

Website for Human Resources Management in a Public Institution Using Caché Object-Oriented Model

Aurelia Pătrașcu

Petroleum-Gas University of Ploiesti, 39 Bucuresti Blvd., Ploiești, Romania
e-mail: patrascuaura@yahoo.com

Abstract

In this paper, the author presents the design of a website for human resources management. The first part of the paper presents the human resources management concept and describes the job analysis process. The main concepts of the object oriented technology and the Caché object-oriented model are also introduced. The second part of the paper contains a description of the application for human resources management that allows users: to add elements in the application, to display personal information of employees and their professional training, the departments they are working in, the salary, bonuses and deductions and to search the training or employment wage according to the employees' mark, the related employees according to the department code and also, the bonuses and deductions applied to employees on the basis of the bonuses or deductions code.

Key words: *Caché object model, Human resources management, job analysis*

JEL Classification: C88

Introduction

Management performance is necessary for an organization to develop and operate. The management practice is as old as organizations, which means a special seniority that dates from the organisation of social life in human communities (Emilian et al 2007).

One of the necessary conditions for Romania to align the human resources management to the existing requirements and standards on the EU market is represented by the rapid integration of the newest trends in the field of information technology.

Undoubtedly, the market of the software developed for the human resources departments has been and it still is dominated by the software designed for salaries. However, an extending trend of these solutions has been registered recently, transforming them from simple salary 'computing' into total human resources management software.

In this article, the author intends to develop a Website for human resources management in a public institution.

Human Resources Management represents the special part of the organization's general management which deals with the personnel issues in an integrated, global, interdisciplinary and professional vision and which consists in ensuring the fulfilment of the organization's objectives by recruitment, selection, development and its proper human resources use so as to fully

capitalize the professional and creative potential of all members (Emilian et al 2003; Mathis and Jackson 2007; Prodan and Rotaru 2006; Russu and Gheorghe 2004; Sims 2007).

Job analysis is the process that determines the tasks, activities, competences and specific responsibilities of a job, as well as the type of person recommended for it.

Job analysis consists in:

- *Job Description* which is a list of tasks, activities, competences, responsibilities, formal relations and working conditions of a job;
- *Job Specification* that represents the list of “human and professional requirements” necessary to the job employment, namely: knowledge, skills, experience and personal attributes (attitudes, personality characteristics, aspirations etc.).

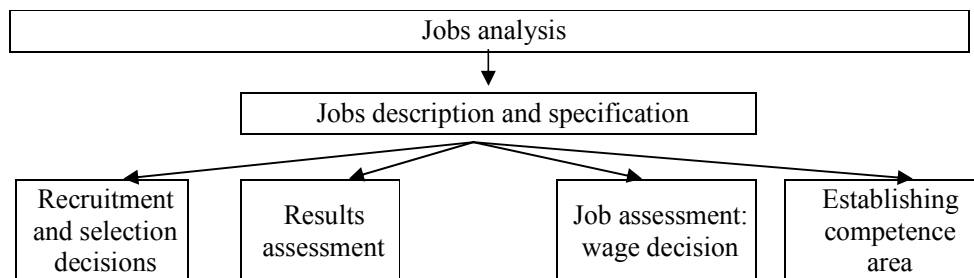


Fig.1. Using information from job analysis.

Source: Emilian et al, 2003.

The job analysis steps are:

Step 1: Specifying the purpose of the information, the type of data that must be obtained and the techniques that will be used for this purpose.

Step 2: Ensuring the required information acquisition. The following elements are used for this step:

- Organization charts;
- Process diagrams;
- Job description (if it already exists, it can be a good starting point from which an updated job description can be prepared).

Step 3: Selecting the representative positions for analysis. This is necessary when there are several similar positions.

Step 4: The proper job analysis, which leads to gathering information concerning the activities implied by the job, the working conditions and human requirements necessary to perform the activities. This step involves using one or more job analysis techniques.

Step 5: Verifying the information obtained through direct discussions with the person occupying the analyzed position or with his hierarchical manager. It is a good opportunity to ask for the job occupant’s agreement or to change the description of the activities he performs.

Step 6: Developing the job description and specifications as a result of the achieved analysis.

Object-Oriented Technology for Human Resources Management

Objects are key elements of object-oriented programming and represent real or abstract individual entities with a well-defined role within a system. An object has an *identity*, *state* and *behaviour*. Everything that objects know (*state*) and can achieve (*behaviour*) is expressed by means of sets of *properties* (or *attributes*) and *operations* (or *methods*). Thus, an object state is

given by its property values at a time. Operations are procedures or functions that allow you to change these values and implement the object behaviour (Lungu, Sabău and Velicanu 2003; Lungu and Bâra 2007).

Object-oriented applications are composed of many objects that interact and communicate among themselves via *messages*. A *message* represents a request addressed to an object for executing a certain operation. The message consists of three distinct elements: the object identity which the message is sent to, the name of the operation which is to be executed and a list of parameters necessary for the operation. The three pieces of information are enough for an object that receives a message to be able to perform the desired operation. Therefore, the messages transmission mechanism allows communication between objects in the different contexts (or processes) or on different computing systems. Objects' interfaces define their *type*. Objects that have the same interface belong to the same type. Since the interface is a subset of operations that defines objects' behaviour, it is also called *externally observable behaviour*.

Identifying a set of objects that have common properties and behaviours is called *classification*. *Class* is another basic concept of the object-oriented programming and represents an abstraction of common elements (*properties and operations*) shared by many objects and describes their implementation. Objects are concrete representations of classes and the creating process of an object based on the definition of a class is called *Instantiation*.

An association between two classes and their specific elements, as well as the objects they store and the interactions realized between these objects are simplified in Figure 2.

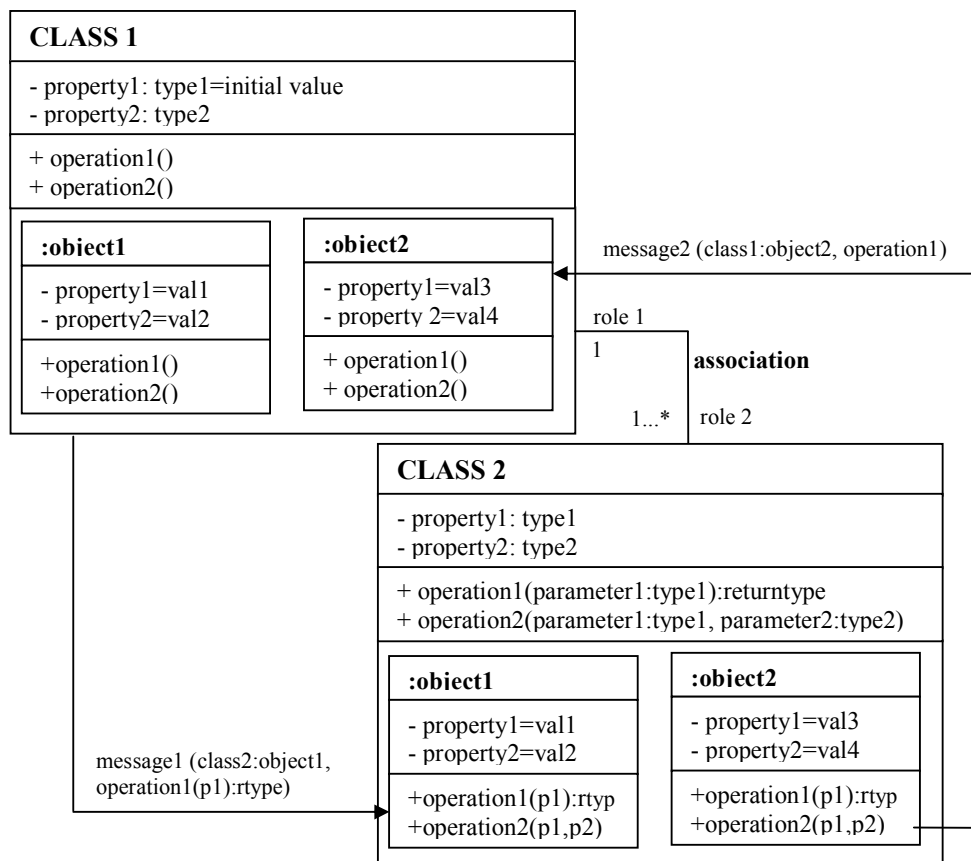


Fig. 2. A simplified representation of an association between two classes

The object-oriented model of the Caché postrelational database management system used at web site design adheres to the ODMG Standard from the Object Data Management Group. The basic operations for Caché Objects are based on object classes and the subsequent compilation of these to produce runtime executables. Caché supports all principles of modern object technology for creating, storing, loading and manipulating object instances (Intersystems 2010; Kirsten et al 2003; Pătraşcu and Tănăsescu 2007; Tănăsescu and Pătraşcu 2007).

Caché Server Pages (CSP) is a set of technologies built in Caché that offers the possibility of web applications achievement and the rapid development.

Dynamic web pages (CSP files) contain HTML or XML, so that they can be created or modified using any text editor or any tool for creating Web pages available on the market. Applications can be designed as if they were a series of static web pages. Navigating through these pages is automatically solved by Caché.

CSP uses *two methods* for building web applications:

- creating classes derived from % CSP.Page class, which generates HTML statements as a response to HTTP received requests;
- using HTML files, which are automatically converted to CSP classes.

These two techniques can be combined to achieve applications with maximum flexibility and for each of them Caché Studio can be used as programming environment.

Website design for human resources management

A Caché web application consists in (Intersystems 2010; Kirsten et al 2003):

- a set of web pages that define what the user actually sees;
- application code which is responsible for handling user's requests, processing the logic part of the application and storing and retrieving data;
- data that are stored on the server and include relatively static information.

Caché allows the class definition in two ways: using Caché Studio tool and using the Class Definition Language (CDL).

For the application on human resources management, Caché Studio tool was used. This is a standard Windows application that uses a number of windows to display and enable different aspects editing. The main components of the Caché Studio user interface are: *the classes / routines / CSP files editor*, the *project view* that displays the content of the current project tree structure, the *class inspector* and the *message window*.

The application includes nine classes: Angajat, AngajareSalariat, Departament, Pregătire, IstoricSalariu, Reţinere, ReţineriSalariat, Prima and PrimeSalariat that are presented in Figure 3.

Angajat class contains five properties (Table 1): Marca, Nume, Telefon, Adresa, DataNastere and two indexes related to the two properties, called MarcaIndex and NumeIndex.

Table 1. Characteristics of the Angajat class properties

Property name	Data type
Marca	%Library.Integer[Required]
Nume	%Library.String[Required]
Telefon	%Library.String[Required]
DataNastere	%Library.Date(FORMAT=4)[Required]
Adresa	%Library.String[Required]

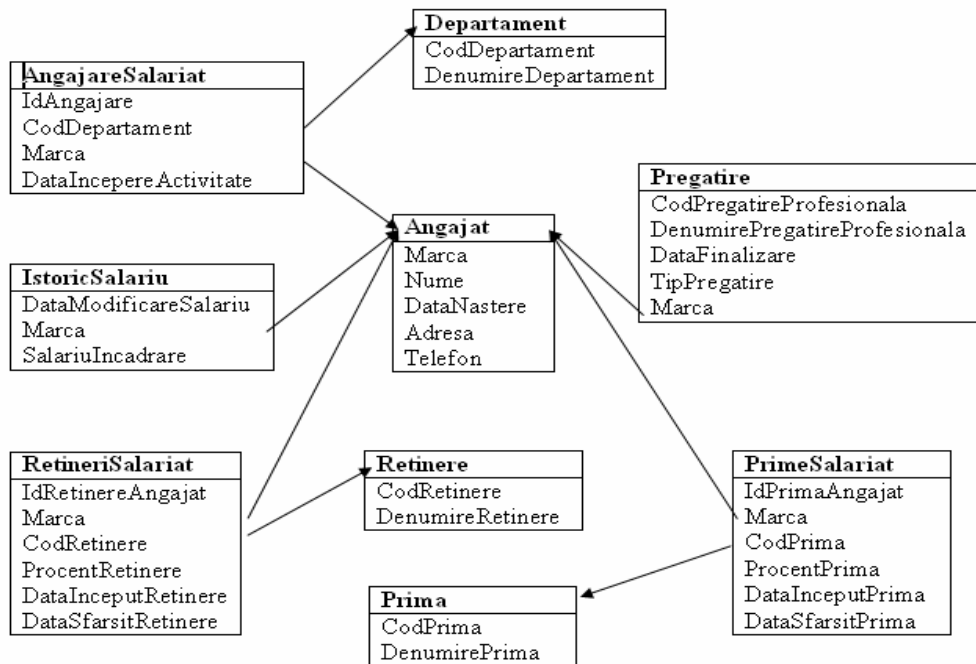


Fig. 3. Human resource management application classes.

After describing the nine classes, the information shall be filled in the database, the easiest method of adding data in a database being the use of a web form.

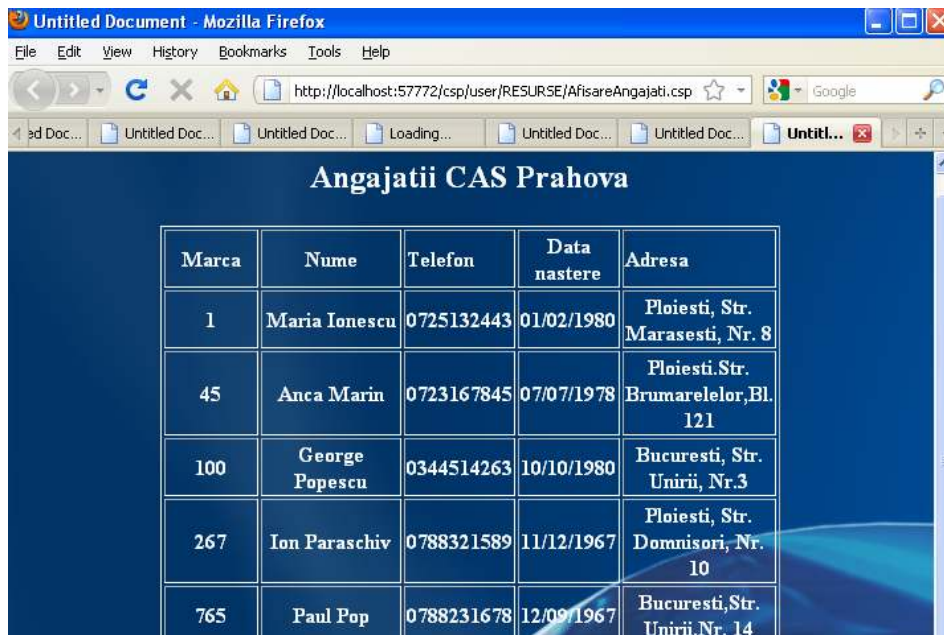
The application contains nine data input forms, each of them corresponding to one hierarchical class, namely: FormularAngajat (Figure 4), FormularAngajareSalariat, FormularDepartament, FormularPregatire, FormularIstoricSalariu, FormularRetinere, FormularRetineriSalariat, FormularPrima and FormularPrimeSalariat.

Fig. 4. Employee input form.

To view the information from the application classes, the following CSP files were created: AfisareAngajati.csp for displaying the CAS Prahova employees and their personal data, DepartamentAngajati.csp for displaying employees distributed by departments, PregatireAngajati.csp for displaying professional training of employees, SalariuIncadrareAngajati.csp for displaying employees' salary, PrimeAngajati.csp for showing raises granted to employees and RetineAngajati.csp to show employee related deductions.

For example, the author has chosen AfisareAngajati.csp web page (Figure 5), the other display pages being created in a similar way.

The page AfisareAngajati.csp has been created in Macromedia Dreamweaver to display the information stored in the Angajat class. This page was customized by placing the title "Prahova County CAS Employees" at the top of the page that was formatted using the Properties box. On this page, a table has been inserted to organize the list of employees as well.



Marca	Nume	Telefon	Data nastere	Adresa
1	Maria Ionescu	0725132443	01/02/1980	Ploiesti, Str. Marasesti, Nr. 8
45	Anca Marin	0723167845	07/07/1978	Ploiesti, Str. Brumarelelor, Bl. 121
100	George Popescu	0344514263	10/10/1980	Bucuresti, Str. Unirii, Nr.3
267	Ion Paraschiv	0788321589	11/12/1967	Ploiesti, Str. Domnisorii, Nr. 10
765	Paul Pop	0788231678	12/09/1967	Bucuresti, Str. Unirii, Nr. 14

Fig. 5. AfisareAngajati.csp page.

A CSP file that contains two frames has been created for performing a search in the application. In the web application for human resources management the following searches were made: search of employees' professional training by mark, search of employee by department code, search of employment wage by mark, search of employee's deduction by deductions code and search of employees' bonus by bonus code.

The following csp files have been created for searching the employees' training by mark: AfisareAngajat.csp that shows the Prahova CAS employees', CautareAngajat.csp that enables selection of the mark on which the training is shown and Cautare2.csp that brings together the two previous pages.

In the two frames that are displayed in dynamic lines (TopFrame and Mainframe), Cautare2.csp page (Figure 6) contains the information from pages CautareAngajati.csp and AfisareAngajati.csp.



Fig. 6. Cautare2.csp Page.

In CautareAngajati.csp page it was added a form that contains a button called Cautare and a list of values (called “select”) that can be used to choose the selection criteria. After enabling the Cautare button, the AfisarePregatire.csp page is shown in the Mainframe. The “select” list values are generated by the MarcaAngajat query defined in Angajat class. Each item in the list has a value represented by object ID and a text represented by Marca property. Further on, a fragment of the CautareAngajati.csp code is presented:

```
<body>
<form action="AfisarePregatire.csp" method="post" name="form1" target="mainFrame">
<span class="style2"><strong><strong>Cautarea pregatirii </strong> angajatilor dupa Marca</strong><
<select name="select" size="1">
<csp:QUERY NAME="ListaA" CLASSNAME="ResurseUmane.Angajat" QUERYNAME="MarcaAngajat">
<csp:while Condition="ListaA.Next() ">
<option value="#(ListaA.Get("ID"))#"(ListaA.Get("Marca"))#"></option>
</csp:while>
<input type="submit" name="Submit" value="Cauta"/>
</select>
</form>
</body>
```

AfisareAngajati.csp page contains a table with two rows and five columns that show the data on the employees of CAS Prahova sorted by Marca property using Listaangajati <csp:query> tag that contains the result of Angajati query execution defined in Angajat class. For exemplification, a fragment from the AfisareAngajati.csp page code is presented:

```

<body>
<div align="center" class="style1">
<p>Angajatii CAS Prahova</p>
<table width="" border="1">
<csp:query name="Listaangajati" classname="ResurseUmane.Angajat" queryname="Angajati">
<tr>
<td width="67"><div align="center" class="style4">Marca</div></td>
<td width="64"><div align="center"><strong><span class="style4">Nume</span></strong></div></td>
<td width="51"><span class="style8">Telefon</span></td>
<td width="73"><div align="center"><strong><span class="style4">Data nastere </span></strong></div>
<td width="61"><span class="style8">Adresa</span></td>
</tr>
<csp:while condition="Listaangajati.Next()">
<tr>
<td><div align="center" class="style9">#(Listaangajati.Get("Marca"))#</div></td>
<td><div align="center" class="style9">#(Listaangajati.Get("Nume"))#</div></td>
<td><div align="center" class="style9">#(Listaangajati.Get("Telefon"))#</div></td>
<td><div align="center" class="style9">#(Listaangajati.Get("DataNastere"))#</div></td>
<td><div align="center" class="style9">#(Listaangajati.Get("Adresa"))#</div></td>
</tr>
</csp:while>
</table>
<p><a href="Cautare.csp"></a><a href="Afisare.csp">

</body>

```

AfisarePregatire.csp page, that is loaded when the Cauta button from the CautareAngajati.csp page form is enabled, contains a dynamic query defined in SQL using the special tag <script> and the %request object.

For displaying the selected employee professional training, a table with two rows and four columns has been build, including dynamic query results. Next, an extract from the AfisarePregatire.csp page code is presented:

```

<script language="SQL" name="PregatireAngajati">
SELECT Marca->Marca,Marca->Nume,TipPregatire,DenumirePregatireProfesionala FROM ResurseUmane.Pregatire
WHERE #($SELECT(%request.Data("select",1) '=' " : "Marca=" _ %request.Data("select",1),1:"))#
ORDER BY TipPregatire
</script>

<table align="CENTER" width="" border="1">
<tr>
<td width="67"><div align="center"><strong><span class="style4">Marca</span></strong></div></td>
<td width="75"><div align="center"><strong><span class="style4">Nume</span></strong></div></td>
<td width="96"><div align="center"><strong><span class="style4">Tip Pregatire</span></strong></div>
<td width="137"><div align="center"><strong><span class="style4">Denumire Pregatire</span></strong></div>
</tr>
<csp:while condition="PregatireAngajati.Next()">
<tr>
<td><div align="center" class="style6"><strong>#(PregatireAngajati.Get("Marca"))#</strong></div></td>
<td><div align="center" class="style6"><strong>#(PregatireAngajati.Get("Nume"))#</strong></div></td>
<td><div align="center" class="style6"><strong>#(PregatireAngajati.Get("TipPregatire"))#</strong></td>
<td><div align="center" class="style6"><strong>#(PregatireAngajati.Get("DenumirePregatireProfesionala"))#</strong></td>
</tr>
</csp:while>
</table>

```

Human Resource Management web application consists of 40 CSP files that can be used: to *add* elements in the application, to *display* personal information of employees, their professional training, departments they are working in, employment salary, applied bonuses and deductions and to *search* the training or employment wage by employees' mark, the related employees

according to the department code and, also, the bonuses and deductions applied to employees on the basis of the bonuses or deductions code.

Conclusions

In this article the author aims to deal with the human resources management issue in a public institution.

The Human resources management system, created using the postrelational Caché database management system (DBMS), allows real-time update and query information about these activities, achieves data management, employees' management in the organizational structure, monitoring the employees' effectiveness and evaluating their performances.

The human resources management system was created using object-oriented technology and Caché object-oriented model because:

- The data and the data processing are no longer distinctly represented, but they are encapsulated in objects;
- The model used for the application design is flexible and easy to maintain;
- The common elements of all system's components are explicitly represented;
- The main concepts of the Caché object-oriented model are objects and literals (data types);
- Caché objects store persistent objects in database, using a storing strategy specified by the user.

References

1. Emilian, R., Țigu, G., State, O., Țuclea, C., *Managementul resurselor umane*, Editura ASE, București, 2003.
2. Emilian, R. (coord.), Țigu, G., Emilian, L., Schiopu, A., VasIU, M., *Management operațional*, Editura ASE, București, 2007.
3. InterSystems Corporation, *Caché*, <http://www.intersystems.com/cache/index.html> [accessed on July 1, 2010].
4. Kirsten, W., Ihringer, M., Kuhn, M., Rohrig, B., *Object-Oriented Application Development Using the Caché Postrelational Database*, Springer-Verlag, 2003.
5. Lungu, I., Sabău, Gh., Velicanu, M., *Sisteme informatice. Analiză, proiectare și implementare*, Editura Economică, București, 2003.
6. Lungu, I., Băra, A., *Sisteme informatice executive*, Editura ASE, București, 2007.
7. Mathis, R.L., Jackson, J.H., *Human Resource Management*, South-Western College Pub, 12 Edition, 2007.
8. Pătrașcu, A., Tănăsescu, A., "Managementul documentelor într-o universitate", în *Managementul cunoașterii în universitatea modernă*, coord.: Bodea C. și Andone I., Editura ASE București, pg. 180-204, 2007.
9. Prodan, A., Rotaru, A., *Managementul resurselor umane*, Editura Sedcom Libris, București, 2006.
10. Russu, C., Gheorghe, I., *Managementul resurselor umane. Note de curs. Studii de caz*, Editura ASE, București, 2004.
11. Sims, R.R., *Human Resource Management, Contemporary Issues, Challenges and Opportunities*, Information Age Publishing, 2007.
12. Tănăsescu, A., Pătrașcu, A., *The Publishing and Research Activity Document Management System*, International Scientific Conference "European Integration – New Challenges for the Romanian Economy, 3rd edition, Oradea, May 25-26, 2007, Analele Universității din Oradea, Seria Științe Economice, Ediție pe suport CD-ROM, TOM XVI, pg. 919-924, 2007.

Site Web pentru managementul resurselor umane într-o instituție publică folosind modelul orientat-obiect Caché

Rezumat

În acest articol, autorul prezintă proiectarea unui site web pentru managementul resurselor umane. În prima parte a lucrării se prezintă conceptul de management al resurselor umane și se descrie procedeul de analiza postului. De asemenea, se prezintă conceptele principale ale tehnologiei orientate-obiect și modelul orientat-obiect Caché. Cea de-a doua parte a lucrării conține o descriere a aplicației pentru managementul resurselor umane care permite utilizatorilor să adauge elemente în cadrul aplicației, să afișeze informațiile personale ale angajaților, pregătirea profesională a acestora, departamentele în care lucrează, salariul de încadrare, primele și reținerile aplicate și să caute în funcție de marca angajaților pregătirea sau salariul de încadrare al acestora, în funcție de codul departamentului salariații din departamentul respectiv și în funcție de codul primelor sau reținerilor, sporurile și reținerile practicate angajaților.