

Using Distributed Applications in Personal Recruitment Management

Marian Pompiliu CRISTESCU¹, Liviu CIOVICA¹, Laurentiu CIOVICA¹, Florin MARTIN²

¹Academy of Economic Studies, Bucharest

²Faculty of Economic Science, „Lucian Blaga” University of Sibiu

mp_cristescu@yahoo.com, ciovica.liviu@yahoo.com, ciovica_laurentiu@yahoo.com,
florin.martin@ulbsibiu.ro

In any company, no matter the dimensions, a very important aspect is the selection of human resources and establishing a constant correspondence between the particularities of the job opened and candidate's profile. Especially in big companies, engaged in big projects, and which must be performed in a fast rhythm, the recruitment process is in a continuum progress, due to the demand rhythm, the work force mobility and to assuring a diversify knowledge data base. Experience has shown that a selection made in advance in the candidates CV's who apply for a scientific position, before programming the technical interview with them, is a real time saving for both the employers and the candidates

Keywords: Distributed Application, Personal Recruitment, Management, Human Resources

1 Introduction

In the process of filtering the candidates for an interview, HR department representatives are involved and even the RD department who is interested in hiring new candidates for specific jobs.

First category, HR represents the “interface with the exterior” because receives all the inputs from possible candidates, and has the purpose to establish a preliminary sorting, before proposing the candidates for interview, and the second category, the technical specialists, are those who impose the technical criteria, after that the candidates CV's are filtrated.

In the case of the multinational company Juniper Networks, the statistic shows that only 1 of the 10 candidates for software development job has enough potential that technical interview stage to justify

Thus, to provide context for the interaction between various groups involved in the process of filtering the candidates, it is proposed to develop a RIA, called HR Warehouse, which is available on the company intranet, so all teams between stakeholders and to automate the process of selection of candidates for job.

2 HR Warehouse application functionality

As a distributed application, which must access all the professional categories involved in the recruitment company, HR Warehouse will reside on a server where application will be made available on the company intranet. The application requires a browser to run. Since Vaadin foundation support, in which the application is built, is GWT, it will run guaranteed in every browser compatible with GWT like: Internet Explorer, Firefox, Opera, and Safari. Since the company is on the intranet HR Warehouse is accessible via a URL like:

[http ://< address> :< port>/HRWarehouse/](http://< address> :< port>/HRWarehouse/),

where the parameter “address” that designates the address location is the application server where the application is located and the parameter “port” refers to the application server port running HR Warehouse. In the case the server uses Tomcat the port is usually 8080.

Application framework is similar in terms of appearance with a desktop application, windows are de common organization forms, panels and views, not the pages as is natural in traditional web application

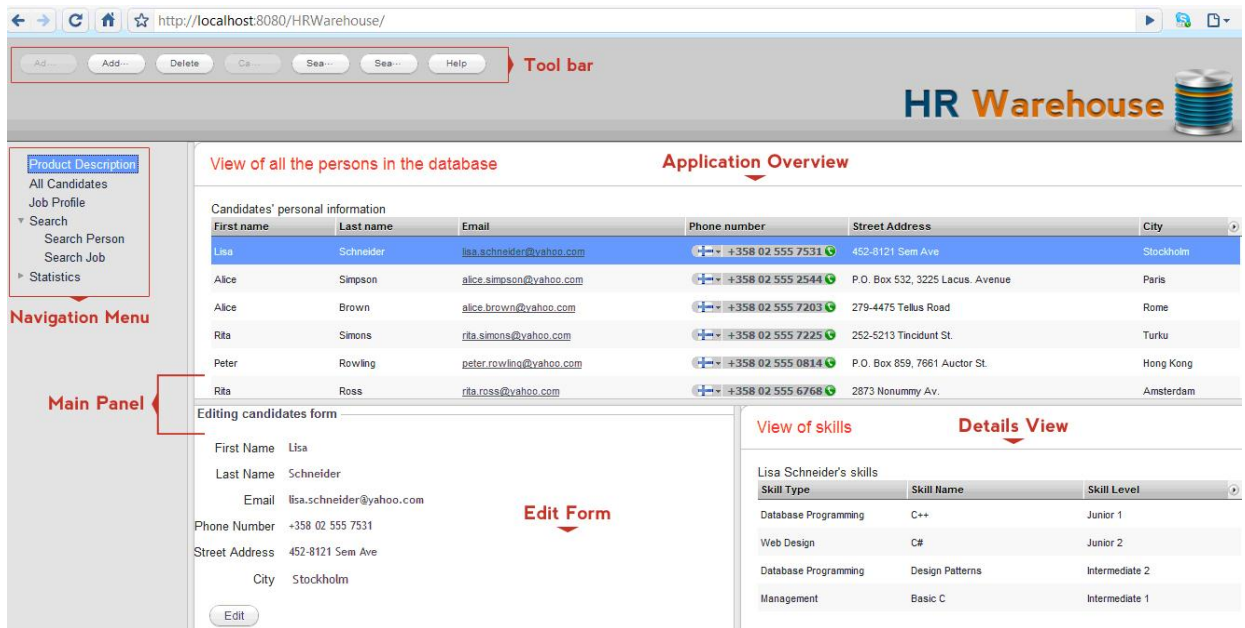


Fig. 1. Graphical organization of a HR Warehouse application

Figure 1 is a section of the HR Warehouse. To increase application quality and performance perceived by the user interface on the module with the principle structure, thus the organization views the application is done using SplitPanel panels with vertical or horizontal tables to accommodate all interconnected in the same window. Permanent elements that are part of each frame of the application are detailed below:

The functionality of the main panel is to act as a container, and to provide the necessary context for the user to perform operation of input/output application. In terms of space most of the display surface area of the application is occupied, but also all the components of the application, is resizable by the user. Navigation Menu functionality is to guide the user through application.



Fig. 2. Navigation Menu of the HR Warehouse application

To rank the application sections, the menu is represented as a navigation tree, with nodes indented to establish parent – child type relationships that are established between similar sections and functionality. Deployment option is available by selecting the downward triangle next to the parent node; Current node

is identified by selection of the dark – blue outline around it. When you select any node in the tree, the main panel presents the view for operating and the utility bar activates the action buttons compatible with current section.

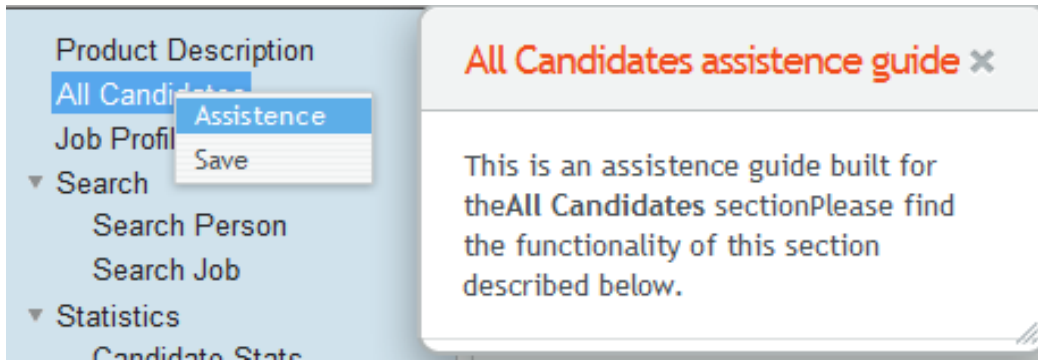


Fig. 3. Contextual menu of the HR Warehouse application navigation tree

Another functionality related to the navigation tree is shown in Figure 3 in the user friendly contextual menu to right click operation executed on any node. The options are

“Assistance” and “Save”. “Save” option saves the current state of the session and “Assistance” opens a pop-up which synthesize the possible actions for that section.

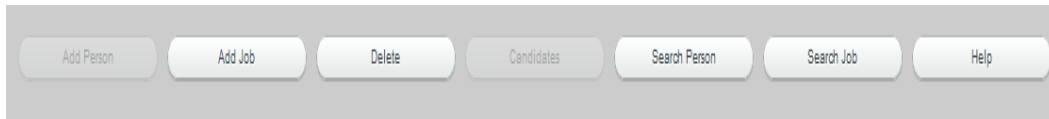


Fig. 4. HR Warehouse utility bar

Figure 4 describes another permanent element in the HR Warehouse structure, the utility tools. This component occupies the top of the display surface and consists of a sequence of buttons, each performing operation to add, search, and delete profiles of the candidates. The buttons are active or inactive, depending on the current section of the user and taking

into consideration the compatibility of the operation with that section. Regarding the utility, HR Warehouse address two main parts of the recruitment process: managing candidates and managing the vacancies. Managing candidates is available as a starting point by selecting the navigation node “All Candidates”.

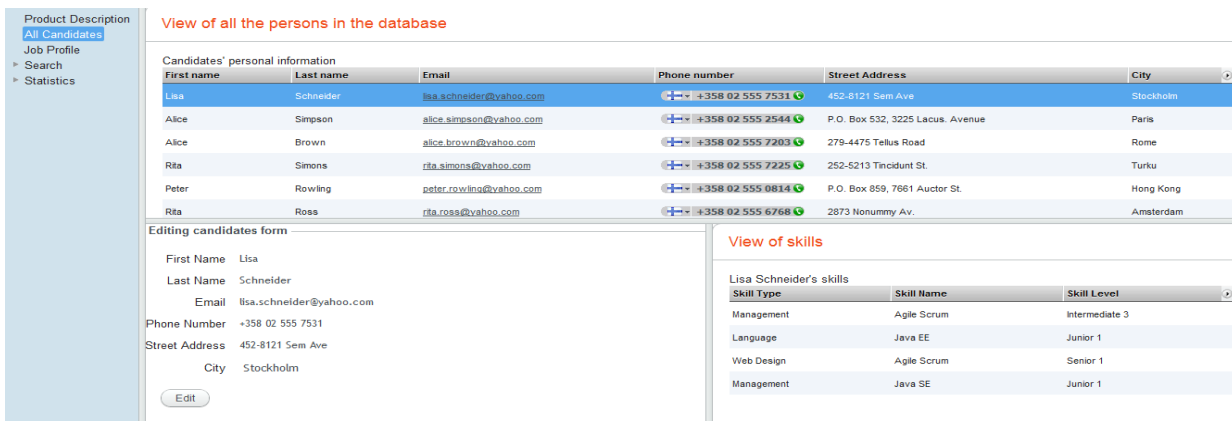


Fig. 5. All Candidates main section

From this view it will be explored the collection of existing candidates in the database. This is done in the upper section, in the table of personal information candidates. In the ta-

ble we found the name, email address, phone number, address and the city of the candidate.

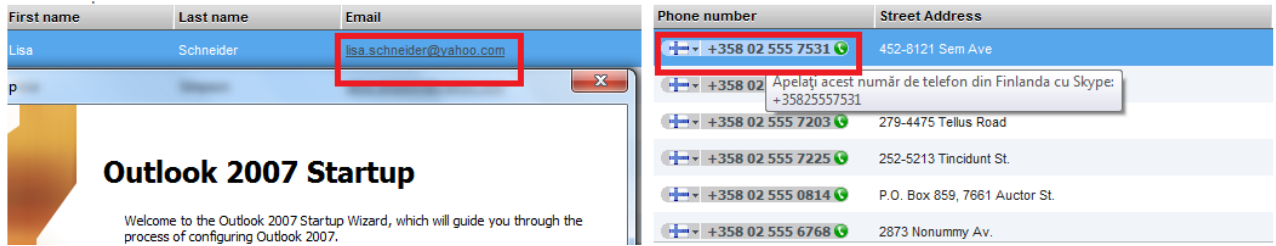


Fig. 6. Candidates contact information

With the field of mail and phone number, the HR department has the ability to contact the candidates with only one click. Accessing the hyperlink address for mail client will automatically start writing a new message and by pressing the phone icon will initiate Skype services to start a conversation. These two

features are illustrated in Figure 6. View table of candidates is made in the upper section, and in the table below, presented in Figure 5, you can see his collection of skills. Deleting a candidate is done by pressing the “Delete” button from the tool bar.

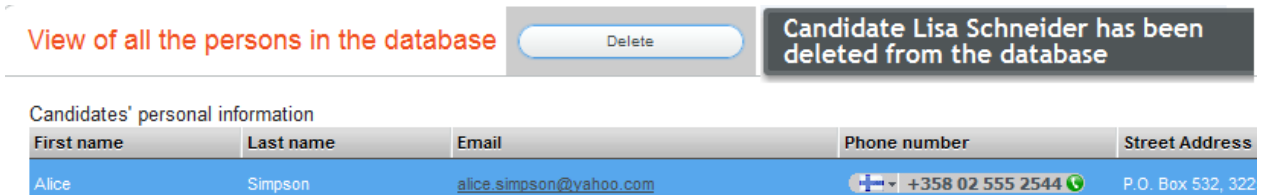
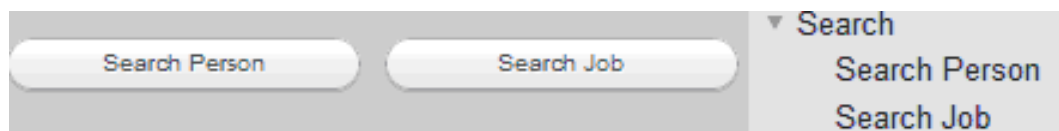


Fig. 7. Deleting a candidate from the database

In Figure 7 it is described the deleting operation. After pressing, the selected row disappear from the table and a notification message notifies its disappearance from the database. Simultaneously, the cursor moves to the next table entry. The next major section

of the application, with importance for all users is the multi – criteria search. It is available for the tow categories of information handled: those relating to clients or to the vacancies.



Acces - utility buttons

Acces - navigator tree

Fig. 8. Access ways to the search section

Relative to the description of Figure 8, all areas are homologous in terms of functionality and are accessible from the navigation tree by expanding the general search “Search” and select one of the children, “Search Per-

son” or “Job Search”, or through the homologous buttons from the tool bar. Views in the HR Warehouse are opened by opening the search action, are illustrated in Figure 9.

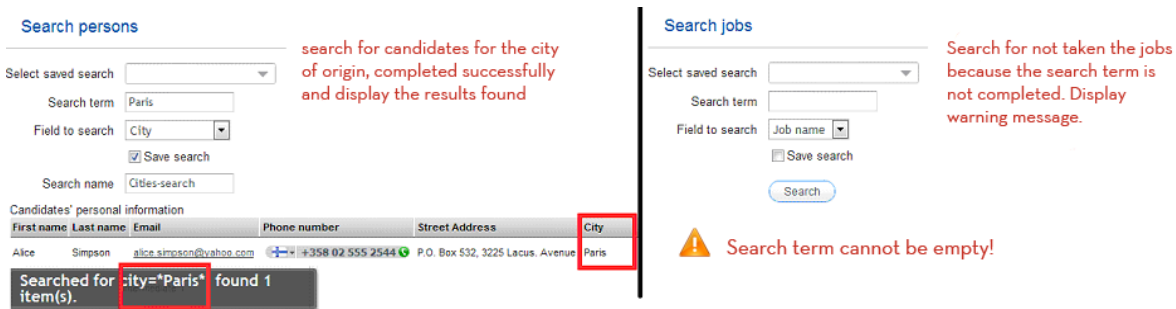


Fig. 9. Search action functionality

Summarized in Figure 9 is the search functionality, presented in parallel for both categories managed by the application. The search form allows you to enter the search term by which the search will be made, and the selection criteria by which filtering are performed. Last major section of the HR Warehouse is the statistic one. This meets the functionality

of high – level monitoring of the recruitment process, representing interest especially for the category of users that are part of the management to assess progress of the recruitment process. Their accesses are achieved by expanding the node “Statistics” in the navigation tree and select the appropriate child. Figure 10 illustrates the functionality of the areas of statistics.

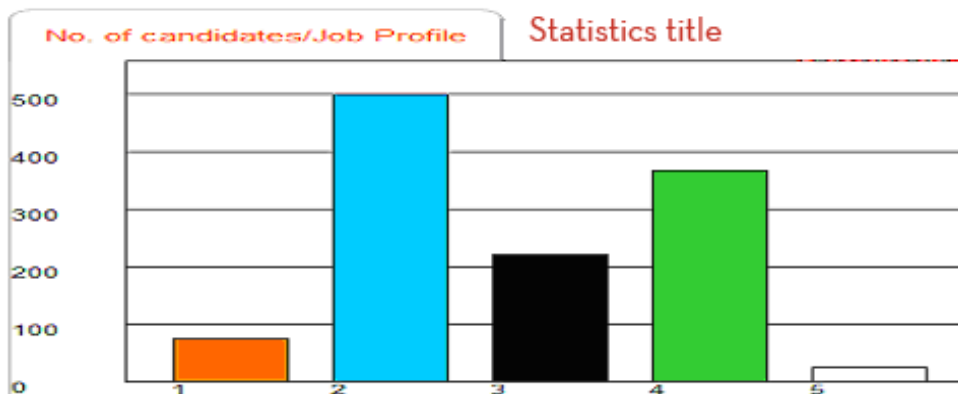


Fig. 10. Graphic display – statistic section

View panels are in the form tab type, which displays dynamic and animated graphics. The title is currently top of the tab. Chart shown in Figure 10 shows the number of candidates per vacancy in the company. The data used for this graph come from tables displayed in other sections. To achieve the necessary correlation between the number of candidates and profiles, run a query that extracts all relevant profiles based on ID field in the database.

3 Designing and implementing graphic interface of a HR Warehouse application

HR Warehouse application is a Rich Internet Application (RIA), having as main characteristic, the proximity to a functional desktop

application, and therefore subject to a high degree of interactivity. Thus, an important component in the application design it refers to building GUI. Graphic interface is the way for the user to enter data into the application, and the way that display the results. So it is justified to pay attention to inputs and outputs.

In the design process where identified 12 tasks through which the interface is operated. Each operation requires a graphic component that interacts with the interface, event or action that informs the user to begin processing the application, processing them effectively, the obtain result, its mode of display graphic component that hosts the outcome, and ways to improve interface quality. In addition to

disseminating operations is needed a diagram of graphics modules that interacts in HR Warehouse, and the roles attached to these modules.

Particularity of the profile module is his link with the management module of the filtered candidates for interview, the communication being accomplished by O6 operation.

Further details will be presented regarding the graphic interface implementation, based on the input formats typologies. HR Warehouse input formats are of several types:

Forms are the common input format for the application. They are built by extending the core framework classes of Vaadin. The HR Warehouse appearance form is simple. Presentation format of the fields is vertical. A `VERTICAL_LAYOUT` instruction is used. This approach streamlines the data entry.

A data source is a class that makes the transition from the database table and the interface component that needs information. Each data source type class extends the parent class `BeanItemContainer` from the framework. This class uses Java Generics parent so that they can be customized for a single data type. In addition, is still a component form used for data entry, often presented in the UI of the application: *pop-up window*. This type of

component is used when you need different levels of hierarchical data entry descent, used to retain the original context visible, and separation of independent sections, and make them visible by a specific animation to the window.

All classes that model such classes extend the “Window” class of the framework. As for the form in which manufacturer is the design aspect window, and the area that defines the types of events that will respond to the stimulus provided by the user. Defining events that meet the component is achieved through mechanism such *listener*. One of their most common uses is found when you want to describe actions following a button event, ore events related to the status window: closed, open and minimized. Further is presented a part of the code definition of events managed by the window to add technical qualities of a candidate, present in the application.

4 Application Design

The application is structured using packages in order to differentiate sections in packages having a single purpose. The structure is presented in Figure 11.

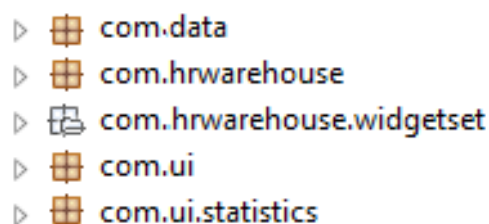


Fig. 11. HR Warehouse Application Package Layout

Thus, the main package, the one which is responsible with the entire application management is the “com.hrwarehouse” package; responsible also in making the required links between the packages.

All the data processing logic resides under the “com.data” package and is also responsible with the database interaction.

The package “com.hrwarehouse.widgetset” is a common package for all Vaadin based application being generated in order to hold the

generated code responsible in handling the created widgets.

The package “com.ui” holds all those classes responsible in defining all the application graphical components and its child “com.ui.statistics” contains all those classes used to build the logic required in gathering, processing and displaying statistical data.

Taking in consideration the fact that the application is a Vaadin based application, it is expected the fact that the entire design to be similar with the Vaadin framework. Thus, in

our approach, the main class, residing under the “hrwarehouse”, is called HrwarehouseApplication which extends the framework’ Application class. This class is the parent of each principal class from within a Vaadin based project.

The HrwarehouseApplication class plays an important role, being the main manager, responsible in communicating with all the project layers: data layer, business logic layer and with the presentation layer.

The application design uses the Plain Old Java Object (also known as POJO) concept, for creating 1 to 1 relationships between the Java object and its correspondent database table. Thus, a POJO describes the state and the object comportment throughout a set of attributes, accessed through getters and setters, and describes also the table prototype where the records are going to be persisted.

Within our application, six POJO resides. Each one of them, inherits an abstract class called AbstractPojo. Its role is to expose minimum functionalities, such as: assuring the ID field, persisted as primary key by the ORM; the version field generation, used to

facilitate the optimistic search in the database.

The AbstractPojo is not a manual implemented class due to the fact that is a class member of the org.vaading.approfondation.persistence.data, built upon JPA (Java Persistence Layer API), in order to ease the ORM implementation.

The JPA framework is the execution context for objects such as POJOs. The framework role is to manage the relational data used in the J2EE projects. Direct usage of JPA is rarely met. In most cases, the framework is used as integrated parts in complex libraries like Hibernate. In our application, JPA is used directly, without complementary libraries due to performance reasons:

- Complex libraries, like Hibernate, extend the application start-up time
- The written code grows in complexity, when objects are mapped to tables, contexts are initialized and session managed.
- The number of sixth POJOs is not big enough to justify the usage of libraries like Hibernate in order to gain on behave of their benefits in favor of application performance

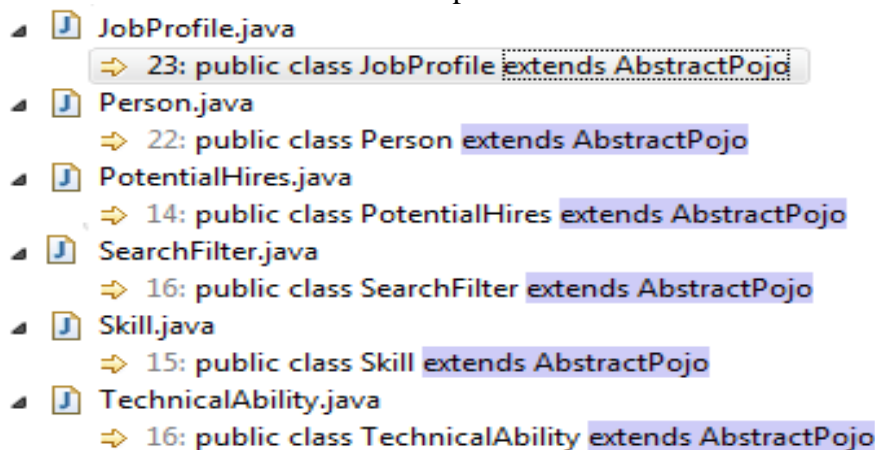


Fig. 12. Pojo objects

In Figure 12, the sixth objects, are listed. They are:

- JobProfile – class used to hold information about the company’s opened position
- Person – class used to hold the information about the post candidate

- PotentialHires – corresponding class for the potential hired person structure. It was necessary to include, candidates in two separated classes based on their nature, on their different evolution mod, from application perspective, candidates which do not reach the interview stage, and candidates which have the chance to go to the interview, thus becoming possible hires.

For the latest, supplementary information is required in order to be processed by the application, thus the new type is justified. From point of view of set inclusion, we state the fact that all candidates of PotentialHires type are found also in the Person types, but not the other way around.

- Search Filter – class holding filtering information. This class can hold search criteria for often used searches. Is the only class which doesn't have a link with POJOs.
- Skill – class responsible in holding the technical characteristics of the candidate, relevant for the CV. Based on it, the candidate's expertise level is established along with its compatibility with a given position
- TechnicalAbility – class responsible in holding the opened position technical specifications. Based on it, the position value, in points, is established and the compatible CV searched is performed.

In a tight link with POJOs, the database design is done. The way in which the database is aware of which class is mapped to which table, is done through the mappings found in

the persistence.xml file. This file is a part of the project structure, and besides mappings describes also the connection settings.

The file describes, all the required aspects for the persistence layers. The first step is to define the persistence unit. In our case, the unit is called „warehouse”, after the product name. Next, the persistence unit description is followed through the use of standard attributes. First attribute is the name of the persistence provider – in our case org.eclipse.persistence.jpa.

The next step is to define the list of classes, mapped to tables. Afterwards, the database driver properties are set. In our case the Derby standard properties are set, such as: the driver type – embedded (which supposes the fact that the driver is hosted within the Virtual Machine); the database name – HrDb; the execution mode – create update (meaning the database is created at the first application run, if not found); connection credentials; eclipse link standard settings, which handles the main persistence details and describes the way in which the tables are created on database startup.

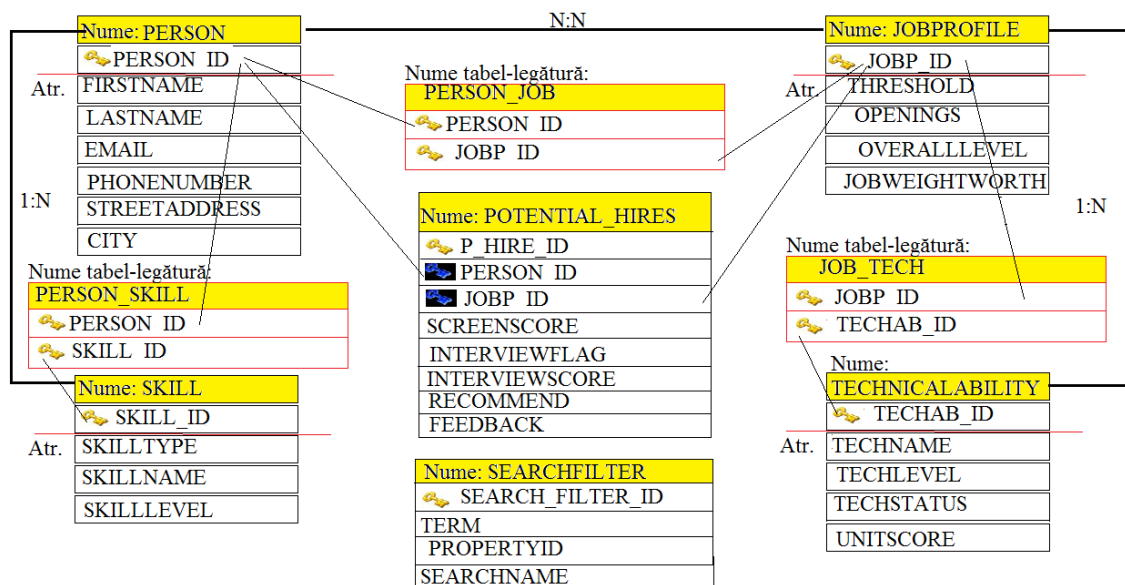


Fig. 13. Conceptual schema of application's tables

As shown in Figure 13, in the HR Warehouse application database table design, the number of POJOs is found. The relationship estab-

lished by the ORM between a POJO and its corresponding table is 1 to 1.

5 Conclusion

With a signified percent in the recruitment process is the technical interview, occasion what gives to employers and candidates the chance to interact, to evaluate each other and to notice if the collaboration is oportune.

The big number of CV's what reaches the HR department attract a considerably amount of time in which each candidates is identified, called and invited to the interview. This time effort and resources is not justified if the CV of a potential candidate is insufficient for the job, and because of this future investment becomes useless.

In the case of the multinational Juniper Networks, the statistics shows that only 1 of 10 candidates for a software development job has sufficient potential so that the technical interview phase is justified.

A component bigger than the forms, meet in the interface, through which the data is introduced, is the simple panel. This appears as an instance in the search section, where the user puts the criteria to be searched.

References

- [1] F.M. Boian and R.F. Boian, *Tehnologii fundamentale Java pentru aplicații WEB*, Editura Albastră, Cluj-Napoca, 2004, ISBN 973-650-131-0, 469 pg.
- [2] L.L. Constantine and L.A.D. Lockwood, *Software in Use: A Practical Guide to the Models and Methods of Usage-Centered Design*, Publisher: Addison Wesley Professional, 1999, ISBN-10: 0201924781, 579 pg;
- [3] D. Coward, "Java SE:A Youthful Maturity," *Oracle Conference*, April 2011;
- [4] M. Gronroos, *Book of Vaadin*, Oy IT Mill Ltd, 2010, 270pg;
- [5] I. Ivan, C Boja, *Practica optimizării aplicațiilor informatice*, Editura ASE, București, 2007, 483 pg;
- [6] T. Stitt, "An Introduction to High-Performance Computing," developed by the *Swiss National Supercomputing Centre, Switzerland as part of the LinkSCEEM Conference*, 6-8 October, Cyprus, 2009, 43 pg.



Marian-Pompiliu CRISTESCU has graduated the Faculty of Planification and Economic Cybernetics in 1985, he holds a PhD diploma in Economics from 2003, obtained at the Faculty of Cybernetics, Statistic and Economic Informatics from the Academy of Economic Studies in Bucharest. Between 1985 and 1991 he worked as an analyst – programmer at the I.A.C.M Computation Office in Olt – Slatina and at the Electronic Computation Territorial Center in Sibiu. In 1991 he joined the university teaching system, going

through all didactic positions starting with teaching assistant, lecturer in 2001 and assistant professor since 2008. Presently, he is a full assistant professor in Economic Informatics at the Faculty of Economic Sciences - "Lucian Blaga" University of Sibiu. He is the author of 8 books and over 60 scientific articles in the field of software quality, programming environments, data bases and economic informatics systems. He is equally focused on software development, being the author or co-author of over 25 programming systems for economic management. He has received many diplomas to certify his achievements in the scientific research domain. He has participated as a project director or as a leading team member at 8 research contracts. He is an active member of the scientific and editing committee for the following magazines and journals: Economic Informatics, Journal of Applied Quantitative Methods, The Economic Magazine edited by the Academy of Economic Studies of Moldavia – Chișinău and the „Lucian Blaga” University of Sibiu. He has participated in the scientific committee of over 10 national and international conferences, for the Informatics section and has coordinated the editing of 2 volumes with the projects of some international scientific conferences.



Ion – Liviu CIOVICĂ has graduated the Faculty of Science, in 2009 gaining a Bachelor of Science degree in Information Technology with a thesis on “Software application for medical diagnosis assisted by computer”. He is currently a student at the Faculty of Science master program, field of Cybernetics, Statistics and Economic Informatics.



Vasile-Laurențiu CIOVICĂ has graduated the Faculty of Science, in 2008 gaining a Bachelor of Science degree in Information Technology with a thesis on Translators and Interpreters for Code Generation and Software Optimization. In 2010 he gained a Master of Management degree in the field of Cybernetics, Statistics and Economic Informatics with a thesis on Intelligent Agents. He is currently a PhD student at Academy of Economic Studies in Bucharest. Between 2006 and 2010 he worked as a programmer at a company from Sibiu. Since January 2010 he works as an Independent Consultant. He is the author and co-author of more than 12 scientific articles in the field of software quality and optimization, code generation techniques, collaborative systems, data bases, programming environments and techniques, mobile platforms and economic informatics systems. Besides the scientific activity he is also an active software developer, being the author of few applications. Some of the created applications were presented to different student’s scientific conferences where he was distinguished with 1 excellence award, 6 first awards, 1 second award and 2 third awards. His area of interests includes among others: software quality, optimization techniques and algorithms, code generation techniques, economic informatics systems, intelligent and collaborative systems, mobile platforms.



Florin-Marius MARTIN is assistant professor at the Faculty of Economic Sciences, Lucian Blaga University in Sibiu. He graduated at the Faculty of Economic Sciences (LBUS) in 2008 in economy of trade, tourism and services specialization, obtaining the economist degree. He is a PhD candidate from October 2008 at the Economic Informatics Department from Academy of Economic Studies, Bucharest, Romania . He is the author and co-author of series of articles about the development of SMEs through the adoption of ICT, increasing competitiveness through innovation and information technologies, digital economy, etc. Field of activity is economic and digitizing business computing. His study focuses on the analysis of information and communication technology, electronic business and digital economy.