending on this type, user has access to different sets of available functions. Unauthenticated users have access to next functionality:

- Choose dictionary by selecting first and second languages.
- Search dictionary articles by word entered to search input field.
- Search dictionary articles by first letter of a word.
- View dictionary article.

The system administrators have access to the same functionality as unauthenticated users and to additional functionality:

- Upload “.docx” format files with dictionary articles to add multiple articles to the system.
- Create, update and delete languages for dictionaries.
- Create, update and delete dictionary articles.

The root administrators have access to the same functionality as administrators and in addition to register new administrators and to remove administrators from the system.

3.3 Microservices architecture solution

There were requirements to consider system for future extension and decouple different pieces of functionality from each other. So, it was decided to use microservice architecture to meet these requirements. The proposed system architecture is shown as UML component diagram [11] in Figure 10.

This diagram consists of the following software components:

- *Client*, in this case client is web application created using React.js library [12], however since software system exposes REST API [13] this means that other web application or mobile application, or other software system could be a client.
- *API Gateway*, implementation of design pattern of the same name, provides one entry point to the system, communication with *Client* and other system components implemented using HTTP.

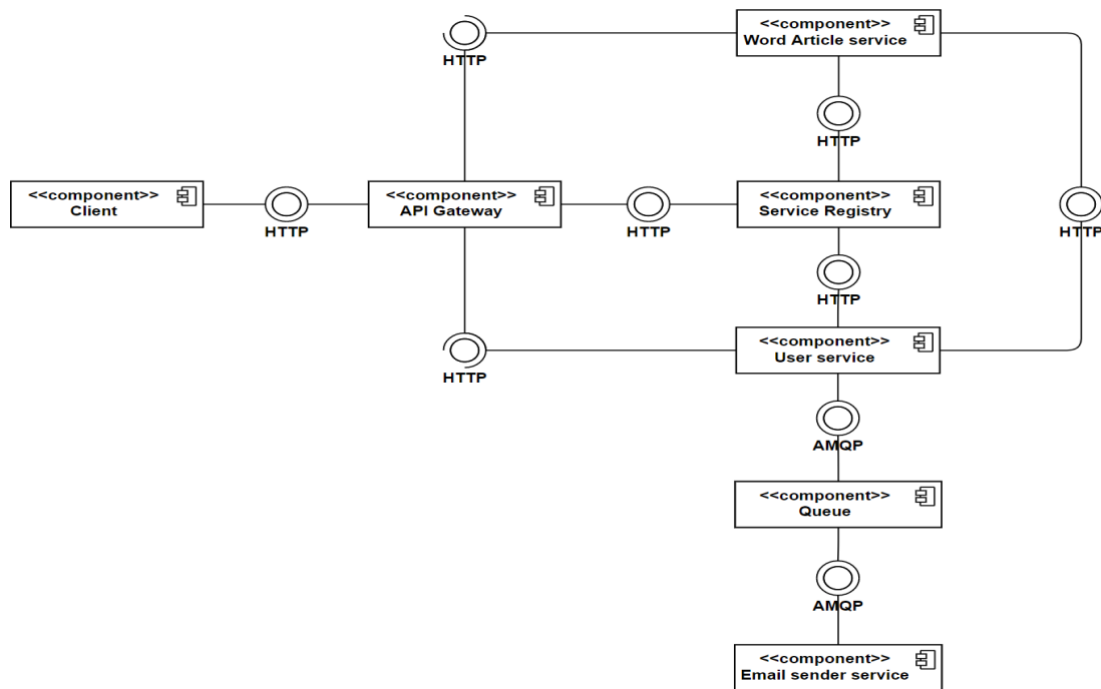


Figure 10: Elaborated EDS solution as a UML component diagram

- *Service Registry*, implementation of design pattern of the same name as well, registers address of active system components (IP address and port) and provides this data to other components for further communication.
- *Word Article service*, manages dictionary articles, provides functionality for unauthenticated users and system administrators. Request authorization is performed by using self-encoded JSON Web Token (JWT) token, that should be passed in *Authorization* HTTP header, when request is received, and token extracted then it's sent to *User service* for validation.
- *User service*, stores information about administrator, root administrators and issued JWT tokens, provides REST API for registering, removing administrators, password updating, registers and removes root administrators, also provides REST API for issuing, updating and validating JWT tokens, after new administrator registration send data about email, that should be sent to the *Queue* component using AMQP [14] protocol.
- *Queue*, message queue, that is used to store information about emails that should be sent.
- *Email sender service*, service for sending emails, retrieves data about emails from *Queue* using AMQP protocol.

As the data storage model for dictionary articles, it was decided to use document-oriented model with MongoDB, since all operations related to articles management operates with whole article object. There is no functionality that updates only some part of article in one place, and it should be updated in other places. Also, there are no complicated queries that would be effective to implement with relation data model.

3.4. Intelligent user interface layout

Considering the listed above shortcomings 1 – 2 of the reviewed existing EDS (see in Section 2), it was decided to add alphabet to the UI of developed solution to implement search by first letter of word, and the text index was created in the system database.

The UI defines how interface will look like. It defines what colors would be used, font styles etc. Graphical interface is a main way to interact with modern information systems. If graphical interface not changed, you could change interaction object without changing interaction principle with it. If user know how to use one system, he could easily learn how to use another system with similar interface. Combination of convenient interface with well-structured navigation increase users' impressions. The main task during any UI design is to simplify its functions, so user can get more results with less effort [15]. The Web-UI was implemented using JavaScript library "React.js" and component library "Material-UI". Dictionary article creation form is shown in Figure 11.

Figure 11: The dictionary article creation form

The all main pages have the following layout:

- Menu with links to other pages.
- Dictionary selection.
- Languages alphabets.
- Page-specific content.

The dictionary article creation form consists of two visual frames: the data entering frame for first language and corresponding frame for second language. If there are two or more words with different list of definitions than the list with these words is present, like for German words in this case; input field for word; input field for morphology endings; input field for morphology category; checkbox that designates that current word has semantically different definitions; button that allows you to add word to current list of definitions; if there are two or more words than button to remove the last one is present; input field for definition; input field for stylistic marker; input field for example; checkbox for equal meaning; button to add second language equivalents; button to remove definition; button to add definition; button to add word; if there are two or more words than button to remove selected word is present; button to save dictionary article. The page with search results is shown in Figure 12.

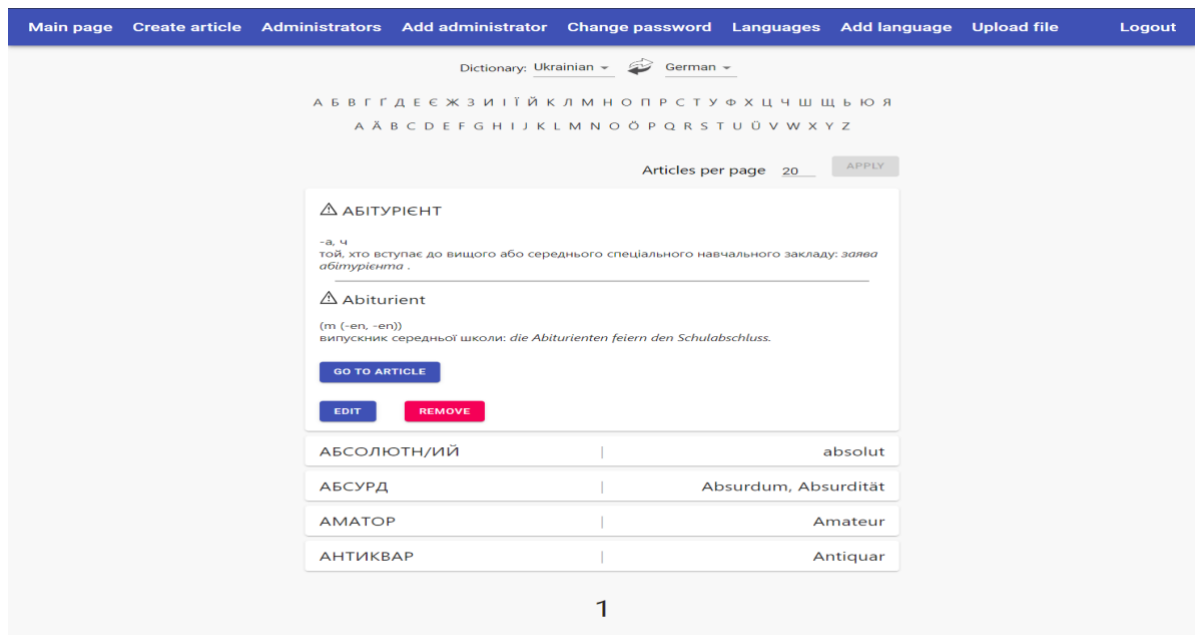


Figure 12: Search results

The page with search results consists of the following visual frames:

- Articles per page component, allows you to specify desired number of articles per page in interval from 10 to 100 elements per page.
- Button to apply entered value. If you change the value than you will continue to view article from the page that contains list of articles displayed before you change number of elements at the page, in other words you won't be redirected to first page.
- Expandable list of articles. When unexpanded only words of article are displayed, in expanded state you can see triangle with exclamation mark inside which represents false parallel flag, first language word, morphology endings and category, first definition. Next you can see corresponding word from second language, button that allows you to view whole article, button that allows administrators to edit article and button that administrators to remove article.

If there are more article that specified "Articles per page" count than at the bottom of results page there would be available links to other pages. On current figure there is only one page.

4. Method for the system testing and analysis of the obtained results

There are different ways to evaluate designed interface, one of them is to invite stakeholders, ask them to use interface to complete list of tasks. Users' interaction with interface is recorded and then analyzed. If you don't have ability to perform such tests, you could use quantitative methods. Such methods allow to replace contentious questions about interface evaluation to simple calculations.

There are some quantitative methods to evaluate interface: the Hick's Law, the Fitts's law, and the model of *Goals, Objects, Methods and Selection rules* (GOMS) [16].

The Fitts's law evaluates efficiency based on the size of the element and the distance from the cursor to its center and shown by the formula (1)

$$T = a + b \log_2 \left(\frac{D}{S} + 1 \right) \quad (1)$$

where T is a mean time spent to perform an action, a is a mean time of starting/ending movement, b – is a value that depends on the typical speed, D is a distance from the starting point to the center of the target, S is a width of the target measured along the axis of motion.

The Fitts's law is usually accompanied by Hick's law, which states that the reaction time when choosing one of several alternatives depends on the number of options. Hick's law is shown by the formula (2)

$$T = a + b \log_2 (n + 1) \quad (2)$$

where T – total reaction time, a – delay before the task, b – individual coefficient of decision-making speed. n – the number of alternative elements from which to choose one.

The GOMS model allows to evaluate UI using time intervals for specific simple actions that have been determined experimentally. GOMS has the next list of actions with its time intervals [16]:

- Key pressing, denoted as “K” = 0.2 s.
- Cursor moving to a specific position, denoted as “P” = 1.1 s.
- Moving hand from mouse to keyboard or vice versa, denoted as “H” = 0.4 s.
- Mental decision of next action, denoted as “M” = 1.35 s.
- Waiting response from system, denoted as “R” (this action depends on a concrete computer system and it should not be used in the calculation).

However, such actions could be extended with respect to the specifics features of a target UI. In our case we are dealing with search results that could contain quite big amount of dictionary articles ordered alphabetically. To navigate throw it is common practice to use a scrolling function. That’s why the standard GOMS model presented above could be extended by the two additional actions

- Scrolling time that is required for navigation to the target element of the UI, denoted as “S” and in average = 1,5 s
- Object focusing time that is required to catch a necessary information from the UI, denoted as “F” and in average = 0.3s

Thus, the new enhanced GOMS+ model allows to provide a more accurate assessment of the EDS / UI quality.

It was decided to use the GOMS model to evaluate the implemented UI, since it provides all necessary values and doesn’t require to define some values in experimental way like Fitts’s and Hick’s laws (see the formulas (1)-(2) above). We will evaluate UI by performing search of word articles that are represents lexical parallel in our eUGDLP facility. Using GOMS rules we will receive next navigation sequence: “HPPKKKKKKKPPHPKMSF”. This means next: first “H” letter means taking computer mouse in the hand; first “P” letter means moving cursor to specific position; first “K” letter means click with mouse; second “H” letter means moving hand from mouse to keyboard; eight “K” letters mean input of eight characters of the word; third “H” letter means moving hand from keyboard to mouse; second “P” letter means moving cursor; last “K” letter means mouse click.

So, we have done the next actions: selected input field using mouse, entered eight characters and pressed search button using mouse. Next, we need to apply the GOMS rules to evaluate navigation sequence. After applying all GOMS+ model rules we receive next sequence “8K6P2H1M1S1F”. Next, we need to map each letter to its time interval and sum all numbers and we will get 12.15 s. This is because in other interfaces you can submit your request by pressing “Enter” key, and in the developed solution you need to press the “Search” button with the mouse. The best result was achieved not only by submitting requests by “Enter” key, but also by setting focus on input field after page is upload, which allows to omit mouse movements to select the input field. Taking into account all the mentioned above, it was decided to implement an autofocus feature on an input field, after a web page is already loaded. Having the such improvements with the new GOMS+ rules, the generated activity sequence would be the next: “6K5P2H1M1S1F”, which is equal to 10.65 s. In the same way we have estimated the UI in the existing EDS described in Section 2. The obtained results are shown in Table 1.

Table 1
Web interfaces evaluation results using GOMS+ after enhancements

Software solution	Initial UI design. Search dictionary article and word meaning (s)	Target UI design. Search dictionary article and word meaning (s)	GOMS+ value
eUGDLP	12,15	10,65	6K5P2H1M1S1F
Ukrainian academic dictionary	10,35	10,35	6K3P3H1M2S1F
СЛОВНИК.УА	10,15	10,15	7K3P2H1M2S1F
УКРЛІТ.ОРГ	11,85	11,85	8K4P3H1M2S1F
Dictionary.com	7,35	7,35	8K2P1H1M1S1F
Cambridge dictionary	7,55	7,55	7K2P2H1M1S1F
Macmillan Dictionary	7,65	7,65	6K2P2H1M1S2F

As it can be seen from the Table 1, the implemented eUGDLP solution has the best evaluated value since it has specific data structure for lexical parallels that allows users to search more efficiently comparing with other solutions. The eUGDLP provides functionality that is not available fast in all considered analogs, like ability to search by input string, or search by alphabet letters that is absent for several analogs. Besides that, analyzing the data presented in the Table 1, it is to mention, that the improvements done in the target UI design version has the better estimation GOMS+ value in compare with the initial one, which can be calculated as follows (approx.): $(1 - 10,65/12,15) * 100\% = 12,35\%$.

Finally, based on the results shown on Table 1, we can state, that the developed eUGDLP solution provides the best usability for lexical parallels than the considered analogs. That's for the front-end functionality, and for the back-end solutions we developed the several microservices that communicate using REST API [13] and AMQP [14], in order to develop software services independently, and to increase system's availability and scalability.

5. Conclusions and future work

In this paper we have motivated an actuality of the problem how to design and to implement the advanced solutions in the domain of electronic dictionary systems (EDS) creation, in particular, with respect to the such linguistic phenomena as lexical parallels in Ukrainian and German languages. The performed overview of some existing domestic and international EDS solutions has shown their specific features and shortcoming as well, that allows us to propose the own framework for EDS development. Within this one the data-centered system design approach was chosen, where the specific noSQL data model was elaborated for the dictionary entry's representation with its lexical parallels. The microservices system software architecture was proposed in order to increase system's availability and scalability. Furthermore, the special attention was paid to the problem to design and to implement the advanced UI solutions for the target EDS, and for its quality assessment the enhanced version of the so-called Goals, Objects, Methods and Selection rules (GOMS) model was proposed and applied successfully (titled as the GOMS+). The obtained test results shown the acceptable quality of the elaborated Web-based EDS, and it allows to make the positive conclusions about the proposed approach.

Further work in this research is supposed to compare the quality assessment results with usage of another UI models, to provide more representative amount and complexity of testing data, and to develop a comprehensive methodology for a performance evaluating of a prospective EDS with respect to the phenomena of lexical parallels in other languages (for instance, English and Polish) for the specific professional groups of their end users.

6. Acknowledgments

The authors would like to thank prof. Volodymyr Dubichynskyi (University of Warsaw, Poland) for his useful advises during the system requirements analysis for the target eUGDLP solution, and bachelor student Bach Viet Yen (V.N. Karazin Kharkiv National University, Ukraine) for her active participation on the software development in this project.

7. References

- [1] Kollé Padmaja, Bhagat Snehal and Zade Shruti. "Ontology based Domain Dictionary". International Conference on Smart City and Emerging Technology (ICSCET), (2018).
- [2] Wenming Yong, "Design and Implementation of Intelligent English Electronic Dictionary System Based on Internet of Things", Wireless Communications and Mobile Computing, (2021). doi:10.1155/2021/5586662.
- [3] Anokhina Tetiana, Kobyakova Iryna and Shvachko Svitlana. Testing Tools for Writing and Publishing E-dictionaries. CEUR Workshop Proceedings, 2870 (2021): 590–600.
- [4] Volodymyr Dubichynskyi, Rustam Ganzayev, Tilmann Reuther. Theoretische Grundlagen und elektronisches Wörterbuch der lexikalischen Parallelen des Ukrainischen und Deutschen // accepted for the publication and to be issued in: Wiener Slawistischer Almanach, Band 88, Berlin: Peter Lang Verlag, 2022.

- [5] Ukrainian academic dictionary, 2018. URL: <http://sum.in.ua/>.
- [6] Slovnyk.ua Portal of Ukrainian language and culture, 2022. URL: <https://slovnyk.ua/>.
- [7] Ukrlit.org Public electronic dictionary of the Ukrainian language, 2021. URL: <http://ukrlit.org/slovnyk>.
- [8] Dictionary.com Meanings and Definitions of Words at Dictionary.com, 2022. URL: <https://www.dictionary.com/>.
- [9] Cambridge Dictionary Make your words meaningful, 2022. URL: <https://dictionary.cambridge.org/>
- [10] Macmillan Dictionary Free English Dictionary and Thesaurus, 2022. URL: <https://www.macmillandictionary.com/>
- [11] I. Sommerville, Software Engineering, 10th Edition, Addison Wesley (2015)
- [12] React – A JavaScript library for building user interfaces, 2022. URL: <https://reactjs.org/>
- [13] REST API Tutorial, 2022. URL: <https://restfulapi.net/>
- [14] AMQP Tutorial, 2022. URL: <https://www.wallarm.com/what/what-is-amqp>
- [15] G. Krishna, The Best Interface Is No Interface: The Simple Path to Brilliant Technology (Voices That Matter), 1st. ed., New Riders Pub (2015).
- [16] J. Marsh, UX for Beginners: a Crash Course in 100 Short Lessons, 1st. Edition, O'Reilly Media (2016).