NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

THESIS

DESIGN AND IMPLEMENTATION OF A DATABASE FOR AN INTEGRATED SYSTEM FOR DAILY MANAGEMENT IN AN INDUSTRIAL AND COMMERCIAL ORGANIZATION

by

Noureddine Trigui

September 2004

Thesis Advisor:	Man-Tak Shing
Thesis Co-Advisor:	Doron Drusinsky

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2004	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Design and Implementation of a Database for an Integrated System for Daily Management in an Industrial and Commercial Organization			**5. FUNDING NUMBERS**
6. AUTHOR(S) Noureddine Trigui			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The purpose of this research is to define a centralized database containing all necessary information related to the daily management in an industrial and commercial organization that is publicly owned and equipped with civil personality and financial autonomy. The system is composed of the following subsystems:

- Subsystem "Human resource management"
- Subsystem "Provisioning"
- Subsystem "Financial, budgetary and accounting management"

The three subsystems should be installed in a central site and at regional sites. Each site will have its own database. The central database will be supplied with the data, which come from the other sites at the end of the day or according to need via modems. It is necessary to develop a tool for remote database queries in order to accomplish this work. The platform on which the application must be executed is IBM-INFORMIX running on top of the WINDOWS operating system. The database will be a relational database. The framework used in the design and modeling consists of:

- Object Oriented Analysis (OOA), which enables the development of high quality software by defining the problem structure.
- The Delphi Language, which provides a robust development environment.

The installation of the solution will be executed according to the following scenario:

- Client/Server architecture with the object oriented development tool DELPHI.
- The database will be installed on the central and regional servers.
- The application will be installed on the end users' stations.
- Data access will be through an open ODBC.

This software will present an integrated solution that will provide centralized and accurate data, so that data will be used to derive the right decision at the best time.

14. SUBJECT TERMS Database Design and Implementation, Object Oriented Analysis, OOA, Requirements Specification, Conceptual Model, Object Model, Delphi Language, IBM INFORMIX Database Management System, MERISE, Client Server Architecture, Use Case, Sequence Diagram, Relational Database, Entity-Relationship, Data Dictionary.			**15. NUMBER OF PAGES** 149
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

DESIGN AND IMPLEMENTATION OF A DATABASE FOR AN INTEGRATED SYSTEM FOR DAILY MANAGEMENT IN AN INDUSTRIAL AND COMMERCIAL ORGANIZATION

Noureddine Trigui
Major, Tunisian Army
B.S., Faculté des Sciences Economique et de Gestion de Sfax, 1989

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SOFTWARE ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2004

Author:	Noureddine Trigui
Approved by:	Man-Tak Shing Thesis Advisor
	Doron Drusinsky Thesis Co-Advisor
	Peter Denning Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The purpose of this research is to define a centralized database containing all necessary information related to the daily management in an industrial and commercial organization that is publicly owned and equipped with civil personality and financial autonomy. The system is composed of the following subsystems:

- Subsystem “Human resource management”
- Subsystem “Provisioning”
- Subsystem “Financial, budgetary and accounting management”

The three subsystems should be installed in a central site and at regional sites. Each site will have its own database. The central database will be supplied with the data, which come from the other sites at the end of the day or according to need via modems. It is necessary to develop a tool for remote database queries in order to accomplish this work. The platform on which the application must be executed is IBM-INFORMIX running on top of the WINDOWS operating system. The database will be a relational database. The framework used in the design and modeling consists of:

- Object Oriented Analysis (OOA), which enables the development of high quality software by defining the problem structure.
- The Delphi Language, which provides a robust development environment.

The installation of the solution will be executed according to the following scenario:

- Client/Server architecture with the object oriented development tool DELPHI.
- The database will be installed on the central and regional servers.
- The application will be installed on the end users’ stations.
- Data access will be through an open ODBC.

This software will present an integrated solution that will provide centralized and accurate data, so that data will be used to derive the right decision at the best time.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS

AS	Assumptions
BO	Business Objectives
CI	Communication Interface
CO	Design and Implementation Constraints
DBMS	Database Management System
DE	Dependencies
DFD	Data Flow Diagram
FE	Major Features
IDS	Informix Dynamic Server
ISYDMA	Integrated System for Daily Management Activities
MDA	Model Driven Architecture
MERISE	Method for the Study and Implementation of Business Information System
OE	Operating Environment
OMT	Object Modeling Techniques
OOA	Object Oriented Analysis
PE	Performance Requirements
PRAIN	Professional Association of Industry
RI	Business Risks
SC	Success Criteria
SE	Security Requirements
SI	Software Interfaces
SRS	Software Requirement Specification
UD	User Documentation
UI	User Interfaces

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank everyone who helped in the completion of this thesis. I especially would like to thank my advisors Dr. Man-Tak Shing and Dr. Doron Drusinsky for all their assistance. I also wish to express a heartfelt “thanks” to all the Software Engineering professors, and in particular, Dr. Shing, Dr. Michael and Professor Riehle for their extreme patience in attempting to impart their knowledge to a foreign student trying to learn new technologies.

I would like to express my undying love and gratitude to my wife, Hedgier, who has always stood by me in all my endeavors providing support and swift kick when necessary, and to my daughters Amina, Amal, and Olfa, whose mere presence is a constant reminder that I must be continuously learning.

Finally, I sincerely thank all NPS staff for providing me the necessary help. Without all of your contributions this would not have been possible. Thank you.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

The utility of databases (DB) is evident in today's world. It is often necessary to store and retrieve data via a database in real time. This project aims to design a centralized database that contains all relevant information of the daily management activities of the Professional Association of Industry (PRAIN) and tools for querying the database locally and remotely.

Geographically, the users are distributed at many sites. Their desire to have access to the database will be met by the development of two tools, one for the server and another for the client. The server contains the database itself with all information stored in binary format. The client contains programs that allow querying and updating the database via a simple and friendly graphical user interface.

The realization of this project is based on the following methodology. The analysis utilized in this research is the Object-Oriented Analysis (OOA). It is inspired by several authors such as Coad and Yourdon, Davis, Rumbaugh and others [Coad and Yourdon, 1992; Rumbaugh and Al, 1996; Davis, 1993]. The purpose of this analysis is to specify the database by using the French method "MERISE" [Gaby, 1993] in order to specify the conceptual data model (CDM). Another tool used is Power AMC 9 to draw the conceptual and the object models presented in Figures 5, 6, 7, 9, 10, and 11. This tool supports both the MERISE and OMT methods for the conceptual, physical and object models. To realize the development of the project, the author used Delphi as a development language and IBM-Informix as a Database Management System (DBMS).

B. AREA OF RESEARCH/SCOPE

The majority of the Tunisian organization applied a French design methodology that allows their teams to complete their projects successfully, within the cost and time planned. This method is called **MERISE** (Method for the Study and Implementation of Business Information System). It is a dynamic modeling method, which models the behavior of an information system during the analysis and design phases.

The MERISE method is based on the separation of the data and processes to be executed in several conceptual and physical models. This separation ensures the longevity of the model.

The goal of this section is to present several rules that allow the transformation of a MERISE 1 or 2 non-object-oriented design to a MERISE 3 object-oriented design, which requires a short introduction of the MERISE method.

1. Introduction to the MERISE Method and the Actual Problem

The first version of MERISE was introduced in 1978-1979 with the aid of several branches of the French government. By the mid-1980's, it became a standard in France and other predominantly French speaking countries in financial information system development. MERISE is a systemic method that allows for a global definition of information [1]. The first version of MERISE tried to create a proposed model from different aspects such as organizational and technical as well as the strategy of the company. This is achieved through three levels, the conceptual, organizational and operational levels.

a. Conceptual Level

The goal of this level is to model a database and make the necessary changes to the information system without modifying the organizational aspect, thereby dictating the functions of the current system. The conceptual level is comprised of the following models:

- CDM (Conceptual Data Model), equivalent to an **Object/Class Diagram.**
- MCT (Conceptual Process Model); equivalent to a **State Diagram.**

b. Organizational Level

The goal of this level is to apply the concepts derived in the conceptual level to incorporate a time frame for the project, the scope of the project and the actors to be involved. The Organizational level is comprised of the following models:

- MOT (Organizational Process Model); equivalent to a DFD**.**
- MLD (Logical Data Model)

c. Operational Level

This level will execute the implementation of the techniques presented in the previous levels. The Operational level is comprised of the following models:

- MOT (Organizational Process Model)
- PDM (Physical Data Model).

In order to take into consideration the extensions and the improvement related to the organizational and technical evolution, MERISE 2 was created. The entity-relationship model used in the first version of MERISE for the data modeling experienced several deficiencies. One group of researchers introduced an extension to the model, the concepts of generalization and specialization in order to deal with the concepts of inheritance and integrity constraints in November 1990. MERISE 2 provides the conceptual level the Data Flow Diagram (DFD) and the Conceptual Model Process Analytical (MCTA) for the process, which handles the data related to the process during the design phase, and the concepts of life cycle of an object in order to take into consideration the steps followed by an object during the life cycle.

After the conceptual level, MERISE 2 takes into consideration the organizational level at which the organization, the human resources and the budget were created. For the logical level, MERISE 2 defines the user interfaces and the data distribution. Finally, the physical level remains unchanged.

The third version of MERISE, OOM, dated 1992, is completely marked by object oriented concepts.

2. Transformation to Object Oriented Method

A study was completed by a group of researchers [2] whose main goal was to design a reengineering platform for legacy systems. The group proposed a rule-based approach for a systematic object-oriented transformation of a MERISE analysis. Those rules resulted from a comparison of the MERISE method and the OMT method. Table 1 shows the similarities between the two method's model.

MERISE	**OMT**
Conceptual Data Model (CDM) Entity-relationship diagram **Rule example**: *a type of entity becomes a class*	Object Model Object/Class diagram
Flow graph (MCC) Actor-flow graph **Rule example**: *an actor become a class*	Scenario Event flow diagram
Process Conceptual Model (MCT) MCT diagram **Rule example**: *the initial event becomes a particular state*	Dynamic Model State diagram
Process Organizational Model (MOT) MOT Diagram **Rule example**: *an operation become a process*	Functional model Data Flow diagram

Table 1. Comparison of the MERISE and OMT Methods

3. Results and Possible Improvements

The transformation of a MERISE analysis into an objected oriented technique will improve the maintenance process and the software quality. The transformation from a MERISE analysis schema to an object-oriented analysis schema is an ongoing research, which will lead to the realization of a semi-automatic tool for this transformation. [2]

C. METHODOLOGY

1. Why Object Oriented Methodology

Object Oriented Methodology (OOM) is a systematic development approach encouraging and facilitating re-use of software components. With this methodology, a computer system can be developed on a component basis, which enables the effective re-use of existing components and facilitates the sharing of its components by other systems. The adoption of OOM can achieve higher productivity, lower maintenance costs and better quality [3].

The keys reasons and advantages of the Object Oriented Analysis (OOA) are presented for guiding two categories of people. For the project manager, this section explains the need to encourage the team members to utilize the OOA. For a team member, this section presents one argument to use to convince responsible personnel of the importance of the OOA.

Coad and Yourdon (1992) have developed the OOA of which the concepts are derived from the semantic modelisations of data and the Object Oriented languages.

The following are several reasons and the advantages of using the OOA.

- Better tackle the questions specific to the domain of the problem. The OOA stresses the comprehension of various domains of the problem.
- Ameliorate the interaction between the expert domain of the problem and the analyst. The OOA organizes the analysis and the specification by utilizing organizational methods that are closest to human thinking.
- Increase the internal coherence of analysis results. The OOA reduces the separation between the different analysis activities by treating the attributes and the methods as unified entities.
- Clearly represent the common elements. The OOA utilizes inheritance to identify the common elements of the attributes and the methods.
- Enhance stability when evolving software for requirement change and when constructing a similar system.
- Promote code reuse.

D. ENVIRONMENT

1. Delphi 7

Borland Delphi7 Studio is the first step for the Delphi developer in the migration from Win32 to Microsoft.NET development for the Windows platform. Delphi 7, like its highly successful predecessors, is a Win32 development environment with new Model Driven Architecture (MDA), Web application development, cross-platform features, and pre-release technology designed to assist the Delphi developer in entering the world of .NET. With the new Delphi environment, developers can also port their applications cross-platform to Linux, potentially increasing their return on investment. By integrating leading development solutions into a single easy-to-use package, Delphi 7 simplifies the application life cycle and speeds time-to-market.

2. IBM Informix Dynamic Server (IDS 9.30)

The IBM Informix Dynamic Server continues a long standing tradition within IBM and Informix of delivering a first-in-class database engine. It combines the robustness, high performance, availability and scalability needed by today's application. [10]

E. ASSUMPTIONS

Throughout this thesis, the assumption is that the reader is familiar with object oriented programming techniques, and has a general understanding of UML representation and the SQL language.

F. ORGANIZATION

This thesis is divided into five chapters. Chapter I presents the background of the problem, the area of research, the methodology and the environment used. Chapter II describes the Requirements Analysis through use cases and the development of a conceptual model. Chapter III details the design phase. Chapter IV provides a prototype developed in the Windows environment with Delphi 7 using IBM Informix Dynamic Server 9.3 as the database. Chapter V provides a conclusion and recommendations for future work.

II. REQUIREMENTS SPECIFICATION

Requirements are defined during the early stage of system development as a specification of what to implement. They are descriptions of how the systems should run, application domain information, constraints on the system's operation, or specifications of a system's property or attribute.

The purpose of this section is to identify and document requirements for the new integrated system in a form that clearly communicates the intent of the PRAIN organization.

It is necessary to recognize the importance of correct and thorough requirements specification as one of the most important parts of the design effort. The detailed specifications resulted from discussions. These requirements were established to provide enough information regarding the system to make it possible to begin contemplating the conceptual model for the software engineering effort.

The primary goal of developing the Integrated System for Daily Management Activities (ISYDMA) in the PRAIN organization is to provide a capability for investigating problems using an efficient automated tool from a central or regional location. The system should provide an intuitive graphical user interface encompassing all functionality of the current ISYDMA system. In addition, it should be designed so as to provide the capability for code reuse. In order to facilitate rapid application development methods, the system must be implemented using the Informix Database and the Delphi development tool. It must be able to run on all Intel Pentium X (or compatible) platforms running Microsoft Windows 98 or any more recent operating system. Finally, to the maximum extent possible, the system should be developed to insulate it from compatibility problems associated with upgrades in operating systems, programming languages, and versions of Informix Database.

The ISYDMA system must be compatible with many different types of hardware ranging from notebook and desktops on the client side, to large enterprise servers from server sites. The system must be able to process data in real-time and should provide an "adequate" level of usability based on the following hardware specifications.

A. HARDWARE SPECIFICATION

1. Server Side

- Computer server architecture: Intel or compatible 1 GHz or higher;
- Memory (RAM): 512 MB min;
- Hard Disk space: 120 GB or higher;
- Monitor: 800 X 600 or higher resolution required;
- Mouse: Microsoft or compatible;
- CD-ROM: required.

2. Client Side

- Computer CPU: Intel or compatible 233 MHz or higher;
- Memory (RAM): 128 MB min;
- Hard Disk space: 40 GB or higher;
- Monitor: 800 X 600 or higher resolution required;
- Mouse: Microsoft or compatible;
- CD-ROM: required.

Two versions of the program are required. One is for the central site and the second for the regional sites. Specific requirements for the central version are needed. Care should be taken to provide as many opportunities as possible for code reuse.

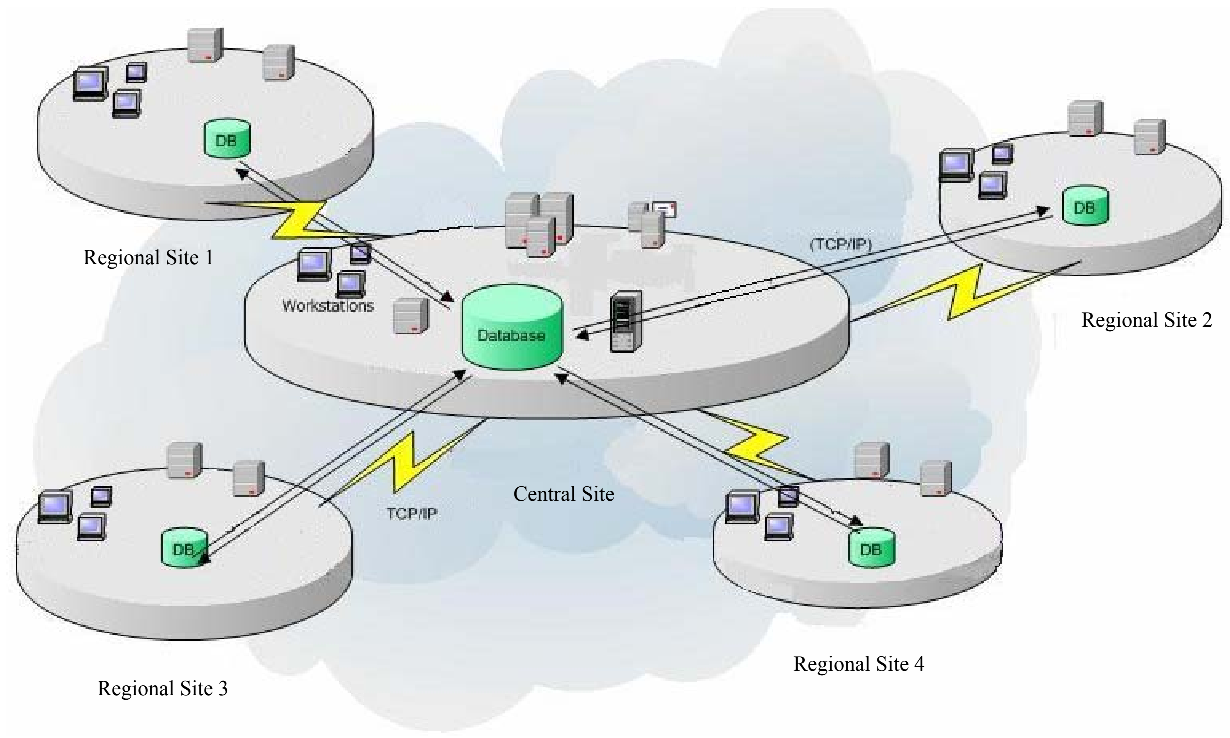


Figure 1. System Architecture

B. VISION AND SCOPE DOCUMENT

1. Business Requirements

a. Background, Business Opportunity, and Customer Needs

Within the framework of the implementation of its handbook of administrative, financial and technical procedures, the PRAIN proposes to realize a mission of study and data-processing applications with the following responsibilities:

- The study and design of the new data-processing application integrated in a total system;
- The realization of the various applications of the system;
- The installation of the various applications and assistance for new applications.

The objectives of this mission are:

- Ensure coherence between the organization and management objectives identified by its organizational charts;
- Ameliorate the efficiency of the data-processing tools;
- Optimize the information circuits in order to ensure better management of the resources;
- Assure better quality services and better follow-up budgetary processes;

- Facilitate communication between the various units;
- Provides the regional sites with their own management systems;
- Allow the central site to be centralizing and a distributor of information;
- Develop the applications in a uniform system.

b. Business Objectives and Success Criteria

BO-1: Reduce the wait time in the manual procedures.

BO-2: Increase the average effective work time by 50% for every activity.

BO-3: Have a reliable information system.

BO-4: Have a good archival system that will serve as system reference.

SC-1: Have the majority of the employees who presently use the manual system use the automated system for managing the overall daily activities.

SC-2: Achieve an increase in the satisfaction of users from the new system.

c. Business Risks

RI-1: Too few employees might use the system, reducing the return on investment from system development and changes in management operating procedures.

RI-2: Some employees might be afraid to apply the new system's procedures, which would reduce employee satisfaction with the system and possibly its usage.

2. Vision of the Solution

a. Vision Statement

For many industrial organizations desiring an automated system to help employees manage daily activities, the ISYDMA system is a client-server application that will integrate different subsystems in order to manage human resources, provisioning, financial, budgeting, and accounting. Every organization will save time and will increase the productivity of their employees with this system.

b. Major Features

FE–1: Human resource management

- Personnel management
- Payroll
- Education process

FE-2: Provisioning management

- Managing cash purchase

- Managing purchase orders
- Managing markets

FE-3: Accounting management

FE-4: Budget management

FE-5: Finance management

FE-6: Performance

- Fast Response

FE-7: Security

- Maintain privacy and data integrity.

c. Assumptions and Dependencies

AS-1: Computers and printers will be available in every unit at the central and regional sites to permit users to process the daily management activities with accurate information.

AS-2: Technical staff will be available to assist all users upon implementation of the system.

DE-1: If connectivity to the server is lost, it is possible to use a computer from another unit to access the database.

3. Scope and Limitations

a. Scope of Initial Release

Features have been prioritized and scoped to available resources. Throughout this thesis, it is assumed that the reader is familiar with object oriented programming techniques as well as a general understanding of UML Notation and SQL language.

4. Business Context

a. Stakeholder Profiles

Stakeholder	***Major Value***	***Attitudes***	***Major Interests***	***Constraints***
Database Administrator	Improved database performance and reliability, ensure the continuity of the productivity	This project should not compromise existing systems nor add an administrative burden	Security; Added Network and CPU load; Administrative overhead	Minimal impact to existing systems; Little if any ongoing maintenance activities
Regional interest user	Needs to set up the system in order to have coherent information	Cooperative	Minimal new technology needed; concern about transferring information	Network performance and might not have technical staff support
Central interest user	Needs to set up the system in order to collaborate with the regional sites	Cooperative	Concern about having accurate information and the method of obtaining said information	Availability of network
Special Interest user	Needs accurate information	Cooperative	Concern about having accurate information in order to update the databases	Availability of data storage

Table 2. Stakeholder Profiles

b. Project Priorities

Dimension	**Driver**	**Constraint**	**Degree of Freedom**
Schedule			Release 1 planned to be available by 12/31/04, release 2 by 05/30/05; delays of approximately three weeks is acceptable
Features		All features scheduled for release 1.0 must be fully operational	
Quality		95% of user acceptance tests must pass; all security tests must pass; compliance with corporate security standards must be demonstrated for all secure transactions	
Staff	Projected team size is half-time project manager, two developers, and half-time tester; additional half-time developer and half-time tester will be available if necessary		
Cost			Budget overrun up to 15% acceptable

Table 3. Project Priorities

C. SOFTWARE REQUIREMENT SPECIFICATIONS (SRS)

1. Introduction

a. Purpose

This SRS describes the software functional and nonfunctional requirements for release 1.0 of the Integrated System for Daily Management Activities (ISYDMA) in PRAIN. This document is intended to be used by the members of the project team implementing and verifying the correct functioning of the system. Unless otherwise noted, all requirements specified here are of high priority and committed for release 1.0.

b. Project Scope and Product Features

The ISYDMA in PRAIN will support the integration of distributed cross-disciplinary data sources into coherent knowledge bases for managing daily activities. Section B.1 provides a detailed project description and section B.2 lists the features scheduled for full or partial implementation in this release.

2. Overall Description

a. Product Perspective

The ISYDMA in PRAIN is a new system that aims to assemble distributed cross-disciplinary data sources into a coherent knowledge base to support collaboration within the different sites. The context diagram in Figure 2 illustrates the external entities and system interfaces for release 1.0. The system is expected to evolve over several releases.

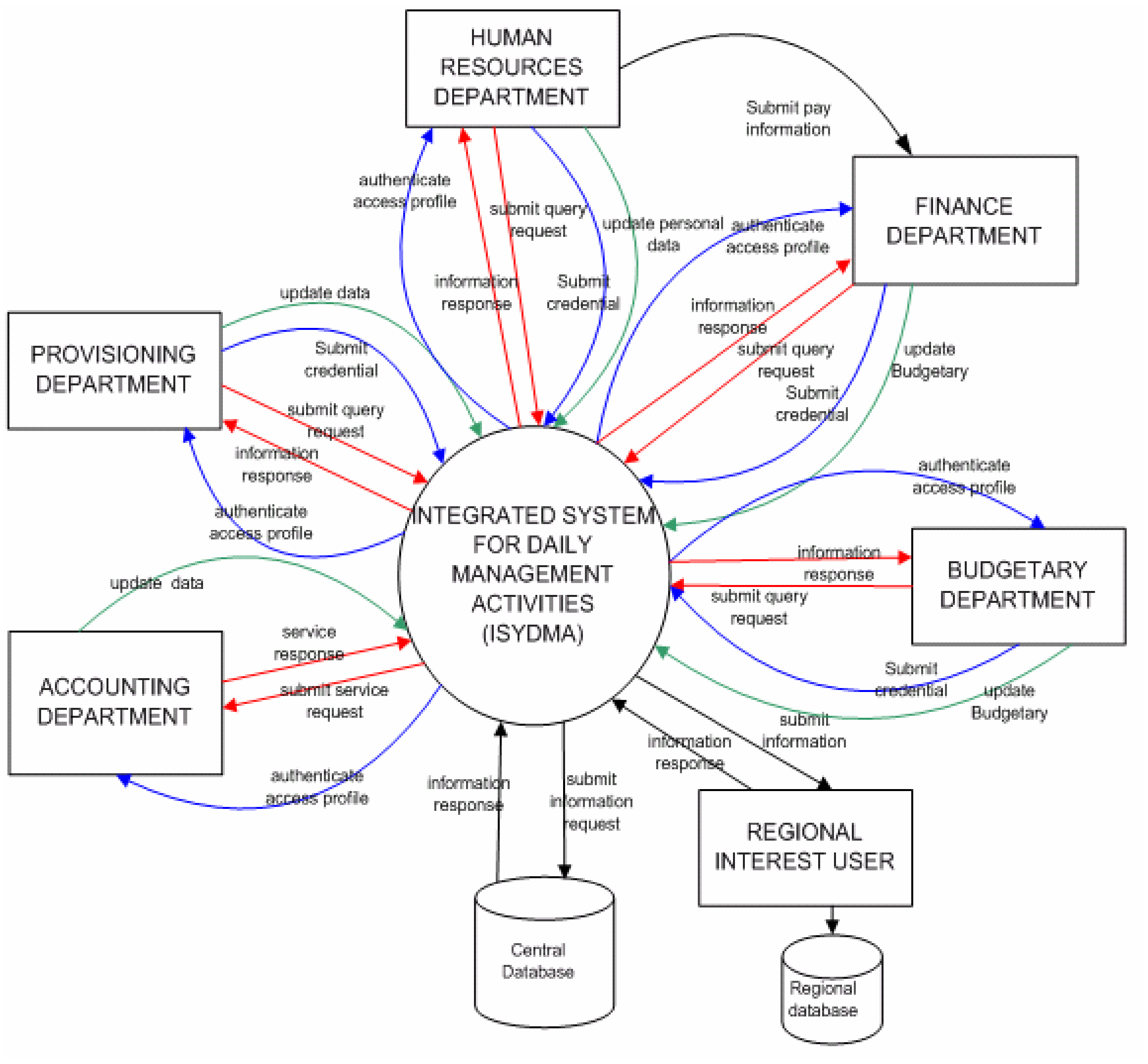


Figure 2. Context Diagram for Release 1.0 of the Integrated System for Daily Management Activities.

b. User Classes and Characteristics

User Classes	Characteristics
Human Resources department	The Human Resources department is responsible for the follow-up of the recruitment and attendance, promotion process, managing the personnel carrier, managing the personnel vacations, preparing personnel payroll as well as managing per diem, bonuses, advances, professional performance review, and the educational process.
Provisioning department	The provisioning system is responsible for the management of suppliers, purchase orders, invoices, and price quotations.
Accounting department	The accounting system ensures the following functionalities: add and modify the system constants and the entries, and the management of the immobilization.
Finance department	The finance system closely linked to the human resource system. It is also responsible for personnel payroll, contractor payment and advances, supplier payment, credit management, and management of the central cashier.
Budget department	The budgetary system permits budget planning and proposal for pledging funds, the definition of the different parameters and nomenclatures, and the follow-up of the annual budget.
Database Manager	Responsible for the security of the overall system

Table 4. User Classes and Characteristics

c. Operating Environment

OE-1: The ISYDMA shall operate with a graphical user interface and client-server architecture.

OE-2: The ISYDMA shall operate over a secure data transmission to ensure security of personal and sensitive information.

d. Design and Implementation Constraints

CO-1: The system's design, development, and maintenance documentation shall conform to the IEEE 1016 [5], 1074 [6], and 1219 [7] standards

e. User Documentation

UD-1: The system shall provide a help system that illustrates all system functions.

f. Assumptions and Dependencies

AS-1: The system aims to facilitate and improve the quality of the services offered.

DE-1: The operation of the ISYDMA depends on the availability of the network when a query is requested from the system.

DE-2: The operation of the ISYDMA depends on the performance of the database and the overall system.

3. System Features

a. Pay Processing

The system shall provide a functionality that allows updating the pay components. Each component is identified by its code and designation. The property of each component and its type are obligatory and are used at the time as pay generation. The assignment of the components of the pay is automatic and occurs at the same time as the creation or updating of the statutory personal card. The amount of the component for each person is calculated based on the associated function, category, and rank. Certain component amounts are calculated based on the assigned percentage (supplementary hours, per diem, and production bonuses)

b. Codification

The system shall provide a functionality that allows creating and updating any components used for the pay module. Those components are classified by function, rank, category, and type. Classification determines the amount of the component for each person.

c. Personal Information

The system shall also provide a functionality that allows the creation and the establishment of personal information. This interface must provide two parts. One is the identification part and it contains the ID, first name, last name, photograph and main department. The second part contains information about personal, administrative, and family data as well as fixed and variable components.

d. Attendance Processing

The system shall provide a functionality that allows the introduction of personnel attendance for every month of pay. It presents the personnel ID, the number of days absent, and the type of absence, either a justified or a non justified absence.

e. Leave Processing

The system shall provide a functionality that allows the tracking of the leave process. This interface shall contain the year of the leave, personnel ID, the references of leave requests, dates of leave, type of the leave, and address during leave.

f. Traveling

The system shall provide a functionality that allows the recording of the traveling done by person for each month of pay. This interface primarily contains the year of pay, the month of pay, personnel ID, type of travel (short duration or long duration), the number of nights and the amount.

g. Supplementary Hours

The system shall provide a functionality that allows the recording of the supplementary hours (overtime) done by person in each month of pay. It should contain the year and the month of pay, the personnel ID and the number of hours worked. Following the validation of the number of hours worked, the system will ascertain the hourly rate and determine the overtime rate, which depends on the personal category and type.

h. Performance Review

The system shall provide functionality related to the performance review and absences during the six-month period. Two types of performance reviews are professional and production bonuses. The interface should contain the year of the pay, the review period 1 or 2 consisting of six months each, personnel ID and the number of days absent.

i. Medical Expenses

The system shall provide a functionality that allows the recording of the medical expenses that should be refunded. The act of recording is completed from the invoice provided by the insurance company. The interface should contain the year and the month of the pay, the insurance invoice number, the invoice's total amount, the affiliation insurance number, personnel ID, name and the amount to be refunded.

j. Temporary Duty

The system shall provide a functionality that allows the recording, modification, searching and deletion of the information related to the temporary duty to other establishment. This interface should deal with the following temporary duty information: the year, the reference number, date approved, personnel ID, name, type duty (in the same organization or to another organization), beginning date, end date, and assigned department.

k. Promotions

The system shall provide a functionality that allows the recording, modification, searching and deletion of the information related to personnel promotions. This interface should provide the following information: the type of operation (promotion to a higher level, reclassification, promotion in rank or function), the decision identification, date of the decision, personnel ID, name, old position, new position (rank, function, category, the amount of the basic wage, and date beginning the new position).

l. Retirement Calculation

The system shall provide a functionality that allows the recording, modification, searching and deletion of the information related to the retirement or death of the person. This information is the year of the retirement decision or death notice, the decision identification or death notice, date, personnel ID, name, type (retirement or death), end date of work and personnel address for retirement.

m. Disciplinary Acts

The system shall provide a functionality that allows the recording, modification, searching and deletion of the information related to disciplinary acts. This interface should contain the following information: year of the disciplinary decision, the decision identification, date of the decision, personnel ID, name, type of discipline, and reason, the number of days, beginning date, and end date of the disciplinary act.

n. Resignation

The system shall provide a functionality that allows the recording, modification, searching and deletion of the information related to the resignation of the personnel. This interface should contain the following information: year of the request, reference number, date of the request, personnel ID, name, reason for resigning, and the last date of employment.

o. Education and Training

The system shall provide a functionality that allows the creating, modification, searching and deletion of the information related to the schedule, planning the training courses, calculating the cost of the training, and record the list of the trainees.

p. ***Budget Entities***

The system shall provide a functionality that allows the correction, modification, searching and deletion of the information related to the budget assigned to a project, regional direction, a local coordination and a committee of development. This interface should contain the following information: local coordination code, year, project code, action code, organization code, source code of financing, title, amount of prevision budget, and date of final budget approval.

q. ***Budget Record***

The system shall provide a functionality that allows the creation, modification, searching, deletion, and editing of a budgetary record from a proposal for pledging funds. It also allows the validation of the budget after signature.

r. ***Cost Project***

The system shall provide a functionality that allows the correction, modification, searching, and deletion of the prevision cost of the project.

s. ***Suppliers***

The system shall provide a functionality that allows the creation, modification, searching and deletion of the suppliers' record in order to have a data bank on the suppliers by branch of industry. Also, the functionality should allow the translation of the information from French into Arabic.

t. ***Suppliers Proposal***

The system shall provide a functionality that allows the introduction, modification, searching, deletion, and editing of the information provided by the suppliers' proposal as well as generation of a list comparing the presented proposal to the request for quotation after verifying the imposed criteria.

u. ***Invoices***

The system shall provide a functionality that allows the creation, modification, searching, deletion, and editing of the invoices.

v. ***Purchase Orders***

The system shall provide a functionality that allows the creation, modification, searching, deletion, and editing of the orders provided by the organization to be revised by the supplier.

w. Supplier Payment

The system shall provide a functionality that allows the creation, modification, searching, deletion, and editing of the payment for the profit of the suppliers after providing the totality of the merchandize requested by the orders.

x. Contractor Payment

The system shall provide a functionality that allows the creation, modification, searching, deletion, and editing of the records of the payments to the contractor who has signed a contract with the organization.

y. Contractor Advance

The system shall provide a functionality that allows the follow up of the advances presented to the contractor for a project.

z. Bills

The system shall provide a functionality that allows the follow up of the payment of bills.

aa. Budgetary Prevision

The system shall provide a functionality that allows the creation, modification, searching, deletion, and editing of the budgetary prevision for the current year from a proposal presented to the community for budget approval.

bb. The Journal

The system shall provide a functionality that allows the follow up of the daily accounting actions.

4. External Interface Requirements

a. User Interfaces

UI-1: The ISYDMA System screen displays shall conform to the existing manual forms in order to facilitate the exploitation of those documents

UI-2: The system shall provide a link to explain how to use that screen.

UI-3: The graphical interface shall permit complete navigation and a research information database search selection only using the keyboard, in addition to using mouse and keyboard combinations.

UI-4: The ISYDMA system shall provide a standard and uniform screen that contains a menu bar, tool bar and status bar. The menu bar contains some standard menu options such as file, edit, and selection. The status bar should contain the actual selection and the actual user. The tool bar should group the primary operations that a user is authorized to execute (add, update, record, cancel, delete, research, clear, result, first, previous, next, last and close).

b. Hardware Interfaces

No hardware interfaces have been identified.

c. Software Interfaces

SI-1: The ISYDMA system shall transmit information to the social security department in order to update their database concerning retirement.

SI-2: The ISYDMA system shall transmit information to the central bank in order to update the accounts.

SI-3: The ISYDMA system shall provide an interface that allows for the transfer of information between different systems.

d. Communications Interfaces

CI-1: The ISYDMA system shall send an email message to the external systems to inform any database status change such as software, hardware upgrades, system migration and maintenance of the ISYDMA system.

5. Other Nonfunctional Requirements

a. Performance Requirements

PE-1: The system shall accommodate 300 users during the peak usage time window of 8:00 am to 10:00 am local time, with an estimated average session duration of 50 minutes.

PE-2: All screens containing graphical information shall be fully downloadable in no more than 50 seconds over a 56 KBps modem connection at no less than 90% of the attempts.

PE-3: Responses to queries shall take no longer than 20 seconds to load onto the screen after the user submits the query.

PE-4: The system shall display confirmation messages to users within four seconds after the user submits information to the system.

b. Safety Requirements

No safety requirements have been identified.

c. Security Requirements

SE-1: All network transactions that involve financial and personnel privacy information or personally identifiable information shall be encrypted.

SE-2: Users shall be required to log in to the ISYDMA system for all operations prior to viewing any of the information.

SE-3: Users will be allowed only one login ID and in addition, passwords chosen must consist of at least eight alphanumeric distinct characters.

SE-4: The system shall permit only ISYDMA system administrators who are on the list of authorized administrators to create or edit User profiles and the user personnel privacy database.

SE-5: The system shall permit users to view only their own searched records, and not the searched information performed by other users.

d. Software Quality Attributes

Availability-1: The ISYDMA System shall be available to users 99.9% of the time between 5:00 am and midnight local time

Robustness-1: If the connection between the user and the system is broken prior to the completion of a search on the systems, the ISYDMA system shall enable the user to recover an incomplete search.

D. FUNCTIONAL DECOMPOSITION DIAGRAM

The decomposition diagram shows the top-down functional decomposition or structure of the system. It also provides the beginnings of an outline for drawing the data flow diagrams.

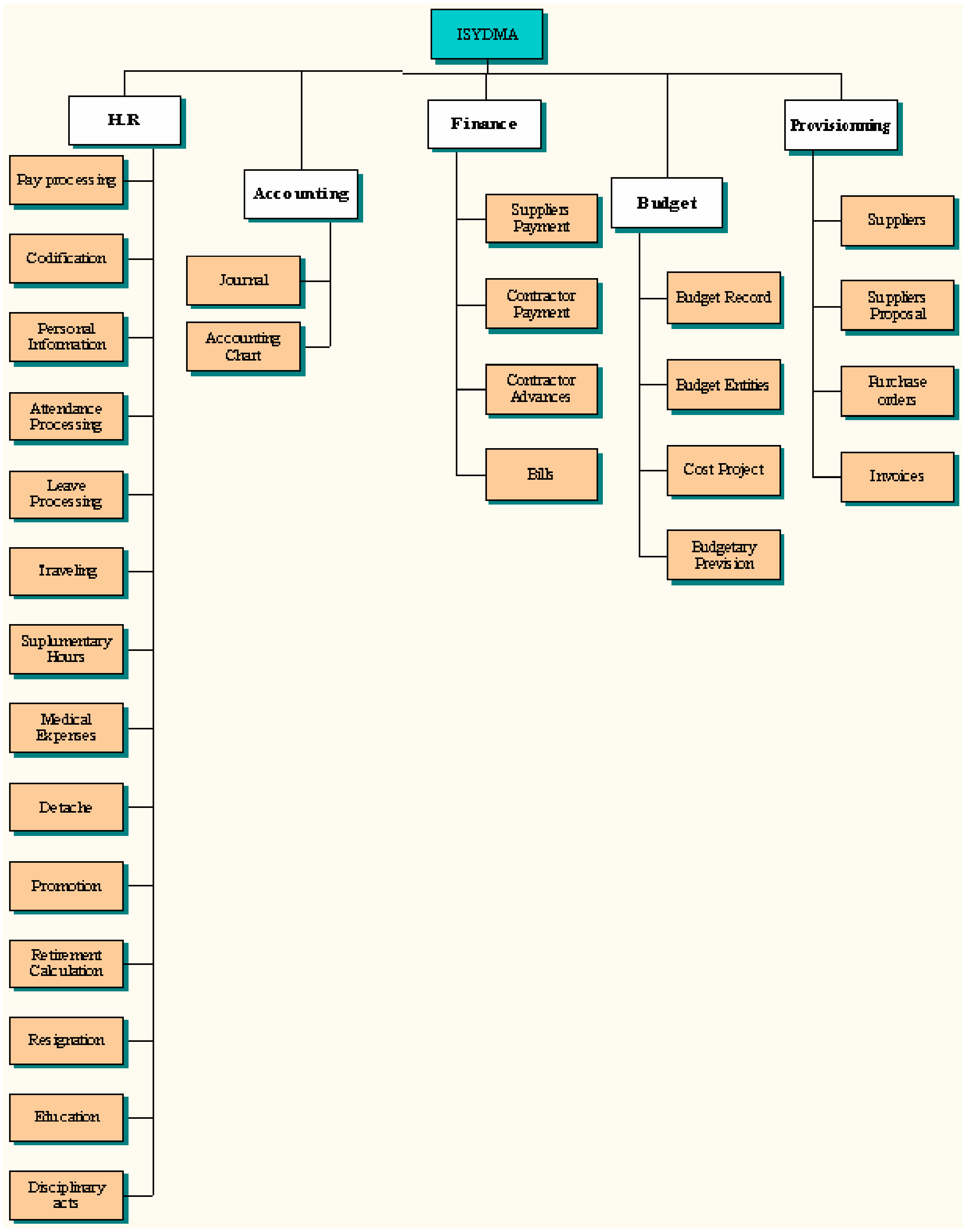


Figure 3. Functional Decomposition Diagram of the ISYDMA System

E. USE CASES

1. Use Case Diagram

Use case modeling was originally conceived by Dr. Ivar Jacobson in 1986 and gained popularity after he published his book, Object-Oriented Software Engineering, in 1992. Dr. Jacobson used use-case modeling as the framework for his objectory methodology, which he successfully used for developing an object oriented information system. Use case modeling has proved to be a valuable tool in meeting the challenges of determining what a system is required to do from a user and stakeholder perspective, and it is now widely recognized as a best practice for the defining, documenting and understanding of an information system's functional requirements [8]. The use cases establish the desired behavior of the system for verifying and validating the system architecture. Many use cases for the ISYDMA system is identified, each corresponding to different functionality. This thesis presents only use cases related to the management of the human resource sub-system as summarized in Figure 4. Detailed description of each use case is presented in Appendix C. Only the major steps that occur most of the time are included in those use cases. Also presented are some preconditions, post-conditions and exceptions that must be handled by the system.

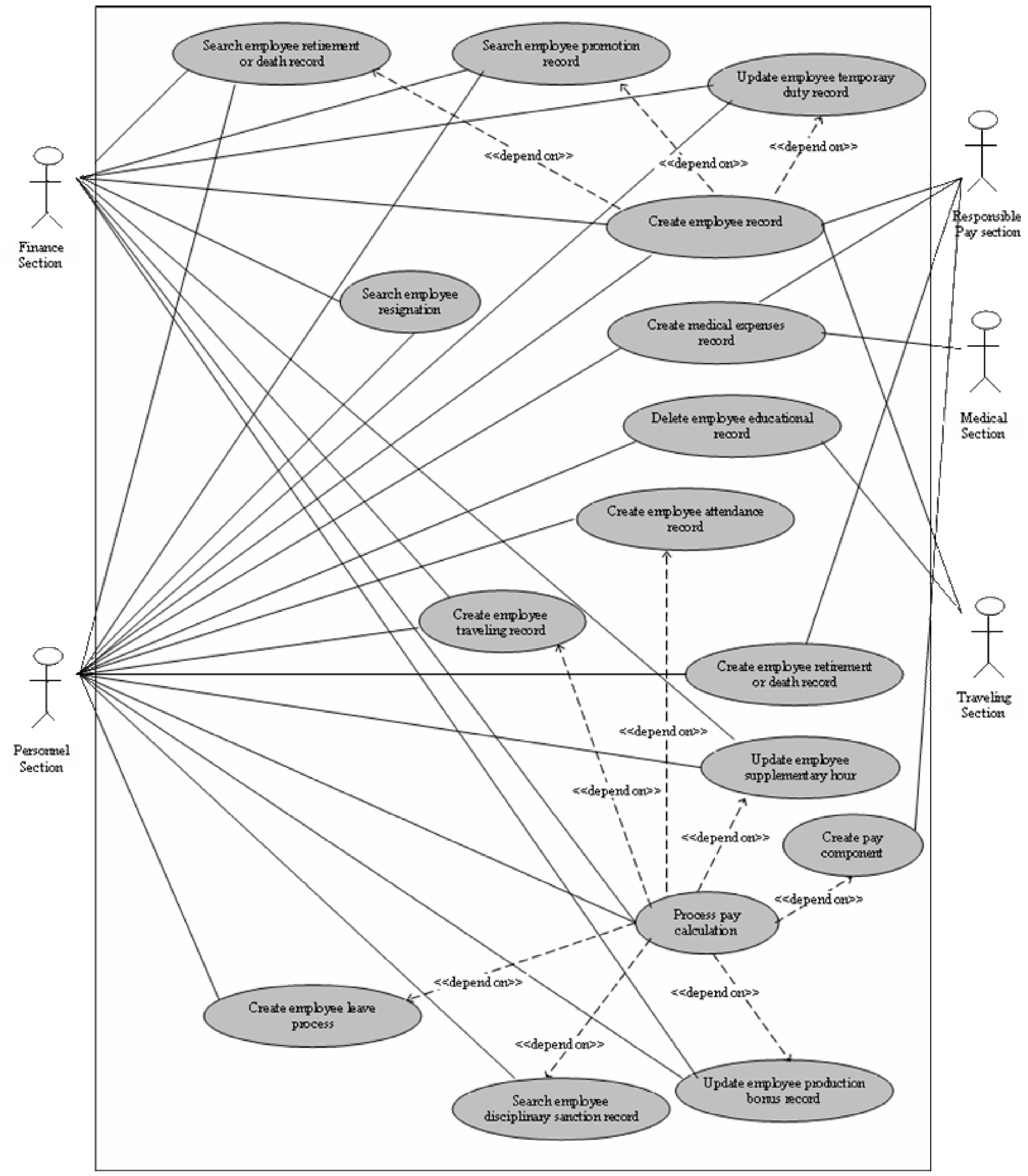

Search employee retirement or death record
Search employee promotion record
Update employee temporary duty record
<<depend on>>
<<depend on>>
<<depend on>>
Finance Section
Responsible Pay section
Create employee record
Search employee resignation
Create medical expenses record
Medical Section
Delete employee educational record
Create employee attendance record
Create employee traveling record
Traveling Section
Personnel Section
Create employee retirement or death record
<<depend on>>
Update employee supplementary hour
<<depend on>>
<<depend on>>
Create pay component
<<depend on>>
Process pay calculation
<<depend on>>
<<depend on>>
Create employee leave process
<<depend on>>
Search employee disciplinary sanction record
Update employee production bonus record

Figure 4. Human Resource Sub-System Use Case Hierarchy Diagram

2. Use Case Glossary

Use-case Name	Use-case Description	Participating Actors and Roles
Create pay component	This use case describes the events of pay section for establishing a new pay component that will serve as a key part for calculation of employee payment.	- Pay section (primary business)
Create employee Record	This use case describes the events of personnel and training section for creating an employee record which is similar for the three types of employee (Statutory employee, Workman employee, agriculture employee). The Employee record is composed of several parts. The first part concerns the identification information, the second the administrative information, and the last the family information.	- Personnel section (primary business) - Training section
Create employee attendance record	This use case describes the events of a personnel section creating a record related to the absence of an employee. Absence can be classified as justified or not justified.	- Personnel section (primary business)
Search employee disciplinary sanction record	This use case describes the events of a personnel section for searching a record related to disciplinary acts.	- Personnel section (primary business)
Update employee temporary duty record	This use case describes the events of a personnel section for creating a record related to the temporary duty record. The temporary duty record can be in another organization or in another department of the same organization.	- Personnel section (primary business)
Update employee production bonus record	This use case describes the events of a personnel section for updating the information related to production bonuses. The production bonuses are calculated once every three months.	- Personnel section (primary business) - Finance section
Search employee resignation record	This use case describes the events of personnel section for searching a record related to an employee's resignation.	- Personnel section (primary business) - Finance section
Delete employee educational record	This use case describes the events of a personnel section for deleting a record containing the different training classes that an employee had attended. The training section can be organized into an internal department or external organization.	- Personnel section (primary business) - Training section
Create employee traveling record	This use case describes the events of a personnel section for creating a record containing the necessary information for travel payments done by an employee during the actual month.	- Personnel section (primary business) - Finance section

Use-case Name	Use-case Description	Participating Actors and Roles
Search employee promotion record	This use case describes the events of a personnel section for searching a record related to the promotion of an employee. An employee's promotion must satisfy certain conditions. There are two types of promotion: rank promotion and category promotion. Each one affects the pay components.	- Personnel section (primary business) - Finance section
Search employee retirement or death record	This use case describes the events of a personnel section for searching a record for retirement and a record upon the death of an employee. Those records affect the pay component.	- Personnel section (primary business) - Finance section
Create medical expenses record	This use case describes the events of a personnel section for creating a record for medical expenses.	- Personnel section (primary business) - Medical section
Create employee leave process	This use case describes the events of a personnel section for creating a record for an employee's leave time. There are many types of leave: annual leave, advance leave, family leave, sick leave, leave without pay, maternity leave.	- Personnel section (primary business)
Update employee supplementary hours record	This use case describes the events of a personnel section for updating a record for the supplementary hour record. Every employee has a maximum number of supplementary hours per month. The amount of money allowed for one supplementary hour depends on the rank and the type of employee.	- Personnel section (primary business) - Finance section
Process pay calculation	This use case describes the events of a personnel section for creating a record for grouping all the components and necessary information to calculate the salary of an employee. The pay calculation is executed every month, and should be started by the second week of the actual month.	- Personnel section (primary business) - Finance section

Table 5. Use Case Glossary

F. DATA MODELING AND ANALYSIS

Systems models play an important role in systems development. Data modeling is a technique for defining business requirements for a database. Data modeling is often called database modeling because a data model is eventually implemented as a database.

The data model is a conceptual representation of the data structures required by a database. The data structures include the data objects, the associations between data objects, and the rules, which govern operations on the objects. As the name implies, the data model focuses on what data is required and how it should be organized rather than what operations will be performed on the data. A data model is independent of hardware or software constraints. Rather than trying to represent the data, as a database would see it, the data model focuses on representing the data as the user sees it in the "real world". It serves as a bridge between the concepts that compose real-world events and processes and the physical representation of those concepts in a database. There are two major methodologies used to create a data model: the Entity-Relationship (ER) approach and the Object Model. This thesis uses the Entity-Relationship approach. [8]

The data model obtains its inputs from the planning and analysis stage. The modeler, along with analysts, collects information about the requirements of the database by reviewing existing documentation and interviewing end-users.

1. Conceptual Data Model Representation (CDM)

A CDM represents the overall logical structure of a database, which is independent of any software or data storage structure. A conceptual model often contains data objects not yet implemented in the physical database. It gives a formal representation of the data needed to run an enterprise or a business activity.

The CDM makes it possible:

- Represent the organization of data in a graphical format to create Entity Relationship Diagrams (ERD)
- Verify the validity of data design
- Generate a Physical Data Model (PDM), which specifies the physical implementation of the database
- Generate an Object-Oriented Model (OOM), which specifies an object representation of the CDM using the UML standard
- Generate a Conceptual Data Model (CDM), to create another model version in order to represent different design stages

A CDM represents the interaction of the following objects:

Object	Tool	Description
Domain	--	Set of values for which a data item is valid
Data item	--	Elementary piece of information
Entity	⊟	Person, place, thing, or concept that has characteristics of interest to the enterprise and information to be stored
Entity attribute	--	Elementary piece of information attached to an entity
Identifier	--	Entity attribute, or a combination of entity attributes, whose values uniquely identify each occurrence of the entity
Relationship	⊶	Named connection or relation between entities (Entity Relationship (ER) modeling methodology)
Inheritance	⊥	Special relationship that defines an entity as a special case of a more general entity
Association	⊖	Named connection or association between entities (MERISE modeling methodology)
Association link	⊶	Link that connects an association to an entity and the definition of the cardinality an entity has relative to another

Table 6. Description of the Objects Used for the CDM (After: [16])

The Conceptual Data Model for the human resource sub-system is divided into three parts. The first part concerns Employee Management (Figure 5). The second part concerns the Follow up of Education (Figure 6). The third part concerns the Order of payoff of personnel (Figure 7). The data model has two outputs. The first is an entity-relationship diagram, which represents the data structures in a pictorial form (Figures 5, 6, 7). The second component is a data dictionary that provides the detail required by the database developer to construct the physical database.

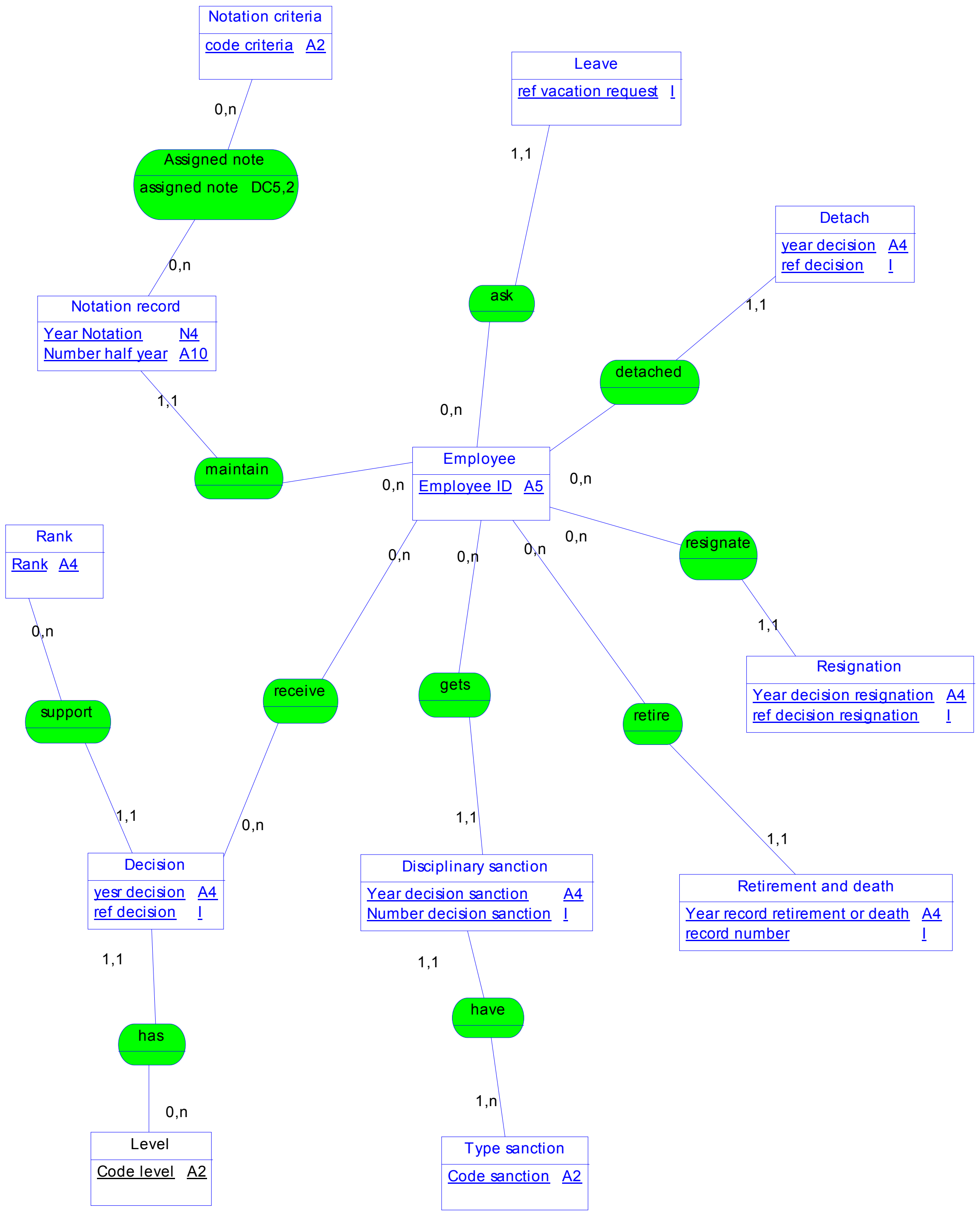


Figure 5. Conceptual Data Model Representation (Employee Management)

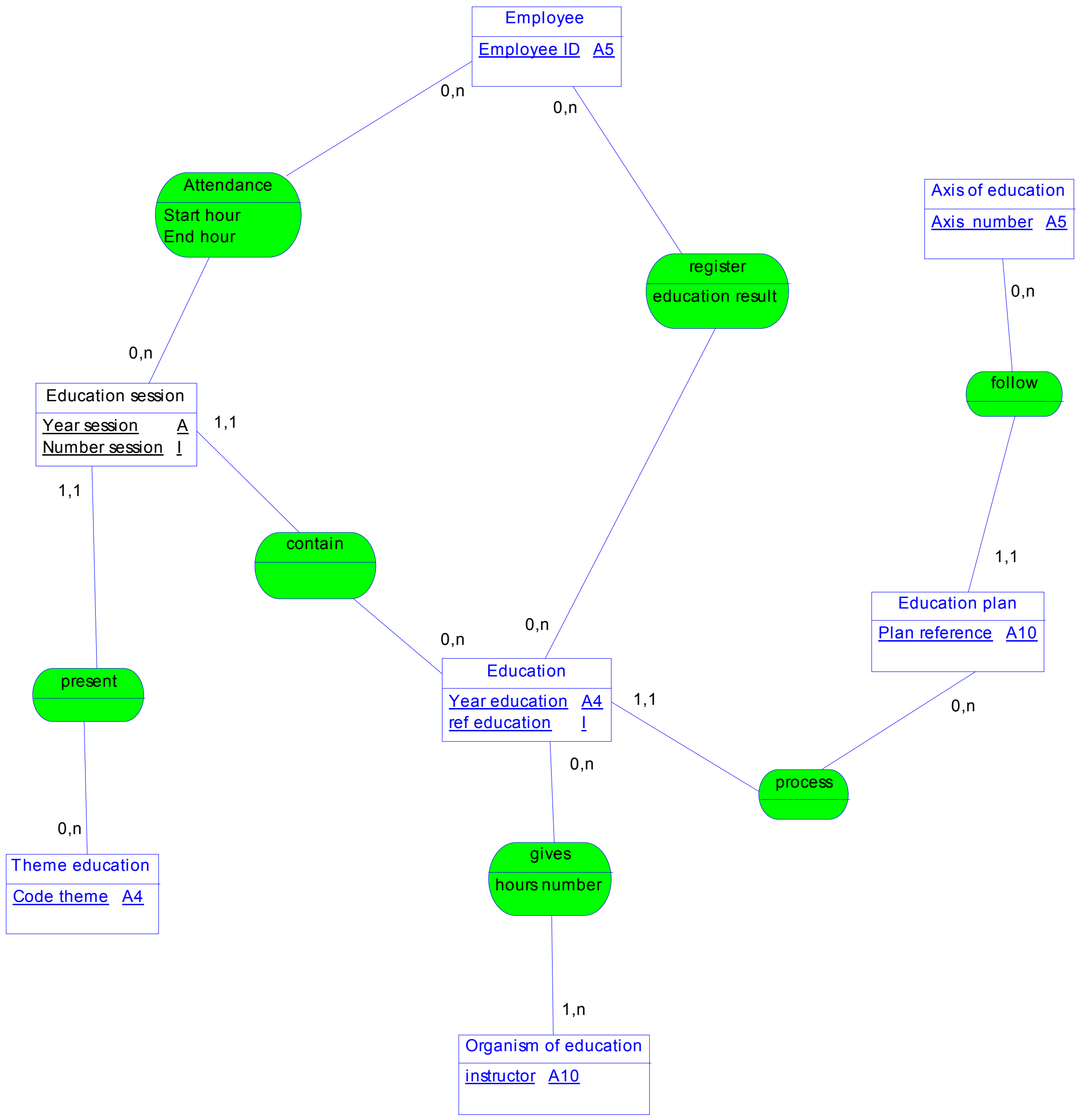


Figure 6. Conceptual Data Model Representation for Education Process

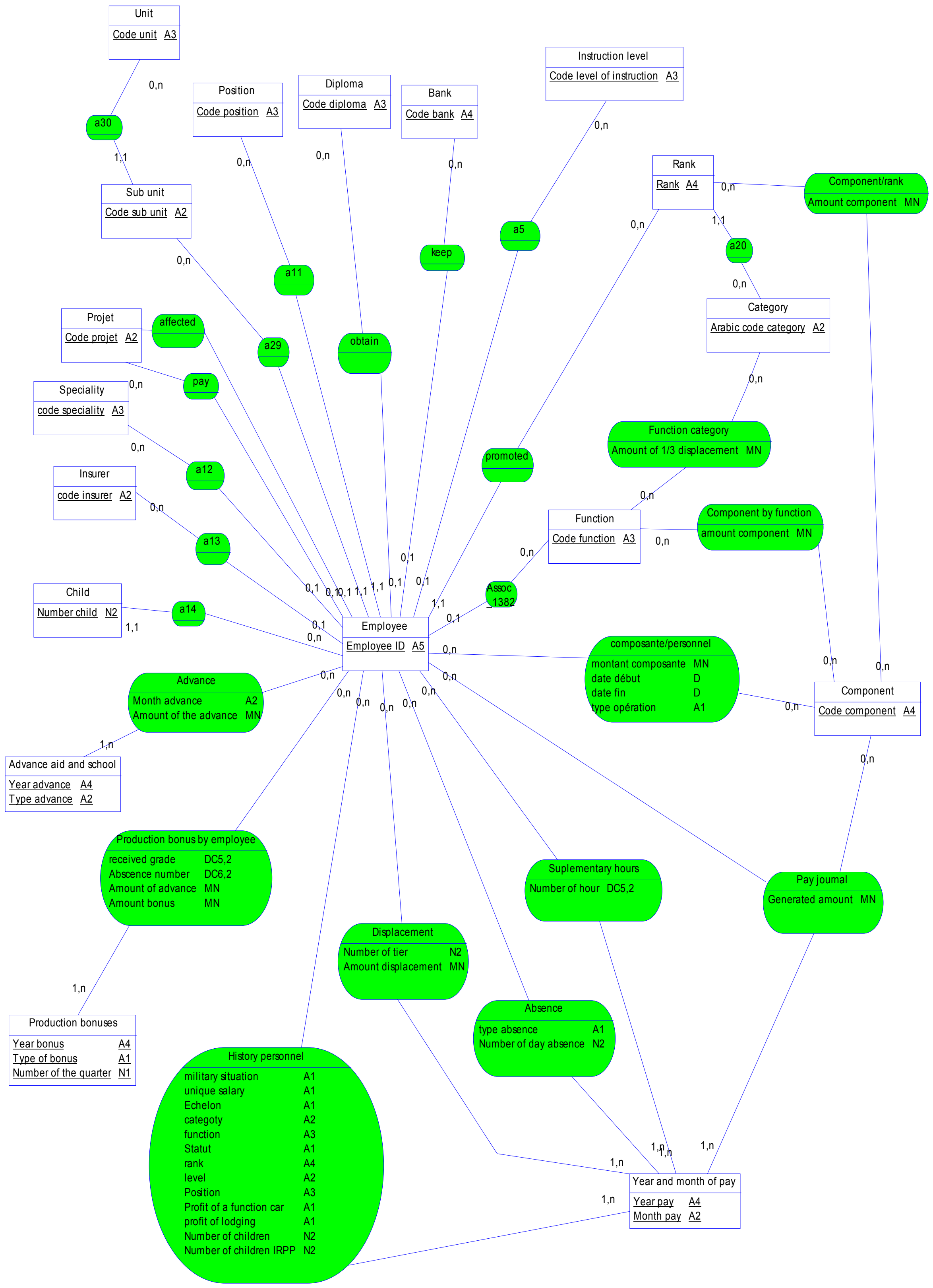


Figure 7. Conceptual Data Model Representation for Order Payoff of Personnel

2. Data Dictionary

The data dictionary encompasses the whole organization, a part of the organization or a database. In its simplest form, the data dictionary is only a collection of data element definitions, according to the descriptions that follow. A more advanced data dictionary contains the database schema with reference keys, while a more advanced data dictionary contains the entity-relationship model of the data elements or objects. The term “data element” used below is the same concept as a “data object” or “object” in some database texts.

- **Data element name:** commonly agreed, unique data element name from the application domain. This is the actual name of this data element.
- **Short description:** description of the element in the application domain.
- **Field name(s):** Field names are the names used for this element in computer programs and database schemas. These are the technical names, often limited by the programming languages and systems.
- **Code format:** Data type (characters, numeric, etc.), size and, if needed, special representation. Common programming language notation, input masks, etc. can be used.
- **Default value:** Data element may have a default value. The default value may be a variable, such as a current date and time of day.

Appendix A shows the list of data collected for the human resource sub system. Appendix B shows the database schema.

III. DESIGN PHASE

A. APPLICATION ARCHITECTURE

The purpose of the first design task is to specify the application architecture that defines the technologies to be used by one, many, or all information systems in terms of their data, processes, interfaces and network components. Thus, designing the application architecture involves considering network technologies and making decisions on the systems' data. Processes and interfaces are distributed among the business locations.

This task is accomplished by analyzing the data model and process models initially created during requirements analysis. The following figure shows the architecture used for the ISYDMA system (client/server system) in which the data and data manipulation layers are placed on servers and others layers are placed on clients, also called two-tier client/server computing.

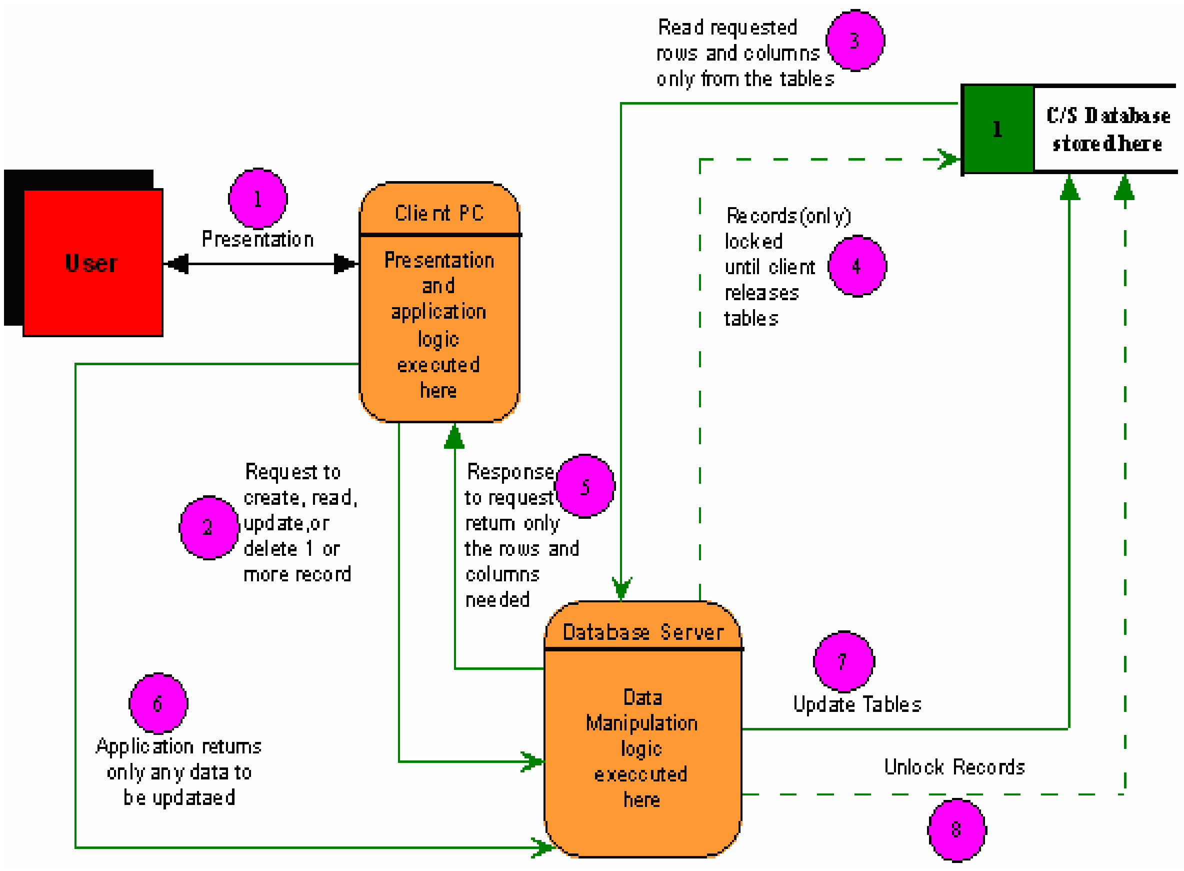

Read requested rows and columns only from the tables
3
1
C/S Database stored here
User
1
Presentation
Client PC
Presentation and application logic executed here
Records (only) locked until client releases tables
4
Request to create, read, update, or delete 1 or more record
2
Response to request return only the rows and columns needed
5
Database Server
Data Manipulation logic exeecuted here
6
Application returns only any data to be updataed
7
Update Tables
Unlock Records
8

Figure 8. Client / Server System: Distributed Data (Two Tiers) (From: [8])

B. OBJECT MODEL REPRESENTATION

In object oriented analysis, the emphasis is on identifying the objects that represent actual data within the business domain. These objects are called entity objects. The transformation from the Conceptual Data Model to the Object Model is an automated process realized by PowerAMC used for representing the Conceptual Data Model (Figures 5, 6, 7).

When generating an Object Oriented Model (OOM) from a CDM, PowerAMC converts CDM objects into specified object language objects as follows:

CDM Objects	Generated object in an OOM
Entity	Class
Attribute	Attribute
Association	Relationship or association
Binary association with attributes	Association class
Inheritance	Generalization

The rules applied for this transformation are as follows.

1. Independent One-to-Many Relationships

In independent one-to-many relationships, the primary identifier of the entity on the one side of the relationship becomes a:

- Primary key in the entity on the one side of the relationship
- Foreign key in the entity on the many side of the relationship

2. Dependent One-to-Many Relationships

In dependent relationships, the primary identifier of the nondependent entity becomes a primary/foreign key in the dependent entity.

3. Independent Many-to-Many Relationships

In independent many-to-many relationships, the primary identifiers of both entities migrate to a join entity as primary/foreign keys.

4. Independent One-to-One Relationships

In independent one-to-one relationships, the primary identifier of one entity migrates to the other generated entity as a foreign key.

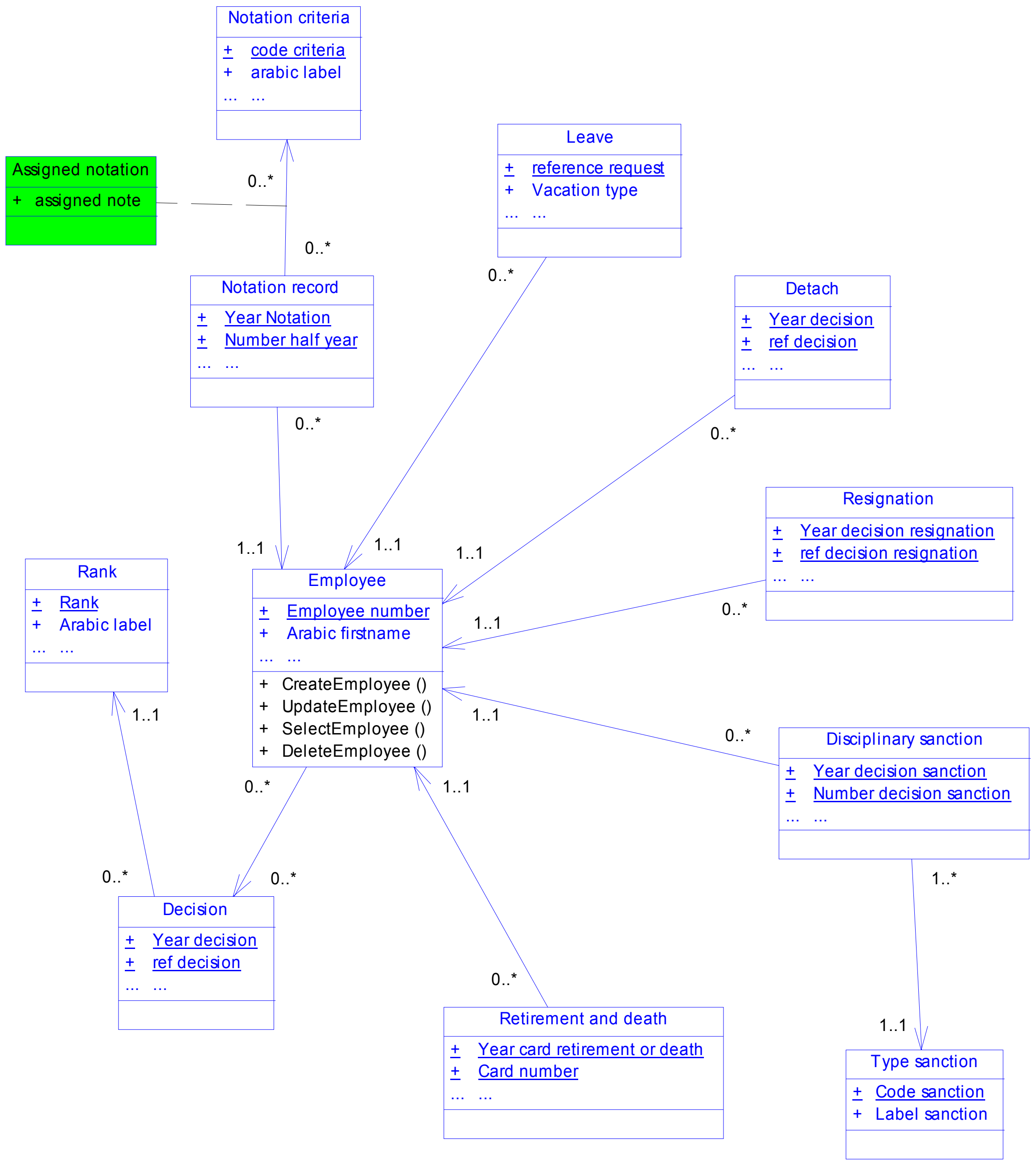


Figure 9.   Object Model Representation (Employee Management)

<u>Note</u>: Underlined attributes denote the keys of the data items.

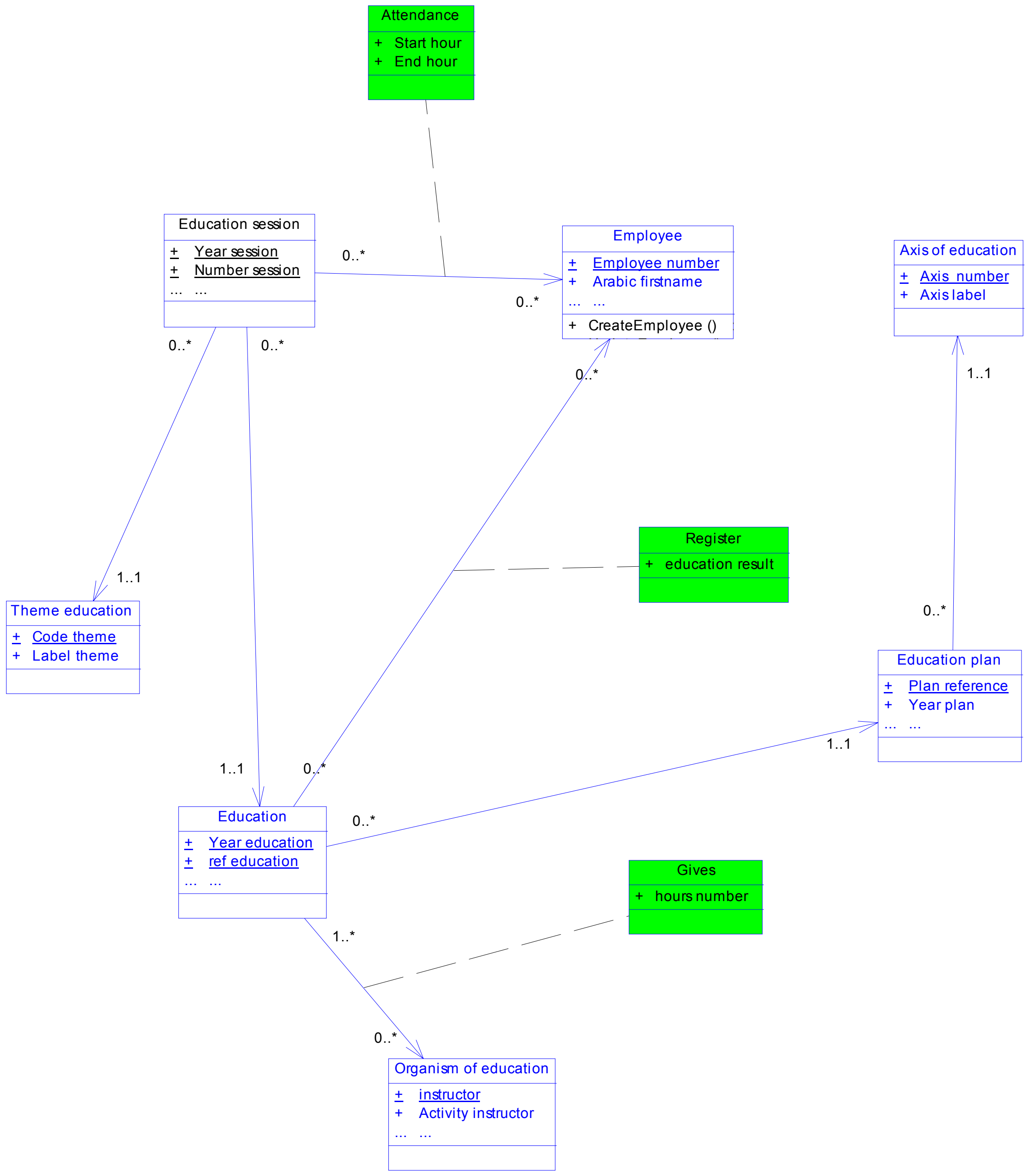


Figure 10. Object Model Representation for Education Process

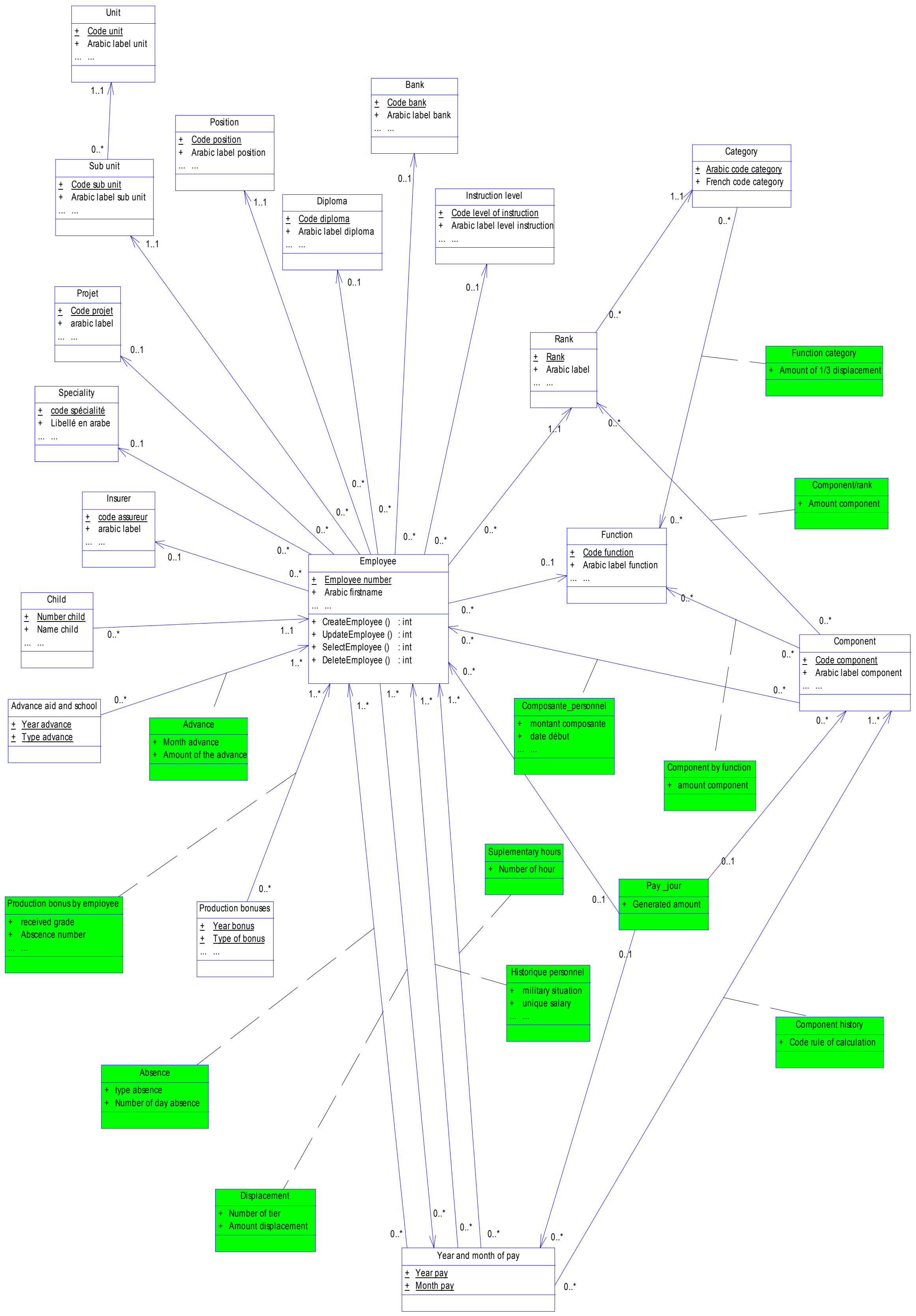


Figure 11. Object Model Representation for Order Payoff of Personnel

C. SEQUENCE DIAGRAM

The sequence diagram depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to execute the functionality of the system. It can be used to derive the testable user interface requirement.

1. Sequence Diagram: *Create function*

The sequence diagram in Figure 12 shows the process for creating data records. An instance of the *User* class invokes a method on the *Identification module process* called *ChooseModule()*. The return value from the actual object is then used to instantiate an instance of *Module* object. A method called *AccessModule()* is then invoked on the new instance. This instantiates an instance of *Display Process* and invokes a method called *DisplayForm()*. The User Class invokes a new method on the *Create Process* that returns a message for authorizing the add record. After receiving the authorization, the User invokes a method in the *Control Process* called *EnterData()*. The control is actually requested from this object. If the input data are valid, then the record is created and the *Validation Process* invokes a method on the *Add Process* called *AddRecord().* Finally, a message confirming the Add record is sent to the user.

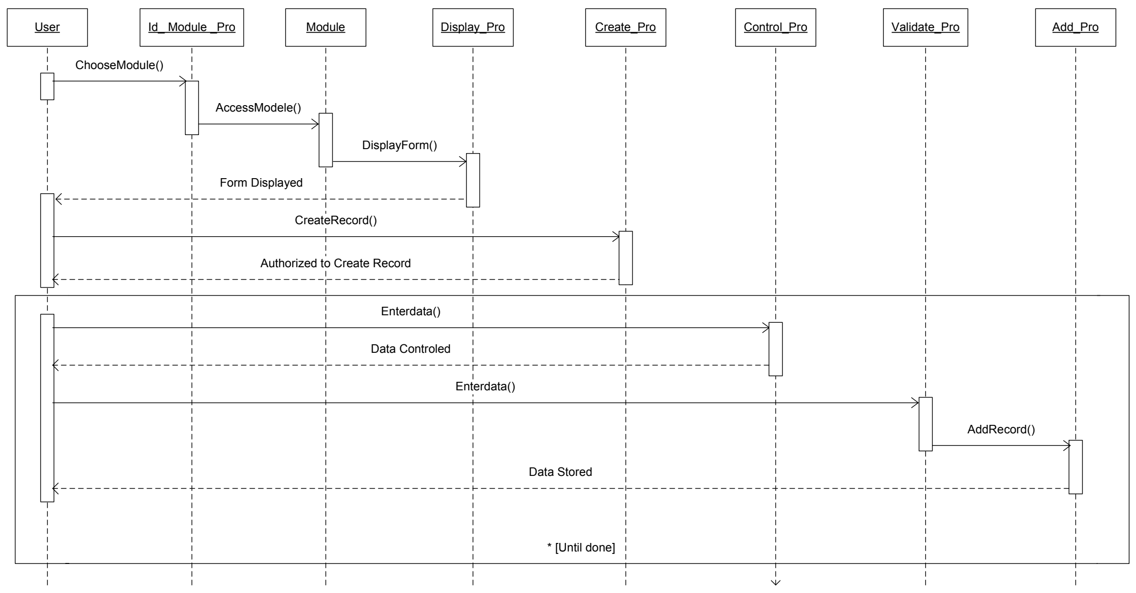


Figure 12. Sequence Diagram for the Create Function of the System

2. Sequence Diagram: *Update function*

The sequence diagram in Figure 13 shows the process for updating data records. An instance of the *User* class invokes a method on the *Identification module process* called *ChooseModule()*. The return value from the actual object is then used to instantiate an instance of *Module* object. A method called *AccessModule()* is then invoked on the new instance. This instantiates an instance of *Display Process* and invokes a method called *DisplayForm()*. The User Class invokes a new method on the *Update Process* that returns a message for authorizing the Update record. After receiving the authorization, the User invokes a method in the *Control Process* called *EnterData()*. The control process invokes a method called *SearchData()* that allows searching the entered information. If the input data are found, then a message Data Found is sent to the User. The User class invokes a new method on the *Update process* called *UpdateData()*. The control is actually requested from this object. If the input data are valid then the record is updated and the *Validation Process* invokes a method on the *Record Process* called *UpdateRecord()*. Finally, a message confirming the Update record is sent to the user.

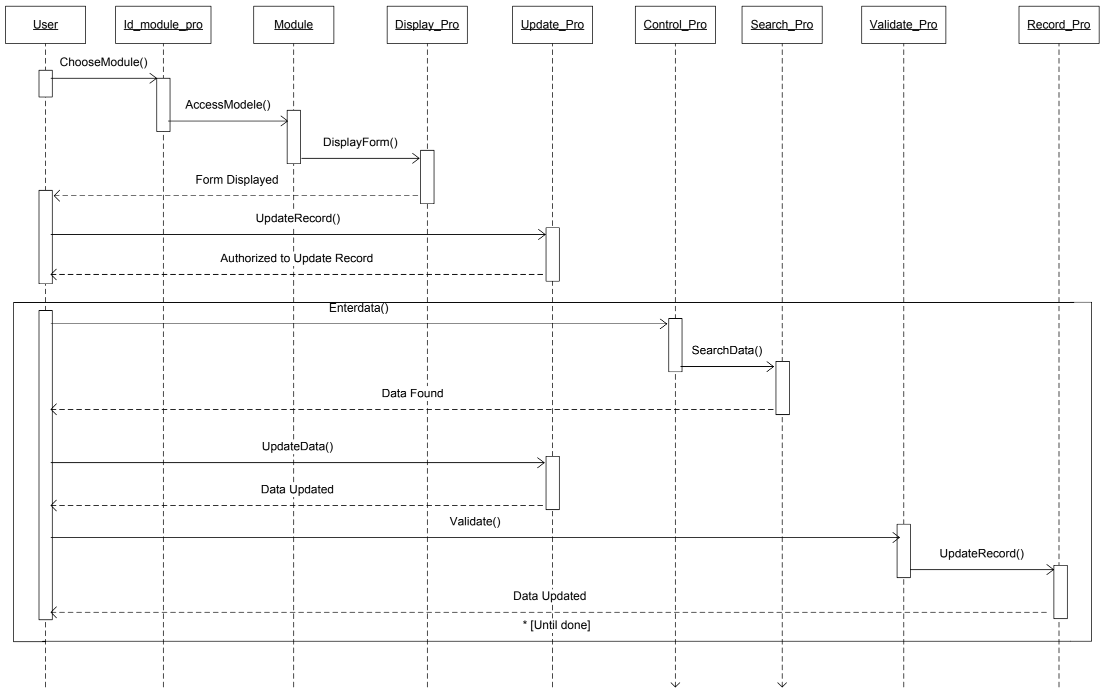


Figure 13. Sequence Diagram for the Update Function of the System

3. **Sequence Diagram:** ***Search function***

The sequence diagram in Figure 14 shows the process for searching data records. An instance of the *User* class invokes a method on the *Identification module process* called *ChooseModule()*. The return value from the actual object is then used to instantiate an instance of *Module* object. A method called *AccessModule()* is then invoked on the new instance. This instantiates an instance of *Display Process* and invokes a method called *DisplayForm()*. The User Class invokes a new method on the *Search Process* that returns a message for authorizing the Search record. After receiving the authorization the User invokes a method in the *Control Process* called *EnterData().* The *control process* invokes a method called *SearchData()* that allows searching the entered information. If the input data are found, then the *search data process* invokes a method on the *display data process* called *Display().* Finally, a message confirming the Update record is sent to the user.

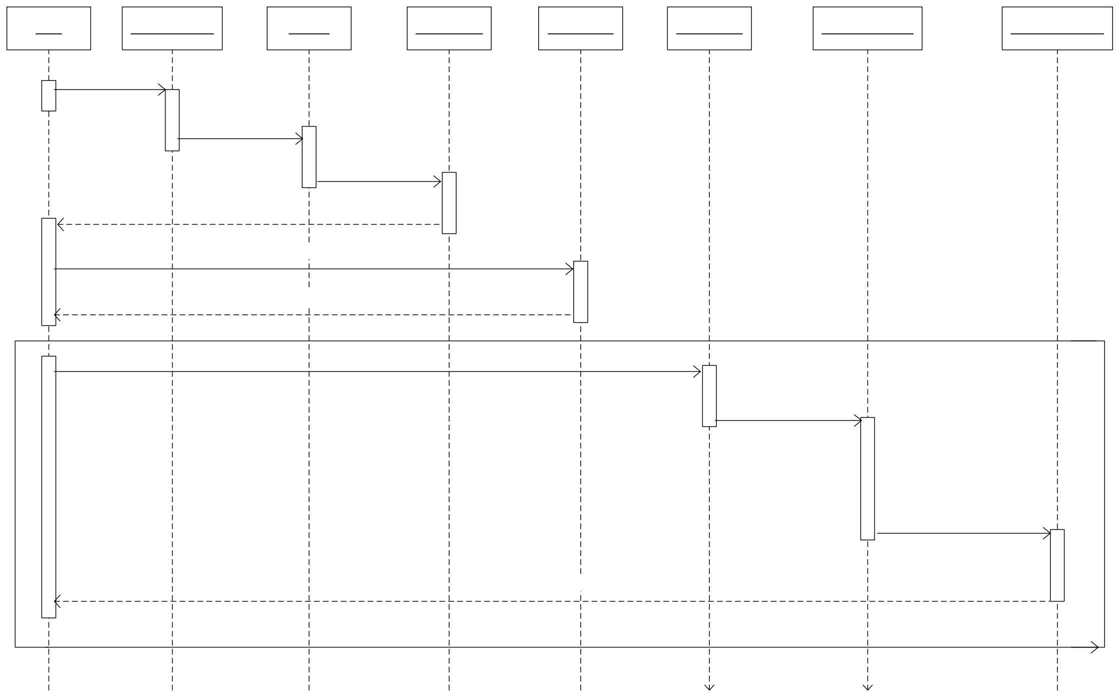


Figure 14. Sequence Diagram for the Search Function of the System

4. Sequence Diagram: *Delete Function*

The sequence diagram in Figure 15 shows the process for delete data records. An instance of the *User* class invokes a method on the *Identification module process* called *ChooseModule()*. The return value from the actual object is then used to instantiate an instance of *Module* object. A method called *AccessModule()* is then invoked on the new instance. This instantiates an instance of *Display Process* and invokes a method called *DisplayForm()*. The User Class invokes a new method on the *Delete Process* that returns a message for authorizing the Delete record. After receiving the authorization the User invokes a method in the *Control Process* called *EnterData()*. The control process invokes a method called *SearchData()* that allows searching the entered information. If the input data are found then a message Data Found is sent to the User. The User invokes a method on the *Delete process* called *DeleteData()*. A Message Data temporary deleted is sent to the user. The *Validation Process* invokes a method on the *Record Process* called *DeleteRecord()*. Finally, a message confirming the Delete record is sent to the user.

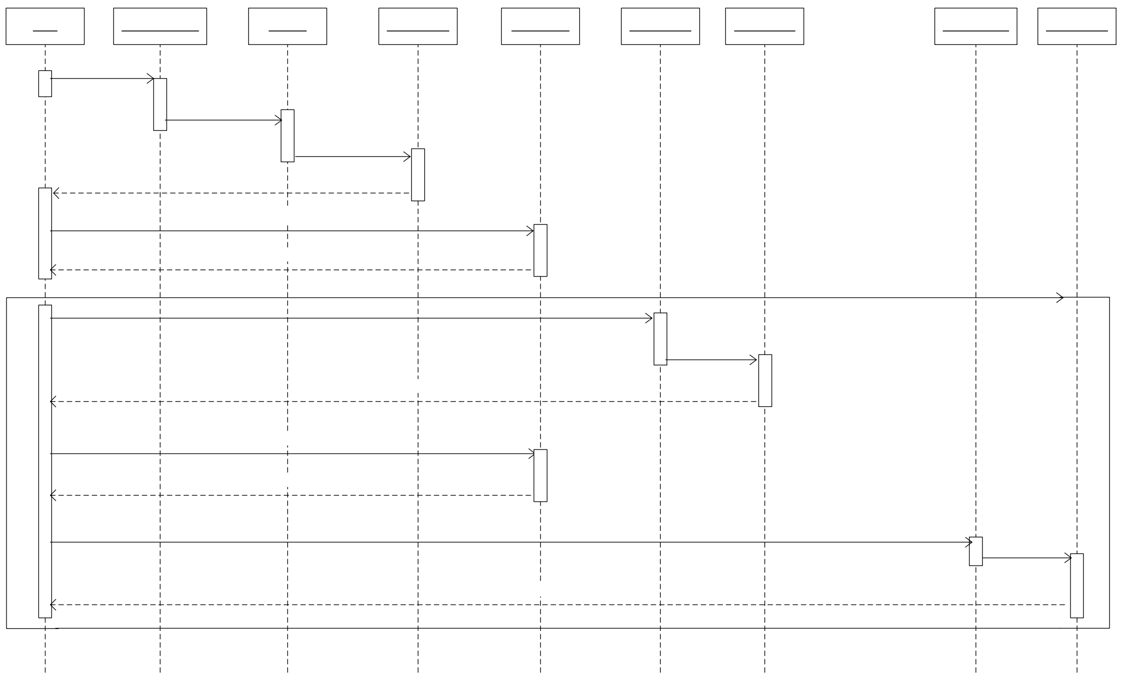


Figure 15. Sequence Diagram for the Delete Function of the System

THIS PAGE INTENTIONALLY LEFT BLANK

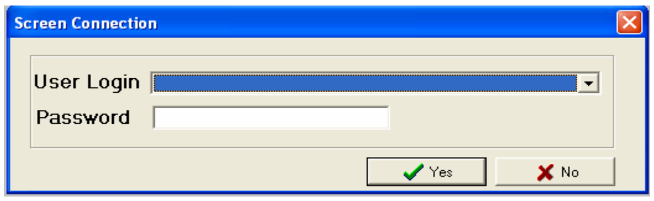
IV. PROTOTYPE

A. COMMON FUNCTIONALITIES

1. How to Connect

When the user starts the application, it is necessary to connect the system to the authorized modules. The process begins with File/Connect in order to access the functionalities of the system.

The following screen appears. The user types in the user ID and password.



2. Feature and Common Functionalities

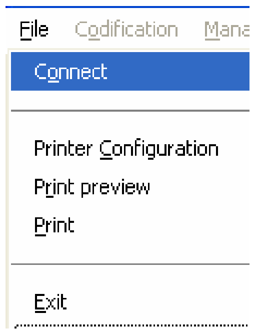
The model presents a standard screen for the entire system composed of the following features:

- Menu bar
- Status bar
- Tools bar
- One or many child window

a. Menu Bar

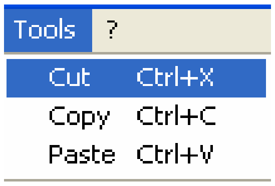
The menu bar has the standard menu options such as File and Tools. Many other functions are available depending on the selected module.

(1) File Menu.



Connect	Identify user
Disconnect	Leave the active connection for the database
Printer configuration	Allow the standard window for printer configuration to open
Exit	Allow existing the module

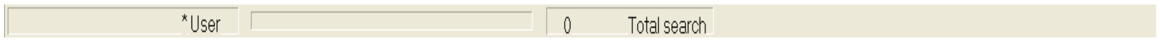
(2) Tools Menu.



- *Cut <Ctrl>+X*	*It is the standard Cut functionality*
- *Copy <Ctrl>+C*	*It is the standard Copy functionality*
- *Paste <Ctrl>+V*	*It is the standard Paste functionality*

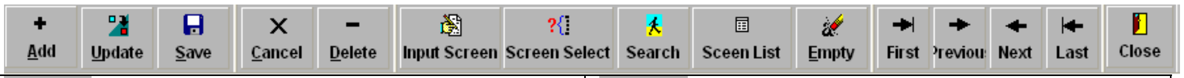
b. Status Bar

The Main window has a state bar on the bottom that indicates the actual selection and the connected user



c. Tools Bar

On the top and under the menu bar, the tools bar groups a set of symbols or icons representing the different operations that a user can utilize by simply clicking with the mouse



Search Allows the execution of search with specified criteria	Empty Allows deletion of search criteria
Screen Select Allows introduction of a criteria selection	Input Screen Allows input of information
Last Go to the last record in the active table	Next Go to the Next record in the active table

Previou Go to the previous record in the active table	First Go to the First record in the active table
Add Allows addition of a record to the active table	Delete Allows deletion of a record from the active table
Update Allows updating a record in the active table	Save Allows saving the active record
Cancel Allows canceling input data	Sceen List Allows showing the selected record in list format

d. Data Manipulation

(1) Select Data. The select action is the first operation done before searching, updating, deleting and editing any information. Before manipulating or processing data, it is necessary to select a set of data that the process will conduct. In order to select data, it is necessary to display the search screen and present the criteria of search. The symbols that can be introduced for a search of criteria can be the following. (=,>,<,>=,<=,<>,!=,:,|,&,*,-) Those symbols are the standard used by Query By Example (QBE).

(2) Add Data. In order to create a new record the user must:

- Open the window corresponding to the specified entity.
- Click on the <Add> icon and start inputting data.
- In order to save data, the user must click on the <Save> icon.
- In order to cancel the inputs, the user must click on the <Cancel> icon.

(3) Update Data. In order to update information:

- Open the window corresponding to the specified entity.
- Click on the <Search> icon and search for the occurrence in the database by way of a search screen presented in the select data section.
- Click on the <Update> icon and do the necessary modification on the data.
- In order to save the data, the user must click on the <Save> icon.
- In order to cancel the modification, the user must click on the <Cancel> icon.

(4) Delete Data. In order to delete information:

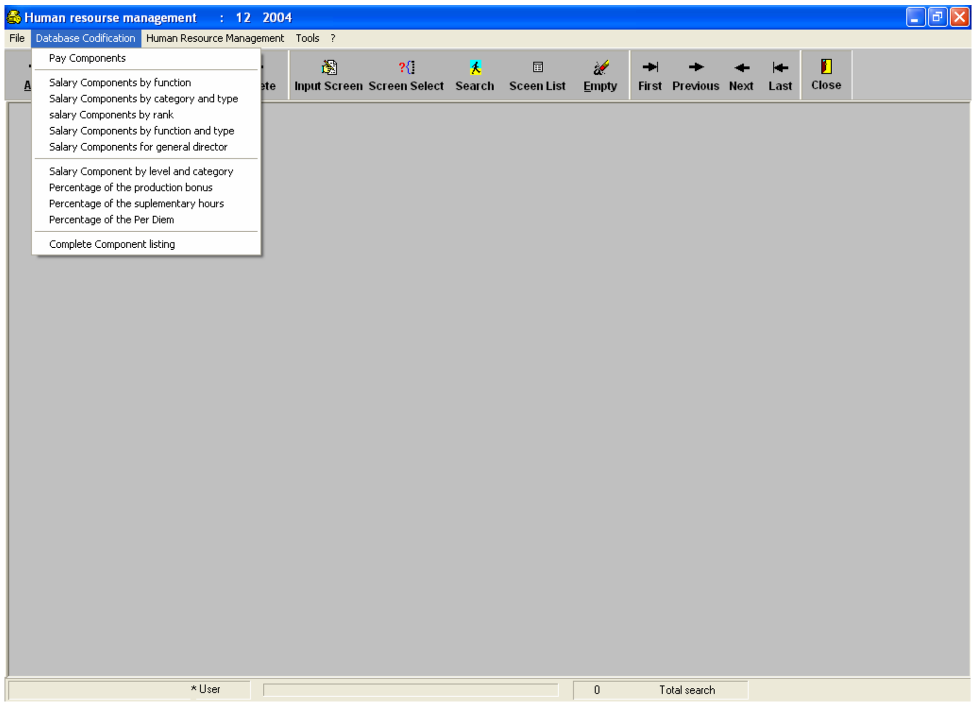
- Open the window corresponding to the specified entity.

- Click on the <Search> icon and search for the occurrence in the database by way of a search screen presented in the select data section.
- Click on the <delete> icon.

B. PRESENTATION OF THE MAIN MENU

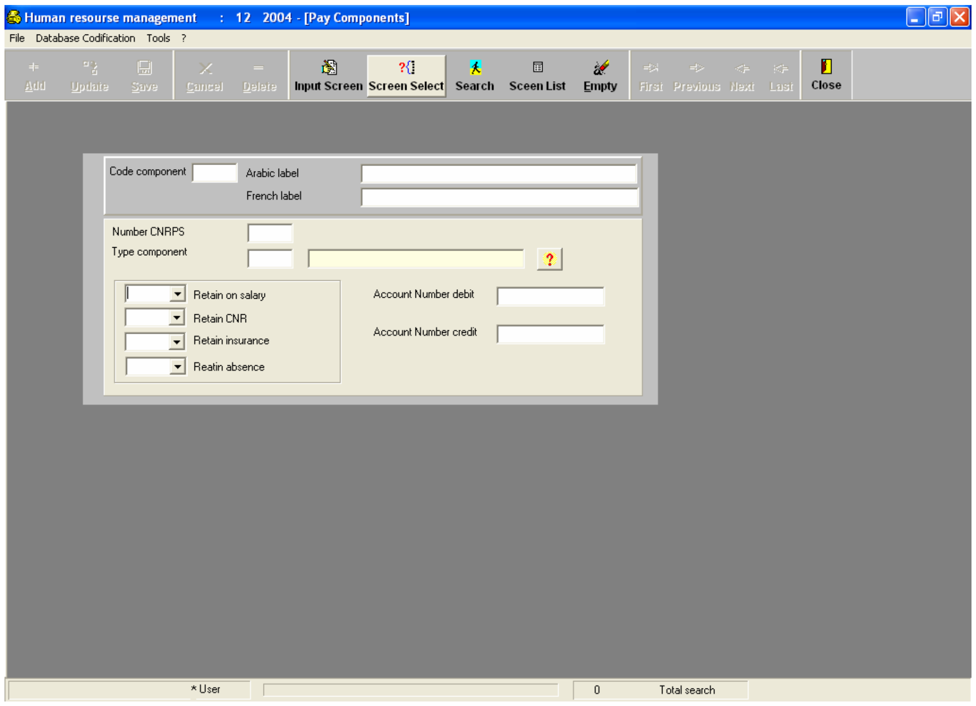
The developed software for the Human Resource subsystem ensures the following functionalities:

1. Database Codification


Human resourse management : 12 2004
File
Database Codification
Human Resource Management
Tools
?
Pay Components
Salary Components by function
Salary Components by category and type
salary Components by rank
Salary Components by function and type
Salary Components for general director
Salary Component by level and category
Percentage of the production bonus
Percentage of the suplementary hours
Percentage of the Per Diem
Complete Component listing
Input Screen
Screen Select
Search
Sceen List
Empty
First
Previous
Next
Last
Close
* User
0
Total search

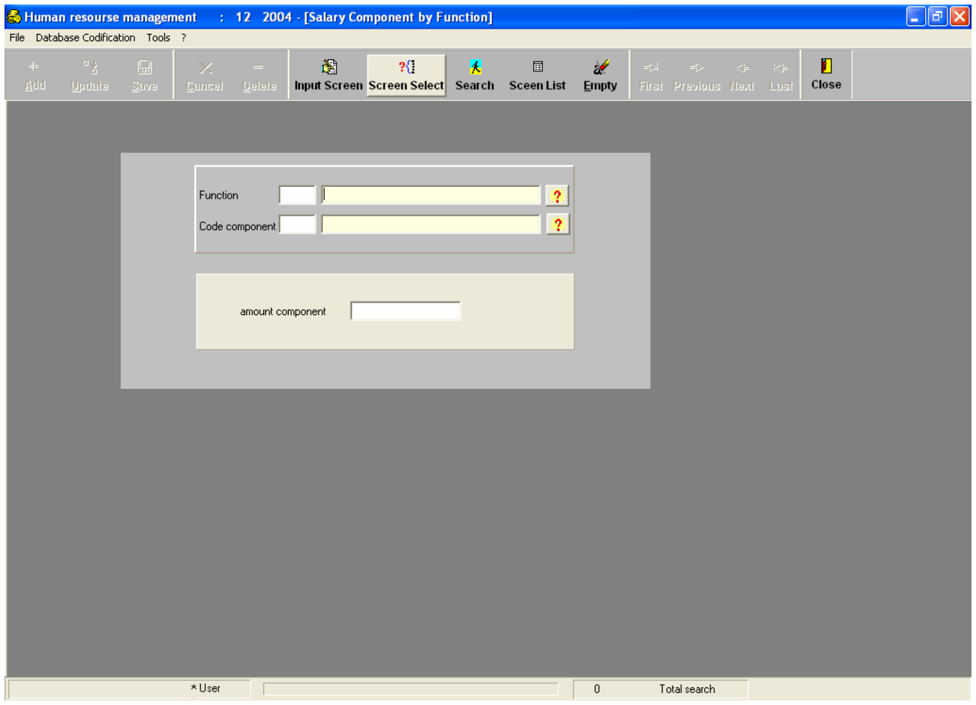
a. ***Pay Component***

The following screen allows the searching of the pay component.


Human resourse management : 12 2004 - [Pay Components]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Code component
Arabic label
French label
Number CNRPS
Type component
Retain on salary
Retain CNR
Retain insurance
Reatin absence
Account Number debit
Account Number credit
* User
0
Total search

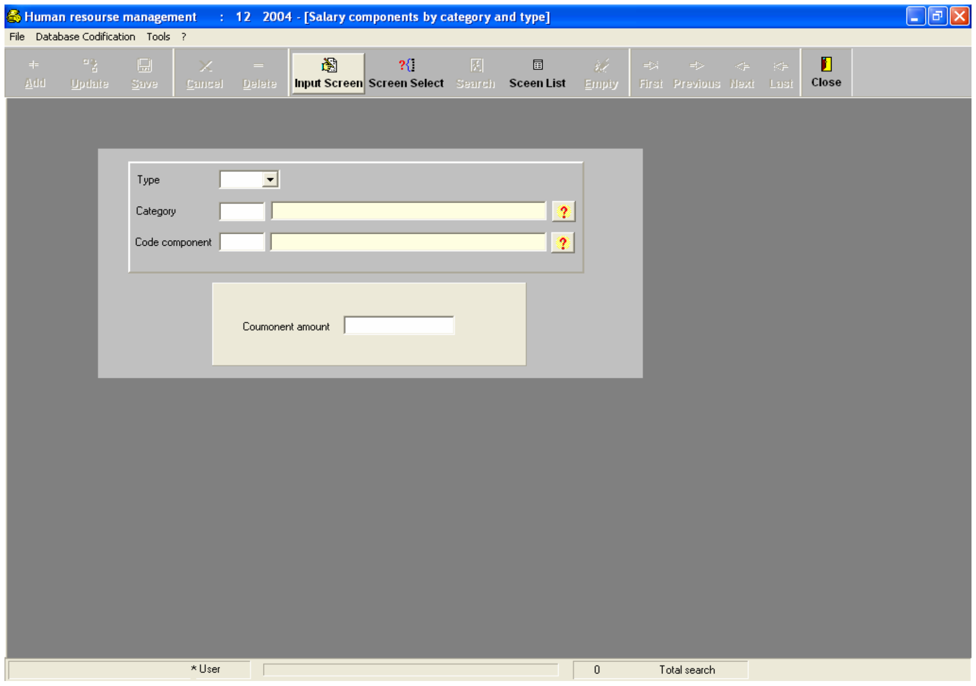
b. ***Salary Component by Function***

The following screen allows the searching of the salary component by function:


Human resourse management : 12 2004 - [Salary Component by Function]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Function
Code component
amount component
* User
0
Total search

c. *Salary Component by Category and Type*

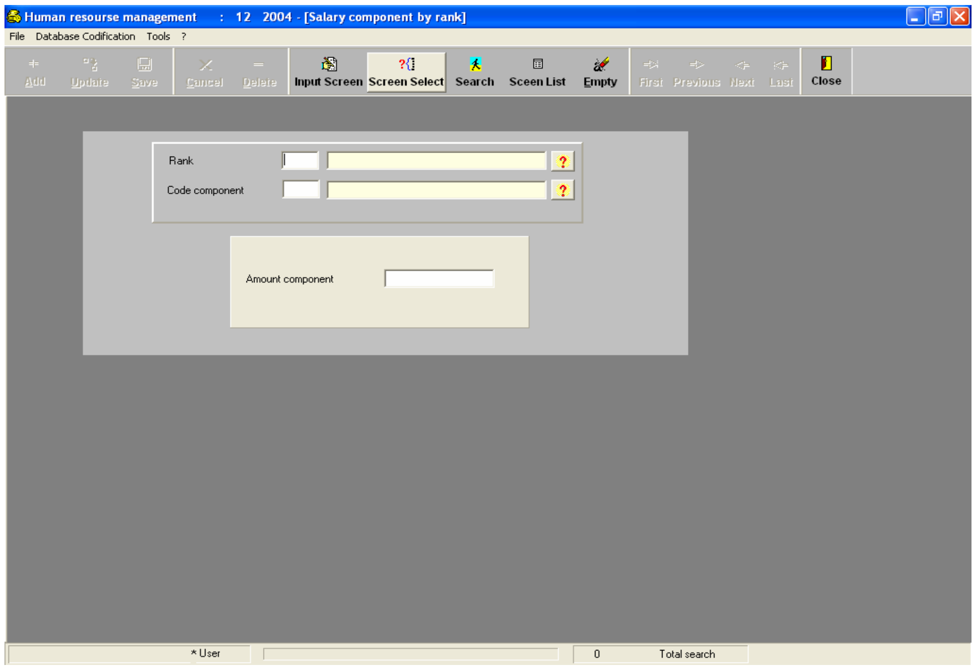
The following screen allows the searching of the salary components by category and type:


Human resourse management : 12 2004 - [Salary components by category and type]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
* User
Total search

Type
Category
Code component
Coumonent amount
0

d. *Salary Component by Rank*

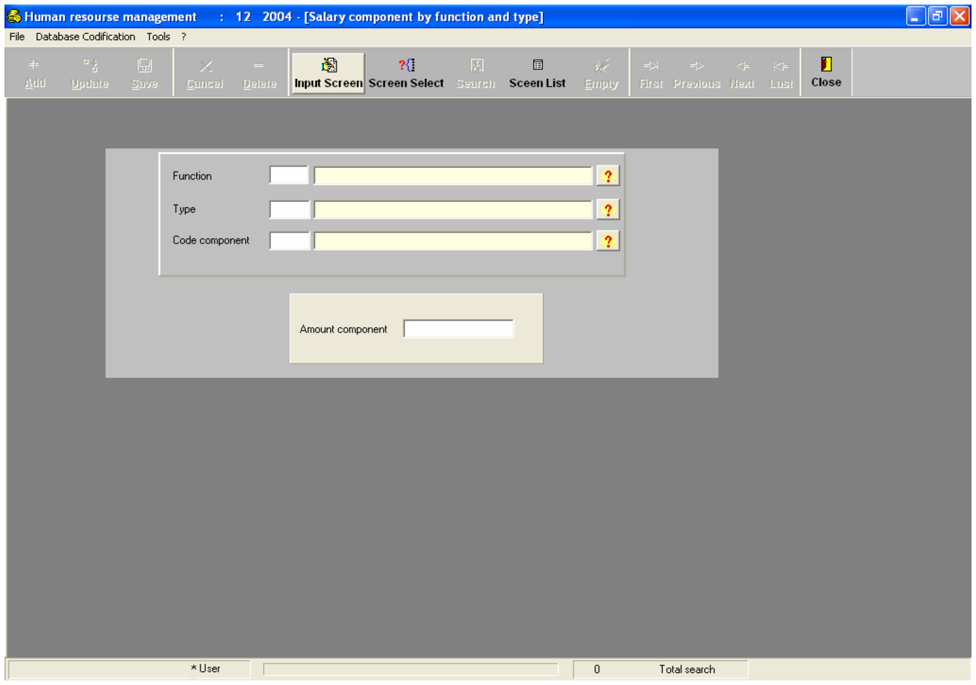
The following screen allows the searching of the salary components by rank:


Human resourse management : 12 2004 - [Salary component by rank]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
* User
Total search

Rank
Code component
Amount component
0

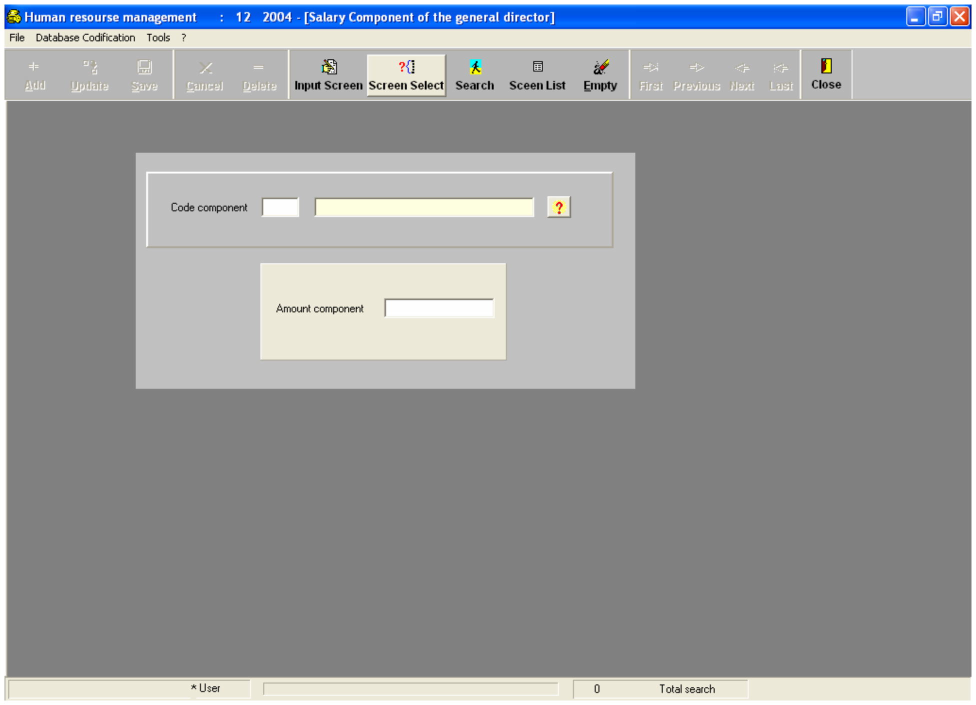
e. *Salary Component by Function and Type*

The following screen allows the searching of the salary components by function and type:


Human resourse management : 12 2004 - [Salary component by function and type]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Function
Type
Code component
Amount component
* User
0 Total search

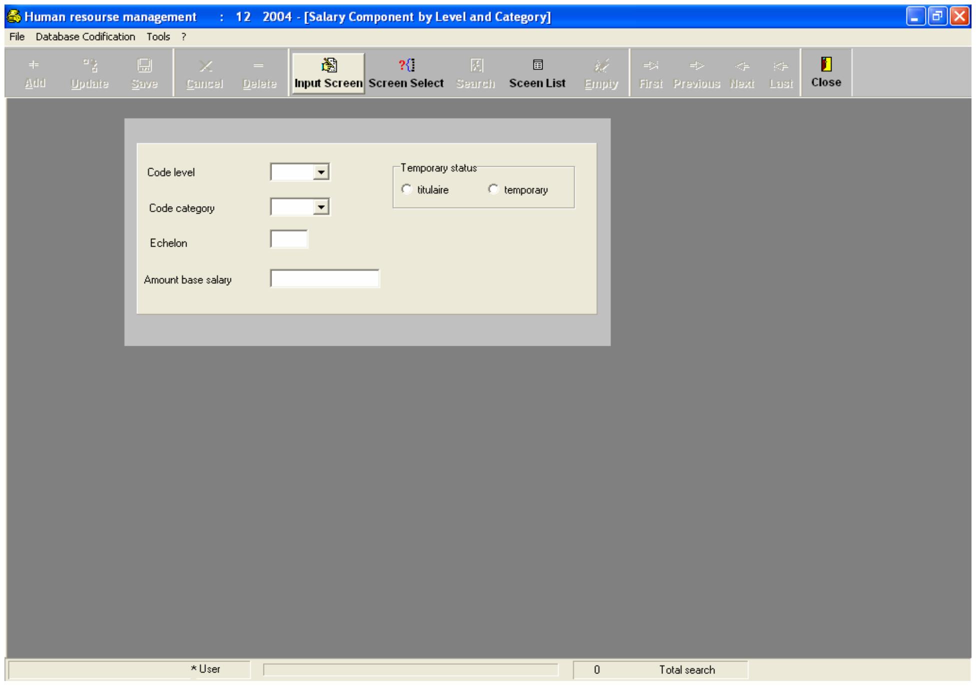
f. *Salary Component for General Director*

The following screen allows the searching of the salary components for the general director:


Human resourse management : 12 2004 - [Salary Component of the general director]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Code component
Amount component
* User
0 Total search

g. *Salary Component by Level and Category*

The following screen allows the searching of the salary components by level and category:


Human resourse management : 12 2004 - [Salary Component by Level and Category]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
* User
0 Total search

Code level

Code category

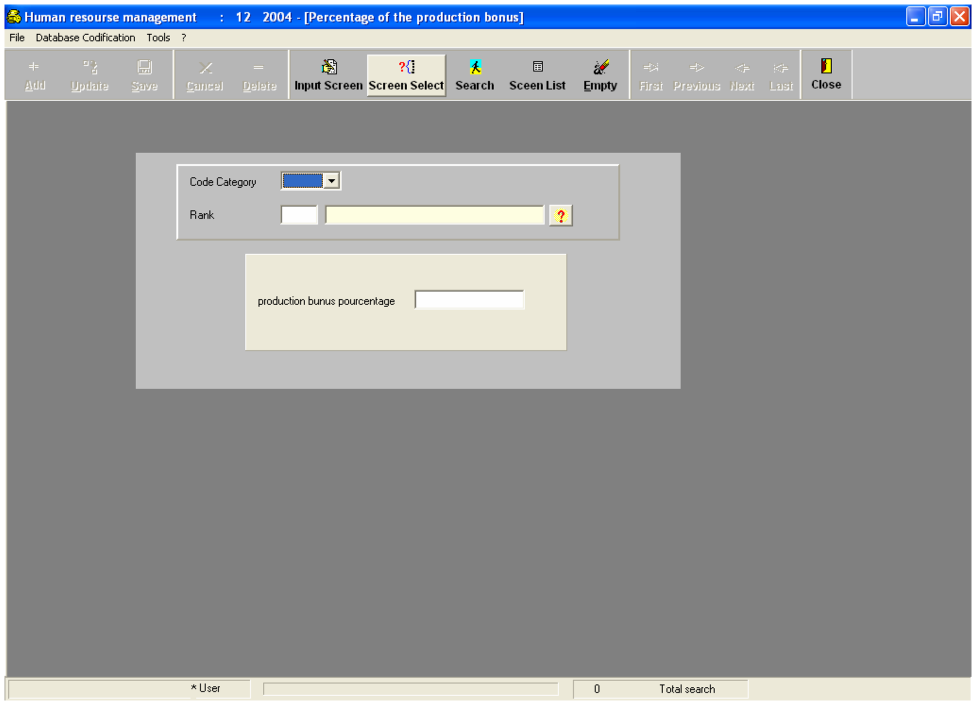
Echelon

Amount base salary

Temporary status: titulaire / temporary

h. *Percentage of Production Bonus*

The following screen allows the searching of the percentage of the production bonus:


Human resourse management : 12 2004 - [Percentage of the production bonus]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
* User
0 Total search

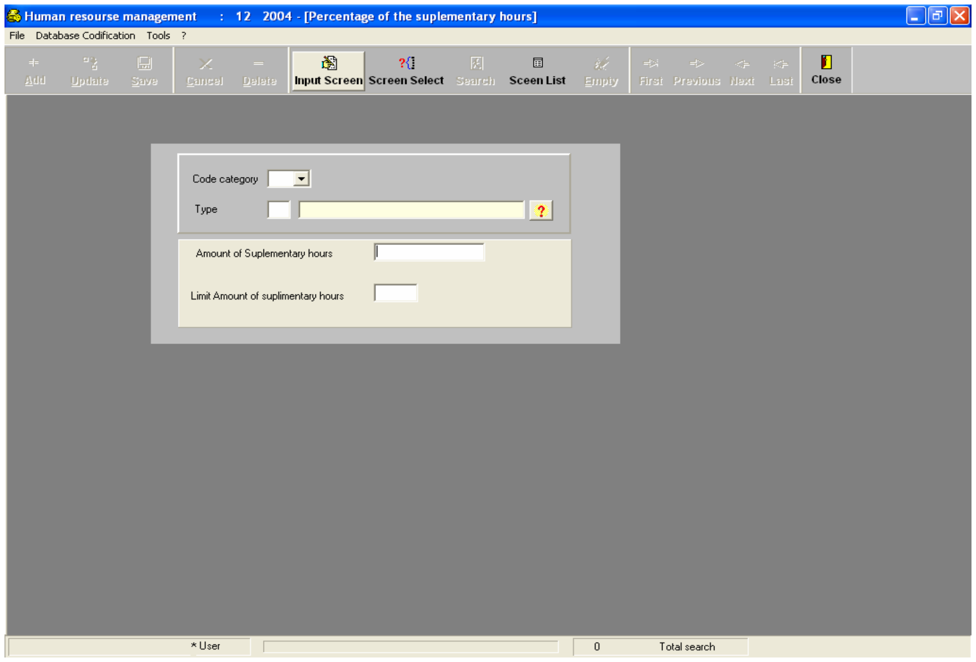
Code Category

Rank

production bunus pourcentage

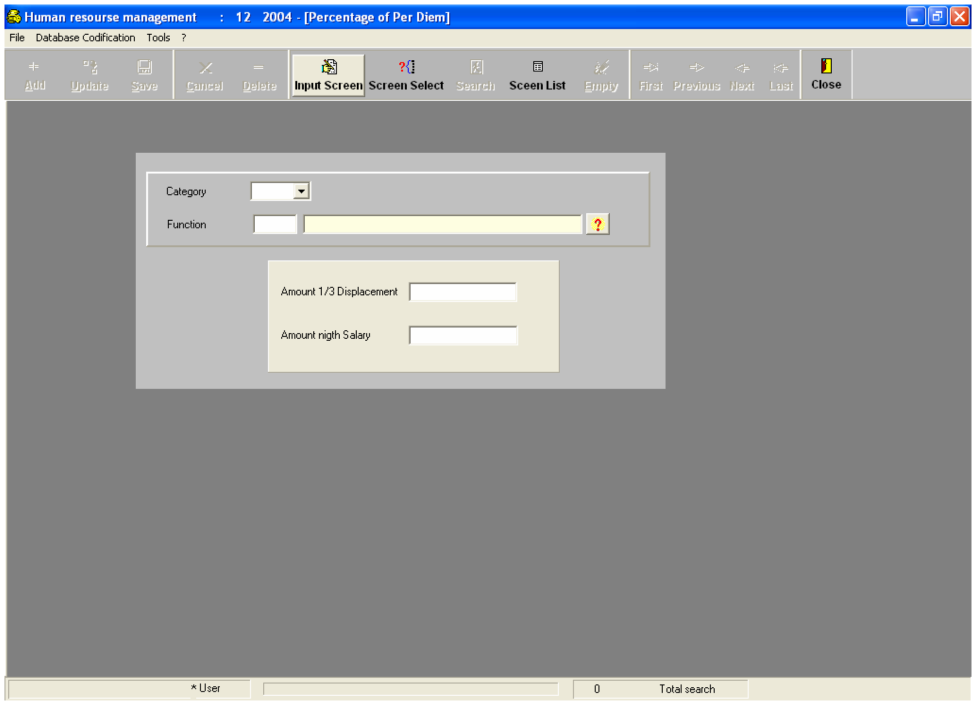
i. Percentage of Supplementary Hours

The following screen allows the searching of the percentage of the supplementary hours:


Human resourse management : 12 2004 - [Percentage of the suplementary hours]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Code category
Type
Amount of Suplementary hours
Limit Amount of suplimentary hours
* User
0 Total search

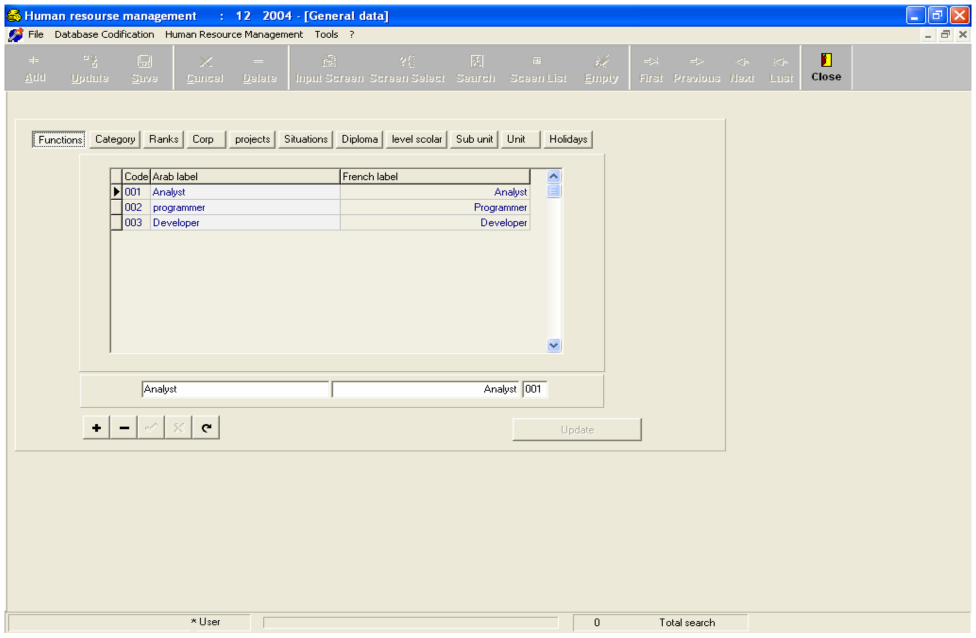
j. Percentage of Per Diem

The following screen allows the searching of the percentage of the Per Diem:


Human resourse management : 12 2004 - [Percentage of Per Diem]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Category
Function
Amount 1/3 Displacement
Amount nigth Salary
* User
0 Total search

k. Complete Components Listing

The following screen allows the searching of the complete components list:


Human resourse management : 12 2004 - [General data]

File Database Codification Human Resource Management Tools ?

Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close

Functions Category Ranks Corp projects Situations Diploma level scolar Sub unit Unit Holidays

Code	Arab label	French label
001	Analyst	Analyst
002	programmer	Programmer
003	Developer	Developer

Analyst | Analyst | 001

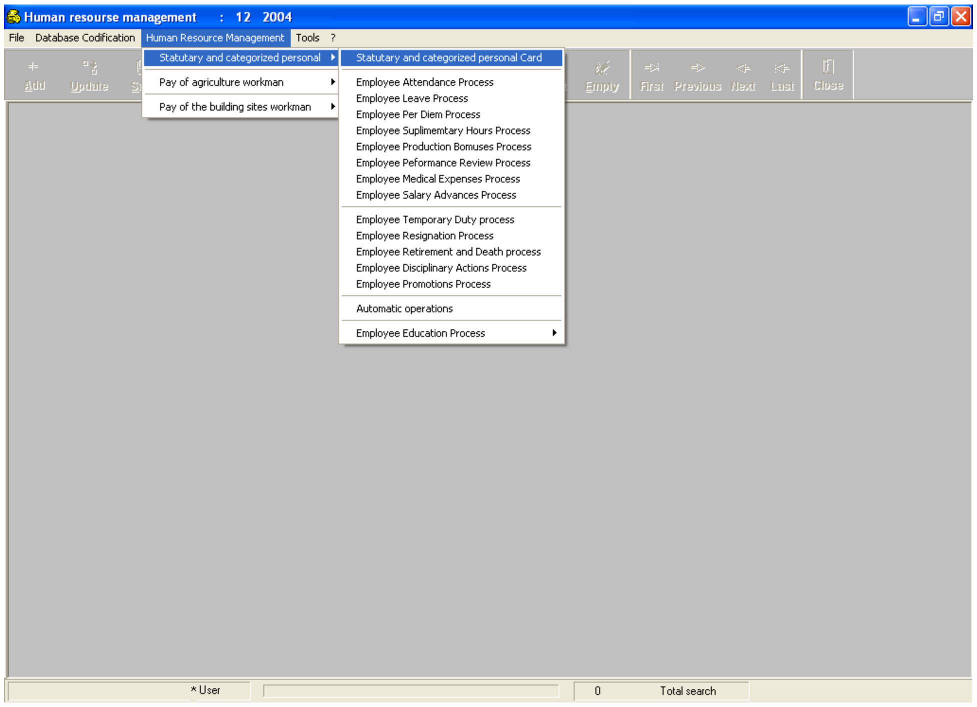
Update

* User 0 Total search

2. Human Resource Management

The next screen presents all the functionalities for the Human Resource management.

a. Statutory and Categorized Personnel


Human resourse management : 12 2004

File Database Codification Human Resource Management Tools ?

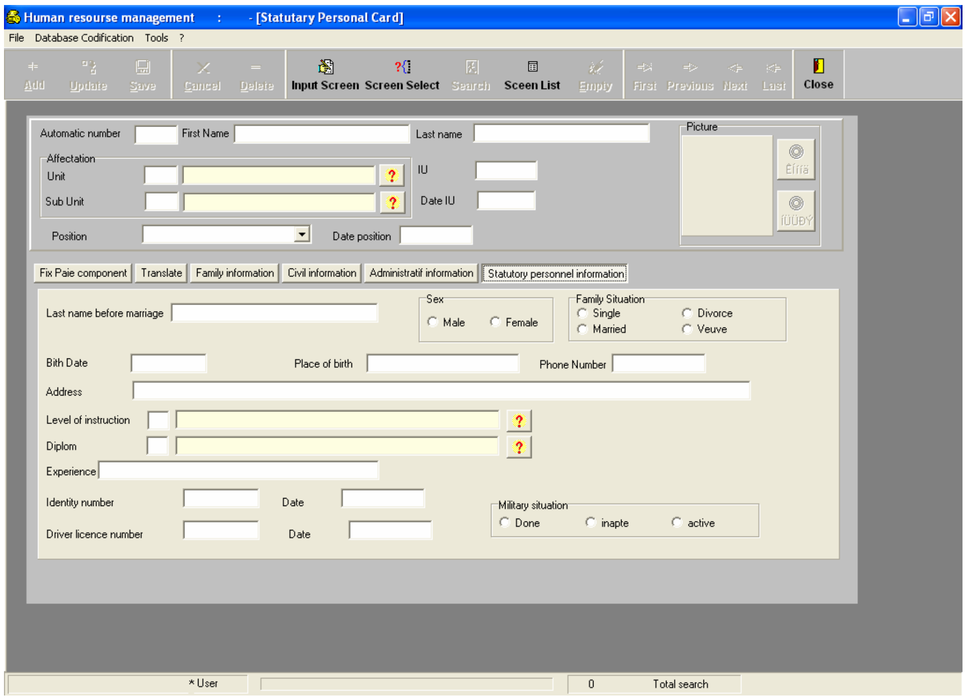
- Statutary and categorized personal
  - Statutary and categorized personal Card
  - Employee Attendance Process
  - Employee Leave Process
  - Employee Per Diem Process
  - Employee Suplimentary Hours Process
  - Employee Production Bomuses Process
  - Employee Peformance Review Process
  - Employee Medical Expenses Process
  - Employee Salary Advances Process
  - Employee Temporary Duty process
  - Employee Resignation Process
  - Employee Retirement and Death process
  - Employee Disciplinary Actions Process
  - Employee Promotions Process
  - Automatic operations
  - Employee Education Process
- Pay of agriculture workman
- Pay of the building sites workman

* User 0 Total search

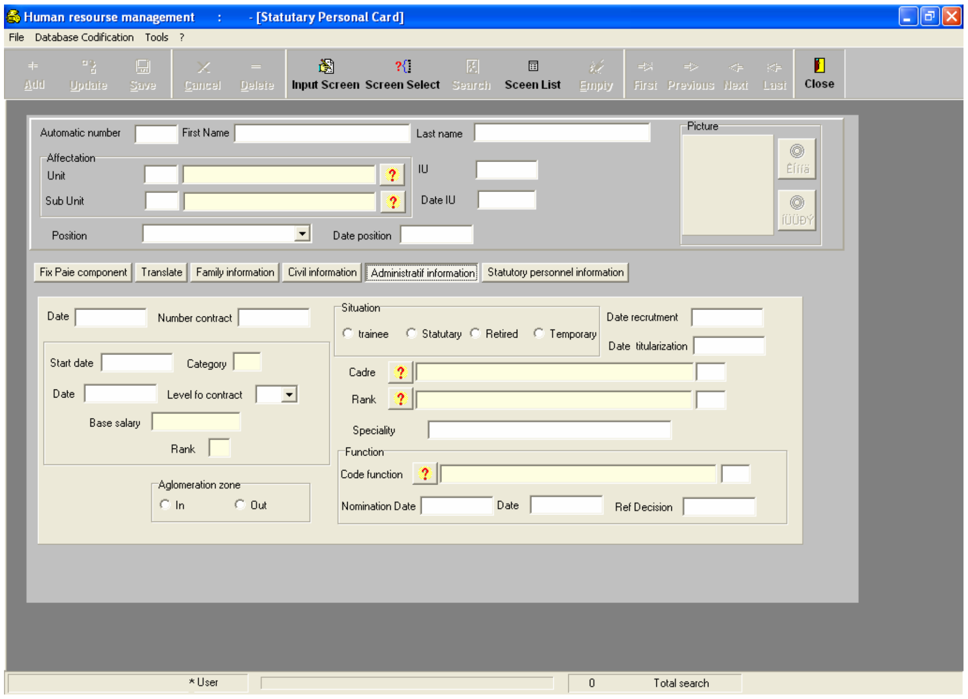
To create and update the personal card, the user must use the following screen, which is composed of two parts:

- A header: It contains personal identification (ID, first name, last name, photograph, etc....)
- List of pages
    - Statutory personnel information
    - Administrative information
    - Civil information
    - Family information
    - Translate (French to Arabic)

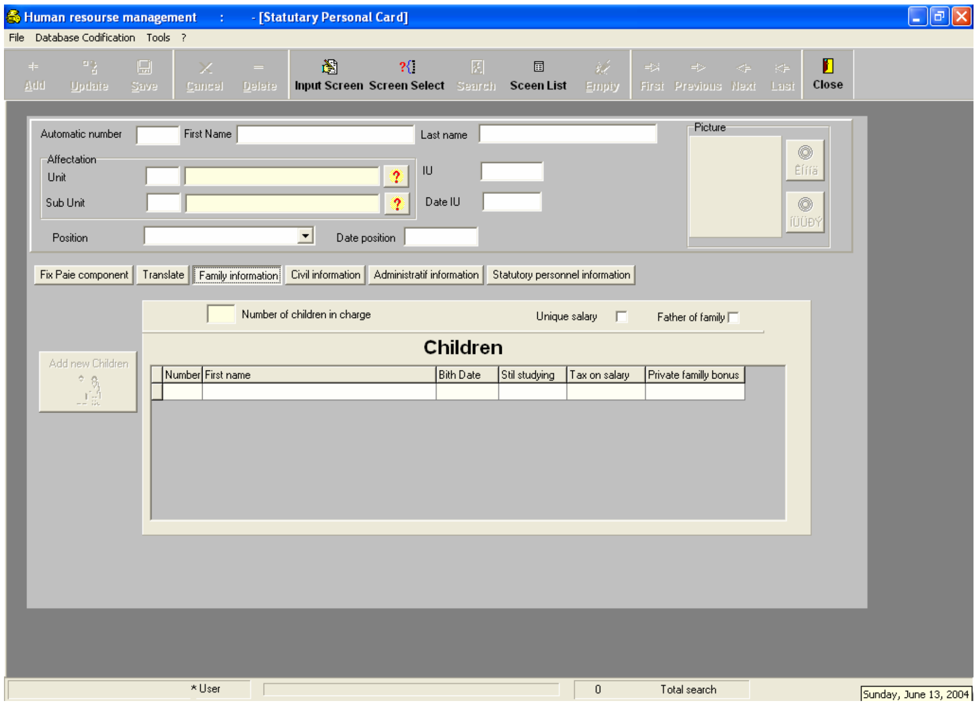
(1) Salary and Categorized Personal Card. This page shows the statutory personal information.



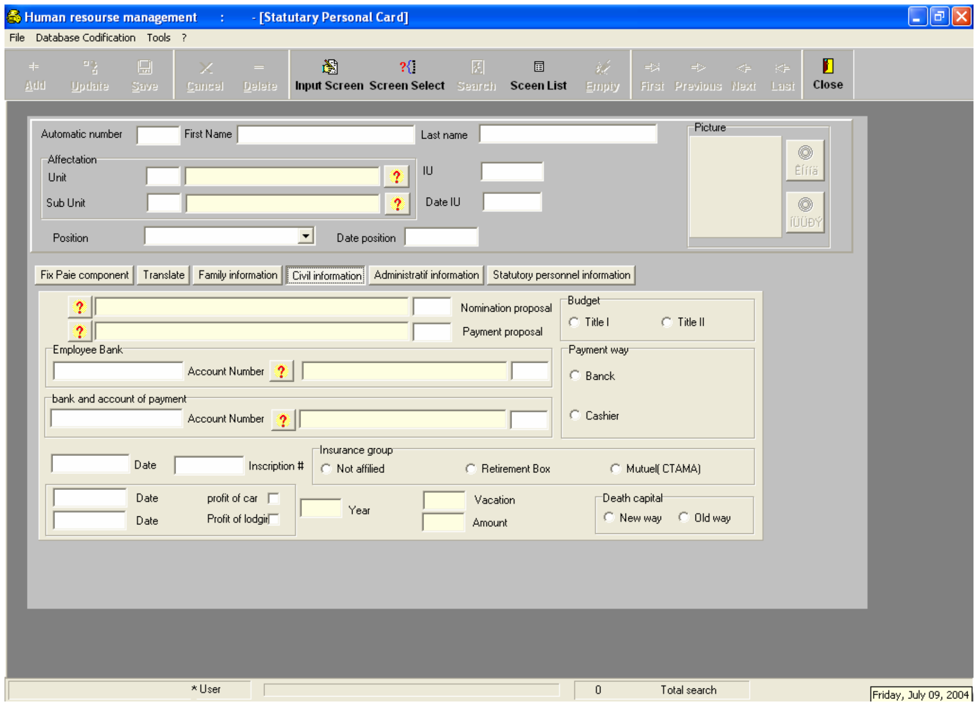
(2) Administrative Information. This page shows administrative information.


Human resourse management : - [Statutary Personal Card]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Automatic number First Name Last name Picture
Affectation
Unit
Sub Unit
IU
Date IU
Position
Date position
Effa
Fix Paie component Translate Family information Civil information Administratif information Statutory personnel information
Date Number contract
Situation
trainee Statutary Retired Temporary
Date recrutment
Date titularization
Start date Category
Date Level fo contract
Base salary
Rank
Cadre
Rank
Speciality
Function
Code function
Nomination Date Date Ref Decision
Aglomeration zone
In Out
* User
0 Total search

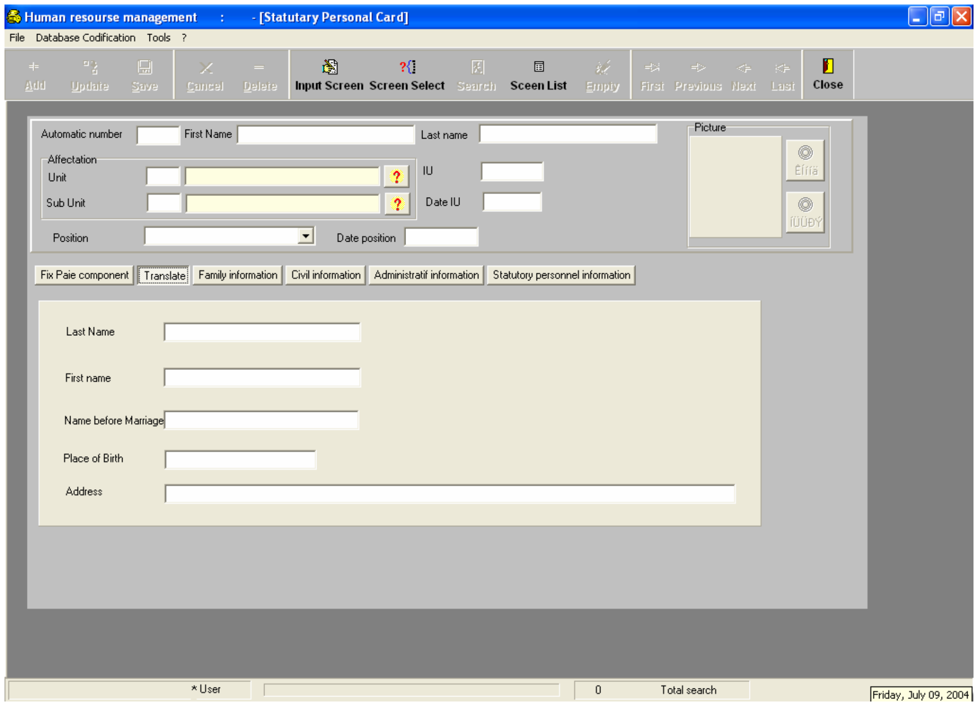
(3) Family Information. This page shows family information.


Human resourse management : - [Statutary Personal Card]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Automatic number First Name Last name Picture
Affectation
Unit
Sub Unit
IU
Date IU
Position
Date position
Effa
Fix Paie component Translate Family information Civil information Administratif information Statutory personnel information
Number of children in charge
Unique salary
Father of family
Add new Children
Children
Number First name Bith Date Stil studying Tax on salary Private familly bonus
* User
0 Total search
Sunday, June 13, 2004

(4) Civil Information. This page shows civil information.

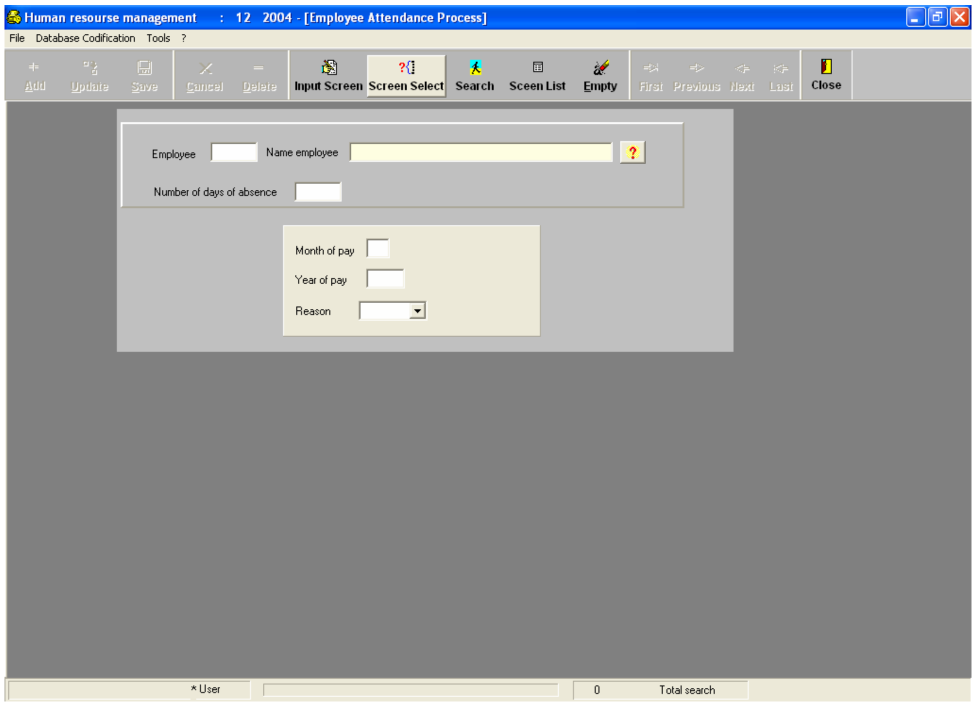

Human resourse management : - [Statutary Personal Card]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Automatic number First Name Last name Picture
Affectation Unit Sub Unit IU Date IU
Position Date position
Fix Paie component Translate Family information Civil information Administratif information Statutory personnel information
Nomination proposal Payment proposal Budget Title I Title II
Employee Bank Account Number Payment way Banck Cashier
bank and account of payment Account Number
Date Inscription # Insurance group Not affilied Retirement Box Mutuel(CTAMA)
Date profit of car Date Profit of lodgin Year Vacation Amount Death capital New way Old way
* User 0 Total search Friday, July 09, 2004

(5) Translate Information. This page shows the translate information (French to Arabic).


Human resourse management : - [Statutary Personal Card]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Automatic number First Name Last name Picture
Affectation Unit Sub Unit IU Date IU
Position Date position
Fix Paie component Translate Family information Civil information Administratif information Statutory personnel information
Last Name
First name
Name before Marriage
Place of Birth
Address
* User 0 Total search Friday, July 09, 2004

b. *Employee Attendance Process*

The following screen shows the employee attendance process:



Human resource management : 12 2004 - [Employee Attendance Process]

Employee | Name employee

Number of days of absence

Month of pay

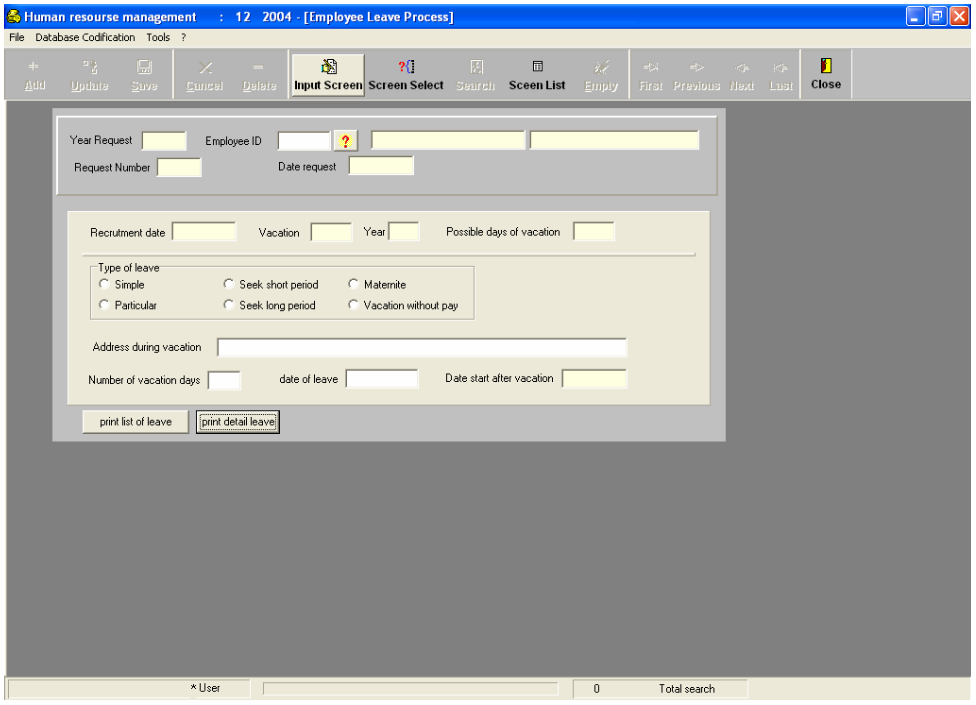
Year of pay

Reason

* User | 0 | Total search

c. *Employee Leave Process*

The following screen shows the employee leave process:



Human resource management : 12 2004 - [Employee Leave Process]

Year Request | Employee ID

Request Number | Date request

Recrutment date | Vacation | Year | Possible days of vacation

Type of leave: Simple, Particular, Seek short period, Seek long period, Maternite, Vacation without pay

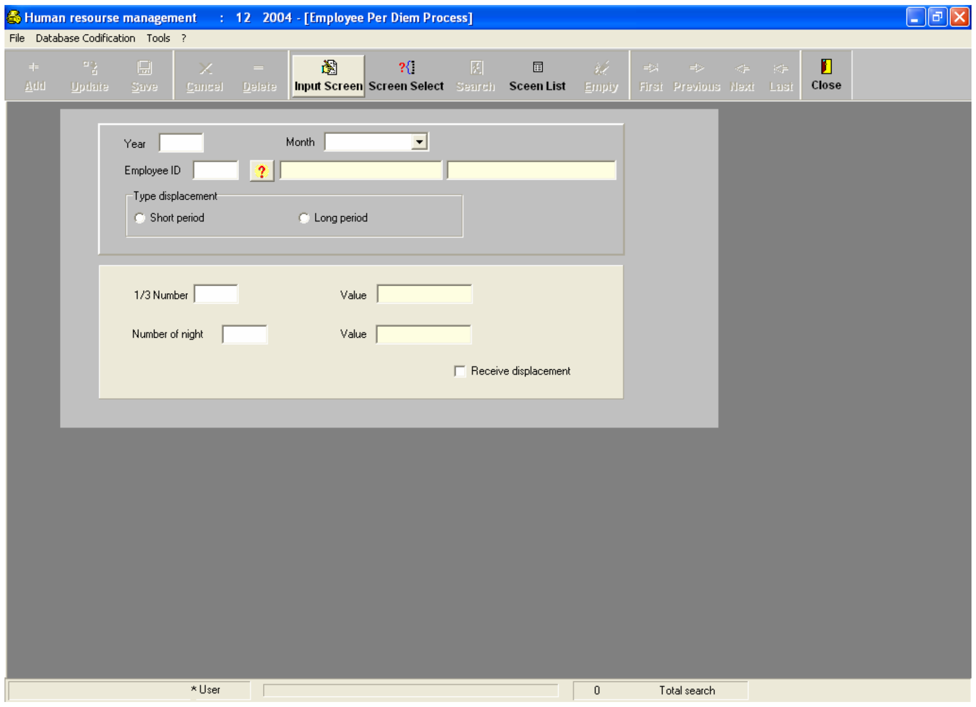
Address during vacation

Number of vacation days | date of leave | Date start after vacation

* User | 0 | Total search

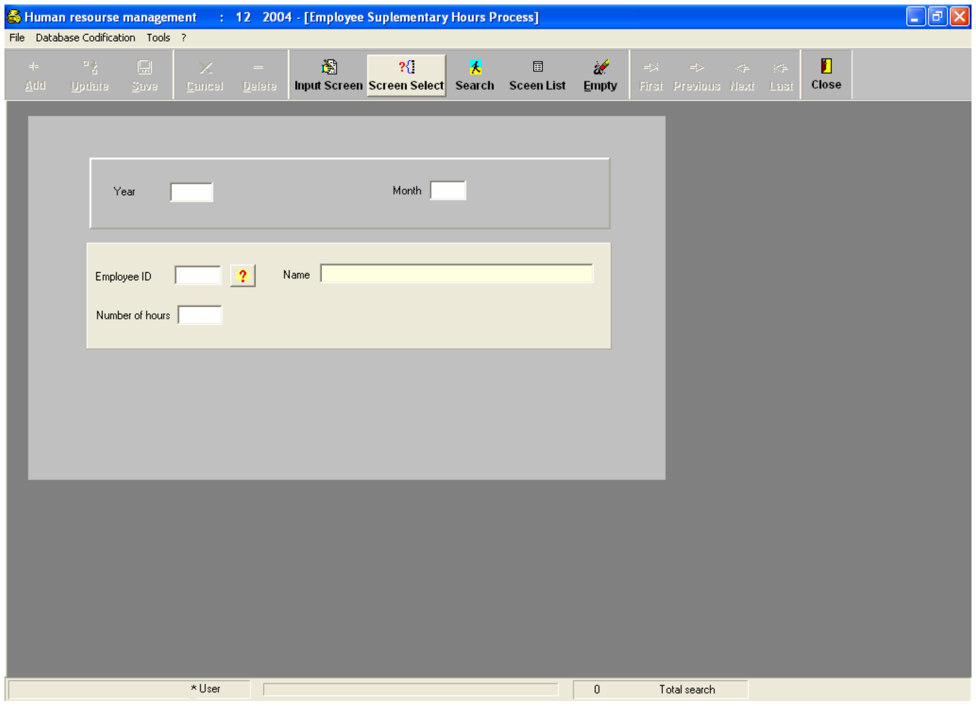
d. Employee Per Diem Process

The following screen shows the employee per diem process:


Human resourse management : 12 2004 - [Employee Per Diem Process]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Year
Month
Employee ID
Type displacement
Short period
Long period
1/3 Number
Value
Number of night
Value
Receive displacement
* User
0
Total search

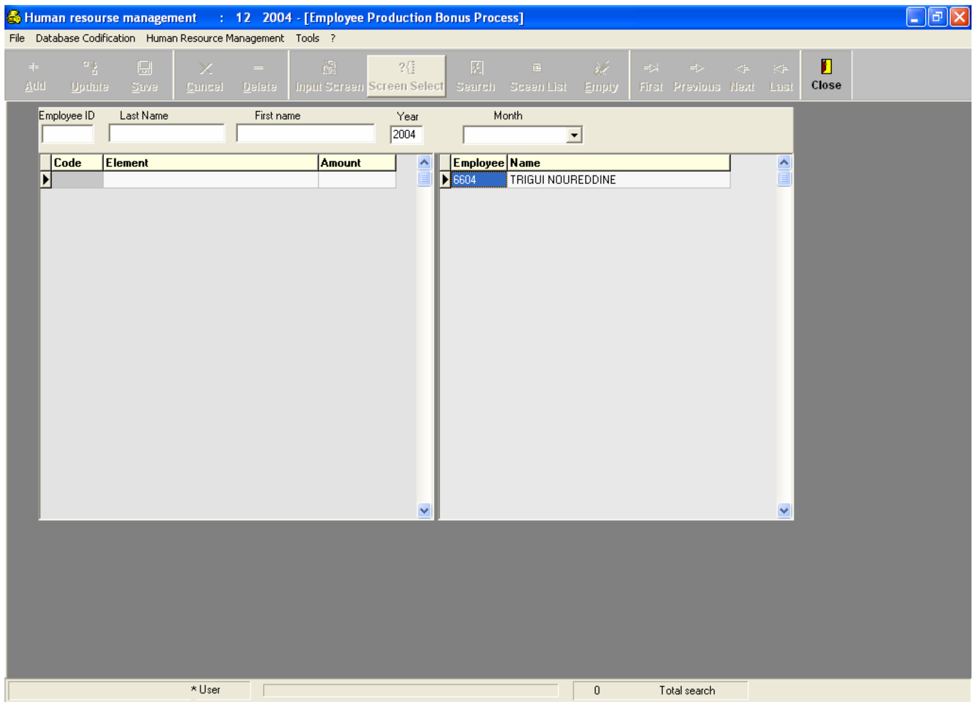
e. Employee Supplementary Hour Process

The following screen shows the employee supplementary hour process:


Human resourse management : 12 2004 - [Employee Suplementary Hours Process]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Year
Month
Employee ID
Name
Number of hours
* User
0
Total search

f. Employee Production Bonus Process

The following screen shows the employee production bonus process:


Human resourse management : 12 2004 - [Employee Production Bonus Process]
File Database Codification Human Resource Management Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Employee ID Last Name First name Year Month
Code Element Amount
Employee Name
* User
Total search

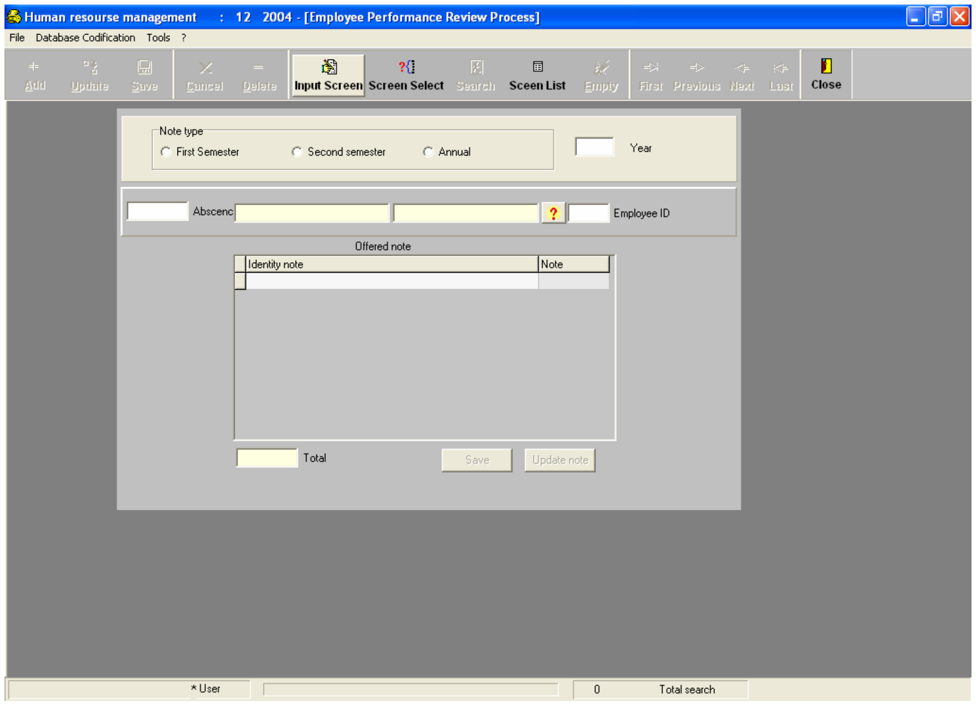
Year: 2004

Employee	Name
5604	TRIGUI NOUREDDINE

0 Total search

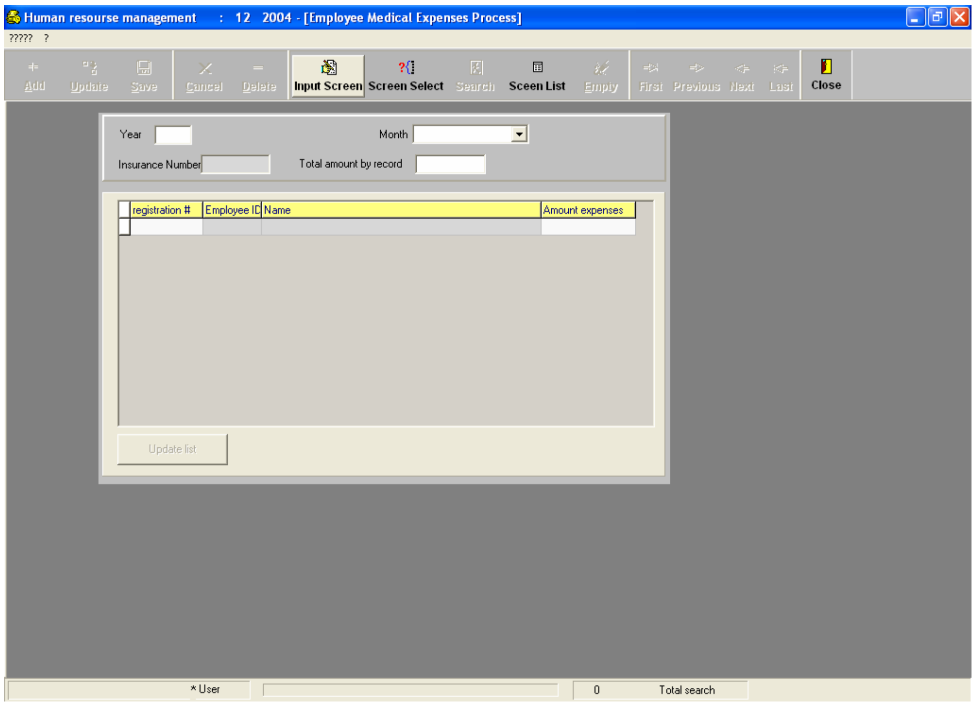
g. Employee Performance Review Process

The following screen shows the employee performance review process:


Human resourse management : 12 2004 - [Employee Performance Review Process]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Note type
First Semester
Second semester
Annual
Year
Abscenc
Employee ID
Offered note
Identity note
Note
Total
Save
Update note
* User
0
Total search

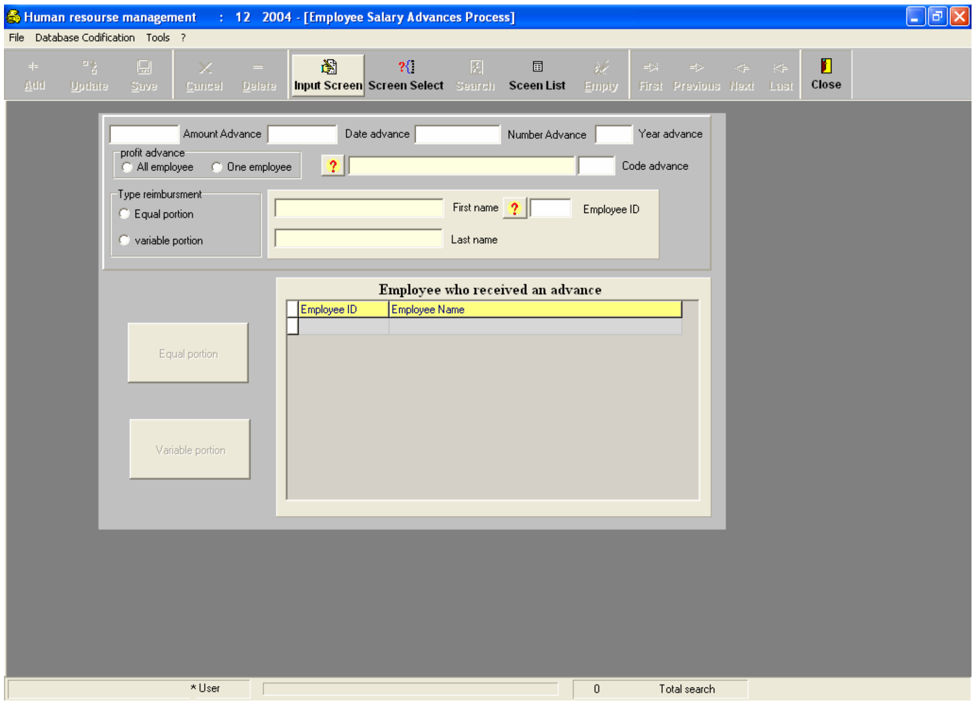
h. *Employee Medical Expenses Process*

The following screen shows the employee medical expenses process:



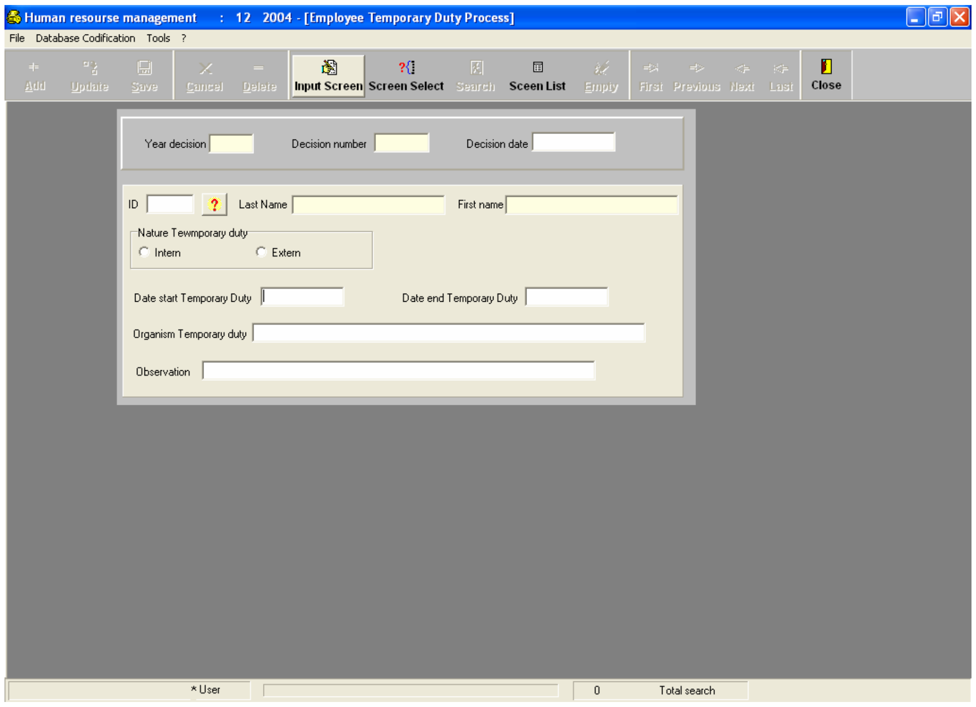
i. *Employee Salary Advances Process*

The following screen shows the employee salary advances process:



j. *Employee Temporary Duty Process*

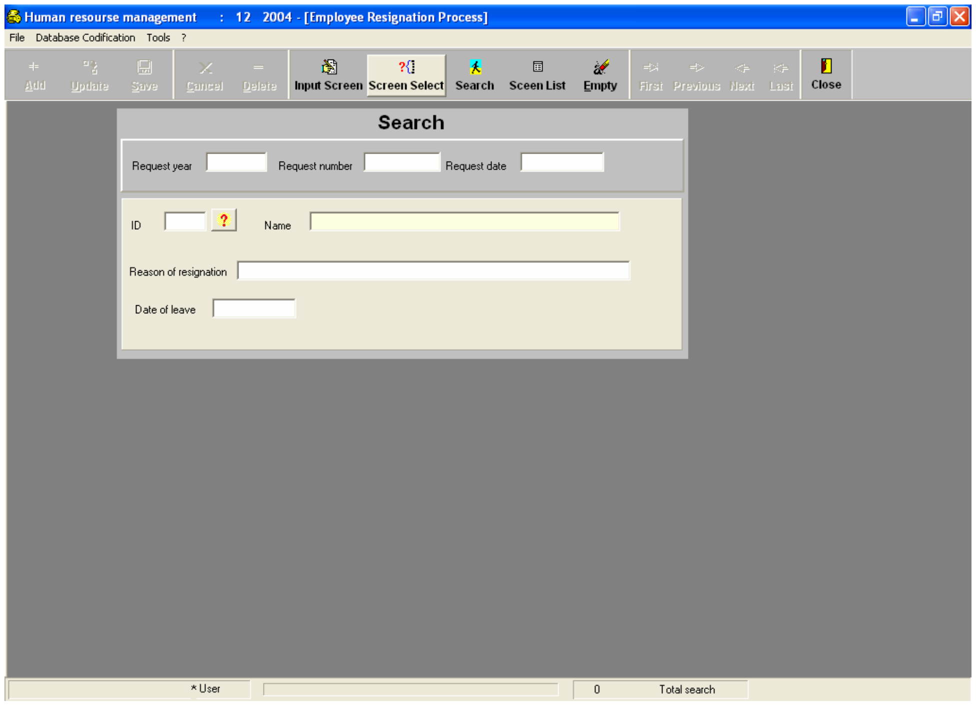
The following screen shows the employee temporary duty process:


Human resourse management : 12 2004 - [Employee Temporary Duty Process]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
* User
0 Total search

Year decision
Decision number
Decision date
ID
Last Name
First name
Nature Tewmporary duty
Intern
Extern
Date start Temporary Duty
Date end Temporary Duty
Organism Temporary duty
Observation

k. *Employee Resignation Process*

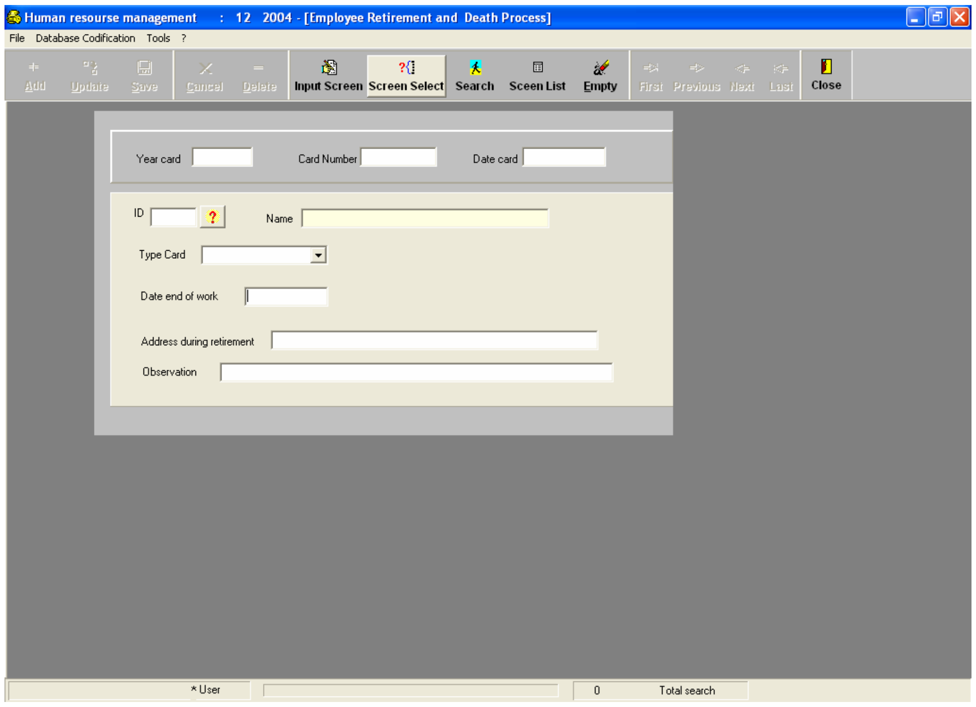
The following screen shows the employee resignation process:


Human resourse management : 12 2004 - [Employee Resignation Process]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
* User
0 Total search

Search
Request year
Request number
Request date
ID
Name
Reason of resignation
Date of leave

l. Employee Retirement and Death Process

The following screen shows the employee retirement or death process:


Human resourse management : 12 2004 - [Employee Retirement and Death Process]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
* User
0 Total search

Year card

Card Number

Date card

ID

Name

Type Card

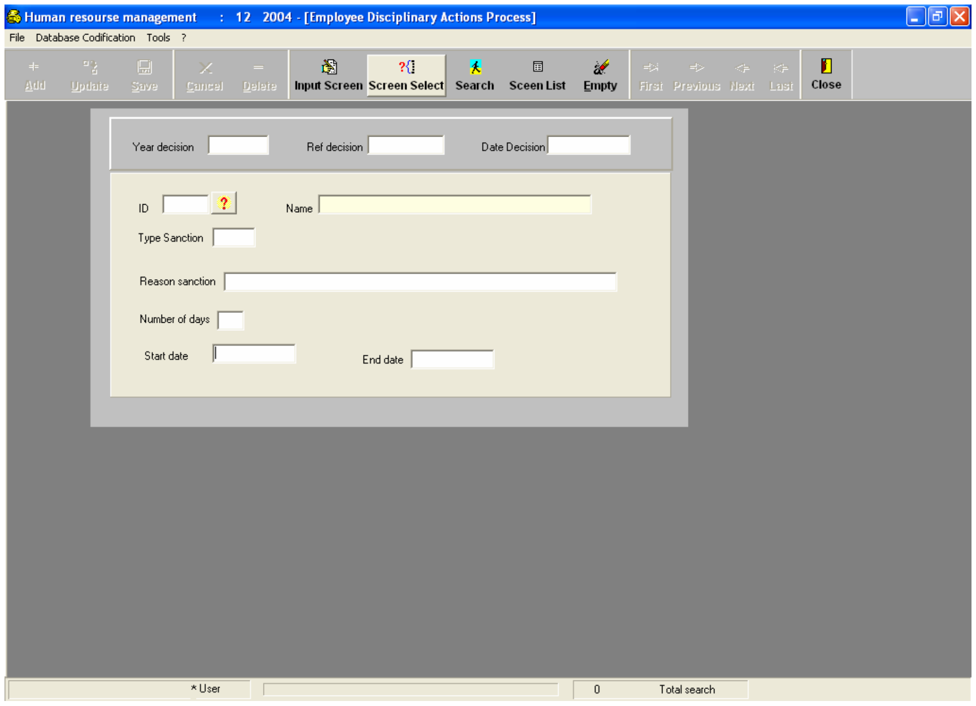
Date end of work

Address during retirement

Observation

m. Employee Disciplinary Actions Process

The following screen shows the employee disciplinary actions process:


Human resourse management : 12 2004 - [Employee Disciplinary Actions Process]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
* User
0 Total search

Year decision

Ref decision

Date Decision

ID

Name

Type Sanction

Reason sanction

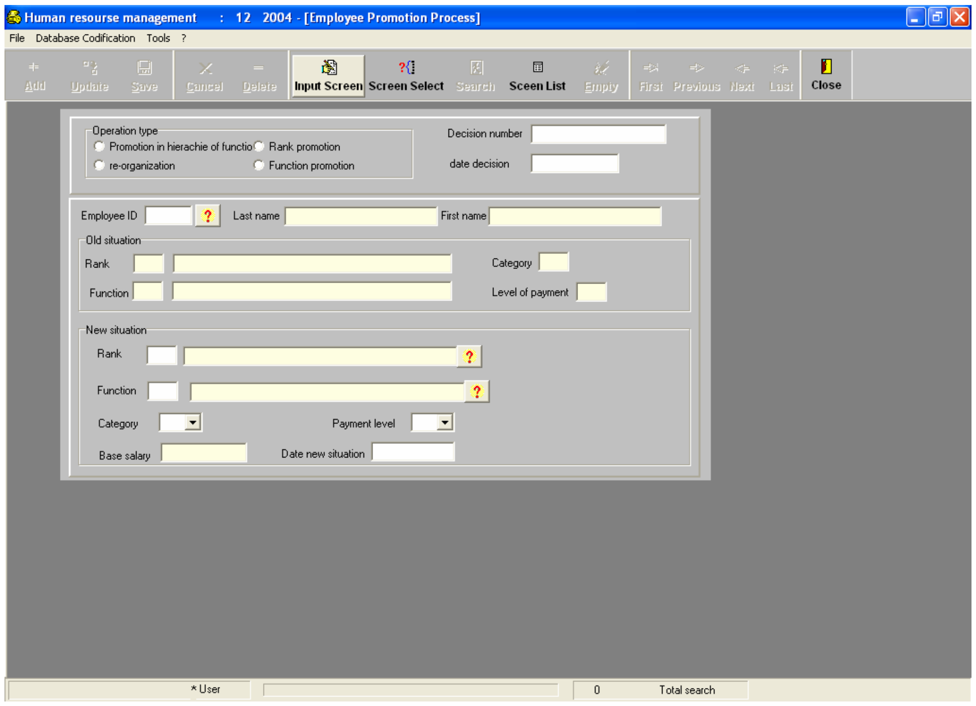
Number of days

Start date

End date

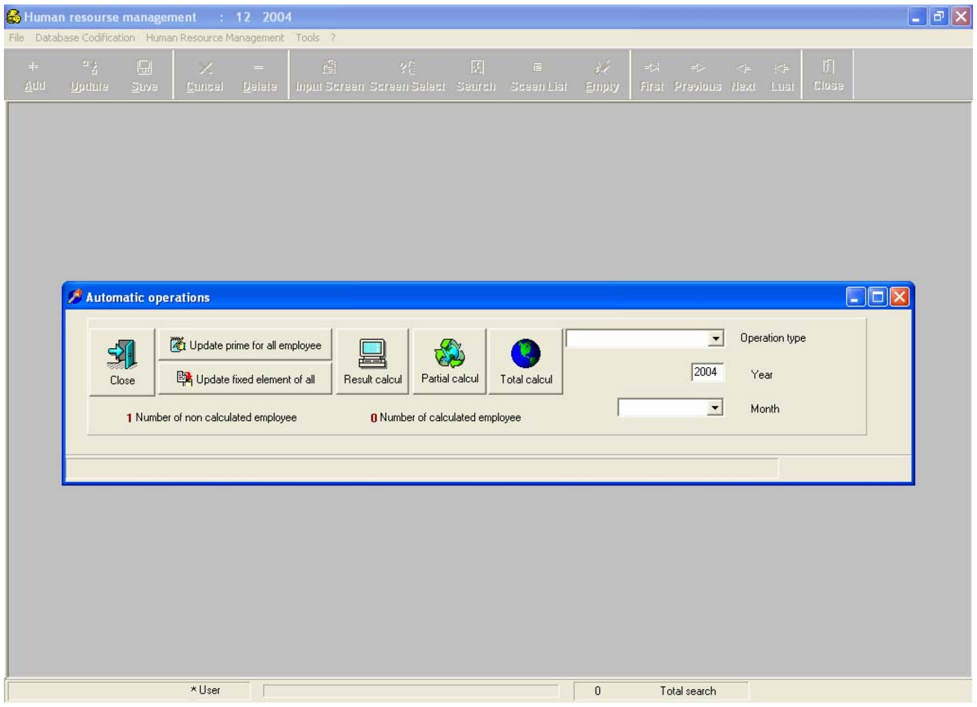
n. *Employee Promotion Process*

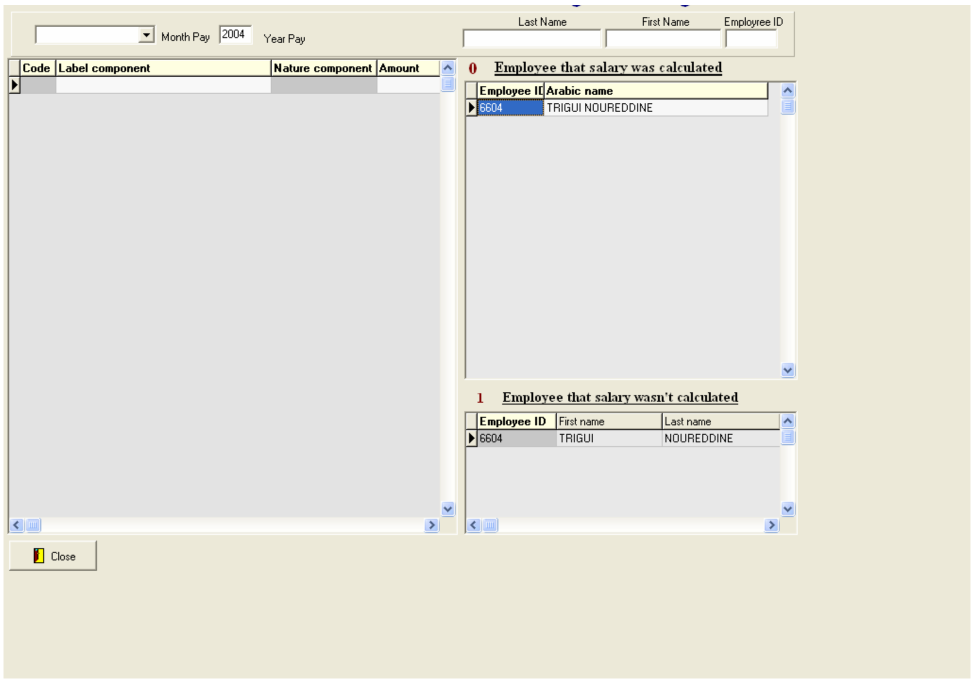
The following screen shows the employee promotion process:



o. *Automatic Operations*

The following two screens show the automatic operations:


Human resource management : 12 2004
File Database Codification Human Resource Management Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Automatic operations
Close
Update prime for all employee
Update fixed element of all
Result calcul
Partial calcul
Total calcul
Operation type
2004
Year
Month
1 Number of non calculated employee
0 Number of calculated employee
* User
0
Total search


Month Pay 2004 Year Pay

Last Name | First Name | Employee ID

Code	Label component	Nature component	Amount

0 Employee that salary was calculated

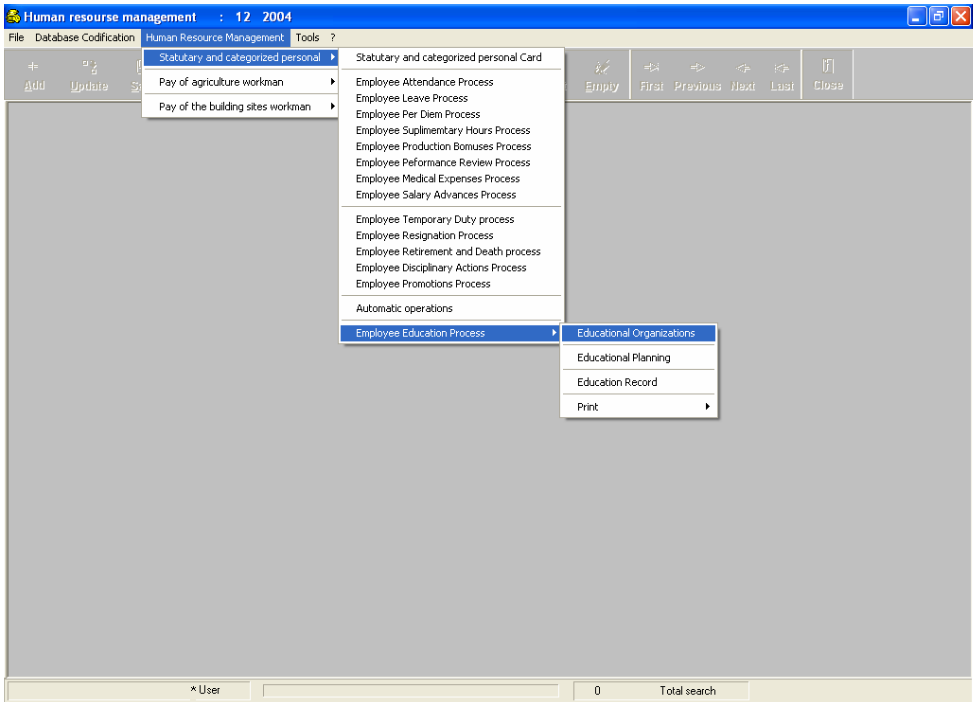
Employee ID	Arabic name
6604	TRIGUI NOUREDDINE

1 Employee that salary wasn't calculated

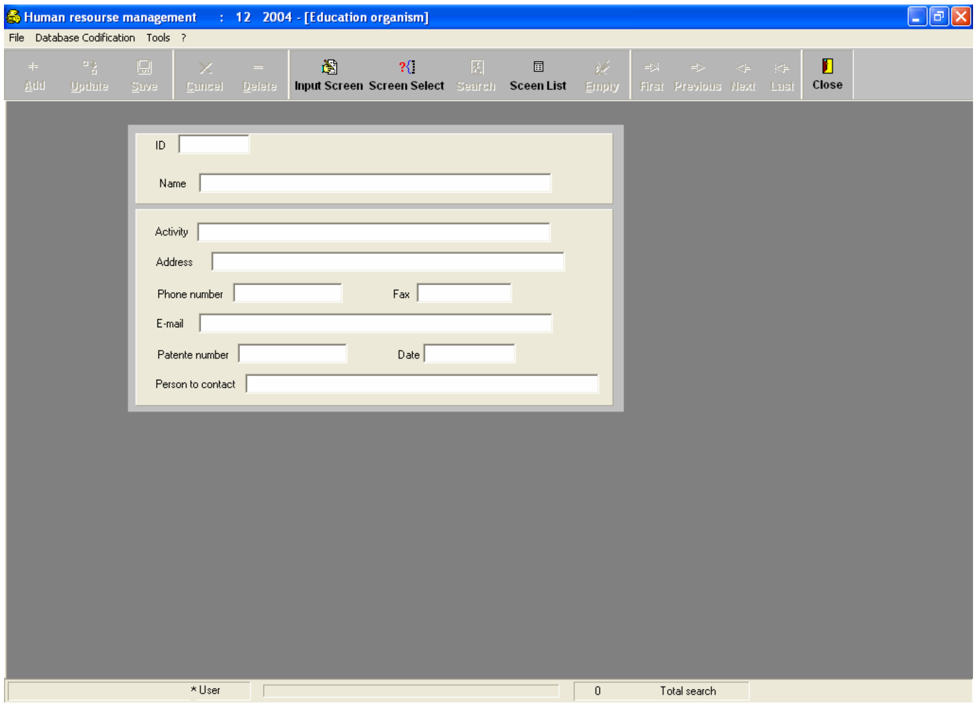
Employee ID	First name	Last name
6604	TRIGUI	NOUREDDINE

p. Employee Education Process

The following screen shows the employee education process:

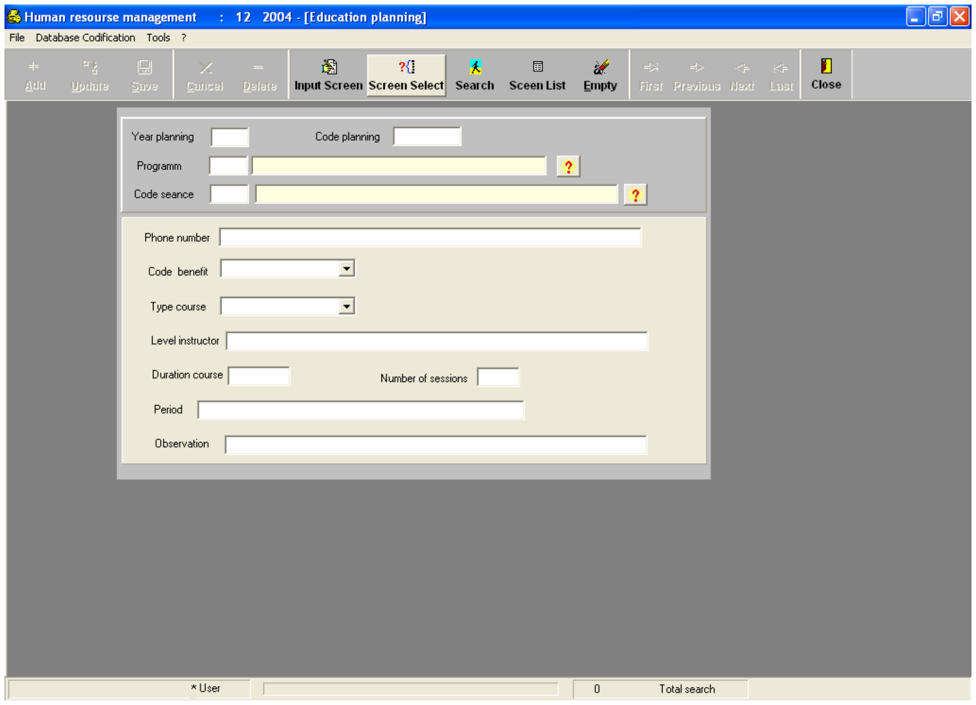

Human resourse management : 12 2004
File
Database Codification
Human Resource Management
Tools
?
Add
Update
Empty
First
Previous
Next
Last
Close
Statutary and categorized personal
Pay of agriculture workman
Pay of the building sites workman
Statutary and categorized personal Card
Employee Attendance Process
Employee Leave Process
Employee Per Diem Process
Employee Suplimemtary Hours Process
Employee Production Bomuses Process
Employee Peformance Review Process
Employee Medical Expenses Process
Employee Salary Advances Process
Employee Temporary Duty process
Employee Resignation Process
Employee Retirement and Death process
Employee Disciplinary Actions Process
Employee Promotions Process
Automatic operations
Employee Education Process
Educational Organizations
Educational Planning
Education Record
Print
* User
0
Total search

(1) Educational Organization. The following screen shows the education organization:


Human resourse management : 12 2004 - [Education organism]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close

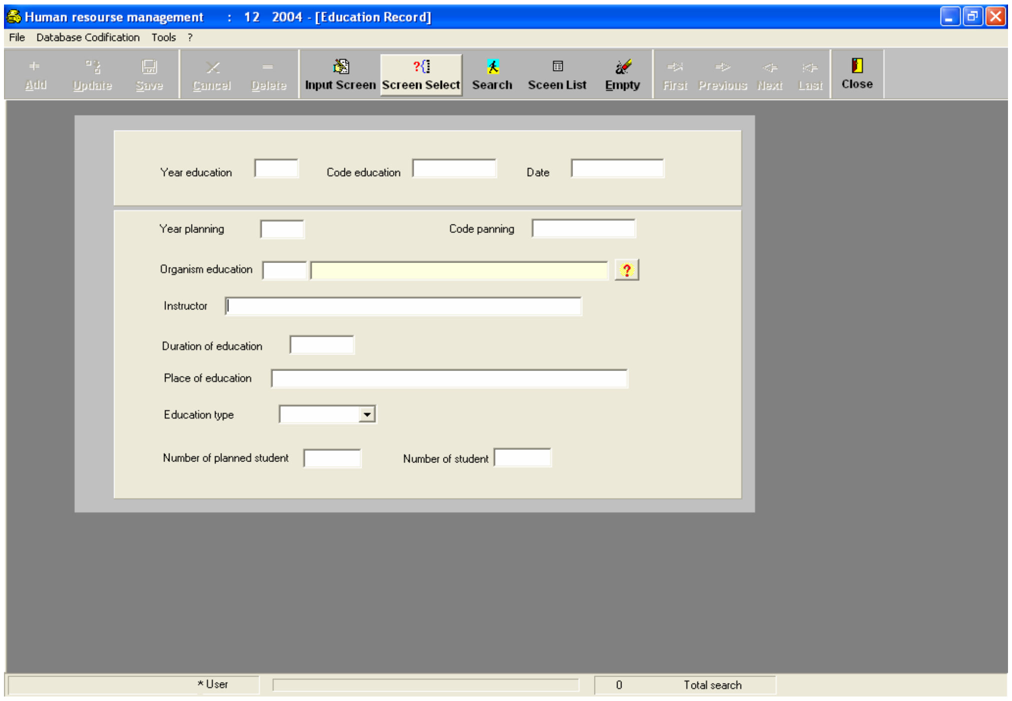
ID
Name
Activity
Address
Phone number Fax
E-mail
Patente number Date
Person to contact

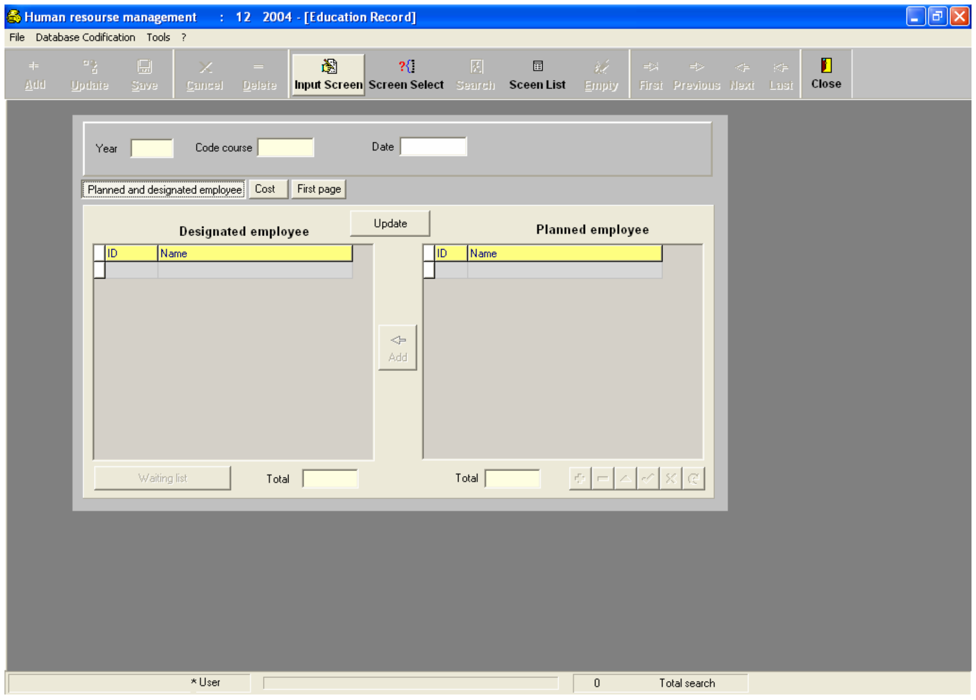
(2) Educational Planning. The following screen shows the education planning process:


Human resourse management : 12 2004 - [Education planning]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close

Year planning Code planning
Programm
Code seance
Phone number
Code benefit
Type course
Level instructor
Duration course Number of sessions
Period
Observation

(3) Educational Record. The two following screens show the education record:


Human resourse management : 12 2004 - [Education Record]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Year education
Code education
Date
Year planning
Code panning
Organism education
Instructor
Duration of education
Place of education
Education type
Number of planned student
Number of student
* User
0
Total search


Human resourse management : 12 2004 - [Education Record]
File Database Codification Tools ?
Add Update Save Cancel Delete Input Screen Screen Select Search Sceen List Empty First Previous Next Last Close
Year
Code course
Date
Planned and designated employee
Cost
First page
Designated employee
Update
Planned employee
ID
Name
ID
Name
Add
Waiting list
Total
Total
* User
0
Total search

The presented prototype could begin the actual development of the human resource subsystem. An attempt was made to develop a standard screen to use for the future development of the other subsystem for the ISYDMA system.

THIS PAGE INTENTIONALLY LEFT BLANK

V. SUMMARY AND CONCLUSION

The author believes that the essential key to success in information system design is effective planning. However, before designing an information system, it is necessary to fully understand exactly what information is being managed and how that information will be used by the end-users. Even so, when the planning is complete, what are the first steps that information designers will take as they begin to work with a given set of content? And what is the actual, concrete tool that developers use to organize, store, manage, and manipulate data? Quite simply, databases are the essential ingredient to managing large amounts of information in useful ways.

Understanding the concept of databases, how they work, and how they can be used is central to understanding many of the issues involved in developing information system design strategies. Information system designers are required to break down a given set of information to its most basic components, identifying each individual element and specifying how all those unique elements relate to each other. This was our focus in this research, so the new system encompasses many of the same purposes and functions. But its methods will be different. Instead of manual procedures, automatic ones will be used to manage the human resource, budget, finance, accounting, and provisioning. The system will keep track of all data in a central database using IBM-INFORMIX Database management system.

In the development of the prototype, the Delphi language was selected because it is a Rapid Application Development (RAD) tool. It combines the power of visual development with a robust, object-oriented language (Object Pascal) and fast, solid, native-code compiler. Delphi has features that promote code-reuse; Delphi includes an "object repository" where we can store commonly used forms and either reuse them or inherit from them to build other forms.

Finally our future work will focus on the design and the implementation of the provisioning, finance, budgetary and accounting subsystem and the integration of those subsystems with the human recourse subsystem.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. DATA DICTIONARY

Data element name	Field name	Code format
Absence number	NBR_ABS	Integer
Account number of education organism	NUM_CMP_FOR	Decimal(12.3)
Activity instructor	ACT_FOR	Char(50)
Address education	ADR_FOR	Char(50)
Address stagier	ADR_STA	Char(50)
Address when retired	ADR_DOM_PEN	Char(50)
Amount Advance month	MNT_AVC_MOI	Decimal(12.3)
Amount base salary	MNT_BAS	Decimal(12.3)
Amount bonus	MNT_PRI	Decimal(12.3)
Amount component function	MNT_COM_FON	Decimal(12.3)
Amount component rank	MNT_COM_GRD	Decimal(12.3)
Amount per diem	MNT_DEP	Decimal(12.3)
Amount given by employee office	MNT_BOU_BUR	Decimal(12.3)
Amount given by the organization	MNT_BOU_EMP	Decimal(12.3)
Amount given for social security	MNT_CNSS	Decimal(12.3)
Amount IRPP	MNT_IRPP	Decimal(12.3)
Amount of 1_3 per diem	MNT_TIER_DEP	Decimal(12.3)
Amount of advance	MNT_AVA	Decimal(12.3)
Amount Production bonuses	MNT_PRI_REN	Decimal(12.3)
Amount salary	MNT_SAL	Decimal(12.3)
Ancient echelon	ANC_ECH	VarChar(2)
Ancient rank	ANC_GRD	VarChar(4)
Ancient level	ANC_NIV	VarChar(2)
Arabic action label	LIB_ACT_ARA	VarChar(40)
Arabic address person	ADR_PER_ARA	VarChar(80)
Arabic code category	COD_CAT_ARA	VarChar(2)
Arabic first name person	NOM_PER_ARA	VarChar(30)
Arabic label bank	LIB_BNQ_ARA	VarChar(30)
Arabic label component	LIB_COM_ARA	VarChar(40)
Arabic label diploma	LIB_DIP_ARA	VarChar(30)
Arabic label function	LIBE_FON_ARA	VarChar(40)
Arabic label insurer	LIB_ASS_ARA	VarChar(40)
Arabic label level instruction	LIB_NIV_ARA	VarChar(30)
Arabic label position	LIB_POS_ARA	VarChar(20)
Arabic label project	LIB_PRJ_ARA	VarChar(80)

Data element name	Field name	Code format
Arabic label rank	LIB_GRA_ARA	VarChar(20)
Arabic label specialty	LIB_SPC_ARA	VarChar(30)
Arabic label sub unit	LIB_SUN_ARA	VarChar(20)
Arabic label unit	LIB_UNI_ARA	VarChar(30)
Arabic last name person	PRN_PER_ARA	VarChar(20)
Arabic name young girl	NJF_PER_ARA	VarChar(20)
Arabic place of birth	LIE_NAI_ARA	VarChar(20)
Area	MIC_ZON	VarChar(30)
Assigned Note	NOT_OBT	Decimal(5.2)
Average note of the half year	NOT_MOY	Decimal(5.2)
Bank account number	NUM_CMP_PER	VarChar(20)
Bank of organization education	BNQ_FOR	VarChar(30)
Calculate component (O, N)	COM_CAL	VarChar(1)
Card number	NUM_FCH	VarChar(20)
Code action	COD_ACT	VarChar(2)
Code bank	COD_BNQ	VarChar(4)
Code budgetary card	FIC_COD_FBU	VarChar(4)
Code component	COD_COM	VarChar(4)
Code criteria	COD_CRI	VarChar(2)
Code diploma	COD_DIP	VarChar(3)
Code function	COD_FON	VarChar(3)
Code insurer	COD_ASS	VarChar(2)
Code level of instruction	COD_NIV_INS	VarChar(3)
Code position (01,02,03,04,05,06)	COD_POS	VarChar(3)
Code project	COD_PRJ	VarChar(2)
Code rule of calculation	COD_REG_CAL	VarChar(10)
Code sanction	COD_SAN	VarChar(2)
Code specialty	COD_SPC	VarChar(3)
Code sub unit	COD_SUN	VarChar(2)
Code theme	COD_THE	VarChar(4)
Code unit	COD_UNI	VarChar(3)
Concerned people	POP_CON	VarChar(40)
Contract number for person under instruction	NUM_CON_STA	Integer
Contract number or decision	NUM_CNT_DEC	VarChar(10)
Cost of education	COU_FOR	Decimal(12.3)
Daily rate	TAU_JOU_MOI	Integer
Date affiliation insurer	DAT_AFF_ASS	Date

Data element name	Field name	Code format
Date card	DAT_FCH	Date
Date contract or decision	DAT_CNT_DEC	Date
Date contract stagier	DAT_CON_STA	Date
Date of the decision	DAT_DEC	Date
Date death child	DAT_DCE	Date
Date decision end carrier	DAT_DEC_FCA	Date
Date decision function	DAT_DEC_FON	Date
Date decision position	DAT_DEC_POS	Date
Date decision sanction	DAT_DEC_DIS	Date
Date decision stage	DAT_DEC_STA	Date
Date delivery CIN	DAT_DEL_CIN	Date
Date departure person	DAT_DEP_PER	Date
Date driver license	DAT_PER_CON	Date
Date effect category	DAT_EFF_CAT	Date
Date effect level	DAT_EFF_NIV	Date
Date effect new position	DAT_EFF_POS	Date
Date end contract	DAT_FIN_CNT	Date
Date end detachment	DAT_FIN_DET	Date
Date end enumeration	DAT_FIN_REN	Date
Date end education	DAT_FIN_FOR	Date
Date end function	DAT_FIN_FON	Date
Date end position	DAT_FIN_POS	Date
Date end sanction	DAT_FIN_MIP	Date
Date end vacation	DAT_RET	Date
Date of Birth child	DAT_NAI	Date
Date of birth person	DAT_NAI_PER	Date
Date profit of a car	DAT_AVA_VOI	Date
Date profit of the lodging	DAT_AVA_LOG	Date
Date Recruitment	DAT_REC	Date
Date request resignation	DAT_DEM_DMS	Date
Date start detachment	DAT_DEB_DET	Date
Date start education	DAT_DEB_FOR	Date
Date start function	DAT_DEB_FON	Date
Date start position	DAT_DEB_POS	Date
Date start sanction	DAT_DEB_MIP	Date
Date start vacation	DAT_SOR	Date
Date of permanent status	DAT_TIT	Date

Data element name	Field name	Code format
Decision end stage	DEC_FIN_STA	VarChar(40)
Decision number detachment	NUM_DEC_DET	VarChar(40)
Decision number end carrier	NUM_DEC_FCA	VarChar(40)
Degree of the sanction (1,2)	DEG_SAN	Integer
Detachment organization	ORG_DET	VarChar(40)
Duration of the session	DUR_SES	Integer
Echelon	ECH_PER	Integer
Efficient salary	MNT_BRUT	Decimal(12.3)
Family situation person (C,M,D,V)	SIT_FAM_PER	VarChar(1)
First name stagier	PRN_STA	VarChar(20)
Education duration	DUR_FOR	Integer
Education expenses	FRA_FOR	Decimal(12.3)
Education result (C,N)	RES_FOR	VarChar(1)
Education type (I,T,E)	TYP_FOR	VarChar(10)
French action label	LIB_ACT_FRA	VarChar(30)
French address person	ADR_PER_FRA	VarChar(80)
French code category	COD_CAT_FRA	VarChar(40)
French first name of person	NOM_PER_FRA	VarChar(20)
French label bank	LIB_BNQ_FRA	VarChar(30)
French label component	LIB_COM_FRA	VarChar(40)
French label diploma	LIB_DIP_FRA	VarChar(30)
French label function	LIBE_FON_FRA	VarChar(40)
French label insurer	LIB_ASS_FRA	VarChar(40)
French label level of instruction	LIB_NIV_FRA	VarChar(40)
French label position	LIB_POS_FRA	VarChar(20)
French label project	LIB_PRJ_FRA	VarChar(80)
French label rank	LIB_GRA_FRA	VarChar(30)
French Label sanction	LIB_SAC	VarChar(30)
French label specialty	LIB_SPC_FRA	VarChar(40)
French label sub unit	LIB_SUN_FRA	VarChar(30)
French Label theme	LIB_THE	VarChar(30)
French label unit	LIB_UNI_FRA	VarChar(30)
French last name of person	PRN_PER_FRA	VarChar(20)
French name young girl	NJF_PER_FRA	VarChar(30)
French place of birth	LIE_NAI_FRA	VarChar(20)
Generated amount	MNT_GEN	Decimal(12.3)
ID card	ID_CARD	VarChar(12)

Data element name	Field name	Code format
Instruction type	TYP_ANI	VarChar(40)
Insurer type (M,C)	TYP_ASS	VarChar(1)
Maximum number of hours	NBR_HER_MAX	Integer
Method of payment (C,B)	MOD_PAY_PER	VarChar(1)
Military situation (A,E,S)	SIT_MIL_PER	VarChar(1)
Month advance	MOI_AVC	VarChar(10)
Month end reimbursement	MOI_FIN_REM	VarChar(10)
Month pay	MOI_PAY	VarChar(10)
Month start reimbursement	MOI_DEB_REM	VarChar(10)
Name child	PRN_ENF	VarChar(40)
Name person under instruction	NOM_STA	VarChar(40)
Nature education (Q,D)	NAT_FOR	VarChar(1)
Nature of departure (1,2,3,4,5)	NAT_DEP	VarChar(1)
Net salary	MNT_NET	Decimal(12.3)
Performance preview type	TYP_NOT	VarChar(1)
Number affiliation insurer	NUM_AFF_ASS	VarChar(15)
Number Budgetary card	FIC_ANN_FBU	VarChar(20)
Number child	NUM_ENF	Integer
Number decision function	NUM_DEC_FON	VarChar(10)
Number decision position	NUM_DEC_POS	VarChar(10)
Number decision resignation	NUM_DEC_DMS	VarChar(10)
Number decision sanction	NUM_DEC_DIS	VarChar(10)
Number driver license	NUM_PER_CON	VarChar(10)
Number half year	NUM_SEM	Integer
Number of child affiliated	NBR_ENF_OUV	Integer
Number of children	NBR_ENF_REE	Integer
Number of children IRPP	NBR_ENF_AFF	Integer
Number of day absence	NBR_JOU	Integer
Number of day of sanction	NBR_JOU_MIP	Integer
Number of hour	NBR_HER	Integer
Number of the quarter (1,2)	NUM_TRI_PRI	Char(10)
Number of the session	NBR_SES	Integer
Number of vacation days	NBJ_CNG	Integer
Number of vacation request	NUM_CNG	Integer
Number of working days	NBR_JOU_TRA	Integer
Observation decision	OBS_DEC	VarChar(50)
Observation detachment	OBS_DET	VarChar(50)

Data element name	Field name	Code format
Observation education	OBS_FOR	VarChar(50)
Opinion of responsible	AVI_HIE	VarChar(50)
Other expenses	AUT_FRA	Decimal(12.3)
Person number	MAT_PER	Integer
Person to contact	PER_CON_FOR	VarChar(40)
Phone number	TEL_PER	VarChar(12)
Picture	PHO_PER	PIC
Place of education	LIE_FOR	VarChar(30)
Plan reference	REF_PLF	VarChar(40)
Profit of a function car (O,N)	BEN_VOI	VarChar(1)
Profit of lodging (O,N)	BEN_LOG	VarChar(1)
Proposed amount	MNT_PRO	Decimal(12.3)
Proposition code project	N_P_COD_PRJ	VarChar(1)
Rank	COD_GRA	VarChar(5)
Realized amount	MNT_REA	Decimal(12.3)
Reason of refusal	MOT_REF	VarChar(50)
Reason of resignation	MOT_DEM	VarChar(50)
Reason of the sanction	MOT_SAN	VarChar(50)
Reduction CNR (O, N)	RET_CNR	VarChar(1)
Reduction for SINDIC (O, N)	RET_SIN	VarChar(1)
Reduction from absence (O, N)	RET_ABS	VarChar(1)
Reduction from salary (O, N)	RET_SAL	VarChar(1)
Reduction insurance (O, N)	RET_ASS	Decimal(12.3)
Reference decision period of instruction	REF_DEC_STA	VarChar(40)
Reference instructor	REF_FOR	VarChar(40)
Residue vacation	REL_CON_PER	Integer
Sex of person (M,F)	SEX_PER	VarChar(1)
Sex person under instruction (M,F)	SEX_STA	VarChar(1)
Social situation	RAI_FOR	VarChar(1)
SSN Card number	NUM_CAR_SEC	VarChar(12)
Supplementary rate of hours	TAU_HOR_HSU	Integer
Taxable Amount	MNT_IMP	VarChar(1)
Total day of work	TOT_JOU	Integer
Type absence (J,N)	TYP_ABS	VarChar(1)
Type advance (A1,A2,S,AO)	TYP_AVC	VarChar(2)
Type card (R,D)	TYP_FCH	VarChar(1)
Type component (+ = More and – = Less)	TYP_COM	VarChar(1)

Data element name	Field name	Code format
Type of detachment (A,D)	TYP_DET	VarChar(1)
Type decision (A,S,P,F)	TYP_DEC	VarChar(1)
Type engagement (PA,PR,FD,AV,TR)	TYP_ENG	VarChar(1)
Type of bonus (T,S)	TYP_PRI	VarChar(1)
Type operation (G,M)	TYPE_OPR_COM	VarChar(1)
Type reimbursement (E,D)	TYP_REM	VarChar(1)
Under 21 years old (O,N)	ENF_CHA	VarChar(1)
Unique salary (O,N)	SAL_UNI_PER	VarChar(1)
Vacation address	ADR_PER_CON	VarChar(50)
Vacation type (R,E,C,L,M,S)	TYP_CNG	VarChar(1)
Way of payment	MOD_REG	VarChar(1)
Year advance	ANN_AVC	VarChar(4)
Year bonus	ANN_PRI	VarChar(4)
Year budgetary card	ANN_FBU	VarChar(4)
Year card retirement or death	ANN_FCH	VarChar(4)
Year contract	ANN_CON_STA	VarChar(4)
Year decision of detachment	ANN_DEC_DET	VarChar(4)
Year decision resignation	ANN_DEC_DMS	VarChar(4)
Year decision sanction	ANN_DEC_DIS	VarChar(4)
Year end reimbursement	ANN_FIN_REM	VarChar(4)
Year education	ANN_FOR	VarChar(4)
Year note	ANN_NOT	VarChar(4)
Year of vacation	ANN_CNG	VarChar(4)
Year pay	ANN_PAY	VarChar(4)
Year plan	ANN_PLF	VarChar(4)
Year proposition	ANN_PRO	VarChar(4)
Year proposition promotion	PRO_ANN_PRO	VarChar(4)
Year residue vacation	ANN_REL_CON	VarChar(4)
Year start reimbursement	ANN_DEB_REM	VarChar(4)

Table 7. Data Element Definition

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. DATABASE SQL SCRIPT

This appendix contains the source code for the implementation of the database prototype created as a proof of concept for the thesis. Although the prototype is a fully functional application, it does not consist of a finished program for distribution. The script is presented in an "as-is" format, intended only as a foundation for future work.

```
create TABLE "informix".n_diplom
  (    cod_dip char(3) not null constraint "informix".n191_335,
    lib_dip_ara char(40),
    lib_dip_fra char(40),
    primary key (cod_dip) constraint "informix".pk_n_diplom  );
create TABLE "informix".n_positi
  (    cod_pos char(3) not null constraint "informix".n193_339,
    lib_pos_ara char(40),
    lib_pos_fra char(40),
    check (cod_pos IN ('01' ,'02' ,'03' ,'04' ,'05' ,'06' )) constraint "informix".ckc_cod_pos_,
    primary key (cod_pos) constraint "informix".pk_n_positi  );
create TABLE "informix".n_specia
  (    cod_spc char(3) not null constraint "informix".n195_344,
    lib_spc_ara char(40),
    lib_spc_fra char(40),
    primary key (cod_spc) constraint "informix".pk_n_specia  );
create TABLE "informix".n_assure
  (    cod_ass char(2) not null constraint "informix".n196_346,
    lib_ass_ara char(20),
    lib_ass_fra char(20),
    plf_ass decimal(12,3),
    primary key (cod_ass) constraint "informix".pk_n_assure  );
create TABLE "informix".critnot
  (    cod_cri char(2) not null constraint "informix".n198_353,
    lib_cri_ara char(30),
    pou_not integer,
    primary key (cod_cri) constraint "informix".pk_critnot  );
create TABLE "informix".categorie
  (    cod_cat_ara char(2) not null constraint "informix".n202_366,
    cod_cat_fra char(2),
    primary key (cod_cat_ara) constraint “informix”.pk_categorie  );
create TABLE “informix”.n_axe
  (    num_axe char(5) not null constraint “informix”.n204_370,
    lib_axe char(60),
    primary key (num_axe) constraint “informix”.pk_n_axe  );
create TABLE “informix”.n_theme
  (    cod_the char(4) not null constraint “informix”.n205_372,
    lib_the char(40),
    primary key (cod_the) constraint “informix”.pk_theme  );
create TABLE “informix”.n_fonction
  (    cod_fon char(3) not null constraint “informix”.n208_380,
    libe_fon_ara char(40),
    libe_fon_fra char(40),
    primary key (cod_fon) constraint “informix”.pk_fonction  );
create TABLE “informix”.n_bank
  (    cod_bnq char(4) not null constraint “informix”.n209_382,
    lib_bnq_ara char(30),
    lib_bnq_fra char(30),
    primary key (cod_bnq) constraint “informix”.pk_banque  );
create TABLE “informix”.prod_bonus
  (    ann_pri integer not null constraint “informix”.n242_517,
    typ_pri char(1) not null constraint “informix”.n200_358,
    num_tri_pri integer not null constraint “informix”.n243_518,
    check (typ_pri IN ('T' ,'S' )) constraint “informix”.ckc_typ_pri_,
    primary key (ann_pri,typ_pri,num_tri_pri) constraint “informix”.pk_primrend  );
create TABLE “informix”.year_month_pay
  (    ann_pay char(4) not null constraint “informix”.n257_535,
    moi_pay char(2) not null constraint “informix”.n257_536,
    primary key (ann_pay,moi_pay) constraint “informix”.pk_annmoipai  );
create TABLE “informix”.resignation
  (    ann_dec_dms char(4) not null constraint “informix”.n262_541,
    num_dec_dms integer not null constraint “informix”.n222_451,
    mat_per char(5) not null constraint “informix”.n222_452,
    dat_dem_dms date,
    mot_dem char(30),
    dat_dep date,
    primary key (ann_dec_dms,num_dec_dms) constraint “informix”.pk_demission  );
create TABLE “informix”.note_rec
  (    ann_not char(4) not null constraint “informix”.n266_548,
    num_sem char(10) not null constraint “informix”.n217_424,
    mat_per char(5) not null constraint “informix”.n217_426,
    not_moy decimal(5,3),
    obs_not char(40),
    typ_not char(1),
    primary key (ann_not,num_sem,mat_per) constraint “informix”.pk_fichnot  );
create TABLE “informix”.suplhours
  (    mat_per char(5) not null constraint “informix”.n242_552,
    ann_pay char(4) not null constraint “informix”.n268_550,
    moi_pay char(2) not null constraint “informix”.n268_551,
    nbr_her decimal(5,3),
    primary key (mat_per,ann_pay,moi_pay) constraint “informix”.pk_heursup  );
create TABLE “informix”.comphist
  (    cod_com char(4) not null constraint “informix”.n240_542,
    ann_pay char(4) not null constraint “informix”.n269_552,
    moi_pay char(2) not null constraint “informix”.n269_553,
    cod_reg_cal char(10) not null constraint “informix”.n240_545,
    primary key (cod_com,ann_pay,moi_pay) constraint “informix”.pk_histcomp  );
create TABLE “informix”.pay_jour
  (    ann_pay char(4) not null constraint “informix”.n273_559,
    moi_pay char(2) not null constraint “informix”.n273_560,
    mat_ouv char(5) not null constraint “informix”.n247_585,
    nbr_jou_tra decimal(5,3),
    tau_jou_moi decimal(12,3),
    mnt_brut decimal(12,3),
    mnt_cnss decimal(12,3),
    mnt_imp decimal(12,3),
    mnt_irpp decimal(12,3),
    mnt_avc_moi decimal(12,3),
    mnt_net decimal(12,3),
    primary key (ann_pay,moi_pay,mat_ouv) constraint “informix”.pk_joupaiper  );
create TABLE “informix”.prodbonusemp
  (    mat_per char(5) not null constraint “informix”.n236_520,
    ann_pri char(4) not null constraint “informix”.n276_565,
    typ_pri char(1) not null constraint “informix”.n236_522,
    num_tri_pri decimal(1) not null constraint “informix”.n236_524,
    not_obt_pri decimal(5,3),
    nbr_abs decimal(6,3),
    mnt_ava decimal(12,3),
    mnt_pri decimal(12,3),
    check (typ_pri IN ('T' ,'S' )) constraint “informix”.ckc_typ_pri1,
    primary key (mat_per,ann_pri,typ_pri,num_tri_pri) constraint “informix”.pk_ligprim  );
create TABLE “informix”.sanction
  (    ann_dec_dis char(4) not null constraint “informix”.n284_573,
    num_dec_dis integer not null constraint “informix”.n223_455,
    mat_per char(5) not null constraint “informix”.n223_456,
    cod_san char(2) not null constraint “informix”.n223_457,
    dat_dec_dis date,
    mot_san char(255),
    nbr_jou_mip integer,
    dat_deb_mip date,
    dat_fin_mip date,
    primary key (ann_dec_dis,num_dec_dis) constraint “informix”.pk_sanction  );
create TABLE “informix”.educ_session
  (    ann_sea char(4) not null constraint “informix”.n285_574,
    num_sea integer not null constraint “informix”.n220_439,
    cod_the char(4) not null constraint “informix”.n220_440,
    ann_for char(4) not null constraint “informix”.n285_575,
    num_for integer not null constraint “informix”.n220_442,
    dat_sea date,
    her_deb_sea date,
    her_fin_sea date,
    fra_res decimal(12,3),
    aut_fra decimal(12,3),
    primary key (ann_sea,num_sea) constraint “informix”.pk_seance  );
create TABLE “informix”.detache
  (    ann_dec_det char(4) not null constraint “informix”.n293_589,
    num_dec_det integer not null constraint “informix”.n293_590,
    mat_per char(5) not null constraint “informix”.n293_591,
    dat_dec_det date,
    typ_det char(1),
    dat_deb_det date,
    dat_fin_det date,
    org_det char(50),
    obs_det char(255)  );
create unique index "informix".pk_detach on "informix".detache (ann_dec_det,num_dec_det);
create TABLE "informix".absence
  (    mat_per char(5) not null constraint "informix".n241_547,
    ann_pay char(4) not null constraint "informix".n241_548,
    moi_pay char(2) not null constraint "informix".n241_549,
    nbr_jou integer not null constraint "informix".n302_624,
    abs_jus char(1),
    primary key (mat_per,ann_pay,moi_pay) constraint "informix".pk_absence  );
create TABLE "informix".n_sanction
  (    cod_san char(2) not null constraint "informix".n304_628,
    lib_san char(50) not null constraint "informix".n304_629,
    primary key (cod_san) constraint "informix".pk_san  );
create TABLE "informix".promotion
  (    ann_dec_pro char(4) not null constraint "informix".n282_570,
    num_dec_pro integer not null constraint "informix".n226_471,
    mat_per char(5) not null constraint "informix".n307_632,
    cod_gra char(4) not null constraint "informix".n226_472,
    cod_niv char(2) not null constraint "informix".n226_473,
    cod_cat char(2) not null constraint "informix".n309_633,
    cod_fon char(3),
    typ_dec char(1),
    dat_dec_pro date,
    anc_cat char(2),
    anc_gra char(4),
    anc_niv char(2),
    anc_fon char(3),
    dat_eff_pos date,
    obs_dec char(255),
    primary key (ann_dec_pro,num_dec_pro) constraint "informix".pk_promotion  );
create TABLE "informix".assignotate
  (    cod_cri char(2) not null constraint "informix".n330_752,
    ann_not char(4) not null constraint "informix".n330_753,
    num_sem char(1) not null constraint "informix".n330_754,
    not_obt decimal(5,3) not null constraint “informix”.n330_755,
    mat_per char(5) not null constraint “informix”.n330_756,
    primary key (cod_cri,ann_not,num_sem,mat_per) constraint “informix”.pk_lignote  );
create TABLE “informix”.orgeducation
  (    cod_org char(10) not null constraint “informix”.n377_911,
    nom_org char(60),
    adr_org char(60),
    act_org char(60),
    tel_org char(15),
    fax_org char(15),
    adr_ele char(20),
    num_pat char(15),
    dat_pat date,
    per_con char(50),
    primary key (cod_org) constraint “informix”.orgformprimarykey1   );
create TABLE “informix”.per diem
  (    ann_pay char(4) not null constraint “informix”.n472_1251,
    moi_pay char(2) not null constraint “informix”.n472_1252,
    mat_per char(5) not null constraint “informix”.n472_1253,
    nbr_tie integer not null constraint “informix”.n472_1254,
    nbr_jou integer,
    typ_dep char(1) not null constraint “informix”.n472_1255,
    mnt_dep decimal(12,3) not null constraint “informix”.n472_1256,
    mnt_nui decimal(12,3),
    dep_pay char(1),
    primary key (ann_pay,moi_pay,mat_per) constraint “informix”.pk_deplaceme  );
create TABLE “informix”.n_rank
  (    cod_gra char(4) not null constraint “informix”.n483_1309,
    cod_cat char(2) not null constraint “informix”.n483_1310,
    lib_gra_ara char(40),
    lib_gra_fra char(40),
    primary key (cod_gra) constraint “informix”.pk_n_grade  );
create TABLE “informix”.compfonc
  (    cod_fon char(3) not null constraint “informix”.n513_1387,
    cod_com char(4) not null constraint “informix”.n513_1388,
    mnt_com decimal(12,3),
    pc_hpc char(1),
  primary key (cod_fon,cod_com,pc_hpc) constraint “informix”.pk_compfonc  );
create TABLE “informix”.comfoncad
 (    cod_fon char(3) not null constraint “informix”.n517_1403,
  cod_cad char(1),
  cod_com char(4) not null constraint “informix”.n517_1404,
  mnt_com decimal(12,3),
  primary key (cod_fon,cod_com,cod_cad) constraint “informix”.comfoncadpk );
create TABLE “informix”.compdg
 (    cod_com char(4) not null constraint “informix”.n520_1410,
  mnt_com decimal(12,3),
  primary key (cod_com) constraint “informix”.compdgpk  );
create TABLE “informix”.compper
 (    mat_per char(5) not null constraint “informix”.n535_1445,
  cod_com char(4) not null constraint “informix”.n535_1446,
  nat_com char(1) not null constraint “informix”.n535_1587,
  mnt_com decimal(12,3),
  ann_deb_com char(4),
  moi_deb_com char(2),
  ann_fin_com char(4),
  moi_fin_com char(2),
  type_opr_com char(1),
  com_aff char(1),
  mnt_glob decimal(12,3),
  nbr_tranch integer,
  com_fix char(1),
  primary key (mat_per,cod_com,nat_com,ann_deb_com,moi_deb_com) constraint
“informix”.ct_compper1  );
create TABLE “informix”.eduplan
 (    ref_plf char(10) not null constraint “informix”.n549_1475,
  num_axe char(5),
  ann_plf char(4) not null constraint “informix”.n549_1476,
  act_plf char(4),
  obj_plf char(255),
  pop_con char(1),
  typ_for char(1),
  typ_ani char(255),
  dur_ses integer,
  nbr_ses integer,
  per_ses char(100),
  obs_for char(255),
  primary key (ref_plf,ann_plf) constraint “informix”.pk_ eduplan);
create TABLE “informix”.child
 (    num_enf integer not null constraint “informix”.n285_896,
  mat_per char(5) not null constraint “informix”.n227_479,
  prn_enf char(25),
  dat_nai date,
  dat_dce date,
  enf_cha char(1),
  enf_irp char(1),
  enf_etd char(1)  );
create unique index “informix”.pk_ child on “informix”. child (mat_per,num_enf);
create TABLE “informix”.compos
 (    cod_com char(4) not null constraint “informix”.n197_348,
  nat_com char(3),
  lib_com_ara char(40),
  lib_com_fra char(40),
  typ_com char(1),
  ret_sal char(1),
  ret_cnr char(1),
  ret_ass char(1),
  ret_abs char(1),
  par_emp char(1),
  ret_sin char(1),
  num_cmp_deb char(10),
  num_cmp_cre char(10),
  num_cnr char(4),
  com_cal char(1),
  reg_cal char(10),
  primary key (cod_com) constraint “informix”.pk_compos  );
create TABLE “informix”.n_natcomp
 (    nat_com char(2) not null constraint “informix”.n582_1538,
  lib_nat_ara char(30),
  lib_nat_fra char(30)  );
create unique cluster index “informix”.ix_n_natcomp1 on “informix”.n_natcomp (nat_com);
create TABLE “informix”.personn
 (    mat_per char(5) not null constraint “informix”.n205_519,
  cod_niv_ins char(3),
  cod_dip char(3),
  cod_gra char(4),
  cod_pos char(2),
  cod_prj_aff char(4),
  spc_per char(30),
  cod_prj_pay char(4),
  cod_uni char(4),
  cod_fon char(3),
  cod_bnq char(4),
  cod_bnq_dom char(4),
  num_cmp_bnq char(20),
  nom_per_ara char(25),
  nom_per_fra char(25),
  prn_per_ara char(25),
  prn_per_fra char(25),
  pho_per char(255),
  njf_per_ara char(25),
  njf_per_fra char(25),
  dat_nai_per date,
  lie_nai_ara char(15),
  lie_nai_fra char(15),
  sex_per char(1)        default 'M',
  adr_per_ara char(255),
  adr_per_fra char(150),
  tel_per char(10),
  num_per_con char(12),
  dat_per_con date,
  sit_mil_per char(1),
  sit_fam_per char(1),
  sal_uni_per char(1)        default 'N',
  chf_fam char(1),
  sta_per char(1),
  pc_hpc_per char(1),
  dat_rec date,
  num_cnt_dec char(10),
  dat_cnt_dec date,
  dat_tit date,
  dat_eff_cat date,
  mnt_bas decimal(13,3),
  ech_per char(2),
  num_dec_fon char(10),
  dat_dec_fon date,
  dat_deb_fon date,
  dat_fin_fon date,
  dat_dec_pos date,
  dat_deb_pos date,
  tit_per char(1),
  mod_pay_per char(1),
  num_cmp_per char(20),
  num_car_sec char(15),
  dat_aff_sec date,
  typ_ass char(1),
  num_aff_ass char(15),
  dat_aff_ass date,
  ben_voi char(1),
  dat_ava_voi date,
  ben_log char(1),
  dat_ava_log date,
  nbr_enf_aff integer,
  primary key (mat_per) constraint "informix".pk_personne  );
create TABLE "informix".avance_per
 (    mat_per char(5) not null constraint "informix".n237_526,
  ann_avc char(4) not null constraint "informix".n258_537,
  num_avc char(15) not null constraint "informix".n652_1633,
  mnt_avc decimal(12,3),
  cod_cl_cai char(4),
  num_mvtc integer,
  cod_cai char(3),
  primary key (mat_per,ann_avc,num_avc) constraint "informix".ct_avance_per1  );
create TABLE "informix".avances
 (    ann_avc char(4) not null constraint "informix".n655_1641,
  num_avc char(15),
  dat_avc date,
  cod_avc char(4) not null constraint “informix”.n655_1642,
  mnt_avc float,
  mat_per char(5),
  typ_rem char(1),
  nbr_rem float,
  ann_deb char(4),
  ann_fin char(4),
  moi_deb char(2),
  moi_fin char(2),
  ben_avc char(1) not null constraint “informix”.n655_1643,
  primary key (ann_avc,dat_avc) constraint “informix”.pk_avances  );
create TABLE “informix”.usr_userinfo
 (    matricule char(3),
  user_name char(20),
  password char(15),
  primary key (matricule) constraint “informix”.ct_usr_userinfo2  );
create TABLE “informix”.usr_module
 (    cod_module char(2),
  nom_module char(20),
  primary key (cod_module) constraint “informix”.ct_usr_module2  );
create TABLE “informix”.usr_droit
 (    matricule char(3),
  cod_module char(2),
  droit char(4)  );
create unique index “informix”.ix_usr_droit1 on “informix”.usr_droit (cod_module,matricule);
create TABLE “informix”.param_edit
 (    mois_edit char(2) not null constraint “informix”.n678_1698,
  an_edit char(4) not null constraint “informix”.n678_1699  );
create unique index “informix”.ix_param_edit on “informix”.param_edit (mois_edit,an_edit);
create TABLE “informix”.education
 (    ann_for char(4) not null constraint “informix”.n703_1778,
  num_for integer not null constraint “informix”.n703_1779,
  ann_plf char(4) not null constraint “informix”.n703_1780,
  ref_plf char(10) not null constraint “informix”.n703_1781,
  dat_for date,
  dur_for integer,
  lie_for char(30),
  nat_for char(1),
  nbr_prev integer,
  nbr_real integer,
  cod_org char(10),
  nom_for char(60),
  fra_trs decimal(12,3),
  ori_trs char(3),
  fra_dep decimal(12,3),
  ori_dep char(3),
  fra_log decimal(12,3),
  ori_log char(3),
  fra_nou decimal(12,3),
  ori_nou char(3),
  fra_stg decimal(12,3),
  ori_stg char(3),
  fra_mis decimal(12,3),
  ori_mis char(3),
  fra_ped decimal(12,3),
  ori_ped char(3),
  fra_ass decimal(12,3),
  ori_ass char(3),
  aut_fra decimal(12,3),
  ori_aut char(3),
  primary key (ann_for,num_for) constraint “informix”.pk_ education);
```
END SCRIPT

APPENDIX C. USE CASE NARRATIVE

Use Case ID:	**1**		
Use Case Name:	**Identification and authentication**		
Created By:	Noureddine Trigui	Last Updated By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Last Updated	July 30,2004
Actors:	Human resource, accounting, finance, budgetary, and provisioning services, market and transit section, patrimonial section		
Description:	Every user of the system must be identified in order to limit the access granted to this user. The identification and authentication is provided by a user login and password stored in the system database.		
Preconditions:	1. User is registered (as central site user, regional site user, and External site) for ISYDMA system.		
Post conditions:	1. User authenticated and role applied 1.a User not logged in 2. Action logged		
Normal Flow:	**1.0 Identification and authentication** 1. User chooses the connect form. 2. System displays login form. 3. User enters use login. 4. User enters PASSWORD. 5. System verifies input data with the stored data. 6. System updates log file to keep track of the action. 7. System displays available menu for the specified user.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available. 2a. User cancels request. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.2 Check user Identification information** (at step 3,4) 1. System informs user that his login and PASSWORD does not match. 2. User enters his identification information again. **1.0.E.3 Can't find information** (at step 6) 1. System informs user that the database is not available now and information are not found.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User shall be able to cancel the Identification request at any time prior to confirming the request.		

Use Case ID:	**2**		
Use Case Name:	**Create Pay Component**		
Created By:	Noureddine Trigui	Last Updated By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Last Updated	July 30,2004
Actors:	Pay section		
Description:	This use case describes the events of pay section for establishing a new pay component, that will serve as a key part for calculation of employee payment.		
Preconditions:	1. User is registered as a user from the pay section		
Post conditions:	1. User authenticated and role applied 1.a User not logged in 2. Action logged		
Normal Flow:	**1.0 Create pay component** 1. User chooses the pay component form. 2. System displays pay component form. 3. User chooses add button. 4. System verifies authorization for the user to add component. 5. User enters information related to the specified component. 6. System controls information and add them to the database.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available. 2a. User cancels request. 2b. System terminates use case. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.2 Check user Identification information** (at step 4) 1. System informs user that he is not allowed to add pay component. 2. User chooses another button. **1.0.E.3 Can't Store information** (at step 6) 1. System informs user that the database is not available now and information is not stored.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User shall be able to cancel the Identification request at any time prior to confirming the request.		

Use Case ID:	**3**		
Use Case Name:	**Create employee record**		
Created By:	Noureddine Trigui	Last Updated By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Last Updated	July 30,2004
Actors:	User from personnel and training section		
Description:	This use case describes the events of personnel and training section for creating an employee record which is similar for the three category of employee (Statutory employee, Workman employee, agriculture employee). The Employee record is composed of several parts. The first part concerns the identification information, second part concerns the administrative information, and the last part concerns the family information.		
Preconditions:	1. User is registered as a user from the personnel or training section		
Post conditions:	1. User authenticated and role applied 1.a User not logged in 2. Action logged		
Normal Flow:	**1.0 Create employee record** 1. User chooses the employee record form. 2. System displays the chosen form. 3. User chooses add button. 4. System verifies authorization for the user to add employee record. 5. User enters information related to the identification part. 6. System controls information and adds them to the database. 7. User chooses to continue entering the administrative, family, civil information. 8. System opens form related to the user choice. 9. User enters information. 10. System validates information and stores them in the database.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available. 2a. User cancels request. 2b. System terminates use case. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.2 Check user Identification information** (at step 4) 1. System informs User that he is not allowed to do this operation. 2. User chooses another action. **1.0.E.3 Can't Store information** (at step 10) 1. System informs User that the database is not available now and information is not stored.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User from the personnel and training section shall be able to cancel the Identification request at any time prior to confirming		

	the request.		
Use Case ID:	**4**		
Use Case Name:	**Create employee attendance record**		
Created By:	Noureddine Trigui	Last Updated By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Last Updated	July 30,2004
Actors:	User from the personnel section		
Description:	This use case describes the events of personnel section for creating a record related to the absence of an employee. Absence can be classified as justified or not justified.		
Preconditions:	1. User is registered as a user from the personnel section allowed to create the information related to the employee absence.		
Post conditions:	1. User authenticated and role applied 1.a User not logged in 2. Action logged		
Normal Flow:	**1.0 Create employee attendance record** 1. User chooses the employee attendance form. 2. System displays the chosen form. 3. User chooses add button. 4. System verifies authorization for the user to add attendance information. 5. User enters information related to the days of absence. 6. System controls information and adds them to the database in order to be taken in consideration when calculating the employee pay.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available. 2a. User cancels request. 2b. System terminates use case. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.2 Check user Identification information** (at step 4) 1. System informs user that he is not allowed to do this operation. 2. User chooses another action. **1.0.E.3 Can't Store information** (at step 6) 1. System informs user that the database is not available now and information is not stored.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User from the personnel section shall be able to cancel the request at any time prior to confirming the request.		

Use Case ID:	**5**		
Use Case Name:	**Create employee leave process**		
Created By:	Noureddine Trigui	Created By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Created:	July 30,2004
Actors:	User from personnel section		
Description:	This use case describes the events of personnel section for creating a record for the vacation of an employee. There are many type of vacation: Normal vacation, exceptional vacation, sickness vacation, vacation without salary, vacation for new born.		
Preconditions:	1. User is registered as a user from the personnel section allowed to create the information related to the employee vacation.		
Post conditions:	1. User authenticated and role applied 1.a User not logged in 2. Action logged		
Normal Flow:	**1.0 Create employee vacations record** 1. User chooses the employee leave process form. 2. System displays the chosen form. 3. User chooses add button. 4. System verifies authorization for the user to add vacation information. 5. User enters information related to type of vacation and the number of vacation days in order to be taken in consideration when calculating the pay. 6. System controls information and adds them to the database.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available. 2a. User cancels request. 2b. System terminates use case. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.2 Check user Identification information** (at step 4) 1. System informs user that he is not allowed to create information related to the employee vacations. 2. User chooses another action. **1.0.E.3 Can't Store information** (at step 6) 1. System informs user that the database is not available now and information is not stored.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User from the personnel section shall be able to cancel the request at any time prior to confirming the request.		

Use Case ID:	**6**		
Use Case Name:	**Create employee traveling record**		
Created By:	Noureddine Trigui	Created By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Created:	July 30,2004
Actors:	User from personnel and training section		
Description:	This use case describes the events of personnel section for creating a record containing the necessary information for the payment of the traveling realized by an employee during his actual month.		
Preconditions:	1. User is registered as a user from the personnel or training section allowed to update the information related to the employee traveling.		
Post conditions:	1. User authenticated and role applied 1.a User not logged in 2. Action logged		
Normal Flow:	**1.0 Create employee traveling record** 1. User chooses the employee traveling form. 2. System displays the chosen form. 3. User chooses the add button. 4. System verifies authorization for the user to add traveling information. 5. User enters information related to the traveling. 6. System controls information and adds the to the database.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available. 2a. User cancels request. 2b. System terminates use case. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.2 Check user Identification information** (at step 4) 1. System informs user that he is not allowed to create information related to the employee traveling record. 2. User chooses another action. **1.0.E.3 Can't Store information** (at step 6) 1. System informs user that the database is not available now and information is not stored.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User from the personnel or training section shall be able to cancel the request at any time prior to confirming the request.		

Use Case ID:	**7**		
Use Case Name:	**Update employee supplementary hours record**		
Created By:	Noureddine Trigui	Created By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Created:	July 30,2004
Actors:	User from personnel and training section		
Description:	This use case describes the events of personnel section for updating a record for the supplementary hour record. Every employee has a maximum number of supplementary hours by month. The amount of money allowed to one supplementary hours depend on the rank and the category of the employee.		
Preconditions:	1. User is registered as a user from the personnel or training section allowed to update the information related to the employee traveling.		
Post conditions:	1. User authenticated and role applied 1.a User not logged in 2. Action logged		
Normal Flow:	**1.0 Update employee supplementary hours record** 1. User chooses the employee supplementary hours form. 2. System displays the chosen form. 3. User chooses the select button. 4. System displays the criteria of selection form. 5. User enters information related to the record to be updated. 6. User pushes the execute search button. 7. System displays information that satisfies the criteria specified. 8. User starts updating information. 9. System controls the entered information. 10. User presses the update button. 11. System verifies authorization for the user to update information. 12. System controls information and updates the database.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available. 2a. User cancels request. 2b. System terminates use case. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.2 Check user Identification information** (at step 11) 1. System informs user that he is not allowed to update information related to the employee supplementary hours. 2. User chooses another action. **1.0.E.3 Can't Store information** (at step 12) 1. System informs user that the database is not available now and information is not stored.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User from the personnel or training section shall be able to cancel the request at any time prior to confirming the request.		

Use Case ID:	**8**		
Use Case Name:	**Update employee production bonus record**		
Created By:	Noureddine Trigui	Created By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Created:	July 30,2004
Actors:	User from personnel and training section		
Description:	This use case describes the events of personnel or training section updating the information related to the production bonuses. The production bonuses are calculated once every 3 months.		
Preconditions:	1. User is registered as a user from the personnel or training section allowed to update the information related to the employee traveling.		
Post conditions:	1. User authenticated and role applied 1.a User not logged in 2. Action logged		
Normal Flow:	**1.0 Update employee production bonus record** 1. User chooses the employee production bonus form. 2. System displays the chosen form. 3. User chooses the select button. 4. System displays the criteria of selection form. 5. User enters information related to the record to be updated. 6. User pushes the execute search button. 7. System displays information that satisfies the criteria specified. 8. User starts updating information. 9. System controls the entered information. 10. User presses the update button. 11. System verifies authorization for the user to update information. 12. System controls information and updates the database.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available. 2a. User cancels request. 2b. System terminates use case. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.2 Check user Identification information** (at step 11) 1. System informs user that he is not allowed to update information related to the employee production bonus. 2. User chooses another action. **1.0.E.3 Can't Store information** (at step 12) 1. System informs user that the database is not available now and information is not stored.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User from the personnel or training section shall be able to cancel the request at any time prior to confirming the request.		

Use Case ID:	**9**		
Use Case Name:	**Update employee detach record**		
Created By:	Noureddine Trigui	Created By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Created:	July 30,2004
Actors:	User from personnel section		
Description:	This use case describes the events of personnel section for updating a record related to the employee-detach record. The detach can be in another organization or in another department of the same organization.		
Preconditions:	1. User is registered as a user from the personnel section allowed to update the information related to the employee detach.		
Post conditions:	1. User authenticated and role applied 1.a User not logged in 2. Action logged		
Normal Flow:	**1.0 Update employee detach record** 1. User chooses the employee detach form. 2. System displays the chosen form. 3. User chooses the select button. 4. System displays the criteria of selection form. 5. User enters information related to the record to be updated. 6. User pushes the execute search button. 7. System displays information that satisfy the criteria specified. 8. User starts updating information. 9. System controls the entered information. 10. User presses the update button. 11. System verifies authorization for the user to update information. 12. System controls information and update the database.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available. 2a. User cancels request. 2b. System terminates use case. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.2 Check user Identification information** (at step 11) 1. System informs user that he is not allowed to update information related to the employee. 2. User chooses another action. **1.0.E.3 Can't Store information** (at step 12) 1. System informs user that the database is not available now and information is not stored.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User from the personnel section shall be able to cancel the request at any time prior to confirming the request.		

Use Case ID:	**10**		
Use Case Name:	**Search employee resignation record**		
Created By:	Noureddine Trigui	Created By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Created:	July 30,2004
Actors:	User from personnel and training section.		
Description:	This use case describes the events of personnel section for searching a record related to resignation of an employee.		
Preconditions:	1. User is registered as a user from the personnel or training section allowed to search the information related to the employee traveling.		
Post conditions:	1. User authenticated and role applied 1.a User not logged in 2. Action logged		
Normal Flow:	**1.0 Search employee resignation record** 1. User chooses the employee resignation form. 2. System displays the chosen form. 3. User chooses the select button. 4. System displays the criteria of selection form. 5. User enters information related to the record to be selected. 6. User pushes the execute search button. 7. System displays information that satisfies the criteria specified.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available. 2a. User cancels request. 2b. System terminates use case. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.3 Can't Find information** (at step 7) 1. System informs user that the database is not available now and information is not stored.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User from the personnel section shall be able to cancel the request at any time prior to confirming the request.		

Use Case ID:	**11**		
Use Case Name:	**Search employee disciplinary sanction record**		
Created By:	Noureddine Trigui	Created By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Created:	July 30,2004
Actors:	User from personnel and training section		
Description:	This use case describes the events of personnel section for searching a record related to the disciplinary acts.		
Preconditions:	1. User is registered as allowed to search the information related to the employee disciplinary sanction.		
Post conditions:	1. User authenticated and role applied 1.a User not logged in 2. Action logged		
Normal Flow:	**1.0 Search employee disciplinary sanction record** 1. User chooses the employee disciplinary form. 2. System displays the chosen form. 3. User chooses the select button. 4. System displays the criteria of selection form. 5. User enters information related to the record to be selected. 6. User pushes the execute search button. 7. System displays information that satisfy the criteria specified.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available 2a. User cancels request. 2b. System terminates use case. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.3 Can't find information** (at step 7) 1. System informs user that the database is not available now and information is not stored.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User from the personnel section shall be able to cancel the request at any time prior to confirming the request.		

Use Case ID:	**12**		
Use Case Name:	**Search employee promotion record**		
Created By:	Noureddine Trigui	Created By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Created:	July 30,2004
Actors:	User from personnel and training section		
Description:	This use case describes the events of personnel section for searching a record related to the promotion of an employee. An employee promotion must satisfy some conditions. There are two type of promotion: rank promotion and category promotion. Each one has an effect on the pay components.		
Preconditions:	1. User is registered as allowed to search the information related to the employee promotion.		
Post conditions:	1. User authenticated and role applied 1.a User not logged in 2. Action logged		
Normal Flow:	**1.0 Search employee promotion record** 1. User chooses the employee disciplinary form. 2. System displays the chosen form. 3. User chooses the select button. 4. System displays the criteria of selection form. 5. User enters information related to the record to be selected. 6. User pushes the execute search button. 7. System displays information that satisfies the criteria specified.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available. 2a. User cancels request. 2b. System terminates use case. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.3 Can't find information** (at step 7) 1. System informs user that the database is not available now and information is not stored.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User from the personnel section shall be able to cancel the request at any time prior to confirming the request.		

Use Case ID:	**13**		
Use Case Name:	**Search employee retirement or death record**		
Created By:	Noureddine Trigui	Created By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Created:	July 30,2004
Actors:	User from personnel and training section		
Description:	This use case describes the events of personnel section for searching a record for the retirement and a record when the employee is dead. Those records have an effect on the pay component.		
Preconditions:	1. User is registered as allowed to search the information related to the employee retirement or death.		
Post conditions:	1. User authenticated and role applied. 1.a User not logged in. 2. Action logged.		
Normal Flow:	**1.0 Search employee retirement or death record** 1. User chooses the employee disciplinary form. 2. System displays the chosen form. 3. User chooses the select button. 4. System displays the criteria of selection form. 5. User enters information related to the record to be selected. 6. User pushes the execute search button. 7. System displays information that satisfies the criteria specified.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available. 2a. User cancels request. 2b. System terminates use case. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.3 Can't find information** (at step 7) 1. System informs user that the database is not available now and information is not stored.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User from the personnel section shall be able to cancel the request at any time prior to confirming the request.		

Use Case ID:	**14**		
Use Case Name:	**Delete employee educational record**		
Created By:	Noureddine Trigui	Created By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Created:	July 30,2004
Actors:	User from personnel and training section		
Description:	This use case describes the events of personnel section for deleting a record containing the different training section that an employee had attended. The training section can be organized in an internal department or external organization.		
Preconditions:	1. User is registered as allowed to delete the information related to the employee educational record.		
Post conditions:	1. User authenticated and role applied. 1.a User not logged in. 2. Action logged.		
Normal Flow:	**1.0 Delete employee educational record** 1. User chooses the employee education form. 2. System displays the chosen form. 3. User chooses the select button. 4. System displays the criteria of selection form. 5. User enters information related to the record to be deleted. 6. User pushes the execute search button. 7. System displays information that satisfies the criteria specified. 8. User presses the delete button. 9. System verifies authorization for the user to delete information. 10. System controls information and updates the database.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available. 2a. User cancels request. 2b. System terminates use case. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.2 Check user Identification information** (at step 9) 1. System informs user that he is not allowed to delete information related to the employee. 2. User chooses another action. **1.0.E.3 Can't Store information** (at step 10) 1. System informs user that the database is not available now and information is not stored.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User from the personnel section shall be able to cancel the request at any time prior to confirming the request.		

Use Case ID:	**15**		
Use Case Name:	**Process Pay calculation**		
Created By:	Noureddine Trigui	Created By:	Noureddine Trigui
Date Created:	March 1, 2004	Date Created:	July 30,2004
Actors:	User from personnel and training section		
Description:	This use case describes the events of personnel section for creating a record for grouping all the component and necessary information to calculate pay salary of an employee. The pay calculation is executed every month, and should be started by the second week of the actual month.		
Preconditions:	1. User is registered as allowed to create the information related to the employee pay calculation.		
Post conditions:	1. User authenticated and role applied 1.a User not logged in 2. Action logged		
Normal Flow:	**1.0 Process Pay calculation** 1. User chooses the process pay calculation form. 2. System displays the chosen form. 3. User chooses create button. 4. System verifies authorization for the user to process the pay calculation. 5. User enters information related to the pay calculation. 6. System controls information and adds them to the database.		
Exceptions:	**1.0.E.1 Option System is not available now** (at step 1) 1. System informs user that this option is not available. 2a. User cancels request. 2b. System terminates use case. 3a. User requests to select another option. 3b. System restarts use case. **1.0.E.2 Check user Identification information** (at step 4) 1. System informs user that he is not allowed to process pay calculation. 2. User chooses another action. **1.0.E.3 Can't Store information** (at step 6) 1. System informs user that the database is not available now and information is not stored.		
Includes:	None		
Priority:	High		
Special Requirements:	1. User from the personnel section shall be able to cancel the request at any time prior to confirming the request.		

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. HUMAN RESOURCE MANAGEMENT SOURCE CODE

```
//#DV1xC00005
unit maingrh;
interface
uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, Menus, connect,ExtCtrls, DBCtrls,
  StdCtrls,  Buttons, DB, DBtables, EiniFile,Quickrpt,
  Grids, DBGrids, DdeMan, Entite, ToolWin, ComCtrls, Tabs, OleCtrls,
  qrextra;
  //Crystal_TLB;
type
    Tfrm_maingrh = class(TForm)
    Printsetup: TPrintDialog;
    Texport: TTable;
    BatchMove: TBatchMove;
    BarreEtat: TPanel;
    Bevel2: TBevel;
    Bevel4: TBevel;
    nbr_rech: TLabel;
    DataL: TDatabase;
    Qrech: TQuery;
    Qupdate: TQuery;
    Qupdatemax: TIntegerField;
    QcountDetail: TQuery;
    QMaxNO: TQuery;
    DSMaxNO: TDataSource;
    bib_DBNavigator: TDBNavigator;
    bib_DBNavigator3: TDBNavigator;
    bib_DBNavigator2: TDBNavigator;
    bib_DBNavigator1: TDBNavigator;
    ToolBar1: TToolBar;
    Panel1: TPanel;
    Label2: TLabel;
    Panel2: TPanel;
    Qseltable: TQuery;
    BTNnouv: TSpeedButton;
    BTNsupp: TSpeedButton;
    BTNannul: TSpeedButton;
    BiB_construire: TSpeedButton;
    BiB_vider: TSpeedButton;
    BiB_ecran: TSpeedButton;
    BiB_annul: TSpeedButton;
    BTNsais: TSpeedButton;
    BTNlist: TSpeedButton;
    BTNsel: TSpeedButton;
    Panel4: TPanel;
    Panel5: TPanel;
    DSFPROJET: TDataSource;
    FPROJET: TTable;
    FPROJETcode_prj: TStringField;
    FPROJETdesi_prj: TStringField;
    FPROJETcode_dev: TStringField;
    FPROJETcode_bai: TStringField;
    FPROJETnume_pre: TStringField;
    FPROJETmont_pre: TFloatField;
    FPROJETdate_sig: TDateTimeField;
    FPROJETdate_val: TDateTimeField;
    FPROJETdate_clo: TDateTimeField;
    BTN_SEPremier: TSpeedButton;
    BTN_SEPrecedent: TSpeedButton;
    BTN_SESuivant: TSpeedButton;
    BTN_SEDernier: TSpeedButton;
    Panel8: TPanel;
    FAGENCE: TTable;
    DSFAGENCE: TDataSource;
    DataD: TDatabase;
    PB: TProgressBar;
    Label1: TLabel;
    EditNOM_UTI: TLabel;
    DBgrh: TDBprogramme;
    Panel3: TPanel;
    MainMenu1: TMainMenu;
    mnu_Fichier: TMenuItem;
    mnu_FSConnecter: TMenuItem;
    N3: TMenuItem;
    mnu_FConfiguration: TMenuItem;
    mnu_FApercu: TMenuItem;
    mnu_FImprimer: TMenuItem;
    N7: TMenuItem;
    mnu_FQuitter: TMenuItem;
     mnu_Edition: TMenuItem;
    mnu_ECouper: TMenuItem;
    mnu_ECopier: TMenuItem;
    mnu_EColler: TMenuItem;
    mnu_Aide: TMenuItem;
    mnu_AIndex: TMenuItem;
    mnu_ARech: TMenuItem;
    N6: TMenuItem;
    mnu_Apropos: TMenuItem;
    SPmntlet_dep: TStoredProc;
    SPmntlet_ordre: TStoredProc;
    BTNmodf: TSpeedButton;
    BTNvalid: TSpeedButton;
//#FV1xC00012
    procedure bib_transfererClick(Sender: TObject);
    procedure BiB_viderClick(Sender: TObject);
    procedure mnu_FSConnecterClick(Sender: TObject);
    procedure mnu_FSDeconnecterClick(Sender: TObject);
    procedure mnu_FConfigurationClick(Sender: TObject);
    procedure mnu_FQuitterClick(Sender: TObject);
    procedure mnu_ECopierClick(Sender: TObject);
    procedure mnu_ECollerClick(Sender: TObject);
    procedure mnu_ECouperClick(Sender: TObject);
    procedure mnu_AIndexClick(Sender: TObject);
    procedure mnu_ARechClick(Sender: TObject);
    procedure mnu_AproposClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
//#FV2xC00013
    procedure activerConnect(enable : boolean);
    procedure activerEdition(enable : boolean);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure BiB_construireClick(Sender: TObject);
    procedure bib_DBNavigatorClick(Sender: TObject; Button: TNavigateBtn);
    procedure BTNnouvClick(Sender: TObject);
    procedure BTNsuppClick(Sender: TObject);
    procedure BTNmodfClick(Sender: TObject);
    procedure BTNvalidClick(Sender: TObject);
    procedure BTNannulClick(Sender: TObject);
    procedure BTN_SEPremierClick(Sender: TObject);
    procedure BTN_SEPrecedentClick(Sender: TObject);
    procedure BTN_SESuivantClick(Sender: TObject);
    procedure BTN_SEDernierClick(Sender: TObject);
    procedure BTNcouperClick(Sender: TObject);
    procedure BTNcopierClick(Sender: TObject);
    procedure BTNcollerClick(Sender: TObject);
    procedure BTNsaisClick(Sender: TObject);
    procedure BTNlistClick(Sender: TObject);
    procedure BTNselClick(Sender: TObject);
    procedure BiB_ecranClick(Sender: TObject);
  private
    FReport : TQuickRep;
    procedure SetReport(Value : TQuickRep);
  public
    property Report : TQuickRep read FReport write SetReport;
  end;
const
//
C_PERSONNE  =1;
C_COMPOS    =2;
C_ENFANT    =3;
C_DETACHE   =4;
C_DEMISSION =5;
C_RTRDEC    =6;
C_SANCTION  =7;
C_PROMOTION =8;
C_OUVPERM   =9;
C_OUVCHANT  =10;
C_ORGFORM   =11;
C_CATGRAD     =12;
C_CATEGNIV    =13;
C_COMPPER     =14;
C_ABSENCE     =15;
C_CONGES      =16;
C_DEPLACEMENT =17;
C_HEURSUP     =18;
C_FICHNOT     =19;
C_LIGNOTE     =20;
C_CATCAD      =21;
C_CATFON      =22;
C_COMPFONC    =23;
C_comcatcad   =24;
```

```
C_comgra      =25;
C_PLANFORM    =26;
C_COMFONCAD   =27;
C_compdg      =28;
C_FORMATION   =29;
C_REMFRAMED   =30;
C_inscription =31;
C_AVANCES     =32;
C_MVTMOIS     =33;
C_COMPOUV     =34;
C_PRESCHANT   =35;
C_CONGESOUV   =36;
//
var
//#DV1xC00030
frm_maingrh              :        Tfrm_maingrh;
//#FV1xC00030
// lien
implementation
 uses grille,about,exportto,navgrh,divgrh,
//#DV2xC00050
  PERSONNE, COMPOS,  CATEGNIV, PLANFORM, ORGFORM, OUVPERM,
  REMFRAMED, CONGES, DEPLACEMENT, HEURSUP, COMPPER, FICHNOT, ABSENCE,
  FORMATION, AVANCES, PRIMREND, DEMISSION, RTRDEC, SANCTION, PROMOTION,
  DETACHE,            MVTMOIS,            PRESCHANT,            OUVCHANT,            SEANCE,
PRESFORM,COMPOUV,CONGESOUV,
  ENFANT, GENERPAIE ,CATGRAD, CATCAD, CATFON, COMPFONC, LIGNOTE, COMCATCAD,
  COMGRA, COMFONCAD, COMPDG, INSCRIPTION, FNOMENC, EDITMENS, generpouv,
  sel_avancable, sel_conges, sel_fichnot, sel_notper, sel_dectriouv,
  generdisk;

//#FV1xC00060

{$R *.DFM}

//#DV1xC00070
{menu fichier}
procedure Tfrm_maingrh.SetReport(Value : TQuickRep);
begin
  FReport:=Value;
end;
procedure Tfrm_maingrh.mnu_FSConnecterClick(Sender: TObject);
begin
  Application.CreateForm(Tfrm_connect,frm_connect);
  frm_connect.ShowModal;
end;
procedure Tfrm_maingrh.mnu_FSDeconnecterClick(Sender: TObject);
begin
  frm_maingrh.dataL.connected:=false;
  //BTNraccourci(0);
  activerConnect(False);
end;
procedure Tfrm_maingrh.mnu_FConfigurationClick(Sender: TObject);
begin
printsetup.execute;
```

```
end;
procedure Tfrm_maingrh.mnu_FQuitterClick(Sender: TObject);
begin
  application.terminate;
end;
//#FV1xC00070
//#DV1xC00080
{menu Edition}
procedure Tfrm_maingrh.mnu_ECopierClick(Sender: TObject);
begin
if ActiveMdiChild.activecontrol is TDBEdit then
  TDBEdit(ActiveMdiChild.activecontrol).CopyToClipboard;
if ActiveMdiChild.activecontrol is TEdit then
  TEdit(ActiveMdiChild.activecontrol).CopyToClipboard;
end;
procedure Tfrm_maingrh.mnu_ECollerClick(Sender: TObject);
begin
if ActiveMdiChild.activecontrol is TDBEdit then
  TDBEdit(ActiveMdiChild.activecontrol).Pastefromclipboard;
if ActiveMdiChild.activecontrol is TEdit then
  TEdit(ActiveMdiChild.activecontrol).PastefromClipboard;
end;
procedure Tfrm_maingrh.mnu_ECouperClick(Sender: TObject);
begin
if ActiveMdiChild.activecontrol is TDBEdit then
   TDBEdit(ActiveMdiChild.activecontrol).cuttoclipboard;
if ActiveMdiChild.activecontrol is TEdit then
   TEdit(ActiveMdiChild.activecontrol).cuttoclipboard;
end;
//#FV1xC00080
//#DV1xC00090
{menu Selection}
procedure Tfrm_maingrh.bib_transfererClick(Sender: TObject);
begin
  if bib_DBNavigator2.datasource<>Nil then
  bib_DBNavigator2.BtnClick(Nbrefresh);
end;
procedure Tfrm_maingrh.BiB_viderClick(Sender: TObject);
begin
  clear_rech;
end;
//#FV1xC00100

//#DV1xC00110
{menu Help}
procedure Tfrm_maingrh.mnu_AIndexClick(Sender: TObject);
begin
Application.HelpCommand(HELP_CONTENTS,0);
end;
procedure Tfrm_maingrh.mnu_ARechClick(Sender: TObject);
begin
Application.helpcommand(HELP_HELPONHELP,0);
end;
procedure Tfrm_maingrh.mnu_AproposClick(Sender: TObject);
begin
frm_about.showmodal
```

```
end;

//#DV1xC00130
procedure Tfrm_maingrh.activerEdition(enable : boolean);
begin
with frm_maingrh do
begin
      mnu_Ecouper.enabled:=enable;
      mnu_Ecoller.enabled:=enable;
      mnu_Ecopier.enabled:=enable;
end;
end;
procedure Tfrm_maingrh.activerConnect(enable : boolean);
begin
with frm_maingrh do
begin
     mnu_FSConnecter.enabled:=not enable;
     mnu_Fimprimer.enabled:=enable;
     mnu_Fapercu.enabled:=enable;
     mnu_Edition.visible:=enable;
end;
end;
procedure Tfrm_maingrh.FormCreate(Sender: TObject);
var v_jour, v_mois, v_annee:word;
begin
  Lire_ini;
  application.BiDiKeyboard:='00001c01';
  application.nonBiDiKeyboard:='0000040c';
  glb_cod_com_bas:='0001';
  frm_maingrh.DBgrh.DerRech := 0;
  frm_maingrh.DBgrh.DerForm := 0;
  frm_maingrh.DBgrh.Derlisting:=0;
  Res_rech:=false;
  nu_table:=1;
  decodedate(date, v_annee, v_mois, v_jour);
  Glb_annee:=inttostr(v_annee);
  if v_mois<10
  then Glb_mois:='0'+inttostr(v_mois)
  else Glb_mois:=inttostr(v_mois);
  Glb_jour:=inttostr(v_jour);
  glb_date:=datetostr(date);
  Barre_etat;
end;
procedure Tfrm_maingrh.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  frm_maingrh.dataL.connected:=false;
  frm_maingrh.dataD.connected:=false;
  activerConnect(False);
end;
procedure Tfrm_maingrh.BiB_construireClick(Sender: TObject);
begin
  if frm_maingrh.DBgrh.DerForm > 0 then
  construire;
  begin
    if frm_maingrh.nbr_rech.Caption='0' then
    MessageDlg('No record can respond to the criteria! ',
           mtInformation, [mbYes], 0);
    naviguer;
    if frm_maingrh.nbr_rech.Caption='1' then BTNsaisClick(BTNsais)
    else
    if frm_maingrh.nbr_rech.Caption>'1' then BTNlistClick(BTNlist)
    else
    BTNselClick(BTNsel);
  end;
end;
procedure Tfrm_maingrh.bib_DBNavigatorClick(Sender: TObject;
  Button: TNavigateBtn);
begin
case Button of
  nbfirst :begin
          BTN_SEPremier.enabled:=false;
          BTN_SEPrecedent.enabled:=false;
          BTN_SESuivant.enabled:=true;
          BTN_SEDernier.enabled:=true;
          end;
   nblast :begin
          BTN_SEPremier.enabled:=true;
          BTN_SEPrecedent.enabled:=true;
          BTN_SESuivant.enabled:=false;
          BTN_SEDernier.enabled:=false;
          end;
   nbNext :begin
          BTN_SEPremier.enabled:=true;
          BTN_SEPrecedent.enabled:=true;
          BTN_SESuivant.enabled:=true;
          BTN_SEDernier.enabled:=true;
          end;
   nbPrior :begin
          BTN_SEPremier.enabled:=true;
          BTN_SEPrecedent.enabled:=true;
          BTN_SESuivant.enabled:=true;
          BTN_SEDernier.enabled:=true;
          end;
end;
naviguer;
end;

procedure Tfrm_maingrh.BTNnouvClick(Sender: TObject);
begin
flag_insert:=0;
if bib_dbnavigator1.datasource<>Nil then
begin
bib_dbnavigator1.BtnClick(Nbinsert);
end;
end;
procedure Tfrm_maingrh.BTNsuppClick(Sender: TObject);
begin
if bib_DBNavigator2.datasource<>Nil then
begin
bib_DBNavigator2.BtnClick(Nbdelete);
end;
```

```
end;
procedure Tfrm_maingrh.BTNmodfClick(Sender: TObject);
begin
flag_insert:=0;
if bib_DBNavigator3.datasource<>Nil then
begin
bib_DBNavigator3.BtnClick(Nbedit);
end;
end;
procedure Tfrm_maingrh.BTNvalidClick(Sender: TObject);
begin
 if bib_DBNavigator3.datasource<>Nil then
 if (bib_DBNavigator3.datasource.dataset.state=dsInsert) or
   (bib_DBNavigator3.datasource.dataset.state=dsEdit) then
    bib_DBNavigator3.BtnClick(Nbpost);
end;

procedure Tfrm_maingrh.BTNannulClick(Sender: TObject);
begin
if bib_DBNavigator3.datasource<>Nil then
 if (bib_DBNavigator3.datasource.dataset.state=dsInsert) or
   (bib_DBNavigator3.datasource.dataset.state=dsEdit) then
   bib_DBNavigator3.BtnClick(Nbcancel);

end;

procedure Tfrm_maingrh.BTN_SEPremierClick(Sender: TObject);
begin
if bib_dbnavigator.datasource<>Nil then
begin
bib_dbnavigator.BtnClick(Nbfirst);
BTN_SEPrecedent.enabled:=false;
BTN_SEPremier.enabled:=false;
end;
end;

procedure Tfrm_maingrh.BTN_SEPrecedentClick(Sender: TObject);
begin
  if bib_dbnavigator.datasource<>Nil then
  begin
    bib_dbnavigator.BtnClick(Nbprior);
    if bib_dbnavigator.datasource.dataset.bof then
    begin
      BTN_SEPrecedent.enabled:=false;
      BTN_SEPremier.enabled:=false;
      BTN_SESuivant.enabled:=true;
      BTN_SEDernier.enabled:=true;
    end;
  end;
end;

procedure Tfrm_maingrh.BTN_SESuivantClick(Sender: TObject);
begin
  if bib_dbnavigator.datasource<>Nil then
  begin
```

```
    bib_dbnavigator.BtnClick(Nbnext);
    if bib_dbnavigator.datasource.dataset.eof then
    begin
      BTN_SEPremier.enabled:=true;
      BTN_SEPrecedent.enabled:=true;
      BTN_SESuivant.enabled:=false;
      BTN_SEDernier.enabled:=false;
    end;
  end;
end;

procedure Tfrm_maingrh.BTN_SEDernierClick(Sender: TObject);
begin
  if bib_dbnavigator.datasource<>Nil then
  begin
    bib_dbnavigator.BtnClick(Nblast);
    BTN_SEPremier.enabled:=true;
    BTN_SEPrecedent.enabled:=true;
    BTN_SESuivant.enabled:=false;
    BTN_SEDernier.enabled:=false;
  end;
end;

procedure Tfrm_maingrh.BTNcouperClick(Sender: TObject);
begin
  if ActiveMdiChild.activecontrol is TDBEdit then
     TDBEdit(ActiveMdiChild.activecontrol).cuttoclipboard;
  if ActiveMdiChild.activecontrol is TEdit then
     TEdit(ActiveMdiChild.activecontrol).cuttoclipboard;

     end;

procedure Tfrm_maingrh.BTNcopierClick(Sender: TObject);
begin
if ActiveMdiChild.activecontrol is TDBEdit then
  TDBEdit(ActiveMdiChild.activecontrol).CopyToClipboard;
if ActiveMdiChild.activecontrol is TEdit then
  TEdit(ActiveMdiChild.activecontrol).CopyToClipboard;

end;

procedure Tfrm_maingrh.BTNcollerClick(Sender: TObject);
begin
if ActiveMdiChild.activecontrol is TDBEdit then
  TDBEdit(ActiveMdiChild.activecontrol).Pastefromclipboard;
if ActiveMdiChild.activecontrol is TEdit then
  TEdit(ActiveMdiChild.activecontrol).PastefromClipboard;

end;

procedure Tfrm_maingrh.BTNsaisClick(Sender: TObject);
begin

  with frm_maingrh do
  begin
     DBgrh.DerRech:= DBgrh.DerForm;
    case DBgrh.DerRech of
      C_PERSONNE    : frm_PERSONNE.NBPERSONNE.PageIndex      :=0;
      C_COMPOS      : frm_COMPOS.NBCOMPOS.PageIndex          :=0;
      C_ENFANT      : frm_ENFANT.NBENFANT.PageIndex          :=0;
      C_DETACHE     : frm_DETACHE.NBDETACHE.PageIndex        :=0;
      C_DEMISSION   : frm_DEMISSION.NBDEMISSION.PageIndex    :=0;
      C_CATEGNIV    : frm_CATEGNIV.NBCATEGNIV.PageIndex      :=0;
      C_RTRDEC      : frm_RTRDEC.NBRTRDEC.PageIndex          :=0;
      C_SANCTION    : frm_SANCTION.NBSANCTION.PageIndex      :=0;
      C_PROMOTION   : frm_PROMOTION.NBPROMOTION.PageIndex    :=0;
      C_OUVPERM     : frm_OUVPERM.NBOUVPERM.PageIndex        :=0;
      C_OUVCHANT    : frm_OUVCHANT.NBOUVCHANT.PageIndex      :=0;
      C_ORGFORM     : frm_ORGFORM.NBORGFORM.PageIndex        :=0;
      C_CATGRAD     : frm_CATGRAD.NBCATGRAD.PageIndex        :=0;
      C_COMPPER     : frm_COMPPER.NBCOMPPER.PageIndex        :=0;
      C_ABSENCE     : frm_ABSENCE.NBABSENCE.PageIndex        :=0;
      C_CONGES      : frm_CONGES.NBCONGES.PageIndex          :=0;
      C_DEPLACEMENT  : frm_DEPLACEMENT.NBDEPLACEMENT.PageIndex:=0;
      C_HEURSUP     : frm_HEURSUP.NBHEURSUP.PageIndex        :=0;
      C_FICHNOT     : frm_FICHNOT.NBFICHNOT.PageIndex        :=0;
      C_CATCAD      : frm_CATCAD.NBCATCAD.PageIndex          :=0;
      C_CATFON      : frm_CATFON.NBCATFON.PageIndex          :=0;
      C_COMPFONC    : frm_COMPFONC.NBCOMPFONC.PageIndex      :=0;
      C_LIGNOTE     : frm_LIGNOTE.NBLIGNOTE.PageIndex        :=0;
      C_COMCATCAD   : frm_COMCATCAD.NBCOMCATCAD.PageIndex    :=0;
      C_COMGRA      : frm_COMGRA.NBCOMGRA.PageIndex          :=0;
      C_COMFONCAD   : frm_COMFONCAD.NBCOMFONCAD.PageIndex    :=0;
      C_COMPDG      : frm_COMPDG.NBCOMPDG.PageIndex          :=0;
      C_FORMATION   : frm_FORMATION.NBFORMATION.PageIndex    :=0;
      C_REMFRAMED   : frm_REMFRAMED.NBREMFRAMED.PageIndex    :=0;
      C_INSCRIPTION  : frm_INSCRIPTION.NBINSCRIPTION.PageIndex:=0;
      C_PLANFORM    : frm_PLANFORM.NBPLANFORM.PageIndex      :=0;
      C_AVANCES     : frm_AVANCES.NBAVANCES.PageIndex        :=0;
      C_MVTMOIS     : frm_MVTMOIS.NBMVTMOIS.PageIndex        :=0;
      C_COMPouv     : frm_COMPOUV.NBCOMPOUV.PageIndex        :=0;
      C_PRESCHANT   : frm_PRESCHANT.NBPRESCHANT.PageIndex    :=0;
      C_CONGESOUV   : frm_CONGESOUV.NBCONGESOUV.PageIndex    :=0;

    end;

    active_desactive_bouton;
  end;
  BTNsais.Flat:=false;
  BTNlist.Flat:=true;
  BTNsel.Flat:=true;

end;

procedure Tfrm_maingrh.BTNlistClick(Sender: TObject);
begin
  with frm_maingrh do
  begin
    DBgrh.DerRech:= DBgrh.DerForm;
    case DBgrh.DerRech of
      C_PERSONNE    : frm_PERSONNE.NBPERSONNE.PageIndex      :=1;
      C_COMPOS      : frm_COMPOS.NBCOMPOS.PageIndex          :=1;
```

```
        C_ENFANT      : frm_ENFANT.NBENFANT.PageIndex          :=1;
        C_DETACHE     : frm_DETACHE.NBDETACHE.PageIndex        :=1;
        C_DEMISSION   : frm_DEMISSION.NBDEMISSION.PageIndex    :=1;
        C_CATEGNIV    : frm_CATEGNIV.NBCATEGNIV.PageIndex      :=1;
        C_RTRDEC      : frm_RTRDEC.NBRTRDEC.PageIndex          :=1;
        C_SANCTION    : frm_SANCTION.NBSANCTION.PageIndex      :=1;
        C_PROMOTION   : frm_PROMOTION.NBPROMOTION.PageIndex    :=1;
        C_OUVPERM     : frm_OUVPERM.NBOUVPERM.PageIndex        :=1;
        C_OUVCHANT    : frm_OUVCHANT.NBOUVCHANT.PageIndex      :=1;
        C_ORGFORM     : frm_ORGFORM.NBORGFORM.PageIndex        :=1;
        C_CATGRAD     : frm_CATGRAD.NBCATGRAD.PageIndex        :=1;
        C_COMPPER     : frm_COMPPER.NBCOMPPER.PageIndex        :=1;
        C_ABSENCE     : frm_ABSENCE.NBABSENCE.PageIndex        :=1;
        C_CONGES      : frm_CONGES.NBCONGES.PageIndex          :=1;
        C_DEPLACEMENT : frm_DEPLACEMENT.NBDEPLACEMENT.PageIndex:=1;
        C_HEURSUP     : frm_HEURSUP.NBHEURSUP.PageIndex        :=1;
        C_FICHNOT     : frm_FICHNOT.NBFICHNOT.PageIndex        :=1;
        C_CATCAD      : frm_CATCAD.NBCATCAD.PageIndex          :=1;
        C_CATFON      : frm_CATFON.NBCATFON.PageIndex          :=1;
        C_COMPFONC    : frm_COMPFONC.NBCOMPFONC.PageIndex      :=1;
        C_LIGNOTE     : frm_LIGNOTE.NBLIGNOTE.PageIndex        :=1;
        C_COMCATCAD   : frm_COMCATCAD.NBCOMCATCAD.PageIndex    :=1;
        C_COMGRA      : frm_COMGRA.NBCOMGRA.PageIndex          :=1;
        C_COMFONCAD   : frm_COMFONCAD.NBCOMFONCAD.PageIndex    :=1;
        C_COMPDG      : frm_COMPDG.NBCOMPDG.PageIndex          :=1;
        C_FORMATION   : frm_FORMATION.NBFORMATION.PageIndex    :=1;
        C_REMFRAMED   : frm_REMFRAMED.NBREMFRAMED.PageIndex    :=1;
        C_INSCRIPTION   : frm_INSCRIPTION.NBINSCRIPTION.PageIndex:=1;
        C_PLANFORM    : frm_PLANFORM.NBPLANFORM.PageIndex      :=1;
        C_AVANCES     : frm_AVANCES.NBAVANCES.PageIndex        :=1;
        C_MVTMOIS     : frm_MVTMOIS.NBMVTMOIS.PageIndex        :=1;
        C_COMPouv     : frm_COMPOUV.NBCOMPOUV.PageIndex        :=1;
        C_PRESCHANT   : frm_PRESCHANT.NBPRESCHANT.PageIndex    :=1;
        C_CONGESOUV   : frm_CONGESOUV.NBCONGESOUV.PageIndex    :=1;
      end;
      BTN_maj(0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1);
      end;
   BTNsais.Flat:=true;
   BTNlist.Flat:=false;
   BTNsel.Flat:=true;

end;

procedure Tfrm_maingrh.BTNselClick(Sender: TObject);
begin
   with frm_maingrh do
   begin
      DBgrh.DerRech:= DBgrh.DerForm;
      case DBgrh.DerRech of
        C_PERSONNE    : frm_PERSONNE.NBPERSONNE.PageIndex      :=2;
        C_COMPOS      : frm_COMPOS.NBCOMPOS.PageIndex          :=2;
        C_ENFANT      : frm_ENFANT.NBENFANT.PageIndex          :=2;
        C_DETACHE     : frm_DETACHE.NBDETACHE.PageIndex        :=2;
        C_DEMISSION   : frm_DEMISSION.NBDEMISSION.PageIndex    :=2;
        C_CATEGNIV    : frm_CATEGNIV.NBCATEGNIV.PageIndex      :=2;
        C_RTRDEC      : frm_RTRDEC.NBRTRDEC.PageIndex          :=2;
        C_SANCTION    : frm_SANCTION.NBSANCTION.PageIndex      :=2;
        C_PROMOTION   : frm_PROMOTION.NBPROMOTION.PageIndex    :=2;
        C_OUVPERM     : frm_OUVPERM.NBOUVPERM.PageIndex        :=2;
        C_OUVCHANT    : frm_OUVCHANT.NBOUVCHANT.PageIndex      :=2;
        C_ORGFORM     : frm_ORGFORM.NBORGFORM.PageIndex        :=2;
        C_CATGRAD     : frm_CATGRAD.NBCATGRAD.PageIndex        :=2;
        C_COMPPER     : frm_COMPPER.NBCOMPPER.PageIndex        :=2;
        C_ABSENCE     : frm_ABSENCE.NBABSENCE.PageIndex        :=2;
        C_CONGES      : frm_CONGES.NBCONGES.PageIndex          :=2;
        C_DEPLACEMENT : frm_DEPLACEMENT.NBDEPLACEMENT.PageIndex:=2;
        C_HEURSUP     : frm_HEURSUP.NBHEURSUP.PageIndex        :=2;
        C_FICHNOT     : frm_FICHNOT.NBFICHNOT.PageIndex        :=2;
        C_CATCAD      : frm_CATCAD.NBCATCAD.PageIndex          :=2;
        C_CATFON      : frm_CATFON.NBCATFON.PageIndex          :=2;
        C_COMPFONC    : frm_COMPFONC.NBCOMPFONC.PageIndex      :=2;
        C_LIGNOTE     : frm_LIGNOTE.NBLIGNOTE.PageIndex        :=2;
        C_COMCATCAD   : frm_COMCATCAD.NBCOMCATCAD.PageIndex    :=2;
        C_COMGRA      : frm_COMGRA.NBCOMGRA.PageIndex          :=2;
        C_COMFONCAD   : frm_COMFONCAD.NBCOMFONCAD.PageIndex    :=2;
        C_COMPDG      : frm_COMPDG.NBCOMPDG.PageIndex          :=2;
        C_FORMATION   : frm_FORMATION.NBFORMATION.PageIndex    :=2;
        C_REMFRAMED   : frm_REMFRAMED.NBREMFRAMED.PageIndex    :=2;
        C_INSCRIPTION : frm_INSCRIPTION.NBINSCRIPTION.PageIndex:=2;
        C_PLANFORM    : frm_PLANFORM.NBPLANFORM.PageIndex      :=2;
        C_AVANCES     : frm_AVANCES.NBAVANCES.PageIndex        :=2;
        C_MVTMOIS     : frm_MVTMOIS.NBMVTMOIS.PageIndex        :=2;
        C_COMPouv     : frm_COMPOUV.NBCOMPOUV.PageIndex        :=2;
        C_PRESCHANT   : frm_PRESCHANT.NBPRESCHANT.PageIndex    :=2;
        C_CONGESOUV   : frm_CONGESOUV.NBCONGESOUV.PageIndex    :=2;

      end;
      BTN_maj(0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,1,1,1);

      delimiter;
    end;
    BTNsais.Flat:=true;
    BTNlist.Flat:=true;
    BTNsel.Flat:=false;
  end;

  procedure Tfrm_maingrh.BiB_ecranClick(Sender: TObject);
  begin
  if ActiveMdichild <> NIL then
     ActiveMdichild.close;
  frm_grille.hide;

  end;

  procedure Tfrm_maingrh.G16Click(Sender: TObject);
  begin
    Application.CreateForm(Tfrm_PERSONNE,frm_PERSONNE);
    frm_PERSONNE.show;

  end;

  procedure Tfrm_maingrh.G01Click(Sender: TObject);
```

```
begin
  Application.CreateForm(Tfrm_COMPOS,frm_COMPOS);
  frm_COMPOS.show;

end;

procedure Tfrm_maingrh.G07Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_CATEGNIV,frm_CATEGNIV);
  frm_CATEGNIV.show;
end;

procedure Tfrm_maingrh.G40Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_PLANFORM,frm_PLANFORM);
  frm_PLANFORM.show;
end;

procedure Tfrm_maingrh.G39Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_ORGFORM,frm_ORGFORM);
  frm_ORGFORM.show;
end;

procedure Tfrm_maingrh.G42Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_OUVPERM,frm_OUVPERM);
  frm_OUVPERM.show;
end;

procedure Tfrm_maingrh.G24Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_REMFRAMED,frm_REMFRAMED);
  frm_REMFRAMED.show;

end;

procedure Tfrm_maingrh.G17Click(Sender: TObject);
begin
end;

procedure Tfrm_maingrh.G18Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_ABSENCE,frm_ABSENCE);
  frm_ABSENCE.show;
end;

procedure Tfrm_maingrh.G19Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_CONGES,frm_CONGES);
  frm_CONGES.show;
end;

procedure Tfrm_maingrh.G20Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_DEPLACEMENT,frm_DEPLACEMENT);
  frm_DEPLACEMENT.show;
end;

procedure Tfrm_maingrh.G21Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_HEURSUP,frm_HEURSUP);
  frm_HEURSUP.show;
end;

procedure Tfrm_maingrh.G22Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_PRIMREND,frm_PRIMREND);
  frm_PRIMREND.show;
end;

procedure Tfrm_maingrh.G23Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_FICHNOT,frm_FICHNOT);
  frm_FICHNOT.show;
end;

procedure Tfrm_maingrh.G41Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_FORMATION,frm_FORMATION);
  frm_FORMATION.show;
end;

procedure Tfrm_maingrh.G25Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_AVANCES,frm_AVANCES);
  frm_AVANCES.show;
end;

procedure Tfrm_maingrh.G26Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_DETACHE,frm_DETACHE);
  frm_DETACHE.show;
end;

procedure Tfrm_maingrh.G27Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_DEMISSION,frm_DEMISSION);
  frm_DEMISSION.show;
end;

procedure Tfrm_maingrh.G28Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_RTRDEC,frm_RTRDEC);
  frm_RTRDEC.show;
end;

procedure Tfrm_maingrh.G29Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_SANCTION,frm_SANCTION);
  frm_SANCTION.show;
end;
```

```
procedure Tfrm_maingrh.G30Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_PROMOTION,frm_PROMOTION);
  frm_PROMOTION.show;
end;

procedure Tfrm_maingrh.G45Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_MVTMOIS,frm_MVTMOIS);
  frm_MVTMOIS.show;
end;

procedure Tfrm_maingrh.G52Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_PRESCHANT,frm_PRESCHANT);
  frm_PRESCHANT.show;
end;

procedure Tfrm_maingrh.G51Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_OUVCHANT,frm_OUVCHANT);
  frm_OUVCHANT.show;
end;

procedure Tfrm_maingrh.G34Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_sel_avancable,frm_sel_avancable);
  frm_sel_avancable.showmodal;
end;

procedure Tfrm_maingrh.G43Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_CONGESOUV,frm_CONGESOUV);
  frm_CONGESOUV.show;
end;

procedure Tfrm_maingrh.G03Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_COMCATCAD,frm_COMCATCAD);
  frm_COMCATCAD.show;

end;

procedure Tfrm_maingrh.G02Click(Sender: TObject);
begin
   Application.CreateForm(Tfrm_COMPFONC,frm_COMPFONC);
  frm_COMPFONC.show;

end;

procedure Tfrm_maingrh.G08Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_CATGRAD,frm_CATGRAD);
  frm_CATGRAD.show;
end;
```

```
procedure Tfrm_maingrh.G09Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_CATCAD,frm_CATCAD);
  frm_CATCAD.show;
end;

procedure Tfrm_maingrh.G10Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_CATFON,frm_CATFON);
  frm_CATFON.show;

end;

procedure Tfrm_maingrh.G04Click(Sender: TObject);
begin
Application.CreateForm(Tfrm_COMGRA,frm_COMGRA);
frm_COMGRA.show;

end;

procedure Tfrm_maingrh.G05Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_COMFONCAD,frm_COMFONCAD);
  frm_COMFONCAD.show;

end;

procedure Tfrm_maingrh.G06Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_COMPDG,frm_COMPDG);
  frm_COMPDG.show;
end;

procedure Tfrm_maingrh.G11Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_FNOMENC,frm_FNOMENC);
  frm_FNOMENC.show;
  BTN_maj(0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0);

end;

procedure Tfrm_maingrh.G44Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_COMPOUV,frm_COMPOUV);
  frm_COMPOUV.show;
end;

procedure Tfrm_maingrh.G46Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_GENERPOUV,frm_GENERPOUV);
  frm_GENERPOUV.showModal;
end;

procedure Tfrm_maingrh.G33Click(Sender: TObject);
begin
```

```
  Application.CreateForm(Tfrm_EDITMENS,frm_EDITMENS);
  frm_EDITMENS.showModal;
end;

procedure Tfrm_maingrh.G31Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_GENERPAIE,frm_GENERPAIE);
  frm_GENERPAIE.showModal;
end;

procedure Tfrm_maingrh.G35Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_sel_conges,frm_sel_conges);
  frm_sel_conges.showmodal;

end;

procedure Tfrm_maingrh.G36Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_sel_fichnot,frm_sel_fichnot);
  frm_sel_fichnot.showmodal;
end;

procedure Tfrm_maingrh.G37Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_sel_notper,frm_sel_notper);
  frm_sel_notper.showmodal;
end;

procedure Tfrm_maingrh.G50Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_sel_dectriouv,frm_sel_dectriouv);
  frm_sel_dectriouv.showmodal;

end;

procedure Tfrm_maingrh.G48Click(Sender: TObject);
begin
 with frm_maingrh do
  begin
  end ;
end;

procedure Tfrm_maingrh.G49Click(Sender: TObject);
begin
 with frm_maingrh do
  begin
  end ;
end;

procedure Tfrm_maingrh.N1Click(Sender: TObject);
begin
 with frm_maingrh do
  begin
  end ;
end;
```

```
procedure Tfrm_maingrh.N5Click(Sender: TObject);
begin
 with frm_maingrh do
  begin
  end ;
end;

procedure Tfrm_maingrh.N9Click(Sender: TObject);
begin
 with frm_maingrh do
  begin
  end ;

end;

procedure Tfrm_maingrh.N15Click(Sender: TObject);
begin
  Application.CreateForm(Tfrm_GENERDISK,frm_GENERDISK);
  frm_GENERDISK.showModal;

end;

procedure Tfrm_maingrh.N17Click(Sender: TObject);
begin
 with frm_maingrh do
  begin
  end ;

end;

procedure Tfrm_maingrh.N18Click(Sender: TObject);
begin
 with frm_maingrh do
  begin
  end ;
end;

procedure Tfrm_maingrh.N20Click(Sender: TObject);
begin
 with frm_maingrh do
  begin
  end ;
end;

procedure Tfrm_maingrh.N22Click(Sender: TObject);
begin
 with frm_maingrh do
  begin
  end ;
end;

procedure Tfrm_maingrh.N24Click(Sender: TObject);
begin
 with frm_maingrh do
  begin
  end ;
end;

procedure Tfrm_maingrh.N31Click(Sender: TObject);
begin
 with frm_maingrh do
  begin
  end ;
end;

procedure Tfrm_maingrh.N29Click(Sender: TObject);
begin
 with frm_maingrh do
  begin
  end ;
end;

procedure Tfrm_maingrh.N28Click(Sender: TObject);
begin
 with frm_maingrh do
  begin
  end ;
end;
procedure Tfrm_maingrh.N26Click(Sender: TObject);
begin
 with frm_maingrh do
  begin
  end ;
end;
end.
//#FV1xC00150
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Eric Harley. "*Yourden System Method vs. MERISE*. URL: http://www.cdf.toronto.edu/~g8kimchi/, (July 2004).

[2] Annya Romanczuk and Izabele Borne. "*Objectifying a MERISE Analysis Using Transformation Rules,*" URL: http://www.iam.unibe.ch/~famoos/ESEC97/submissions/romanczuk.pdf (April 2004).

[3] Peter Biggs, "*A Survey of Object Oriented Methods.*" URL: http://students.cs.byu.edu/~pbiggs/survey.html, (June 2004).

[4] HR Press "*The Largest Selection of Software for Human Resource Management, Workplace Safety, and Employee Testing and Training."* URL: http://www.hrpress-software.com/peopleman.html, (March 2004).

[5] IEEE 1016-1998, Recommended Practice for Software Design Descriptions.

[6] IEEE 1074-1997, Standard for Developing Software Life Cycle Processes.

[7] IEEE 1219-1998, Standard for Software Maintenance.

[8] Jeffrey L. Whitten, Lonnie D. Bently, and Kevin C. Dittman, "*System Analysis Design Methods,*" 6th ed.

[9] Jean-Pierre Fournier. "*MERISE*". URL: http://www.infeig.unige.ch/support/se/lect/gl/meth/web.html, (April 2004).

[10] Ron Flannery, *User Informix Handbook*, Informix Press, 2000. URL: http://www.informixhandbook.com/, (March 2004).

[11] Dean Leffingwell and Don Widrig, *Managing Software Requirement: A Unified Approach*, Addison-Wesley, 2000.

[12] Simon Bennett, Ken Lunn and John Skelton, *Schaum's Outlines of UML*, McGraw Hill, 2001.

[13] Object Management Group, *UML Resource Page*. URL: http://www.omg.org/uml/, (March 2004).

[14] SYBASE "PowerDesigner Conceptual Data Model User's Guide." URL: http://sybooks.sybase.com/, (June 2004).

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dr. Peter Denning
   Computer Science Department
   Naval Postgraduate School
   Monterey, California

4. Dr. Man-Tak Shing
   Computer Science Department
   Naval Postgraduate School
   Monterey, California

5. Dr. Doron Drusinsky
   Computer Science Department
   Naval Postgraduate School
   Monterey, California

6. Noureddine Trigui
   Tunisia