DESIGN AND IMPLEMENTATION OF A SEMANTIC ONTOLOGY FOR THE ADMINISTRATION DEPARTMENTS OF PRIMARY AND SECONDARY EDUCATION

by

ELEFTHERIOS TZAGKARAKIS

B.A. in Informatics, Athens University of Economics and Business, 2001

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SCHOOL OF ENGINEERING

HELLENIC MEDITERRANEAN UNIVERSITY

2020

Approved by:

Major Professor
Dr. Nikolaos Papadakis

Abstract

Advances in computers and communications have significantly changed almost every aspect of our daily activity. In this maze of change, governments around the world cannot remain indifferent. Public administration is evolving and taking on a new form through e-Government. A large number of organizations have set up websites establishing an online interface with the citizens and businesses with which it interacts. However, few organizations provide integrated services of stage 4 through their websites and these are mainly central organizations of national scope. Most organizations, especially the decentralized agencies of the ministries and local authorities, provide up to stage 2 services and a few up to stage 3 and 4 services. Besides, these services are mainly focused on serving citizens and businesses and less on providing services to employees. In this thesis, we describe the process of developing a Semantic Ontology to support the administrative procedures of the Decentralized Organizations of the Ministry of Education of Greece. Finally, we describe the development of an e-government portal that offers stage 1-4 services to employees, using the above ontology for modeling and data management.

Περίληψη

Η πρόοδος στον τομέα των υπολογιστών και των επικοινωνιών έχει αλλάξει σημαντικά όλες σχεδόν τις πτυχές της καθημερινής μας δραστηριότητας. Σε αυτόν τον κυκεώνα των αλλαγών δεν μπορεί να μείνει αμέτοχο το κράτος και η δημόσια διοίκηση. Η ηλεκτρονική διακυβέρνηση είναι πλέον επιτακτικά αναγκαία και αποτελεί φυσική εξέλιξη των πραγμάτων. Ένας αρκετά μεγάλος αριθμός οργανισμών έχει προχωρήσει στην δημιουργία ιστοτόπων εδραιώνοντας μια ηλεκτρονική διεπαφή με τους πολίτες και τις επιχειρήσεις με τις οποίες συναλλάσσεται. Είναι όμως λίγοι οι οργανισμοί οι οποίοι παρέχουν μέσω των ιστοτόπων τους ολοκληρωμένες υπηρεσίες επιπέδου 4 και πρόκειται κυρίως για κεντρικούς οργανισμούς εθνικής εμβέλειας. Οι περισσότεροι οργανισμοί και κυρίως οι αποκεντρωμένοι οργανισμοί των υπουργείων και οι οργανισμοί τοπικής αυτοδιοίκησης παρέχουν μέχρι υπηρεσίες επιπέδου 2 και λίγοι μέχρι υπηρεσίες επιπέδου 3 και 4. Επίσης, οι υπηρεσίες αυτές προσανατολίζονται κυρίως στην εξυπηρέτηση των πολιτών και των επιχειρήσεων και λιγότερο σε παροχή υπηρεσιών προς τους υπαλλήλους. Στην εργασία αυτή περιγράφουμε την διαδικασία ανάπτυξης μιας οντολογίας του Σημασιολογικού Ιστού για την υποστήριξη των διοικητικών διαδικασιών των Αποκεντρωμένων Υπηρεσιών του Υπουργείου Παιδείας. Στη συνέχεια, περιγράφουμε την ανάπτυξη μιας ηλεκτρονικής πύλης που προσφέρει υπηρεσίες επιπέδου 1-4 σε πολίτες (government to citizen) και υπαλλήλους (government to employee), χρησιμοποιώντας την παραπάνω οντολογία για την μοντελοποίηση και για τη διαχείριση των δεδομένων.

Table of Contents

List of Figures

List of Tables

Acknowledgements

With the completion of this thesis, I would like to thank my supervisor Professor Dr. Nikolaos Papadakis, for his valuable support and guidance.

Dedication

To my family

Introduction

The progress made in the field of computers and among them in the field of networks and the internet has significantly affected all areas of our daily lives. Our daily activities have the potential to be simplified and performed more efficiently. This progress has affected not only the operations in the Private Sector but also the operations of the Public Sector by introducing the concept of e-government. In this context, governments are trying to formulate a central digital policy, by delineating the axes and determining directions for its development. Their goal is to increase and improve the services provided electronically, with minimal physical interaction, and reducing administrative burdens. Special emphasis has been also placed on facilitating access to electronic services through more user-friendly interfaces and one-stop government points.

The establishment of e-government requires the transformation of public administration through the simplification and digitization of administrative procedures. The most effective operational planning presupposes the understanding and modeling of the correct needs of the procedures. This will lead to the necessary redesign of the procedures, to optimize the provided electronic services.

This transformation is not a simple process. The Public Sector has a huge range of functions and services offered to citizens and businesses. The structure and operation of the Public Sector make reforms time-consuming and complex. Reforming usually requires changes in the legislation and a series of circulars that will regulate the individual issues that arise. Besides, the rapid introduction of computers in society in recent decades has created the need for the immediate introduction of computers in Public Administration. This resulted in the development of systems-oriented to the needs of each organization, before the central government had time to adequately formulate and implement a Central Digital Policy. Many of the developed systems attempted to digitalize existing processes without having undergone the necessary administrative transformations. Therefore, these systems face difficulties in cooperating, showing a high degree of heterogeneity. Besides, these systems face difficulties in adapting to the newly reformed framework set by the central government.

If we examine the way e-government has developed in recent years, we see that the key priority is the provision of services to Citizens (government to citizen) and Businesses

(government to business). However, another area of e-government that has not been explored extensively is the provision of electronic services to employees (government to employee). The low priority given to this area is normal, as the administrative burden of handling services to civil servants is generally lower. On the other hand, there are cases of public organizations with a large number of employees and a large geographical spread, such as the Ministry of Education of Greece. In this case, the number of employees makes the administrative burden for their management measurable.

At a theoretical level, the need for online governmental services has been widely recognized. There are many initiatives in recent years around the world and several projects have been developed to promote e-government. Remarkable is also the research in the field of semantic ontologies in e-government. The majority of these proposals use a top-down approach and are mainly concerned with defining horizontal ontologies. But, as Heeks & Bailur (2007) and later Alcaide Muñoz & Rodríguez Bolívar (2017) point out, there is a gap between theory and practice. Despite the plethora of work and suggestions on the use of ontologies in e-government, the percentage of work that has suggestions for practical applications is low.

On a practical level, many organizations have set up websites establishing an electronic interface with the citizens and businesses with which they transact. However, few organizations provide integrated stage 4 services through their websites and these are mainly central national organizations. Most agencies, especially decentralized ministry agencies and local government agencies, provide up to stage 2 services and a few up to stage 3 and 4 services. Also, few agencies have been involved in offering integrated services to their employees. Besides, there are few e-government applications based on semantic ontologies. Finally, it is noteworthy that despite the research carried out, there is difficulty in defining backbone ontologies by governments.

In this paper, we present on the one hand the development of an ontology to support administrative work and on the other hand, a good practice for the process of developing an e-government portal that will provide comprehensive services to employees. The portal is based on the ontology we built, both for the modeling, as well as for data management. We followed a bottom-up approach, starting with the analysis of the requirements of the organizations and continuing with the design and implementation. The decentralized Primary and Secondary

Education Agencies under the Ministry of Education were selected as the scope of application. These organizations have a common administrative framework and are called to serve the needs of more than 200,000 permanent and contract teachers in Greece.

With this work, we propose a model for the development of e-government, through the development of individual specialized vertical ontologies, developed with a bottom-up approach and the interconnection between them. The contribution of this work lies in exploring the development of ontologies for government to employee. With our work, we try to highlight the usefulness of vertical ontologies in the provision of integrated electronic services for both modeling and mainly for data management, by presenting a good practice for e-government.

Chapter 1 - Ontologies

1.1 Definition

The roots of the word ontology date back to antiquity in the domain of philosophy. Over the years, this word has been used by other sciences as well. The word is also used in recent years in the field of computer science. The word ontology has been used to describe various structures ranging from simple classifications, metadata schemas to logical theories.

A brief interpretation of the modern use of the word Ontology has been given by Gruber [1], who defined that: “An ontology is an explicit specification of a conceptualization”. By conceptualization, Gruber refers to the set of objects, concepts, and other entities that exist in a particular area of interest, as well as the relationships with which they are associated.

Guarino [2] refined Gruber’s definition by making clear the difference between an ontology and a conceptualization. So, Guarino defined Ontology as “a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualization of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models.”

An ontology contains the following key components:

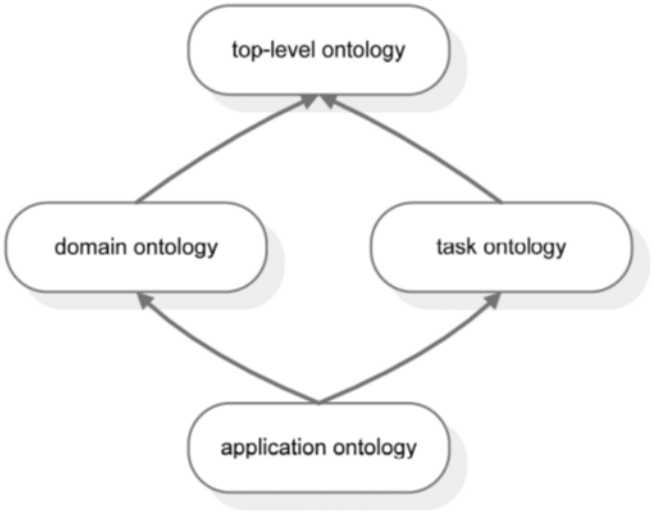
- **Classes**, which are the basic concepts of the cognitive field on which the ontology focuses, and may consist of subclasses, which inherit all the characteristics of the class to which they belong.
- **Properties**, which represent the basic characteristics of the classes.
- **Relationships**, which depict the interaction between the concepts of one field.
- **Restrictions**, which express some limitations in both relationships as well as in the properties of the ontology.
- **Axioms**, which represent proposals that are always true.
- **Instances**, concerning specific elements - examples of classes.

The use of Ontologies has several advantages. The main advantage of ontologies is the modeling of knowledge by offering a common understanding of it. Ontologies are a unifying conceptual framework between different perceptions, and thus help to achieve the organization, classification and management of information.

1.2 Ontologies' Types

There are several approaches to categorizing ontology types. Below we list some of them. Guarino [2] proposes the development of different types of ontologies according to their level of generality as shown in figure 1.1 [2].

Figure 1.1 Types of Ontologies according to Guarino



The different types of ontologies can be distinguished as follows:

- **Top Level Ontologies:** Describe very general concepts (such as space, time, event), which are independent of a specific problem or area of knowledge. It makes sense to have a high-level unified ontology for large user communities.
- **Domain Ontologies:** Describes vocabulary related to a general field of knowledge specifying the concepts introduced in the top-level ontology.
- **Task Ontologies:** Describes vocabulary related to a general activity / task in order to specialize in Top Level Ontologies.

- **Application Ontologies:** They are the most specific ontologies. Concepts in application ontologies often correspond to roles of knowledge domain entities as they perform a specific activity.

Gomez-Perez and Benjamins [3] classify ontologies as follows:

- **Knowledge representation ontologies**: provide representation entities without specifying what specifically they represent.
- **General/ common ontologies:** aim to capture general knowledge about the world, providing basic concepts such as time, space, events, etc.
- **Top-level ontologies:** provide general concepts under which all terms are related to pre-existing ontologies.
- **Metadata ontologies:** provide a vocabulary to describe the information content, which is available online.
- **Domain ontologies:** represent knowledge around a specific field, e.g. public administration etc.
- **Method or Task ontologies:** provide terms that refer to specific tasks

1.3 Semantic Web and Ontologies

The Semantic Web is an extension of the existing Web, with the difference that in the Semantic Web the information is understood not only by humans but also by agent programs. Ontologies use a common vocabulary that allows the exchange of information between the applications of the Semantic Web, in order to be able to represent knowledge in it.

The Semantic Web promotes the interoperability of information. The information is better organized and can be reused by different web applications, such as search engines, web portals, intelligent agents, and web services.

In the Semantic Web, both the processing and the exchange of information are facilitated by the use of ontologies. Ontologies contain a set of concepts and semantic correlations between them. The concepts describe classes of objects, ie concepts-standards related to objects, while the correlations usually concern hierarchical dependencies between the concepts. In addition, in an

ontology there may be properties of concepts, restrictions around them, equivalence and separation relations as well as semantic correlations between terms with the use of logic.

Moreover, the use of ontologies in the Semantic Web allows us to infer and draw conclusions. We can also automatically categorize objects or resources in the class hierarchy when the class to which they belong is not known but their properties are known. The Semantic Web not only contains links from some ontologies that differ from each other, but also contains ontologies that are interconnected and complement each other. This means that one can use some ontology in whole or in part, while one can redefine its terms to serve one's own needs or the needs of a group with common interests.

Chapter 2 E-Government

2.1 Definition

E-government is not just a matter of using computers in public administration or automating old practices. These do not increase the efficiency of the Public Sector, or promote the participation of citizens. As Khalil *et al.* [4] define, “e-government utilizes technology to accomplish reform by fostering transparency, eliminating distance and other divides, and empowering people to participate in the political processes that affect their lives”. E-government is a catalyst for public administration at all levels, as it enables the improvement of efficiency by offering a feasible solution to the conflicting requirements for providing high quality services with fewer resources but also ensures non-discriminatory access to citizens and businesses.

Seifert [5], made another approach to e-government: “E-government involves using information technology, and especially the Internet, to improve the delivery of government services to citizens, businesses, and other government agencies. E-government could enable citizens to interact and receive services from the federal government (or state and local governments) 24 hours a day, seven days a week”.

E-Government is the transformation of the internal and external relations of the public sector through ICT, in order to improve the services provided to citizens and businesses. This requires a significant reorganization and improvement of the services provided. It does not just mean their automation but something much more than that. Essentially, it is an effort in the general context of exploiting modern technologies in order for the ordinary citizen to be able to fulfill his obligations to Public Bodies by using computers and the internet, thus saving valuable time and avoiding bureaucracy. Also in this way there is greater security in the conduct of transactions, it is all more transparent and corruption can be fought since all actions can be controlled.

2.2 Objectives of e-government

There are two strategic goals of e-government:

- The development and provision of electronic services of high quality and low cost to citizens and businesses, which meet the requirements of the modern economic and social environment
- The development of integrated and interoperable information systems of public administration, which assist the management of knowledge. In this way, governments will be able to rationally plan their action, measure the results and make correct and enforceable decisions.

These two objectives aim at the following benefits for the public administration and the citizens:

- Reducing the cost of providing services
- Reducing the need for communication with the public (call centers, counters)
- Better coordination between Bodies - common standards
- Additional benefits from process reorganization, which gradually make better use of information and communication technologies
- Possibility of new services and operating methods (eg tele-work, forums, consultation, distance learning)
- Reduction of service time
- Cost reduction for citizens and businesses
- Increase data security and integrity
- Services provided on a "24 X 7" basis
- Services that do not discriminate on the basis of gender, color, age
- Possibilities of new services (eg e-Democracy)

2.3 Sectors of E-government

Regarding the types of services, there are the following basic models [5]:

- **Government to Government (G2G):** It concerns the relations between public bodies and all those activities that will improve and upgrade the services of the government and lay the foundations for the electronic service of citizens and businesses. Some of the activities it includes are the following:
    - Facilitate and automate inter-organization transactions.
    - Elimination of overlaps and responsibilities.
    - Easy and fast flow of information.
    - Easy and fast handling of documents.
    - Simplification of procedures.
- **Government to Citizens (G2C):** It concerns the relations between public bodies and citizens. It deals with the management of relations with the citizens. Includes some or all of the following activities:
    - Electronic provision of information.
    - Electronic certification of citizens.
    - Electronic submission of applications.
    - Electronic issuance of certificates
    - Possibility of electronic payment.
- **Government to Business (G2B):** Refers to the relationship between public bodies and private companies. It deals with collaborations, joint activities and practices, nationally and internationally. Includes all or part of the following activities:
    - Electronic provision of information information.
    - Electronic certification of the company and operating licenses.
    - Electronic supplies.
    - Facilitation and automation of commercial transactions.

- **Government to Employee (G2E):** Refers to the relationship between government and its employees only. Its purpose is to offer online services to the employees, and at the same time to improve their management. There is not much research in this sector. Many researchers consider it as a part of G2G sector [6]. Includes all or part of the following activities:
    - applying online for an leave
    - checking the balance of leave
    - reviewing salary payment records
    - applying for Certificates

2.4 Advantages of E-government

The advantages of e-government are divided into two main categories as follows:

- **Increase productivity in Public Administration**

This is achieved with many ways. First of all by reducing the cost of services and communication with the public. The use of common standards leadsd to better coordination between actors. Moreover, the better use of ICT leading to reorganization of processes and the possibility of providing new services and operating methods, such as tele-work, distance learning and forums.

- **Better services for citizens and businesses**

This is ensured by reducing the service time and costs for citizens and businesses. It is very important to provide services beyond the typical serving hours of the public sector. The user of an electronic service does not need to know the mode of operation, the structure and the responsibilities of the Public Administration bodies involved in its service. His only responsibility is to receive the result of the service from an exit point without being involved in intermediate stages of service (One Stop).

2.5 Stages of maturity for electronic governance services

There are four stages in classifying the maturity of e-Government services in terms of its internal operation [7].

- **Information:** Through their websites, public bodies publish information material about their services.
- **One-way Interaction:** Citizens have the opportunity to interact with Public Bodies to process services. There are forms available on the web sites which the interested party can download and print. The service process begins with their physical submission to the competent Agency. It is also possible to communicate with the Institutions with the help of e-mail.
- **Two-way Interaction:** Certified citizens-users have access through the web sites to official electronic forms, which they fill in to start the service process electronically. The completion of the process is done in a non-electronic way.
- **Transaction:** Certified citizens-users have the ability to process an service electronically, ie they send the request to start the service to the competent body and receive the administrative act electronically, without requiring any further work from them to complete their service. Services under this level completely replace the corresponding conventional, non-electronic services.

2.6 Ontologies in E-government

In the last decades, there has been an increase in semantic web ontologies that try to model the services offered by the Public Sector. Fraser *et al.* [8] studied knowledge management, which can be used for designing and developing e-government services. They suggest the use of knowledge units which through a domain map are related to the transaction service components that will be implemented. In the context of the SmartGov project, they created a general-purpose ontology that aims to provide a conceptual framework at the cognitive level and not a special-purpose ontology.

Another project that recognizes the need for proper knowledge management and deals with the conceptual level is the OntoGov project [9], [10]. This project tries to address the problems that arise in the provision of services to public bodies due to the frequent changes that take place

in the legislation. For this purpose, they defined a cluster of ontologies for modeling e-government services. The overall goal of the OntoGov project is to develop, test, and validate a semantically enriched platform that will facilitate the consistent restructuring of e-government services.

The ICTE-PAN project dealt with the support and structure of the highest-level functions of the government, such as the planning, implementation, and evaluation of public policies for major and complex social problems. In this work, Loukis [11] distinguished the need to interconnect heterogeneous systems with different backgrounds, interests, and values. He developed a horizontal ontology which, however, can be combined with vertical ontologies in the management of specialized topics.

Peristeras and Tarabanis [12] proposed a top-level enterprise architecture, the Governance Enterprise Architecture (GEA), that aims at the overall description of a governmental model. They suggest five levels for process and object models, the GEA mega-process model, the GEA interaction model, the GEA public policy formulation object model, the GEA service provision object model, and the latest development of the GEA object model for the overall governance system. According to the proposed model, the interaction between the Administration and the citizens is divided into two main parts. The first one, the planning part consists of the necessary actions needed to provide citizens with the necessary information to identify and use the available services. The second consists of the necessary actions to provide the citizen with the product of the service. Their research continued with the implementation of the model using OWL [13]. In this context, they developed an application that attempts to provide the user with access to related functions, based on his profile. They also explored the conceptual mappings between GEA entities and WSMO service model components [14].

GEA was the basis for other works in the development of ontologies, particularly in the area of service detection by citizens. The OntoAL ontology [15] attempts to adequately describe the state structure of Albania, making the necessary modifications to adapt to local conditions. This ontology aims to translate public services into daily activities so that the citizen can easily find services even if they have not been modeled.

Salhofer *et al.* [16] used a model proposed in GEA and constructed a semantic model to support the identification of the services by the citizens. They proposed a Model-Driven

Architecture methodology and they built a framework for the needs of a municipality. Its purpose was to assist users in the search for services and to provide them with relevant access to it, making them independent of domain experts.

There are also many more top-level ontologies for e-government. Indicatively, we mention some of them. Alshehab *et al.* [17] proposed an e-government framework in OWL that uses semantic ontologies for the State of Kuwait. The ontologies are designed for assisting interoperability and interconnecting information from different government organizations. They integrated information from the domains of health care, education, and civil information, taking into account the common information from all these domains. IndiGov-O is another top-level ontology that conceptualizes the structure of government in India. Kumar and Joshi [18], introduce a 4-level hierarchy to represent the ministries and their departments that will form the basis for future extension of e-government.

Apart from top-level ontologies, there is interesting research for the development of vertical ontologies in specific sectors of Public Administration. In the field of Legal Ontologies, Gómez-Pérez *et al.* [19] introduced an e-Government ontology model for the Real-estate transaction domain, in Spain. This model is a part of the EGO Ontology model that was developed within the Reimdoc Project. The aim of the project was the modeling of legal documents and information to enable their retrieval during the processing of transactions with citizens. This issue has also been investigated in other works [20], [21]. The eGRRC framework that was proposed by Hasan *et al.* [22] can be used for modeling legislation but mainly focuses on regulatory requirements compliance and their interrelationships.

Another area that has attracted the interest of researchers is financial e-government ontologies. Brusa *et al.* [23] presented an ontology for the budgetary and financial system of Santa Fe Province in Argentine. They combined ontology development methodologies and software engineering techniques to highlight the advantages of ontology-based applications. One more financial ontology, focusing on public procurement, was developed by Muñoz-Soro *et al.* [24]. The PPROC ontology covers all the stages of procurement processes and contracts. The Zaragoza's City Council and the Provincial Government of Huesca have adopted it. PPROC ontology complies with transparency laws providing open data for public procurement.

Other researchers have also pointed out the adequacy of semantic ontologies in open government and open data. Lourenço *et al.* [25] developed a transparency ontology with OWL and Protégé. They started their research driven by the deduction that there is difficulty in selecting the data that will be used to create an effective ontology. Therefore, they focused on their research into which data is more understandable to citizens, in order to select an optimal set for the ontology. An innovative work was developed at Rensselaer Polytechnic Institute in the field of linked open government data [26]. The result of this work was the TWC LOGD portal, a Semantic Web-Based platform for sharing open data. The approach for the LOGD portal was adopted in the UK and USA for their open data portals. In the UK (Data.gov.uk) there was a top-down implementation, while in the USA (Data.gov) they preferred a bottom-up method.

Savvas and Bassiliades [21] proposed another vertical ontology that aims to support integrated services of a public organization. The ontology consists of the structure of Public Administration in Greece and the documents that are used. The documents are divided into the documents that are produced by the administrative procedures and the documents that concern the legal framework that is applied. They presented the application of the relevant ontology in the administration procedures of local authorities of Greece.

Ontologies beyond their effectiveness in imaging and modeling systems for developing applications that implement services may be particularly useful for the interconnection of heterogeneous systems. There is a lot of relevant work on this subject. Indicatively we mention some of them. Gugliotta *et al.* [27] presented a semantic-based architecture of a one-stop government portal, which on the one hand supports the user in searching for services on interconnected systems and on the other hand ensures the processing of the requested service. EG-BOnt was another approach by Xiao *et al.* [28] for modeling and collaboration on the Government to Business field. Kanagwa *et al.* [29] proposed an e-government interoperability framework in Uganda. They designed a National Enterprise Architecture that is based on a set of related ontologies.

Chapter 3 Proposed Solution

The contribution of the above works is particularly useful and necessary for the mapping of how semantic ontologies can be utilized in e-government. These papers use more a top-down approach and focus on top-level and horizontal ontologies. Most of them try to give general directions for the development and use of ontologies in Public Administration. Despite the relevant research, governments around the world face difficulties to define a central governmental ontology that will support the development of electronic services. Besides, there are not many vertical ontologies, especially in the field of government to employee.

The experience gained from the above work is a basis for the next step in this area. Our proposal is based on the development of vertical ontologies that will result from a bottom-up approach. Each organization or group of organizations with a common working framework must design the ontology that will support its administrative procedures. Beyond that, emphasis should be placed on the interconnection of these ontologies to achieve the desired degree of interoperability.

The heterogeneity that results from this process should not be considered as a deterrent. Given the general difficulty of defining and applying common standards, heterogeneity is part of the system and is already a phenomenon that is being addressed. Heterogeneity is a more general problem that does not occur only in our case. Also, the modeling offered by the use of ontologies helps us to overcome the issues of heterogeneity more easily. Several papers suggest solutions to this issue through the use of ontologies [27]–[29]

For our application, we used semantic ontologies. Semantic Ontologies offer efficient modeling of structure, data, and services. This in addition to the advantages it offers us for design and implementation facilitates the interconnection of heterogeneous systems. Moreover, the reasoning capability enables us to infer information easier from our data

The solution we propose can also be considered in conjunction with other researches that explore the potential for export data from Relational Databases into Semantic Ontologies [30]–[34]. In this way, it is possible to utilize the huge volume of data that already exists in Relational Databases, to facilitate the transition of Information Systems using Relational Databases to Semantic Ontology Systems.

The purpose of our work is to create an ontology to support administrative work and an ontology-based e-government portal that offers integrated services of stages 1-4 to employees. In this context, the ontology manages the modeling and management of knowledge, which is used in services of all 4 stages. The Web Portal, for its part, undertakes the implementation of the services, offering the necessary interfaces to the clients (employees) for the submission of requests to the administration and the receipt of the results. It also provides the necessary interfaces for the administration to access the SPARQL endpoint for data management and request processing.

For the construction of the ontology, we chose the OWL language, because of its stronger semantics and logic relation expressiveness. For the implementation of the ontology we used Protégé 5.5.0 and we verified the ontology with the Pellet reasoner. The portal was built with Open Source tools. The web interface was powered by WordPress and we used PHP custom templates that were handling the user requests and the SPARQL queries. We also set up a SPARQL Endpoint with Apache Jena for data management.

Chapter 4 Methodology and Application framework

4.1 The domain of the Ontology

The ontology we developed aims to support the work of the administrative departments of the Regional Agencies of Primary and Secondary Education of the Ministry of Education of Greece. The Ministry of Education is the Ministry with the largest number of employees in Greece and its agencies cannot be considered as classic public agencies.

The decentralized administrative agencies of the Ministry consist of the Regional Directorates of Primary and Secondary Education, the Directorates of Primary Education, the Directorates of Secondary Education, and the school units. The administration departments of the Directorates have an administrative role, as the Regional Directorates have a coordinating role, while school units have mainly an educational role. The administration departments of Directorates are called to serve the needs of more than 200,000 permanent and substitute teachers serving in primary and secondary education units. In addition to the above, several decentralized bodies have a purely educational role, such as PEKES, KESY, KPE, etc. These agencies perform minimal administrative acts, and their administrative procedures are served by the Directorates of Education.

The staff serving in the administrative and educational structures shows intense mobility. The permanent employees are hierarchically subordinated to one or more school units and to a Directorate of Education where their records are kept. The main volume of administrative procedures is performed by the Directorates and a smaller part of the procedures are performed by the school units, the Regional Directorates, and the Ministry. The same applies to substitute teachers, only their employment relationship is valid for one academic year.

The permanent staff that manages the administrative services is small concerning the needs of the organizations. For this, every year a large number of teachers are seconded to cover administrative needs. Except that, in many positions of Heads are placed teachers. As a result, the Head of an institution that signs the administrative acts of the organization belongs organically to another body. At the same time, the administrative work for him should be processed by another organization.

Moreover, every year a significant number of teachers are transferred, at their request, to an area of responsibility of another directorate of education. This results Each Directorate is required each year to manage staff that is not organically subordinated to it and at the same time work for a part of its staff is handled by other Directorates. According to the legal framework, some permanent administrative procedures (such as leaves, extraordinary fees, etc.) are handled by the Organizations they serve and some others (such as payroll, issuance of a certificate of service status, etc.) are handled by the Organizations that belong organically.

Besides, about 25% of the teachers employed are substitute teachers that are hired each year. The procedures for hiring, managing, and firing them are quite time consuming and are repeated every year. Moreover, for each employee, it is necessary to take into account the administrative acts performed by the Organizations that have served in the past. Therefore, the model that we want to represent is complex and presents several peculiarities.

4.2 Methodology

There are several approaches for designing and implementing e-government ontologies. In general, the methodologies are divided into two categories [35]. In the first category, top-down methodologies attempt to cover the entire range of a domain by describing the concepts and relationships between them. Their goal is to cover all possibilities by creating a rich vocabulary. They are mainly focused on representing knowledge. The second category includes bottom-up methodologies that are aimed at representing experience. Ontologies are built step by step seeking to describe concepts that meet specific needs.

The bottom-up approach certainly has many disadvantages as pointed out Xiao *et al.* (2007) [28]. This approach does not use a common model and shape to represent the concepts. Each ontology uses a different model for its development. Nevertheless, despite the research that has taken place all these years, governments hesitate to adopt top-level ontologies. In our case, however, the bottom-up approach offers greater flexibility to adapt to the needs of the specific domain/organization. Besides, the purpose of ontologies in public administration is the representation of daily life events, which, however, are dynamic and changeable. Ontologies, therefore, need to be revised frequently to meet the needs of society and public administration.

This is easier to do in low-level ontologies. Moreover, these changes and can be combined with the life cycle of the applications used. In the current state of public administration, it is more realistic to build independent vertical ontologies and focus our efforts on interconnecting them.

For the development of our ontology, we adopted classical software development methods. We built the ontology incrementally, using an iterative development procedure.

During the process of developing the ontology, we followed the following steps:

Step 1: Recording status

1. Mapping of the structure of the decentralized services of the Ministry of Education.

The detailed recording was limited to decentralized agencies with administrative work. Apart from this, however, we took into account the general depiction of organizations of other bodies whose administrative acts affect the administrative procedures that we will implement (eg certificates of previous service issued by public or private bodies). For the recording, we relied on the relevant legislation which defines the organization chart of the agencies.

2. Recording of the procedures performed.

In this phase, we proceeded to the simple recording of the services offered to the employees and the citizens. For this recording, we proceeded with interviews with Heads of Administration Departments of four Directorates of Primary Education, four Directorates of Secondary Education, and two Regional Directorates of Education.

Step 2: Selection of procedures to be implemented.

Given the large volume of procedures performed, we proceeded to the selection of procedures aimed at a large number of employees and are repeated periodically. In this way, the result from the use of the ontology and the portal can directly reduce administrative burdens.

For each of the procedures we chose in the previous step, we followed steps 3 to 8.

Step 3: Analysis of procedures.

In analyzing the procedures, we recorded the flow of information and actions required. Specifically, we recorded:

a. The events (triggers) that initiate the process and the data that accompanies them (input data).

b. The individuals and organizations involved in the process.

c. The additional data that needs to be retrieved from the Organizations Records to complete the process.

d. The outputs of the process. We recorded their structure, the data they contain, and the way they are provided to the applicant.

Step 4: Ontology Design:

In this phase, the Entities were modeled and separated into Organizations, People, Documents, and Data participating in the processes. We also noted the relationships between them and the data properties.

Step 5: Implementation of the Ontology:

Then we implemented the Ontology of the previous step with the Protégé tool. The validity of the Ontology was checked with the Pellet reasoner and the OOPS (OntOlogy Pitfall Scanner).

Step 6: Create a test dataset.

To verify the functionality of the ontology we constructed a set of test data, which we uploaded to SPARQL Server.

Step 7: Integrate the process in the Portal

To check the correct design and implementation of the Ontology we tested its functionality in the Web Portal. We developed the user interfaces for each process we added and the necessary code to communicate with SPARQL Server.

Step 8: Test the Portal

During the testing of the functions in the Web Portal we checked the correctness of the design and implementation of the Ontology and proceeded to corrections where necessary, returning to Step 4.

Step 9: Ontology Evaluation

After completing steps 3 to 8 for all the procedures we chose to implement, we proceeded to the overall evaluation of the Ontology and the Web-Portal.

4.3 The Procedures

The procedures that are implemented are mainly divided into decision-making procedures (eg appointment, discharge, approving leave) and procedures that certify a status (eg certificate of service status, certificate of attendance). Also, the procedures can be divided into procedures in which the decision of the Administration is required to be taken into account for their completion (such as granting a leave) and in those that can be processed automatically without the intervention of the Administration (such as issuing a certificate).

The triggers that initiate the process are mainly the will of the citizens or employees that are expressed by submitting an application. The application is an incoming document that contains the details of the applicant, identifies the requested service, and the necessary parameters required to process the application (eg the application form contains the details of the employee, the type of leave, and the dates of leave). Some procedures are initiated ex officio by an organization as they fall under its statutory obligations such as hiring or firing teachers.

Data needed to complete a process consist of the data that are kept in the Agency's records for citizens or employees (eg for the student the years he has studied and in which school units).

Process outputs are documents that validate an act or certify a situation. To create these documents, it is necessary after collecting the necessary data concerning the employee or the citizen, to connect them with the details of the organization that issues the document, the

supervisor who certifies them, and the employee who compiles them. The Administration's Decision is also recorded in case it is required (eg the approval or rejection of an application of leave).

The procedures we chose to implement are the following:

- Issuance of Certificate of Attendance
- Issuance of Certificate of Previous Service
- Issuance of Certificate of Service Status
- Issuance of Certificate of Service Changes
- Granting leaves

Chapter 5 The Ontology

Following the previous analysis, we proceeded to the design of the ontology. The Ontology contains 75 classes, 41 object properties, and 53 data properties.

5.1 Classes

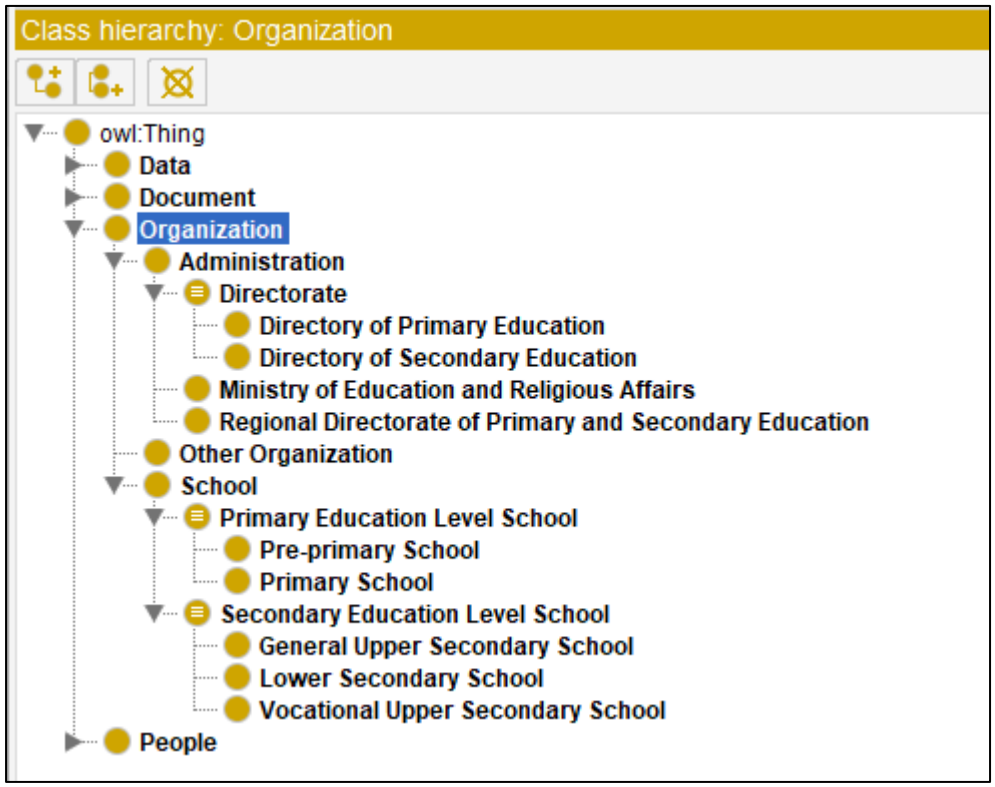
The entities that we represent are classified into the following classes:

- Organizations
- People
- Documents
- Data

5.1.1 Organization

The Organization's class hierarchy is described in the following figure:

Figure 5.1 Class: Organization



The Organization class is used to represent Organizations and their structure and has the subclasses Administration, School, and Other Organization.

Table 5-1 Class Organization

Class: Organization	
rdfs: label	**Organization**
SubClass Of:	Thing
	hasIssued some 'Outgoing Documents'
	hasReceived some Document
	hasReceivedService some PreviousService
Disjoint With:	Data, Document, People

The Administration class represents the Administrative Organizations of the relevant Body and includes the subclasses Ministry of Education and Religious Affairs, Regional Directorate of Primary and Secondary Education and Directorate. The subclass Directorate is divided into the subclasses Directorate of Primary Education and Directorate of Secondary Education. The subclass Ministry of Education and Religious Affairs has been created for the future expansion of Ontology.

Table 5-2 Class Administration and its subclasses

Class: Administration	
rdfs: label	**Administration**
SubClass Of:	Organization
	hasHead some Director
Disjoint With:	School, 'Other Organization'
Class: Ministry	
rdfs: label	**Ministry of Education and Religious Affairs**
SubClass Of:	Administration
	supervises some 'Regional Directorate of Primary and Secondary Education'
Disjoint With:	'Regional Directorate of Primary and Secondary Education', Directorate
Class: RegionalDirectorate	
rdfs: label	**Regional Directorate of Primary and Secondary Education**
SubClass Of:	Administration
	isSupervisedBy only 'Ministry of Education and Religious Affairs'
	supervises only Directorate

Disjoint With:	'Ministry of Education and Religious Affairs', Directorate
Class: Directorate	
rdfs: label	**Directorate**
SubClass Of:	Administration
	isSupervisedBy only 'Regional Directorate of Primary and Secondary Education'
Equivalent To:	'Directory of Primary Education' or 'Directory of Secondary Education'
Disjoint With:	'Ministry of Education and Religious Affairs', 'Regional Directorate of Primary and Secondary Education'
Class: DirectorateOfPrimaryEducation	
rdfs: label	**Directory of Primary Education**
SubClass Of:	Directorate
	supervises only 'Primary Education Level School'
Disjoint With:	DirectorateOfSecondaryEducation
Class: DirectorateOfSecondaryEducation	
rdfs: label	**Directory of Secondary Education**
SubClass Of:	Directorate
	supervises only 'Secondary Education Level School'
Disjoint With:	DirectorateOfPrimaryEducation

The School class represents the types of school. It is distinguished in the subclasses Primary Education Level School and Secondary Education Level School. The subclasses for the Primary Schools are Pre-primary School and Primary School. The subclasses for the Secondary Schools are Lower Secondary School, General Upper Secondary School, and Vocational Upper Secondary School.

Table 5-3 Class School and its subclasses

Class: School	
rdfs: label	**School**
SubClass Of:	Organization
	hasHead some Director
Disjoint With:	Administration, 'Other Organization'
Class: Primary_edu_school	
rdfs: label	**Primary Education Level School**
SubClass Of:	School
	isSupervisedBy only 'Directory of Primary Education'

Equivalent To:	'Pre-primary School' or 'Primary School'
Disjoint With:	'Secondary Education Level School'
Class: Pre-primary_school	
rdfs: label	**Pre-primary School**
SubClass Of:	Primary Education Level School
Disjoint With:	'Primary School'
Class: Primary_school	
rdfs: label	**Primary School**
SubClass Of:	Primary Education Level School
Disjoint With:	'Pre-primary School'
Class: Secondary_edu_school	
rdfs: label	**Secondary Education Level School**
SubClass Of:	School
	isSupervisedBy only 'Directory of Secondary Education'
Equivalent To:	'General Upper Secondary School' or 'Lower Secondary School' or 'Vocational Upper Secondary School'
Disjoint With:	Primary Education Level School'
Class: General_Upper_Secondary_school	
rdfs: label	**General Upper Secondary School**
SubClass Of:	'Secondary Education Level School'
Disjoint With:	'Lower Secondary School', 'Vocational Upper Secondary School'
Class: Lower_Secondary_school	
rdfs: label	**Lower Secondary School**
SubClass Of:	'Secondary Education Level School'
Disjoint With:	'General Upper Secondary School', 'Vocational Upper Secondary School'
Class: Vocational_Upper_Secondary_school	
rdfs: label	**Vocational Upper Secondary School**
SubClass Of:	'Secondary Education Level School'
Disjoint With:	'Lower Secondary School', 'General Upper Secondary School'

Finally the class Other Organization is used to represent organizations that do not belong to the Body of the Ministry of Education, but it is necessary to record them as they have issued acts in the past that affect the official status of our employees.

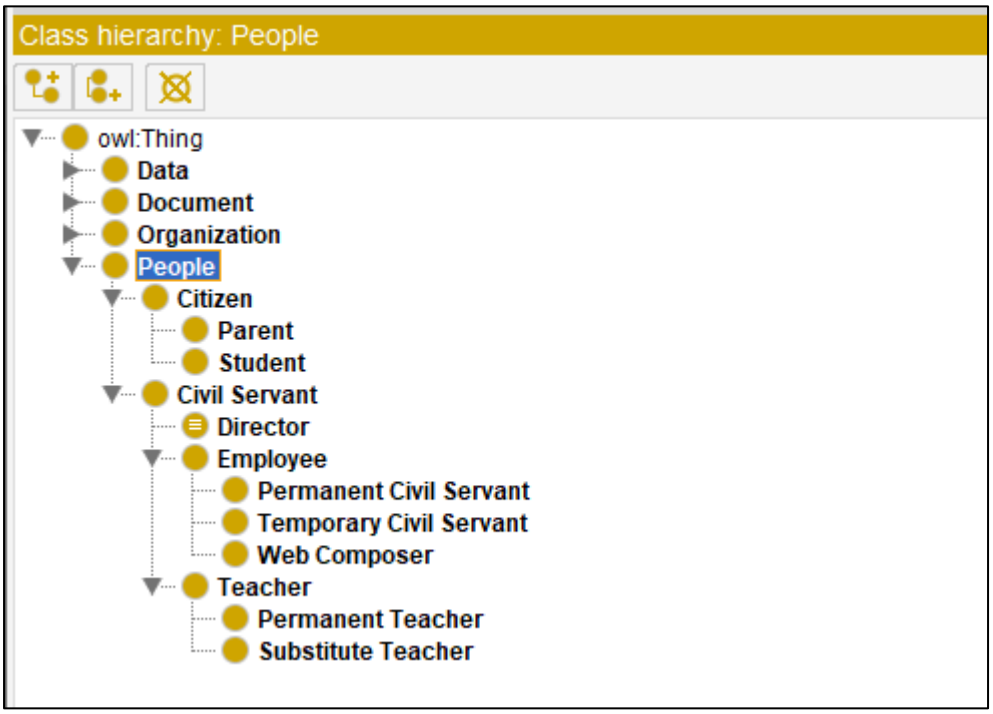
Table 5-4 Class Other Organization

Class: OtherOrg	
rdfs: label	**Other Organization**
SubClass Of:	Organization
Disjoint With:	School, Administration

5.1.2 People

The figure 5.2 presents the class hierarchy of the class People.

Figure 5.2 Class: People



In the class people there are the subclasses Citizen and Civil Servants. Since the vast majority of students attending primary and secondary schools are minors, until adulthood their transactions with public organizations are carried out through their parents. For this reason, the class citizen is divided into Parent and Student subclasses.

Table 5-5 Class People and its subclasses

Class: People	
rdfs: label	**People**
SubClass Of:	Thing
	hasReceived some 'Outgoing Documents'
	hasSigned some Document
Disjoint With:	Data, Document, Organization
Class: Citizen	
rdfs: label	**Citizen**
SubClass Of:	People
Disjoint With:	'Civil Servant'
Class: Parent	
rdfs: label	**Parent**
SubClass Of:	Citizen
	isParentOf some Student
Disjoint With:	Student
Class: Student	
rdfs: label	**Student**
SubClass Of:	Citizen
	hasAttented some Attedance
	isChildOf some Parent
Disjoint With:	Parent

The Civil Servant class contains the subclasses Director, Employee and Teacher. The class Employee refers to civil servants who belong to an Administrative Organizations, while the class Teacher refers to civil servants who belong to a school. Both of these classes are subdivided respectively into two more subclasses to separate permanent civil servants from temporary civil servants. Therefore, the class Teacher has the subclasses Permanent Teacher and Substitute Teacher, while the class Employee has the subclasses Permanent Civil Servant and Temporary Civil Servant. The class Employee has one more subclass called Web Composer that is used for the documents that are issued automatically through the Web Portal.

Table 5-6 Class Civil Servant

Class: CivilServant	
rdfs: label	**Civil Servant**
SubClass Of:	People

	hasComposed some 'Outgoing Documents'
	hasPreviousService some PreviousService
	hasRequestedLeave some Leave
	isEffectedBy some 'Changes in Civil Servant's status'
	isEmployeeOf some Organization
Disjoint With:	Citizen

Table 5-7 Class Employee and its subclasses

Class: Employee	
rdfs: label	**Employee**
SubClass Of:	'Civil Servant'
Disjoint With:	Teacher
Class: Permanent_employee	
rdfs: label	**Permanent Civil Servant**
SubClass Of:	Employee
Disjoint With:	'Web Composer', 'Temporary Civil Servant'
Class: Temporary_employee	
rdfs: label	**Temporary Civil Servant**
SubClass Of:	Employee
Disjoint With:	'Web Composer', 'Permanent Civil Servant'
Class: WebComposer	
rdfs: label	**Web Composer**
SubClass Of:	Employee
Disjoint With:	'Permanent Civil Servant', 'Temporary Civil Servant'

Table 5-8 Class Teacher and its subclasses

Class: Teacher	
rdfs: label	**Teacher**
SubClass Of:	'Civil Servant'
Disjoint With:	Employee
Class: Permanent_teacher	
rdfs: label	**Permanent Teacher**
SubClass Of:	**Teacher**
Disjoint With:	'Substitute Teacher'
Class: Substitute_teacher	
rdfs: label	**Substitute Teacher**
SubClass Of:	**Teacher**
Disjoint With:	'Permanent Teacher'

The class Director represents the civil servants (Employees or Teachers), that are head of an organization and have the authority to sign the issued documents. Since our ontology is oriented towards government to employee support, civil servants play different roles depending on the occasion. In particular, a civil servant may submit an application or process an application of another employee or sign a document as head of an organization.

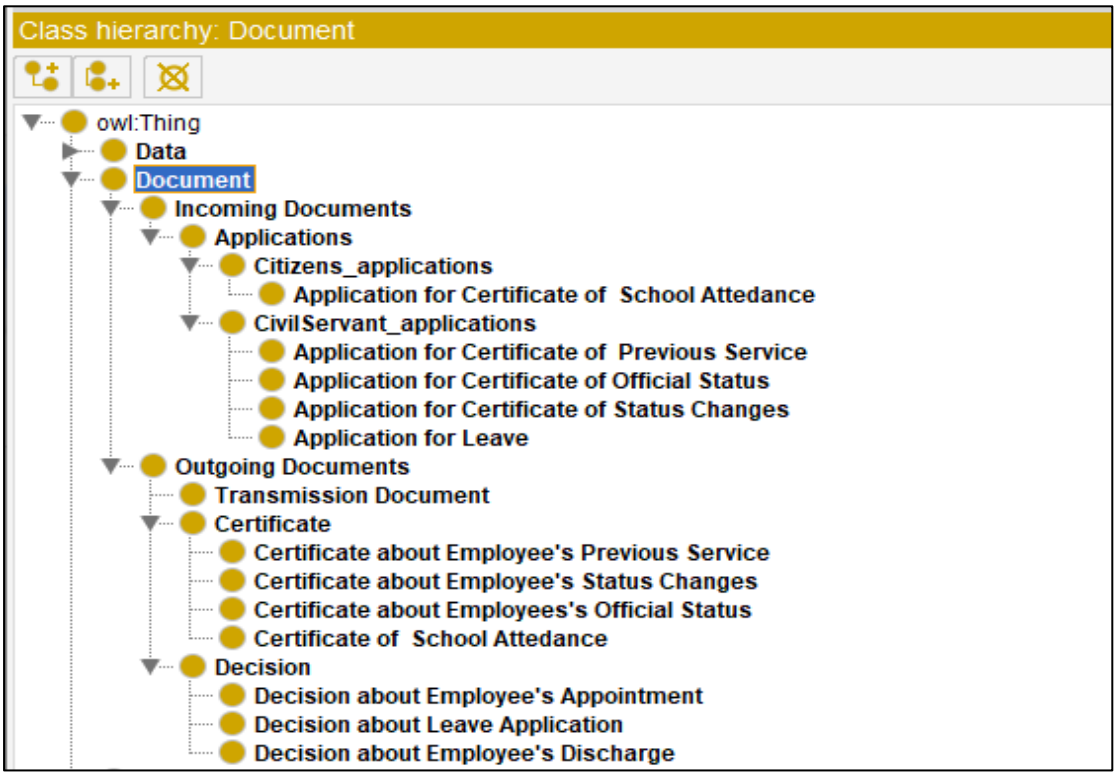
Table 5-9 Class Director

Class: Director	
rdfs: label	**Director**
SubClass Of:	'Civil Servant'
Equivalent To:	isHeadOf some Organization

5.1.3 Document

In the next image we can see the class hierarchy of class Documents.

Figure 5.3 Class:Document



Documents are divided into two major categories which are represented by the respective classes, Incoming Documents and Outgoing Documents. There are many types of Incoming Documents. At this stage, we only use one type, the applications.

Table 5-10 Class Document

Class: Document	
rdfs: label	**Document**
SubClass Of:	Thing
	isSignedBy exactly 1 People
Disjoint With:	Data, Organization, People

Applications are usually the triggers of a procedure and are divided into the subclasses Citizen Applications and Civil Servant Applications. For each of the services we offer, we have created a class that includes the applications for this service. So as a result the class Citizen Applications has the subclass Application for Certificate of School Attedance. The class Civil Servant Applications has four subclasses, Application for Certificate of Previous Service, Application for Certificate of Official Status, Application for Certificate of Status Changes and Application for Leave.

Table 5-11 Class Incoming Document and its subclasses

Class: Incoming_doc	
rdfs: label	**Incoming Documents**
SubClass Of:	Document
	isForwardedTo some Organization
Disjoint With:	'Outgoing Documents'
Class: Applications	
rdfs: label	**Applications**
SubClass Of:	Incoming Documents
Class: Citizens_applications	
rdfs: label	**Citizens_applications**
SubClass Of:	Applications
	isSignedBy some Citizen
Disjoint With:	CivilServant_applications
Class: CertificateOfAttedance_application	
rdfs: label	**Application for Certificate of School Attedance**
SubClass Of:	Citizens_applications

Class: CivilServant_applications	
rdfs: label	**CivilServant_applications**
SubClass Of:	Applications
	isSignedBy some 'Civil Servant'
Disjoint With:	Citizens_applications
Class: ServiceCertiface_application	
rdfs: label	**Application for Certificate of Previous Service**
SubClass Of:	CivilServant_applications
	isProccessedBy only 'Certificate about Employee's Previous Service'
Disjoint With:	'Application for Leave', 'Application for Certificate of Status Changes', 'Application for Certificate of Official Status'
Class: OfficialStatus_application	
rdfs: label	**Application for Certificate of Official Status**
SubClass Of:	CivilServant_applications
	isProccessedBy only 'Certificate about Employees's Official Status'
Disjoint With:	'Application for Leave', 'Application for Certificate of Previous Service', 'Application for Certificate of Status Changes'
Class: StatusChanges_application	
rdfs: label	**Application for Certificate of Status Changes**
SubClass Of:	CivilServant_applications
	isProccessedBy only 'Certificate about Employee's Status Changes'
Disjoint With:	'Application for Leave', 'Application for Certificate of Previous Service', 'Application for Certificate of Official Status'
Class: Leave_application	
rdfs: label	**Application for Leave**
SubClass Of:	CivilServant_applications
	isProccessedBy only 'Decision about Leave Application'
	isRequestingFor some Leave
Disjoint With:	'Application for Certificate of Previous Service', 'Application for Certificate of Status Changes', 'Application for Certificate of Official Status'

The types of outgoing documents that we have chosen to represent, are mainly the derivatives of the procedures and more specifically the Certificates and the Decisions for which we created the respective subclasses. More specifically, the Certificate class has four subclasses, the Certificate about Employee's Previous Service, the Certificate about Employee's Status

Changes, the Certificate about Employees's Official Status and the Certificate of School Attedance. The Decision class has three subclasses, the Decision about Employee's Appointment, the Decision about Employee's Discharge and the Decision about Leave Application. We have also created the subclass of the Transmission Document for future use, as it is a basic type of document used in Public Administration.

Table 5-12 Class Outgoing Document and its subclasses

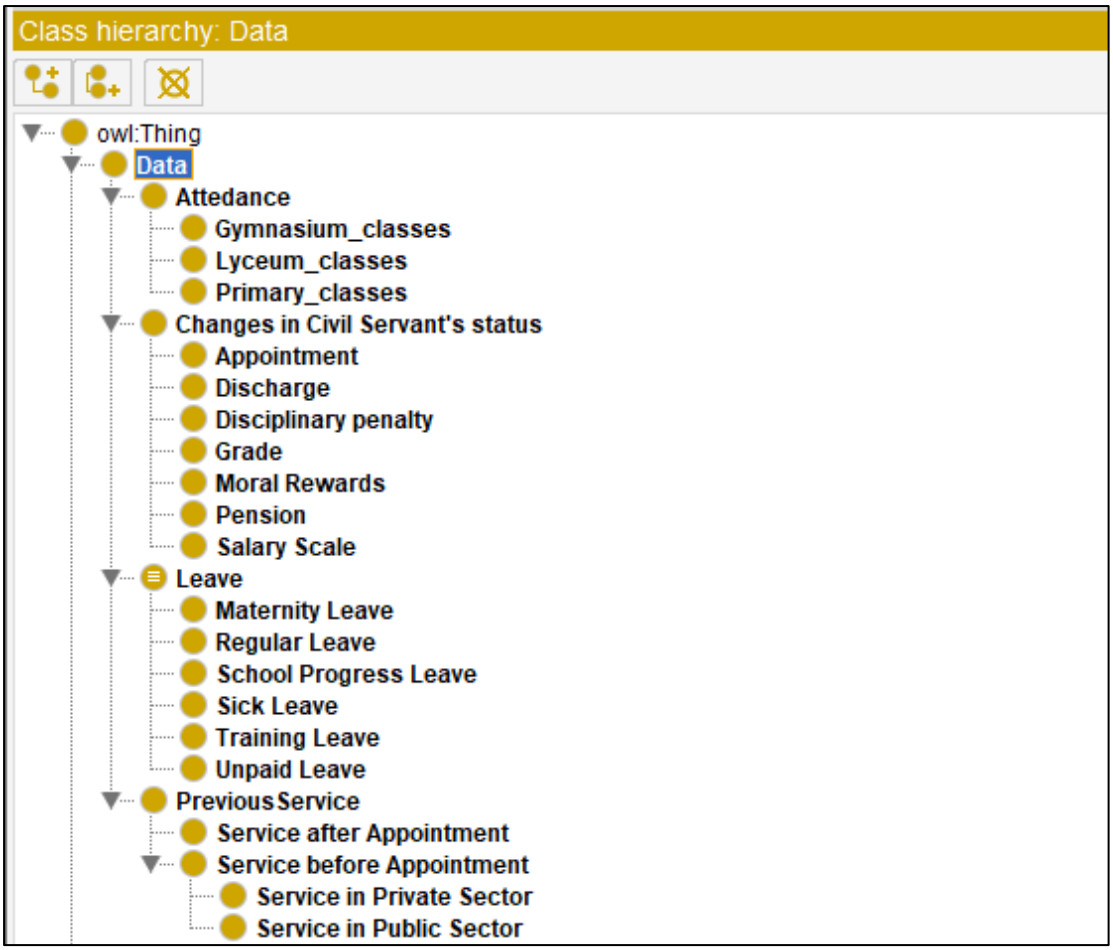
Class: Outgoing_doc	
rdfs: label	**Outgoing Documents**
SubClass Of:	Document
	isComposedBy only 'Civil Servant'
	isForwardedTo some (Organization or People)
	isIssuedBy exactly 1 Organization
	isSignedBy only Director
Disjoint With:	'Incoming Documents'
Class: Certificate	
rdfs: label	**Outgoing Documents**
SubClass Of:	Outgoing Documents
Disjoint With:	Decision, 'Transmission Document'
Class: PreviousService_Certificate	
rdfs: label	**Certificate about Employee's Previous Service**
SubClass Of:	Certificate
	hasProccessed only 'Application for Certificate of Previous Service'
	hasReffered only PreviousService
Disjoint With:	'Certificate of School Attedance', 'Certificate about Employees's Official Status', 'Certificate about Employee's Status Changes'
Class: StatusChanges_certificate	
rdfs: label	**Certificate about Employee's Status Changes**
SubClass Of:	Certificate
	hasProccessed only 'Application for Certificate of Status Changes'
Disjoint With:	'Certificate about Employee's Previous Service', 'Certificate of School Attedance', 'Certificate about Employees's Official Status'
Class: OfficialStatus_certificate	
rdfs: label	**Certificate about Employees's Official Status**
SubClass Of:	Certificate
	hasProccessed only 'Application for Certificate of Official Status'

Disjoint With:	'Certificate about Employee's Previous Service', 'Certificate of School Attedance', 'Certificate about Employee's Status Changes'
Class: attendance_certificate	
rdfs: label	**Certificate of School Attedance**
SubClass Of:	Certificate
	hasProccessed only 'Application for Certificate of School Attedance'
Disjoint With:	'Certificate about Employee's Previous Service', 'Certificate about Employees's Official Status', 'Certificate about Employee's Status Changes'
Class: Decision	
rdfs: label	**Decision**
SubClass Of:	'Outgoing Documents'
Disjoint With:	Certificate, 'Transmission Document'
Class: Appointment_decision	
rdfs: label	**Decision about Employee's Appointment**
SubClass Of:	Decision
Disjoint With:	'Decision about Leave Application', 'Decision about Employee's Discharge'
Class: Discharge_decision	
rdfs: label	**Decision about Employee's Discharge**
SubClass Of:	Decision
Disjoint With:	'Decision about Leave Application', 'Decision about Employee's Appointment'
Class: Leave_decision	
rdfs: label	**Decision about Leave Application**
SubClass Of:	Decision
	hasProccessed only 'Application for Leave'
	hasReffered only Leave
Disjoint With:	'Decision about Employee's Appointment', 'Decision about Employee's Discharge'
Class: TransmissionDoc	
rdfs: label	**Transmission Document**
SubClass Of:	'Outgoing Documents'
	hasProccessed only 'Application for Certificate of School Attedance'
Disjoint With:	Certificate, Decision

5.1.4 Data

The following φιγθρε presents the class hierarchy of the class Data.

Figure 5.4 Class Data



In the Data class, we model the data we manage and are necessary for their processing and the production of outgoing documents. The data of each class is used to handle more than one process and this modeling can be used in future system extensions.

Table 5-13 Class Data

Class: Data	
rdfs: label	**Data**
SubClass Of:	Thing
	isRefferedIn some 'Outgoing Documents'
Disjoint With:	Document, Organization, People

A key element in serving the citizens is the registration of student attendance. In our implementation we use these data for the issuing of the Certificate of School Attendance, but these data can be utilized in future extensions of the ontology and the production of more documents like the Certificate of completion of studies and the Certificate of Acquisition of Citizenship. The class Attendance is divided in three subclasses, the Primary_classes, the Gymnasium_classes and the Lyceum_classes.

Table 5-14 Class Attendance and its subclasses

Class: Attendance	
rdfs: label	**Attendance**
SubClass Of:	Data
Disjoint With:	'Changes in Civil Servant's status', PreviousService, Leave
Class: Primary_classes	
rdfs: label	**Primary Classes**
SubClass Of:	Attendance
Disjoint With:	'Upper Secondary Classes', 'Lower Secondary Classes'
Class: Gymnasium_classes	
rdfs: label	**Lower Secondary Classes**
SubClass Of:	Attendance
Disjoint With:	'Upper Secondary Classes', 'Primary Classes'
Class: Lyceum_classes	
rdfs: label	**Upper Secondary Classes**
SubClass Of:	Attendance
Disjoint With:	'Lower Secondary Classes', 'Primary Classes'

The Changes in Civil Servant's status class contains the data that are kept in the personal records of the employees. These data are useful for the issuing of almost all documents for Civil Servants. They contain information that are referred in them, like the appointment date, the grade etc. The subclasses of this class are Appointment, Discharge, Disciplinary penalty, Grade, Moral Rewards, Pension and Salary Scale.

Table 5-15 Class Changes and its subclasses

Class: Changes	
rdfs: label	**Changes in Civil Servant's status**
SubClass Of:	Data
	hasEffectOn some 'Civil Servant'
Disjoint With:	Attendance, PreviousService, Leave
Class: Appointment_change	
rdfs: label	**Appointment**
SubClass Of:	Changes in Civil Servant's status
Disjoint With:	Grade, 'Salary Scale', 'Disciplinary penalty', 'Moral Rewards', Pension, Discharge
Class: Discharge_change	
rdfs: label	**Discharge**
SubClass Of:	Changes in Civil Servant's status
Disjoint With:	Grade, 'Salary Scale', 'Disciplinary penalty', 'Moral Rewards', Pension, Appointment
Class: DisciplinaryPenalty_change	
rdfs: label	**Disciplinary penalty**
SubClass Of:	Changes in Civil Servant's status
Disjoint With:	Grade, 'Salary Scale', 'Moral Rewards', Pension, Appointment, Discharge
Class: Grade_change	
rdfs: label	**Grade**
SubClass Of:	Changes in Civil Servant's status
Disjoint With:	'Salary Scale', 'Disciplinary penalty', 'Moral Rewards', Pension, Appointment, Discharge
Class: Rewards _change	
rdfs: label	**Moral Rewards**
SubClass Of:	Changes in Civil Servant's status
Disjoint With:	Grade, 'Salary Scale', 'Disciplinary penalty', Pension, Appointment, Discharge
Class: Pension_change	
rdfs: label	**Pension**
SubClass Of:	Changes in Civil Servant's status
Disjoint With:	Grade, 'Salary Scale', 'Disciplinary penalty', 'Moral Rewards', Appointment, Discharge
Class: SalaryScale_change	
rdfs: label	**Salary Scale**

SubClass Of:	Changes in Civil Servant's status
Disjoint With:	Grade, 'Disciplinary penalty', 'Moral Rewards', Pension, Appointment, Discharge

The Previous Service class contains information related to the services of the staff before and after the appointment. The Previous Service class is divided into the Service after Appointment class and the Service before Appointment class. The latter is divided more into the subclasses Service in Private Sector and Service in Public Sector.

Table 5-16 Previous Service and its subclasses

Class: PreviousService	
rdfs: label	**Previous Service**
SubClass Of:	Data
	wasOfferedBy some 'Civil Servant'
	wasProvidedIn some Organization
Disjoint With:	Attendance, 'Changes in Civil Servant's status', Leave
Class: AfterAppointment_service	
rdfs: label	**Service after Appointment**
SubClass Of:	'Previous Service'
Disjoint With:	'Service before Appointment'
Class: BeforeAppointment_service	
rdfs: label	**Service before Appointment**
SubClass Of:	'Previous Service'
Equivalent To:	'Service in Private Sector' or 'Service in Public Sector'
Disjoint With:	'Service after Appointment'
Class: PrivateSector_service	
rdfs: label	**Service in Private Sector**
SubClass Of:	'Service before Appointment'
Disjoint With:	'Service in Public Sector'
Class: PublicSector_service	
rdfs: label	**Service in Public Sector**
SubClass Of:	'Service before Appointment'
Disjoint With:	'Service in Private Sector'

Finally, the Leave class is used to categorize the types of leaves. It has seven subclasses, the Maternity Leave class, the Regular Leave class, the School Progress Leave class, the Sick Leave class, the Training Leave class and the Unpaid Leave class.

Table 5-17 Class Leave and its subclasses

Class: Leave	
rdfs: label	**Leave**
SubClass Of:	isRequestedWith exactly 1 'Application for Leave'
Equivalent To:	Data and ('Maternity Leave' or 'Regular Leave' or 'School Progress Leave' or 'Sick Leave' or 'Training Leave' or 'Unpaid Leave')
Disjoint With:	Attendance, 'Changes in Civil Servant's status', PreviousService
Class: Maternity_leave	
rdfs: label	**Maternity Leave**
SubClass Of:	Leave
Disjoint With:	'School Progress Leave', 'Regular Leave', 'Training Leave', 'Sick Leave', 'Unpaid Leave'
Class: Regular_leave	
rdfs: label	**Regular Leave**
SubClass Of:	Leave
Disjoint With:	'School Progress Leave', 'Training Leave', 'Maternity Leave', 'Sick Leave', 'Unpaid Leave'
Class: SchoolProgres_leave	
rdfs: label	**School ProgresLeave**
SubClass Of:	Leave
Disjoint With:	'Regular Leave', 'Training Leave', 'Maternity Leave', 'Sick Leave', 'Unpaid Leave'
Class: Sick_leave	
rdfs: label	**Sick Leave**
SubClass Of:	Leave
Disjoint With:	'School Progress Leave', 'Regular Leave', 'Training Leave', 'Maternity Leave', 'Unpaid Leave'
Class: Training_leave	
rdfs: label	**Training Leave**
SubClass Of:	Leave
Disjoint With:	'School Progress Leave', 'Regular Leave', 'Maternity Leave', 'Sick Leave', 'Unpaid Leave'
Class: Unpaid_leave	
rdfs: label	**Unpaid Leave**
SubClass Of:	Leave
Disjoint With:	'School Progress Leave', 'Regular Leave', 'Training Leave', 'Maternity Leave', 'Sick Leave'

5.2 Object Properties

In the following tables, we present the object properties that show the relation between our entities. The Object Properties are grouped according to their usage:

Organization Object Properties

Table 5-18 Organization Object Properties

Object Property	Domain	Range	Inverse Of	Usage
supervises	Administration	Administration or School	isSupervisedBy	Used for the depiction of Organization's hierarchy
isSupervisedBy	Administration or School	Administration	supervises	Used for the depiction of Organization's hierarchy
hasHead	Organization	Director	isHeadOf	Defines the head of an organization
hasEmployee	Organization	Civil Servant	isEmployeeOf	Assign an employee to an organization
hasIssued	Organization	Document	isIssuedBy	Connects a document with the issuing organization

People Object Properties

Table 5-19 People Object Properties

Object Property	Domain	Range	Inverse Of	Usage
hasSigned	People	Document	isSignedBy	Assigns the person citizen or civil servant that signed a document (incoming or outgoing)

Citizen Object Properties

Table 5-20 Citizen Object Properties

Object Property	Domain	Range	Inverse Of	Usage
hasAttented	Student	Attendance	-	Connects the student with the classes that has completed
isParentOf	Parent	Student	isChildOf	Matches Parent with Student
isChildOf	Student	Parent	isParentOf	Matches Parent with Student

Civil Servant Object Properties

Table 5-21 Civil Servant Object Properties

Object Property	Domain	Range	Inverse Of	Usage
isHeadOf	Organization	Director	hasHead	Defines the head of an organization
isEmployeeOf	Civil Servant	Organization	hasEmployee	Assign an employee to an organization
hasRequestedLeave	Civil Servant	Leave	isRequestedFrom	Connects Civil Servant with the requested leave
hasPreviousService	Civil Servant	Previous Service	wasOfferedBy	Assigns previous service to an employee
hasEffectOn	Changes in C.Serv status	Civil Servant	isEffectedBy	Connects a Civil Servant with his changes
isEffectedBy	Civil Servant	Changes in C.Serv status	hasEffectOn	Connects a Civil Servant with his changes

Employee Object Properties

Table 5-22 Employee Object Properties

Object Property	Domain	Range	Inverse Of	Usage
hasComposed	Civil Servant	Outgoing Documents	isComposedBy	Matches the composer of a document with the document

Document Object Properties

Table 5-23 Document Object Properties

Object Property	Domain	Range	Inverse Of	Usage
hasProccessed	Outgoing Documents	Incoming Documents	isProccessedBy	Connects the trigger document with the produced document
isProccessedBy	Incoming Documents	Outgoing Documents	hasProccessed	Connects the trigger document with the produced document
hasReceived	Organization or People	Document	isForwardedTo	Shows the flow of the documents
isForwardedTo (isSubmittedTo)	Document	Organization or People	hasReceived	Shows the flow of the documents

hasReffered	Outgoing Documents	Data	isRefferedIn	Assigns the data that are used in an outgoing document
isRefferedIn	Data	Outgoing Documents	hasReffered	Assigns the data that are used in an outgoing document
isComposedBy	Outgoing Documents	Civil Servant	hasComposed	Matches the composer of a document with the document
isIssuedBy	Document	Organization	hasIssued	Connects a document with the issuing organization
isSignedBy	Document	People	hasSigned	Assigns the person citizen or civil servant that signed a document (incoming or outgoing)

Leave Object Properties

Table 5-24 Leave Object Properties

Object Property	Domain	Range	Inverse Of	Usage
isRequestedFrom	Leave	Civil Servant	hasRequestedLeave	Connects Civil Servant with the requested leave
isRequestedWith	Leave	Application for Leave	isRequestingFor	Assigns a leave to an application
isRequestingFor	Application for Leave	Leave	isRequestingFor	Assigns a leave to an application

Previous Service Object Properties

Table 5-25 Previous Service Object Properties

Object Property	Domain	Range	Inverse Of	Usage
hasReceivedService	Organization	Previous Service	wasProvidedIn	Connects a previous service with an organization
wasProvidedIn	Previous Service	Organization	hasReceivedService	Connects a previous service with an organization
wasOfferedBy	Previous Service	Civil Servant	hasPreviousService	Assigns previous service to an employee

5.3 Data Properties

In this section, we define the data properties of our Ontology. We have also group them according to their usage:

Organization Data Properties

Table 5-26 Organization Data Properties

Data Property	Domain	Range	Usage
hasOrgId	Organization	xsd:integer	**Organization's Id**
hasTitle	Organization	xsd:string	**Organization's title**
hasAddress	Organization	xsd:string	**Organization's Address**
hasTel	Organization	xsd:long	**Organization's Telephone**
hasFax	Organization	xsd:long	**Organization's Fax**
hasEmail	Organization	xsd:string	**Organization's Email**
hasWebSite	Organization	xsd:string	**Organization's Web Site**

Document Data Properties

Table 5-27 Document Data Properties

Data Property	Domain	Range	Usage
hasDate	Document	xsd:dateTime	The date of the document
hasRefNum	Document	xsd:integer	The reference number of the document
hasTheme	Document	xsd:string	The theme of the document

People Data Properties

Table 5-28 People Data Properties

Data Property	Domain	Range	Usage
hasSurname	People	xsd:string	Surname
hasFirstName	People	xsd:string	First Name
hasFatherName	People	xsd:string	Father Name
hasMotherName	People	xsd:string	Mother Name
hasIdentityCardNum	People	xsd:string	Identity Card Number
hasTaxIDNum	People	xsd:string	Tax ID Number
hasBirthDate	People	xsd:dateTime	Birth date

Civil Servant Data Properties

Table 5-29 Civil Servant Data Properties

Data Property	Domain	Range	Usage
hasRegistryNumber	Civil Servant	xsd:integer	Employee's Registry Number
hasEducationLevel	Civil Servant	PE or TE or DE or YE	Employee's Educational Level
hasSpeciality	Civil Servant	xsd:string	Employee's Specialization

Application Data Properties

Table 5-30 Application Data Properties

Data Property	Domain	Range	Usage
hasAppStatus	Applications	pending or proccessed	The status of the application

Civil Servant's Changes Data Properties

The Data Properties hasChangeDate, hasDecisionAuthority, hasDecisionDate and hasDecisionRefNum are common to all types of changes. The rest properties depend on the type of the change.

Table 5-31 Civil Servant's Changes Data Properties

Data Property	Domain	Range	Usage
hasChangeDate	Changes in Civil Servant's status	xsd:dateTime	The date from which the change takes effect
hasDecisionRefNum	Changes in Civil Servant's status	xsd:string	Reference Number of the Decision ratifying the change
hasDecisionDate	Changes in Civil Servant's status	xsd:dateTime	The date of the decision
hasDecisionAuthority	Changes in Civil Servant's status	xsd:string	The body that issued the Decision
hasGovGazzeteNum	Appointment or Discharge or Pension	xsd:string	The Government Gazzete Number only for appointment, discharge and pension changes

hasGrade	Grade	xsd:string	The acquired grade, for the grade change
hasPenaltyType	Disciplinary penalty	xsd:string	Type for the Disciplinary penalty change
hasRewardType	Moral Rewards	xsd:string	Type for the Moral Rewards change
hasSalaryScale	Salary Scale	xsd:short	The acquired scale, for the salary scale change

Previous Service Data Properties

Table 5-32 Previous Service Data Properties

Data Property	Domain	Range	Usage
hasServiceTitle	Previous Service	xsd:string	The title describes the type of the previous service
hasServiceStartDate	Previous Service	xsd:dateTime	Starting Date
hasServiceEndDate	Previous Service	xsd:dateTime	Ending Date
hasServiceTotalTime	Previous Service	xsd:duration	Service duration
hasServiceHeadPosition	Previous Service	xsd:boolean	This is applicable only for services in the public sector

Leave Data Properties

Table 5-33 Leave Data Properties

Data Property	Domain	Range	Usage
hasLeaveStartDate	Leave	xsd:dateTime	Starting Date of the leave
hasLeaveEndDate	Leave	xsd:dateTime	Ending Date of the leave
hasLeaveNumOfDays	Leave	xsd:short	Leaves' duration
hasLeaveStatus	Leave	Approved or pending or rejected	The status of the leave

Attendance Data Properties

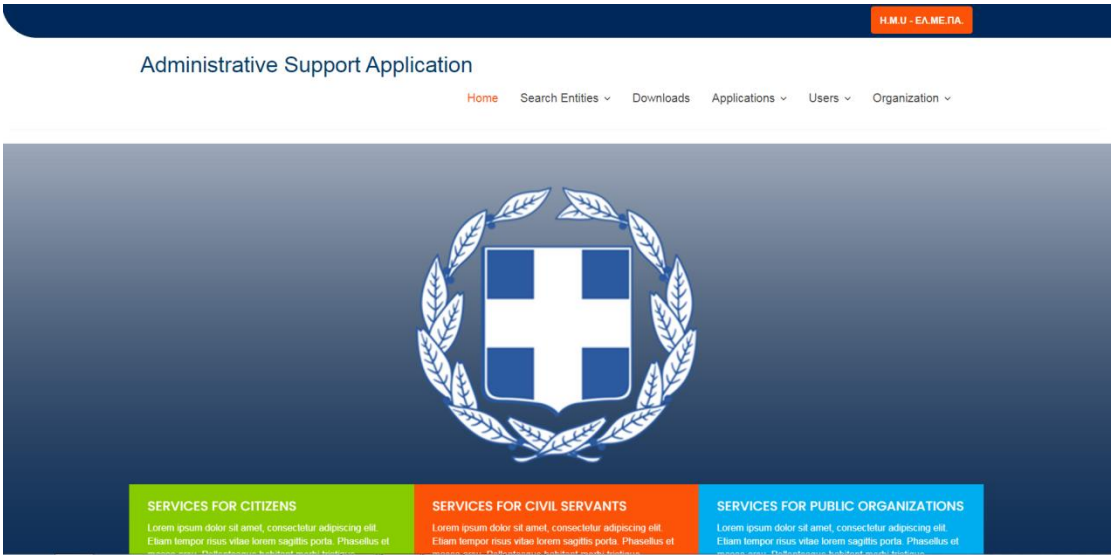
Table 5-34 Attendance Data Properties

Data Property	Domain	Range	Usage
class	Attendance	xsd:string	The attended class
school_year	Attendance	xsd:string	The study year
success	Attendance	xsd:boolean	Whether the class was completed or not

Chapter 6 The Web Portal

The purpose of the portal is to provide stage 1 to 4 services to citizens and employees. We also use a SPARQL Server to manage the data. In this context, we developed the interfaces and the functionality for the connection with the SPARQL endpoint. We created mechanisms for users' login and displaying their personal data, through their Personal Repository At the same time, we developed the relevant interfaces (Organization's Dashboard) for data management on the part of Administration.

Figure 6.1 Web Portal (Homr Page)



6.1 E-government's stages of Services in the Web Portal

Stage 1 (Information)

The web portal contains basic citizen information services. In general, providing information at this stage of services is not very complex. What is interesting is the data retrieval that is performed with SPARQL queries to the Apache Jena Endpoint. From the options menu, users can select the type of information that will be displayed to them.

Stage 2 (Interaction)

An additional feature offered through the 2nd stage services is one-way interaction. Users have the opportunity to interact with the portal by downloading documents and information which they can store on their computer and use for their physical transaction with the Organizations. For the services of this stage there is no communication with the SPARQL Endpoint and all management is accomplished through the Website.

Stage 3 (Two-way Interaction)

The Web Portal supports two-way interaction services by offering to users registration functions. The users can register by providing basic personal information and after the administration's approval, they have access to more online services. After their successful registration they can submit applications to the Administration. For submitting applications communication is necessary with the SPARQL Endpoint for retrieval of the necessary parameters.

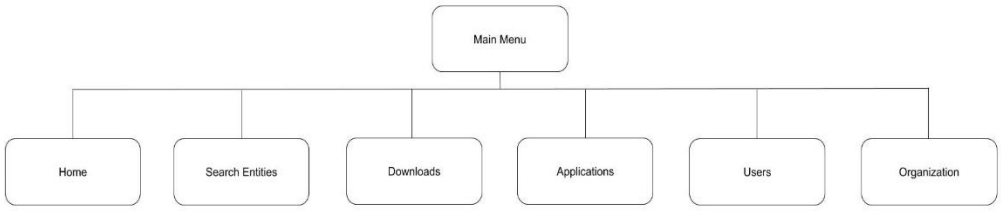
Stage 4 (Transaction)

Stage 4 services are provided through the user's personal repository. The personal repository contains the employee or teacher's electronic file. The user has access to all the data kept by the administration in the file of the Organization. He also has access to his request history and can check the status of his applications. In addition he can download on his computer, the certificates and decisions for which he has applied.

6.2 Web Portal Functions

Some of the functions provided by the Web-portal are available to users without logging in, while others require a login. In the figure below, we see the options provided by the main menu to users.

Figure 6.2 Main Menu Options

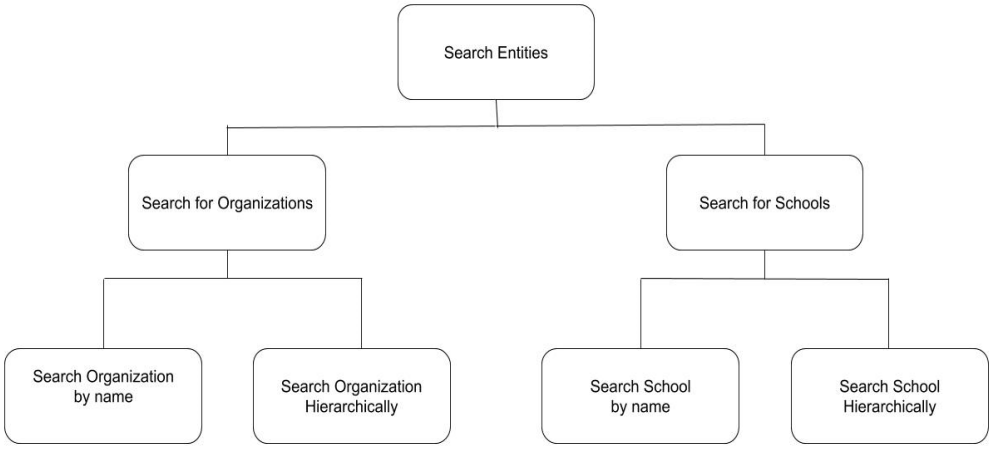


6.2.1 Search for Entities

This function is part of stage 1 services (Information). With this function, the user can search for information related to the decentralized administrative organizations and the school units of Greece. The search can be performed either by the name of the organization or school, or by selecting an organization and find information about its supervised bodies (organizations or schools). To search for the data, the website compiles a SPARQL query, according to the criteria defined by the user. The query is submitted to the SPARQL Endpoint which returns the results to the Web-Portal, which undertakes to present them to the user.

In the following figure, we see the diagrammatic representation of the user's options

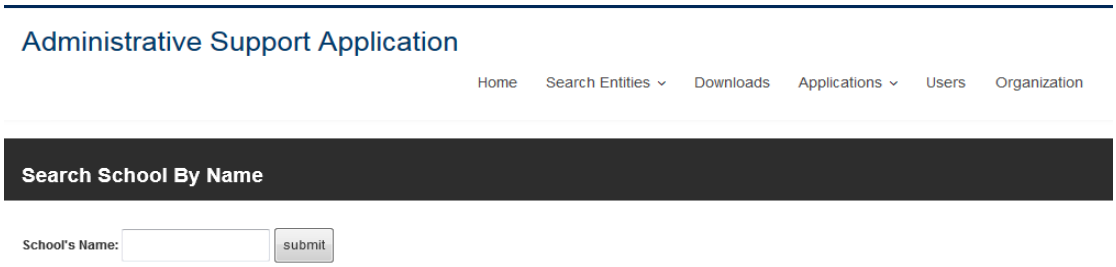
Figure 6.3 Menu item Search Entities



- **Search School by Name**

With this option, the user can search information of a school by entering its name or a part of it.

Figure 6.4 Search School by name



- **Search School Hierarchically**

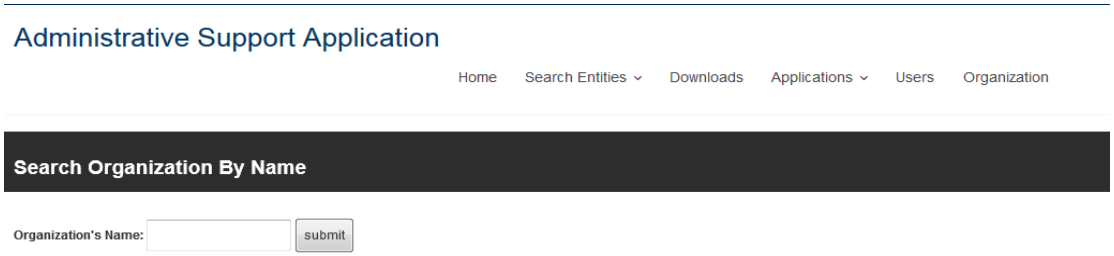
With the hierarchical search, we can see the school units that are supervised by an organization. The search is carried out in three steps in which the user must fill in the Regional Directorate, the Directorate of Education and the type of school he is interested in. Each of the steps includes communication with the SPARQL Endpoint, to provide the necessary data. At the beginning of the hierarchical search (step 1), the first communication with the Endpoint takes place, which returns the Regional Directorates. The user selects the Regional Directorate he desires and submits it to the endpoint. The user may also select all of them by choosing the option All Regional Directorates. In the second step, the endpoint returns the Directorates of Education under the Regional Directorate selected in the previous step. The user selects the Directorate he desires or all of them with the relevant choice “All Directorates”, and his choice is submitted to the Endpoint. Finally, at the third step the Endpoint returns the available School Types. After the user selects the School Type, the final query is submitted and the Endpoint returns the results that are displayed to the user.

Figure 6.5 Search School Hierarchically

- **Search Organization by Name**

With this option, the user can search information of an organization by entering its name or a part of it.

Figure 6.6 Search Organization by Name

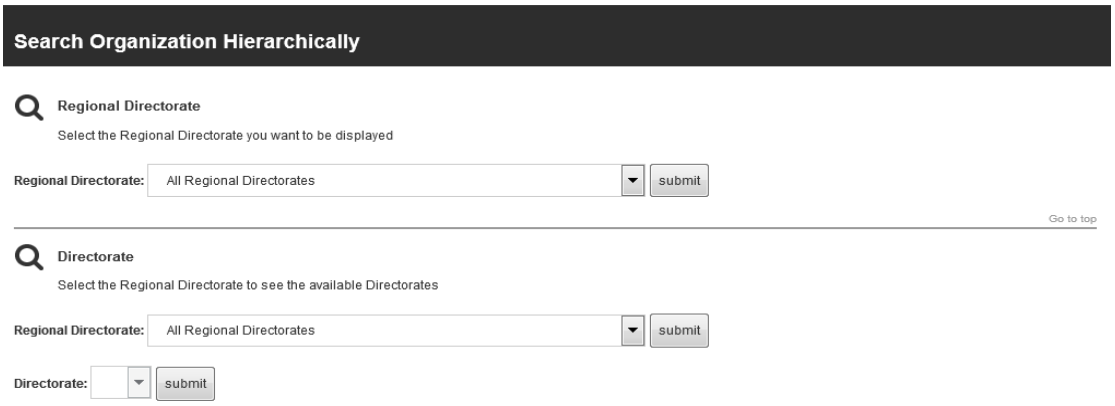


- **Search Organization Hierarchically**

The search for organizations results in the presentation of data for Regional Directories and Directories. For the convenience of users this page is divided into two search areas. In the first, we can display the data of the Regional Directorates that we will select from the dropdown list. For the appearance of Directorates, we first need to select the Regional Directorate, which is in charge of the Directorates, whose details will be displayed.

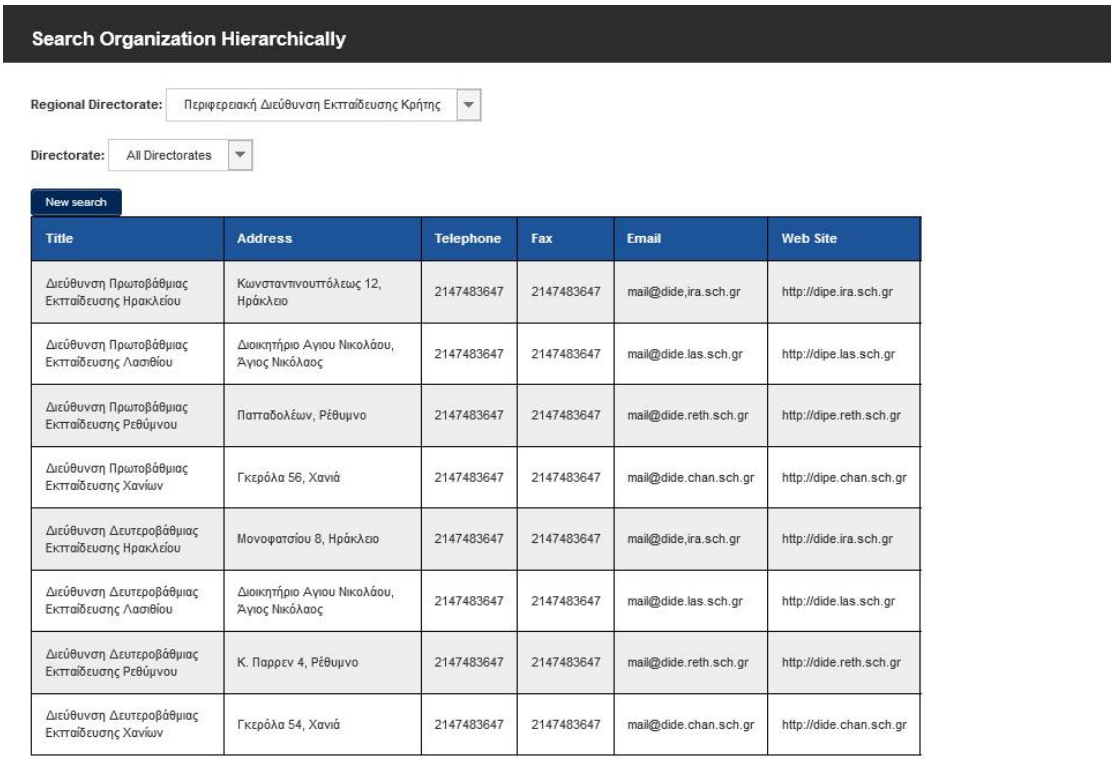
Upon the page loading, a request is submitted to the SPARQL Server for fetching the information about the Regional Directorates, that are necessary for completing the search form.

Figure 6.7 Search Organization Hierarchically



The following figure shows the search results performed through the menu item Search Organization Hierarchically, where we requested information about the organizations that are supervised by the Regional Directorate of Education of Crete.

Figure 6.8 Search results



Search Organization Hierarchically

Regional Directorate: Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης

Directorate: All Directorates

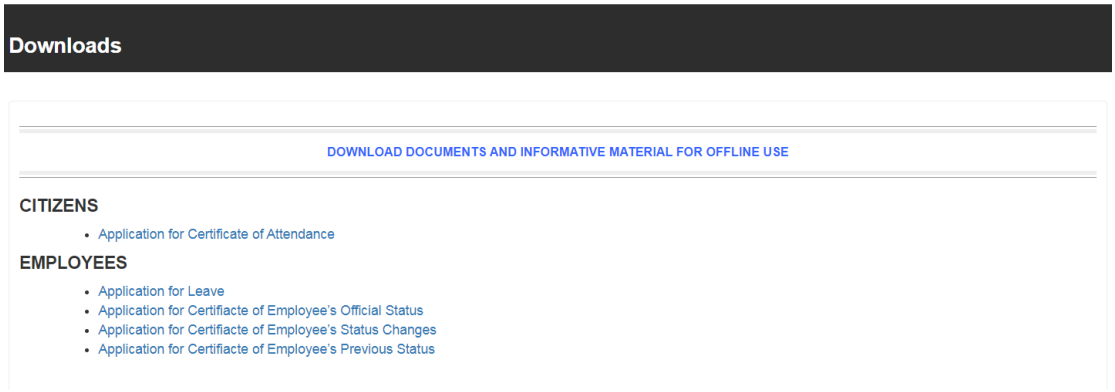
New search

Title	Address	Telephone	Fax	Email	Web Site
Διεύθυνση Πρωτοβάθμιας Εκπαίδευσης Ηρακλείου	Κωνσταντινουπόλεως 12, Ηράκλειο	2147483647	2147483647	mail@dide,ira.sch.gr	http://dipe.ira.sch.gr
Διεύθυνση Πρωτοβάθμιας Εκπαίδευσης Λασιθίου	Διοικητήριο Αγιου Νικολάου, Άγιος Νικόλαος	2147483647	2147483647	mail@dide.las.sch.gr	http://dipe.las.sch.gr
Διεύθυνση Πρωτοβάθμιας Εκπαίδευσης Ρεθύμνου	Πατταδολέων, Ρέθυμνο	2147483647	2147483647	mail@dide.reth.sch.gr	http://dipe.reth.sch.gr
Διεύθυνση Πρωτοβάθμιας Εκπαίδευσης Χανίων	Γκερόλα 56, Χανιά	2147483647	2147483647	mail@dide.chan.sch.gr	http://dipe.chan.sch.gr
Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης Ηρακλείου	Μονοφατσίου 8, Ηράκλειο	2147483647	2147483647	mail@dide,ira.sch.gr	http://dide.ira.sch.gr
Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης Λασιθίου	Διοικητήριο Αγιου Νικολάου, Άγιος Νικόλαος	2147483647	2147483647	mail@dide.las.sch.gr	http://dide.las.sch.gr
Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης Ρεθύμνου	Κ. Παρρεν 4, Ρέθυμνο	2147483647	2147483647	mail@dide.reth.sch.gr	http://dide.reth.sch.gr
Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης Χανίων	Γκερόλα 54, Χανιά	2147483647	2147483647	mail@dide.chan.sch.gr	http://dide.chan.sch.gr

6.2.2 Downloads

With this option, users have access to stage 2 services (Interaction). Here, they can download documents (e.g. applications) and other informative material for offline use.

Figure 6.9 Menu item Downloads



Downloads

DOWNLOAD DOCUMENTS AND INFORMATIVE MATERIAL FOR OFFLINE USE

CITIZENS

- Application for Certificate of Attendance

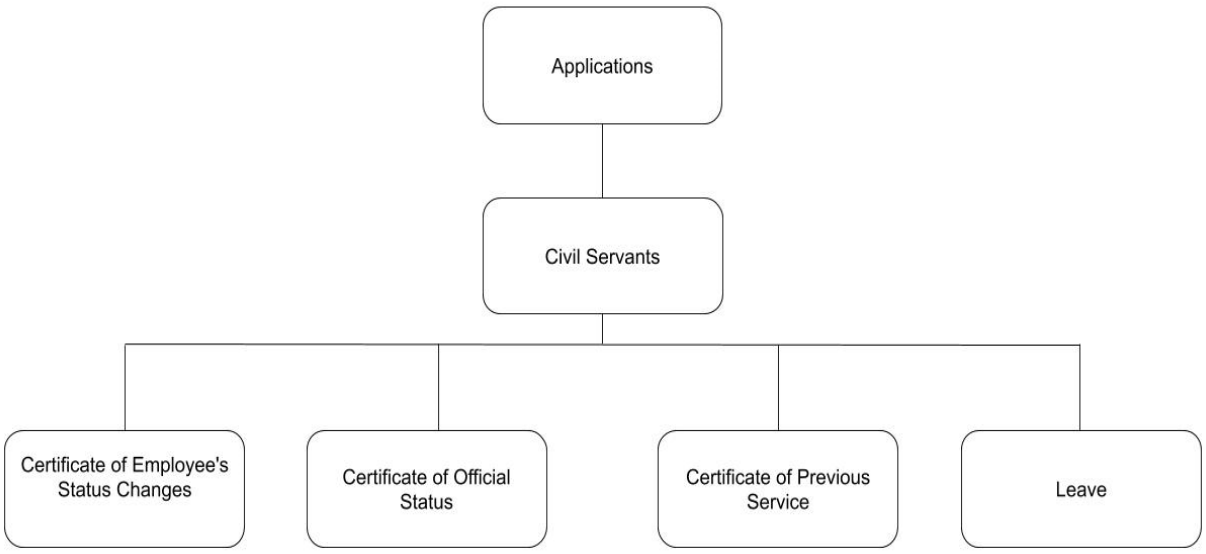
EMPLOYEES

- Application for Leave
- Application for Certifiacte of Employee's Official Status
- Application for Certifiacte of Employee's Status Changes
- Application for Certifiacte of Employee's Previous Status

6.2.3 Applications

The application options are displayed to the user after logging in. The purpose of this is each user depending on his role to access the services that concern him. Online applications are part of stages 3 and 4 services. The user may apply for Certificate of Employee's Status Changes, Certificate of Official Status, Certificate of Previous Service, and Leave. In the following figure we see the diagrammatic representation of the user's options

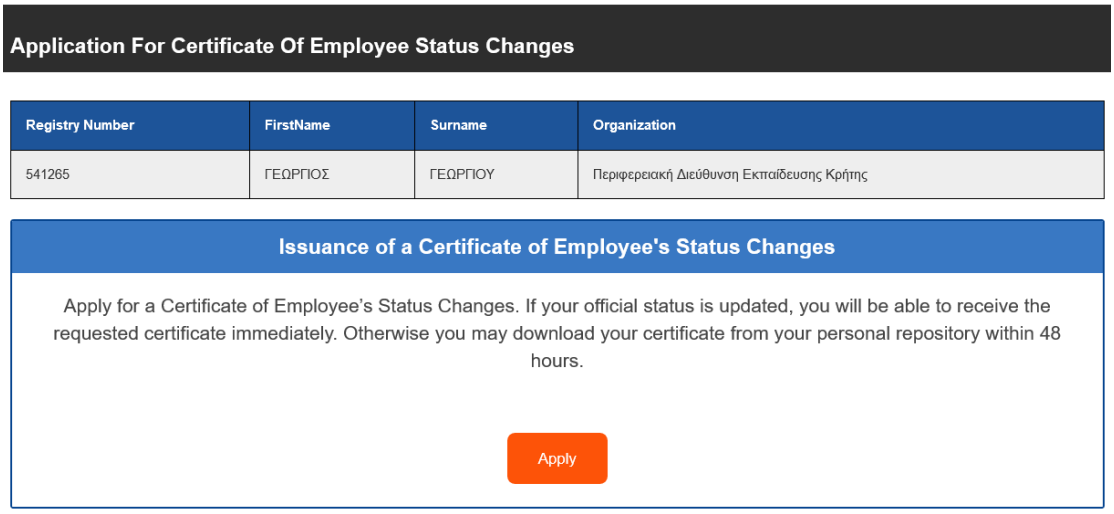
Figure 6.10 Menu item Applications



- **Application for Certificate of Employee Status Changes**

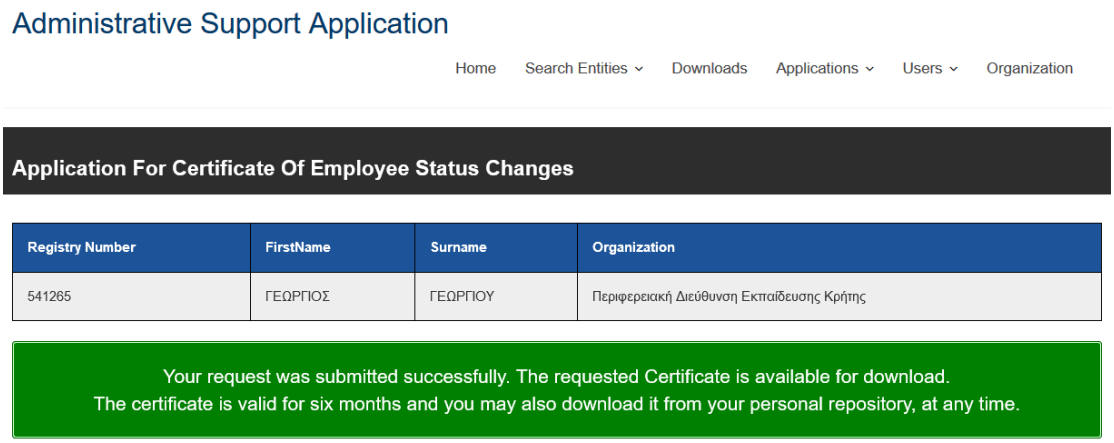
The procedure for the issuance of Certificate of Employee Status Changes is an automated process and an administration’s representative intervention is not required for its completion. The user selecting the menu option for the submission of the certificate is transferred to the screen we see the figure 6.11.

Figure 6.11 Application for Certificate of Employee Status Changes



On this screen, the user is informed about the procedure to be followed to receive the requested certificate and is asked to confirm his intention. If the user proceeds with the submission by clicking the apply button, the application is processed by communicating with the SPARQL Endpoint. If the process is successful, the user receives a corresponding informative message that we see in figure 6.12. The result of the above process is the completion of the issuance of the requested certificate to which the user has access through the Personal Repository, and the administration through the Dashboard.

Figure 6.12 Result of application



- **Application for Certificate of Official Status / Application for Certificate of Service**

The process for applying for and issuing the Certificate of Official Status and Certificate of Previous Service are also automated procedures, which do not require the intervention of the administration and their results are available in the Personal Repository and the Dashboard. These processes are similar to the procedure described above for the Certificate of Employee's Status Changes, so we will not expand on them, and we present them together. In figures 6.13 and 6.14 we can the relevant screenshots.

Figure 6.13 Application for Certificate of Official Status

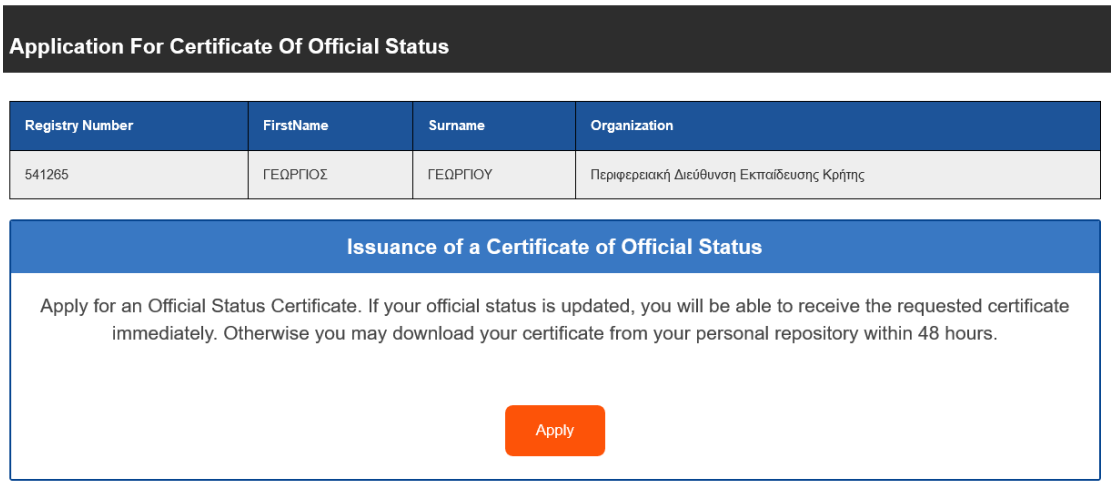
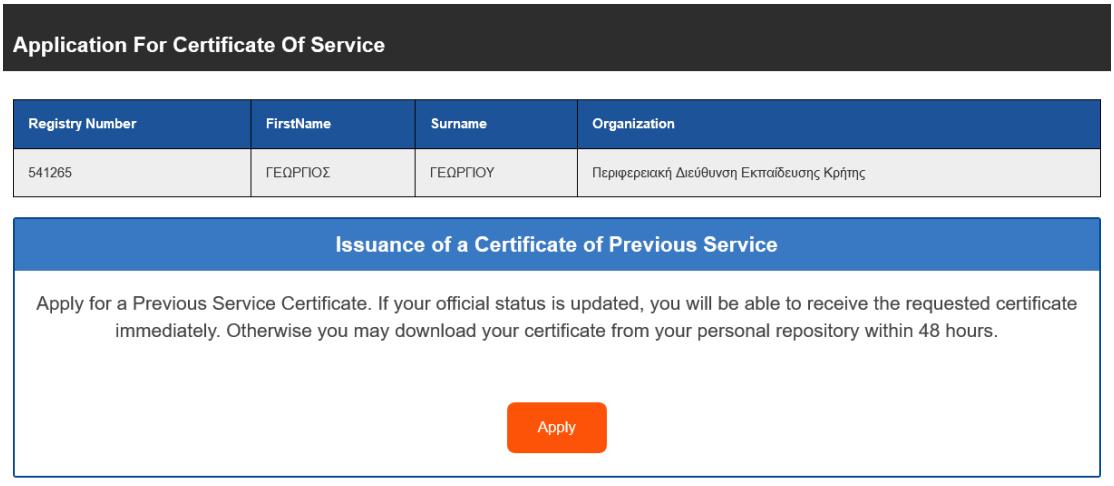


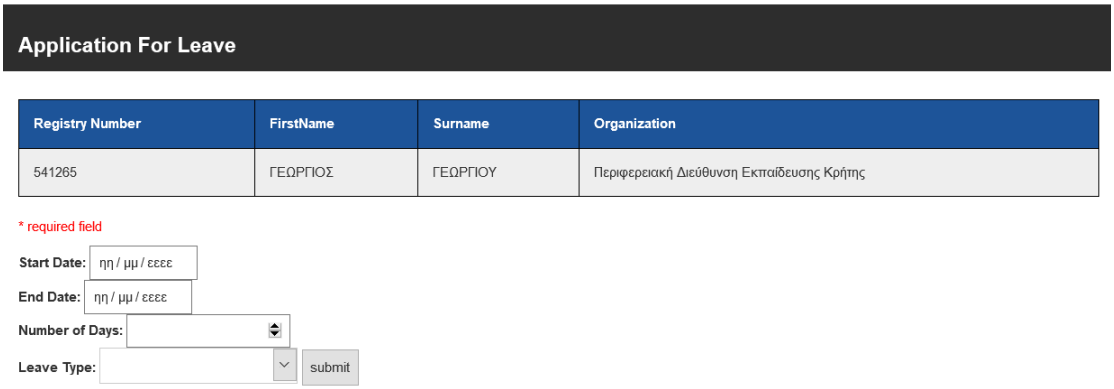
Figure 6.14 Application for Certificate of Service



- **Leave**

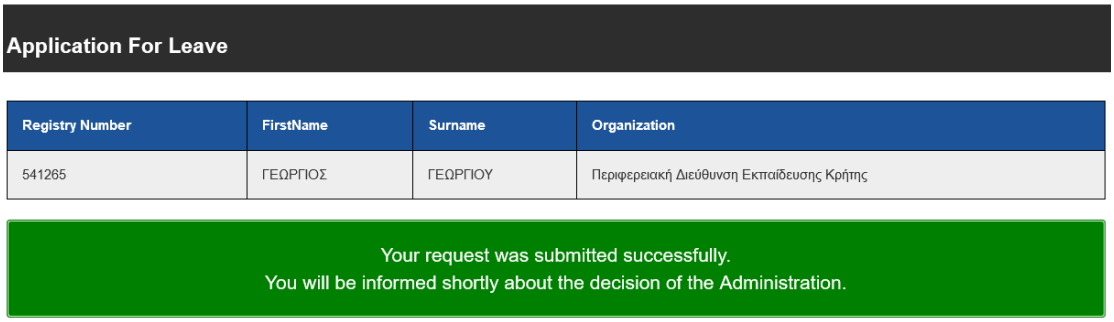
The procedure for granting a leave is different from the applications for certificates we examined above, as the intervention of a representative of the administration is required for its completion. The process is completed with the issuance of the relevant decision which expresses the will of the administration regarding the approval or rejection of the application. The employee can monitor the progress of his application by checking its status through the Personal Repository. When the relevant decision is issued, he is informed about the result of his application. Also, the representative of the administration sees through the Dashboard the pending applications and proceeds to the necessary actions for their processing. In order to submit the application, the applicant needs to fill in the necessary information. This information, as we see in figure 6.15, is the start and end date of the leave, its duration and the type of leave.

Figure 6.15 Application for Leave



To complete the application the user needs to click on the submit button. The request is then forwarded and registered in the SPARQL Endpoint, and the user receives a corresponding informative message as we see in figure 6.16.

Figure 6.16 Application's submission result



6.2.5 Users

- **Sign In**

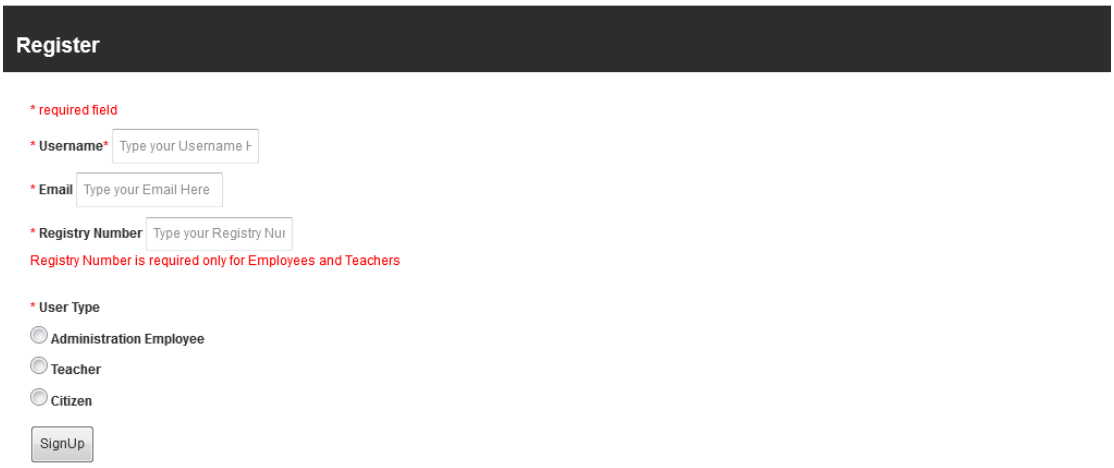
This option provides a simple interface where the user enters his credentials to be identified and access the online services.

Figure 6.17 Sihn In

- **Register**

On the website can register Employees, Teachers and Citizens. The information entered by the new user are his name, his type (Employee, Teacher or citizen) and the registration number (only for employees and teachers). When storing data, the registry number is formatted appropriately to have the same format as the Ontology data property hasRegistryNumber. In this way we connect the website user with the employee whose data are stored in the SPARQL Endpoint. The Web Portal Administrator approves the registration after checking.

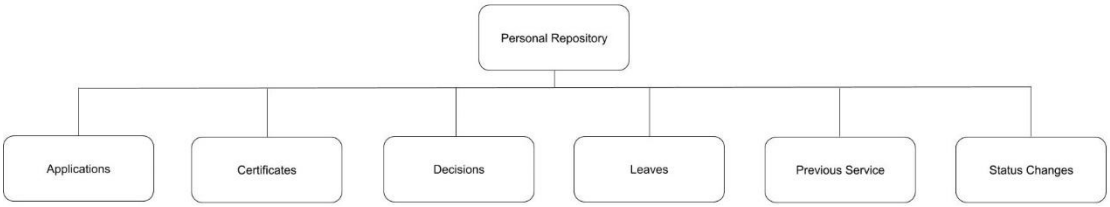
Figure 6.18 Register



- **Personal Repository**

This menu item only appears to logged in users who are clients of the services. Here users have access to their personal data, which is stored in the SPARQL Endpoint. In figure 6.19 we can see available option for the user. The options are divided into two categories, the options associated with the history of the user's transactions with the organization and options related to their personal registry.

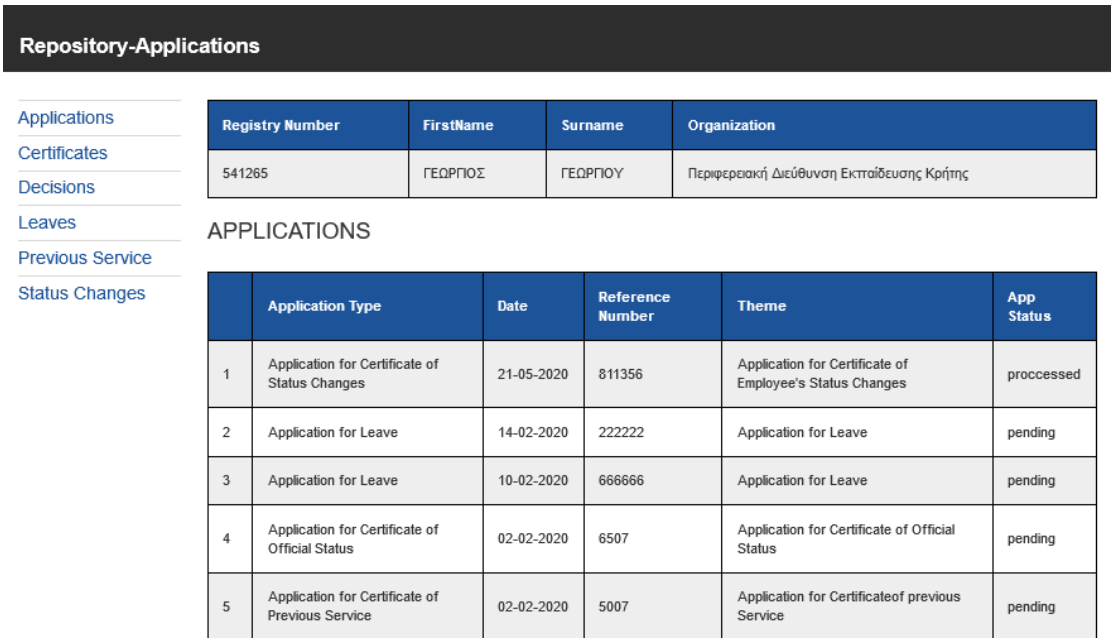
Figure 6.19 Menu item Personal Repository



Applications, Certificates and Decisions options belong to the first category. The user can view all the applications he has submitted to the organization through the portal and display the certificates and decisions issued for him. In figure 6.20 we can see a preview of Applications Section of the Personal Repository. For each application we can see the application type, the date, the reference number of the organization's, the theme and the status of the application. The applications are incoming documents for the Organization, so the preview of them is not necessary for the user or the administration. The information needed for the processing of the application is captured and a template of the application, is not necessary to take form, in order to be used outside of this system.

Unlike applications, certificates and decisions are outgoing documents and need to take form in order to be used outside the current system. figure 6.21 previews the Certificate's section, that contains the Certificates that the organization has issued for the user. For each certificate we can see the application type, the issuing organization, the theme, the reference number, the date, and the requesting application. For each certificate there is a Preview button, so the user can access and download it. The relevant information and action are also provided to the Decision's section.

Figure 6.20 Repository - Applications



Repository-Applications

- Applications
- Certificates
- Decisions
- Leaves
- Previous Service
- Status Changes

Registry Number	FirstName	Surname	Organization
541265	ΓΕΩΡΓΙΟΣ	ΓΕΩΡΓΙΟΥ	Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης

APPLICATIONS

	Application Type	Date	Reference Number	Theme	App Status
1	Application for Certificate of Status Changes	21-05-2020	811356	Application for Certificate of Employee's Status Changes	proccessed
2	Application for Leave	14-02-2020	222222	Application for Leave	pending
3	Application for Leave	10-02-2020	666666	Application for Leave	pending
4	Application for Certificate of Official Status	02-02-2020	6507	Application for Certificate of Official Status	pending
5	Application for Certificate of Previous Service	02-02-2020	5007	Application for Certificateof previous Service	pending

Figure 6.21 Repository - Certificates

Repository-Certificates

- Applications
- Certificates
- Decisions
- Leaves
- Previous Service
- Status Changes

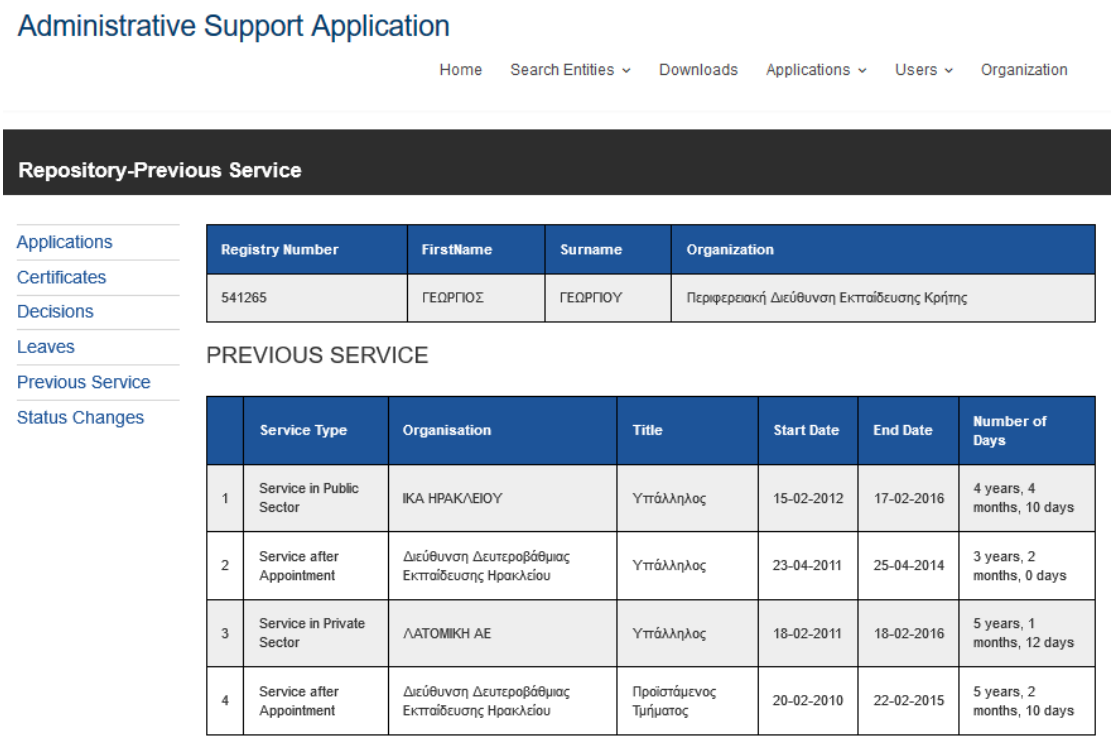
Registry Number	FirstName	Surname	Organization
541265	ΓΕΩΡΓΙΟΣ	ΓΕΩΡΓΙΟΥ	Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης

CERTIFICATES

	Certificate Type	Organization	Theme	Ref Number	Date	Proccessed Application	Actions
1	Certificate about Employee's Status Changes	Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης	Certificate about Employee's Status Changes	811356	21-05-2020	811356/21-05-2020	Preview
2	Certificate about Employee's Status Changes	Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης	Certificate about Employee's Status Changes	7006	31-08-2019	7006/25-08-2019	Preview
3	Certificate about Employee's Previous Service	Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης	Certificate about Employees's Previous Service	5006	30-08-2019	5006/25-08-2019	Preview
4	Certificate about Employees's Official Status	Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης	Certificate about Employees's Official Status	6506	28-08-2019	6506/25-08-2019	Preview

The second category includes the options Leaves, Previous Services and Status Changes. Here the user has a picture of his personal registry and can see his service status changes. He can also see the leaves he has received, and his previous services that have been recognized for salary and grade development. In figure 6.22 we preview the Previous Service Section. The displayed information are the service type, the organization's name, the position (title) he served, the beginning and ending date, and the duration of the service.

Figure 6.22 Repository - Previous Service



6.2.4 Organization

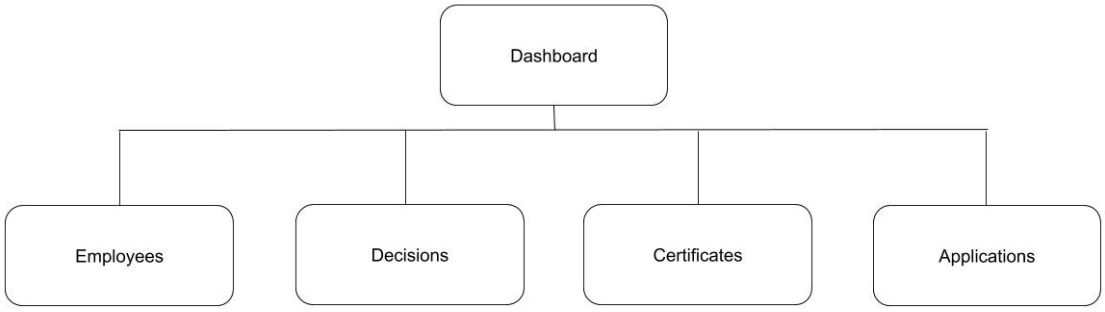
- **Sign In**

This option provides a simple interface where the user that acts on behalf of the Administration enters his credentials to be identified and access the online services.

- **Organization's Dashboard**

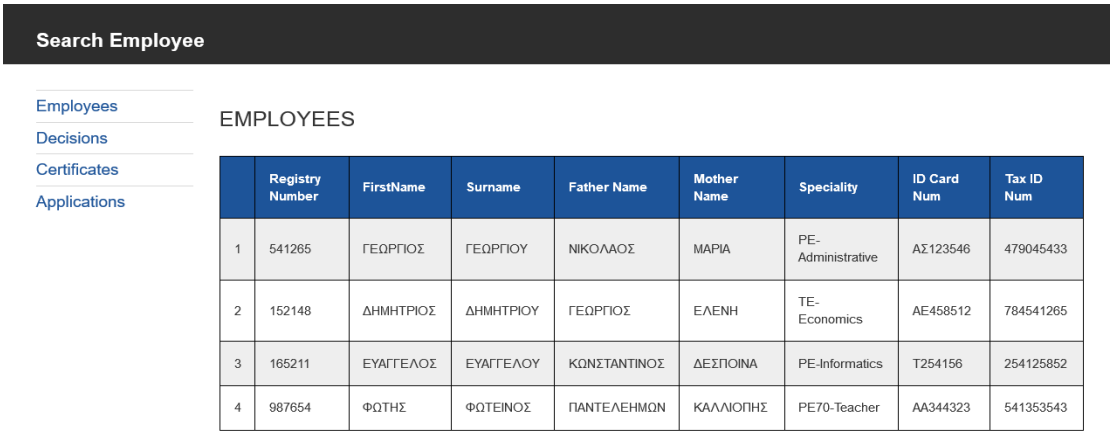
This menu item only appears to logged in users who act as representatives of the administration and are responsible for processing applications. At the home page of the Dashboard we can see the Organization's Information. On the left frame there is the Dashboard's Menu. At the following figure, we can see the available options in the Dashboard.

Figure 6.23 Menu item Dashboard



With the Employee option, the user has access to the employees' registry of the organization. As we see in figure 6.24, in this screen we can see basic information, for each employee.

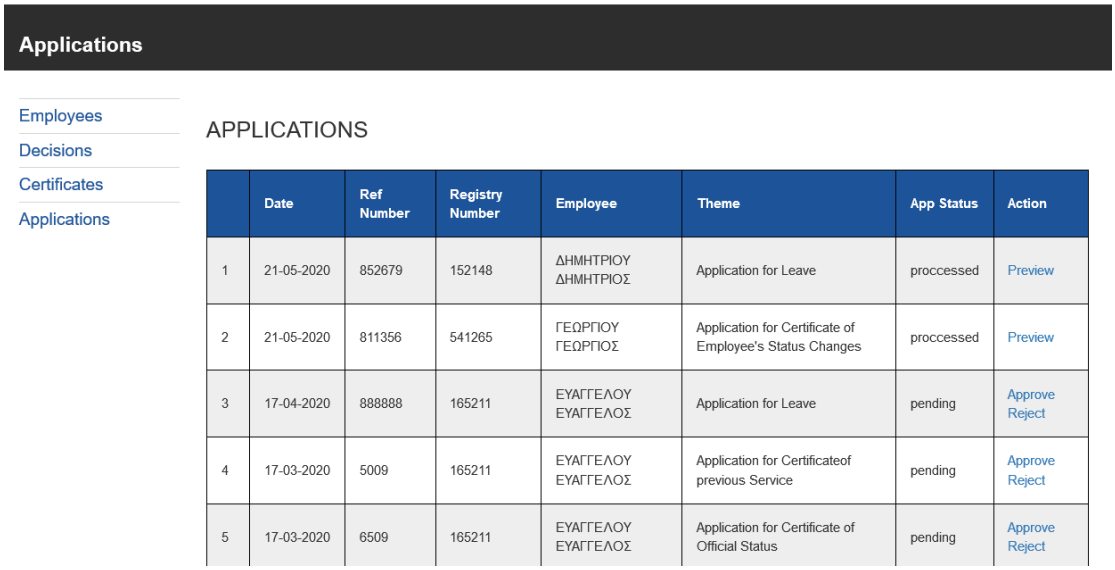
Figure 6.24 Dashboard - Employees



	Registry Number	FirstName	Surname	Father Name	Mother Name	Speciality	ID Card Num	Tax ID Num
1	541265	ΓΕΩΡΓΙΟΣ	ΓΕΩΡΓΙΟΥ	ΝΙΚΟΛΑΟΣ	ΜΑΡΙΑ	PE-Administrative	ΑΣ123546	479045433
2	152148	ΔΗΜΗΤΡΙΟΣ	ΔΗΜΗΤΡΙΟΥ	ΓΕΩΡΓΙΟΣ	ΕΛΕΝΗ	TE-Economics	ΑΕ458512	784541265
3	165211	ΕΥΑΓΓΕΛΟΣ	ΕΥΑΓΓΕΛΟΥ	ΚΩΝΣΤΑΝΤΙΝΟΣ	ΔΕΣΠΟΙΝΑ	PE-Informatics	Τ254156	254125852
4	987654	ΦΩΤΗΣ	ΦΩΤΕΙΝΟΣ	ΠΑΝΤΕΛΕΗΜΩΝ	ΚΑΛΛΙΟΠΗΣ	PE70-Teacher	ΑΑ344323	541353543

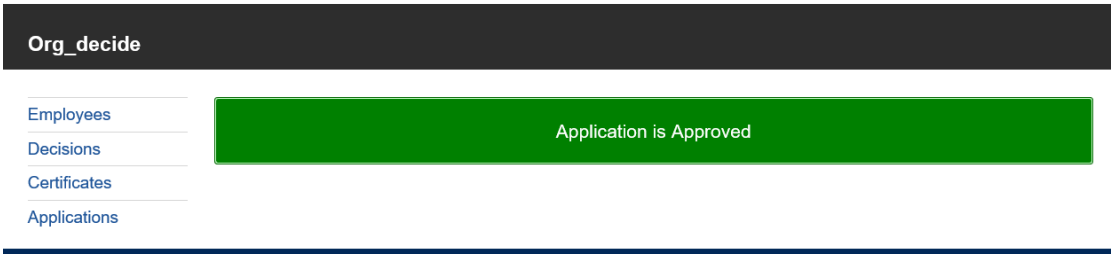
In figure 6.25 we can see the Section Applications of the dashboard. From this option, the user can access all the application that are submitted to his Organization. He can check the status of the applications and proceed to the necessary actions. If an application is processed the action Preview is displayed. From this option the user can preview the issued Certificates and Decisions. If an application is pending, the user can complete it by approving or rejecting the application, by clicking on the relevant option. With this action the Web-Portal updates the SPARQL Server and upon completion an informative message is displayed to the user (figure 6.26). After this, the application's status changes to processed and the decision can be displayed with the Preview option (figure 6.27)

Figure 6.25 Dashboard Applications



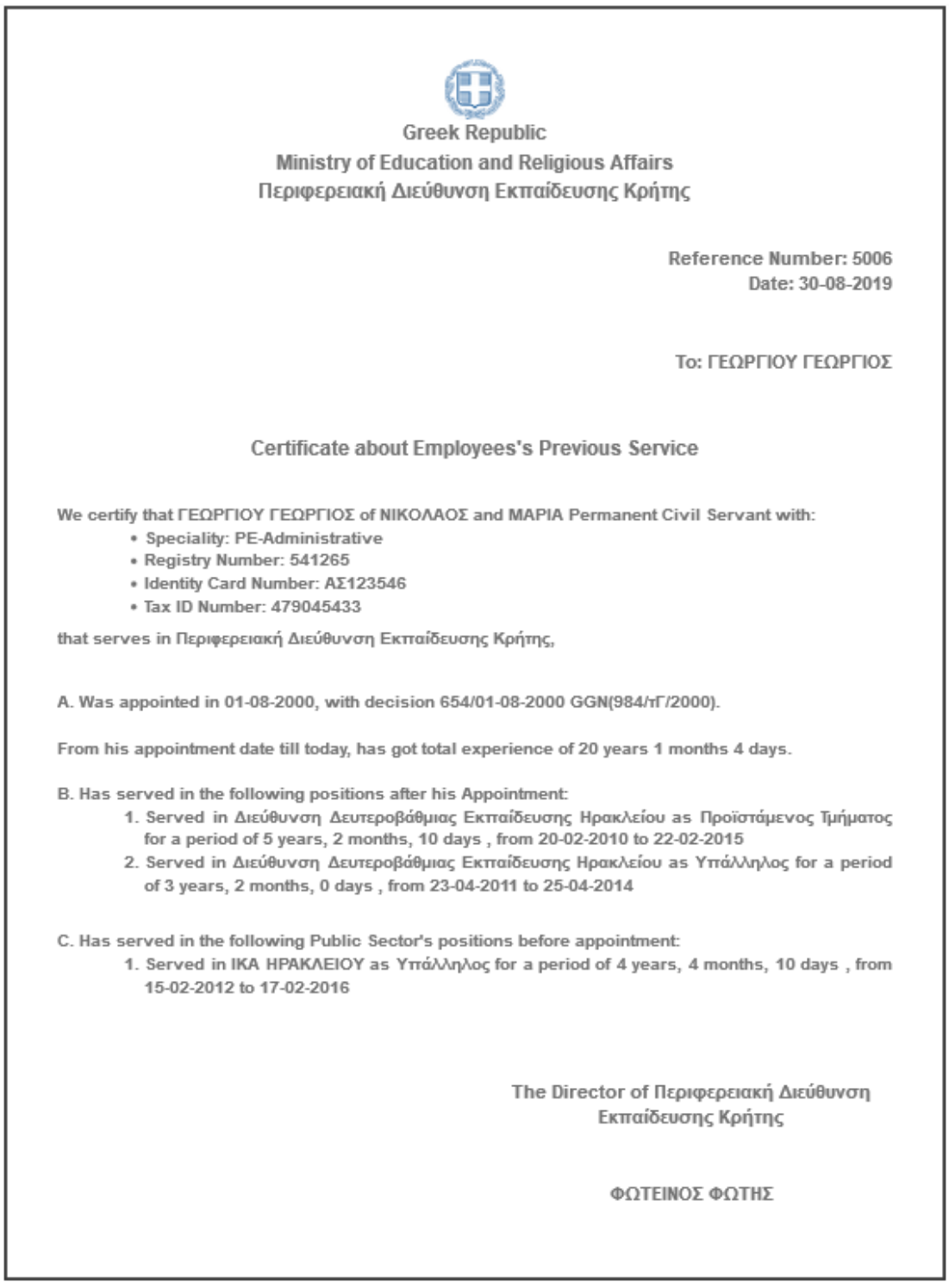
	Date	Ref Number	Registry Number	Employee	Theme	App Status	Action
1	21-05-2020	852679	152148	ΔΗΜΗΤΡΙΟΥ ΔΗΜΗΤΡΙΟΣ	Application for Leave	proccessed	Preview
2	21-05-2020	811356	541265	ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ	Application for Certificate of Employee's Status Changes	proccessed	Preview
3	17-04-2020	888888	165211	ΕΥΑΓΓΕΛΟΥ ΕΥΑΓΓΕΛΟΣ	Application for Leave	pending	Approve Reject
4	17-03-2020	5009	165211	ΕΥΑΓΓΕΛΟΥ ΕΥΑΓΓΕΛΟΣ	Application for Certificateof previous Service	pending	Approve Reject
5	17-03-2020	6509	165211	ΕΥΑΓΓΕΛΟΥ ΕΥΑΓΓΕΛΟΣ	Application for Certificate of Official Status	pending	Approve Reject

Figure 6.26 Application approval



The Decisions and Certificates options are similar to the respective Personal Repository options, but contain the relevant documents for all employees of the Organization.

Figure 6.27 Certificate about Employee's Previous Service



Greek Republic
Ministry of Education and Religious Affairs
Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης

Reference Number: 5006
Date: 30-08-2019

To: ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ

Certificate about Employees's Previous Service

We certify that ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ of ΝΙΚΟΛΑΟΣ and ΜΑΡΙΑ Permanent Civil Servant with:

- Speciality: PE-Administrative
- Registry Number: 541265
- Identity Card Number: ΑΣ123546
- Tax ID Number: 479045433

that serves in Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης,

A. Was appointed in 01-08-2000, with decision 654/01-08-2000 GGN(984/πΓ/2000).

From his appointment date till today, has got total experience of 20 years 1 months 4 days.

B. Has served in the following positions after his Appointment:

1. Served in Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης Ηρακλείου as Προϊστάμενος Τμήματος for a period of 5 years, 2 months, 10 days , from 20-02-2010 to 22-02-2015
2. Served in Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης Ηρακλείου as Υπάλληλος for a period of 3 years, 2 months, 0 days , from 23-04-2011 to 25-04-2014

C. Has served in the following Public Sector's positions before appointment:

1. Served in ΙΚΑ ΗΡΑΚΛΕΙΟΥ as Υπάλληλος for a period of 4 years, 4 months, 10 days , from 15-02-2012 to 17-02-2016

The Director of Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης

ΦΩΤΕΙΝΟΣ ΦΩΤΗΣ

Chapter 7 Implementation of Procedures

The functions of the Web-Portal described above can be grouped according to how they are implemented. In summary, we distinguish the following procedures, which we implemented and adapted to the above functions:

- Preview of an Organization's Incoming Documents (Applications)
- Preview of an Organization's Outgoing Documents (Decisions/Certificates)
- Application and automatic issuance of a document
- Application and manual issuance of a document
- Preview of an Outgoing Document(Decision/Certificate)
- Search by name
- Hierarchical Search

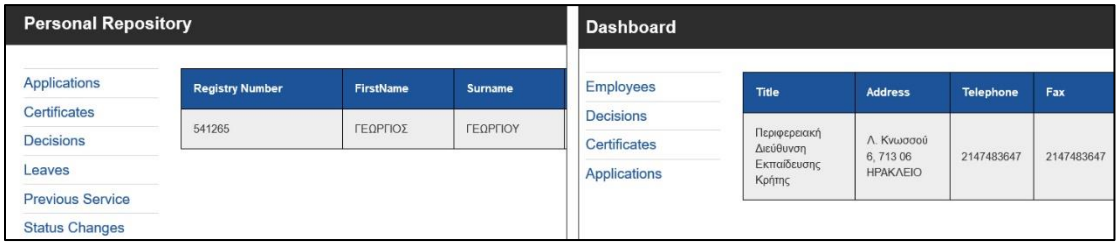
7.1 Preview of an Organization's Incoming Documents (Applications)

The process of viewing the Incoming Documents of an Organization, in our case the applications, are applied for the implementation of the Applications option. This option is available through the Personal Repository and the Administration's Dashboard. In both cases, the procedure is almost the same with minor modifications to the query that fetches the results.

- **Actions performed on the Web-Portal and the SPARQL Endpoint**

When the user selects Personal Repository or Administration's Dashboard from the main menu, a submenu appears in the left frame of the page (Figure 7.1)

Figure 7.1 Personal Repository - Administration's Dashboard



When selecting the above option from the Repository and the Dashboard, the SPARQL Endpoint is contacted and the following queries are submitted respectively. This execution shall

provide us with the details of all applications submitted by the employee and of all applications submitted to the Organization. For the compilation of queries, we use the Registry Number of the employee and Organization, which are stored in the variable nickname of WordPress

In Figure 7.2 we see the query executed by the Repository. The query that we send to Endpoint requests the instances that belong to the CivilServant_applications class (line 16) and that have been signed (Object Property: isSignedBy) by the employee (line 19). The data returned for these instances are the application's type label, id, Reference Number, theme, date, and status. If the application's type is "Application for leave", the Endpoint also returns the Leave's id.

Figure 7.2 SPARQL query for Applications from Repository

```
1  PREFIX sch: <http://www.pdekritis.gr/semantic#>
2  PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
3  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5
6  JSON {
7          'appTypeLabel': ?appTypeLabel,
8          'application': ?application,
9          'refNum': ?refNum,
10         'theme': ?theme,
11         'date': ?date,
12         'appStatus': ?appStatus,
13         'leave': ?leave
14 }
15 WHERE {
16         ?appType        rdfs:subClassOf     sch:CivilServant_applications  .
17         ?appType        rdfs:label          ?appTypeLabel .
18         ?application    rdf:type            ?appType ;
19                         sch:isSignedBy      sch:". $nickname  ." ;
20                         sch:hasRefNum       ?refNum ;
21                         sch:hasTheme        ?theme ;
22                         sch:hasDate         ?date ;
23                         sch:hasAppStatus    ?appStatus .
24         OPTIONAL {  ?application sch:isRequestingFor ?leave.    }
25 }
26 ORDER BY DESC(?date)
```

In Figure 7.3 we see the query executed in the Dashboard. The query that we send to the Endpoint requests the instances that belong to the CivilServant_applications class (line 21) and that have been submitted (Object Property: isSubmittedTo) to the Organization (line 24). The data returned by this query is more than the previous query, as in addition to the data of the application we also need the data of the employee who submitted it. Specifically for the application, these details are the application's type label, type, id, Reference Number, theme, date, and status. Also

for the employee, this information is the employee's id, first name, surname, and registry number. If the application's type is "Application for leave", the Endpoint also returns the Leave's id.

Figure 7.3 SPARQL query for Applications from Dashboard

```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

JSON {
        'appTypeLabel':     ?appTypeLabel,
        'appType':          ?appType,
        'app':              ?app,
        'refNum':           ?refNum,
        'theme':            ?theme,
        'date':             ?date,
        'appStatus':        ?appStatus,
        'employee':         ?employee,
        'firstName':        ?firstName,
        'surname':          ?surname,
        'regNum':           ?regNum,
        'leave':            ?leave
}
WHERE {
        ?appType    rdfs:subClassOf             sch:CivilServant_applications  .
        ?appType    rdfs:label                  ?appTypeLabel .
        ?app        rdf:type                    ?appType ;
                    sch:isSubmittedTo           sch:". $nickname  ." ;
                    sch:hasRefNum               ?refNum ;
                    sch:hasTheme                ?theme ;
                    sch:hasDate                 ?date ;
                    sch:hasAppStatus            ?appStatus ;
                    sch:isSignedBy              ?employee .
        ?employee   sch:hasFirstName            ?firstName ;
                    sch:hasSurname              ?surname ;
                    sch:hasRegistryNumber       ?regNum .
OPTIONAL {?app      sch:isRequestingFor         ?leave. }
}
ORDER BY DESC(?date)
```

The data retrieved by the above queries are displayed in a table by the Web-Portal and the user is given the available actions (Figure 7.4). The available actions are determined by the status of the application. If the status of the application is processed, then the user can view the relevant outgoing document. If its status is pending, the employee user can do nothing. On the contrary, the user representing the administration has the option to choose whether to approve or reject it. The user's decision is captured by pressing the corresponding option approve or reject. The actions performed by pressing the Approve / Reject and Preview buttons are described in paragraph 7.4.

Figure 7.4 Results from the query

	Date	Ref Number	Registry Number	Employee	Theme	App Status	Action
1	07-09-2020	558551	541265	ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ	Application for Certificate of Employee's Status Changes	pending	Approve Reject
2	06-09-2020	463418	152148	ΔΗΜΗΤΡΙΟΥ ΔΗΜΗΤΡΙΟΣ	Application for Certificate of previous Service	pending	Approve Reject
3	06-09-2020	975694	152148	ΔΗΜΗΤΡΙΟΥ ΔΗΜΗΤΡΙΟΣ	Application for Certificate of Official Status	proccessed	Preview
4	05-09-2020	639298	541265	ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ	Application for Leave	proccessed	Preview
5	05-09-2020	649187	541265	ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ	Application for Certificate of previous Service	proccessed	Preview

7.2 Preview of an Organization's Outgoing Documents (Decisions / Certificates)

The procedure followed for viewing the outgoing documents of the Organization (Decisions / Certificates) is similar to the procedure described in the previous paragraph, both for the Repository and the Dashboard. Selecting the options from the auxiliary menu executes SPARQL queries that fetch the necessary data. Then, the Web-Portal undertakes the presentation of the results to the user.

In figure 7.5 we present the excerpts of the queries that respectively set the criteria for the selection of Decisions and Certificates concerning an employee. For decisions, the query requests the instances that belong to the Decision class and are notified (Object Property: isForwardedTo) to the employee. For certificates, the query requests the instances that belong to the Certificate class and are notified (Object Property: isForwardedTo) to the employee.

Figure 7.5 SPARQL query for the Certificates of an employee

```
?decisionType    rdfs:subClassOf*        sch:Decision .
?decisionType    rdfs:label              ?decisionTypeLabel .
?decision        rdf:type                ?decisionType ;
                 sch:isForwardedTo       sch:". $nickname  ." ;


?certificateType     rdfs:subClassOf*    sch:Certificate .
?certificateType     rdfs:label          ?certificateTypeLabel .
?certificate         rdf:type            ?certificateType ;
                     sch:isForwardedTo   sch:". $nickname  ." ;
```

In Figure 7.6 we present the excerpts of the queries that respectively set the criteria for the selection of Decisions and Certificates of an Organization. For decisions, the query requests the instances that belong to the Decision class and have been issued (Object Property: isIssuedBy) by the Organization. For certificates, the query requests the instances that belong to the Certificate class and have been issued (Object Property: isIssuedBy) by the Organization.

Figure 7.6 SPARQL query for the Certificates issued by an organization

```
?decisionType    rdfs:subClassOf*        sch:Decision .
?decisionType    rdfs:label              ?decisionTypeLabel .
?decision        rdf:type                ?decisionType ;
                 sch:isIssuedBy          sch:". $nickname  ." ;


?certificateType     rdfs:subClassOf*        sch:Certificate .
?certificateType     rdfs:label              ?certificateTypeLabel .
?certificate         rdf:type                ?certificateType ;
                     sch:isIssuedBy          sch:". $nickname  ." ;
```

7.3 Application and automatic issuance of a document

Automatic document issuance is a process applied to the Web-Portal for the issuance of Certificate of Employee's Status Changes, Certificate of Official Status, and Certificate of Previous Service. For the correctness of the certificates, it is necessary to immediately enter the system all the information that changes the status of the employee. Also, this procedure is based on the fact that the necessary data for the issuance of the requested certificates have been registered in the file of the Organization after their validity has been checked by the Administration.

- **Steps of manual Procedure**

The traditional process of issuing the above certificates includes the following steps, which require human intervention:

Step 1: Compilation of a printed signed application by the employee, which includes his personal data for identification, ie his name, patronymic, Organic Position, position of service, registration number, specialty, grade, salary scale, home address, and contact numbers. The application also contains the subject of the application and the detailed description of the requested certificate.

Step 2: Receipt and registration of the application.

Step 3: Assignment of the application by the Head to the employee who will process it.

Step 4: Collection by the employee of the necessary data and preparation of the relevant certificate.

Step 5: Verification and validation of the application by the Head of the Organization.

Step 6: Send to the applicant the necessary certificate and archiving of the application and the document produced.

- **Redesign of the process**

The redesign of the process defines the following steps that require human intervention to process it through the Web-Portal:

Step 1: Login of the employee to the Web-Portal with his credentials.

Step 2: Select the desired application from the Main menu and submit the application simply by clicking the apply button, without providing any other information.

Step 3: Receipt of the requested Certificate from Personal Repository.

Comparing the two procedures we see that the redesigned process requires human intervention only on the part of the user, while the processing time of the application is minimized.

- **Actions performed on the Web-Portal and the SPARQL Endpoint**

Let's now look at the tasks performed by the Web-Portal and the SPARQL Endpoint for each of the user actions. We will use the Certificate of Employee's Status Changes as an example.

1st Select the function from the main menu

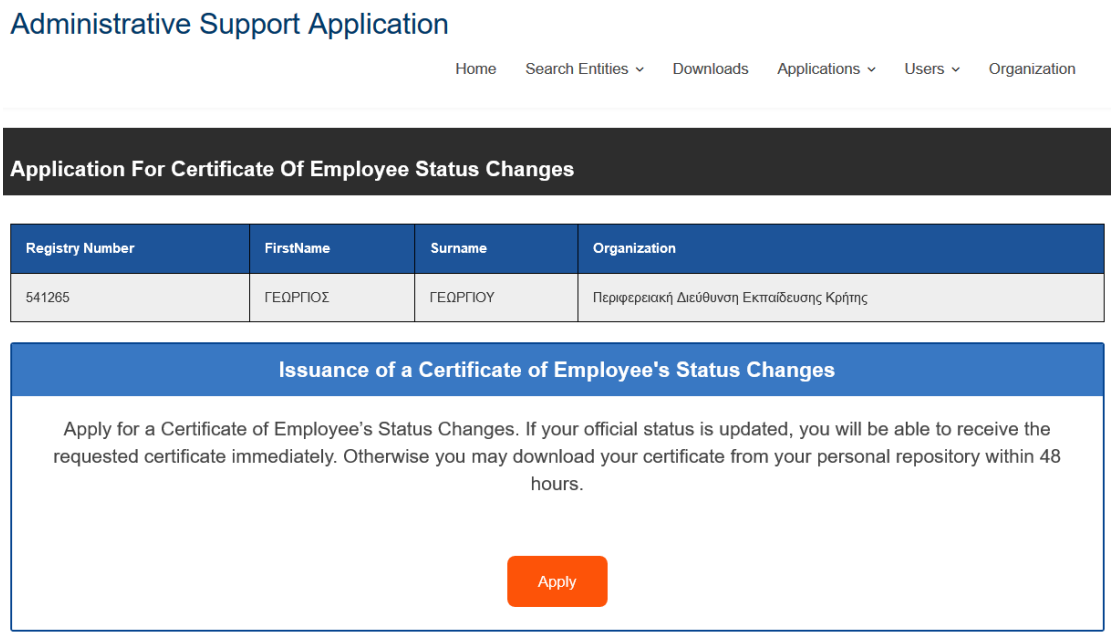
When selecting the function from the main menu, the SPARQL Endpoint is contacted and the following query is submitted (figure 7.7). With this query, we receive from Endpoint the basic data of the employee and the organization, which we display to the user to check their correctness. These details are the Registry Number, First Name, Surname, Organization, the Organization's title, and the head of the Organization. To locate the employee we use the "nickname" variable of WordPress which contains the Registry Number of the employee. This information is used on the one hand to be displayed to the employee before the application is submitted and to make a first check for their correctness. Also, this data is stored in local variables for later use for the compilation of the SPARQL Query that will be executed for the registration of the application and the issuance of the Certificate.

The Web-Portal then displays to the user the page we see in figure 7.8. On this screen, the user previews some basic information about him and is informed about the procedure to be followed to receive the requested certificate and is asked to confirm his intention.

Figure 7.7 SPARQL query for Employee's information

```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

JSON {
        'user':             ?user,
        'registryNumber':   ?registryNumber,
        'firstName':        ?firstName,
        'surname':          ?surname,
        'org':              ?org,
        'isEmployeeOf':     ?orgTitle,
        'orgHead':          ?orgHead
}
WHERE
    {
        ?x                      rdfs:subClassOf*        sch:CivilServant .
        ?user                   rdf:type                ?x ;
                                sch:hasRegistryNumber   ?registryNumber;
                                sch:hasFirstName        ?firstName;
                                sch:hasSurname          ?surname;
                                sch:isEmployeeOf        ?org.
        ?org                    sch:hasTitle            ?orgTitle ;
                                sch:hasHead             ?orgHead .
    filter (?user=sch:". $nickname  ." )
}
```

Figure 7.8 Application for Certificate of Employee Status



2nd Submission of the application

If the user proceeds with the submission by clicking the apply button, the application receives a Reference Number, the SPARQL query is compiled and the SPARQL Endpoint is contacted to execute it.

The Reference Number in our implementation is a random number generated by the Web-portal. As the Official Greek Code of Administrative Procedure provides for a single numbering for the reference numbers of an organization, separate numbering cannot be observed for documents circulating through the portal. Since the interconnection of the application with the organization's protocol is a big undertaking that does not touch the scope of the present research, we have adopted the solution of random Reference Numbers.

As no human intervention is required for the issuance of the certificate, the SPARQL query includes all the necessary entries for the registration of the application and the details of the certificate issued. In figure 7.9 we see the SPARQL query for creating the application and the Certificate of Employee Status Changes. For the syntax of the query, we use data we have stored in variables in the previous step. The variable "randNum" contains the random number we assigned as Reference Number. In the "appdate" variable, we have stored the current date, after

formatting it properly. The variables “nickname” and employee, contain the registry number of the employee. Finally, the variable Org, Director, and WebComposer contain the Organization, the Director that signs the certificate, and the composer of the Document respectively.

The query consists of two parts. In the first one (lines 7-17), we insert the application, and in the second part (lines19-34), we insert the Certificate. In particular, in lines 1-4 we define the prefixes we use. In lines 7-11 we insert the application’s details, which are the type, the reference number, the date, and the theme. Lines 13-14 specify the applicant. In lines 16-17, is inserted the Organization that receives the Application. In lines 19-22 we insert the certificate’s details, which are the type, the reference number, the date, and the theme. Lines 24-25, connect the Application with the Certificate. Lines 27-28 specify the receiver of the Certificate. In lines 29-30 with insert the issuing organization and in lines 31-32 the Director that validates the Certificate by signing it. Finally, lines 33-34 define the composer of the outgoing document. In this case, we insert the user webcomposer, as the process is automated, without human intervention.

Figure 7.9 SPARQL query for registering application - issuing of certificate

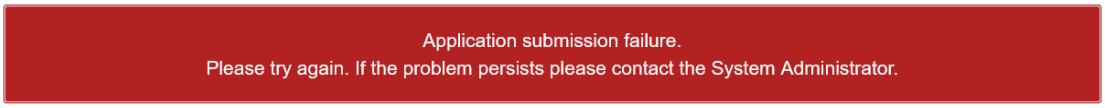
```
1  PREFIX sch: <http://www.pdekritis.gr/semantic#>
2  PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
3  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5
6  INSERT Data {
7      sch:app_statusChanges". $randNum ."     rdf:type            sch:StatusChanges_application .
8      sch:app_statusChanges". $randNum ."     sch:hasRefNum       ". $randNum ." .
9      sch:app_statusChanges". $randNum ."     sch:hasDate         ". $appDate  ." .
10     sch:app_statusChanges". $randNum ."     sch:hasTheme        \"Application for Certificate of Employee's Status Changes\" .
11     sch:app_statusChanges". $randNum ."     sch:hasAppStatus    \"proccessed\" .
12
13     sch:app_statusChanges". $randNum ."     sch:isSignedBy      sch:" . $nickname . " .
14     sch:" . $nickname . "                   sch:hasSigned       sch:app_statusChanges". $randNum . " .
15
16     sch:app_statusChanges". $randNum ."     sch:isSubmittedTo   <". $org ."> .
17     <". $org ."">                           sch:hasReceived     sch:app_statusChanges". $randNum ."  .
18
19     sch:certStatusChanges_". $randNum ."    rdf:type            sch:StatusChanges_certificate .
20     sch:certStatusChanges_". $randNum ."    sch:hasRefNum       ". $randNum ." .
21     sch:certStatusChanges_". $randNum ."    sch:hasDate         ". $appDate  ." .
22     sch:certStatusChanges_". $randNum ."    sch:hasTheme        \"Certificate about Employee's Status Changes \" .
23
24     sch:certStatusChanges_". $randNum ."    sch:hasProccessed   sch:app_statusChanges". $randNum ." .
25     sch:app_statusChanges" . $randNum ."    sch:isProccessedBy  sch:certStatusChanges_". $randNum ." .
26
27     sch:certStatusChanges_". $randNum ."    sch:isForwardedTo   <". $employee ."> .
28     <". $employee ."">                      sch:hasReceived     sch:certStatusChanges_". $randNum ." .
29     sch:certStatusChanges_". $randNum ."    sch:isIssuedBy      <". $org ."> .
30     <". $org ."">                           sch:hasIssued       sch:certStatusChanges_". $randNum ." .
31     <". $director ."">                      sch:hasSigned       sch:certStatusChanges_". $randNum ." .
32     sch:certStatusChanges_". $randNum ."    sch:isSignedBy      <". $director ."> .
33     sch:certStatusChanges_". $randNum ."    sch:isComposedBy    <". $webComposer ."> .
34     <". $webComposer ."">                   sch:hasComposed     sch:certStatusChanges_". $randNum ." .
35 }
```

3rd Result of the application

Depending on the result of the query's execution the web portal displays to the user a corresponding message of success or failure that we see in figure 7.10 and 7.11.

Figure 7.10 Successful application result

Figure 7.11 Application submission failure



The result of the above process is the completion of the issuance of the requested certificate to which the user has access through the Personal Repository and the administration through the Dashboard.

7.4 Application and manual issuance of a document

The process of manual issuance of a document is similar to the process of automatic issuance of a document. It differs as, before its issuance, the intervention of a representative of the Administration is required. For this reason, it is divided into two parts. The first part includes the application of the employee and the second the decision of the Management. This procedure applies to the employee's application for leave.

- **Steps of manual Procedure**

The traditional process of issuing the above certificates includes the following steps, which require human intervention:

Step 1: Compilation of a printed signed application by the employee, which includes his personal data for identification, ie his name, patronymic, Organic Position, position of service, registration number, specialty, grade, salary scale, home address, and contact numbers. The application also contains the subject of the application and the type of leave requested, its duration, and the start and end dates of the leave.

Step 2: Receipt and registration of the application.

Step 3: Assignment of the application by the Head to the employee who will process it.

Step 4: Approval or rejection of the application by the Head

Step 5: Collection by the employee of the necessary data and preparation of the relevant decision.

Step 6: Verification and validation of the application by the Head of the Organization.

Step 7: Send to the applicant the necessary certificate and archiving of the application and the document produced.

- **Redesign of the process**

The redesign of the process defines the following steps that require human intervention to process it through the Web-Portal:

Step 1: Login of the employee to the Web-Portal with his credentials.

Step 2: Select the desired application from the menu of user functions and fill in the necessary elements of the application. Submit the application by pressing the apply button.

Step 3: Login of the administration representative to the Web-Portal with his credentials.

Step 4: View of pending applications in the Dashboard.

Step 5: Informing the Head, making a decision, and registering it, through the Dashboard

Step 6: Receipt of the requested Certificate from Personal Repository.

Comparing the two procedures we see that in the redesigned process the human intervention is reduced both on the part of the user and the part of the administration, and at the same time, the processing time of the application is minimized.

- **Actions performed on the Web-Portal and the SPARQL Endpoint**

Let's now look at the tasks performed by the Web-Portal and the SPARQL Endpoint for each of the user actions.

Part 1 - Apply

1st Select the function from the main menu

When selecting the function from the main menu, the SPARQL Endpoint is contacted and the query described in paragraph 7.3 is submitted (figure 7.7). With this query, we get from Endpoint the basic data of the employee and the organization, which are stored in local variables to be used later for the compilation of the SPARQL query that will be executed for the registration of the application.

Another query is also submitted which presents the available leave types, which we see in figure 7.12.

Figure 7.12 SPARQL query of leave types

```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

JSON {
        'leaveType':        ?leaveType,
        'leaveTypeLabel':   ?leaveTypeLabel

}
WHERE
{
        ?leaveType          rdfs:subClassOf     sch:Leave ;
                            rdfs:label          ?leaveTypeLabel .
}
```

2nd Completion of the application and submission

After completing the application by filling the necessary fields in the Leave form, the user clicks the submit button. After this the application receives a Reference Number, the SPARQL query is compiled and the SPARQL Endpoint is contacted to execute it. In figure 7.13 we see the SPARQL query for creating the application and the individual of the class Leave.

For the syntax of the query, we use data we have stored in variables in the previous step. The variable “randNum” contains the random number we assigned as Reference Number. The variable “nickname”, contains the registry number of the employee and the variable Org contains the Organization. We also use the parameters that the user entered with the completion of the Leave form. These parameters are stored in the POST variables $_POST["leaveType"] that

contains the type of the leave and $_POST["numOfDates"] that contains the number of the leaves' days. The start and end day, that are stored in the relevant POST variables, are formatted properly before used in the query and we store them in the local variables $startDateVar and $endDateVar respectively.

The query creates an application and a leave. We create a separate individual for leave because it is data that must be kept in the personal records of the employee. In lines 7-11 we insert the leave's details, which are the type, start date, end date, number of dates, and the status which is set to pending. In lines 13-17 we insert the application's details, which are the type, the reference number, the date, the theme, and the status which is also set to pending. Lines 19-20 specify the applicant. Lines 22-23, connect the Application with the Leave and 25-26 connects the employee with the Leave. In lines 28-29 is inserted the Organization that receives the Application.

Figure 7.13 SPARQL query for registering application and leave

```
1  PREFIX sch: <http://www.pdekritis.gr/semantic#>
2  PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
3  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5
6  INSERT Data {
7                  sch:leave_". $randNum ."        rdf:type                <". $_POST["leaveType"]  .">  .
8                  sch:leave_". $randNum ."        sch:hasLeaveStartDate   ". $startDateVar  ." .
9                  sch:leave_". $randNum ."        sch:hasLeaveEndDate     ". $endDateVar  ." .
10                 sch:leave_". $randNum ."        sch:hasLeaveNumOfDays   ". $_POST["numOfDates"]  ." .
11                 sch:leave_". $randNum ."        sch:hasLeaveStatus      \"pending\" .
12
13                 sch:app_Leave". $randNum ."     rdf:type                sch:Leave_application .
14                 sch:app_Leave". $randNum ."     sch:hasRefNum           ". $randNum ." .
15                 sch:app_Leave". $randNum ."     sch:hasDate             ". $appDate  ." .
16                 sch:app_Leave". $randNum ."     sch:hasTheme            \"Application for Leave\" .
17                 sch:app_Leave". $randNum ."     sch:hasAppStatus        \"pending\" .
18
19                 sch:app_Leave". $randNum ."     sch:isSignedBy          sch:" . $nickname . " .
20                 sch:" . $nickname . "           sch:hasSigned           sch:app_Leave". $randNum . " .
21
22                 sch:app_Leave". $randNum ."     sch:isRequestingFor     sch:leave_". $randNum ." .
23                 sch:leave_". $randNum ."        sch:isRequestedWith     sch:app_Leave". $randNum ." .
24
25                 sch:" . $nickname . "           sch:hasRequestedLeave   sch:leave_". $randNum ." .
26                 sch:leave_". $randNum ."        sch:isRequestedFrom     sch:" . $nickname . " .
27
28                 sch:app_Leave". $randNum ."     sch:isSubmittedTo       <". $org .">  .
29                 <". $org .">                    sch:hasReceived         sch:app_Leave". $randNum ."  .
30 }
```

Part 2 - Processing the application

1ov Selecting option "Applications" from the Dashboard Menu

The application is processed in the Administration's Dashboard. In order, the representative of the Administration, to have access to it, he must log in with the corresponding credentials and select the option "Applications".

The query in Figure 7.2 is then executed, which we described in section 7.1. The web-portal then displays the results and the user is allowed to approve or reject the pending applications.

2nd Approval or rejection of an application

By clicking the approve or reject option the control is transferred to the PHP function which manages the creation of the decision, as we see in figure 7.14. In this function, we send the decision for approval or rejection which is recorded in a GET variable. We also send in separate variables, the application's reference number, and the registry number of the employee.

Figure 7.14 Approve - Reject leave

```
<td><?php if (strcmp($resultApps['appStatus'],'proccessed') == 0) {?>
            <a href="<?php echo $home1.$redirectURL.$resultApps['refNum'];?>">Preview</a>
        <?php   } else {
            $nickname = explode("#",$resultApps['employee']);
        ?>

            <a href="<?php echo $home1."/org_decide?ref=".$resultApps['refNum']."&dec=approve&reg=".$nickname[1];?>">Approve</a>
            <a href="<?php echo $home1."/org_decide?ref=".$resultApps['refNum']."&dec=reject&reg=".$nickname[1];?>">Reject</a>

        <?php } ?>
</td>
```

At the start of the function execution, as in the previous procedures, the query described in paragraph 7.3 is executed (figure 7.7). With this query, we get from Endpoint the basic data of the employee and the organization, which are stored in local variables to be used later for the compilation of the SPARQL query that will be executed for the registration of the application.

Then, depending on whether the application has been approved or not, we give values to some local variables that we will need in the SPARQL query that we will execute to register the decision. These are the "theme" variable that contains the subject of the "Approval of application for leave" or "Rejection of application for leave" decision and the "leaveStatus" variable that contains the "approved" or "rejected" leave status.

After that, the query we see in figure 7.15 is executed which creates the relevant decision. The query begins with the deletion of the leave's status and the application's status (lines 6-9), which is pending, in order to register their new status with the creation of the decision (lines 11-32). Lines 13-14 update the status of the leave (approved/rejected) and the application for leave(processed). Lines 16-19 enter the elements of the Decision (type, reference number, date, and theme). With lines 21-22 we connect the decision with the application. Lines 24-31 link the

leave to the applicant, to the issuing Organization, to the Director, and to the document's composer.

The result of the above process is the completion of the processing of the requested leave. The produced document is accessible, to the user through the Personal Repository, and to the administration through the Dashboard.

Figure 7.15 SPARQL query for issuing a decision

```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

DELETE Data {
                sch:app_Leave". $_GET['ref'] ."          sch:hasAppStatus    \"pending\" .
                sch:leave_". $_GET['ref'] ."             sch:hasLeaveStatus  \"pending\" .
};


INSERT Data {
                sch:app_Leave". $_GET['ref'] ."          sch:hasAppStatus    \"proccessed\" .
                sch:leave_". $_GET['ref'] ."             sch:hasLeaveStatus  \"" . $leaveStatus . "\" .

                sch:decisionLeave_". $_GET['ref'] ."     rdf:type            sch:Leave_decision .
                sch:decisionLeave_". $_GET['ref'] ."     sch:hasRefNum       ". $_GET['ref'] ." .
                sch:decisionLeave_". $_GET['ref'] ."     sch:hasDate         ". $decDate ." .
                sch:decisionLeave_". $_GET['ref'] ."     sch:hasTheme        ". $theme ." .

                sch:decisionLeave_". $_GET['ref'] ."     sch:hasProccessed   sch:app_Leave". $_GET['ref'] ."  .
                sch:app_Leave". $_GET['ref'] ."          sch:isProccessedBy  sch:decisionLeave_". $_GET['ref'] ." .

                sch:decisionLeave_". $_GET['ref'] ."     sch:isForwardedTo   <". $employee ."> .
                <". $employee .">                        sch:hasReceived     sch:decisionLeave_". $_GET['ref'] ." .
                sch:decisionLeave_". $_GET['ref'] ."     sch:isIssuedBy      <". $org ."> .
                <". $org .">                             sch:hasIssued       sch:decisionLeave_". $_GET['ref'] ." .
                <". $director .">                        sch:hasSigned       sch:decisionLeave_". $_GET['ref'] ." .
                sch:decisionLeave_". $_GET['ref'] ."     sch:isSignedBy      <". $director ."> .
                sch:decisionLeave_". $_GET['ref'] ."     sch:isComposedBy    <". $webComposer ."> .
                <". $webComposer .">                     sch:hasComposed     sch:decisionLeave_". $_GET['ref'] ." .
}
```

7.5 Preview of an Outgoing Document (Decision/Certificate)

The outgoing documents that are issued, are not static documents that are stored in a server storage. Every time a user requests to preview them, the Web-Portal creates them dynamically by collecting data with queries submitted to the SPARQL Endpoint. The user, of course, can download them to his computer.

The preview option for viewing the documents is made through the Decisions and Certificates options of the Personal Repository and the Administration's Dashboard, as described above. The only element that needs to be passed to the PHP function that manages the display is the Reference Number of the document, as shown in figure 7.16.

Figure 7.16 PHP excerpt for previewing a decision

```
<td><a href="<?php echo $home1."/decleavepreview?dec=".$resultDecisions['refNum'];?>">Preview</a> </td>
```

7.5.1 Document structure

Before proceeding with the actions taken in the Web-Portal and in Endpoint, let us look at the structure of the documents that we create. In figure 7.17 we see the structure of a certificate and in figure 7.18 the structure of a decision. The format of the documents we present is a simplified version of the standards set by the Official Greek Code of Administrative Procedure, to facilitate implementation.

Both types of documents contain the title of the organization and the hierarchy to which it belongs. They also contain reference number, release date, notifications, and theme. Finally, at the end of the document is the name and signature of the head of the Organization that validates the document.

Let's look at the elements that differentiate the two types of documents. The certificates contain basic information about the employee, which, however, is important for his identification and the brief description of his official status. These data are the name, patronymic, specialty, registration number, ID number, and Tax ID number. The certificates also contain a section listing the information certified by the organization. The Decisions, in addition to the common points with the Certificates, contain the section of the relevant legislation and the operative part. The relevant legislation contains the legislation and the events that were taken into account for the drafting of the decision. The operative part contains a description of what the administration decided about the request of the applicant.

7.5.2 Actions performed on the Web-Portal and the SPARQL Endpoint

- **Preview of a Certificate**

Once the control is transferred to the function that manages the display of the certificates, a series of SPARQL queries are executed which collect the necessary data for constructing the

document. If the queries are executed successfully then the Web-Portal compiles the document and displays it to the user.

Let's first look at which queries are executed by the three types of certificates that we manage and then we will analyze separately the specifics of each certificate. These queries will provide us with information about the employee, the organization, and the document.

First, based on the Registry Number of the employee stored in the variable “nickname”, the following query is executed which brings the full details of the employee. These data are the label of the type of employee (teacher or employee), and the civil servant’s registryNumber, firstName, surname, fatherName, motherName, educationLevel, specialty, idenityCardNum, taxIDNum (lines 27-38). Also for the organization the employee serves are the organization’s id, title, Address (lines 39-41) and it’s director’s Surname and First name (lines 42-43).

Figure 7.17 Structure of a certificate

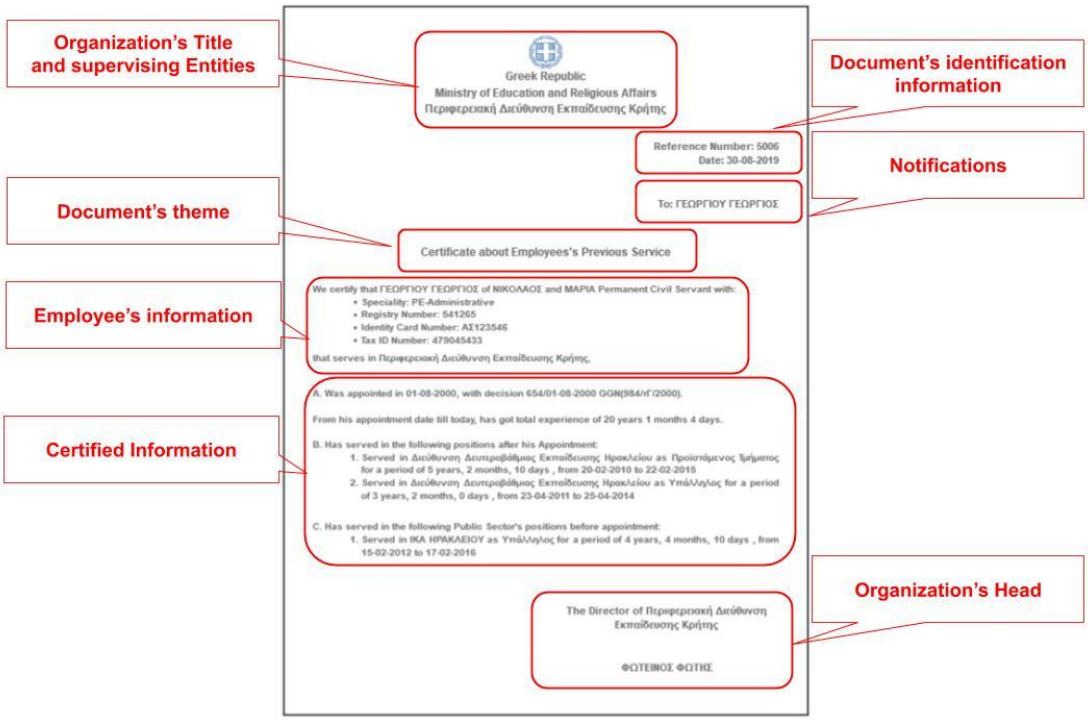


Figure 7.18 Structure of a decision

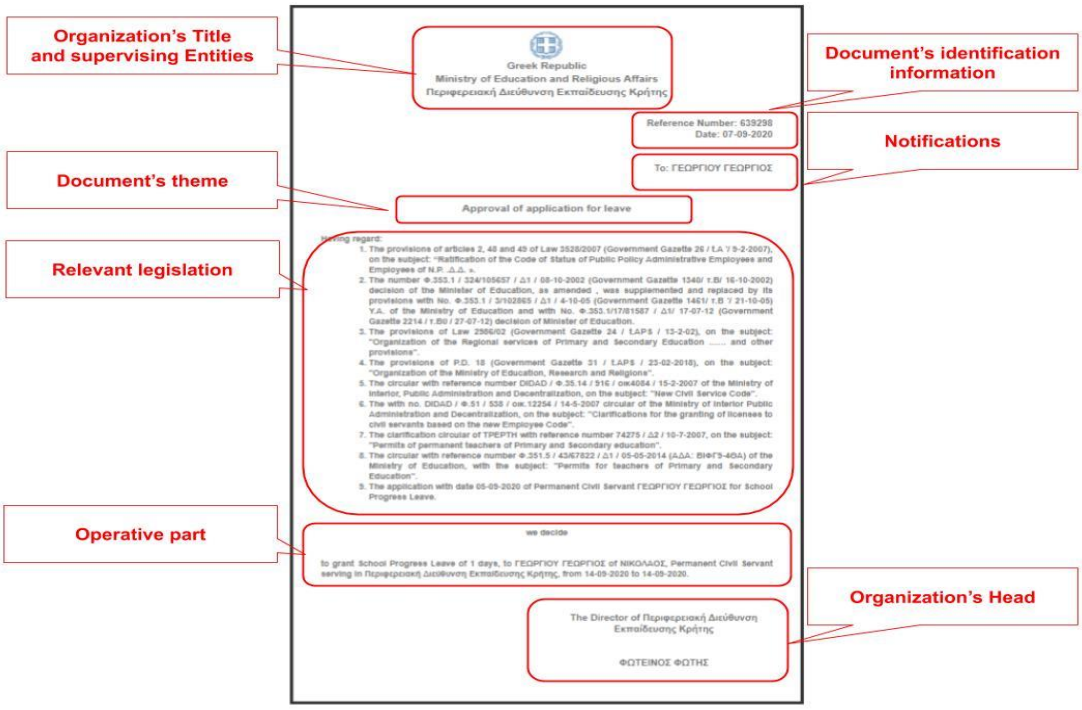


Figure 7.19 SPARQL query for employee's information

```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

JSON {
        'y':                            ?y,
        'civilServantTypeLabel':        ?civilServantTypeLabel,
        'registryNumber':               ?registryNumber,
        'firstName':                    ?firstName,
        'surname':                      ?surname,
        'fatherName':                   ?fatherName,
        'motherName':                   ?motherName,
        'educationLevel':               ?educationLevel,
        'speciality':                   ?speciality,
        'idenityCardNum':               ?idenityCardNum,
        'taxIDNum':                     ?taxIDNum,
        'org':                          ?org,
        'isEmployeeOf':                 ?orgTitle,
        'orgAddress':                   ?orgAddress,
        'dirSurname':                   ?dirSurname,
        'dirFirstName':                 ?dirFirstName
}
WHERE
{
        ?civilServantClass  rdfs:subClassOf*        sch:CivilServant .
        ?civilServantClass  rdfs:label              ?civilServantTypeLabel .
        ?civilServant       rdf:type                ?civilServantClass ;
                            sch:hasRegistryNumber   ?registryNumber;
                            sch:hasFirstName        ?firstName;
                            sch:hasSurname          ?surname;
                            sch:hasFatherName       ?fatherName;
                            sch:hasMotherName       ?motherName;
                            sch:hasEducationLevel   ?educationLevel;
                            sch:hasSpeciality       ?speciality;
                            sch:hasIdenityCardNum   ?idenityCardNum;
                            sch:hasTaxIDNum         ?taxIDNum;
                            sch:isEmployeeOf        ?org.
        ?org                sch:hasTitle            ?orgTitle ;
                            sch:hasAddress          ?orgAddress;
                            sch:hasHead             ?director .
        ?director           sch:hasSurname          ?dirSurname ;
                            sch:hasFirstName        ?dirFirstName .
        filter (?civilServant=sch:". $nickname  ." )
}
```

The following query (figure 7.20) searches based on the Registry Number of the employee, the Organization to which it belongs, and then the supervisory bodies of the Organization. The items searched are the organization's id, supervisor organization, and its title. These details will be placed on the front of the Certificate.

Figure 7.20 SPARQL query for organization's hierarchy

```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

JSON {
        'org':              ?org,
        'superOrg':         ?superOrg,
        'superOrgTitle':    ?superOrgTitle
}
WHERE
{
        sch:". $nickname  ."    sch:isEmployeeOf        ?org.
        ?org                    sch:isSupervisedBy*     ?superOrg .
        ?superOrg               sch:hasTitle            ?superOrgTitle .
} ORDER BY  ASC(?superOrg)
```

In figure 7.21 we see the query which presents the data of the document. The information needed to execute it, is the Reference Number and the document type. A common tactic of the public sector is to use the Reference number of the incoming document (application) to issue the outgoing document (certificate or decision). Therefore, the type of document in the query we compile is outgoing. The items returned are the document's id, date, and theme.

Figure 7.21 SPARQL query for documents information

```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

JSON {
        'doc':      ?doc,
        'date':     ?date,
        'theme':    ?theme
}
WHERE
{
        ?doc        sch:hasRefNum       ". $doc  ." .
        ?doc        rdf:type*           ?y .
        ?y          rdfs:subClassOf*    sch:". $docType ." .
        ?doc        sch:hasDate         ?date ;
                    sch:hasTheme        ?theme .
}
```

Let's now look at the extra queries that run depending on the type of each certificate and how the data is combined to produce it.

- **Certificate of Employee's Status Changes**

The Certificate of Employee's Status Changes is the certificate with the most information about the employee as it lists in detail all the information that concerns him. The data collected are:

A. Changes in status

Changes in service status are distinguished in appointment, discharge, Disciplinary penalty, Grade, Moral Rewards, Pension, and Salary Scale. To collect this data, the query of figure 7.22 is executed. The search is based on the registry number of the employee and we look for which instances of the Changes class affect him (Object Property: isEffectedBy). The data provided for all types of changes are its type and label (lines 22, 32), the change's date (line 23). We also request information for the Decision that validated the change. These data are the decision's authority organization, date, reference number (lines 24-27). For each type of change, different items are returned (lines 27-31). Specifically for the appointment, dismissal, and retirement, the Government Gazette of the decision is returned. Also for the changes Grade, Salary Scale, Reward, and Penalty for each of them the corresponding data is returned.

Figure 7.22 SPARQL query for changes

```
1  PREFIX sch: <http://www.pdekritis.gr/semantic#>
2  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5
6  JSON {
7          'change':               ?change,
8          'changeTypeLabel':      ?changeTypeLabel,
9          'changeDate':           ?changeDate,
10         'decisionAuthority':    ?decisionAuthority,
11         'decisionDate':         ?decisionDate,
12         'decisionRefNum':       ?decisionRefNum,
13         'govGazzeteNum':        ?govGazzeteNum,
14         'grade':                ?grade,
15         'penaltyType':          ?penaltyType,
16         'rewardType':           ?rewardType,
17         'salaryScale':          ?salaryScale
18 }
19 WHERE
20     {
21             sch:". $nickname ." sch:isEffectedBy            ?change.
22             ?change             rdf:type                    ?changeType;
23                                 sch:hasChangeDate           ?changeDate ;
24                                 sch:hasDecisionAuthority    ?decisionAuthority ;
25                                 sch:hasDecisionDate         ?decisionDate ;
26                                 sch:hasDecisionRefNum       ?decisionRefNum .
27 OPTIONAL {  ?change             sch:hasGovGazzeteNum        ?govGazzeteNum . }
28 OPTIONAL {  ?change             sch:hasGrade                ?grade . }
29 OPTIONAL {  ?change             sch:hasPenaltyType          ?penaltyType . }
30 OPTIONAL {  ?change             sch:hasRewardType           ?rewardType . }
31 OPTIONAL {  ?change             sch:hasSalaryScale          ?salaryScale . }
32             ?changeType         rdfs:label                  ?changeTypeLabel .
33 } ORDER BY  ASC(?changeTypeLabel)
```

B. Unpaid Leaves

With the query of figure 7.25, we receive the data for the Unpaid Leaves that the employee has received. This element is important because it is deducted from time taken into account for his official development. The query looks for the instances that belong to the Unpaid_leave class and have been requested (Object Property: isRequestedFrom) by the employee. The data that are returned are leave's id, start date, end date, duration. The query also fetches the Organization that issued the leave, and the decision's reference number and date.

Figure 7.23 SPARQL query for unpaid leaves

```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

JSON {  'leave':            ?leave,
        'leaveStartDate':   ?leaveStartDate,
        'leaveEndDate':     ?leaveEndDate,
        'leaveNumOfDays':   ?leaveNumOfDays,
        'refNum':           ?refNum,
        'date':             ?date,
        'orgTitle':         ?orgTitle
}
WHERE
    {
        ?leave      rdf:type                sch:Unpaid_leave ;
                    sch:isRequestedFrom     sch:".$nickname." ;
                    sch:hasLeaveStartDate   ?leaveStartDate ;
                    sch:hasLeaveEndDate     ?leaveEndDate ;
                    sch:hasLeaveNumOfDays   ?leaveNumOfDays  ;
                    sch:hasLeaveStatus      \"approved\" ;
                    sch:isRequestedWith     ?app .
        ?app        sch:isProccessedBy      ?decision .
        ?decision   sch:hasRefNum           ?refNum ;
                    sch:hasDate             ?date ;
                    sch:isIssuedBy          ?org .
        ?org        sch:hasTitle            ?orgTitle .
}
```

C. Total Experience

Another important element that is mentioned in the certificate is the total time of service after the permanent appointment. To calculate it, we do not do any additional query but we use the date of appointment and the date of issuance of the certificate submitted with the queries we made above. From the date of the issue of the certificate, we subtract the date of appointment and the total service time is obtained.

D. Experience before Appointment

In figure 7.24 we see the query with which we request the services offered before the permanent appointment in the public sector. Essentially the query searches the instances of the PublicSector_service class offered (Object Property: wasOfferedBy) by the employee. The data returned are the service's start date, end date, title, total time, and the organization's title.

Figure 7.24 SPARQL query services in Public Sector

```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

JSON {
        'service':              ?service,
        'serviceStartDate':     ?serviceStartDate,
        'serviceEndDate':       ?serviceEndDate,
        'serviceTitle':         ?serviceTitle,
        'serviceTotalTime':     ?serviceTotalTime,
        'orgTitle':             ?orgTitle
}
WHERE
    {
        ?service    rdf:type                sch:PublicSector_service ;
                    sch:wasOfferedBy        sch:". $nickname  ." ;
                    sch:hasServiceStartDate ?serviceStartDate ;
                    sch:hasServiceEndDate   ?serviceEndDate ;
                    sch:hasServiceTitle     ?serviceTitle ;
                    sch:hasServiceTotalTime ?serviceTotalTime ;
                    sch:wasProvidedIn       ?org .
        ?org        sch:hasTitle            ?orgTitle .
    } ORDER BY  ASC(?serviceStartDate)
```

E. Experience in Executive Positions

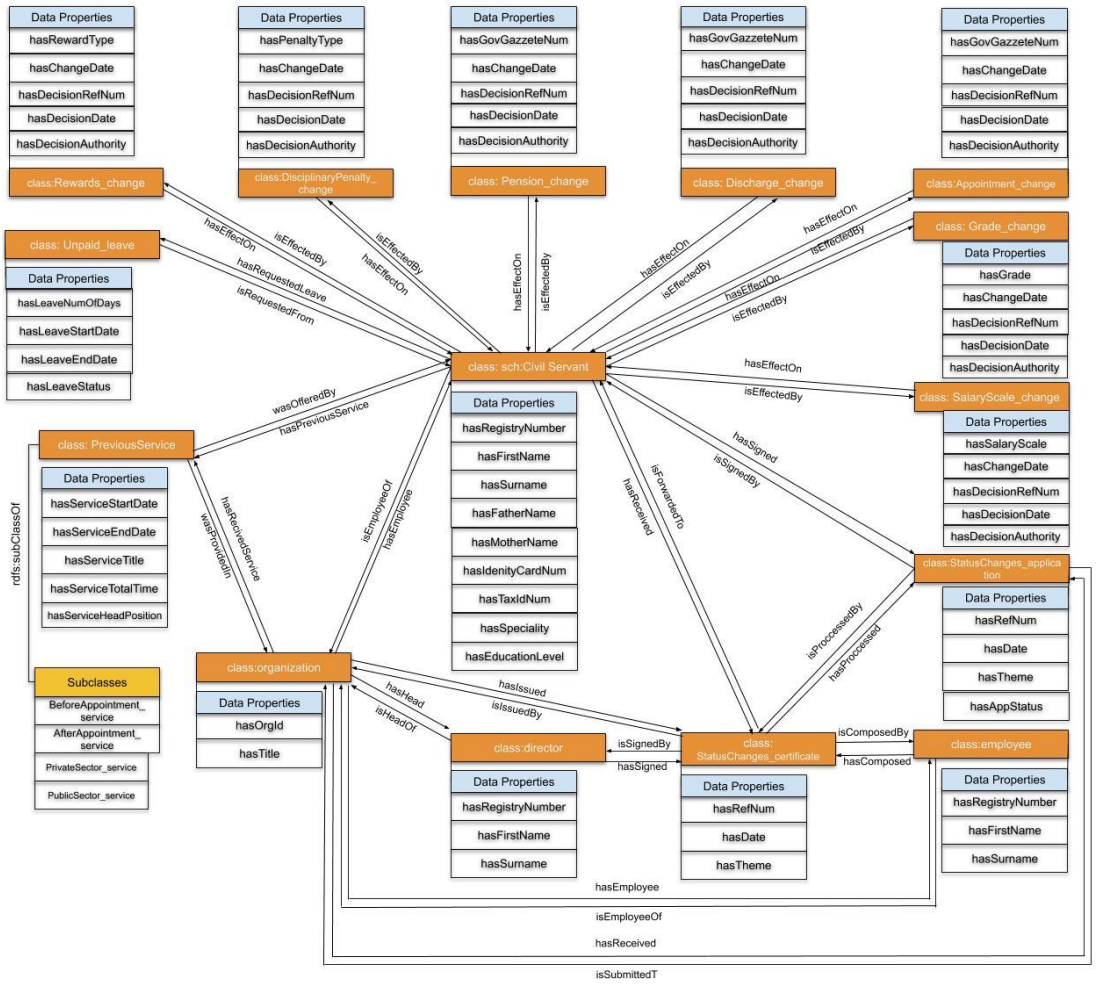
Another element listed on the certificate is tenure in positions of responsibility. To search for them, we run the query in figure 7.25. The query requests the instances of the AfterAppointment_service class offered (Object Property: wasOfferedBy) by the employee and the previous service was in a position of responsibility (Data Property: hasServiceHeadPosition == TRUE). The data fetched by this query are the service's start date, end date, title, total time, and the organization's title.

In figure 7.26 we can see the Classes, the Object Properties, and the Data Properties that we have to use in order to produce the Certificate of Employee's Status Changes.

Figure 7.25 SPARQL query for service after appointment

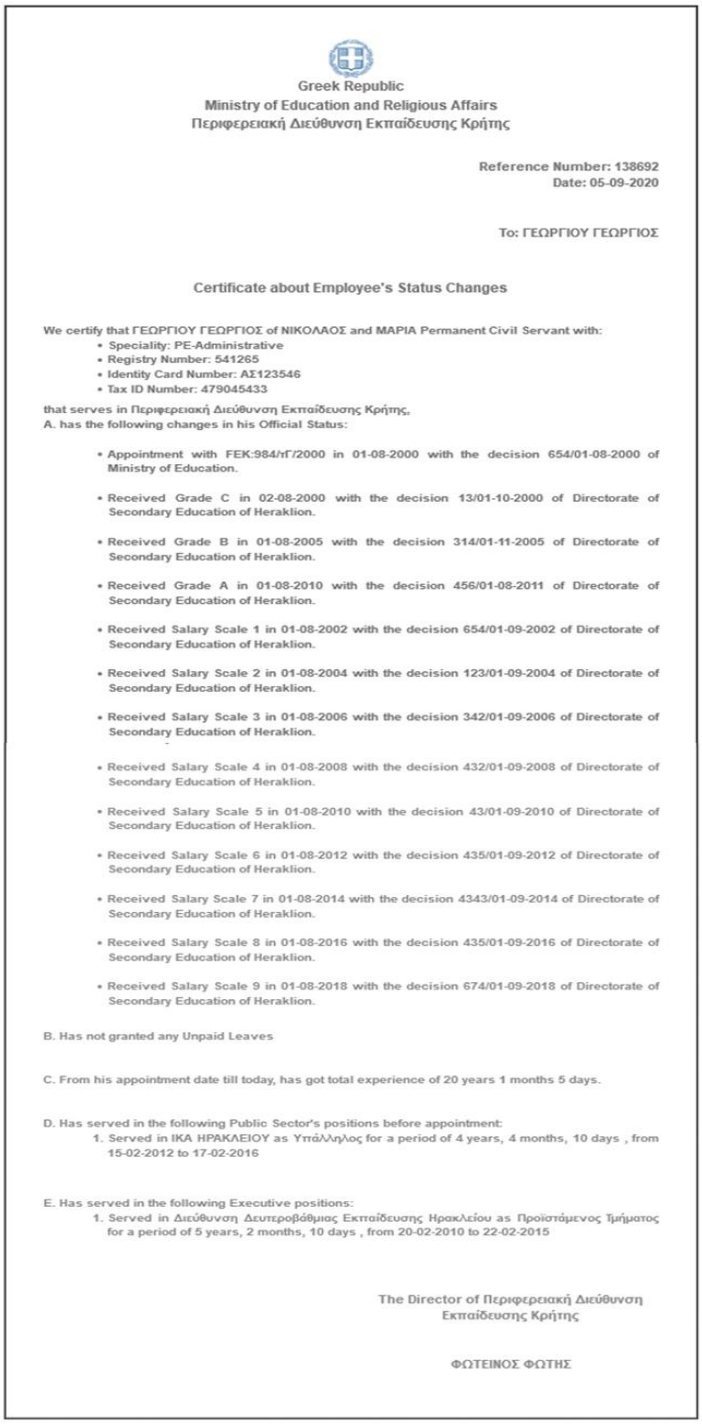
```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

JSON {
        'service':              ?service,
        'serviceStartDate':     ?serviceStartDate,
        'serviceEndDate':       ?serviceEndDate,
        'serviceTitle':         ?serviceTitle,
        'serviceTotalTime':     ?serviceTotalTime,
        'orgTitle':                 ?orgTitle
}
WHERE
    {
        ?service    rdf:type                    sch:AfterAppointment_service ;
                    sch:wasOfferedBy            sch:". $nickname  ." ;
                    sch:hasServiceStartDate     ?serviceStartDate ;
                    sch:hasServiceEndDate       ?serviceEndDate ;
                    sch:hasServiceTitle         ?serviceTitle ;
                    sch:hasServiceTotalTime     ?serviceTotalTime ;
                    sch:wasProvidedIn           ?org ;
                    sch:hasServiceHeadPosition  TRUE .
        ?org        sch:hasTitle                ?orgTitle .
} ORDER BY  ASC(?serviceStartDate)
```

Figure 7.26 Certificate of Employee's Status Changes - Classes, Object, and Data Properties



The result of all of the above is the Certificate we see in figure 7.27

Figure 7.27 Preview of Certificate of Employee's Status



Greek Republic
Ministry of Education and Religious Affairs
Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης

Reference Number: 138692
Date: 05-09-2020

To: ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ

Certificate about Employee's Status Changes

We certify that ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ of ΝΙΚΟΛΑΟΣ and ΜΑΡΙΑ Permanent Civil Servant with:

- Speciality: PE-Administrative
- Registry Number: 541265
- Identity Card Number: ΑΣ123546
- Tax ID Number: 479045433

that serves in Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης,
A. has the following changes in his Official Status:

- Appointment with FEK:984/τΓ/2000 in 01-08-2000 with the decision 654/01-08-2000 of Ministry of Education.
- Received Grade C in 02-08-2000 with the decision 13/01-10-2000 of Directorate of Secondary Education of Heraklion.
- Received Grade B in 01-08-2005 with the decision 314/01-11-2005 of Directorate of Secondary Education of Heraklion.
- Received Grade A in 01-08-2010 with the decision 456/01-08-2011 of Directorate of Secondary Education of Heraklion.
- Received Salary Scale 1 in 01-08-2002 with the decision 654/01-09-2002 of Directorate of Secondary Education of Heraklion.
- Received Salary Scale 2 in 01-08-2004 with the decision 123/01-09-2004 of Directorate of Secondary Education of Heraklion.
- Received Salary Scale 3 in 01-08-2006 with the decision 342/01-09-2006 of Directorate of Secondary Education of Heraklion.
- Received Salary Scale 4 in 01-08-2008 with the decision 432/01-09-2008 of Directorate of Secondary Education of Heraklion.
- Received Salary Scale 5 in 01-08-2010 with the decision 43/01-09-2010 of Directorate of Secondary Education of Heraklion.
- Received Salary Scale 6 in 01-08-2012 with the decision 435/01-09-2012 of Directorate of Secondary Education of Heraklion.
- Received Salary Scale 7 in 01-08-2014 with the decision 4343/01-09-2014 of Directorate of Secondary Education of Heraklion.
- Received Salary Scale 8 in 01-08-2016 with the decision 435/01-09-2016 of Directorate of Secondary Education of Heraklion.
- Received Salary Scale 9 in 01-08-2018 with the decision 674/01-09-2018 of Directorate of Secondary Education of Heraklion.

B. Has not granted any Unpaid Leaves

C. From his appointment date till today, has got total experience of 20 years 1 months 5 days.

D. Has served in the following Public Sector's positions before appointment:
1. Served in ΙΚΑ ΗΡΑΚΛΕΙΟΥ as Υπάλληλος for a period of 4 years, 4 months, 10 days , from 15-02-2012 to 17-02-2016

E. Has served in the following Executive positions:
1. Served in Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης Ηρακλείου as Προϊστάμενος Τμήματος for a period of 5 years, 2 months, 10 days , from 20-02-2010 to 22-02-2015

The Director of Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης

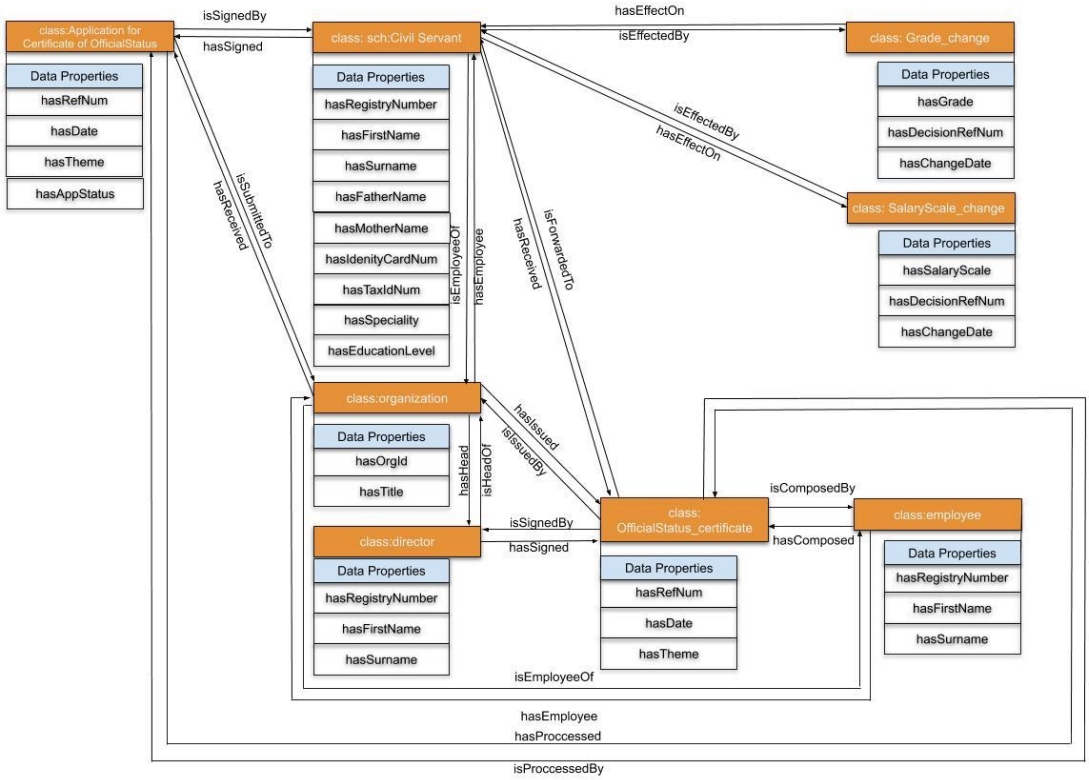
ΦΩΤΕΙΝΟΣ ΦΩΤΗΣ

- **Certificate of Employee's Official Status**

The Certificate of Employee's Official Status is a certificate which is a brief description of the official status of the employee. This is essentially a summary of the Certificate of Employee's Status Changes. For the data we need to compile it, we run some queries which provide us with the necessary information. The information we need is the details of the appointment and the current service. We have presented these queries above.

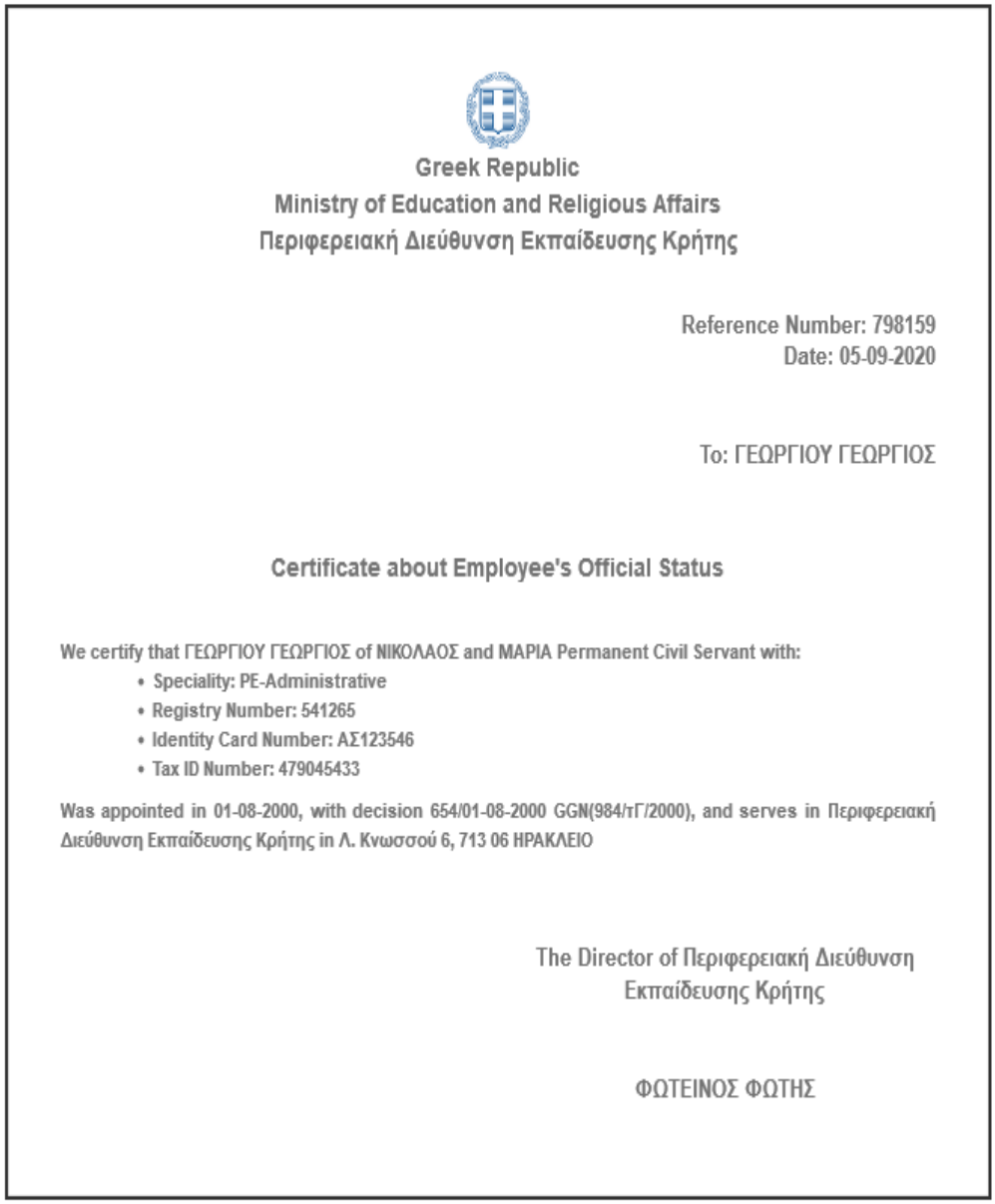
In figure 7.28 we can see the Classes, the Object Properties, and the Data Properties that we have to use in order to produce the Certificate of Employee's Official Status.

Figure 7.28 Certificate of Employee's Official Status - Classes, Object, and Data Properties



The result of all of the above is the Certificate we see in figure 7.29.

Figure 7.29 Preview of Certificate of Employee's Official Status



Greek Republic
Ministry of Education and Religious Affairs
Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης

Reference Number: 798159
Date: 05-09-2020

To: ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ

Certificate about Employee's Official Status

We certify that ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ of ΝΙΚΟΛΑΟΣ and ΜΑΡΙΑ Permanent Civil Servant with:

- Speciality: PE-Administrative
- Registry Number: 541265
- Identity Card Number: ΑΣ123546
- Tax ID Number: 479045433

Was appointed in 01-08-2000, with decision 654/01-08-2000 GGN(984/τΓ/2000), and serves in Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης in Λ. Κνωσσού 6, 713 06 ΗΡΑΚΛΕΙΟ

The Director of Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης

ΦΩΤΕΙΝΟΣ ΦΩΤΗΣ

- **Certificate of Employee's Previous Service**

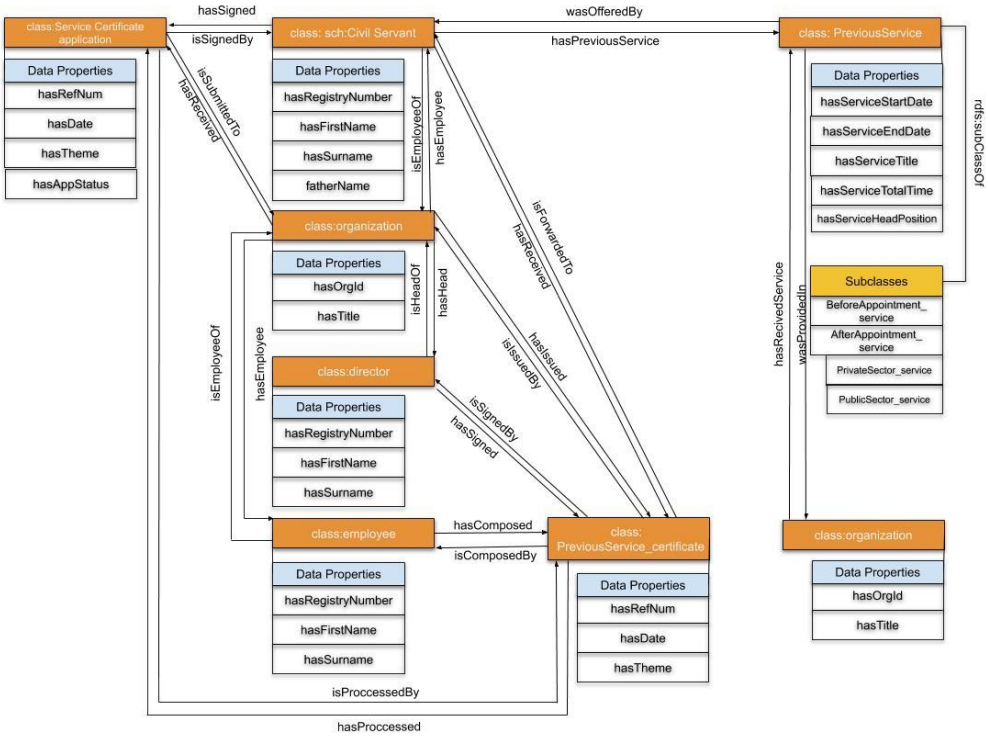
The Certificate of Employee's Previous Service contains a detailed description of the employee's services before and after his appointment. In order to collect the necessary data, in addition to the common queries mentioned at the beginning, some additional queries are executed. These queries are intended to gather information about his appointment details, post-appointment

services, and pre-appointment services. The last query was described in the description of the Certificate of Employee's Status Changes (figure 7.24). The items returned are the service's start date, end date, title, total time, and the organization's title.

The same information is returned from the query regarding the services after the appointment (figure 7.25). Besides, for each service, it is also mentioned if it was a position of responsibility. The search criterion is the instances of the AfterAppointment_service class offered (Object Property: wasOfferedBy) from the employee with the Registry Number that is in the "nickname".

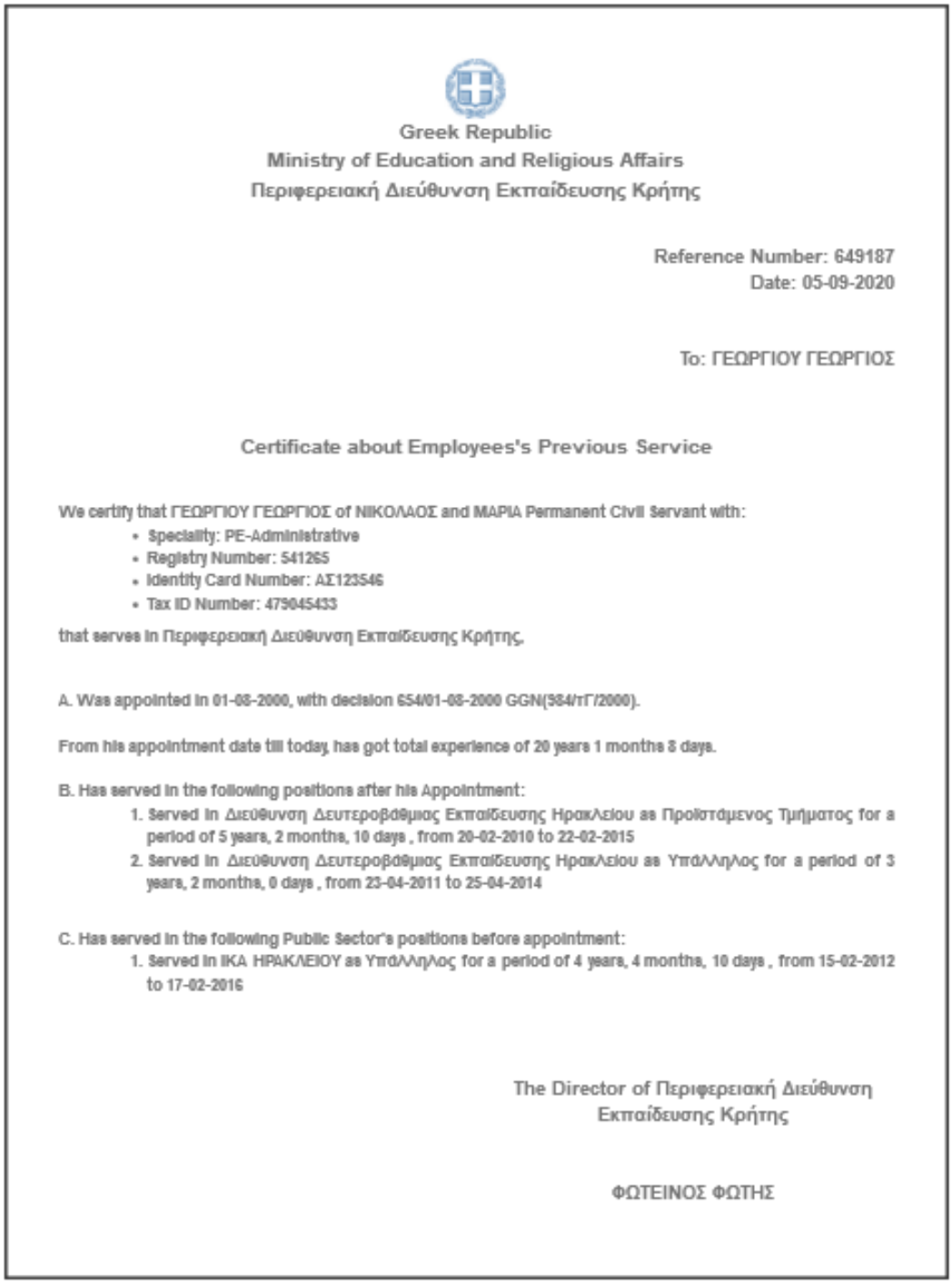
In figure 7.30 we can see the Classes, the Object Properties, and the Data Properties that we have to use in order to produce the Certificate of Employee's Previous Service.

Figure 7.30 Certificate of Employee's Previous Service - Classes, Object, and Data Properties



The result of all of the above is the Certificate we see in figure 7.31.

Figure 7.31 Preview of Certificate of Employee's Previous Service



Greek Republic
Ministry of Education and Religious Affairs
Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης

Reference Number: 649187
Date: 05-09-2020

To: ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ

Certificate about Employees's Previous Service

We certify that ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ of ΝΙΚΟΛΑΟΣ and ΜΑΡΙΑ Permanent Civil Servant with:

- Speciality: PE-Administrative
- Registry Number: 541265
- Identity Card Number: ΑΣ123546
- Tax ID Number: 479045433

that serves in Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης,

A. Was appointed in 01-08-2000, with decision 654/01-08-2000 GGN(384/τΓ/2000).

From his appointment date till today, has got total experience of 20 years 1 months 8 days.

B. Has served in the following positions after his Appointment:

1. Served in Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης Ηρακλείου as Προϊστάμενος Τμήματος for a period of 5 years, 2 months, 10 days , from 20-02-2010 to 22-02-2015
2. Served in Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης Ηρακλείου as ΥπάΛληλος for a period of 3 years, 2 months, 0 days , from 23-04-2011 to 25-04-2014

C. Has served in the following Public Sector's positions before appointment:

1. Served in ΙΚΑ ΗΡΑΚΛΕΙΟΥ as ΥπάΛληλος for a period of 4 years, 4 months, 10 days , from 15-02-2012 to 17-02-2016

The Director of Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης

ΦΩΤΕΙΝΟΣ ΦΩΤΗΣ

- **Preview of a Decision**

This process is used to view the leaves' decisions. To create the Decision, it is first necessary to collect some basic information, as we described during the process of issuing the certificates. With the queries of figures 7.19, 7.20, and 7.21 we collect data about the employee, the organization, and the document respectively. The difference is that the query that fetches the information about the document, is executed twice. The first time we search the details of the Decision that has been issued and the second the details of the application with which we requested the leave.

The additional information we need to look for is the leave information. The query that fetches them is shown in figure 7.32. For the query, we use the Reference Number of the leave's application. The query first requests the instance of the application that has a Reference Number (Object Property: hasRefNum) the number we transfer as a parameter (line 16). It then searches for the instance of the leave that is requested (Object Property: isRequestedWith) with the above application (line 17). Then, the query requests and receives the leave's type, start date, end date, duration, status, and type's label.

The template used for the decision has the same basic features as the certificate template. It indicates the details of the organization and the hierarchy of the organizations supervising it, the reference number, and the date of issue, the subject of the decision, and the signatory of the document.

The decision also contains the Relevant Legislation section. This section refers to the relevant legislation on which the decision is based. This point also refers to the application that caused the issuance of this decision, which we retrieved with the query mentioned above. Finally, depending on the status of the leave (approved or rejected), the text that will be included in the operative part of the Decision is selected.

In figure 7.33 we can see the Classes, the Object Properties, and the Data Properties that we have to use in order to produce the Decision: The result of all of the above is the Certificate we see in figure 7.34.

Figure 7.32 SPARQL query for a decision

```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

JSON {
        'leave':            ?leave,
        'leaveType':        ?leaveType,
        'leaveStartDate':   ?leaveStartDate,
        'leaveEndDate':     ?leaveEndDate,
        'leaveNumOfDays':   ?leaveNumOfDays,
        'leaveStatus':      ?leaveStatus,
        'leaveTypeLabel':   ?leaveTypeLabel
}
WHERE
    {   ?app        sch:hasRefNum           ". $dec  ." .
        ?leave      sch:isRequestedWith     ?app .
        ?leave      rdf:type*               ?leaveType ;
                    sch:hasLeaveStartDate   ?leaveStartDate ;
                    sch:hasLeaveEndDate     ?leaveEndDate ;
                    sch:hasLeaveNumOfDays   ?leaveNumOfDays ;
                    sch:hasLeaveStatus      ?leaveStatus .
        ?leaveType  rdfs:subClassOf         sch:Leave ;
                    rdfs:label              ?leaveTypeLabel .
}
```

Figure 7.33 Decision for a leave - Classes, Object, and Data Properties

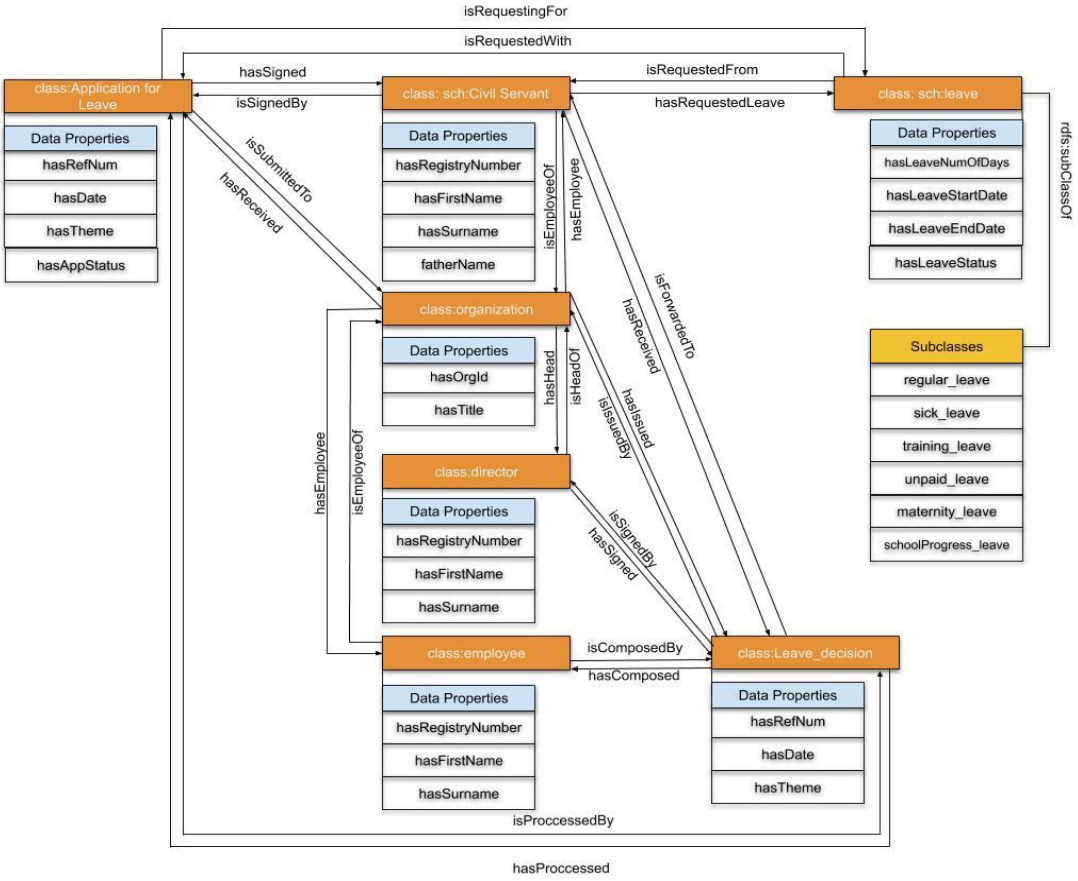
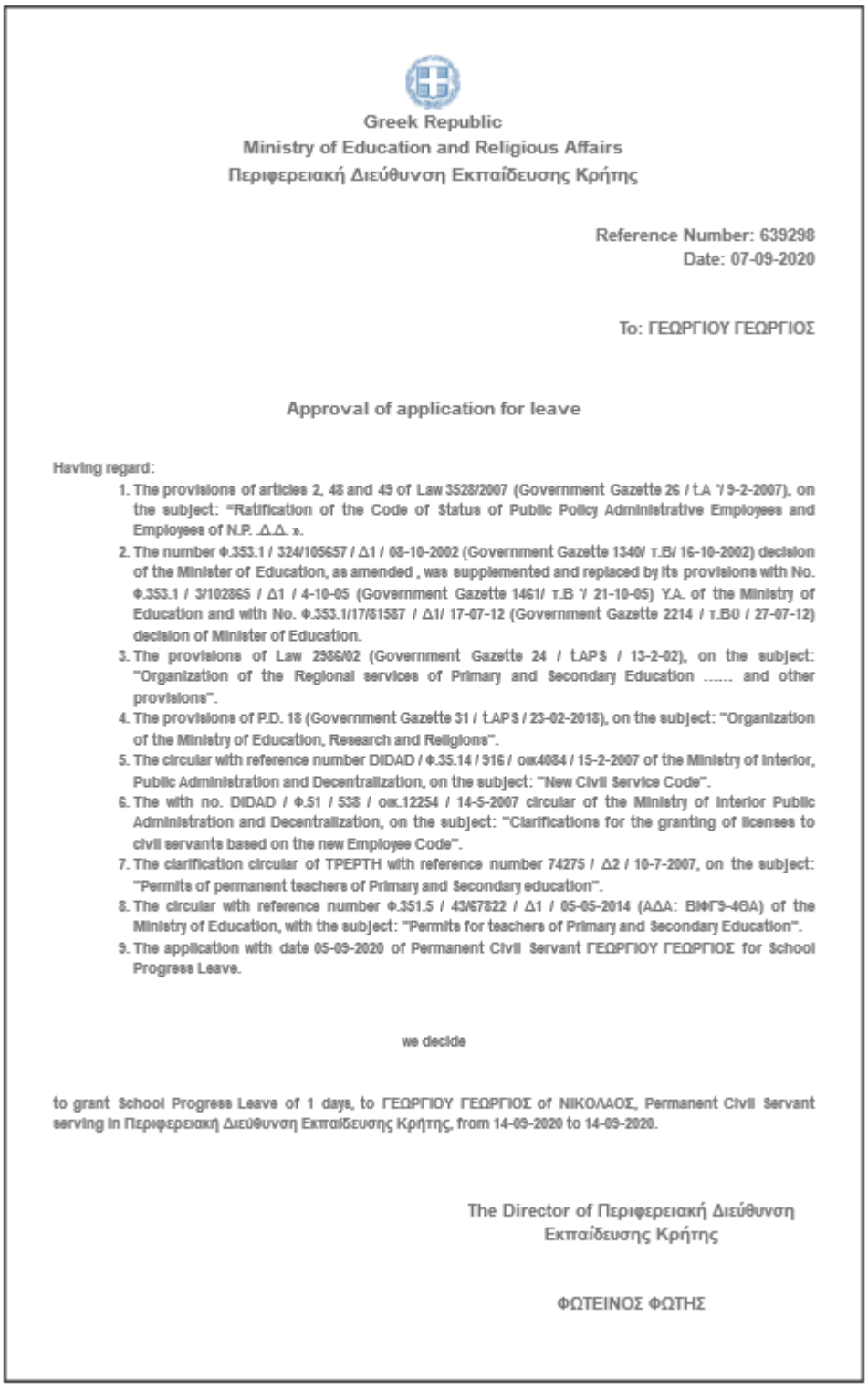


Figure 7.34 Preview for a Decision for a leave



Greek Republic
Ministry of Education and Religious Affairs
Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης

Reference Number: 639298
Date: 07-09-2020

To: ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ

Approval of application for leave

Having regard:

1. The provisions of articles 2, 48 and 49 of Law 3528/2007 (Government Gazette 26 / τ.Α '/ 9-2-2007), on the subject: "Ratification of the Code of Status of Public Policy Administrative Employees and Employees of N.P. .Δ.Δ. ».
2. The number Φ.353.1 / 324/105657 / Δ1 / 08-10-2002 (Government Gazette 1340/ τ.Β/ 16-10-2002) decision of the Minister of Education, as amended , was supplemented and replaced by its provisions with No. Φ.353.1 / 3/102865 / Δ1 / 4-10-05 (Government Gazette 1461/ τ.Β '/ 21-10-05) Y.A. of the Ministry of Education and with No. Φ.353.1/17/81587 / Δ1/ 17-07-12 (Government Gazette 2214 / τ.Β0 / 27-07-12) decision of Minister of Education.
3. The provisions of Law 2986/02 (Government Gazette 24 / τ.ΑΡ$ / 13-2-02), on the subject: "Organization of the Regional services of Primary and Secondary Education ..... and other provisions".
4. The provisions of P.D. 18 (Government Gazette 31 / τ.ΑΡ$ / 23-02-2018), on the subject: "Organization of the Ministry of Education, Research and Religions".
5. The circular with reference number DIDAD / Φ.35.14 / 916 / οικ.4084 / 15-2-2007 of the Ministry of Interior, Public Administration and Decentralization, on the subject: "New Civil Service Code".
6. The with no. DIDAD / Φ.51 / 538 / οικ.12254 / 14-5-2007 circular of the Ministry of Interior Public Administration and Decentralization, on the subject: "Clarifications for the granting of licenses to civil servants based on the new Employee Code".
7. The clarification circular of ΤΡΕΡΤΗ with reference number 74275 / Δ2 / 10-7-2007, on the subject: "Permits of permanent teachers of Primary and Secondary education".
8. The circular with reference number Φ.351.5 / 43/67822 / Δ1 / 05-05-2014 (ΑΔΑ: ΒΙΦΓ9-4ΘΑ) of the Ministry of Education, with the subject: "Permits for teachers of Primary and Secondary Education".
9. The application with date 05-09-2020 of Permanent Civil Servant ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ for School Progress Leave.

we decide

to grant School Progress Leave of 1 days, to ΓΕΩΡΓΙΟΥ ΓΕΩΡΓΙΟΣ of ΝΙΚΟΛΑΟΣ, Permanent Civil Servant serving in Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης, from 14-09-2020 to 14-09-2020.

The Director of Περιφερειακή Διεύθυνση Εκπαίδευσης Κρήτης

ΦΩΤΕΙΝΟΣ ΦΩΤΗΣ

7.6 Search by name

This service is offered to users of the Web-Portal without login, as it is a stage 1 service. The process of "search by name" applies to both the search for Administrative Organizations and the search for school units. The process is simple both from the user's point of view and from the web-portal and endpoint side. By selecting the option from the main menu, the user is transferred to a form where he fills in the name of the entity he is looking for or a part of it and submits his query by pressing the relevant button.

The Web-Portal then compiles the SPARQL query by integrating the user input and submitting it to the SPARQL Endpoint. The Endpoint then returns the search results, which the Web-portal presents to the user.

In figure 7.35 we present the query for the search for a school unit. With this query, we ask Endpoint to return all the instances that belong to the School class (line 15) and we filter them based on the user input (line 25). The information returned to us is the school's title, address, telephone, fax, website, email, and the title of the supervising organization.

Figure 7.35 SPARQL query for searching school by name

```
1  PREFIX sch: <http://www.pdekritis.gr/semantic#>
2  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4  JSON {
5          'address': ?address,
6          'tel':  ?tel,
7          'fax': ?fax,
8          'title':  ?title,
9          'email':  ?email,
10         'webSite': ?webSite,
11         'dirSupTitle': ?dirSupTitle
12 }
13 WHERE
14 {
15     ?x rdfs:subClassOf* sch:School .
16     ?y rdf:type ?x ;
17         sch:hasTitle ?title ;
18         sch:hasAddress ?address ;
19         sch:hasTel ?tel ;
20         sch:hasFax ?fax ;
21         sch:hasEmail ?email ;
22         sch:hasWebSite ?webSite ;
23         sch:isSupervisedBy ?dirSup .
24     ?dirSup sch:hasTitle ?dirSupTitle .
25     filter regex(?title, \"".$_POST['inputname']."\", \"i\")
26 }
```

The same query is executed when searching Organization by name, except that in it we ask to return the instances that belong to the Administration class (line 15) and we filter them based on the user input (line 25).

Figure 7.36 shows the results of a search where user input was number “4”.

Figure 7.36 Search results

Title	Address	Telephone	Fax	Email	Web Site	Supervisor
4ο Δημοτικό Σχολείο Χανίων	Πλατεία Αρκαδίου, 731 31 Χανιά	2147483647	2147483647	mail@4dim-chanion.chan.sch.gr	https://4dimchanion.blogspot.com/	Διεύθυνση Πρωτοβάθμιας Εκπαίδευσης Χανίων
4ο ΓΕΝΙΚΟ ΛΥΚΕΙΟ ΗΡΑΚΛΕΙΟΥ	ΦΙΛΙΠΠΟΥΠΟΛΕΩΣ 45, ΤΚ 71305	2147483647	2147483647	mail@4lyk-irakl.ira.sch.gr	http://4lyk-irakl.ira.sch.gr	Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης Ηρακλείου
4ο Γενικό Λύκειο Χανίων	Εμμ. Μουντάκη 1	2147483647	2147483647	mail@4lyk-chanion.chan.sch.gr	http://4lyk-chanion.chan.sch.gr	Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης Χανίων
4ο Γυμνάσιο Χανίων	Απτέρων 2 Κουμπές, 73134, Χανιά	2147483647	2147483647	4gymchan@sch.gr	http://4gym-chanion.chan.sch.gr/	Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης Χανίων
4ο ΕΠΑ.Λ. ΗΡΑΚΛΕΙΟΥ (ΕΣΠΕΡΙΝΟ)	ΣΚΕΠΕΤΖΗ 31, ΤΚ 71307	2147483647	2147483647	mail@4epal-esp-irakl.ira.sch.gr	http://4epal-esp-irakl.ira.sch.gr	Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης Ηρακλείου

7.7 Hierarchical Search

This service is also offered to users of the Web-Portal without login, as it is a stage 1 service. The hierarchical search presents the supervised bodies of an organization (schools and administrations). The data we view are stored in Endpoint and with the appropriate queries, we can view it. The process is similar in the search for schools and the search for organizations. Below we describe the hierarchical search of schools.

Step 1: Selecting the option from the main menu the following query (figure 7.37) is executed. The query provides the details of the Regional Directorates. The user can select one or all the Regional Addresses to view the Directorates under them.

Step 2: After the user selects for the Regional Directorates, a SPARQL query is compiled that provides the Directorates’ information, which we see in figure 7.38. The user can select one or all the Directorates.

Step 3: Then the query is executed that fetches the type of schools which we see in figure 7.39.

Figure 7.37 SPARQL query for Regional Directorates

```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
JSON {
        'x': ?x,
        'title': ?title,
        'address': ?address,
        'tel':  ?tel,
        'fax': ?fax,
        'email':  ?email,
        'webSite': ?webSite
}
WHERE
{
        ?x rdf:type      sch:RegionalDirectorate ;
                         sch:hasTitle ?title ;
                         sch:hasAddress ?address ;
                         sch:hasTel ?tel ;
                         sch:hasEmail ?email .
}
```

Figure 7.38 SPARQL query for Directorates

```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

JSON {
    'z':        ?z,
    'title':    ?title,
    'address':  ?address,
    'tel':      ?tel,
    'fax':      ?fax,
    'email':    ?email,
    'webSite':  ?webSite
}
WHERE
{
    ?y      rdfs:subClassOf*    sch:Directorate .
    ?z      rdf:type            ?y ;
            sch:hasTitle        ?title ;
            sch:hasAddress      ?address ;
            sch:hasTel          ?tel ;
            sch:hasFax          ?fax ;
            sch:hasWebSite      ?webSite ;
            sch:hasEmail        ?email ".  $cond0 ." .
}
```

Figure 7.39 SPARQL query for School's types

```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

JSON {
        'schType':          ?schType,
        'schTypeLabel':     ?schTypeLabel
}
WHERE
{
    ?schType    rdfs:subClassOf*    sch:School ;
                rdfs:label          ?schTypeLabel .
}
```

After the user selects the type of school the last query is compiled which returns the search results (figure 7.40). The choices made by the user at each step are stored in a table that is transferred by a POST variable to the next step.

Figure 7.40 SPARQL query with all user's options

```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

JSON {
        'address':      ?address,
        'tel':          ?tel,
        'fax':          ?fax,
        'title':        ?title,
        'email':        ?email,
        'webSite':      ?webSite,
        'dirSupTitle':  ?dirSupTitle
}
WHERE
{       ". $cond2 ."    sch:hasTitle        ?title ;
                        sch:hasAddress      ?address ;
                        sch:hasTel          ?tel ;
                        sch:hasFax          ?fax ;
                        sch:hasEmail        ?email ;
                        sch:hasWebSite      ?webSite ". $cond1 ." ;
                        sch:isSupervisedBy  ?dirSup .
        ?dirSup         sch:hasTitle        ?dirSupTitle ". $condReg  ." .
}
```

Chapter 8 Evaluation

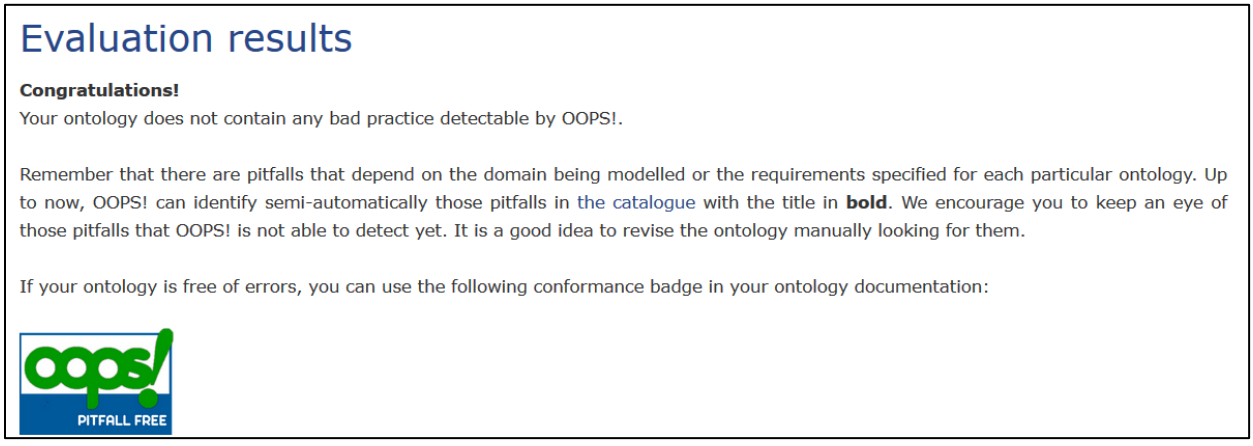
The evaluation of all this work was carried out in two stages. In the first stage, the Ontology was checked and evaluated. In the second stage, the Web-Portal was evaluated.

8.1 Evaluation of the Ontology

For the evaluation of the Ontology we first used the Pellet reasoner. Pellet is a complete OWL-DL reasoner. It is open-source and it accessible through various interfaces. In our case we used the Pellet plugin of Protégé. As its developers [36] state "It offers a panoply of features including conjunctive query answering, rule support, E-Connection reasoning, and axiom pinpointing, among others". Using Pellet we checked the Ontology for inconsistencies and errors that may have occurred during the design and implementation.

After this, we used another tool to check the Ontology for inconsistencies and pitfalls. OOPS (OntOlogy Pitfall Scanner) [37], is a tool for detecting pitfalls in ontologies. This tool checks an ontology and predicts potential problems that may arise. This prediction is based on a list of bad practices "pitfalls". OOPS tries to identify features that often represent a problem or that could lead to ontology errors. As we see in Figure 8.1 our ontology is free of pitafalls.

Figure 8.1 Evaluation results from OOPS



8.2 Evaluation of the Web-Portal

In addition to the evaluation of the Ontology, we also proceeded to the evaluation of the Web-Portal. The dataset we used to test the functions of the Web-portal contains a sufficient number of individuals for evaluation. For this purpose, a survey was conducted in which 42 people used the Web-Portal and then completed a questionnaire in which they recorded their experience of using it. Using Pellet we checked the Ontology for inconsistencies and errors that may have occurred during the design and implementation.

The participants in the research were civil servants, administrative employees, and teachers, who serve in decentralized organizations of the Ministry of Education. As the Ontology is oriented towards the development of functions in the field of government to employee, the participants had the advantage that they could participate in a dual role. More specifically, they were invited to use the Web-Portal both as users who wish to submit their requests, but also as representatives of the administration that needs to process requests that have been submitted to the Organization. For this purpose, they were granted credentials for their connection to different roles.

8.2.1 Description of the Questionnaire

The questionnaire was completed anonymously and contained 28 questions divided into 4 sections. In the first section, the user entered his demographic data. In the second part, the user was asked to answer questions about his experience as a user, customer of the services. The third section contained questions about his experience as a user representative of the administration. Finally, in the fourth section, the user answered questions about his overall experience from the platform. Let's see the questions in detail:

Section 1: Demographics

In this section, we use multiple choice questions, for the collection of demographic data.

- **What is your gender?**
- **Which age group do you belong to?**

We have grouped the ages in the groups 18-29, 30-39, 40-49, and above 50 years old. The user selects his age group by selecting the corresponding radio button.

- **How many years of working experience do you have?**

We have grouped the working experience in the groups 0-5, 6-10, 11-15, 16-20, 20-25, and above 25 years. The user selects his experience group by selecting the corresponding radio button.

- **Are you trained in computer use?**

Section 2: Evaluation as an employee

The questions in this section are answered by completing a linear scale. The rating of the scale is from 1 (not at all) to 5 (very much). These options correspond to the respective options on the Likert scale: 1-Very Dissatisfied, 2-Not Satisfied, 3-Neutral, 4-Satisfied, 5-Very Satisfied. The questions that users are asked are the following:

- **How satisfied are you with using the Login / Register feature?**
- **How satisfied are you with using the Search by Name feature?**
- **How satisfied are you with the use of the Hierarchical Search function?**
- **How satisfied are you with the use of the Certificate Application feature?**
- **How satisfied are you with using the Leave Application feature?**
- **How satisfied are you with the use of Personal Repository feature?**

Section 3: Evaluation as a representative of the administration

The questions in this section are also answered by completing the same linear scale, from 1 (not at all) to 5 (very much), that we referred above. The questions in this section are:

- **How satisfied are you with using the Administration's Dashboard feature?**
- **How satisfied are you with the use of the Approve / Reject application?**
- **How satisfied are you with the use of the Decision View feature?**
- **How satisfied are you with the use of the Certificate View feature?**
- **How satisfied are you with the use of the Employee View feature?**
- **How satisfied are you with using the View Applications feature?**

- **How satisfied are you with the use of the Certificate View feature?**
- **How satisfied are you with the use of the Decision View feature?**

Section 4: Overall Evaluation of the Web-Portal

The first three questions in this section are answered by completing the linear scale, we used in the previous sections.

- **How satisfied are you with the use of the Web-Portal?**
- **How is the Web-Portal evaluated in terms of ease of use?**
- **How satisfied are you with the speed with which your requests are processed?**

The next six questions are divided into three pairs. The first question of each pair is answered with yes or no and asks if the user encountered any type of difficulties. In case he encountered any difficulties, he completes with a free text the second question of the pair, providing clarifications about the difficulties he encountered.

- **Did you encounter difficulties in understanding the functions?**
- **If you had difficulties in understanding the functions, please describe the difficulties you encountered.**
- **Did you encounter any technical problems?**
- **If you encountered technical problems, please describe the problems you encountered.**
- **Did you successfully complete all your requests?**
- **If you encountered problems completing your requests, please describe the issues you encountered.**

In the last question, the user can fill in with free text comments and suggestions for the improvement of the Web Portal.

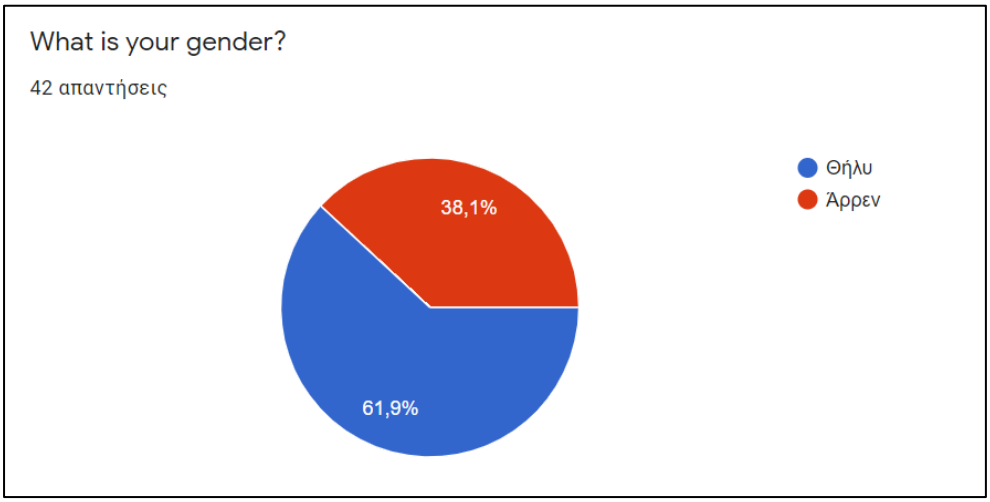
- **Suggestions for improving the functions of the Web-Portal:**

8.2.2 Results of the Survey

Section 1: Demographics

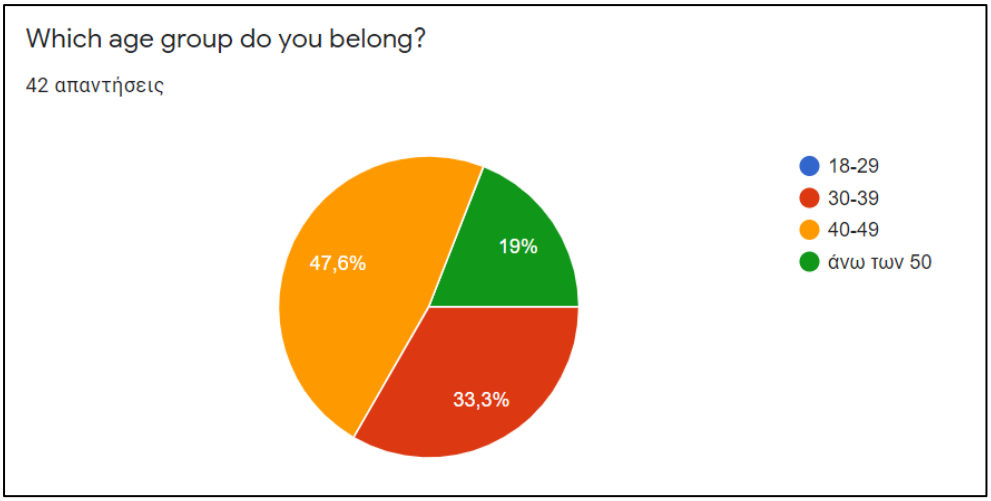
The research involved 42 people, 26 women (61.9%), and 16 men (38.1%).

Figure 8.2 Participant's gender



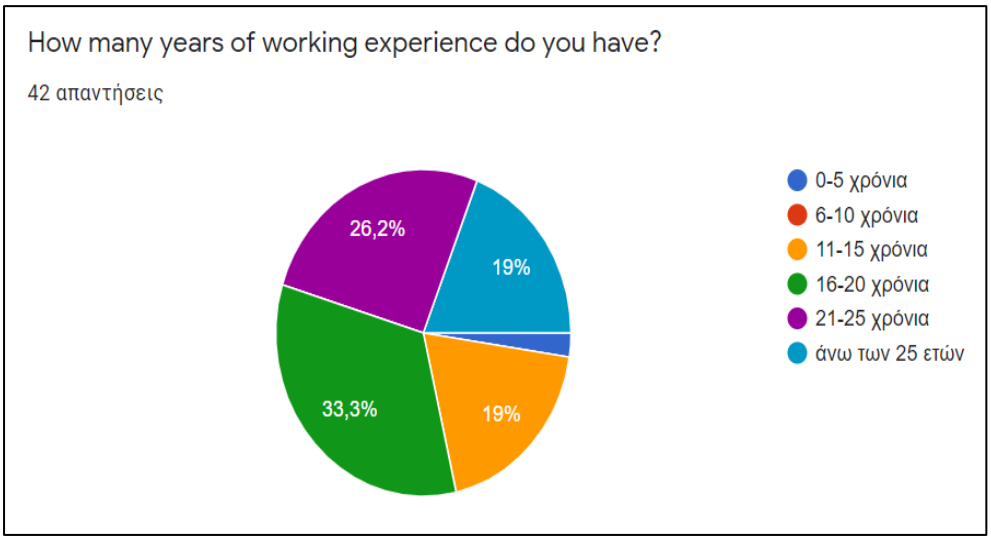
None of the participants belonged to the age group 18-29. In the age group 30-39, belonged to 14 participants (33.3%), while in the age group 40-49 belonged 20 participants (47.6%). Finally, eight people (19%) were over 50 years old.

Figure 8.3 Participant's age



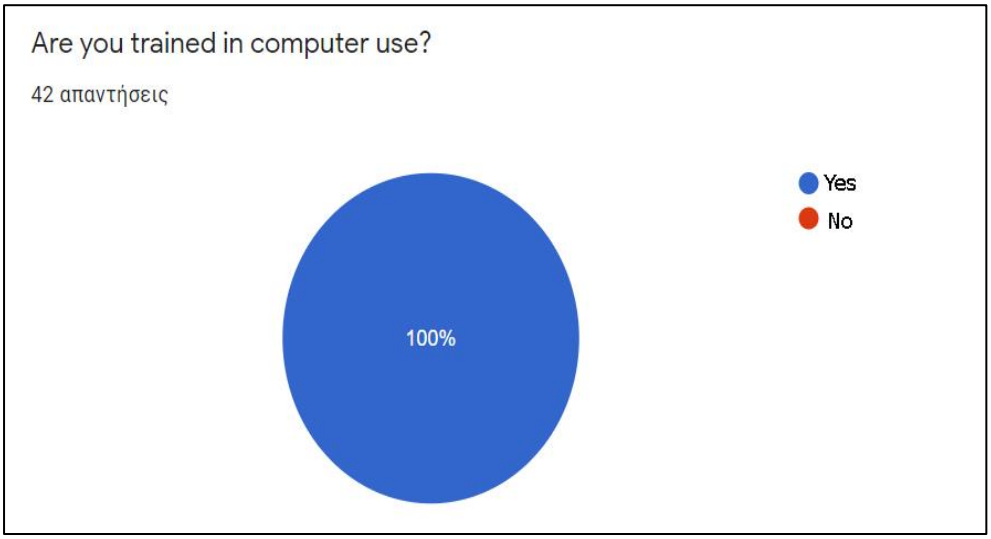
Regarding work experience, we observe that no participants were belonging to the group 6-10 years, while the group 0-5 years had only one person (2.4%). The main reason for this is the reduction of recruitment in the Public Sector in the last decade. Beyond that, 8 participants (19%) have 11-15 years of work experience, while 14 people (33.3%) have worked for 16-20 years. In the group of 21-25 years, there were 11 employees (26.2%). Finally, 8 people (19%) have more than 25 years of work.

Figure 8.4 Participant's working experience



All participants in the survey have at least an elementary education in using computers.

Figure 8.5 Participant's computer training



Section 2: Evaluation as an employee

In this section, users responded about their experience as Web-Portal client users. Essentially, the functions they used and evaluated were login, search by name, hierarchical search, certificate request, leave request, and the Personal Repository where users have access to data concerning them.

The user rating in this section was positive. In all the functions the vast majority of users stated that they are very satisfied with the functions they used. The Login/Register function left the users very satisfied at 90.5% and satisfied at 9.5%. The search by name received very good reviews at 97.6% (very satisfied), while the hierarchical search did not have the same success collecting good reviews at 85.7%(very satisfied), and good reviews at 14.3%(satisfied). Regarding the functions of the applications, the application for a certificate and the application for a leave left the users very satisfied, at a rate of 97.6%. On the other hand, the functions of the Personal Repository, although they received only positive reviews, did not seem to satisfy the users as well, as 83.3% of the users stated that they are very satisfied and 16.7% satisfied.

Figure 8.6 User's satisfaction with Login/Register

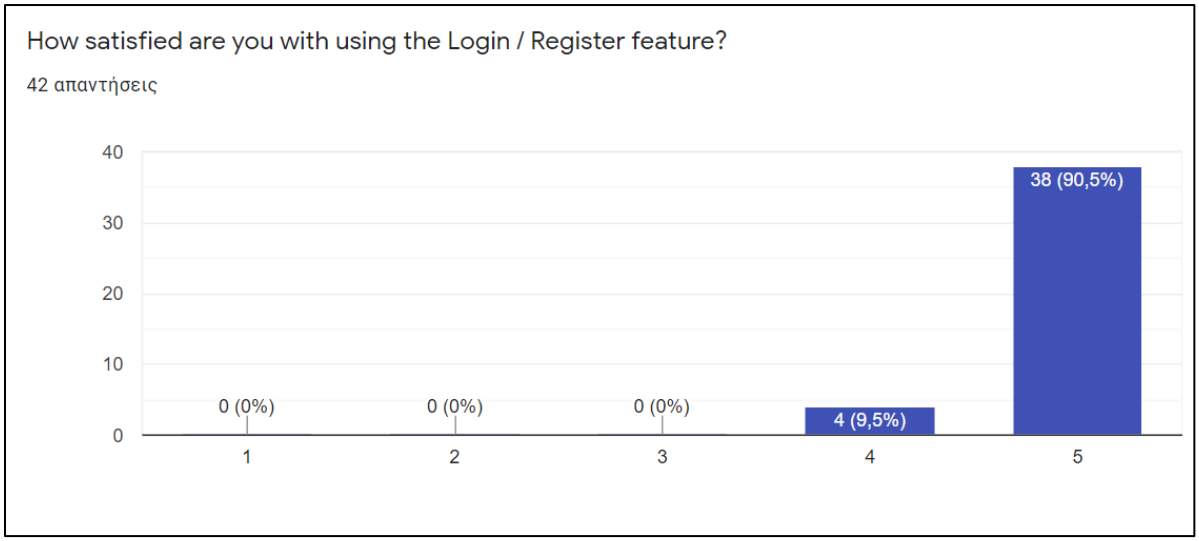


Figure 8.7 User's satisfaction with Search by name

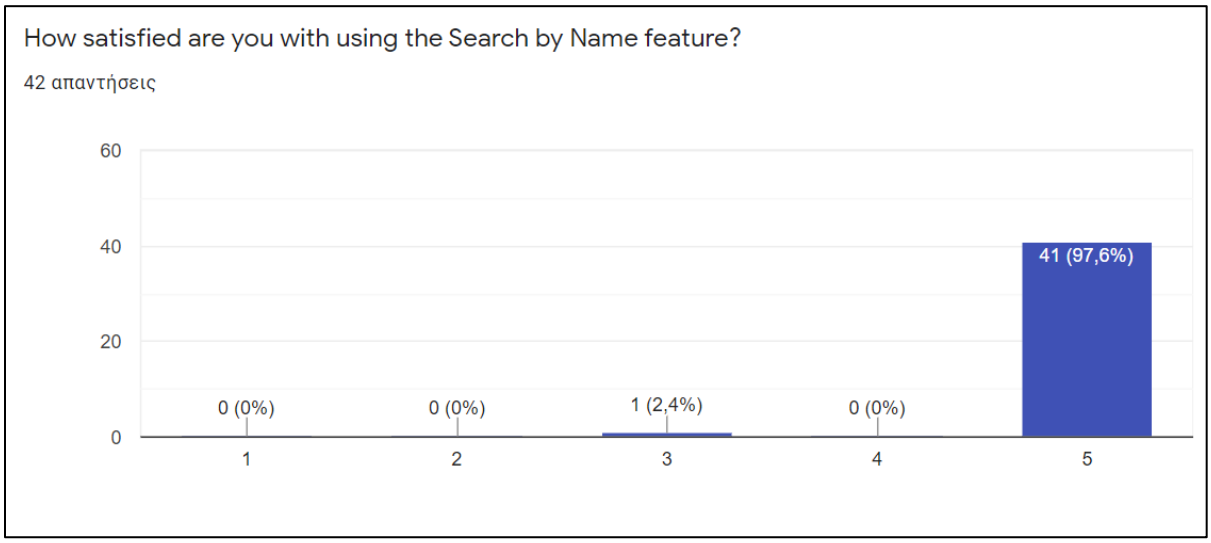


Figure 8.8 User's satisfaction with hierarchical search

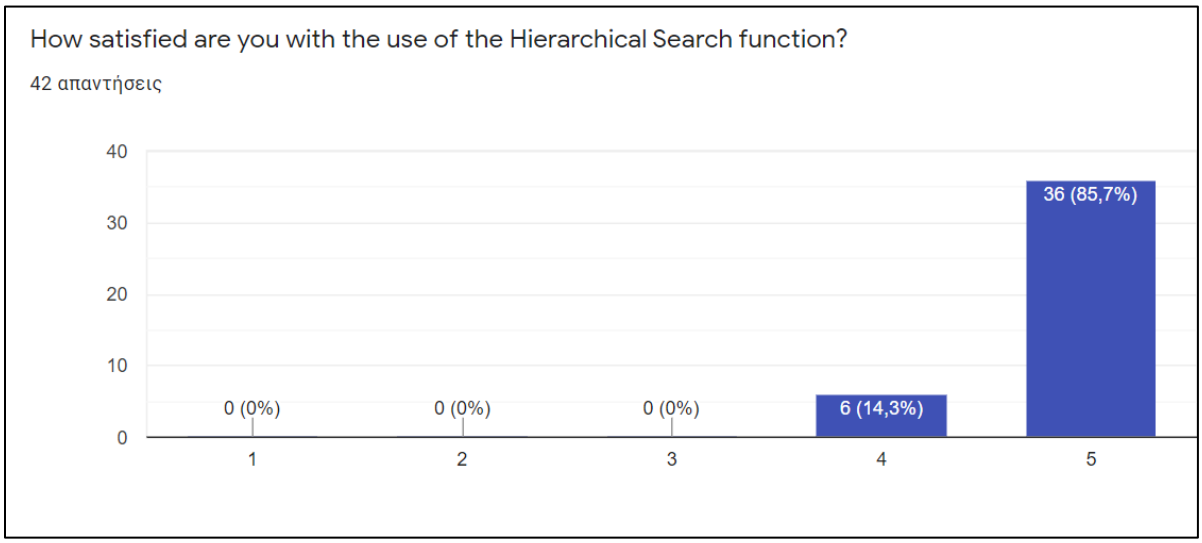


Figure 8.9 User's satisfaction with Application for Certificate

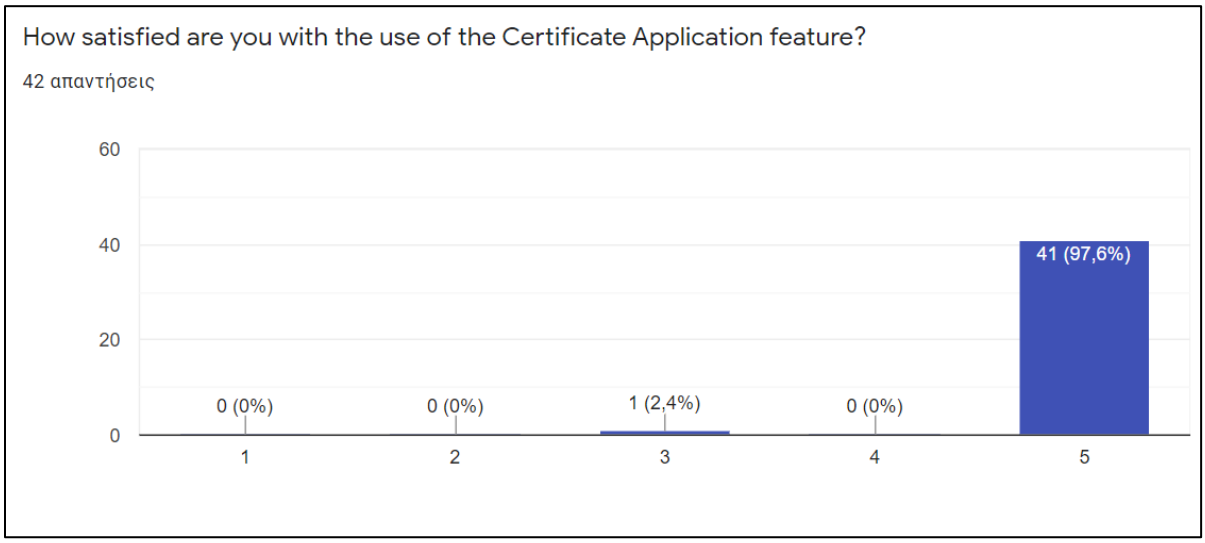


Figure 8.10 User's satisfaction with Application for leave

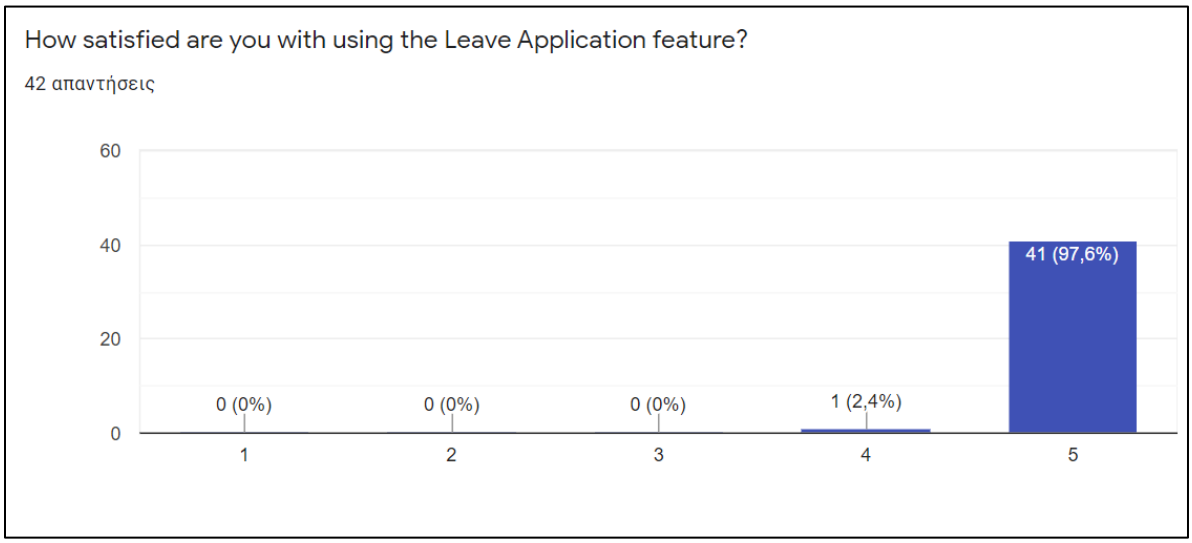
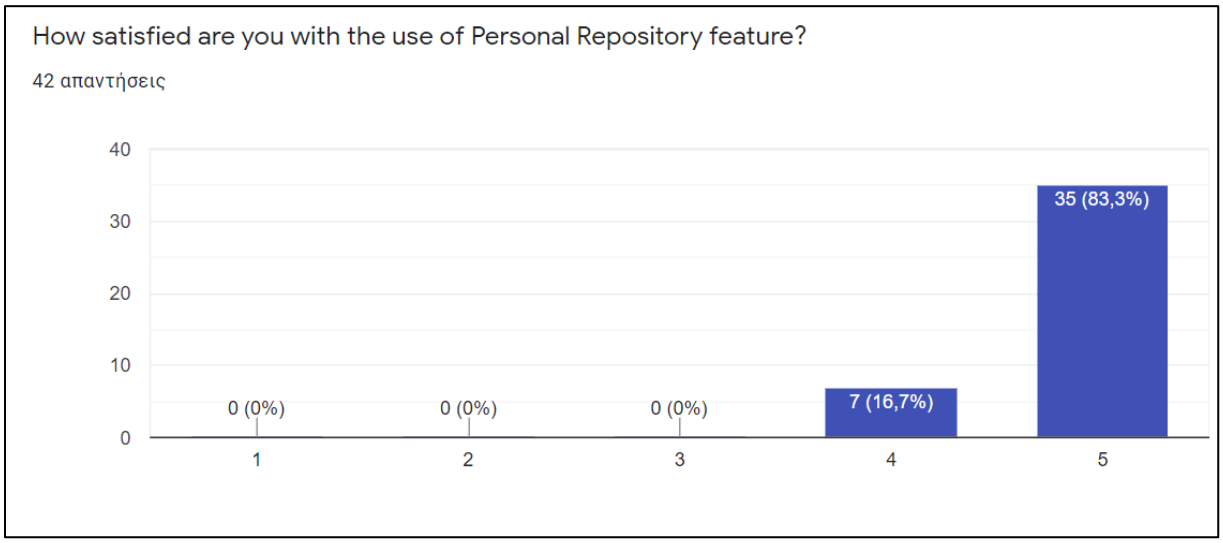


Figure 8.11 User's satisfaction with Personal Repository



Section 3: Evaluation as a representative of the administration

In this section, users recorded their views on the functions that the administration representative can use to manage the personnel and process their requests. Administration's Dashboard functions received positive reviews and 83.3% of users stated they were very satisfied, 14.3% were satisfied, and 2.4% were neutral.

The approval/rejections function experienced overwhelming user acceptance, with 97.6% saying they were very satisfied and 2.4% saying they were satisfied. Positive reviews were also received, but not so strongly, by the functions of displaying Decisions and Certificates. For both of them, the users stated very satisfied at a rate of 90.5%. User reviews were also positive for the Employees View and Applications View features, but with lower acceptance rates. Specifically, for both of them, the users stated very satisfied at a rate of 81%. For Employee View 11.9% of users stated satisfied and 7.1% expressed a neutral opinion, while for Applications View 14.3% of users stated satisfied and 4.8% expressed a neutral opinion. Better acceptance received the Decision and the Certificate View, for which the users stated very satisfied at a rate of 97.6% and 95.2% respectively. Moreover, users were satisfied with these features at a rate of 2.4% and 4.8% respectively.

Figure 8.12 User's satisfaction with Administration's Dashboard

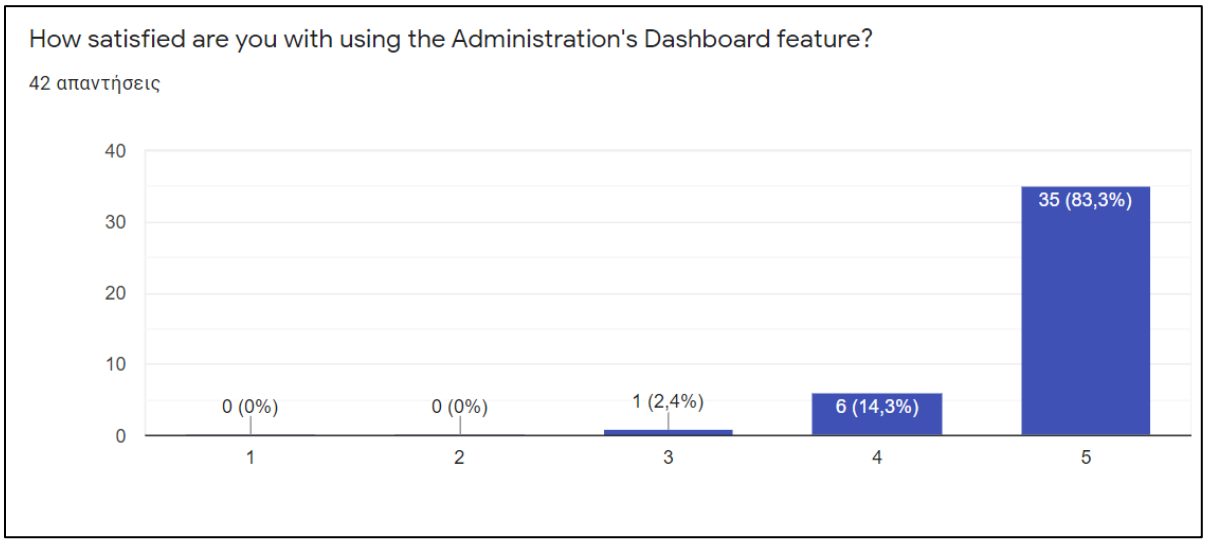


Figure 8.13 User's satisfaction with Approve/Reject of an application

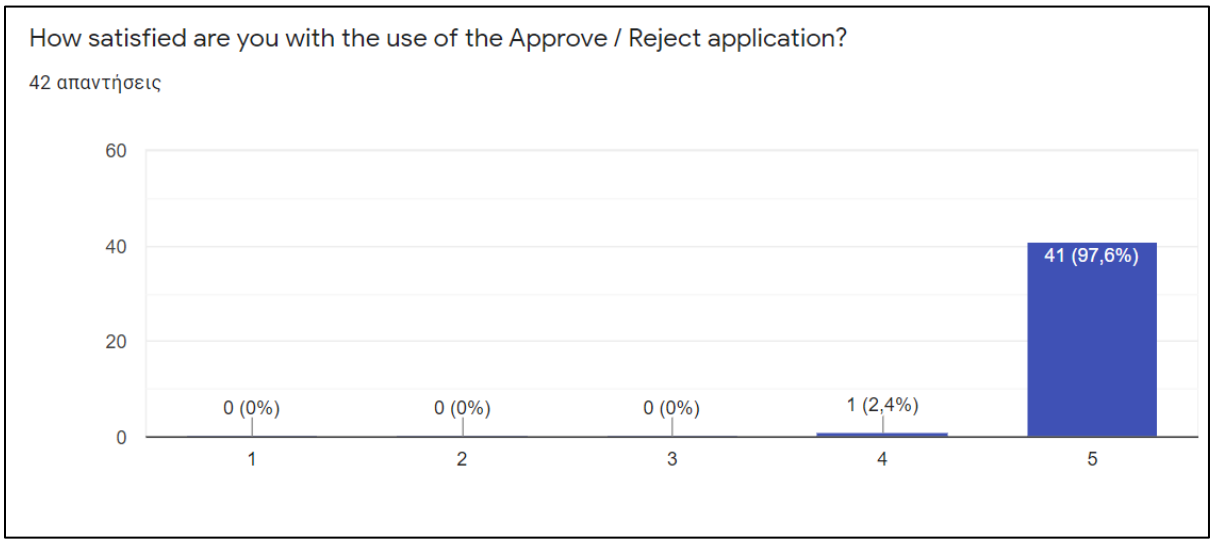


Figure 8.14 User's satisfaction with preview of a Decision

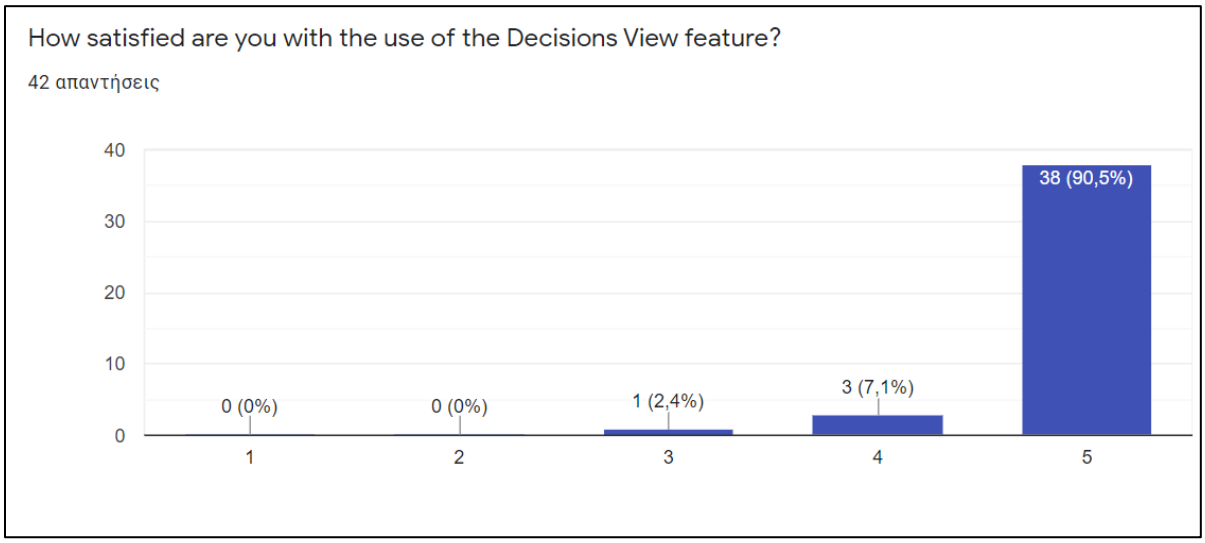


Figure 8.15 User's satisfaction with the preview of a Certificate

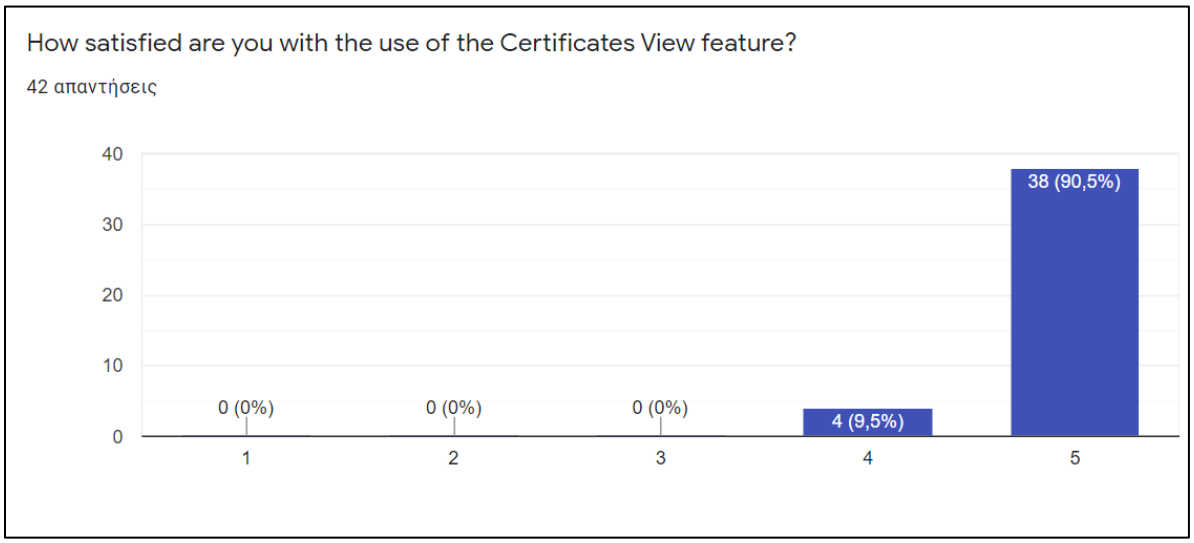


Figure 8.16 User's satisfaction with Employee's view

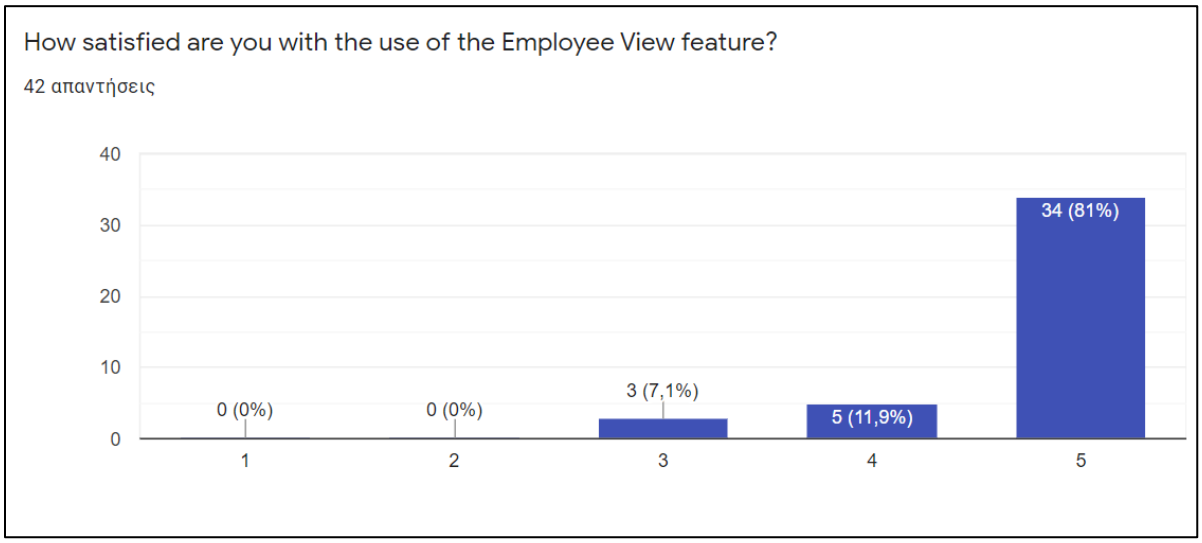


Figure 8.17 User's satisfaction with Applications' view

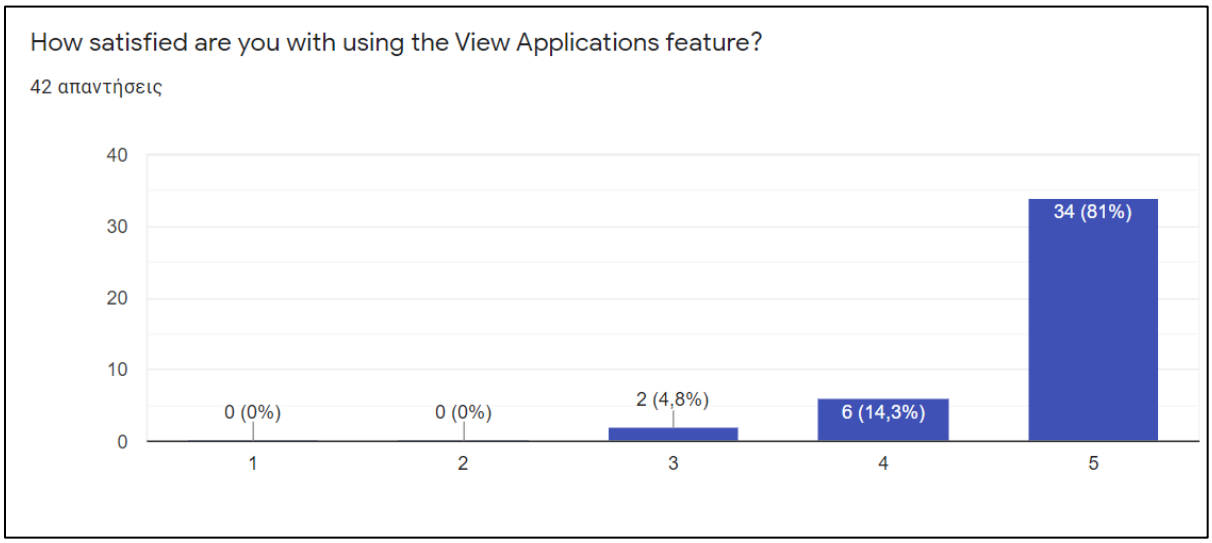


Figure 8.18 User's satisfaction with Certificate's view

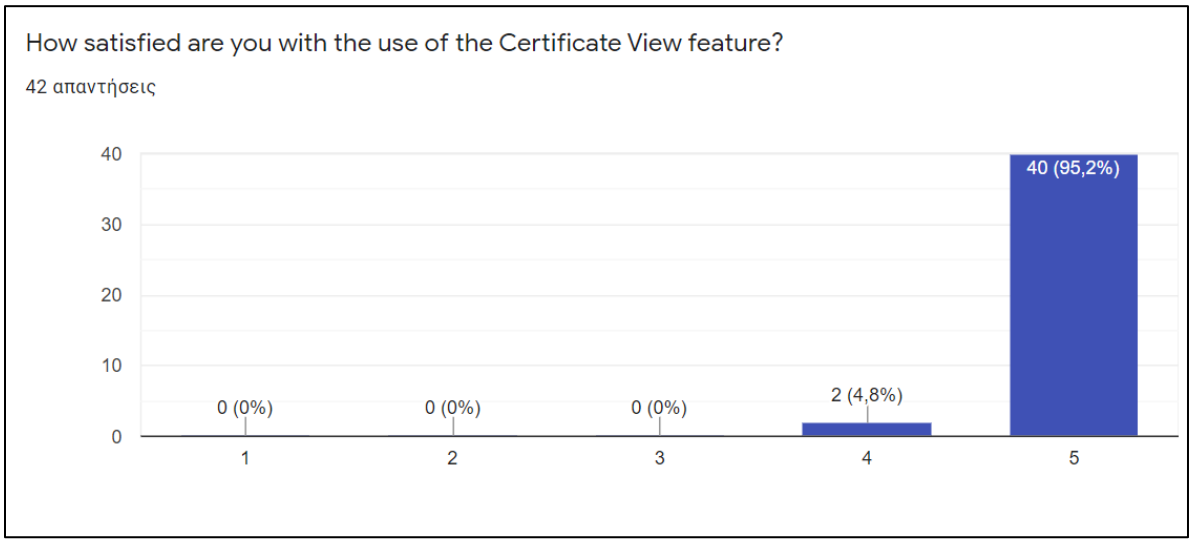
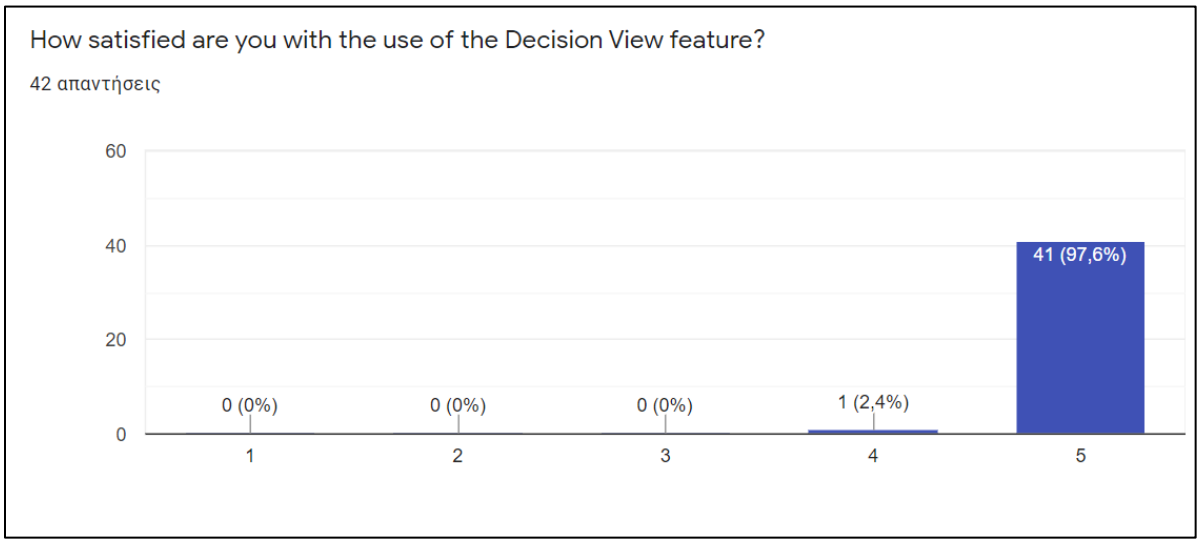


Figure 8.19 User's satisfaction with Decision's view



Section 4: Overall Evaluation of the Web-Portal

In the overall rating of the Web-Portal, the users stated very satisfied at 81% and satisfied at 16.7% while there was a 2.4% that stated neutrally. There were no negative answers. In terms of ease of use, the users stated very satisfied at 76,2% and satisfied at 23,8%. The processing speed was quite satisfactory, and the users stated very satisfied at 95,2%, satisfied at 2,4%, and neutral at 2,4%. Finally, it was very important that all users easily understood the functions of the Web Portal, completed all their requests, and did not encounter any technical difficulties.

Figure 8.20 User's satisfaction with the Web-Portal

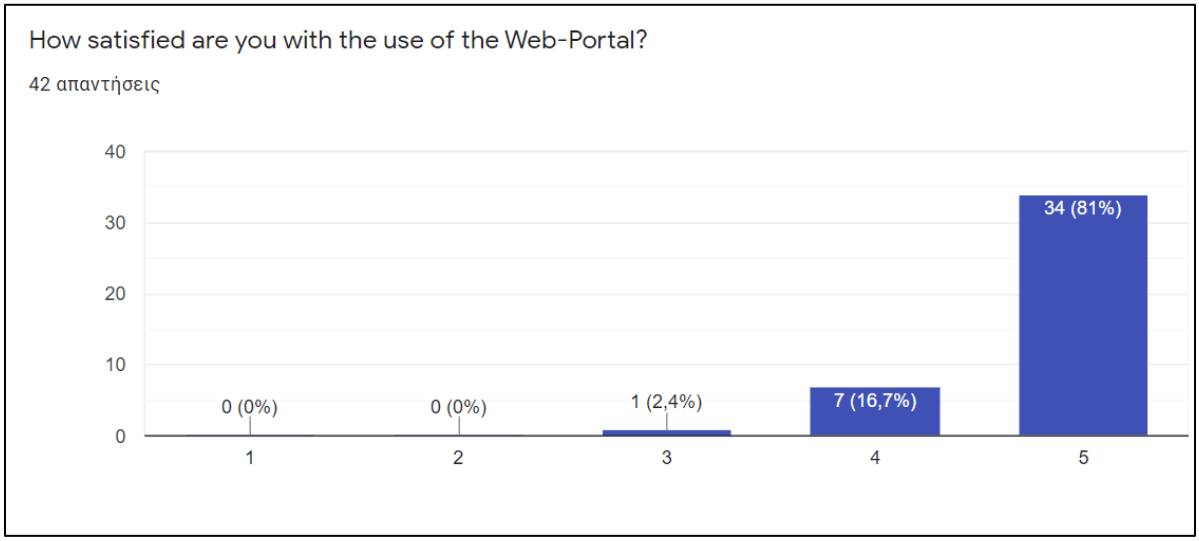


Figure 8.21 User's satisfaction with the ease of use

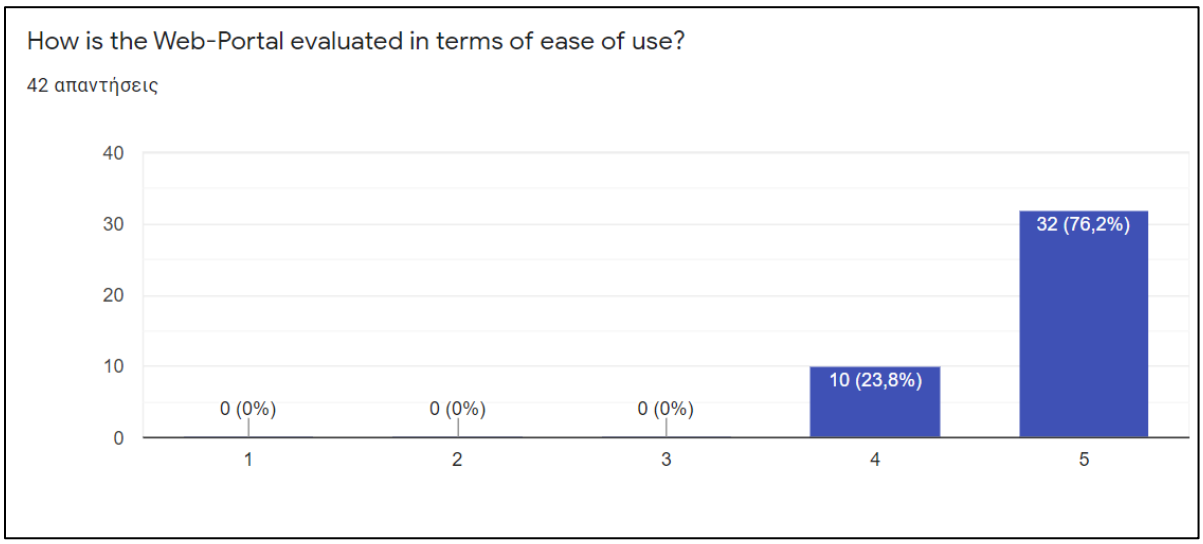


Figure 8.22 User's satisfaction with speed of processing

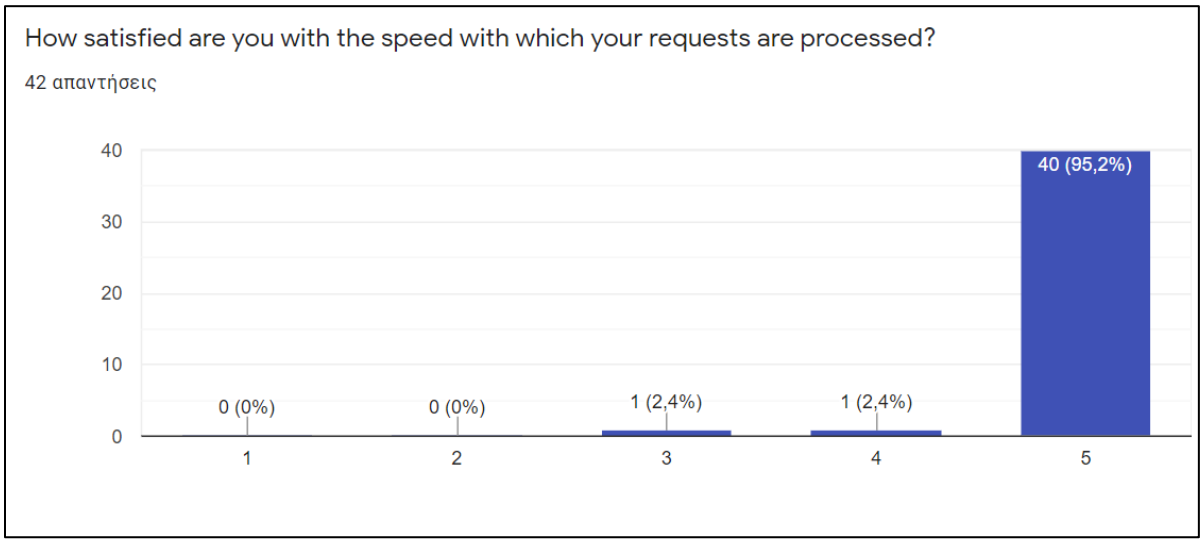
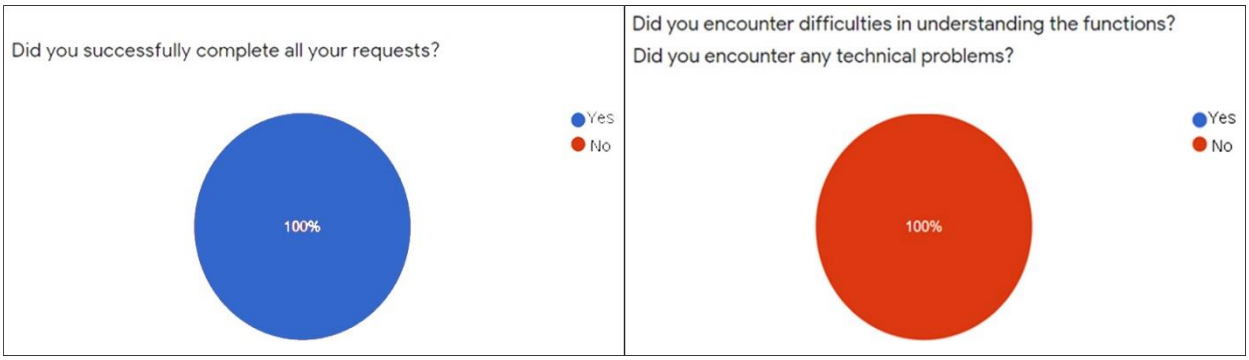


Figure 8.23 Difficulties, Technical Problems, Completion of requests



8.2.3 Conclusions of the evaluation

The evaluation of the Web Portal was positive. It is noteworthy that there were no negative reviews at all (Very Dissatisfied or Not Satisfied). Users were satisfied with the overall operation of the platform and showed ease of adaption by easily understanding and performing the functions. An important role in the increased satisfaction of the users was played by the fact that they did not face any technical problems and there was no delay in processing their requests.

Users stated high satisfaction with features that required action, such as a certificate application, a leave application, and a leave approval/rejection. Regarding the data display functions, high satisfaction rates were recorded for the viewing of individual documents, such as the Decision view and the Certificate view. Regarding the aggregate data views, and in general the functions of the Personal Repository and Administration's Dashboard, the reviews were positive but with a lower rating from the users.

The evaluation of the Web-Portal, although not directly evaluating the Ontology, helps us to draw useful conclusions about it. The fact that all the requests were completed successfully, within normal time and with no technical problems, is a sign of good operation of the Ontology. Also, the functions that received lower ratings, received these scores due to the presentation of the functions on the Web-Portal. In general, the cooperation between Web-Portal and SPARQL Endpoint was effective and the services offered to their users left them satisfied.

Chapter 9 Conclusions

9.1 Future Work

Our work is an initial approach to the development of vertical ontologies, which is open to a variety of extensions. Several issues can be explored at a later stage. First of all, there are plenty of administrative tasks that can be incorporated. Also, an issue that needs to be explored is the interconnection with other systems. The utilization of the third-party certification and specifically through the API of the General Secretariat of Information Systems will increase interoperability capabilities. Another issue on which there is already relevant research and which will expand the capabilities of the ontology is the incorporation of document templates.

9.2 Conclusion

In recent decades, progress in the field of information technology and communications has given a huge boost to the evolution of society, giving it an additional dimension, the digital dimension. In this cycle of change, governments are called upon to meet the needs of society and to transfer their activities to the digital world, through e-government. In this context, many government organizations have established online presence points offering online services, to citizens, businesses, and other government bodies. However, the services they offer extend mainly to stages 1 and 2 and less to stages 3 and 4. Also, few organizations have been involved in providing online services to their employees.

Various solutions have been adopted for the development of online services. A very useful and effective tool for the development of such services is the Semantic Ontologies. Semantic Ontologies offer better modeling of the structure of the entities, services, and data involved in the processes. Moreover, the use of ontologies has the advantage of inference ability, through reasoning.

At the same time, governments around the world, recognizing the need for e-government, have supported several initiatives and research in this area. These efforts have helped to layout general frameworks. However, this task is quite complex and encounters many difficulties, as the

scope of the public sector is huge. As a result, governments find it difficult to adopt horizontal ontologies that will be the backbone for the development of vertical ontologies.

The solution we propose to address the above issues is based on the development of vertical ontologies to serve the specific needs of each organization. The approach that we propose for the development of e-government systems certainly raises issues of heterogeneity. However, given the difficulty in implementing strict frameworks for e-government, this is a more general problem that does not occur only in our case. Also, the modeling offered by the use of ontologies helps us to overcome the issues of heterogeneity more easily.

In order to investigate the effectiveness of using ontologies to offer stages 1-4 services to employees, we developed an ontology and a Web-Portal that support the administrative procedures of an organization. The ontology was designed from scratch, using a bottom-up approach, and was customized for the special needs of the organization. The Web-Portal undertook to provide the services of stages 1-4, providing the necessary interfaces to the users and processing the necessary communication with the SPARQL Endpoint.

From the developer's point of view, the process of developing the services using the ontology was flexible and functional. Accessing the SPARQL Endpoint was a simple process that did not require complex queries. The evaluation of the users, after the use of the Web-Portal, was positive. Users used all four stages of services and completed successfully their requests without encountering any technical problems. Implementing services using Semantic Ontologies was not something that was perceived by the users. The cooperation of the Web-Portal with the SPARQL Endpoint was effective in meeting the needs of the users both for the employees and for the administration.

Our approach provides an adequate solution for the implementation of integrated e-government services, which can be effectively applied, apart from government to employee, and to the rest sectors of e-government.

References

[1] T. R. Gruber, “A translation approach to portable ontology specifica-tions,” in *Knowledge Acquisition*, 1993, pp. 199–220.

[2] N. Guarino, Ed., “Formal ontology in information systems,” in *Proceedings of the first international conference (FOIS’98), June 6-8, Trento, Italy (Vol. 46)*, 1998.

[3] A. Gómez-Pérez and R. Benjamins, “Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods,” *Proc. 16th Int. Jt. Conf. Artif. Intell. Work. KRR5 Ontol. Probl. Methods Lesson Learn. Futur. Trends | 16th Int. Jt. Conf. Artif. Intell. (IJCAI’99*, vol. 18, pp. 1–15, 1999.

[4] Mohsen A. Khalil; Bruno D. Lanvin; Vivek Chaudhry, “The E-Government Handbook for Developing Countries,” *World Bank/Centre for Democracy Technology*, no. November. 2002.

[5] Jeffrey W. Seifert, “A Primer on E-Government: Sectors, Stages, Opportunities, and Challenges of Online Governance,” 2001.

[6] M. Alshehri and S. Drew, “E-Government Fundamentals Author E-GOVERNMENT FUNDAMENTALS,” 2018.

[7] K. Hinkelmann, B. Thönssen, and D. Wolff, “Ontologies for E-government,” in *Theory and Applications of Ontology: Computer Applications*, no. September, Dordrecht: Springer Netherlands, 2010, pp. 429–462.

[8] J. Fraser *et al.*, “Knowledge management applied to e-government services: The use of an ontology,” *Lect. Notes Artif. Intell. (Subseries Lect. Notes Comput. Sci.*, vol. 2645, pp. 116–126, 2003, doi: 10.1007/3-540-44836-5_13.

[9] D. Apostolou, L. Stojanovic, T. P. Lobo, J. C. Miró, and A. Papadakis, “Configuring e-government services using ontologies,” *IFIP Adv. Inf. Commun. Technol.*, vol. 189, pp. 141–155, 2005, doi: 10.1007/0-387-29773-1_10.

[10] L. Stojanovic, N. Stojanovic, and D. Apostolou, “Change management in e-government: OntoGov case study,” *Electron. Gov.*, vol. 3, no. 1, pp. 74–92, 2006, doi: 10.1504/EG.2006.008493.

[11] E. N. Loukis, “An ontology for G2G collaboration in public policy making, implementation and evaluation,” *Artif. Intell. Law*, vol. 15, no. 1, pp. 19–48, 2007, doi: 10.1007/s10506-007-9041-5.

[12] V. Peristeras and K. Tarabanis, “Governance Enterprise Architecture (GEA): Domain models for e-governance,” *ACM Int. Conf. Proceeding Ser.*, vol. 60, pp. 471–479, 2004, doi: 10.1145/1052220.1052280.

[13] S. K. Goudos, V. Peristeras, N. Loutas, and K. Tarabanis, “A public administration domain ontology for semantic discovery of eGovernment services,” *2007 2nd Int. Conf. Digit. Inf. Manag. ICDIM*, vol. 1, no. November, pp. 260–265, 2007, doi: 10.1109/ICDIM.2007.4444233.

[14] W. Xia, T. Vitvar, V. Peristeras, A. Mocan, S. K. Goudos, and K. Tarabanis, “WSMO-PA: Formal specification of public administration service model on semantic web service ontology,” *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, 2007, doi: 10.1109/HICSS.2007.618.

[15] J. Shehu and E. Xhina, “Modeling an ontology for public E-Government Services in Albania,” 2019, doi: 10.33422/ret.2019.09.504.

[16] P. Salhofer, B. Stadlhofer, and G. Tretter, “Ontology driven e-government,” *Polit. Democr. E-Government Particip. Serv. Deliv.*, vol. 7, no. 4, pp. 383–401, 2010, doi: 10.4018/978-1-61520-933-0.ch023.

[17] A. A. Alshehab, A., Naser, N. A., & Hawaf, “Semantic Integration Sharing for E-Government Domains Ontology : Design and,” vol. 97, no. 6, pp. 1820–1831, 2019.

[18] A. Kumar and A. Joshi, “IndiGov-O: An ontology of Indian government to empower digital governance,” *India Int. Conf. Inf. Process. IICIP 2016 - Proc.*, pp. 4–9, 2017, doi: 10.1109/IICIP.2016.7975373.

[19] A. Gómez-Pérez, F. Ortiz-Rodríguez, and B. Villazón-Terrazas, “Legal ontologies for the Spanish e-Government,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4177 LNAI, pp. 301–310, 2006, doi: 10.1007/11881216_32.

[20] B. Molnár, A. Béleczki, and A. Benczúr, “Application of legal ontologies based approaches for procedural side of public administration: A case study in Hungary,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9831 LNCS, pp. 135–149, 2016, doi: 10.1007/978-3-319-44159-7_10.

[21] I. Savvas and N. Bassiliades, “A process-oriented ontology-based knowledge management system for facilitating operational procedures in public administration,” *Proc. Int. Work. Res. Issues Data Eng. - Distrib. Object Manag. -RIDE-DOM*, vol. 36, no. 3, pp. 77–84, 2001, doi: 10.1016/j.eswa.2008.05.022.

[22] M. M. Hasan, Di. Anagnostopoulos, G. Kousiouris, T. Stamati, P. Loucopoulos, and M. Nikolaidou, “An Ontology based framework for e-Government regulatory requirements compliance,” *Int. J. E-Services Mob. Appl.*, vol. 11, no. 2, pp. 22–42, 2019, doi: 10.4018/IJESMA.2019040102.

[23] G. Brusa, M. L. Caliusco, and O. Chiotti, “Building ontology in public administration: A case study,” *CEUR Workshop Proc.*, vol. 226, no. January, pp. 16–30, 2006.

[24] J. F. Muñoz-Soro, G. EstebAn, O. Corcho, and F. Serón, “PPROC, an ontology for transparency in’public procurement,” *Semant. Web*, vol. 7, no. 3, pp. 295–309, 2016, doi: 10.3233/SW-150195.

[25] R. P. Lourenço, S. Jorge, and H. Rolas, “Towards a transparency ontology in the context of open government,” *Electron. Gov.*, vol. 12, no. 4, pp. 375–394, 2016, doi: 10.1504/EG.2016.080440.

[26] L. Ding *et al.*, “TWC LOGD: A portal for linked open government data ecosystems,” *J. Web Semant.*, vol. 9, no. 3, pp. 325–333, 2011, doi: 10.1016/j.websem.2011.06.002.

[27] A. Gugliotta, L. Cabral, J. Domingue, V. Roberto, M. Rowlatt, and R. Davies, “A semantic web service-based architecture for the in- teroperability of e-government services Conference Item A Semantic Web Service-based Architecture for the Interoperability of E-government Services,” *Event London*, pp. 21–30, 2005.

[28] Y. Xiao, M. Xiao, and H. Zhao, “An ontology for e-government knowledge modeling and interoperability,” *2007 Int. Conf. Wirel. Commun. Netw. Mob. Comput. WiCOM 2007*, pp. 3605–3608, 2007, doi: 10.1109/WICOM.2007.891.

[29] B. Kanagwa, J. Nakatumba-Nabende, R. Mugwanya, E. Kigozi Kahiigi, and S. Ngabirano, “Towards an Interoperability e-Government Framework for Uganda,” no. August 2018, Springer International Publishing, 2018, pp. 16–28.

[30] M. Asfand-E-Yar and R. Ali, “Semantic integration of heterogeneous databases of same domain using ontology,” *IEEE Access*, vol. 8, pp. 77903–77919, 2020, doi: 10.1109/ACCESS.2020.2988685.

[31] M. R. Chbihi Louhdi and H. Behja, “Ontology learning from relational databases: Transforming recursive relationships to OWL2 components,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 10, pp. 265–270, 2019, doi: 10.14569/ijacsa.2019.0101037.

[32] M. A. G. Hazber, R. Li, X. Gu, and G. Xu, “Integration mapping rules: Transforming relational database to semantic web ontology,” *Appl. Math. Inf. Sci.*, vol. 10, no. 3, pp. 881–901, 2016, doi: 10.18576/amis/100307.

[33] K. Munir and M. Sheraz Anjum, “The use of ontologies for effective knowledge modelling and information retrieval,” *Appl. Comput. Informatics*, vol. 14, no. 2, pp. 116–126, 2018, doi: 10.1016/j.aci.2017.07.003.

[34] H. Tissot, C. A. G. Huve, L. M. Peres, and M. D. Del Fabro, “Exploring logical and hierarchical information to map relational databases into ontologies,” *Int. J. Metadata, Semant. Ontol.*, vol. 13, no. 3, pp. 191–208, 2019, doi: 10.1504/IJMSO.2019.099834.

[35] A. Doniger, “Position Paper – W3C Workshop on Semantic Web in the Energy Industries ; Part I : Oil & Gas,” *W3C Work. Semant. Web Oil Gas Ind.*, vol. 1, pp. 1–4, 2008.

[36] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, “Pellet: A practical OWL-DL reasoner,” *Web Semant.*, vol. 5, no. 2, pp. 51–53, 2007, doi: 10.1016/j.websem.2007.03.004.

[37] M. Poveda-Villalón, A. Gómez-Pérez, and M. C. Suárez-Figueroa, “OOPS! (OntOlogy Pitfall Scanner!): supporting ontology evaluation on-line,” *Int. J. Semant. Web Inf. Syst.*, vol. 10, no. 2, pp. 7–34, 2014.