Преглед НЦД 17 (2010), 1–8

Sanja Životić, Saša Malkov,
Nenad Mitić, Žarko Mijajlović
Matematički fakultet, Beograd

A CULTURAL HERITAGE REGISTER PROTOTYPE

Abstract. In the Republic of Serbia the cultural heritage has been digitized by many institutions and individuals. Although a large number of resources with digitized material of different kinds are available on the Internet, it is not easy to find any specific one. We have developed a prototype of central register of digitized cultural heritage. Our solution is based on a database of digital resources metadata and provides a simple way to locate any kind of digital resources. Resources can be located either by name or by their characteristics. The register can be regularly updated from information sources using XML files.

Keywords: register, catalogue, digitization, cultural heritage

1. Introduction

National cultural heritage consists of social values and physical artifacts that are inherited from past generations. Tangible objects of value are usually held in libraries, museums, galleries, monasteries and similar institutions all over the country. Damage or destruction of those objects would be a huge and unrecoverable loss for both the nation and the mankind. The digitization of cultural artifacts is one way to protect them against loss or damage and to provide simple, safe and wider public availability.

The Ministry of Culture of the Republic of Serbia supports the usage of information technology to preserve cultural identity and to raise consciousness about significance of cultural heritage for society and future generations [1]. Many institutions (including National Museum, National Library of Serbia, Archive of Serbia, Institute for the Protection of Cultural Monuments of the Republic of Serbia and many others) digitize their artifacts. As digitization becomes more common, both the number and the variety of types of digitized cultural resources rapidly grow each year. That makes the problem of finding any specific resource more difficult. There are many general search engines (like Google), which allows the searching of publicly available digital resources. However, the results include data from different sources, which make difficult the recognition of relevant sources in the domain. A central register of digitized cultural heritage could provide a simple way to search for any digital resource in one place. Moreover, the central register could help in synchronizing the digitization efforts. One of the synchronizing goals is to prevent multiple digitization of the same content. Another potential purpose of the register is to act as a tool for organizing and publishing the digitized content.

2. Problem definition and register design

In this paper, the original artifacts are referred to as “content” while digitized artifact representations are referred to as “resources”. Each of the contents may have multiple digital representations. For example, the book “Tsar Radovan's treasure” by Jovan Dučić can be

represented by files in both PDF and DOC formats; similarly the painting “Field of happiness” by Dragan Malešević Tapi can be represented by files in JPEG, BMP or PNG format. There are many factors that justify the presence of multiple resources per content, including image resolution, scanning technique, lighting and many others. The sonverse holds true as well: one digital resource can represent more than one content. For example, one JPEG image can contain multiple paintings, or one PDF document can hold together a writer's photographs and books.

The register is conceptually defined as a list of digital resources reference marks. A register is a genuine register only – it contains no digital resources. It is limited to available metadata and global references to the digital resources and their appropriate content. The digital resources are referenced using URIs, which allows potential register users to reach them at their source. Thus, even if the register itself is not updated with current version of resources metadata, the latest information about digital resources and the appropriate metadata will be available to the users.

There are a lot of different possible types of content. The prototype implementation supports a limited set of basic document types and corresponding properties, but the underlying design is able to cope with new ones. Properties of the resources stored in the register depend on the resources type. In the scope of this paper, discussion is limited to books, magazines, paintings, photographs, and audio/video materials.

Register design process covered problems related to resources and corresponding sources (metadata, source information, collections); security problems (including data security, access policy, user authorities, permitted operations, etc.); availability and performance problems; administration problems (including initially loading, updating, and backup), as well as problems concerned to construction of register user interface.

2.1 Resource attributes and organization. Each registered digital resource is identified by its name and attributes. The number of attributes and their values depend on the resource type. There are some attributes that are common for all resources – like content location (URI address of original artifacts), MIME type of resource, language of the resource description, keywords that are most important when searching for the resources, etc. Some other attributes depend tightly on resource type –for example publisher for the book type or size (width and height) for the picture type. Because of the wide content domain, the register must support metadata specification in different languages. The code of used language is stored according to ISO 639-1 standard. Also, the register allows using different alphabets for the same language (like Cyrillic or Latin for Serbian).

Digital resources are organized into digital collections. One resource can be found in different collections. The book “Tsar Radovan's treasure” from the previous example can be a part of both collections “The most popular books in Serbia” and “Selected works of Jovan Dučić”. Also, a single collection can (and usually does) consist of many resources. As a consequence of this organization, there is a need for separate attributes that describe the resource collections. The most of the attributes hold information that can be interesting to potential users and are subjects for searching and reporting.

2.2 Security and administration problems. Not all digital resources are public. Information sources can have different security policies for data access. Considering the fact that register database contains only metadata and not the actual digital resources, permission policies are defined by information sources and they take effect on the attempt to access resources. That conceptual standpoint allows the prototype to neglect the resources permission policies.

However, the production version should feature some permission managing capabilities, including the presentation of the policies and searching by policies.

There are several kinds of register users: administrators, source representatives and (ordinary) users. Users can have different roles in the system. The system provides user authentication and authorization. Data in the register can be public or private. Any user can search and access all public data in the register. Private data can be accessed by authorized persons only. Administrators can manage data of different information sources, while source representatives are allowed to manage only the sources that they represent.

The problem of initial register load must be solved together with owners of digitized resources. Nowadays, there are no rules (laws, etc.) related to sharing of digitized resources of cultural heritage. As a consequence, loading information in register is dependent on cooperation with resource owners. To provide the current data, register has to be regularly updated. Updating can be done either automatically or manually. Both the automated and the manual updates have their pros and cons. While automatic updates are easier and make the register the most up-to-date, manual updates are better if only some specifically selected resources should be updated. Moreover, manual update allows the source authorities to manage the representative collections of the resources. System administrators and source representatives can manage data in two completely different ways. The first and basic one is by importing metadata from an XML document. System validates the XML document using XML scheme and stores parsed data into register database. Metadata format is flexible and supports specifics of different information sources. The other way of managing register data is by adding, changing and deleting individual data manually in the user interface.

Register administrator is responsible for the backup and recovery. Although the source representatives can provide backup/recovery for their resources, the best solution is to centralize this operation for all registered resources.

2.3 User interface, availability and performance. Access to register can be made either over Web application or over specially tailored application. Web interface offers access to wide group of potential users but also brings additional security problems. The prototype interface is implemented as Web application based on Java. The application supports all necessary levels of security and different levels of authorization for users and administrators.

In the register development, one of the main requirements was to provide a good performance regardless of number of users or number of resources. In production, number of resources can be huge, and the volume of associated metadata can be several gigabytes. Also, the number of concurrent users of Web based user interface can be very high. Relational database is a natural choice for register data storage implementation because it has embedded mechanisms which can handle both concurrent access and data growth.

3. Database design

The register implementation is based on a relational database. The database scheme is presented in Entity Relationship (ER) diagram [2] in Figure 1. Entities on the diagram correspond to database tables. Each table row represents an instance of the appropriate entity. Relationship lines represent the foreign keys. The central entity is *ResourceSource* that represents institutions and individuals that provide information about their content (*Content*), digital resources (*DigitalResource*) and collections (*Collection*). Among other attributes, digital resources are described by their location on the web, size, MIME type (*MimeFormat*) and keywords that are important for searching. Digital collections can contain resources from different sources. One digital resource can be a part of multiple digital collections, so there is

a need for the additional entity (*CollectionMember*) which describes this relation. Contents can have more than one author (*Author*). Bidirectional association between content and author is represented with entity *AuthorContent* that has extra attributes which describe the rank and the role of each author in the process of content creation.

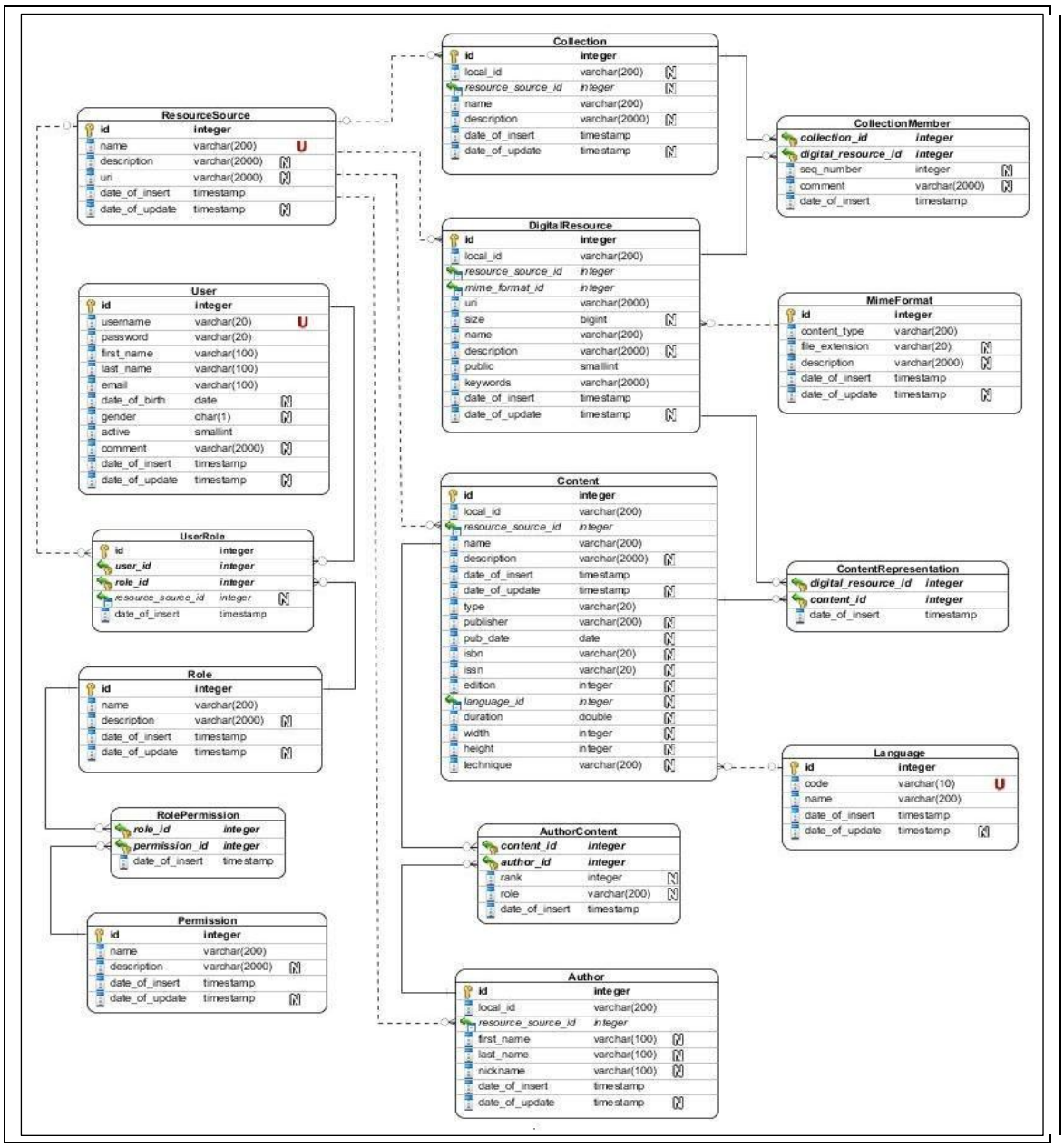

Figure 1: Database scheme

The register prototype supports only the most common types of content. Database can be extended to support new types without significant changes in database structure, using Entity-Attribute-Value model (EAV), also known as Object-Attribute-Value model and Open Schema [3]. EAV data model is often used when the number of different entity attributes is unlimited. However, in practice only a small set of attributes is used. Database would have two additional tables: one that defines allowed content attributes and another one that stores content identifier, attribute identifier and the attribute value.

```
<sources>
	<source sourceId=”1”>
		<name>Библиотека Петар Кочић</name>
		<description>...</description>
		<uri>www.petarkocic.rs</uri>
	</source>
</sources>
<contents>
	<content sourceId=”1” localId=”c1”>
		<name>Хазарски речник</name>
		<description>...</description>
		<type>book</type>
		<publisher>Народна књига<publisher>
		<pubDate>2006-09-02</pubDate>
		<edition>2</edition>
		<isbn>978-8673464565</isbn>
		<language>SR</language>
	</content>
	<content sourceId=”1” localId=”c2”>
		<name>Кутија за писање</name>
		<type>book</type>
		<language>SR</language>
	</content>
	...
</contents>
<authors>
	<author sourceId=”1” localId=”a1”>
		<firstName>Милорад</firstName>
		<lastName>Павић</lastName>
	</author>
	...
</authors>
<authorContents>
	<authorContent sourceId=”1” contentId=”c1” authorId=”a1”>
		<role>писац<role>
	</authorContent>
	<authorContent sourceId=”1” contentId=”c2” authorId=”a1”>
		<role>писац<role>
	</authorContent>
	...
</authorContents>
<digitalResources>
	<digitalResource sourceId=”1” localId=“dr1”>
		<mimeType>application/pdf</mimeType>
		<uri>http://www.petarkocic.rs?id=1612</uri>
		<name>Хазарски речник</name>
		<public>1</public>
	</digitalResource>
	...
<digitalResources>
…
```

Register users (*User*), their roles (*Role*) and set of permissions (*Permission*) for each role are represented with relevant entities. Associations between user and role, as well as between role and permission are described as additional entities *UserRole* and *RolePermission.*

Database triggers are defined on all tables. Create and update operations on table rows fire triggers that store exact time of operation in the database, for each table row. Entities have extra attributes that store information on creation time (*date_of_insert*) and last modification (*date_of_update*).

The register retrieves information from sources by importing metadata using XML documents. The structure of XML documents, validated by XML scheme, corresponds to the structure of register database. A partial example of valid XML document is presented above.

4. Search techniques

We expect that register based on presented concepts may have a lot of (concurrent) users. That requires highly efficient searching methods. The register prototype provides two different types of searching. The first one is the so-called simple search that finds digital resources by entered keywords or phrases, using the full-text search technique [4]. Full-text queries perform linguistic search against text data in full-text indexes. This is simple only from the usage point. One of the primary problems is that DB2 Net Search Extender (nor the other similar tools) does not support the Serbian language. On the other hand, it allows using of thesaurus. The thesaurus is a document, structured like a network of nodes linked together by associative or synonym relations. To make the linguistic search possible, we defined and used thesaurus where, among others, all words with the same root or just in different alphabets (Cyrillic or Latin) are specified as synonyms. We started with a simple list of Serbian words that we got from MySpell package [5]. Then, we defined an algorithm for grouping synonyms and used it to make the thesaurus. Our solution doesn't provide perfect matching results for all words, but the matching is good enough for the prototype phase. An example of the full-text search is presented in Figure 2 (matching results with metadata are presented below the search box).

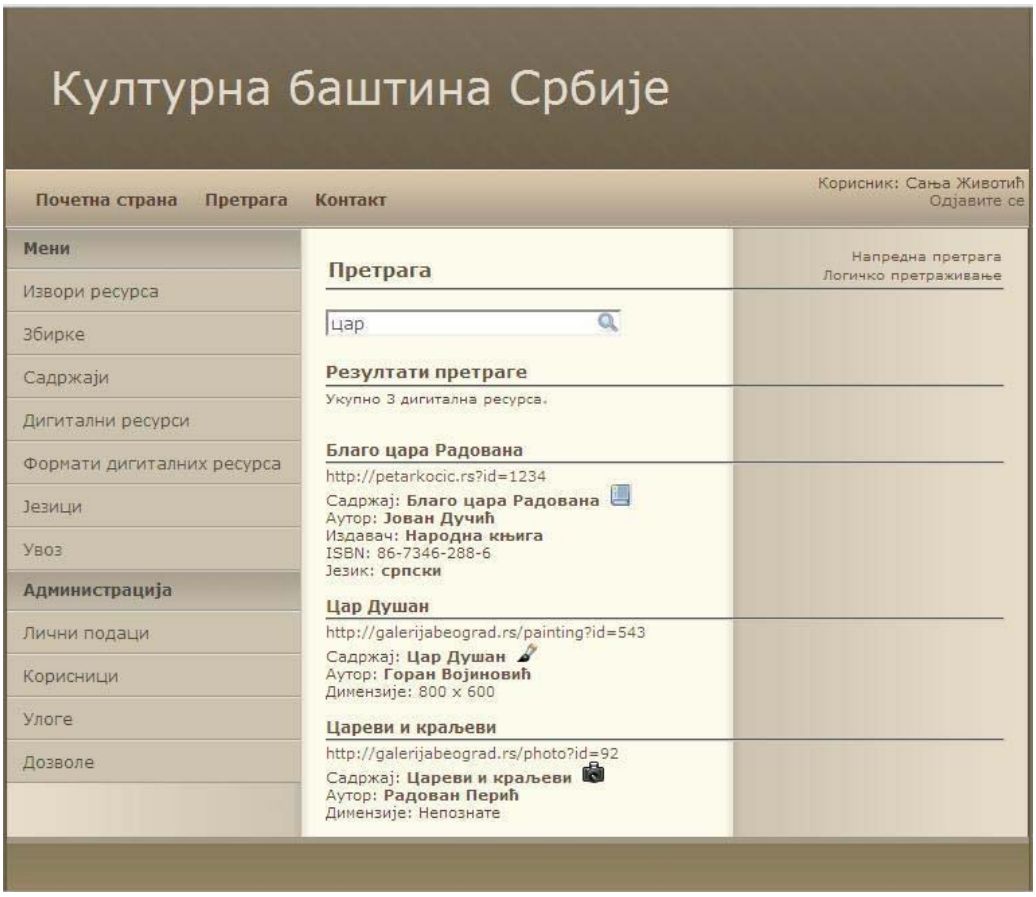

Figure 2: Simple search

The other type of searching is the advanced search. Advanced search allows the specification of various advanced criteria to narrow the search results. Depending on the types of resource and the represented content, the search criteria can vary. Regardless of content type, users can always enter basic search criteria like information source, resource name and format, content type, author name, etc. In example presented in Figure 3, because books are specified as the content type, the appropriate additional type-specific criteria are presented, like publisher, date of publishing, ISBN and language.

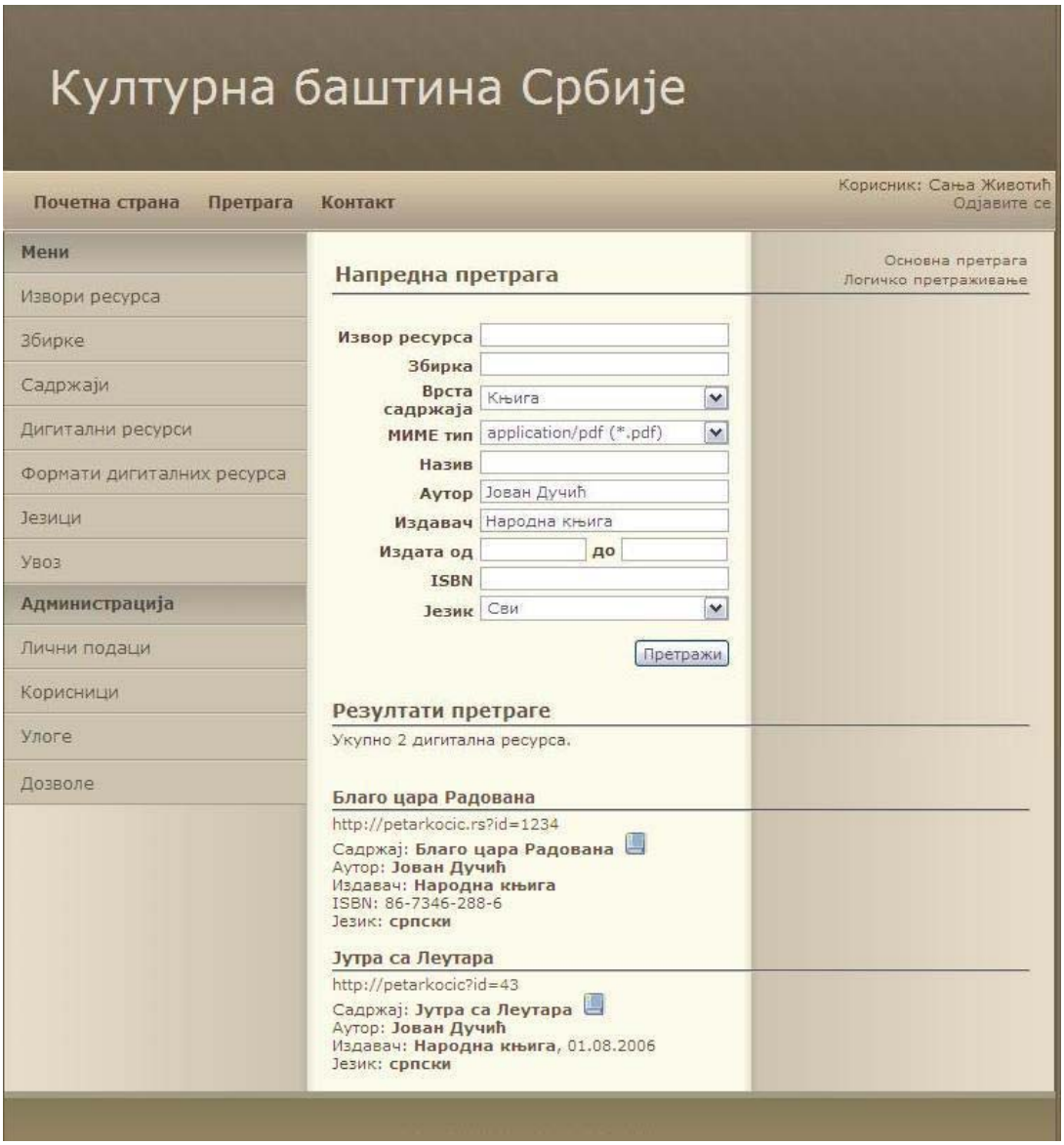

Figure 3: Advanced search

5. Implementation tools

Register prototype is implemented in object oriented programming language Java [6]. Java Server Pages (JSP) technology [7] and the open source servlet container Apache Tomcat [8] are used for development of dynamic web content. Data storage and manipulation are

handled by relational database management system IBM DB2 [9]. DB2 Net Search Extender [10] is utilized for full-text search. Object-relational mapping (ORM) library Hibernate, [11] solves object-relational impedance mismatch problems.

6. Conclusion

The central register of digitized cultural heritage provides a simple way to locate digital resources. The register is designed to be efficient and flexible. In agreement with source representatives, it could support additional metadata. For initial data loading and data maintenance it requires cooperation with institutions and individuals to provide information on digitized resources.

The application of the register in practice would provide not only the efficient search of digital resources, but support the process of digitization all over the country and significantly contribute to the preservation of cultural identity and heritage.

References

[1] Ministry of Culture of Republic of Serbia, Web Site http://www.kultura.gov.rs/?jez=sc&p=1
[2] Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman, *System Analysis and Design Methods*, 5th ed., McGraw-Hill, 2001
[3] Carol Friedman, George Hripcsak, Stephen B. Johnson, James J. Cimino, Paul D. Clayton, *A Generalized Relational Schema for an Integrated Clinical Patient Database, in Proceedings of the Annual Symposium on Computer Application in Medical Care*, 1990, 335–339.
[4] Chavdar Botev, Sihem Amer-Yahia, Jayavel Shanmugasundaram, *Expressiveness and Performance of Full-Text Search Languages, in Yannis Ioannidis (ed), Advances in database technology: EDBT 2006 : 10th International Conference on Extending Database Technology, Munich, Germany, March 26–31,* Springer, 2006, 349–367
[5] Goran Rakić, *MySpell package for Serbian language in GNU Aspell*, 2005, http://srpski.org/aspell/
[6] Ken Arnold, James Gosling, *The Java Programming Language*, Java Series, Sun Microsystems, 1996.
[7] Hans Bergsten, *JavaServer pages*, O'Reilly Media, 2003
[8] Jason Brittain, Ian F. Darwin, *Tomcat: The Definitive Guide*, O'Reilly Media, 2007
[9] *IBM DB2 Version 9.7: SQL Reference*, IBM Corporation, 2009.
[10] *IBM DB2 Version 9.7: DB2 Net Search Extender documentation*, IBM Corporation, 2009
[11] Christian Bauer, Gavin King, *Java Persistence with Hibernate*, Manning, 2006,

e-mail: {sanjaz,smalkov,nenad,zarkom}@matf.bg.ac.rs

Corresponding author: Sanja Životić sanjaz@matf.bg.ac.rs