

Faculty of Engineering of the University of Porto
Master on Informatics and Computing Engineering



**Integrating FEUP's Print Service and Information System at
CICA**

Curricular Internship Report 2007/2008

Jessica Martínez Carrillo

FEUP Supervisor: Prof. José Manuel de Magalhães Cruz
CICA Supervisor: Engenheiro António Francelino Gomes Viana

February 2008

I dedicate this to my family and friends.

Resumo

Descreve-se o processo de desenvolvimento de uma nova funcionalidade dos serviços de impressão da FEUP, efectuado no âmbito do programa de estágios do Mestrado Integrado em Engenharia Informática e Computação, na Faculdade de Engenharia da Universidade do Porto.

O trabalho foi desenvolvido no Centro de Informática Prof. Correia de Araújo (CICA) e teve como objectivo dar aos utilizadores do sistema informático da FEUP a possibilidade de trocaram a senha global de acesso ao sistema informático em qualquer uma das impressoras Xerox multifunção administradas pelo CICA. As principais motivações para o trabalho foram a redução do número de solicitações de trocas de senha ao serviço de atendimento ao público do CICA (HelpDesk) e a possibilidade de se prescindir de um computador para se efectuar tais trocas. Pretendia-se, também, adquirir melhores conhecimentos técnicos sobre a tecnologia Extensible Interface Platform (EIP) da Xerox.

O projecto foi desenvolvido com base em tecnologias genéricas, HTML, JavaScript, Perl e na recente tecnologia EIP, específica à Xerox.

O trabalho foi muito bem sucedido, todos os objetivos propostos foram atingidos e o sistema está em fase de testes de campo, prevendo-se que venha a ser disponibilizado aos utilizadores da FEUP muito em breve.

Abstract

It is described the process of the development of a new functionality of the FEUP's printing services, made under the program internship of the Integrated Master in Informatics and Computation Engineering, at the Faculty of Engineering of Porto's University.

The work was developed at the Computing Centre "Professor Correia de Araújo" (CICA) and the target was to provide the computer system's users in FEUP, the possibility of changing their global access password to the computer system in any of the Xerox multifunction printers managed by CICA. The main motivations for the project were to reduce the number of requests for password changing, submitted to the HelpDesk service of CICA and making those changes possible without a normal computer. Also, it was intended to acquire better know-how on the Extensible Interface Platform (EIP) of Xerox.

The project was developed on base generic technologies, HTML, JavaScript, Perl and on the most recent EIP technology, specific of Xerox.

All the proposed goals were met and the system is in the field testing phase; it is expected to be available to the FEUP's users shortly.

Acknowledgements

I thank my parents for always being with me specially now, that we are apart and for always backing me up on everything I want to do in life.

I thank my sister for all her support and comprehension.

I thank my aunt Carmen for all her help.

I thank Nico for always giving me a smile.

I thank my friends for giving me all their support.

I thank my new friends in Oporto, for all the good moments lived and for making me feel at ease and keeping me company so that I never felt lonely.

I thank Engineer António Francelino Gomes Viana for all his help and availability for me.

I thank Professor José Manuel Magalhães Cruz for all his help and support with the memory.

I thank Professor António Augusto de Sousa for all his help with the Erasmus application and also with the choosing of the Project.

I thank all my colleagues at CICA for all their help.

I thank Lina Silva for all her help with the application and her care.

I thank Noémia Margarida Pereira Taboada for all her help in the Portuguese course offered by FEUP.

I thank my Xerox contact for all his help as without it, everything would have been different.

Thanks to all the people at the internship institution that was directly involved on the work made or contributed to its development.

Thanks to all the FEUP people that somehow supported the work success or contributed to its development.

Table of Contents

1 Introduction.....1

 1.1 CICA introduction 1

 1.2 The Project 2

 1.3 Study and prototype development 3

 1.4 Organization of the report..... 3

2 Analysis of the problem4

 2.1 The printing system at FEUP during 2000-2005 4

 2.1.1 Problems identified in 2005.....4

 2.1.2 Objectives for 20064

 2.1.3 Some numbers during 2000-20054

 2.1.4 Technical architecture during 2000-2005.....5

 2.2 Current objectives5

 2.3 Project 5

 2.4 System benefits..... 6

 2.5 Current Technical Architecture and Infrastructure of FEUP's printing services..... 6

 2.6 Work plan7

 2.6.1 Tasks explanation7

3 Technological revision.....9

 3.1 The SiFEUP 9

 3.1.1 Provision of Access.....9

 3.1.2 The SiFEUP's team.....9

 3.1.3 Resources Available 10

 3.2 The linserv..... 11

 3.3 The password change via SMS 11

 3.4 Xerox EIP Technology..... 11

 3.4.1 Benefit for the end user 13

 3.4.2 System integrator / Information Technology advantages 13

 3.4.3 Examples where EIP can be used..... 13

 3.4.4 Study forum EIP Xerox..... 14

 3.5 Base technologies for the project 15

4 Specification17

 4.1 Requirements..... 17

 4.1.1 External factors17

 4.1.2 Internal factors 17

 4.1.3 User interface and human factors 17

 4.2 Description of operation 18

 4.3 Use Cases 18

 4.3.1 Use Case diagram of the system 19

 4.3.2 Design of external interfaces.....20

 4.4 ChangePassSystem states diagram..... 22

5 Implementation.....23

 5.1 Changes regarding the Specification 23

5.2	Some details	23
5.3	Additional applications	23
5.4	Client Registration.....	23
5.5	Things needed	25
5.6	Application.....	25
5.7	Files description	26
5.8	Problems encountered	28
5.9	Function's description	29
5.10	Tests made	33
6	Conclusion.....	35
6.1	Personal conclusions	35
	Bibliography.....	36
	Glossary	38
ANNEX A:	How the linserv program works	40
ANNEX B:	Description of the SDK	42
ANNEX C:	Description of other tests	46

1 Introduction

The project was developed at FEUP (Faculty of Engineering University Porto) more specifically, at its computers centre, CICA. The project started on February 20 and finished on July 7, 2008. The main goal was to develop an application, in which the user can change his/her system password, without using the CICA's assistance. Thus, the user can change the password using the printers which are placed throughout the faculty; he/she only needs their own security PIN¹. The security PIN is not the same as the PIN code, that is to say, the user has it in order to have more security when printing. The PIN code is like using a magnetic card.

The main phases of the project were: to study the EIP² (Extensible Interface Platform) Technology from XEROX, to study the integration of the printing services at FEUP with its information system, and to develop a service prototype which would allow FEUP users to change their passwords of the global computing system directly from multifunction printing devices.

In this way, the user would have yet another way to change passwords. He/she could do it before by going to the HelpDesk, by web or by SMS (Short Message Service).



Figure 1: CICA logotype.

1.1 CICA introduction

The Prof. Correia de Araújo Computer Centre (CICA) is the entity responsible for the planning, development, implementation, management and maintenance of computer resources and services and of communication networks to the FEUP community, by promoting its use and innovation.

The users are allowed in computer rooms from 8:00 to 20:00 and a HelpDesk is open from 9:00h until 18:00h and the students helping service, in the computer areas, is from 8:00h until 20:00h. In case of having a problem, a user can report it by email address is: helpdesk@fe.up.pt. The main computer resources managed by CICA (e.g. Internet access) are available 24 hours per day.

Eng. Tito S. Vieira is the chief executive officer at CICA, divided in five units. The units are:

1. **Information Systems Unit (USINF)**, headed by engineer Susana Santos Gaio. **USINF** is in charge of planning, development, implementation and maintenance of information systems.
2. **Infrastructures and Communication Networks Unit (UIRC)**, headed by engineer Jorge Rocha. **UIRC** is in charge of planning, implementation, management and maintenance of communications network's services and infrastructures.

¹ Security PIN is the secret number sequence that allows the user to use the printing services.

² Xerox EIP is a software platform upon which developers can use standard web-based tools to create server-based applications that can be configured for the Multifunction Printer's touch-screen user interface.

- 3. Micro informatics and User's Support Unit (UMSU)**, headed by engineer António Viana. **UMSU** is in charge of planning, implementation and management of users support services and computer rooms. In this unit, the HelpDesk is responsible to inform, advise and solve computer user's problems. This service may be requested by any person registered in the SiFEUP (Information System of Faculty of Engineering of University Porto).

To use this service you could either:

- Go in person to the ground floor of building D;
- Write an e-mail message;
- Use the telephone;
- Use the trouble ticket system validation.

- 4. Quality and Security (UQS)**, headed by engineer Tito Carlos S. Vieira. **UQS** has a number of tasks from which a selection is mentioned:

Regarding quality: standardizing technical procedures and putting them into practice; evaluating the services provided; promoting initiatives and procedures towards the systematically adoption of a quality politics and its corresponding monitoring in all fields.

Regarding security: planning the computing security of CICA by defining the strategy, the procedures and suitable practice for the various areas; defining and spreading good computer security measures on the various areas in the academic community; helping all units on the elaboration of contingency plans and defining a global contingency plan;

- 5. Systems Administration Unit (UAS)**, headed by engineer Carmen Lima. **UAS** is in charge of the planning, implementation, management and maintenance of the computing services and infrastructures.

1.2 The Project

The project consisted on the development a web application, through which a user could modify his/her global information system password (Unix/Linux system, SiFEUP information system, Windows systems, TCPGate³ firewall system, e-learning systems and MySQL database manage), using his/her security PIN, from most of the campus printers, managed by CICA. A user would have two options to modify the password: the first one would be choosing himself/herself a password and the second one would be to get an automatically generated password. Either way, the password could be printed, if the user so wished. The application is linked to a program. It will be seen in another chapter in what this program consists of. This program is already created and is linked to the database.

In short, the project consists of adding this functionality to a multifunction printer.

³ TCPGate is a service that allows access to FEUPnet from outside, using an access via modem, ISDN (Euro-ISDN) or via Internet.

1.3 Study and prototype development

My work part consisted on developing the aforementioned application, interfacing it to the already existing program. I had to work with this program developer, in order to find out how to interface to it, so that the database could be accessed and modified.

1.4 Organization of the report

The report is organized as follows:

Chapter 1: contains information about CICA, a brief project explanation and a little introduction; **Chapter 2:** contains information about the planning of the necessary work and a detailed analysis of the problem; **Chapter 3:** contains information about the current solutions, the necessary technology and the language used to develop the application; **Chapter 4:** contains detailed information about the solution, including its specification; **Chapter 5:** contains information about the implementation, problems found, and necessary differences from the specification; **Chapter 6:** presents the final conclusions of the project; **Glossary:** contains a table with acronyms and a table with an explanation of the project terms; **Annex A:** contains information about how the linserv program works; **Annex B:** contains Software Development Kit (SDK⁴) detailed information; **Annex C:** contains other tests that have been made during the project.

⁴ SDK is typically a set of development tools that allows a software engineer to create applications for a certain software package, software framework, hardware platform, computer system, operating system, or similar platform.

2 Analysis of the problem

In this chapter, the overall analysis of the problem will be presented.

2.1 The printing system situation at FEUP during 2000-2005

In the period from 2000 to 2005, the users had access to the printers installed in the computer rooms in a way controlled by the printing quota system.

There were 40 computer rooms with a printer inside in the room. The concept of direct printing was, in other words, that only one user could print in that room and at a time. A user could only send his/her printing job to a specific room. Each job had printed identification printed header sheet that the user did not have to pay for. Users received an initial printing quota of 250 pages, and could be extended by giving payment to the HelpDesk, of CICA.

In SiFEUP their balance and the logs of impressions could be checked.

2.1.1 Problems identified in 2005

The problems that came across in 2005 were: very high exploration cost; printing with low-quality; volume Printing: 15% of printing sheets where header sheets with identification (supported by CICA), 80% of the work had less than 7 pages, the 5% were non collected printed jobs; the number of pages were not counted correctly; to print on a different printer, you had to move to another room; printing could only be done inside the rooms.

2.1.2 Objectives for 2006

The following objectives were defined in 2006: replacing the hardware; keeping with SiFEUP integration, self-service system, concept of printing directly into the classroom; introducing a new concept "Follow-You Printing"⁵: it consists of being able to print from any printer inside the campus, any time and day of the week; having service in campus, including Wireless and VPN; having: color printing, scanning documents and copy service; implementing monitoring system; having a simplified management; having controlled expenses.

2.1.3 Some numbers during 2000-2005

The equipments, number of users, etc. during the period of 2000-2005 are shown as follows: 40 computer rooms (+ the 700 PC's); 40 printers (one per room); 3200 PC's connected to FEUPNet; 4791 wireless users; 8091 users; 4 million printers during 2005.

The FEUP studied several solutions and chose Xerox with the Equitrac software⁶.

There exists a monitoring consisting of: pro-active system about the printer's alert status; centralized logs system; system blocks the equipment automatically when in trouble.

⁵ Follow-You Print: the work is sent to a printing queue at the server; it can be released at any equipment.

⁶ Equitrac is software that provides document cost management (auditing, allocation, recovery) and output management solutions for printers, copiers and multifunction devices.

In the Printer Status it is possible to see its details, and the system notifies when a printer does not work correctly. Furthermore, it informs of the documents printing time, the users, etc. UMSU Control Centre informs about the printers that do not work correctly.

2.1.4 Technical architecture during 2000-2005

Fig 2 shows the architecture of the printing system of FEUP during 2000-2005. VLAN⁷ for printing aggregated 60 printers. VLAN was connected to the printing server (PrintServer) that at the same time had the printing database. The print server had two Network Interface Card (NIC⁸). For security reasons, a firewall was used. The users could have access to the FEUPNet and through its FEUPSig, the users could send jobs to be printed.

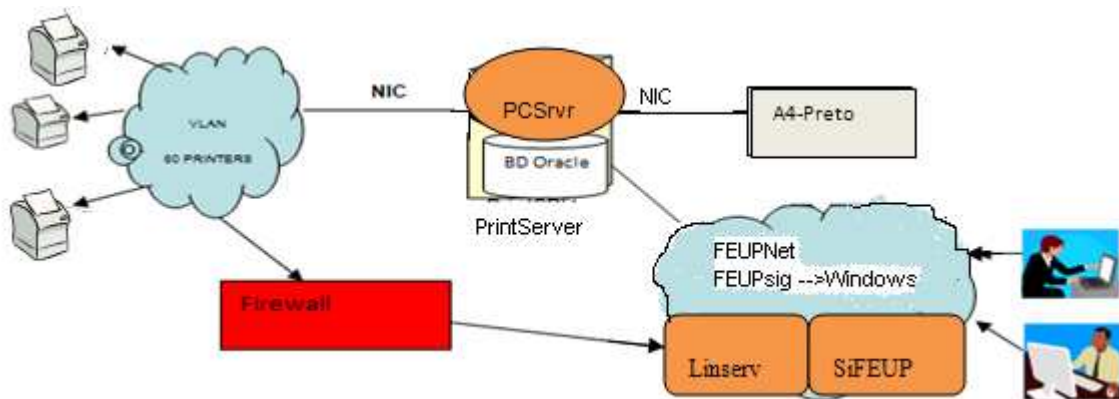


Figure 2: Architecture of the printing system of FEUP during 2000-2005.

2.2 Current objectives

The following are the objectives expected to be achieved with this project: to reduce the number of service requests to HelpDesk, for password changing; changing the password can be done at any Xerox printer; changing the password can be done without using a computer; changing the password can be done at any time and day of the week; to know the EIP Xerox technology.

2.3 Project

This project provides:

1. Access for all SiFEUP users enabling them to change their global information system password from any printer on campus. It may be made through the printer graphical interface. Users will be able to change their password from the printers at any time and day of the week. The purpose is that the users stop using the HelpDesk service within CICA, largely because HelpDesk tasks are meant to changing password only;

⁷ VLAN has the same attributes as a physical LAN, but allows end stations to be grouped together even if they are not located on the same LAN segment.

⁸ NIC is a piece of computer hardware designed to allow computers to communicate over a computer network.

2. The user will be identified by the printer when using his/her card or when entering the security PIN. In this way, printers can only be used for changing password by SiFEUP pre-registered users;
3. The application will communicate with “linserv” in order to change password for all the informatics services and resources.

2.4 System benefits

The main benefits expected for the CICA with this new functionality are:

1. Expense savings on HelpDesk personnel, as the changing passwords tasks are performed automatically;
2. Providing a secure channel to be able to change passwords in a printer, without using a computer;
3. Allowing users to be able to change passwords at any time, day and place at FEUP.

2.5 Current Technical Architecture and infrastructure of FEUP's printing services

Fig 3 shows the architecture and infrastructure of FEUP printing services. The current technical architecture is different from the previous one, as it includes 4 drivers: Geral_A4, Geral_A3, Geral_Cor_A4 and Geral_Cor_A3. Now it uses the Equitrac program, which runs on the printing server, the linserv program and an internal firewall. The firewall controls the communications according to the network policy that the company has defined. It protects the internal network from any attempt of unauthorized access from the Internet.

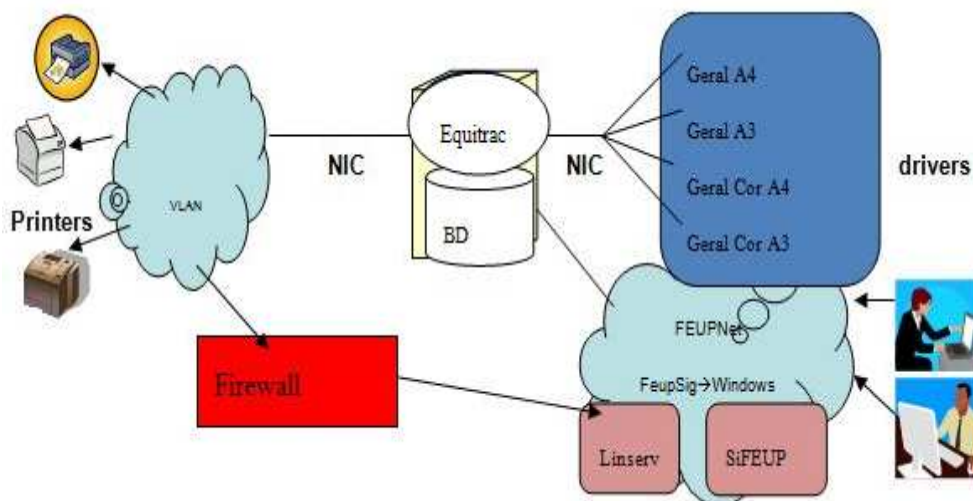


Figure 3: Architecture and infrastructure of FEUP's printing services.

Equitrac Office provides end users with a secure, convenient and mobile print workflow while controlling costs and simplifying administration of the printing system. Equitrac Office enables single sign-on access to devices and services, personal print queues to maximize document security and mobility, and rules to create cost-effective printing behaviour.

The key features are: track how printers and copiers are being used and by whom; analyze which devices are being over-and under-used and recognize peak usage periods; allocate expenses by department, cost centre, project, or client.

The Extensible Interface Platform Xerox-capable devices enable these Equitrac Office features: secure, mobile printing, that allows users to release their print jobs securely at any EIP-enabled device; no additional hardware is needed, because user ID code and billing entries are made directly on the Xerox user interface; monitor, measure, and manage printing costs to enforce quotas and other color print / copy rules as well as to track and report on usage costs by individual, department or other user grouping - all by leveraging the same user ID data utilized to access the Xerox device.

2.6 Work plan

The table in Fig. 4 shows the tasks planning that were proposed for the development of the project. It is the work plan that I have followed.

	Task Name	Duration	Start	Finish	Predecessors
1	Analysis of CICA structure and organization	3 days	Wed 20/02/08	Fri 22/02/08	
2	Analysis of FEUP's printing service infra-structure	2 days	Mon 25/02/08	Tue 26/02/08	1
3	Analysis of FEUP information system services	2 days	Wed 27/02/08	Thu 28/02/08	2
4	Study Xerox EIP Technology	10 days	Fri 29/02/08	Thu 13/03/08	3
5	Pre-specification	4 days	Fri 14/03/08	Wed 19/03/08	4
6	Mount server, add applications to a printer, test printer	4 days	Thu 20/03/08	Tue 25/03/08	5
7	Pre-Implementation	7 days	Wed 26/03/08	Thu 03/04/08	6
8	Pre-Tests	8 days	Wed 26/03/08	Fri 04/04/08	6
9	Programming template	1 day	Fri 04/04/08	Fri 04/04/08	7
10	Wiki	65 days	Fri 04/04/08	Thu 03/07/08	7
11	Specification	2 days	Mon 07/04/08	Tue 08/04/08	8
12	Implementation	35 days	Wed 09/04/08	Tue 27/05/08	11
13	Tests	35 days	Wed 09/04/08	Tue 27/05/08	11
14	User's manual	4 days	Wed 28/05/08	Mon 02/06/08	13
15	Personal conclusions	2 days	Tue 03/06/08	Wed 04/06/08	14
16	Summary in Portuguese and English	5 days	Thu 05/06/08	Wed 11/06/08	15
17	Write the memory	20 days	Wed 28/05/08	Tue 24/06/08	13

Figure 4: Final work plan.

2.6.1 Tasks explanation

In the following paragraphs each of the proposed tasks will be explained.

Analysis of CICA structure and organization: the structure and organization of the CICA building had to be analyzed. The services offered, the floors and the departments structure, had to be analysed too; **Analysis of FEUP's printing service infra-structure:** the university printers infra-structure had to be studied and a technical architecture sketch was to be made; **Analysis FEUP information system services:** the following services had to be analysed: SiFEUP, linserv and password change via SMS; **Study Xerox EIP Technology:** the EIP forum, the SDK and the Xerox services, had to be studied; **Pre-specification and Specification:** had to be done twice as the objectives have changed along the development (the original objectives were not very clear and other ideas emerged to change the password); **Mount server, add applications to a printer, test printer:** the Apache server had to be installed in my computer, a network board card (NIC) had to be added to allow connection to the printers network, printer applications had to be added in the printers in order to test them and begin to work with the printer; **Pre-Implementation and Implementation:** had to be

done twice as the specification changed. The application has been implemented; **Pre-tests and Tests**: had to be done twice as the implementation had to be changed. The developed code had to be tested on the printer; **Programming template**: had to be made to always put the code forward the same way; **Wiki**: a docuWiki has been created, in order to add information about the Project, minutes, templates, reports and interesting things about the project; **User's Manual**: a user's manual had to be created; **Personal conclusions**: the conclusions of the project had to be written; **Summary in Portuguese and English**: a summary of the Project had to be written in the two languages.

The work proceeded in close range to the proposed plan as the effective timeline of the project shows (Fig. 5).

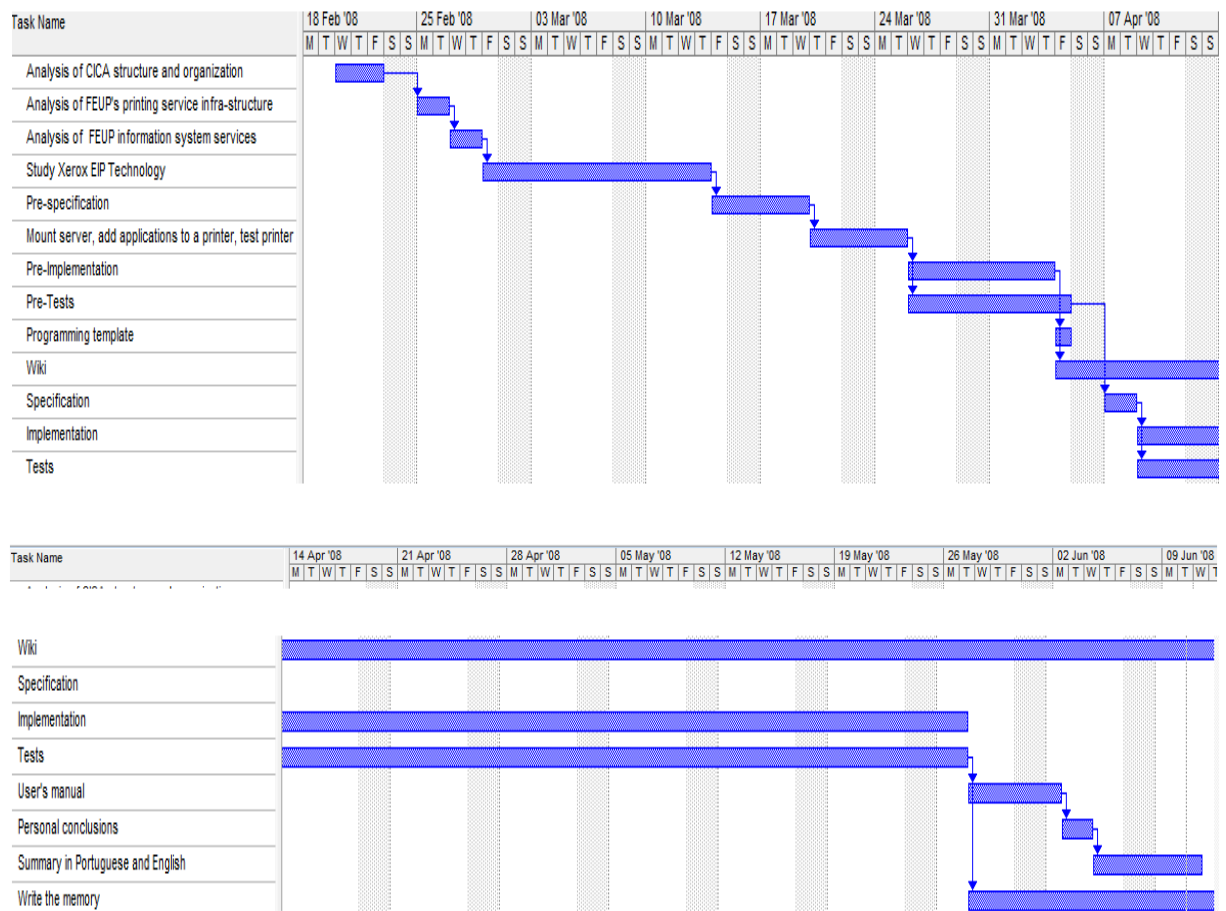


Figure 5: Gantt diagram of executed tasks.

3 Technological Revision

The current solutions to change the system password will now be analysed, as well as the carried out solution in this project.

3.1 The SiFEUP

The Information System of FEUP (SiFEUP) is both a service and an infrastructure. The key service objectives are to inform and facilitate the access to the Faculty resources, at the same time as increasing team work cooperation amongst FEUP's members.

The SiFEUP allows access to all the information concerning the Faculty, including the academic information of the students, as well as providing enrolment services, such as application for certificates and classes. Documentation in support of various courses, information on the location of buildings and rooms, the contact information for teachers and students, communication facilities, are also available through the system.

The document "Base Development of SiFEUP" – [16] clarifies the general objectives and operation of the system, characterizes the information area and describes its architecture and development methodology.

All elements of the FEUP academic community, including students from year 1 to doctoral students, besides having a webpage, visible on the Internet, can also create their own information resources in SiFEUP.

3.1.1 Provision of Access

Upon enrolment at FEUP, all students have automatically created accounts on several computing environments: (UNIX / Linux machines, Windows machines, SiFEUP, TCPGate firewall to local network usage from the Internet, etc). Their passwords are delivered at the time of registration. In the case of students requiring special accounts, they can go to the CICA's HelpDesk.

These facilities remain valid throughout their academic path at FEUP.

Since September 2004 the existence of an institutional email address covering all users was established. The e-mail address cannot be changed in SiFEUP and all the messages concerning life at FEUP can be sent to it.

3.1.2 The SiFEUP's team

The SiFEUP's team has the task of developing the infrastructure that allows the publishing of data in the system. The publication comes from several sources: the academic information, from the Academic Unit; the news, from the Image and Communication Unit; the funded projects and the courses information, from directors offices, etc. A SiFEUP's team provides technical support and advice necessary to the correct use of the system.

3.1.3 Resources Available

Fig. 6 shows a table with the main information areas covered by SiFEUP.

Area	Resource	Description
General	FEUP	Introduction, history and organization of FEUP
	News	General and local news
	Legislation	Laws, statutes, minutes
Programmes	Study plans	Programmes description
	Description	Programmes official Information
	Web pages	Support to specific resources in the web
	Educational service	Educational service functions
	Timetables	Teachers, classes and halls timetable
	Training classes	Training classes enrollment
	Accreditation	Reports of accreditation and external evaluation
	Pedagogical Inquest	Pedagogical inquests' results
	Continuous education	Programmes description
Students	Official web page	Personal institutionally managed data
	Student card	Classified registry
	Personal web page	Personally managed generic data
	Certificates	On-line certificates orders
	Statistics	Academic results' statistics
	Printing quotas	Accounting of printing services used
	Payments	Payments through ATM
Staff	Official web page	Personal institutionally managed data
	Personal web page	Personally managed generic data
I&D	Projects	Description, participants, results
	Scientific articles	Bibliography, abstracts
Other	Buildings floors	Buildings floors and departments
	Halls	Halls' description
	Computing resources	Hardware and software availability, maintenance
	Resources reservations	Halls, equipments and general reservations
	Trouble Tickets	Support services management to the users
	Dynamic e-mail	Dynamic distribution Lists
	Forums	Debate areas
	Searching	General searching

Figure 6: Main information areas covered by SiFEUP.

SiFEUP advantages: the users have access to information 24 hours a day.

Inconveniences: if the user does not remember the password he/she would not be able to use the system and will have to go to the HelpDesk. Furthermore, if the user does not remember the password and wants to use the faculty computers would not be able to do it.

3.2 The Linserv

Linserv is the main tool in the HelpDesk service. It is a tool for managing all the following accounts, from the overall faculty informatics system:

- **Windows**
- **Unix/Linux**
- **SiFEUP**
- **TcpGate**

Linserv can be accessed through the web address: <https://umsucc.fe.up.pt/Portal>, by special authorised personnel. Linserv is connected to the database and is implemented in Perl.

Advantages: it is a fast and easy to use program. One can quickly see whether there has been any error when changing passwords in any of the accounts.

Inconveniences: In order to use this program, you have to use the HelpDesk service, since CICA workers are the only ones allowed to manage the users' accounts. The password can only be changed this way on CICA opening hours.

Annex A presents a description of the linserv operation.

3.3 The password change via SMS

The "Password Recovery System" is a tool that allows the users to have access, simply and quickly, to recover / amend the same way as done via FEUP's computer systems, using SMS technology. Firstly, the user has to fill in a form with the mobile number; the user receives a SMS with a verification code afterwards. The user has to type in the verification code. If it is correct, a question and an answer has to be written. When the user decides to use the password recovery system, he/she has to type in the answer written in the form and if it is correct.

All users can have access to the system using the SIFEUP validation service.

Advantages: it is not necessary to use the CICA service.

Inconveniences: if the user loses the password and did not have the form data filled in, he/she would not be able to use the service nor use their account. They will have to go to CICA's HelpDesk.

3.4 Xerox EIP Technology

Xerox announced the Extensible Platform Interface in October 2006. Xerox was not the first provider of a software platform Multifunction printer (MFP), but profited from the experience

of the pioneers. But, Xerox has been the first MFP supplier to adopt and implement the "Web 2.0"⁹ trend and a related technology, Service-Oriented Architecture (SOA¹⁰).

Xerox's EIP is used with Xerox multifunction printers. With EIP, those printers can adapt their way of working to the environment where they are used. End users can store and print information easily and securely. Developers can quickly create applications that are easily customized to the touch screen user's interface of the devices.

A Multifunction printer (**MFP**) is directly connected to a network and offers several functions, such as printing, scanning, copying and faxing. They are toner-based machines, either color or monochrome.

Printing involves receiving a stream of computer data and instructions on how to represent the data on paper. Scanning is generated by the creation of new data in the form of images. Users can scan a document on an MFP and store images inside of the device, or place them in a folder somewhere on the network. Those images can also be sent directly to an email address.

There are currently two main alternatives for the creation of software for MFP: embedded Java and an approach to Web services/web browser.

Java is a powerful and widely accessible programming environment and there are many Java programmers. One or more Java applications can be developed to reside within an MFP. Java is easier to use than traditional programming languages. Because a Java application must be installed within each MFP where it is used, this approach can create administrative overhead for the management and updating of applications.

The alternative is to use Web Services that take advantage of international standards of the Internet as HTML, XML, WSDL, SOAP and AJAX to create applications, to develop the user interface screens and administrate this software.

One possible disadvantage of a network based approach is that if the network shuts down, the application cannot be accessed. Of course, if the network shuts down, a Java application in an MFP is not in a position to send images over the network, and the printer will not be available for printing, just for copying.

Another problem based on the network is that if a large number of MFPs interact with an application on a server, that network could create a bottleneck. However, in most applications, two or more servers may be deployed to adequately serve the expected demands. In addition, the image compression capability helps reducing network traffic.

The software on the server that hosts Web pages is then used to put forward information to the user on the MFP Control Panel. The data that the user through the control panel goes communicate to the server and, at the same time, can be used on the communication with other applications.

It is easier to develop screens and intuitive control panel with a browser. As the applications are not within the MFP, this reduces the need for processing power or memory on the device and but increases network dependence.

⁹ Web 2.0 is a term describing the trend in the use of World Wide Web technology and web design that aims to enhance creativity, information sharing, and, most notably, collaboration among users.

¹⁰ SOA is a computer system's architectural style for creating and using business processes, packaged as services, throughout their lifecycle.

It is more practical to have multiple applications running on a server rather than having them all placed in MFPs. And, most important, the software is easier to deploy, manage and update.

From the two alternatives, I chose to use web services, because since the beginning it was agreed to program with HTML and JavaScript. Anyway, the option to choose Java is not possible as we cannot add any additional software to the multifunction printers. I also observed in the Xerox forum that it was recommended to use web services. Having tested the web services, I could see that the answering time were good: the queries invoked directly at the printer (display user and obtain a security PIN) were answered and showed at the touch screen soon enough. Xerox supplies JavaScript for those web services that must be called within the EIP Web browser. Xerox only use/support Visual Studio and C# because that is what the majority of their partners use.

3.4.1 Benefits for the end user

- Simplifies complicated workflow by facilitating the use of multifunction printer, making the overall work easier;
- Transforms paper documents into digital information, making it easier to edit, store and share;
- Adjusts the printer to the user needs, and not vice versa;
- Completes certain tasks directly on the multifunction printer, including the reception of documents from a network without the need of a PC.

3.4.2 System integrator / Information Technology advantages

- Adds solutions to an existing infrastructure;
- Manages centralized solutions from anywhere in the world;
- Multifunction printer adapts well to businesses;
- Creates customized solutions easily. EIP is based on Web standards like HTML, CSS, XML and JavaScript. It also uses standard secure protocols: HTTPS and SSL;

3.4.3 Examples where EIP can be used

Some examples of where EIP can be used: use menus and specific terms of the company or working groups as "Search customers database", "Send forms to the complaints department" or "Send by fax to suppliers"; all personal preferences can be displayed on the touch screen system multifunction by passing one's card; converts a complex workflow into a simple process that requires only clicking a few buttons; digitizes and stores in a repository documents currently on paper just by pressing a button;

In Figure 7 it is show the necessary structure for using an EIP application in a web server: the web applications are in the application server and the user interacts with the application through the tactile screen. In Figure 8 a block and layers picture of the general architecture

standards web such as HTML, CSS, XML, JavaScript and other secure protocols such as HTTPS and SSL can be used.

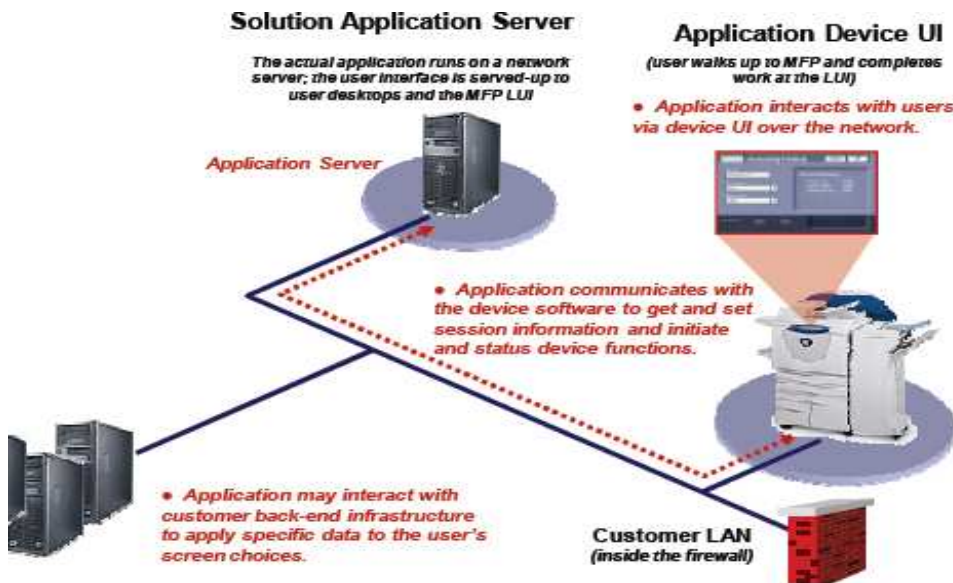


Figure 7: Typical EIP Application System.

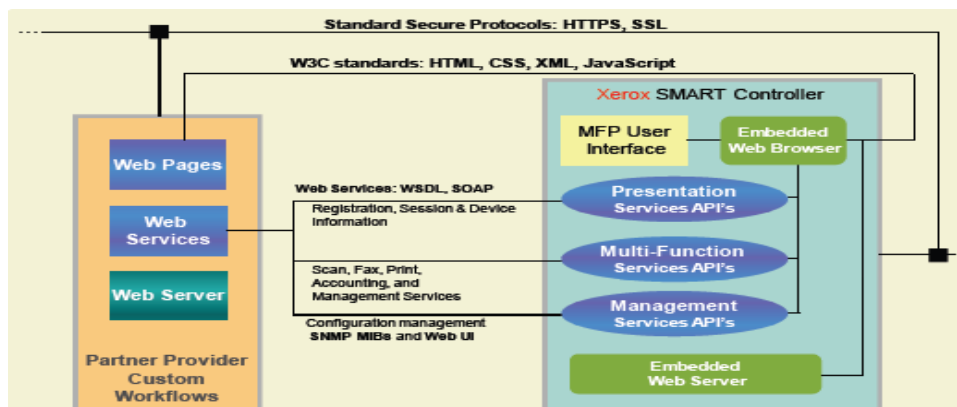


Figure 8: EIP System Overview

3.4.4 Study forum EIP Xerox

By reading and participating in the Xerox EIP forum, a number of practical and useful information were acquired.

The developers of Xerox EIP use:

- **Integrated Development Environment:** NetBeans 6, MS Visual studio 2005 (under Vista or XP), Text editor, Eclipse with Zend PDT (PHP);
- **Languages:** Java/Groovy, C#, VB, PHP, PHP 5 & Zend Framework, VB.NET, C++;
- **Web server platform:** Tomcat/Apache, ISS;

Some interesting problems to the user on to the developer found in the mentioned forum:

Problem with the buttons register in the main screen (it shows: print, email, etc): If we have two registered services and the buttons are superimposed.

Solution:

- 1) Press Log In/Out to log in/out
- 2) Machine Status
- 3) Tools
- 4) System Settings > Common Service Settings > Screen/Button Settings
- 5) All Services
- 6) Make sure you have a "Custom Service" for each registered service.

Problem: how to print a document that has been scanned from MFP.

Solution: The first way is to build a LPR command line string and launching a shell command to run it. Something like `lpr -S192.168.1.10 -Plp printfilename.pdf`. The other way is to create a TCP/IP socket to port 9100 of the MFP and put the binary data from the file through this socket.

Problem: Displaying pdf-files in the embedded browser.

Solution: There is no built-in way to display PDFs in the browser. You would need to render the PDF into one of the supported image formats.

Problem: Add a button which calls a keyboard to unfold it and the user enters some text.

Solution:

```
<div> lime pit text <input name="text1" id="text1" type="text" class="xrx: keyboardtext">
</div>
showKeyBoard ();

<link href="xrx_widgets/XRXWidgets.css" type="text/css" rel="stylesheet"/>
<script src="xrx_widgets/XRXWidgets.js" language="JavaScript"></script>
<script src="KeyboardInput.js" language="JavaScript"></script>

xrx_load_widgets();
```

Conclusion: Additional technologies to solve the core problem of the project need not be studied because the multifunction Xerox printers only support this technology, which is developed to be used together with its own printers.

3.5 Base-Technologies for the project

The base-technologies necessary for the project are the following:

JavaScript: it is an interpreted programming language (that is to say, it does not require compilation) used mainly in web pages, with a similar syntax to that of the Java language and C language.

Perl: it is an interpreted language that takes features from C, from the interpreted shell (sh), AWK, Sed, Lisp and, to a less extent, from lots of other programming languages. It is very much used in system administration scripts.

HTML (HyperText Markup Language), it is the base language of web pages. It is used to describe the structure and contents of web pages, which can include text, images and multimedia objects. HTML can also describe, to a certain extent, the appearance of a document and can include *scripts* (for example, JavaScripts), which can affect the web navigators behaviour and other HTML processors.

The problem of an interpreted language is that you could have left something in the code, for example, declared a variable when testing the code it does not work. On the other hand, if the syntax is correct, it is faster, because it is not necessary to compile the code. In the case of Perl, it is a language that can be compiled or interpreted afterwards.

An advantage that offers HTML and JavaScript is the independence of the platform to be used: they are not bounded to any specific hardware platform or operative system. I could not choose the programming languages as I have been assigned them from the first moment. I am using Perl as it was the development language for the linserv program.

In order to program with HTML and JavaScript, I have used the WordPad program and for the Perl program, I have used Emacs. I could have used the Emacs editor for all the html files but at the beginning I started programming with Windows.

In order to add applications to the printers, an Apache server has been installed on my computer.

It is very important to add the folder `XRXWebServices`¹¹ to the correct directory, because some applications do not work well, because the applications need some of the functionalities in the folder itself. The `XRXWebServices` contains widgets. Widget is a little application or program, usually displayed in small files which are executed by a Widget Engine.

The server election has been free; I have chosen the Apache server and have used it on Windows, because the linserv server uses Apache. Furthermore, it is a robust and easy server to use.

¹¹ `XRXWebServices` contains style sheets and necessary widgets.

4 Specification

In this chapter, detailed information about the solution is given, including its specification.

4.1 Requirements

The requirements have been classified according to the Meyer quality factors catalogue. The quality factors are the factors that determined the quality of a software product. There are two types of quality factors: the external and the internal.

4.1.1 External factors

External factors are those which are perceived by the users. The requirements for this project are:

- **Reliability:** the software performs the functions as they were defined in the requirements and in the specification;
- **Compatibility:** the software should be easily combined with other equipment;
- **Efficiency:** the software uses the minimal computing resources to operate correctly and agreeably;
- **Ease of use:** the software is understandable by people with different levels of experience, so that they can use it with a minimum of technical help.

4.1.2 Internal factors

Internal factors of interest to developers and administrators are the following:

- **Modularity:** the software is easily extendable and reusable;
- **Timeliness:** the software is usable upon prescribed delivery time;
- **Integrity:** the software operates correctly, without corrupting valid data;
- **Economic:** the software fulfils the goals with a minimum data of development and operational resources.

4.1.3 User Interface and human factors

For our application we will take into consideration only one type of user, the user registered in SiFEUP. It is not required that the user has advanced computer skills, who knows only how to deal with easiness in the management of Xerox printers. The application's main goal is to be easy to use and the most intuitive possible.

The errors that will be considered during the process are on the data the user enters to the application. In the event that this data is not valid, the system informs the user of the error. Other possible errors had been considered as well, like for example having problems in returning to the main menu.

The system must be able to detect errors and to display the type of error that occurs and, if the problem cannot be solved, the system will abort the action that was taking place.

4.2 Description of operation

The main operation consists of changing the user password of the following computer system and services: Unix/Linux, SiFEUP, Windows, TCPGate, e-learning and MySQL directly from a Xerox Multifunction printer. The user will have to go to a printer, from the touch screen select the icon Password, and follow the steps that are prompted by the application. For security, the user is asked to enter his/her security PIN; if the user does not have one, the operation cannot continue. In the future, there might be an option to change the password without the security PIN. After entering a security PIN, the user is informed by the system about its validity; in case of error the user has two more attempts, after which the operation is aborted by the system. The user can try again but will have to start from scratch. If the PIN validation is successful, the user could then choose to either type a new password or accept a system generated one. If the user chooses the option of automatic password, the system will show the user the accounts change status afterwards. On the contrary, if the user chooses to type in its own password, the system will inform him/her about its validity and about the accounts change status. To have the new password validated, the user has to write it twice, after which the system will check that the password is correct and inform the user about the success of the change. If one error is detected, the system will report it to the user, indicate the type of error and allow the user to type in another password and repeat the validation process. Finally, the system will take the user back to the main menu.

4.3 Use Cases

The application has a single use case, that of changing the system password:

Use Case: changePassSystem

Actors: subscribers of SiFEUP

Main Course:

- 1) The user goes to the printer.
- 2) Enters the pin code or passes his/her identity card.
- 3) Enters the security PIN or just clicks [Enter].
- 4) Chooses the option: "Change system password".
- 5) The login name will be displayed. The system will ask the user if he/she really wants to change the password. The user then has the option to answer yes or no.
- 6) The user clicks [Yes], and enters a security PIN at the system request.
- 7) The user clicks [Next].
- 8) The system validates the security PIN and informs whether the security PIN is correct.
- 9) The system displays the login name and two fields to enter the new password. The password must have a minimum of 6 characters and cannot contain symbols.

- 10) The user writes the new password twice and clicks [Next].
- 11) The system informs the user that the change has been carried out successfully.
- 12) Finish the use case.

Extensions:

No: user responds that after all, he/she does not want to change the password (5)

- 1) Finish the use case.

Exit: the user clicks [Exit] (7)

- 1) Finish the use case.

The security PIN is not correct (8):

- 1) The system will alert the user that the security PIN is incorrect. The user has another attempt.
- 2) The user writes an incorrect security PIN again, the system informs the error.
- 3) Finish the use case.

New password is incorrect (10):

- 1) The system reports either that the password must contain a minimum of six characters and symbols are not allowed or the new password and repeat password are not match.

Error: an error will occur when trying to change the password (10)

- 1) Finish the use case.

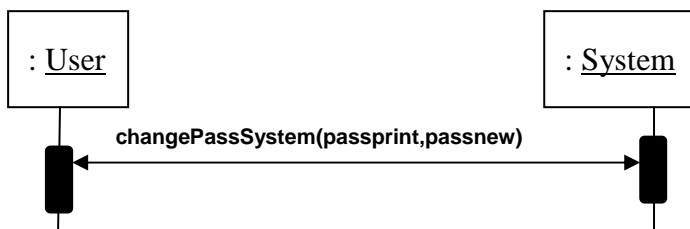


Figure 9: Operation sequence diagram: changePassSystem.

4.3.1 Use Case diagram of the system

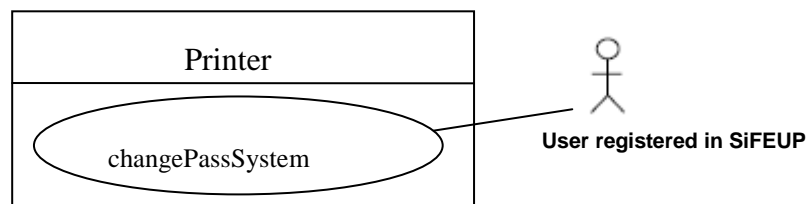


Figure 10: Use case diagram of the system.

4.3.2 Design of external interfaces

The whole process of password changing, from the point of view of the user is now shown, by means of pictures of the user interface on the touch-screen of a Multifunction printer. The Fig. 11 shows the main menu which contains the options: copy, e-mail, print and change password.

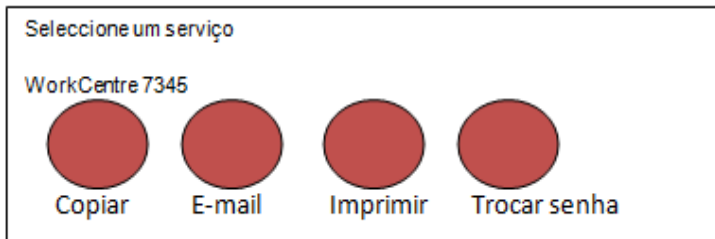


Figure 11: Current Xerox's MFP main view: selection of a service.

Change password of the system

In Fig. 12, the system asks the user if he/she wants to change the password to access to Unix, SiFEUP, TCPGate, e-learning and MySQL.

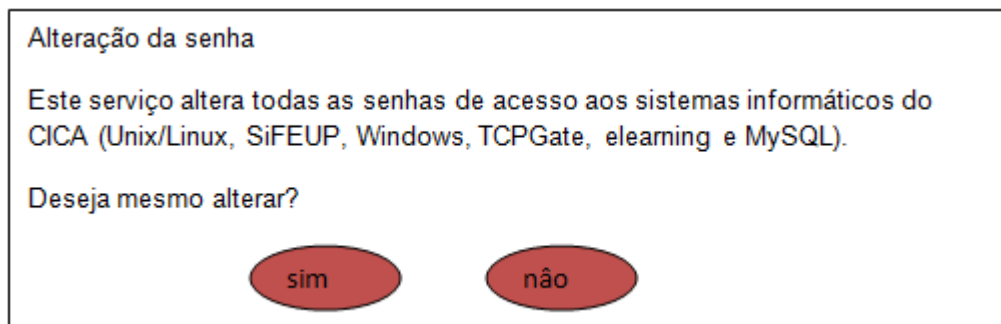


Figure 12: Change password of the system.

User clicks yes

In Fig. 13 it is shown the interface after the user has pressed the "Sim" button of Fig. 12 for security reasons the user has to introduce his/her security PIN.

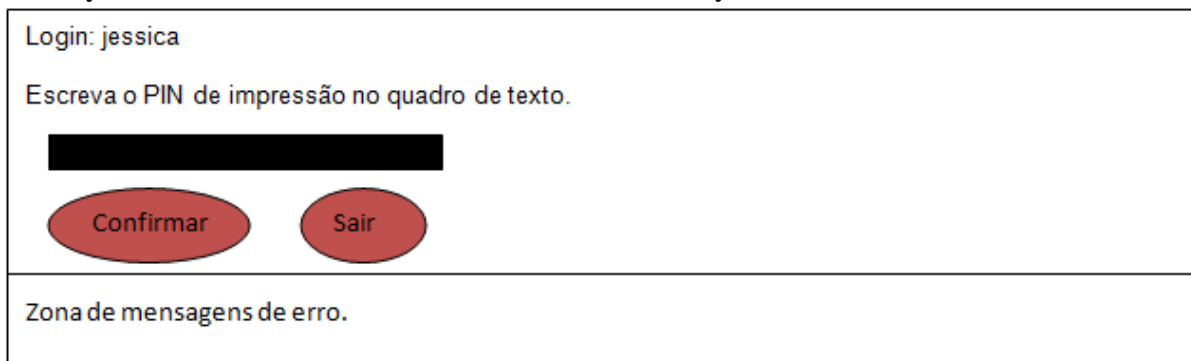


Figure 13: Type the security PIN.

Error messages area

In the following figure an error is shown, in which case the user has written his/her security

PIN incorrectly and can type in the security PIN again.

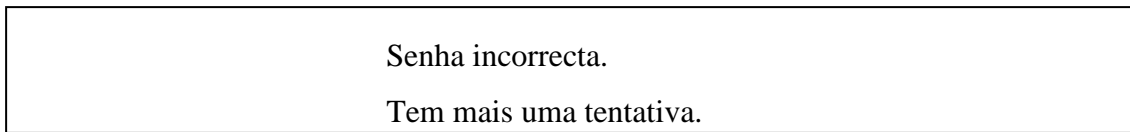


Figure 14: Error message.

Change password

In the following figure the interface is shown, in which the user has to type in his/her new password and type it in twice. The new password cannot have less than 6 characters and neither can have symbols.

Figure 15: Change the system password.

Error messages area

The following figure shows an error with the new password introduced.

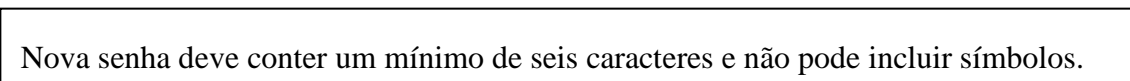


Figure 16: Error message.

Correct password

The following figure informs that the password change has been made correctly.



Figure 17: Correct password.

4.4 ChangePassSystem states diagram

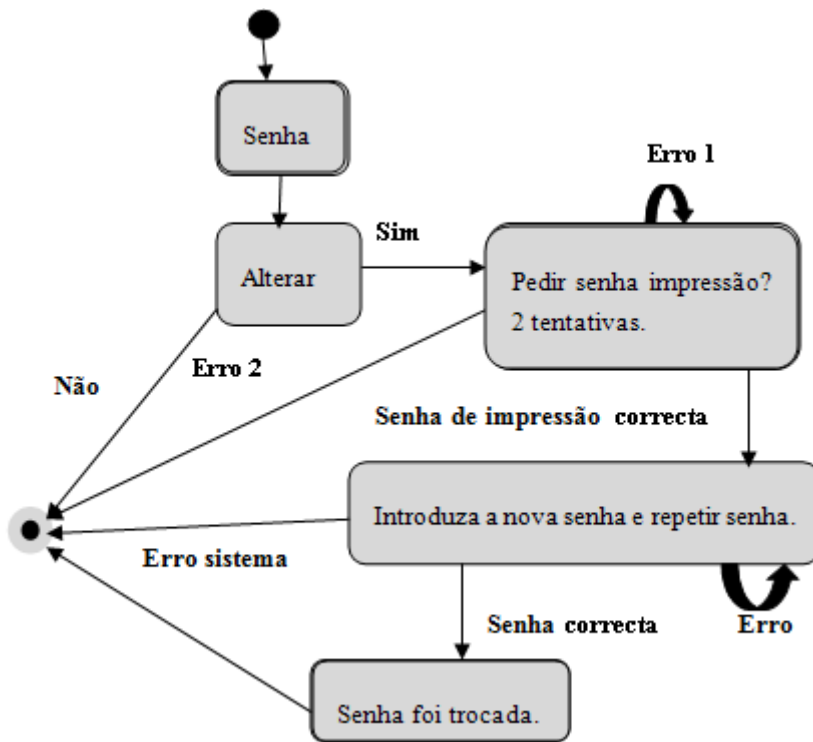


Figure 18: ChangePassSystem states diagram.

5 Implementation

In this chapter a more detailed description of the changing of passwords process is given. In chapter 3, the selection of the languages and integrated development environment (IDE) used during the project were justified. Also, the widgets used were provided by Xerox. They were in libraries that included web pages styles (.css) and other operations.

5.1 Changes regarding to the Specification

A new option has been introduced: generate a password automatically. This option generates a password automatically and the user can accept this new password or require the system to generate another one. Another change, was introduced: the user does not type his/her new password twice, as to type it again the user has to look very carefully in the keyboard to be able to write it properly.

Lastly, the new password does not get printed out, as there is no information as how to print a document from a multifunction printer. The Xerox contact informed me that this information will be available next year.

Anyway, we are interested in knowing how printing jobs work from any multifunction printer, not one in particular.

5.2 Some details

The Firefox browser works well with the system and better than the Internet Explorer. We can also use the Windows simulator browser that comes with the SDK. Xerox recommends for the development of the EIP pages, to use the Windows ANT navigator shown in the SDK or Firefox, it does not recommend using Internet Explorer for any page debug since the pages will not render service correctly.

We just need the SDK presentation services and printing services to carry out the password change on the information system.

In the SDK there is a sample Registration Client that shows how to add a button. The registration service has a method called "PutRegistration" that will place the button on the User Interface of machines that support EIP.

5.3 Additional applications

The sequence of operation should be: put the code on the web server and type in the registration; then add the url for these files. For the EIP application to work on a print server, the print server would have to be running a web server application.

5.4 Client registration

First we have to run the file: Xerox.EIPSDK.DEviceConfigApp. We add the printer IP and click connect. By clicking on GetDeviceInfo, information from the printer, serial number, name, model, etc. is displayed.

It is necessary for the developer to attach the client registration in the SDK. You need .NET 1.1 and WSE2.0sp3. The registration is necessary in order to add functions from any

computer. The registration client essentially creates a shortcut between the printer and our web application.

To make the client registration it is necessary to enter the printer IP and the password. Click "Connect". Then if we click on the "List", it displays the label name: Follow-You and the corresponding checksum.

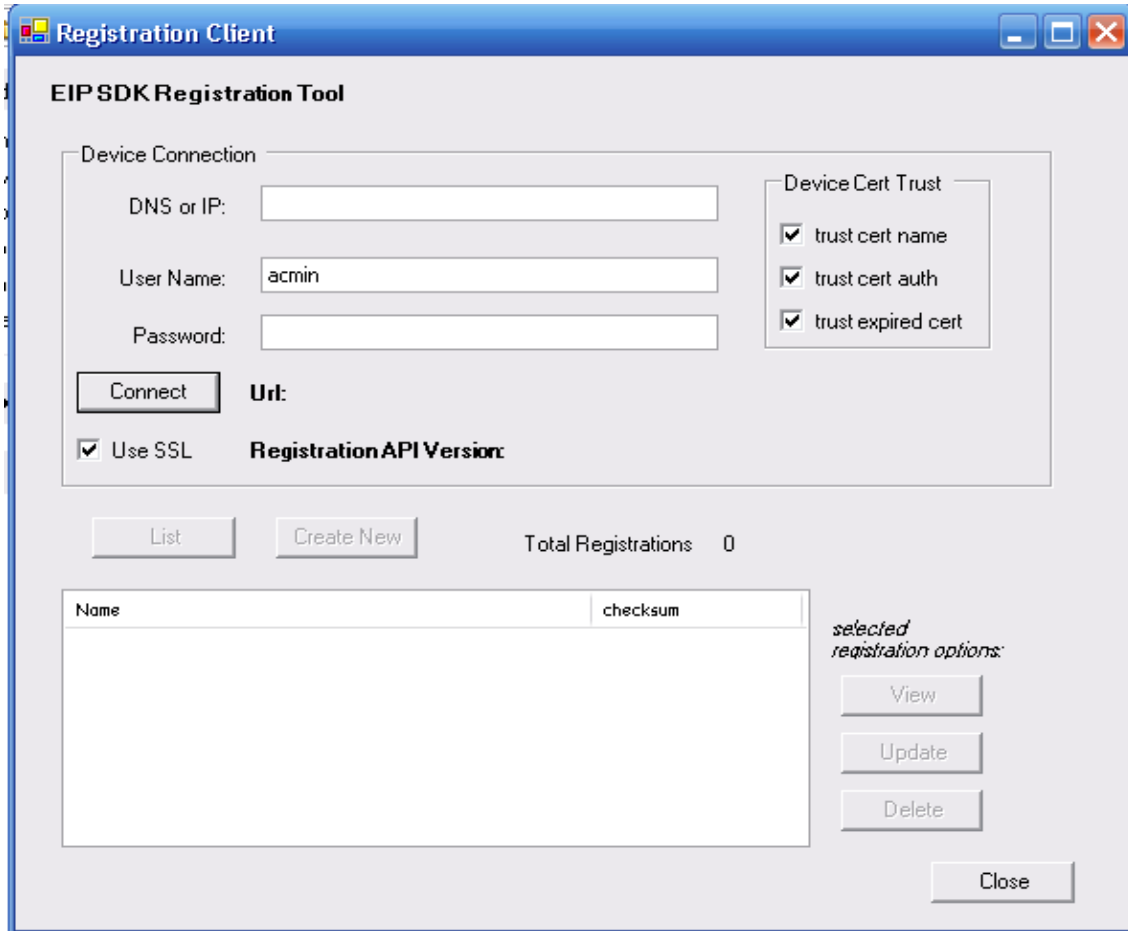


Figure 19: Client registration of EIP application.

To register a feature we have to click: "Create New", we have to fill in the following fields:

Registration Name: name of the registration;

ServiceURL: place where the application is;

DescriptionURL: is the place where we put the XML file with the name of the application. In this field we write the url. The file description:

```
<?xml version="1.0" encoding="utf-8" ?>
<displayText xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.xerox.com/webservices/office/dsp/xeip">
    <label>Trocar senha</label>
</displayText>
```

We write the name on the <label> of the application that appears in the printer.

Admin Description: is optional;

Small Icon URL: is optional, you can write the place where the icon is. Support icons with the format: gif, jpg and png.

Large Icon URL: you can leave this blank, it is not used;

Tools Icon URL: is the same, as for the Small Icon URL, these only shows:” web interface” on the printers;

Vendor: the creator;

Name	checksum
Follow-You	19871647
Change password	294171828
test	-978

Figure 20: Client registration with a new application (change password).

5.5 Things needed

We need to create a web server containing the web applications. And we also need to register the client on any computer to record the web applications. The client registration updates the user's touch screen interface on the machine that displays the web application output for the project application.

Not all Xerox MFP support the system:

- Xerox Workcentre 7245 (Color) - does not support EIP.
- Xerox Workcentre 73xx (Color) - supports EIP.
- Xerox Workcentre 128 - does not support EIP.

To do the implementation we will use the Xerox Workcentre 7328.

To send the HTMLs, the XML file and the XRXWebServices folder to the “linserv”, we used the FileZilla program.

We had to introduce the “linserv” IP, the user's name, password and the port. Afterwards, we dragged the files from the local address to the remote address.



Figure 21: FileZilla program.

5.6 Application

The application is structured as follows:

Cgi-print folder: containing all the applications, XML file and the XRXWebServices folder.

Applications: index, index1, index2, pass, index4, exit are the HTML.

Label is an XML file which has the name of the new printer option, in which case, it contains “Trocar senha”. The file description can be found in section 5.4.

Files in Perl are change.pl, status.pl, these files contain the access to the database to validate the password and to inform whether there is a problem with the new password.

5.7 Files description

Below, a description of the contents of all files used is given.

index.html: the system shows the user's login and enquires whether he/she wants to change the system passwords, the user then can reply yes or not.

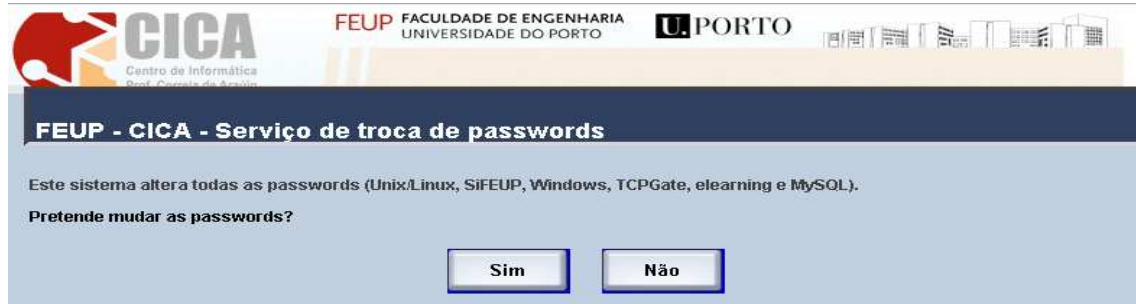


Figure 22: index.html screenshot.

index1.html: the system shows the user's login and he/she has to type in the security PIN in order to change the system password. The system checks whether the security PIN is correct or not.



Figure 23: index1.html screenshot.

index2.html: the system provides the user with three options: choosing a new password; using the system generated password automatically; leave an application.



Figure 24: index2.html screenshot.

pass.html: the system shows the user's login and asks the user to type in the new password.



Figure 25: pass.html screenshot.

index4.html: the system shows the user's login and the new password generated.



Figure 26: index4.html screenshot.

exit.html: the system informs the change is correct.



Figure 27: exit.html screenshot.

change.pl: the system checks whether the user and the new password are correct.



Figure 28: change.pl screenshot.

status.pl: the system shows the user's accounts change state.

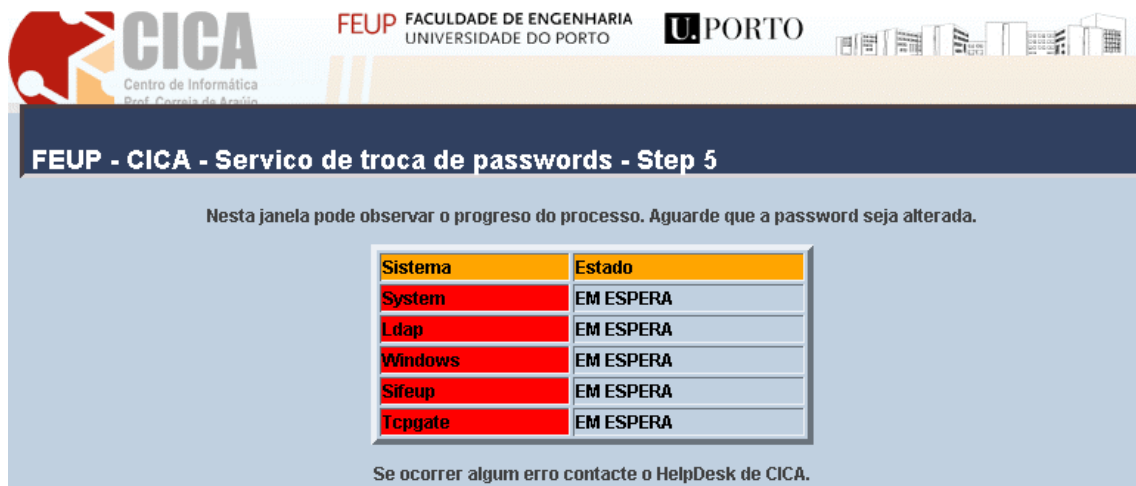


Figure 29: status.pl screenshot.

5.8 Problems encountered

1. The machines do not support proxy servers, yet this will be an issue only if the printer is directly connected to the Internet.
2. Loading a common web page would be fine but there are some issues:
 - There are no scroll bars on the printer's browser; so, if the page is too big, it will just be truncated;
 - There is no support for "plug-ins"; so, Flash, ActiveX, and similar pages that require a plug-in will not work. The ability to type in a text box will not work since you would need to load up the keyboard widget in order to access it;

When working a lot with Firefox, we have to take into account that it is necessary to clean the cache from time to time, as the application does not do what it is expected to do because of that. This problem can be solved cleaning the cache each time the HTML file is shut.

Another problem I came across with when sending the form data, was that there cannot be a form inside another form.

Another problem that I came across when I tested the applications was that I wasted a lot of time as the identification failed to work sometimes, so I could not test or work. Therefore we had to call a technician to solve the problem. The problem is when the user goes to the printer and it is necessary to type in his/her security PIN, sometimes does not work and the keyword

takes a long time to come out to be able to type in the PIN. The same happens when using cards.

Another problem is that there is no documentation on how to print in the Xerox web page in SDK section; this information will be added by Xerox next year. A few things to print have been tested, but it has not been possible. A new command was tested with the IP printer and the document to print. We are interested in that the system detecting which printer is in use. For that reason, the printing of the new password would not be done. Because it is necessary to know the printer IP in which the user would like to print the new password.

Another problem we came across with was when installing a firmware new version 1.233.3 in the printer; the code provided by the user system password, stopped working. This is a problem as the code had no need to access the database and was an efficient code itself. The previous firmware version was 1.231.5. To solve this problem, we consulted SDK developers, with Xerox, a message was sent to Xerox forum and a technician came over to uninstall the current version. Eventually, the best solution to this problem was to install the previous version.

5.9 Function's description

Some tests have been made in my computer and others in the printer I am working on. The functions that can only be tested in the printer are: showing the PIN and username and the system password changes. The rest of the functions could be tested from the computer.

Following *requestSessionInfo* obtains the username and the security PIN. This code is efficient as we do not have to consult/check the database to look for the security PIN.

```
function requestSessionInfo() {
    try {
        xrxSessionGetSessionInfo("https://localhost", displaySessionInfo, callbackfail);
    }
    catch ( e ) {
        Util.writeInfo("Error getting session info: " + e);
    }
}
// Callback from request for session info. Display the resulting data.
function displaySessionInfo(sessRequest, sessResponse) {
    var sessInfoObj = xrxSessionParseGetSessionInfo(sessResponse);
    if (!sessInfoObj) {
        Util.writeInfo("displaySessionInfo: null sessionInfo");
    }
    try {
        var name = null; var contentArea1 = document.getElementById('programcontent1');
        name = xrxGetElementValue( sessInfoObj, 'username' );
        contentArea1.innerHTML = "<b>username: </b>" + ((name != null)?name: "NONE");
        name = xrxGetElementValue( sessInfoObj, 'password' );
        contentArea1.innerHTML += "<br/><b>password: </b>" + ((name != null) ?name:"NONE");
        this.Pass = name;
    }
    catch ( e ) {
        Util.writeInfo("displaySessionInfo: " + e);
    }
    }
}
```

Figure 29: Function requestSessionInfo().

In the function *displaySessionInfo*, a global variable for store the value of the password is used to be able to compare it later. We are not interested in comparing it with *displaySessionInfo* as such function shows the username when initiating the web page and we have to compare the password after the user has typed in the security PIN.

The next code uses numerical buttons like those in a keyboard. The user will have to type in the security PIN pressing the buttons and when he /she finishes press the OK button to assure everything is correct.

Advantages over the previous codes: there is no need to check that the user types any other non numerical characters, the keyboard does not take long in loading up as it displays in the screen directly and we save checking ups.

The user has only two attempts to type in the PIN correctly; in the case he/she does not type it in correctly the system will come out of the application; as a security precaution. When the user types in the PIN correctly, the system will show the next page. We use six global variables: **Code** keeps the PIN introduced value, **Pass** keeps the security PIN provided by the printer, **Try** the number of attempts to type in the security PIN correctly, **name** keeps the username provided by the printer, **Color** contains a false initialised vector that indicates no number has been pressed and **write** is a variable I use to write an asterisk each time the user presses a number.

```

var code = 0; var Pass = null; var name= null; var Try = 0; var write= "";
var Color = new Array('false', 'false', 'false', 'false', 'false', 'false', 'false', 'false', 'false', 'false');
function Enter_code(number) {
    if (this.name == null) { alert("Aguarde, por favor até que apareça o username."); } //if
    else {
        for (i=0; i < 10; i++) {
            if (Color[i] == true) {
                document.getElementById(i).style.color = '#000000'; Color[i] = false;
            } //if
        } //for
        Color[number] = true; code = code * 10 + number; write = write + "*";
        document.getElementById(number).style.color = '#ff0000';
        document.getElementById('programcontentnote11').innerHTML = "Caracteres introduzidos:" + write;
    } //else
}
function correct() { // Use the template function such as the one below to capture the keyboard value.
    if (this.Try == 2) { alert("Só tem duas tentativas."); exitApp( ); return (false); } //if
    else {
        this.Try++;
        if (code == 0) {
            alert("Só é possível mudar de senha com uma senha de impressão."); return (false);
        } //if
        if (code == this.Pass) { parent.location.href="index2.html" return (true); } //if
        else {
            alert("Password incorrecto, tente novamente!!!");
            document.getElementById('programcontentnote11').innerHTML = "Número de caracteres
            introduzidos: 0"; write = ""; code = 0; return (false);
        } //else
        return (true);
    } //else
}
function deletePass() { // This function deletes the characters introduced up to the moment
    document.getElementById('programcontentnote11').innerHTML = "Número de caracteres introduzidos: 0"
    code = 0; write = "";
}

```

Figure 30: Functions: Enter_code(), correct() and deletePass().

I have created the form as follows: I create the 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 buttons, corresponding to the only ones value that a security PIN can have and another button to delete the security PIN.

```
<table id="Table4" cellspacing="4" width="650" border="0" >
  <form name="codepad">
    <p align="center">
      <tr>
        <td><input style="WIDTH: 120px; HEIGHT: 60px" id ="1" onclick="Enter_code(1)" type="button" value=" 1
        "/></td>
        ....
        <td><input style="WIDTH: 120px; HEIGHT: 60px" id ="11" onclick="deletePass()" type="button" value="
        <" /></td>
      </tr>
    </p>
  </form>
</table>
```

Figure 31: Form with buttons.

The following function *exitApp* shows how to come out from the current page to return to the initial menu.

```
function exitApp() { //Exit the EIP application
    xrxSessionExitApplication("https://127.0.0.1", exitCallback);
}
function exitCallback (request, response, status) {
    alert("Exit Failure! Status returned: " + status + ": " + xrxGetStatusText());
}
```

Figure 32: Function *exitApp*().

In order to generate a password automatically, I have used the following function *generatePassword*. This function ensures that the password has a minimum of 6 characters and a maximum of 8. It also ensures that the new password would not have symbols.

```
function generatePassword() {
    var chars = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
    var randomstring = '';
    for (var i=0; i<8; i++) {
        var rnum = Math.floor(Math.random() * chars.length);
        randomstring += chars.substring(rnum,rnum+1);
    } //for
    this.Pass = randomstring;
    document.getElementById('programcontentnote').innerHTML = "<b>A password nova é " + this.Pass;
}
```

Figure 33: Function *generatePassword*().

The following function *enterUpperCase* puts the keywords letters in capital or lowercase.

```
function enterUpperCase() {
    if (this.majus == false) {
        this.majus = true;
        for (i=10; i<36; i++) {
            document.getElementById(i).value = document.getElementById(i).value.toUpperCase();
        }
    } //if
    else {
        this.majus = false;
        for (i = 10; i < 36; i++) {
            document.getElementById(i).value = document.getElementById(i).value.toLowerCase();
        }
    } //else
    document.getElementById('programcontentnote11').innerHTML = "Caracteres introduzidos: "+ p;
}
```

Figure 34: Function *enterUpperCase()*.

The following function *Enter_code* shows an alert to the user if the login has not been shown on the screen. In that case, we could have problems with the login, as an error message will appear saying that the user does not exist. On the contrary, if the login appears, it checks whether the key pressed is capital or lowercase, if the capital is being pressed and it is not a number, the capital letter is placed in the number variable. And for last the function *processCode* is called.

```
function Enter_code(number, num) {
    if (this.name == null) { alert("Aguarde, por favor até que apareça o username."); }
    else {
        for (i = 0; i < 36; i++) {
            if (Color[i] == true) {
                document.getElementById(i).style.color = '#000000'; Color[i]=false;
            } //if
        } //for
        Color[num] = true; document.getElementById(num).style.color = '#ff0000';
        if (isNaN(number) == true) {
            if (this.majus == true) { number = number.toUpperCase(); } //if
        } //if
        processCode(number);
    } //else
}
```

Figure 35: Function *Enter_code*.

The function *processCode* adds to the Pass variable, the letter that has a number variable and

afterwards, it keeps the password value and the login in the form variables.

```
function processCode(number){
    Pass = Pass + number; write = write + "*"; document.forms[2].Pass.value = Pass;
    document.getElementById('name').value = this.name;
    document.getElementById('programcontentnote11').innerHTML = " Caracteres introduzidos: " +write;
}
```

Figure 36: Function processCode.

The keyboard has the following buttons: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z. It also has a button to put the keyboard letters in capital or small letters and a button to delete the password. I have created the keyboard as follows:

```
<table id="Table4" cellspacing="3" width="650">
  <p align="center">
    <tr>
      <td><input style="WIDTH: 108px; HEIGHT: 28px" id = '1' onclick="Enter_code('1',1)" type="button" value="1" /></td>
      ....
      <td><input style="WIDTH: 120px; HEIGHT: 28px" onclick="enterMajus()" type="button" value=" Caps " /></td>
      <td><input style="WIDTH: 120px; HEIGHT: 28px" onclick="deletePass()" type="button" value=" <-- " /></td>
    </tr>
  </table>
```

Figure 36: Table with buttons.

5.10 Test made

Below samples on the tested security PIN is shown, where the entry data are shown, the exit data and an explanation of the testes made.

Data entry	Output	Explanation
123456	Password incorrecto, tente novamente!!!	The user types in 123456, the system displays a message informing the user that the security PIN is not correct and invites to try again.
14567	Password incorrecto, tente novamente!!!	Is the same as the first one, but with other digits.
My correct password	index2.html	The user types in the security PIN correctly and show the next page.
	Só é possível mudar de senha com uma senha de impressão	The user does not type anything, the system shows a message informing that it is only possible to change password if he/she has the security PIN.
1234	Password incorrecto, tente novamente!!!	The user types 1234, the system shows a message informing that the security PIN is

		not correct.
1234	Só tem duas tentativas. Sair	The user types in 1234 again and the system shows a message informing that he/she has only two attempts and will come out of the application.

6 Conclusions

The Project fulfills all the agreed requirements. Some of the initial requirements have been changed such as, printing a page with the new user password, because Xerox web pages do not have this information in the printing services section. I got in touch with a Xerox worker so that he could inform us about printing services and he told me that such functionality would become available next year. Also, the option to automatically generate a password has been added, which is an alternative to letting the user choose the new one.

The greater difficulty of the Project has been to find out how to add the applications to the Multifunction printers, as the documentation Xerox offers is sometimes a bit unclear. Once it is known how to add the applications, it is only a matter of programming and testing. For the development of the code, some examples provided by a Xerox worker, have been very useful. Specifically, it has been useful for the project the following functionalities: quit the application, getting the identification, the login and security PIN.

The advantages this new application offers are: the user can change his/her system password, at any time and day of the week, it is not necessary to use a computer, nor use the system offered by HelpDesk. It is a real application and it is another way of being able to change the system password. This alternative, allows all the students community, teachers and FEUP workers to change their password from any Xerox multifunction printer.

The development of this project has been as follows: firstly I started collecting information on the printing system of FEUP, I analyzed the technical architecture, I also read the documentation about EIP available Xerox web pages, consulted the EIP forum, analyzed the requirements, made the implementation and conducted the tests; at the same time, I have been keeping a docuWiki with information about the project, writing a user manual and the present memory.

I have not had any previous training by the company, as it was not necessary.

6.1 Personal conclusions

At a personal level, this internship has been a good experience, since it has been useful to know another way to work at a different country. Furthermore, it has been useful to learn new things and new programming languages such as Perl. HTML, already I knew, but have been able to extend my knowledge; JavaScript is very similar to Java. So, I have not had many problems. When you work with different programming languages, you realize its advantages and disadvantages, what can be of use in the future.

Generally, everything has been all right, the working atmosphere has been good and whenever I needed help, people tried to help me. I have been placed in a shared office and provided with a computer so that I could develop my job.

Since the first moment, I liked the idea of the Project and I thought it was interesting. At the beginning, I thought it was going to be more complicated as it was something I had never done before and new nothing about Xerox EIP technology.

Bibliography

- [1] **FEUP – CICA – Serviços CICA**
http://www.fe.up.pt/si/web_base.gera_pagina?P_pagina=21237, accessed on February 2008
- [2] **Quais são as responsabilidades da equipa do SiFEUP?**
<http://faqs.fe.up.pt/FAQs/Quais%20são%20as%20responsabilidades%20da%20equipa%20do%20SiFEUP.pdf>, accessed on February 2008
- [3] **Xerox Extensible Interface Platform (EIP) Advanced Development Software for Xerox Multifunction**
<http://www.office.xerox.com/eip/>, accessed on February 2008
- [4] **Xerox**
http://www.xerox.com/go/xrx/template/inv_rel_newsroom.jsp?ed_name=ESP_ServiciosWebNuevosDispositivos&app=Newsroom&view=NewsroomLite&format=article&Xentry=ESP&Xlang=es_ES, accessed on February 2008
- [5] **Chiletech**
<http://www.chiletech.com/link.cgi/Empresas/X/XEROX/19388>, accessed on February 2008
- [6] **Xerox anuncia nuevas incorporaciones a su línea de productos multifunción**
http://www.financialtech-mag.com/000_estructura/index.php?id=24&idb=143&ntt=9388&sec=27&vn=1, accessed on February 2008
- [7] **Xerox cambia el entorno de oficina con la nueva tecnología EIP**
<http://www.idg.es/pcworldtech/mostrarVideo.asp?id=465>, accessed on February 2008
- [8] **Infonws – Xerox y sus socios comerciales presentaron ocho softwares para mejorar el manejo de documentos**
http://www.infonews.info/portal/index.php?option=com_content&task=view&id=7204&Itemid=109, accessed on February 2008
- [9] **Xerox ofrece soluciones de documentos –eSemanal**
http://www.esemanal.com.mx/articulos.php?id_sec=2&id_art=5045, accessed on February 2008
- [10] **SDKs, Redistributables & Service Packs**
<http://msdn2.microsoft.com/en-us/netframework/aa731542.aspx>, accessed on March 2008
- [11] **Visual Studio .NET. Manual de ASP.NET. Tutorial de ASP.NET. WebEstilo**
<http://www.webestilo.com/aspnet/aspnet02.phtml>, accessed on March 2008

- [12] **Intel NetportExpress 10 y 10/100 Servidores de impresión – Configuración de Windows 2000**
<http://www.intel.com/support/mt/sp/netport/10100/sb/cs-015195.htm>, accessed on April 2008
- [13] **Gestión y creación de colas de impresión**
<http://www.trucoswindows.es/topico-1198-gestion-y-creacion-de-colas-de-impresion.html>, accessed on April 2008
- [14] **Uso de las impresoras**
http://www.freebsd.org/doc/es_ES.ISO8859-1/books/handbook/printing-using.html, accessed on April 2008
- [15] **Tutoriales de Perl**
http://perlenespanol.baboonsoftware.com/tutoriales/html/desplegar_html.html, accessed on May 2008
- [16] **Desenvolvimento do SiFEUP**
https://www.fe.up.pt/si/web_base.gera_pagina?p_pagina=1571, accessed on June 2008

Acronyms

CICA	Prof. Correia de Araújo Computer Centre
EIP	Extensible Interface Platform
FEUP	Faculty of Engineering of University Porto
HTML	HyperText Markup Language
IDE	Integrated Development Environment
IP	Internet Protocol
IT	Information Technology
MFD	MultiFunction printer
NIC	Network Interface Card
PIN	Personal Identification Number
SDK	Software Development Kit
SiFEUP	Information System of Faculty of Engineering of University Porto
SMS	Short Message Service
SOA	Service Oriented Architecture

Glossary

Equitrac	Equitrac is software that provides document cost management (auditing, allocation, recovery) and output management solutions for printers, copiers and multifunction devices.
Follow-You Print	The work is sent to a printing queue at the server; it can be released at any equipment.
NIC	NIC is a piece of computer hardware designed to allow computers to communicate over a computer network.
SDK	SDK is typically a set of development tools that allows a software engineer to create applications for a certain software package, software framework, hardware platform, computer system, operating system, or similar platform.
Security PIN	Security PIN is secret number sequence that allows the user to use the printing services.
SIGARRA	SIGARRA previously named SIFEUP at FEUP, is a service that allows the registering, referencing and dealing with information on various processes of the university.
SOA	SOA is a computer system's architectural style for creating and using business processes, packaged as services, throughout their lifecycle.

TCPGate	TCPGate is a service that allows access to FEUPnet from outside, using an access via modem, ISDN (Euro-ISDN) or via Internet.
VLAN	VLAN has the same attributes as a physical LAN, but allows end stations to be grouped together even if they are not located on the same LAN segment.
Web 2.0	Web 2.0 is a term describing the trend in the use of World Wide Web technology and web design that aims to enhance creativity, information sharing, and, most notably, collaboration among users.
Widget	Widget is a little application or program, usually displayed in small files which are executed by a Widget Engine.
Xerox EIP	Xerox EIP is a software platform upon which developers can use standard web-based tools to create server-based applications that can be configured for the Multifunction Printer's touch-screen user interface.
XRXWebServices	XRXWebServices contains style sheets and necessary widgets.

ANNEX A: How the linserv program works

As said in chapter 3, linserv is the tool used by the HelpDesk personnel to manage the FEUP's user's accounts. Its user interface will be here described in more detail.

Add menu

It has the following fields: login, name, SiFEUP, password, group (teacher, a graduate student, master student, doctoral students, researchers), department, shell, accounts (Ldap, Windows, TcpGate, IBM Cluster). You have the option to print information in English and in Portuguese.

It also allows you to create temporary accounts for internal/external FEUP training. Another option is to consult the alias to check whether they exist or not. You can find the alias reports on the student's login. You can also change a user's nickname, or given him/her one.

Figure 37: Add menu.

Seek menu

This menu allows queries on the main database. It is available in the login or in the user code. In this window you can find the following fields:

- **Log:** it contains all the alternated passwords and comments on the database.
- **Edit:** allows editing, altering data users. Also allows students to migrate accounts.
- **Password:** it allows changing all passwords.
- **Diary:** schedule allows locking/unlocking accounts.

There are several systems that can be shut down or have user accounts blocked to them. It contains the following systems:

- **Ldap** authenticates emails, Linux, Moodle and Digitool. You can delete, block, generate a new password and verify the quota;
- **Windows** authenticates Windows and Samba;
- **SiFEUP** authenticates in SiFEUP;
- **TcpGate** authenticates a wireless, VPN and TcpGate;
- **Voip** authenticates on Voip.



Figure 38: Seek menu.

Extra menu

In this menu you can change the password (SiFEUP), you have to manually enter the password. The other option is altering accounts (HelpDesk) allows unlocking, locking or generation of new passwords.

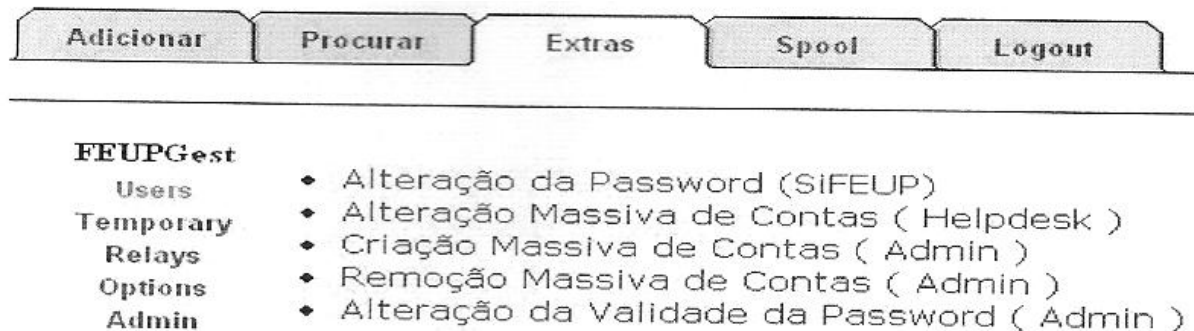


Figure 39: Extra menu.

Spool Menu

This menu allows verification of the passwords alterations made at that moment by the users.

Logout Menu

Ends the session to which we are connected and it allows other users to initiate a new session.

ANNEX B: Description of the SDK

As said in Chapter 3, the software development tools used for the project were: Ms Visual Studio 2005 and Emacs. Some aspect of their use will now be more detailed. This annex contains information about the documents and files that are found in the Xerox web page in SDK section.

Accounting Services

Consolidated JBA (Job Based Accounting) Sim Test Matrix.ZIP: contains 18 excel documents for doing tests;

jba.ZIP: contains 5 files, I installed JBA SDK. It has created a folder which contains: SDK license agreement.

JBA-API-V4.0.6.pdf: contains information on command post and get with examples.

JBA-API-V4-0.6.pdf: contains information about commands;

JBA Solution Test Plan and Report Template ver1.03.0903: contains information for doing tests;

JBASimIntegratedRelease11.EXE: I installed J2SE Runtime Environment 5.0 Update 1. Typical version. Apache Tomcat Version 5.5.4. Connect 8080 in the port, user: admin, password: feup08. JBA also installs a Simulator. Installation complete. In this directory some examples can be found *C:\Program Files\Apache Software Foundation\Tomcat 5.5\webapps\jsp-examples*;

OffboxValidationSDK1[1].2B.1006.ZIP: I installed Xerox Offbox Validation SDK 1.1.2;

Presentation Services

Using “Presentation Services”, SDK developers can control applications as they appear or present themselves, or display on Xerox tactile devices.

EIP Technology presentation.pdf: contains SSL configuration, explains how to print, scan, and recommends how to use Firefox or a Simulator (the navigator of EIP).

- **Device provides embedded browser and platform API** (Application Programming Interface)
 - Widgets;
 - Web Services;
 - Web browser;
 - SNMP (Simple Network Management Protocol);
 - Plus legacy APIs (scan, accounting, etc.).
- **Partner web server provides HTML pages compliant with style guide**
- **Technologies for Web Development that can use**
 - HTTP 1.1;
 - HTML 4.01;
 - CSS 2.0;
 - Server-side scripting;
 - Javascript (ECMAScript V3);

- AJAX (JavaScript+DOM);
- Web services;
- DOM Level 2;

Web services

- Validate web service version;
 - Will also ensure that EIP is currently enabled;
- Register application;
 - URLs: application, icon, label (http or https);
 - Application name and company name;
- Name must be unique;

Xerox embedded browser extensions

Widgets

- Xerox widgets (provided as samples, some embedded);
- HTML/javascript widgets should be implemented according to style guide recommendations;

Graphic formats that can use

- JPEG;
- PNG;
- GIF;
- AnimatedGIF;

JavaScript events

- Suspend;
- Resume;
- Key press events (including green “Start” button);

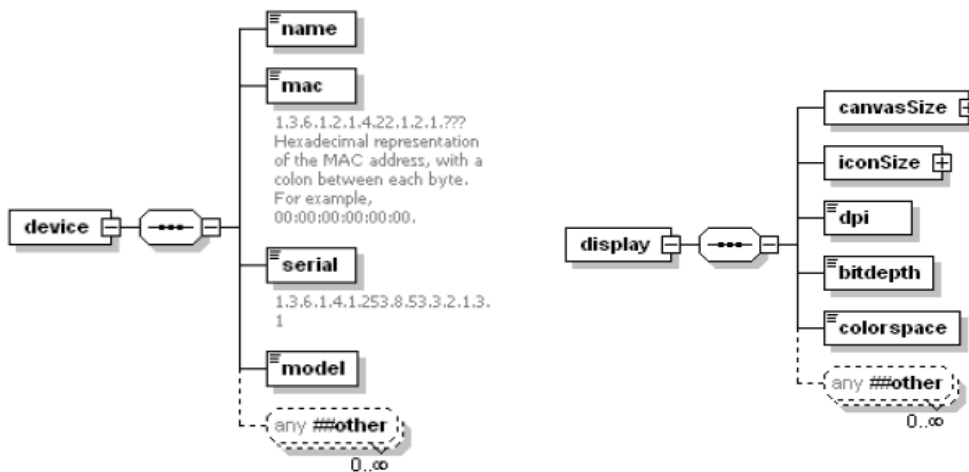


Figure 40: Information about the device and display.

Registration API (public)

- Allows registration of new services;
- Expected extension: ability to modify built-in services;

Session API (local only)

- GetSessionInformation;
- Returns username, password, accounting codes, e-mail address;
- Expected extensions: ability to set security, accounting data;

Protocol Configuration API (public)

- GetDeviceInformation;
- Returns S/N, Mac Address, Model, display parameters, SW versions, etc.
- Expected extensions: protocol enabling, settings, etc.

Xerox Scan Extension API (local only)

- InitiateScan, GetXrxResourceSimple;
- Initiates scan, returns supported scan parameters;
- Expected extensions: Microsoft Web Services for Devices (WSD);

EIPSDKSetup.msi: requires the installation: NET Framework version 1.1.4322, downloaded from: <http://www.microsoft.com/downloads/details.aspx?FamilyId=9B3A2CA6-3647-4070-9F41-A333C6B9181D&displaylang=en>. I installed .NET Framework SDK.



Figure 41: Installation of Microsoft .NET Framework SDK.

Fax Services

In this section we do not need anything, because it was not necessary for the project. Nevertheless, for completeness, the FaxSimTool contains:

Fax Receive Test Cases and Matrix.ZIP: contains examples of the cases to be scanned.

FaxSimTool 1[1].6.1203.ZIP: I installed FaxSimTool, serial number: 1. FaxSimTool is a Java based application whose main purpose is to serve fax jobs collected from various Xerox Document Centres to a Fax Server Solutions or Applications. FaxSimTool assists our Partners to test their application during the development cycle and at the Solution Validation Test cycle. It explains how it is to install it.

FaxSimTool v2[1].0.0304: FST JobLogs Test Documentation 0205.ZIP: explains how to test FaxSimTool.

XBP Fax Solution Test Plan and Report Template v1[1].01.0403.DOC: explains how should a document be to do tests.

Scanning Services

ParserSDK 4[1].2.0 1006.ZIP: the file installs the parserSDK Xerox JTL Parser SDK 4.2.0. ParserSDK contents: the Job Template Language Parser (JTL) SDK Version 4.2.00 includes the following components: documentation; sample applications; **TPDemo:** sample application illustrating all the primary parser methods; **build-job or Log Parser:** this illustrates how to

parse the results of a build-job; **language Version or Editor**: this illustrates how to modify the language version stored in a template; **multi-Attribute Editor value**: this illustrates how to read and modify multivalued attributes; **XNSParser**: this includes files and DLL for the XNS Parser, version 4.2.00;

Required Minimal Scan Test.ZIP: contains a folder with tests, and the code simulation xml and VBScript.

Scan Template Management SDK 103 1006.ZIP: installed the scan template management SDK.

ScanSDK.ZIP: installed **Xerox Scan SDK**: contents: **developer_Guide.pdf**: provides an overview of scan implementation and guidelines for integration; **attribute_Manual.pdf**: provides a developer reference manual for Scan attributes and Services used in templates; **Xerox HTTP Filing Protocol.pdf**: provides details on Xerox HTTP/S implementation for file transfer of scan outputs; ScanOutputSamples for integration testing and application development. Sample Template; httpScriptSamples (for HTTP/S file transfer); samples (range of samples from current and old devices); outputUsingCommonXST (samples from devices using common.xst the base xst); templates (base xst file sample).

SMARTsend2[1].0WebAPI1.0 SDK.ZIP: SMARTsend WebAPI SDK provides documentation and examples to support the use of SMARTsend WebAPI, a set of web services that support SMARTsend Workflow management activities. Version 1.0 of the SDK documents WebAPI version 1.0, which is included with SMARTsend version 2.0. The WebAPI SDK includes: a set of WSDL (Web Services Description Language) files that describe each of the WebAPI web service interfaces; several QuickStart samples and guides to get you started in creating applications that leverage the SMARTsend WebAPI; a SampleCode example that demonstrates a more complex application for managing Workflow publications and credentials. The sample code is intended for use in a Visual Studio .Net Windows application project;

StandardScanTestProcedure and ReportTemplate 1 00.ZIP: contains a document with information about the tests.

XDCLibrary v2 09 1206.ZIP: the file cannot be opened.

ScanSDK Installer.zip: I installed scanSDK.

Authentication Services

Authentication services SDK allows developers to create applications that support authentication and provides selective access to the features of the devices Xerox.

Convenience Authentication Install v102 1006.ZIP: contains files to install the Convenience Authentication. I Installed: a web setup application for the server simulator application; a source code for server simulators; a source code for device simulator; a GSOAP headers.

Printing Services

There are no documents or files in Printing Services section. According to Xerox, this information would be available from 2009 onwards.

ANNEX C: Description about other tests

Next, it is shown some code that has been tested before reaching the current design, with its advantages and disadvantages.

The function *validate* contains a form with a single field: security PIN. It contains a button to validate the form. Inconveniences: with this keyboard it is necessary to check that the user only types in numbers, not symbols.

```
function validate (formulario) {
    formulario.password.focus(); var checkOK = "0123456789";
    var checkStr = formulario.password.value; var allValid = true; var decPoints = 0; var allNum = "";
    for (i = 0; i < checkStr.length; i++) { //Verify that all characters are numeric
        ch = checkStr.charAt(i);
        for (j = 0; j < checkOK.length; j++)
            if (ch == checkOK.charAt(j)) break;
        if (j == checkOK.length){ allValid = false; break; } //if
        allNum += ch;
    } //for
    if (!allValid) { //inform that the characters are incorrect
        alert("Escreva só números no campo \"password\"."); formulario.password.focus(); return (false);
    } //if
    else { //verify that the password is 123456
        if (formulario.password.value == 123456) { alert("Password correcto");
            formulario.password.focus(); return (true);
        } //if
        else { alert("Password incorrecto"); formulario.password.focus(); return (false);
        } //else
    } //else
    return (true);
}
```

Figure 42: Function validate.

The form has been created as follows:

```
<form method="POST" name="registro" onSubmit="return validate(this)" ID="Form1">
  <table border="0" width="350" cellspacing="3" ID="Table4">
    <tr>
      <td width="50%" bgcolor="#FFF3D6" align="right">Password</td>
      <td width="50%" bgcolor="#FFDC88"><input type="password" name="password" size="20" ID="Text4"></td>
    </tr>
  </table>
  <p> <input type="submit" value="Enviar datos" name="enviar" ID="Submit1"> </p>
  <input class="xrx:command" id="Button2" style="WIDTH: 82px; HEIGHT: 35px; BACKGROUND-COLOR: blue"
  onclick="exitApp();" type="button" size="32" value="No" name="Button2" input="input" />
</form>
```

Figure 43: Form with one field: password and button send.

Conducted testing

Data entry	Output	Explication
123456	Password correcto	The user introduces 123456 and then the system shows that the security PIN is correct.
123	Password incorrecto	The user introduces 123456 and then the system shows that the security PIN is incorrect.
ola	Escreva só números no campo Password	The user introduces “ola”, the system informs the user that he/she can only type numbers in the security PIN field.

Inconveniences of the current code: it has to check whether the characters are valid; in the final code, the user can only press numbers.

Another test was to integrate Google and the university website in the printer, but the site is too large to be displayed on the printer. It is not practical, as we cannot scroll to move freely in the web page.

```
<html>
<head>
</head>
<body>
  <p> Queres acceder a internet? </p>
  <p> <a href="http://www.google.com">Google</a> </p>
  <p> <a href="http://www.fe.up.pt">FEUP</a> </p>
</body>
</html>
```

Figure 44: Google and FEUP's web.

For security reasons, the user has only two attempts to type in the security PIN, in the case the user types in the security PIN wrongly; the system will get out of the application. This way, if the user wants to introduce the security PIN, he/she will have to start from 0. The security PIN must be numerical. In the case the security PIN is correct, the system will show another website so the system password can be changed. The keyboard that calls the printer takes a few seconds to load up and allows the user to type symbols, alphanumerical characters, etc. Therefore, it is necessary to check up more often making it more inefficient.

Inconveniences compared to the final application code: checking that all the characters are correct and the fact that the tactile keyboard takes longer in loading.

```
var Pass = null; var Try = 0;
function reportValue(id) { // Use the template function such as the one below to capture the keyboard value.
  this.Try++;
  if (this.Try == 3) {
    alert("Só tem duas tentativas"); exitApp(); return (false);
  } //if
  else { if (node.value.length == 0) { node.focus();
    alert("Só é possível mudar de senha com uma senha de impressão."); return (false);
  } //if
  var node = document.getElementById(id); var valid = "0123456789-"; var hyphencount = 0;
  var checkOK = "0123456789"; var checkStr = node.value; var allValid = true; var decPoints = 0
```

```

var allNum = "";
for (i = 0; i < checkStr.length; i++) {
    ch = checkStr.charAt(i);
    for (j = 0; j < checkOK.length; j++)
        if (ch == checkOK.charAt(j)) break;
    if (j == checkOK.length) { allValid = false; break; } //if
    allNum += ch;
} //for
if (!allValid) {
    alert("Escreva só números no quadro de texto");
    node.focus(); node.value = null; return (false);
} //if
else {
    if (node.value == this.Pass) {
        alert("Password correcto!!!"); parent.location.href="index2.html" return (true);
    } //if
    else {
        alert("Password incorrecto!!!"); node.focus(); node.value = null; return (false);
    } //else
} //else
return (true);
} //else
}
<input class="xrx:keyboardtext" lang="en" id="kbd1" type="password" size="30" name="jeff"
onsave="reportValue('kbd1')" />

```

Figure 45: Function reportValue.

Conducted testing

Data entry	Output	Explication
123456	Password incorrecto!!! Because my password no is 123456	The user introduces 123456 and then the system shows that the security PIN is incorrect.
olaola	Escreva só números no quadro de texto	The user introduces “olaola” as I mentioned earlier, the security PIN can only have numerical characters.
123456780000	Password incorrecto!!!	Is the same as the first but with more digits.
My correct password	Password correcto!!!	The user types the security PIN correctly; the system shows a message informing him/her it is correct.
	Só é possível mudar de senha com uma senha de impressão	The user does not type anything, the system informs him/her that it is only possible to change password if he/she has the security PIN.
Types wrongly the security PIN twice.	Só tem duas tentativas, gets out of the application.	If the user types wrongly the security PIN twice, the system shows him/her a message informing that he/she has only two attempts and gets out of the application.