A web based DBMS supporting development of

Armenian Science Citation Index

by

Davit Abovyan

Diploma in Macroeconomics, Yerevan State University of Economy, 2004

A thesis submitted in partial satisfaction of

the requirements for the degree of

Master of Science

in

Computer & Information Science

in the

COLLEGE OF SCIENCE AND ENGINEERING

of the

AMERICAN UNIVERSITY OF ARMENIA

Supervisor: _____

Signature: _____ Date:_____

Committee Member: _____

Signature: _____ Date:_____

Committee Member: _____

Signature: _____ Date:_____

Committee Member: _____

Signature: _____ Date:_____

Abstract

A web based DBMS supporting development of Armenian Science Citation Index

Author: Davit Abovyan

In the scope of this project a web based data management system should be developed for Armenian Science Citation Index (ASCI) which can later grow into a web portal including complete and comprehensive bibliographical data on all scientific publications in Armenia, their citations in other publications worldwide, a centralized multi-functional information system that will incorporate 'bibliometric' data for Armenian researchers, such as their affiliations, journal and other publication citations, etc. ASCI related data is currently collected by Center for Scientific Information Analysis and Monitoring at Institute for Informatics and Automation Problems (CSIA). CSIA uses MS Access files for that purpose and manually input data. Part of such data is send to Russian eLibrary.ru system to make it available to the world.

This web site will support in relevant data collection by providing centralised point of access and data consistency and integrity checks which will partially solve above mentioned data related issues. These data, being publicly available and searchable, could be used as a tool to analyze and evaluate Armenian researchers' impact based on their publications record and parameters, such as the Citations Impact Factor. It will increase the scope of relevant date available to public by web services and will decrease the time spent by used to get their needed data. It can also be a start point to develop an easy search and collaboration tool for Armenian scientists locally as well as in diaspora - a tool to find other Armenian researchers in the same field of study.

Keywords: bibliographic, DBMS, Java, RestAPI

Licenses for Software and Content

Software Copyright License (to be distributed with software developed for masters project)

Copyright (c) 2018, Davit Abovyan 

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(This license is known as "The MIT License" and can be found at http://opensource.org/licenses/mit-license.php)

Content Copyright License (to be included with Technical Report)

LICENSE

Terms and Conditions for Copying, Distributing, and Modifying

Items other than copying, distributing, and modifying the Content with which this license was distributed (such as using, etc.) are outside the scope of this license.

1. You may copy and distribute exact replicas of the OpenContent (OC) as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the OC a copy of this License along with the OC. You may at your option charge a fee for the media and/or handling involved in creating a unique copy of the OC for use offline, you may at your option offer instructional support for the OC in exchange for a fee, or you may at your option offer warranty in exchange for a fee. You may not charge a fee for the OC itself. You may not charge a fee for the sole service of providing access

Table of content

About the purpose of the project

In the scope of this project a web based data management system should be developed for Armenian Science Citation Index (ASCI) which can later grow into a web portal including complete and comprehensive bibliographical data on all scientific publications in Armenia, their citations in other publications worldwide, a centralized multi-functional information system that will incorporate 'bibliometric' data for Armenian researchers, such as their affiliations, journal and other publication citations, etc. ASCI related data is currently collected by Center for Scientific Information Analysis and Monitoring at Institute for Informatics and Automation Problems (CSIA). CSIA uses MS Access files for that purpose and manually input data. Part of such data is send to Russian eLibrary.ru system to make it available to the world.

This web site will support in relevant data collection by providing centralised point of access and data consistency and integrity checks which will partially solve above mentioned data related issues. These data, being publicly available and searchable, could be used as a tool to analyze and evaluate Armenian researchers' impact based on their publications record and parameters, such as the Citations Impact Factor. It will increase the scope of relevant date available to public by web services and will decrease the time spent by used to get their needed data. It can also be a start point to develop an easy search and collaboration tool for Armenian scientists locally as well as in diaspora - a tool to find other Armenian researchers in the same field of study.

CSIA provided sample input data which was used to develop requirements. Later data inputs and administration will also be controlled by CSIA.

Development code

1. Back-end part

Github: https://github.com/davit-abovyan/bibliographic/tree/master-REST

2. Front-end part

Github: https://github.com/davit-abovyan/bibliographic_front

Tech approach

3. High level tech approach

Main approach:

Considering that potential requirements on later possible growth and integrations with other systems and the current tendencies in web development web application should be RESTful, resulting full separation of back-end from front-end.

Benefits are as follows:

- When extending application usage from browser to desktop and/or mobile no change will be made in back-end.
- External integrations with the system will be available,
- Full separation of front-end from back-end brings to more clear code which is easier to maintain.
- Further development will require development team instead of single developer in which case no full stack developer will be required, work will be easily divided between developers with more concrete specification eliminating lots of problems raised on teamwork.
- Later code refactoring due to change in technologies will affect only half of the code base in particular point of time.

Methodology used:

Agile is used as a software development methodology as it is expected that requirements will be changed during the development and the list of features to be implemented during this project can be extended or reduced depending on actual progress. Two sprints is expected each with two month duration, First sprint for research on technologies, tools, libraries to be used and their testing, second sprint for actual implementation.

Version Control:

GitHub is used as a version control tool, as it is already an industry standard, it is free (unless private repo is needed).

Encoding:

UTF-8 encoding is used to support Armenian and Russian unicode. It is considered in database, Spring servlet as well as in web view.

2. Back-end tech approach

Programming language:

Java (version 8) is used as a programming language for back-end as it is considered as an efficient (time and memory) and effective (writing code) language for web applications, it is well known (easy replacement of developer, reach and stable libraries).

Framework used:

Spring framework is used, as being free and the most popular Java framework for web applications. In Spring for setups XML is used and for the rest annotations. Besides Spring main tools, Spring security is used for authorization and authentication, JDBCTemplate is used for work with database.

Build and dependency control tool:

Maven is used as a build and dependency control tool, it is effective, well known and popular tool.

Integrated development environment (IDE):

Intellij is used as and IDE as it is well developed and have very powerful tool. It works fine with Spring, mySQL, Git, Maven. It is not free but education purposes it provide free license.

Coding style used:

Naming - for variable and method name camelcase is used with lower case first character, name should include the description of the field with max limit of five words, constants are all in upper case with words divided by dash.
Commenting - JavaDocs style is used for commenting methods signature, single line comments are used otherwise in any place where additional information will be useful.

Code cleanup - "*Later means never*" is the slogan used, code is being cleaned during its initial writing process (at least tried to do so).

DESIGN APPROACHES USED:

Application layering: Considering possible changes in future and grow in code base several levels of indirection is used. For each specific object each of the below layers are used:

- DAO (data access object) layer - all functionality related to work with database (CRUD operations) is done within this layer. Each DAO layer is implemented for a particular type of database (MySQL in our case). All the other layers works with database only through this layer.
- Service layer - all business logic is implemented here, it uses DAO layer to work with database, as well as service layer for other objects.
- Controller layer - this layer is the endpoint for all http request, it uses service layer to serve the requests.

Indirection in inheritance: Interfaces and several level of inheritance are used for better code reuse and fewer changes experience. In Service layer DAO layer objects are used indirectly through interfaces as in case of change or addition of other database type nothing should be changed except DAO layer.

REST API: Main requirements for RESTful services is be kept, CRUD is implemented by POST, GET, UPDATE and DELETE http methods, API support JSON object, API is versioned with no change to already provided contract, endpoints are in following the format /api/vX.X/object_name/

Database access: For work with database Spring JDBC is used. Although it requires more code than JPA or Spring Data, but it gives more flexibility and more control. For effective use of database connects, pool is used.

Exception handling: Custom exceptions are developed in order to catch all main exceptions and rethrown in DAO and Service layers, so it can be properly handled in API responses. All try/catch blocks are eliminated from DAO and Service layers being catched in Controller.

Third party libraries: For effective manipulation of data in JSON format Gson library is used (Google implementation of JSON),

Logging: In order to track and analyze unexpected behavior of already deployed web application logs are being recorded using log4j library.

Testing: To ensure that continuous integration of new features into application and bug fixed don't introduce new bugs automation tests are developed. For unit and integration testing Junit library is used

Application deployment: Tomcat is used as an application server, it is free and feets Spring well. The back-end part of application is deployed in a hosted virtual machine together with MySQL server.

3. Database tech approach

Database used:

MySQL database with InnoDB engine is used as it is free and fully satisfy all technical requirements for such web application, it is very popular and well developed.

Naming:

For field/table naming dash is used to separate words, index name ends with _idx, and foreign keys with _fk.

Other aspects:

Data integrity are assured in database level using NOT NULL and UNIQUE constraints and default values. Predicted limit of characters for VARCHAR fields is used, otherwise maximum.

4. Front-end tech approach

JavaScript framework used:

Angular 4 with TypeScript is used as a front-end framework, it is well developed and very popular and lots of libraries are developed for free use. For deployments it is compiled to relatively small minified files. Angular 4 has some limitations but is much easier to develop than ReactJS which is the main competitor with Angular.

Front-end library used:

Bootstrap 4 is used for quick formatting of web view, although it supports responsiveness, but the way Angular 4 is used in web application, such responsiveness mainly loss its credits. HTML5 and CSS3 are used in Bootstrap 4.

Images used:

Actually no image is used, instead custom fonts (Fontawesome) replacing icons are used. Main benefits in comparison with images are:

- It is light weight;
- It doesn't need additional http requests to server (browser could make very limited number of request simultaneously);
- In scaling it doesn't lose any accuracy;
- It can be placed in any place where text is allowed and the color change is as easy as for text.

Cascading style sheets:

LESS technology is used to generate css files. It has some kind of object oriented features such as inheritance, variables for code reuse, methods. It could be used for view customization.

5. Project initial timeline

The project was planned to be completed in 4 months period with following high level breakdown:

General task description	***January***	***February***	***March***	***April***
Literature review				
Analysis of existing data				
Discuss and document all possible use cases and gather requirements				
Create technical approach for front-end				
Create technical approach for back-end				
Create technical approach for database				
Create development iterations and task breakdown for each iteration				
Development of features based on iteration				
Review changed requirements and reschedule iterations				
Develop new features based, bug fixing				
Create automation test, final deployment				
Project finalization/documentation				

References

Materials needed to get fair understanding of the final outcome of the project

Wikipedia

1. https://en.wikipedia.org/wiki/Citation_index
2. https://en.wikipedia.org/wiki/Scopus
3. https://en.wikipedia.org/wiki/Web_of_Science

Websites

4. https://clarivate.com/products/web-of-science/
5. https://elibrary.ru/defaultx.asp
6. http://csiam.sci.am/

Information relevant from local viewpoint

Data/information provided by Center for Scientific Information Analysis and Monitoring, Institute for Informatics and Automation Problems

Technical publications/materials

1. **Core Java** - Tenth edition, Cay S. Horstmann, December 2015, *ISBN 9780134177304*
2. **Spring in action** - Fourth edition, Craig Walls, 2015, *ISBN 9781617291203*
3. **Database System Concepts** - Sixth edition, Abraham Silberschatz, 2011, *ISBN 9780073523323*
4. **Ang-book The complete book on Angular 5** - N. Murray, F.Coury, A.Lerner
5. **Apache Tomcat 7** - User Guide, Apache Software Foundation, 2011
6. **Clean Code** - A Handbook of Agile Software Craftsmanship, Robert C. Martin, 2008