WEB CRAWLER FOR INDEXING SCIENTIFIC PAPERS

Melanya Hambardzumyan

Resum— En aquest projecte s'aborda el problema que molts investigadors i professionals relacionats amb àrees de la ciència i la tecnologia tenen, la de buscar articles d'interès publicats en revistes online. Encara que hi ha eines de recerca disponibles per a realitzar aquesta tasca, aquestes sovint són molt genèriques o complicades de configurar.

En aquest treball es desenvolupa un web crawler que fa una recerca en la llibreria digital del IEEE Xplore i mostra articles que continguin paraules clau o autors prèviament especificades.

Aquesta eina agilitzarà una part essencial de la feina de cerca d'articles en pàgines webs, ja que l'usuari no haurà d'entrar manualment a les pàgines i revisar cada revista per veure si hi ha algun article d'interès. En compte d'això només haurà de clicar un botó i la cerca es farà de forma automàtica.

Paraules clau— Web crawler, spider, JSF, JEE, Aplicació web, IEEE Xplore Digital Library

Abstract— This project addresses a problem that many researchers and professionals related to areas of science and technology have, that of searching articles of interest published in online journals. Although there are searching tools available to do this task, they often are very generic or difficult to configure.

In this project a web crawler is developed. It does a customized search in the IEEE Xplore Digital Library and shows papers containing keywords or authors, previously specified.

This tool will speed up an essential part of the papers search in the web page, since the user does not need to enter manually to the journals and check each new volume and issue to see if there are papers of interest. Instead, the user only has to click a button and the search will be carried out automatically.

Index Terms— Web crawler, spider, JSF, JEE, Web application, IEEE Xplore Digital Library

1 Introduction

1.1 Motivation

RESEARCHERS around the world want to keep up with the most recent issues published in the journals they are interested in. The problem is that there is not a tool that does the automated search of the issues in an easy, quick and efficent way.

The problem is that there are many journals and websites that contain content that researchers are interested in. There are tools like RSS that simplify a little these problems but no tool allows to do the search in a personalized way.

The development of the project has focused on solving, at least providing an alternative, to his problem to allow researchers to search articles they have interest in.

1.2 State of Art

The Institute of Electrical and Electronics Engineers (IEEE) is a worldwide association of professionals of Electrical Engineering, Electronics, Computers, etc. [1] which aims to standardize methods in technical fields.

Founded in 1884, IEEE is the largest company of associated professionals, with more than 395,000 members in 160 Countries. This project is based on the IEEE Digital Library Xplore, where the search in journals will be done. In the IEEE Xplore Digital Library [2] more than 170 journals with scientific content published by IEEE and associate professionals can be found. It contains more than 2 million documents in HTML format, with 20,000 new ones added each month.

There are several mechanisms that offer parts of the features designed to develop in this project. For instance the IEEE Advanced Search offers the functionality to search articles based on introduced keywords, but the search is done in the entire database, not in specific journals and issues as it is aimed in this project.

- *E-mail de contacte: melanya.hambardzumyan@gmail.com*
- *Menció realitzada: Tecnologies de la Informació.*
- *Treball tutoritzat per: Francesc Aulí Llinas (Departament d' Enginyeria de la Informació i de les Comunicacions.)*
- *Curs 2015/16*

Most of this type of engines are limited and too "simple" because they search in the entire database or they are very complex as they have to be configured to perform custom searches.

1.3 Scope

A web crawler [3], [4], also known as a web spider, is a program which, given a number of "seed" pages, browses the WWW in search of these pages and downloads them. The crawler also extracts hyperlinks from the visited pages and does this process recursively in the extracted hyperlinks. Figure 1 shows a very simple architecture of a generic crawler. It works as explained above. There is a queue of URLs that are scheduled to visit. When accessing these URLs, the metadata is extracted and stored in the database and other URLs are extracted of these pages, which are added to the queue of URLs to visit.

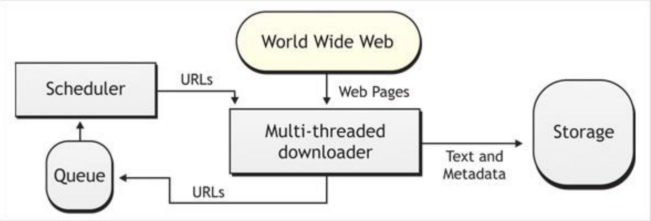


Fig. 1. Generic web crawler architecture [5].

The Web Crawler developed in the project is a tool for realizing personalized searches in the IEEE Xplore Digital Library. The tool is as easy to use as to first of all to sign up, and once signed up and logged in, select the journals the user is interested in. Additionally the user has to introduce the keywords and/or authors he is interested in, to carry out the search.

1.4 Paper Structure

The paper is structured in five sections. The motivation, as well as the state of art and scope of the project are described in Section 1. Section 2 is the thesis proposal. It contains the project specifications as well as objectives, their categorization and requirements. Methodology, project schedule and stages are described in Section 3, project plan. This section contains detailed explanation of the performed tasks as well as the project calendar. Section 4 contains meticulous description of the functionality of the application as well as explanation of the architecture and design, back-end and the web interface. This section also explains in great detail the programming language and its features, tools used to develop the application, the database and the user interface. Section 5 contains the results of the project. The results include a review of the objectives and whether they have been fulfilled or not. In addition, the features of the application are explained. The last section closes the paper with the project conclusions.

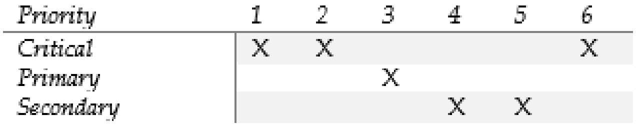
2 Thesis Proposal

The aim of this project is to develop a search engine, that employs a Web Crawler, which performs custom searches in the IEEE Xplore Digital Library based on parameters such as the user preferences, authors and journals in which the user has interest in.

This search engine is a web application where the user can log in and perform the search employing on the parameters mentioned above, and get all the results matching the specifications in a single page.

2.1 Objectives

1. Basic operation of the search engine. When the *Search* button is clicked, the information has to be searched and displayed according to the search options and keywords specified.
2. Search the information in the proper fields.
   When the user introduces the keywords in the specified area, the searching engine has to look for the keyword in the title of an article.
   When the user introduces the author he is interested in in the specified area, the searching engine has to look for the authors involved in an article despite of the article title.
3. Save and retrieve information.
   When a user signs up in the application, the username and password are stored in an xml document, which fulfills the database function.
   When a registered user loges in the application, first the username and password are retrieved from the database, and once logged in, a list is shown to the user with the information of the last journal volume and issue consulted. This information is also retrieved from the database.
   Once the search is done, and the user has finished, the new searching information is stored in the database.
4. Design and implementation of a simple user interface.
   One of the complains with other searching engines was that they were complicated to use or complex to configure, so one of the objective is to make a user friendly interface, easy to use, easy to understand and easy to handle.
5. Optimization of the source code.
   There are many ways to obtain and manipulate html code and one objective is to use new technologies oriented to optimize these tasks.
   This is not only a help in terms of programming it is also a good opportunity to learn new technologies.
6. Finish the project in the scheduled time.



Priority	1	2	3	4	5	6
Critical	X	X				X
Primary			X			
Secondary				X	X	

Fig. 2. Objectives priority table.

Figure 2 shows the categorization of the project objectives. The most critical objectives are the ones in which the main functionality relays, such as the searching function and the proper uptake of the keywords. The other very critical objective is the time factor. The project has to finish as scheduled.

The primary objective is that of the tasks related to the database. Without the database implementation, the Web Crawler simply will be for occasional use as the user will not be able to store his preferences. Plus, the non-compliance of the functionality, will mean that changes will have to be applied to the initial project proposal in order to make it simpler.

Finally, the secondary objectives are the optimization of the source code and the design of the simple user interface. This means that if these two objectives are not achieved, they will not prevent the project completion.

2.2 Requirements

From the objectives, it has been established a number of application requirements. These have been divided between functional and non-functional.

2.2.1 Functional

- The application must register new users.
- The application must allow access to already registered users.
- The application must access properly to the database to retrieve the specified user information.
- When registering, if the user does not enter the same password in the password and verification fields, the application must not allow the registration of the new user.
- Once logged in, the application must display correctly last volumes and issues of the consulted journals.
- The application has to show new issues and/or volumes published since the last query.
- The application must display a catalog of journals which can be added to the search.
- The application must make searches of article in new issues and/or new journals selected.
- The application must make the search for the keywords entered.
- The application must make the search for the authors introduced.
- The application must display correctly the results of the search.
- The application must update user information in the database with new data, either because a new journal is added to the search, or there has been new published issues and these have been selected to do the search in.

2.2.2 Non-functional

- Source code must be written in java.
- The user interface must be implemented in XHTML.
- Establish the minimum number of connections to the IEEE Xplore Digital Library in order to avoid timeouts.
- Eliminate article duplications after the search result.
- The application must provide error messages that are informative and oriented towards the end user.

3 Project Plan

3.1 Methodology

The project management is a guiding methodology for the processes and tasks to be performed during the project. Management methodologies are step-by-step patterns, methodologies, processes and logically related practices for successful planning and execution of projects from beginning to end [6].

The methodology followed in the project has been based and depending on the needs and type of project, which is a web application.

The methodology used in this project has been Agile Software Development [7]. The properties of agile is that this methodology is iterative and incremental, with "sprints" every few weeks.

As shown in figure 3, the project consists of stages of development. Once a milestone is reached, after testing and deciding whether the success or failure of the phase, the project will continue as the same or readjustments and changes will be introduced.

Instead of defining every little detail regarding the design or implementation at the beginning of the project, the main objectives and line of work were defined. From that point the development started, and as the sprints passed, the followed tasks started to take form.This does not mean that agile is a chaotic or less defined methodology than others, but compared to other methodologies such as Waterfall, it is more flexible and less static.

The chosen methodology is optimum for the project because instead of having planned all the tasks form the beginning to the end, the project has been splitted into various sprints, with several functionalities in each. This makes it easier to overlook the entire project, and allows for quick changes if any task does not fit the project.

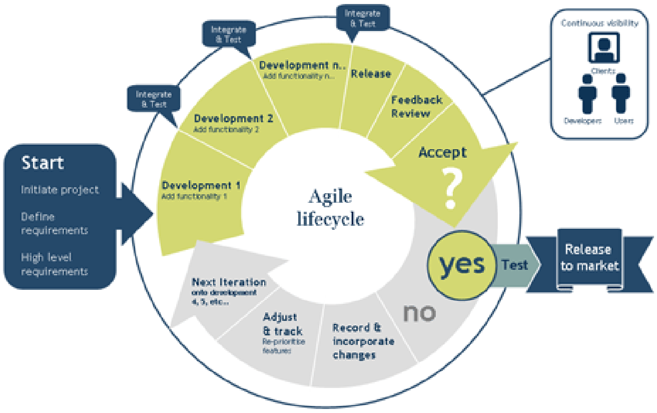
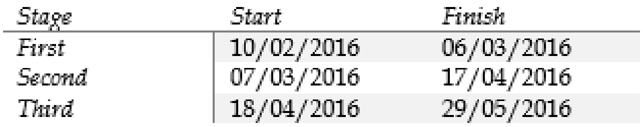


Fig. 3. Agile Methodology Lifecycle [7].

3.2 Project Stages

As stated in the previous section, the detailed planning of the tasks has been done iteratively, but major tasks were planned from the beginning, as well as the main stages. Figure 4 shows the beginning and end of each stage.



Stage	Start	Finish
First	10/02/2016	06/03/2016
Second	07/03/2016	17/04/2016
Third	18/04/2016	29/05/2016

Fig. 4. Project stages calendar.

3.2.1 First Stage

The first stage covers the beginning of the project and its planning.

The project started with the first meeting with the tutor where the idea of the project, functionality, goals, and more were discussed. The definition of objectives, planning of project and scheduling the development was done. In the first stage basically it has been done everything related to the planning in order to be able to start the programming part from that moment on.

3.2.2 Second Stage

In the second stage first of all the working environment with the server and the libraries has been set.

It has been programmed the first application functionalities. The first stage has been a learning stage of programming with the jsoup library (detailed in the next section).

In this step there have been performed simple tasks such as login or index user interface. Help page or the page that is accessed after login were also implemented.

Also it has been implemented the connection and html data manipulation of the source web page. In this aspect it has been implemented the functionality of journal names obtainment from the journal id.

It has also been done the integrations of the various functions and testing tasks have been started.

3.2.3 Third Stage

In the final stage is where most of the programming of the application has been done. It has been realized tasks related to the collection of data from both the journal and of the articles.

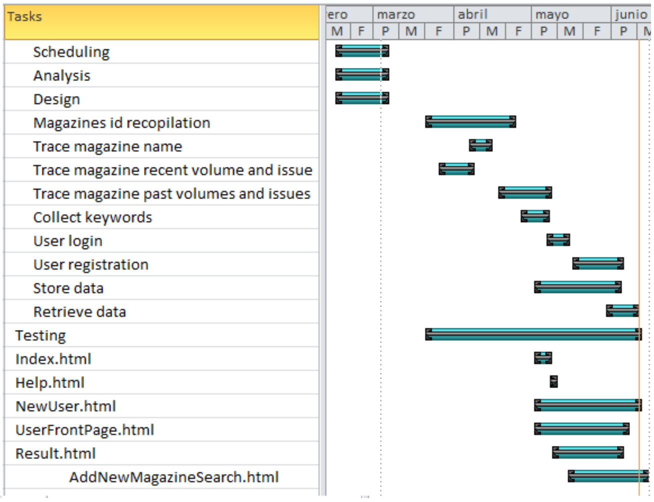
Tasks of storing information in the database as well as the recovery of these have been made.

It has been done the main functionality of the Web Crawler which is the search tool and everything related to obtaining information of the journals.

In the third stage the implementation of the application has been done and the project has been closed.

3.3 Schedule

Project planning is a difficult task because of the uncertainities in the project. In this project it was the lack of similar experiences and bias in estimates which difficulted a correct initial scheduling. So it is fair to say that the schedule of minor tasks was defined during the project development. The final tasks and plannings can be seen in figure 5, the Gantt chart.



Tasks
Scheduling
Analysis
Design
Magazines id recopilation
Trace magazine name
Trace magazine recent volume and issue
Trace magazine past volumes and issues
Collect keywords
User login
User registration
Store data
Retrieve data
Testing
Index.html
Help.html
NewUser.html
UserFrontPage.html
Result.html
AddNewMagazineSearch.html

Fig. 5. Gantt chart.

4 Development

The developing environment consists in Eclipse Mars.1, Glassfish Edition 4, Java Server Faces 2.2 and jsoup library.

JSF [8] is a Java Web application Framework, part of the Java EE standard, which uses the Model View Controller architecture pattern to separate the interface from the application processing core. This allows abstraction and reuse of the code and it offers scalability as the new functionality changes will only affect the information processing classes while the view will not be aware of it, and vice versa.

The JSF application runs on the Glassfish web container. Glassfish Server Edition 4 is used as it implements technologies defined in the Java Enterprise Edition platform [8].

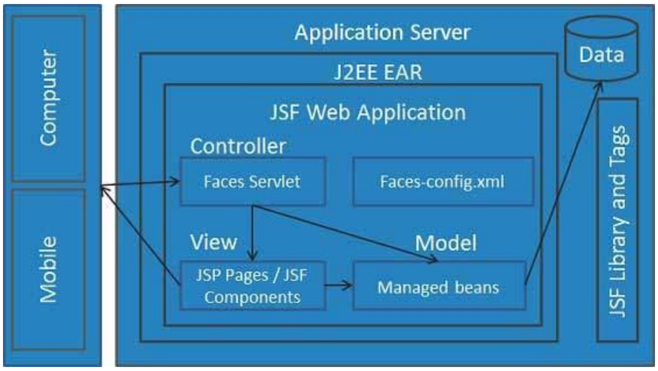


Fig. 6. JSF Architecture [9]

Figure 5 shows the architecture of JSF, which contains:

- Application server: Glassfish
- Managed beans [10]: These are regular java beans managed by JSF. The managed bean annotation is set when an object is referenced from the web page. These have associated a lifecycle which, if not specified, by default are @RequestScoped.
  In addition a managed bean instance can be injected in another managed bean by @ManagedProperty annotation.
- Faces Servlet: intercepts requests made by the client and generates a response.
- JSP Page: JSF contains tag library which contain HTML tags which must follow the XML standards (close open tags, simple tags are line breakers must also be closed, etc.).
- Faces-config.xml: Contains the configuration of JSF application.

4.1 Architecture and Design

To understand better the architecture and the operation of the program from figure 7, it will be explained through an example.

The IEEE Xplore Digital Library contains many Journals and publications. The journals have an identifier associated called punumber. The journal that is selected to illustrate the example is Transactions on Affective Computing. This journal has a punumber, as well as all the other journals, and volumes and issues, which are published periodically. The combination of volume an issue also has an identifier associated, isnumber.

Let's assume that our user is registered on the platform and has made the last query of the volume 6, issue 3 of the above specified journal. The application retrieves this information from the database, stored in form of identifiers (punumber-isnumber), and shows it to the user. Besides this, the application accesses the journal and checks if new volumes and/or issues have been published after the last query. If it is so, the application shows the user the new issues in form of checkboxes and the user will decide if he is interested in these new issues. In our example, it has been published volume 6, issue 4 and volume 7, issue 1, which both of them the user selects.

In addition to the new issues, the user also has the option to add new journals to the search query. In this second case it will automatically been taken the last issue of the last published volume. In our example, let’s say that the user is interested in Artificial Life and he selects the journal.

Arrived at this point, the user only has to entry the keywords in the specified fields and click Search. The application searches the keywords in the selected issues

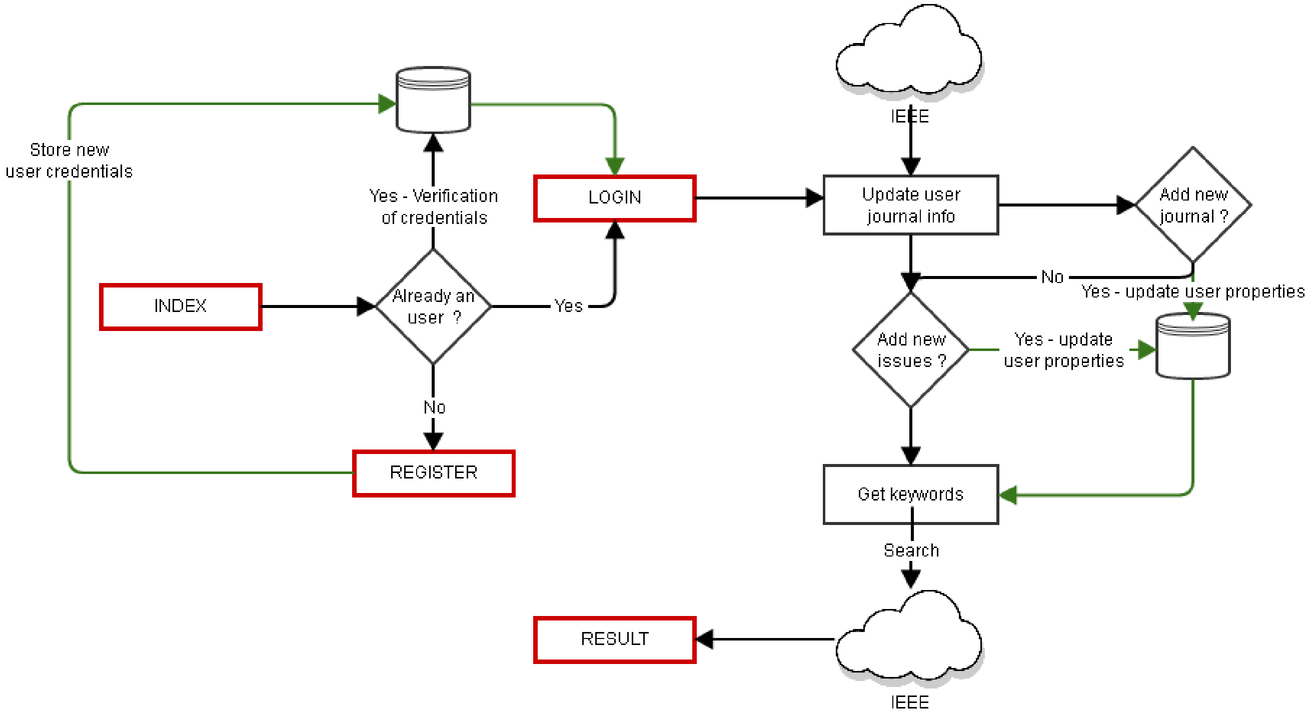


Fig. 7. Architecture of the Web Crawler. The red rectangles represent the views, the rhombus represent a questions with a Yes/No answer, the black rectangles represent an action, the two databases represent the main database in the application and the clouds with the IEEE description represent the IEEE Xplore Digital Library web page.

and journals and these results are returned to the user.

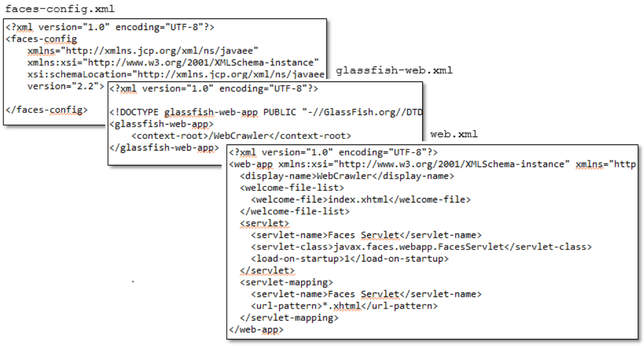
Once the search is over, the changes are saved in the database. If it has been chosen new issues of journals that were already followed, the value of these are updat-ed with the new isnumber. If a journal has been added to the search, a new node is created in the database with the credentials of the new journal (punumber-isnumber).

To end the example, our user had selected two new issues of the journal he had in the database, so the information of Transactions on Affective Computing is updated with the isnumber of volume 7, issue1 (the most recent one). Given that our user has selected a new journal, the credential of this journal and the newest issue are appended to the database.

The next time that our user accesses the platform, he will see two journals in his list.

4.2 Back-end

It has been created a Dynamic Web Project in Eclipse for the application development. The principal folders of the project are *Java Resources* where the project packages and classes are. The *Web Content* folder contains the view documents, application configuration files and libraries. The configuration files are the three shown in figure 8.



faces-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<faces-config
    xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    version="2.2">

</faces-config>
```

glassfish-web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE glassfish-web-app PUBLIC "-//GlassFish.org//DTD
<glassfish-web-app>
    <context-root>/WebCrawler</context-root>
</glassfish-web-app>
```

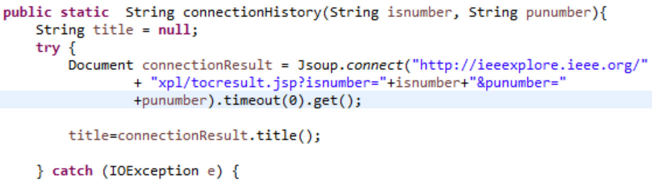
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http
  <display-name>WebCrawler</display-name>
  <welcome-file-list>
    <welcome-file>index.xhtml</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>*.xhtml</url-pattern>
  </servlet-mapping>
</web-app>
```

Fig. 8. Web Content configuration files

As already specified above, the implementation of the Crawler has been done in java. Since the extraction of most data is made from a web page, it has been added a library to the project called jsoup.

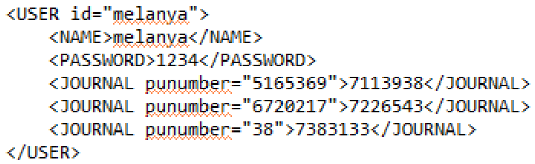
jsoup [11] is a library that allows, among other things, parsing, extraction or HTML data manipulation in java. Features as connection to a URL or extraction of HTML segments are used. Figure 9 shows how easy a connection is established and in this case, how the title of the HTML page is extracted.



```
public static  String connectionHistory(String isnumber, String punumber){
    String title = null;
    try {
        Document connectionResult = Jsoup.connect("http://ieeexplore.ieee.org/"
                + "xpl/tocresult.jsp?isnumber="+isnumber+"&punumber="
                +punumber).timeout(0).get();

        title=connectionResult.title();

    } catch (IOException e) {
```

Fig. 9. Example of extraction of the title of HTML page with jsoup.

In addition to data extraction, some attributes are stored in a database. Since the storing of user data or search information has not been a priority, it has been chosen a simple method to store it, through an XML document.

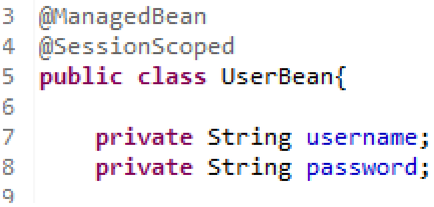


```
<USER id="melanya">
    <NAME>melanya</NAME>
    <PASSWORD>1234</PASSWORD>
    <JOURNAL punumber="5165369">7113938</JOURNAL>
    <JOURNAL punumber="6720217">7226543</JOURNAL>
    <JOURNAL punumber="38">7383133</JOURNAL>
</USER>
```

Fig. 10. Section of the database, xml document.

As it can be seen in the figure 10, the username and password are stored as a plain text. The last issues of each journal consulted are also stored in the file in the form of identifiers.

As previously mentioned, JSF annotations have been used to register managed beans. The registration in JSF looks like in figure 11:



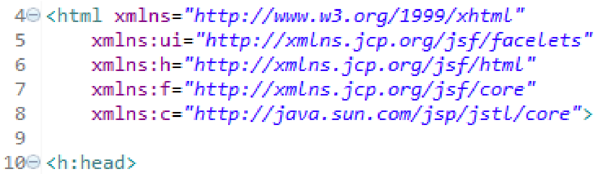
```
@ManagedBean
@SessionScoped
public class UserBean{

    private String username;
    private String password;

```

Fig. 11. Representation of JSF annotations

The annotation @SessionScoped represents that the bean will be alive during all the navigation session.

4.3 Web interface

The web interface has been implemented with the JSF Facelets. JFS offers tag libraries which have been included in the XHTML document (figure 9):



```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
    xmlns:h="http://xmlns.jcp.org/jsf/html"
    xmlns:f="http://xmlns.jcp.org/jsf/core"
    xmlns:c="http://java.sun.com/jsp/jstl/core">

<h:head>
```

Fig. 11. Tag library configuration

As for the interface, they have been used basic web components as buttons, output labels, multi-line text input areas, lists and many other elements.

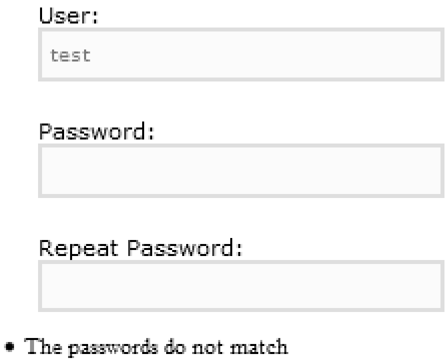
The project also contains a cascading style sheet for the application web interface style.

5 RESULTS

The results of the project have been satisfactory. Looking back to the objectives it can be declare that both critic and primary objectives have been fulfilled. This can be stated categorically as they are related to functionalities. As for the secondary objectives, a simple and easy interface with which to interact has been implemented. As for code optimization, a specific library has been incorporated to facilitate the development and simplify the source code.

Also if it is looked back to the requirements, they all have been met. As for these, some examples.

One requirement was to display error messages to inform the user that something was not OK. In figure 12 it can be seen the warning message when during registering the introduced passwords don't match. Similar messages are displayed when the register is successfully completed, or when during authentication the username or password is wrongly introduced.



User:
test

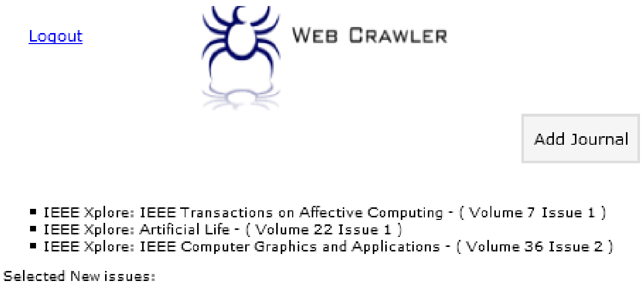
Password:

Repeat Password:

- The passwords do not match

Fig. 12. Message displayed when the passwords do not match.

Another example is, when logged in, the application shows the journals from the user database. In figure 13 can be seen the journals for the user *melanya*.

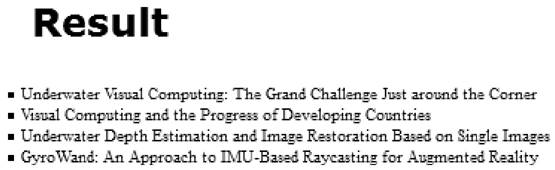


- IEEE Xplore: IEEE Transactions on Affective Computing - (Volume 7 Issue 1)
- IEEE Xplore: Artificial Life - (Volume 22 Issue 1)
- IEEE Xplore: IEEE Computer Graphics and Applications - (Volume 36 Issue 2)

Selected New issues:

Fig. 13. Journals from database for user *melanya*.

Finally, a snapshot of the result page which shows articles containing the keyword *and*:



Result

- Underwater Visual Computing: The Grand Challenge Just around the Corner
- Visual Computing and the Progress of Developing Countries
- Underwater Depth Estimation and Image Restoration Based on Single Images
- GyroWand: An Approach to IMU-Based Raycasting for Augmented Reality

Fig. 14. Result page

In the annex all the pages that are accessed when operating with the application can be seen.

6 CONCLUSIONS

It has been developed a tool for professionals or simply people interested in keeping up with the latest articles on their subjects of interest. The design and implementation has been from scratch and it has been managed to meet the proposed objectives on functionality, data treatment and interface. Changes have been done during some procedures because of unforeseen barriers but they have not been an impediment to complete the project.

As a result, on a personal level, my knowledge on JSF and data extraction of web pages has been expanded, in the process of which, I have experienced a few surprises in terms of design, structure or implementation of the IEEE Xplore Digital Library.

The Web Crawler developed in the project is a first version to which lots of upgrades can be applied. It is true that the objectives raised at the beginning have been met, but it is also true that these objectives were relatively simple because of the consideration that it is a project developed by one person, it has to begin from zero, and there is a time limitation.

On the one hand the storage part could be improved significantly, because the way it is implemented now provides capacity constraints and also is not optimal. The implementation and connection to a database will mean an opening gate to many other upgrades. On the other hand, the implementation is such to search in only one website. Of course other web pages that offer similar content could be added to the application, in fact, it would be an easy task because it would take the addition of a few new modules, and as the project is modular, few modifications would be required.

As for the interface, only the surface has been scratched. The goal was to make a simple interface, but it also can be improved, if the changes above are applied.

From a personal standpoint, it has been a great opportunity to experience what it takes to make a project from the very beginning. It has been a learning process and expansion of personal boundaries.

I have learned to be more autonomous and do whatever it takes to accomplish the goals set at the beginning.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my tutor, Francesc Aulí Llinas, for counselling me. For his patience and his support at every stage of the project. I am very grateful that I have been given the freedom to decide how to develop the project and decide all the changes to apply, yet his support has been unconditional. Support that has had to overcome good and bad stages as this road has been full of obstacles.

I also would like to thank my closest friends Nadia and Javier who comforted and encouraged me when I had problems, and brought a smile to my face.

REFERENCES

[1] History of IEEE
https://www.ieee.org/about/ieee_history.html
Last visited on 20/06/2016

[2] About IEEE Xplore® Digital Library
http://ieeexplore.ieee.org/xpl/aboutUs.jsp
Last visited on 20/06/2016

[3] Marc Najork, "Web Crawler Architecture", Microsoft Research, https://www.microsoft.com/en-us/research/wp-content/uploads/2009/09/EDS-WebCrawlerArchitecture.pdf. 2009

[4] Carlos Castillo, Mauricio Marin, Andrea Rodriguez, Ricardo Baeza-Yates, Center for Web Research, "Scheduling Algorithms for Web Crawling", http://chato.cl/papers/castillo04_scheduling_algorithms_web_crawling.pdf. 2004

[5] Crawlers & Data Quality
http://arcgate.com/blog/crawlers-data-quality/
Last visited on 25/06/2016

[6] Project Management Methodology: Definition, Types, Examples
http://www.mymanagementguide.com/basics/project-methodology-definition/
Last visited on 06/06/2016

[7] Metodología Agile como patrón pedagógico de ciertos aprendizajes
http://www.acanelma.es/2014/02/metodologia-agile-como-patron.html
Last visited on 06/06/2016

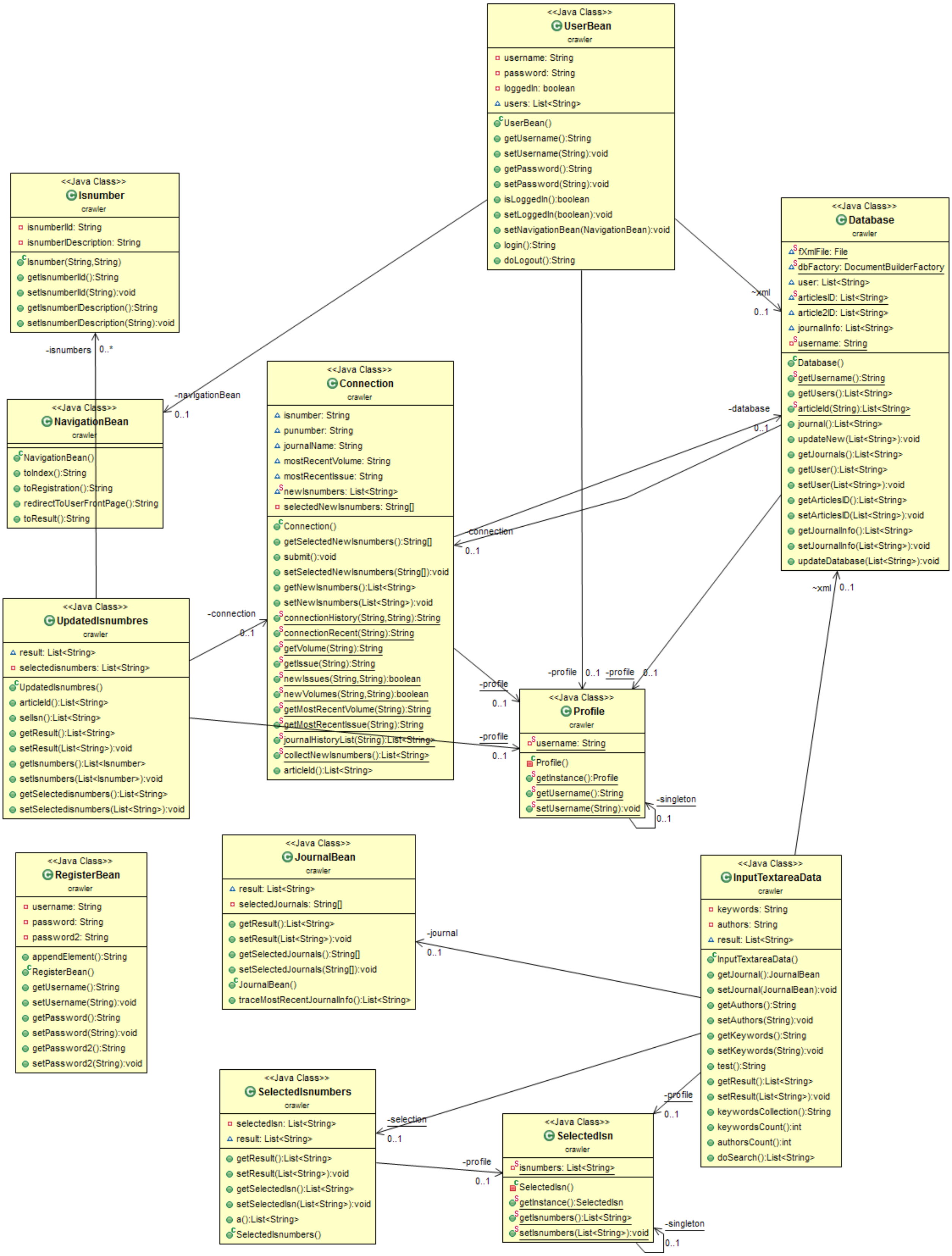
[8] Hans Bergsten, "JavaServer Faces", Chapter 1, 2004

[9] JSF – Architecture
http://www.tutorialspoint.com/jsf/jsf_architecture.htm
Last visited on 22/06/2016

[10] JSF 2 Tutorial Series
http://www.coreservlets.com/JSF-Tutorial/jsf2
Last visited on 22/06/2016

[11] jsoup: Java HTML Parser
https://jsoup.org/
Last accesses on 05/06/2016

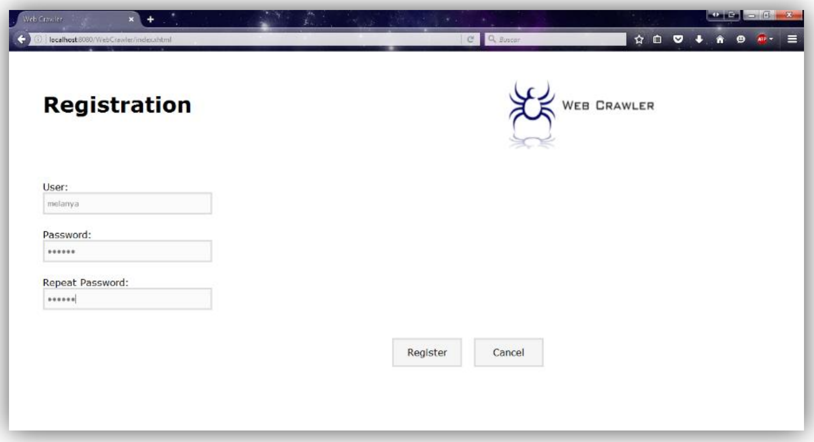
ANNEX



A1. UML DIAGRAM

A 2. WEB CRAWLER UI

Registration page:
Index:



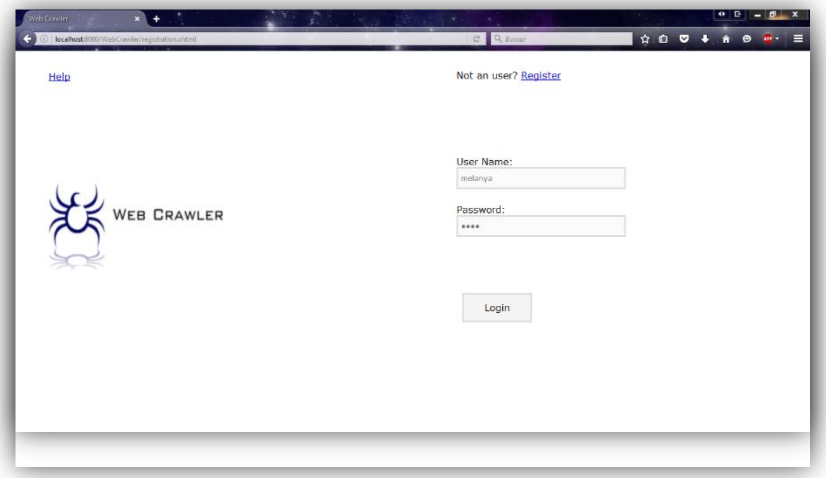
Registration

User:
melanya

Password:
••••••

Repeat Password:
••••••

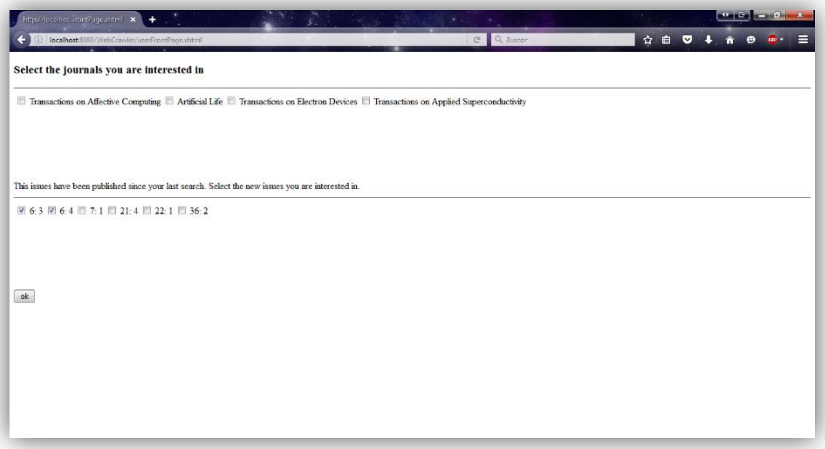
User profile page:



User Name:
melanya

Password:
••••

Add journal:
User profile page after selecting new issues and introduction of keywords:


https://localhos...rontPage.xhtml
localhost:8080/WebCrawler/userFrontPage.xhtml
Buscar

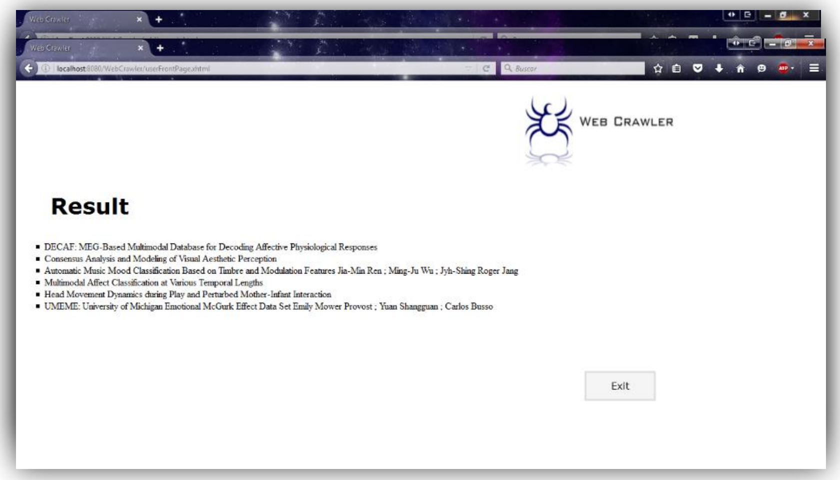
Select the journals you are interested in

□ Transactions on Affective Computing □ Artificial Life □ Transactions on Electron Devices □ Transactions on Applied Superconductivity

This issues have been published since your last search. Select the new issues you are interested in.

☑ 6: 3 ☑ 6: 4 □ 7: 1 □ 21: 4 □ 22: 1 □ 36: 2

Result:



Result

- DECAF: MEG-Based Multimodal Database for Decoding Affective Physiological Responses
- Consensus Analysis and Modeling of Visual Aesthetic Perception
- Automatic Music Mood Classification Based on Timbre and Modulation Features Jia-Min Ren ; Ming-Ju Wu ; Jyh-Shing Roger Jang
- Multimodal Affect Classification at Various Temporal Lengths
- Head Movement Dynamics during Play and Perturbed Mother-Infant Interaction
- UMEME: University of Michigan Emotional McGurk Effect Data Set Emily Mower Provost ; Yuan Shangguan ; Carlos Busso