



UNIVERSIDADE DA BEIRA INTERIOR
Faculty of Engineering
Department of Informatics

New Concepts Integration on E-Learning Platforms

Tiago Miguel Carrola Simões

Dissertation to obtain the Degree of Master of Science in
Informatics Engineering
(2nd cycle of study)

Supervisor: Prof. Dr. Joel José Puga Coelho Rodrigues

Covilhã, Maio de 2012

Acknowledgements

This dissertation would not be possible without the collaboration of some persons and entities, which directly and indirectly helped to make all this possible.

My sincere thanks to professor Joel Rodrigues, my mentor and coordinator, for the opportunity to work in this project and also for his advice and support in all challenges and steps.

I am most grateful to the University of Beira Interior for allowing me the opportunity to work on the funded E-Learning 3.0/5635 QREN project. Thanks to all partners involved in the project (ISEP, iZone, and Maisis), to professor Abel Gomes and to all other friends and family not mentioned here. A special thanks also to the Next Generation Networks and Applications Group (NetGNA), *Instituto de Telecomunicações*, Covilhã delegation, for all the support, excellent environment and cooperation inside the group.

Last but not least, a special thanks to my girlfriend Carina for all the support and patience during the long hours of work, where nothing would be possible without her.

Abstract

The learning experience has evolved into the virtual world of the Internet, where learners have the possibility to shift from face-to-face learning environments to virtual learning environments supported by technologies. This concept, called e-learning, emerged in the early 1960s where a group of researchers from the Stanford University, USA began experimenting different ways to publish and assign learning content using a computer. These experiments were the beginning that led to the creation of countless learning platforms, initially constructed in standalone environments and later ported to the Internet as Web-based learning platforms. As initial objectives, these learning platforms include a collection of features to support instructors and learners in the learning process. However, some of these platforms continued to be based on an old instructor-centered learning model and created a collection of outdated technologies that, given the current need to a learner-center learning model and the existence of Web 2.0 technologies, become inadequate. As a solution to address and overcome these challenges, a friendly user interface and a correct root incorporation of Web 2.0 services a platform designed to focus the learning experience and environment personalization into the learner is needed to propose.

In an operating system (OS) context the graphic user interface (GUI) is guided by a collection of approaches that details how human beings should interact with computers. These are the key ideas to customize, install, and organize virtual desktops. The combination of desktop concepts into a learning platform can be an asset to reduce the learning curve necessary to know how to use the system and also to create a group of flexible learning services. However, due to limitations in hypertext transfer protocol-hypertext markup language (HTTP-HTML) traditional solutions, to shift traditional technologies to a collection of rich Internet application (RIA) technologies and personal learning environments (PLEs) concepts is needed, in order to construct a desktop-like learning platform. RIA technologies will allow the design of powerful Web solutions containing many of the characteristics of desktop-like applications. Additionally, personal learning environments (PLEs) will help learners to manage learning contents.

In this dissertation the personal learning environment box (PLEBOX) is presented. The PLEBOX platform is a customizable, desktop-like platform similar to the available operating systems, based on personal learning environments concepts and rich Internet applications technologies that provide a better learning environment for users. PLEBOX developers have a set of tools that allow the creation of learning and management modules that can be installed on the platform. These tools are management learning components and interfaces built as APIs, services, and objects of the software development kit (SDK). A group of prototype modules

were build for evaluation of learning and management services, APIs, and SDKs. Furthermore, three case studies were created in order to evaluate and demonstrate the learning service usage in external environments. The PLEBOX deployment and corresponding features confirms that this platform can be seen as a very promising e-learning platform. Exhaustive experiments were driven with success and it is ready for use.

Resumo

A experiência de aprendizagem baseada em tecnologias evoluiu para o mundo virtual da Internet, onde os alunos têm a possibilidade de mudar uma aprendizagem presencial em sala de aula para uma aprendizagem baseada em ambientes virtuais de aprendizagem suportados por tecnologias. O conceito de e-learning surgiu nos anos sessenta (1960) quando um grupo de investigadores da Universidade de Standford, nos Estados Unidos, começaram a experimentar diferentes formas de publicar e atribuir conteúdos de aprendizagem através do computador. Estas experiências marcaram o começo que levou à criação de inúmeras plataformas de aprendizagem, inicialmente construídas em ambientes isolados e depois migradas para a Internet como plataformas de aprendizagem baseadas na Web. Como objectivos iniciais, estas plataformas de aprendizagem incluem um conjunto de recursos para apoiar professores e alunos no processo de aprendizagem. No entanto, algumas destas plataformas continuam a ser baseadas em velhos modelos de aprendizagem centrados no professor, criadas com base em tecnologias ultrapassadas que, dadas as necessidades actuais de um modelo de aprendizagem centrado no aluno e da existência de tecnologias baseadas na Web 2.0, se tornaram inadequadas. Como abordagem para enfrentar e superar estes desafios propõem-se uma plataforma focada na personalização do ambiente de aprendizagem do aluno, composta por uma interface amigável e uma correcta incorporação de raiz de serviços da Web 2.0.

No contexto dos sistemas operativos (SOs) o *graphic user interface* (GUI) é desenhado tendo em conta um conjunto de abordagens que detalha como as pessoas devem interagir com os computadores. Estas são as ideias chave para personalizar, instalar e organizar áreas de trabalho virtuais. A combinação do conceito *desktop* com uma plataforma de aprendizagem pode ser um trunfo para reduzir a curva de aprendizagem necessária para saber como utilizar o sistema e também para criar um grupo de serviços flexíveis de aprendizagem. No entanto, devido as limitações em soluções tradicionais *hypertext transfer protocol - hypertext markup language* (HTTP - HTML), é necessário migrar estas tecnologias para um grupo de tecnologias *rich Internet application* (RIA) e conceitos presentes em ambientes personalizados de aprendizagem (*personal learning environment* - PLE) para construir uma plataforma baseada em ambientes de trabalho virtuais de aprendizagem. As tecnologias RIA irão permitir a criação de soluções Web poderosas que contêm muitas das características disponíveis em aplicações desktop. Adicionalmente, o conceitos de PLE irá ajudar os alunos a gerir os seus próprios conteúdos de aprendizagem.

Nesta dissertação, com base nas características apresentadas anteriormente, é apresentada a *personal learning environment box* (PLEBOX). A plataforma PLEBOX é uma solução de aprendizagem parametrizável com um ambiente de trabalho semelhante aos sistemas

operativos actuais, baseando-se em *personal learning environments* e tecnologias RIA que fornecem um melhor ambiente de aprendizagem para os seus utilizadores. Os programadores da PLEBOX têm ao seu dispor um conjunto de ferramentas que permitem a criação de módulos de aprendizagem e administração que podem ser instalados na plataforma. Estas ferramentas são componentes de aprendizagem e interfaces construídos como APIs, serviços e objectos do *software development kit* (SDK). Foi construído um conjunto de módulos com o objectivo de avaliar e demonstrar os serviços de aprendizagem, os serviços de gestão, APIs e SDKs. Para além disso, foram criados três casos de estudo para avaliar e demonstrar a utilização dos serviços de aprendizagem em ambientes externos. O desenvolvimento efectuado até ao momento na PLEBOX e respectivos recursos confirma que esta plataforma pode ser vista com uma promissora plataforma de aprendizagem (e-learning), totalmente modular e adaptativa. Realizaram-se experiências exaustivas para testar a plataforma e estas foram realizadas com sucesso num ambiente real, estando assim a plataforma pronta para exploração real.

Keywords

E-learning, E-learning Solutions, Learning Management System, Personal Learning Environment, E-learning Technologies, Personal Learning Environment Box.

Contents

Acknowledgements	iii
Abstract	v
Resumo	vii
Keywords	ix
Contents.....	xi
List of Figures	xv
Acronyms	xix
Chapter 1 – Introduction	1
1.1 Focus.....	1
1.2 Objectives	3
1.3 Main Contributions.....	4
1.4 Dissertation Organization	4
Chapter 2 – Related Work	7
2.1 Introduction.....	7
2.2 E-Learning.....	8
2.3 E-Learning Platforms and Technologies	10
2.3.1 Open-source E-Learning Platforms	13
2.3.2 Commercial E-Learning Platforms.....	15
2.4 Personal Learning Environments	16
2.5 Rich Internet Applications	20
Chapter 3 – Requirements Analysis	23
3.1 User Cases Diagrams.....	23
3.1.1 APIs/Services and SDKs.....	23
3.1.2 Non-Learning Modules and Widgets.....	30
3.1.3 Learning Modules	31
3.2 The Relational Structured Database	37
4.3.1 Entity Relationship Model	37
3.3 Class Diagrams	39

Chapter 4 – PLEBOX Technologies.....	45
4.1 Introduction	45
4.2 Resources	46
4.2.1 Windows Server 2008 R2	46
4.2.2 Internet Information Services 7	46
4.2.3 Visual Studio 2010	47
4.2.4 Relational Database Management System – MySQL	47
4.2.5 ASP.NET and the Microsoft .NET Framework.....	48
4.2.6 REST and SOAP Web Services.....	48
4.2.7 Windows Communication Foundation 4.0.....	49
4.2.8 JavaScript Libraries	49
4.3 Database Views and Stored Procedures	50
4.3.1 Database Views	50
4.3.2 Stored Procedures.....	51
4.4 Architectures	53
4.4.1 Version 1.0.....	53
4.4.2 Version 2.0.....	55
4.4 Server and Client Architectures.....	58
4.5 Internal and External Development Scenarios	58
Chapter 5 – PLEBOX Components Design.....	61
5.1 Non-Learning Components	61
5.2 Learning Components.....	62
5.4 Software Development Kit Component.....	63
Chapter 6 - System Demonstration and Validation.....	67
6.1 Platform Usage	67
6.1.1 PLEBOX User Interface Main Environment Elements	67
6.1.2 Modules Management	70
6.1.3 Software Development Kit	70
6.1.4 Learning Modules.....	74
6.2 Aggregation Examples	82
6.2.1 Example 1 – ASP.NET Web Application.....	83
6.2.2 Example 2 – Web-Based JavaScript Application	83
6.2.3 Example 3 - Android Application	84

Chapter 7 – Conclusions and Future Work	85
References	89

List of Figures

Figure 1 - LMS Market Share for USA Institutions (<i>source: http://goo.gl/9iHT2</i>).....	2
Figure 2 - PLEBOX Illustration Concept.	3
Figure 3 - Web 2.0 Components Illustration.	7
Figure 4 - Learning Objects and SCORM Packages.	11
Figure 5 - Cost of Content Integration before and after SCORM.....	12
Figure 6 - Illustration of the Personal Learning Environment concepts.	17
Figure 7 - Guidelines for an Efficient Personal Learning Environment.	19
Figure 8 - Illustration of Rich Internet Applications concepts.	21
Figure 9 - Adopted Development Model.	23
Figure 10 - Main User Case diagram.	24
Figure 11 - Modules Management User Case Diagram.	24
Figure 12 - Menu Management User Case Diagram.....	25
Figure 13 - Windows Management User Case Diagram.	25
Figure 14 - Database Management User Case Diagram.	25
Figure 15 - Notifications Management User Case Diagram.	26
Figure 16 - Interface Objects Provider User Case Diagram.	26
Figure 17 - Settings Management User Case Diagram.	27
Figure 18 - Utilities Provider User Case Diagram.....	27
Figure 19 - Input/Output Management User Case Diagram.....	27
Figure 20 - Handle Users User Case Diagram.	28
Figure 21 - Handle Courses User Case Diagram.....	29
Figure 22 - Handle Curricular Units User Case Diagram.....	29
Figure 23 - Handle Repositories User Case Diagram.	29
Figure 24 - Handles Contents User Case Diagram.	30
Figure 25 - Manage Modules User Case Diagram.....	30
Figure 26 - Manage Notifications User Case Diagram.	31
Figure 27 - Profile User Case Diagram.	31
Figure 28 - Modules User Case Main Diagram.	32
Figure 29 - Manage Content User Case Diagram.....	32
Figure 30 - Manage Content Standards User Case Diagram.....	32
Figure 31 - Manage Content Types User Case Diagram.....	33
Figure 32 - Manage Courses User Case Diagram.	33
Figure 33 - Manage Curricular Units User Case Diagram.	33
Figure 34 - Manage Courses and Curricular Units User Case Diagram.....	34
Figure 35 - Manage Learners User Case Diagram.	34
Figure 36 - Manage Notifications User Case Diagram.	34

Figure 37 - Manage Repositories User Case Diagram.	35
Figure 38 - Manage Users Groups Module User Case Diagram.	35
Figure 39 - Manage Users Module User Case Diagram.	36
Figure 40 - Collaboration With Learners User Case Diagram.	36
Figure 41 - Collaboration With Instructors User Case Diagram.	36
Figure 42 - Assignments Module User Case Diagram.	37
Figure 43 - Access and Utilities Components Class Diagram.	39
Figure 44 - Desktop Components Class Diagram.	40
Figure 45 - User Class Diagram.	40
Figure 46 - Content Class Diagram.	41
Figure 47 - Courses Class Diagram.	41
Figure 48 - Curricular Units Class Diagram.	42
Figure 49 - Notifications Class Diagram.	42
Figure 50 - Repositories Class Diagram.	43
Figure 51 - PLEBOX Logo Illustration.	45
Figure 52 - WCF Services Basic Concept.	49
Figure 53 - Conceptual Zones and Respective Views.	50
Figure 54 - Conceptual Zones and Respective Stored Procedures.	51
Figure 55 - Architecture of PLEBOX V1.0.	54
Figure 56 - Architecture of PLEBOX V2.0.	57
Figure 57 - Server Hardware Architecture.	58
Figure 58 - Internal and External Development Scenarios.	59
Figure 59 - Publish Profile.	59
Figure 60 - Concept Sketch of the PLEBOX's Desktop	61
Figure 61 - Concept Sketch of the PLEBOX's Learning Components.	62
Figure 62 - Concept Sketch of the PLEBOX's Software Development Kit.	63
Figure 63 - Window Interface Object.	64
Figure 64 - Dynamic Table Interface Object.	64
Figure 65 - Upload Interface Object.	64
Figure 66 - List Interface Object.	64
Figure 67 - Balloon Sub-Object.	65
Figure 68 - Modal Window Sub-Object.	65
Figure 69 - Tab Interface Object.	65
Figure 70 - Color Picker Interface Object.	65
Figure 71 - Button Image Sub-object.	65
Figure 72 - Images Slide Sub-object.	65
Figure 73 - Authentication User Interface Element.	68
Figure 74 - Desktop User Interface Element.	68
Figure 75 - User Custom Desktop.	69
Figure 76 - User Profile Widget.	69

Figure 77 - Google+ Widget.....	69
Figure 78 - Twitter Widget.	69
Figure 79 - Notifications Widget.	69
Figure 80 - Manage Modules.	70
Figure 81 - Window Interface Object.	71
Figure 82 - Change the Icon of the Window.....	71
Figure 83 - Maximized Window.....	71
Figure 84 - Window as Icon.	71
Figure 85 - Make the Window Transparent.	71
Figure 86 - Dynamic Table Interface Object.....	71
Figure 87 - Records Search.	71
Figure 88 - Records Control.	72
Figure 89 - List Interface Object.	72
Figure 90 - Upload Interface Object.	72
Figure 91 - Sub-Object Balloon.	73
Figure 92 - Sub-Object Modal Window.	73
Figure 93 - Popup Interface Objects Usage Example.	73
Figure 94 - Tab Interface Object - Label Encapsulated.	74
Figure 95 - Tab Interface Object - Label and Textbox Encapsulated.	74
Figure 96 - Color Picker Interface Object Inside the Tab Interface Object.....	74
Figure 97 - Panel Interface Object.	74
Figure 98 - Images Slider Interface Sub-Object.	74
Figure 99 - Image Button Interface Sub-Object.	74
Figure 100 - List Users.....	75
Figure 101 - Add User.	75
Figure 102 - Edit User Information.	75
Figure 103 - Edit User Associations.....	75
Figure 104 - Manage Courses.	76
Figure 105 - Manage Types of Courses.	76
Figure 106 - Manage Curricular Units.	76
Figure 107 - Edit Curricular Unit.....	76
Figure 108 - Manage Content.....	77
Figure 109 - Manage Content User Association.	77
Figure 110 - Manage Content Curricular Unit Association.	77
Figure 111 - Manage Content Standards.	77
Figure 112 - Edit Standards.....	77
Figure 113 - Manage Types.	78
Figure 114 - Edit Type.....	78
Figure 115 - Manage Notifications.....	78
Figure 116 - Manage Repositories.....	79

Figure 117 - Edit Repository.....	79
Figure 118 - Manage Courses and Curricular Units.....	80
Figure 119 - Assign Content to Curricular Unit.	80
Figure 120 - View Contents of the Selected Curricular Unit.	80
Figure 121 - View Content.....	80
Figure 122 - Edit Content.....	80
Figure 123 - Manage Learners.....	81
Figure 124 - View Learner Information.	81
Figure 125 - Instructor's Collaboration.	81
Figure 126 - Send Notification(s) to Learner(s).	81
Figure 127 - Assignments.	82
Figure 128 - Perform Learning Task.....	82
Figure 129 - Learner's Collaboration.	82
Figure 130 - Send Notification(s) to Instructor.	82
Figure 131 - Example 1 (ASP.NET Web Application).....	83
Figure 132 - Example 2 (Web-Based JavaScript Application).	83
Figure 133 - Authentication.	83
Figure 134 - Modules List.	83
Figure 135 - Example 3 (Android Application).....	84
Figure 136 - Mobile Login.	84
Figure 137 - Assignments List.	84
Figure 138 - Viewing Content.	84

Acronyms

CMS	Content Management System
CPU	Central Processing Unit
ER	Entity-Relationship model
FTP	File Transfer Protocol
FTPS	File Transfer Protocol Secure
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IEEE	Institute of Electrical and Electronics Engineers
IO	Input/Output
IT	Information Technology
LCMS	Learning Content Management System
LDAP	Lightweight Directory Access Protocol
LMS	Learning Management System
LO	Learning Object
LOM	Learning Object Metadata
LTSC	Learning Technology Standards Committee
NNTP	Network News Transfer Protocol
OS	Operating System
PLE	Personal Learning Environment
PLEBOX	Personal Learning Environment Box
RAM	Random-Access memory
REST	Representational State Transfer
RIA	Rich Internet Application
SCORM	Sharable Content Object Reference Model
SDK	Software Development Kit
SLK	SharePoint Learning Kit
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
UBI	University of Beira Interior
W3C	World Wide Web Consortium
WCF	Windows Communication Foundation
WWW	World Wide Web

Chapter 1 - Introduction

This chapter details the focus, objectives, contributions, and an overview about the structure of this dissertation. The following sections provide an introduction to the e-learning topic, platforms, technologies, open issues, and also an introduction to a new learning platform, called personal learning environment box.

1.1 Focus

Distance learning comes from the use of postal service to receive educational tools, submit papers, and receive the corresponding feedback. Nowadays, thanks to the Internet, the submission and feedback can be virtually instantaneous where the availability of telecommunication networks and information and communication technologies in education became the key element in the process of changing the current school and educational systems [1-2].

In the past few years, e-learning has emerged as a promising solution to a lifelong learning and on-the-job work force training. E-learning can be defined as technology-based learning in which learning materials are delivered electronically to remote learners via a computer network [3]. This concept is supported by a set of learning platforms, which are based on learning solutions installed on Web servers that are used to administer, distribute, and supervise the educational activities of an organization or institution [4].

E-learning platforms are mainly described as learning management system (LMS), content management system (CMS), or learning content management system (LCMS). First, learning management system is a technological solution that aims to manage learning users, resources, progresses, content registration, classroom, and a set of different ways to deliver learning content assigned to a set of users or courses [5]. Second, content management system aims to create, deploy and retain a range of learning objects that may or may not be in learning standard. Third, learning content management system is a combination of learning management system and content management system features. Out of this division, there is also a Web-based environment designed to support the learner, called personal learning environment (PLE) that aims to combine a set of available learning and Web 2.0 technologies (e.g. LMS, CMS, Blogs, Wikis, and Chats) to provide a number of settable learning environments. In contrast to traditional LMS-driven solutions, a PLE takes a learner-centric approach where its technologies belong and are controlled by individual learners [6].

The concept of learning object (LO) is derived from object-oriented programming ideas where each object represents a set of entities and its respective characteristics. The main idea, in an e-learning topic, considers that any LO can be shared between learning platforms. Furthermore, a learning object can reduce the development cost and accelerate the

development of teaching materials [7]. On the other hand, learning standards like the sharable content object reference model (SCORM), offers ways to represent, package, and publish a range of learning objects.

A number of technological learning solutions for schools, universities, and other organizations emerged over the years and, as shown in the Figure 1, platforms like Blackboard and Moodle are having a great success in different American institutions. However, many of these solutions are not being created taking into account a new set of concepts and technologies. Instead they have been wrongly stretched and patched to use technologies like mobile frameworks and rich Internet applications frameworks. One ideal approach may involve the incorporation of these concepts at the root of the learning solution, offering a proper encapsulation of current services. Furthermore, these systems use a set of deployment learning ways that somehow are outdated to meet the expectations of our learners where each learner has their one way to learn and assimilate knowledge. As evidence, a development in education has shifted its emphasis from instructor-centered learning to learner-centered learning [8]. In conclusion, current learning solutions have dominated Internet-based education for the past two decades, yet, some of these traditional solutions does not implement current Internet technologies [9].

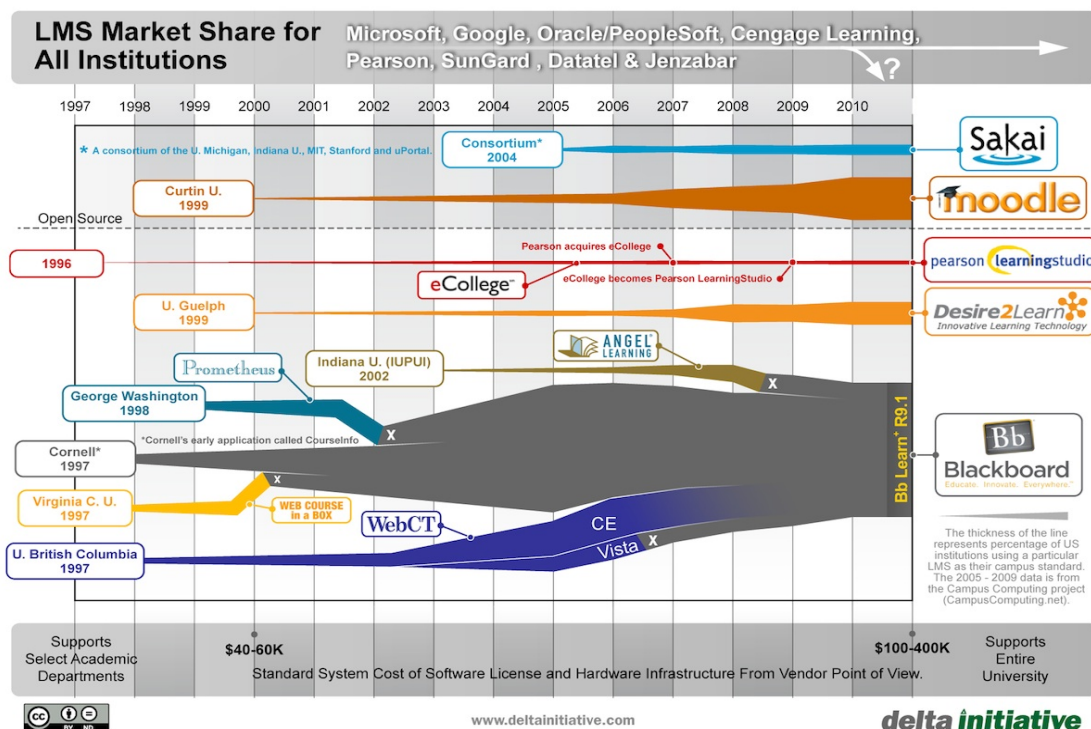


Figure 1 - LMS Market Share for USA Institutions (source: <http://goo.gl/9iHT2>).

Taking into account the above-mentioned problems, a new solution should be made from scratch considering a new innovative user interface that uses a set of concepts present in rich internet applications (RIA), a set of services that are driven to be used not only inside the

platform but also outside of the platform and, finally, a set of learning services and APIs created to centralize the learning experience into the learner. In this context the personal learning environment box (PLEBOX) was proposed. The concept of the box, illustrated in Figure 2, outlines the abstract concept that represents the encapsulation of a number of learning environments, learning modules, non-learning modules, and users.

The PLEBOX platform aims to address some of the above-identified issues and challenges providing a different number of services, APIs, and SDKs to develop learning and non-learning modules that can be easily installed “inside the box”. Furthermore, the provided learning services and non-learning services, SDKs, learning and non-learning APIs could be also used to develop internal components. On top of this, the personal learning environment box platform provides a desktop-like user interface to aggregate modules and components, not denying the possibility to incorporate learning and non-learning features into a set of distinct external environments (e.g. Blackboard, Joomla, Android application, Windows mobile application, Moodle, and SharePoint 2007/2010).



Figure 2 - PLEBOX Illustration Concept.

1.2 Objectives

The main objective of this dissertation is the analysis and construction of a number of application programming interfaces (APIs) in order to provide LMS functionalities and simplify the development of modules and features for the personal learning environment box platform (PLEBOX). In order to achieve this main objective, the following partial objectives were identified:

- Study of the related work about e-learning and tools;
- Detailed analysis of the first PLEBOX prototype;
- Requirement analysis;
- Construction and validation of a new LMS layer;
- Construction and validation of a set of APIs focusing on facilitate the development of components (e.g. modules, features);
- Construction and validation of a set of APIs focusing on managing the system;

- Construction and validation of a set of APIs focusing on supplying learning features.

The main objective has been widely overcome, and a new set of objectives were proposed. It includes the ability to publish a set of learning and non-learning operations to be used not only within the platform but also outside of the platform. A second objective includes the reunification of development operations into an SDK that will provide a range of interface objects to produce and deploy, as fast as possible, a set of components (modules and features). In order to achieve these main objectives, the following partial objectives were identified:

- Study service-oriented architectures;
- Study necessary changes to be applied to the PLEBOX platform;
- Deployment of learning and non-learning features as Web services;
- Graphical construction of interface objects (e.g. dynamic tables and internal popups) to be used in the development of components for the box;
- Construction and deployment of a group of modules on the PLEBOX environment (e.g. courses management, curricular units management, or users management module).

1.3 Main Contributions

This dissertation focuses on researching new learning methods and concepts for learning processes based on technologies. These concepts represent the foundations of the joined forces of a personal learning environment and a learning management system, integrated into the PLEBOX platform.

The first scientific contribution refers to the construction of a learning environment and a set of common personal learning environment features. A number of learning and administration APIs were created to support the development and management of box components. This contribution was published on the Computer Applications in Engineering Education [14].

The second scientific contribution refers to the deployment of the PLEBOX platform into an innovative system model for distant and mobile learning, only possible given a set of learning-related APIs, SDKs, and Web services available on the PLEBOX side. This contribution was submitted for an international journal (Appendix 2).

1.4 Dissertation Organization

This dissertation is organized in seven chapters and the chapters are organized as follows. This chapter, the first, presents the focus of the dissertation including a perspective of the topic, issues, objectives, contributions, and the organization of the document. Chapter 2 describes the related work in e-learning platforms and associated technologies, personal learning environments, and rich Internet applications. Each topic is defined and also a set of

related associated technologies are identified. Chapter 3 outlines the requirements analysis that was carried out prior to the improvement of the platform, focusing on the relational structured database model, user cases, and class diagrams. Chapter 4 outlines a detailed description of the PLEBOX platform, used technologies, database views, stored procedures, architectures, and development scenarios. Chapter 5 presents the concepts linked to each PLEBOX service, API, and SDK. Chapter 6 details the system demonstration and validation, providing a set of screenshots to demonstrate the platform, modules, and three services aggregation examples. Finally, in chapter 7 is drawn the conclusion and suggested directions for further works.

Chapter 2 - Related Work

This chapter details the related work in e-learning, platforms, related technologies, personal learning environments, and also an overview about rich internet technologies. This chapter focuses mainly on finding issues surrounding the e-learning topic.

2.1 Introduction

The traditional world wide web (WWW), proposed by Tim Berners Lee in 1989, was based on static websites, where information was available for browsing without user interaction [2]. Nowadays, the world wide web progressed into a more interactive experience and the old concepts of static web pages became a thing of the past. Everything and everyone, somehow, are connected and the process of learning has evolved from classrooms and physical environments to a virtual and adaptive environment. This is particularly possible by the introduction of new characteristics and Web technologies that form the concept of Web 2.0. In Figure 3 is shown a list of components that form this concept, which comprises a range of components, like collaborative applications (e.g. blogs and wikis), content syndication (RSS), web services, advance gaming support, mashups, dynamic content management to create, edit and remove content, and finally the component that outlines cloud tags. As proposed Web 2.0 is the second phase in the Webs evolution, and is also called the wisdom Web, people-centric Web, participative Web, and read/write Web. Version 2.0 harnesses the Web in a more interactive and collaborative manner, emphasizing peers social interaction and collective intelligence, and presents new opportunities for leveraging the Web and engaging its users more effectively [10].



Figure 3 - Web 2.0 Components Illustration.

The dominant learning approach in technology enhanced learning has been for many years the information transfer, which is based on the central figure of the instructor whose primary purpose is the provision of educational contents to be transferred to learners that consume these contents in passive ways. The concept of learning is vast, and many authors present different definitions for learning, however is considered that the process of learning is a complex process, driven by social, biological and physical environments [11]. It is also consider that the learning process takes place throughout ones life where there are different kinds of learning styles as well as learning domains. These learning styles and domains can be different from person to person. If we can identify our own learning style it will be a great help to learn quickly and systematically [12]. There is also a range of different categories for learning styles that fit the need of individual learners. The visual learning category where the learner is best when information is presented visually and in a written language, the auditory learning category that details that the user is better when information is presented in an oral language format, and finally the tactile/kinesthetic learning category where leaners are better when they are physically involved in an activity [13]. For making all this possible is necessary that software technologies fulfill the need of adapting the solution to the user and their own way of learning, incorporating new and old concepts in one unique solution and start treating the user as major role in the platform that need unique characteristics for personalization, content management, and other major features that enroll the learner into a better learning environment.

2.2 E-Learning

Historically when Thomas Edison invented motion pictures in 1891, he predicted that motion pictures one day replace textbooks. Nevertheless, textbooks are still considered as the essential material in traditional learning as well as tools for distance education. During the Second World War, films were used for training and later, open universities began to use videotapes and TVs as a mean of communication media for learning. During the late 1970s and the early 1980s, computer-based training (CBT) added interactions in automatic training. And finally in the 1990s, multimedia technologies further enhanced training contents and the popularity of the Internet and WWW technologies started a new era of distance learning.

There are many definitions of electronic learning (e-learning), depending on the target, however many of them mean exactly the same. With this in mind, we can consider that e-learning refers to the use of information and communications technologies (ICTs) to transform and support the learning process, and the evolution of e-learning has resulted from the emergence of the information society (IS) and it has greatly impacted on the global socio-cultural and economic development [14]. Simões *et al.* [14] and Wu *et al.* [15] argue that the development of e-learning has not been tested, and it is still in its infancy, considering that e-learning is at the initial stage of development and that the relevant theoretical research still need to be further improved. Rodrigues and Sabino [2] define e-learning as the use of

Internet technologies to provide, at the distance, a set of solutions for the improvement or the acquisition of knowledge and practical applicability of the same. E-learning can also be defined as a learning modality that facilitates the creation of a learning environment supported by the information and communication technologies [16], which can be used to obtain and share information [17]. Two last definitions state that e-learning is a generic term which encompasses the learning channels and systems such as remote delivery of multimedia content, email, video conferencing, online interactive classes, and learning management systems [18], also stating that e-learning is about connecting learners and instructors to each other and to professional support services, which in turn results in a learning platform [18].

The evolution of e-learning passes from a set of versions that have been discussed. There are three phases inherent to the development of e-learning processes: e-learning 1.0, 2.0 and the future version 3.0. The author Oliveira *et al.* [16] establishes that e-learning 1.0 is based, essentially, on reading and not on interaction, existing a great focus on the available content. In the other hand version 2.0 is focused on the participation and contribution, promoted through the use of tools and Web 2.0 services like blogs, wikis, tags, social bookmarking, and RSS feeds. E-learning 2.0 outlines the Web as a space of participation [16]. It is considered that the next version, 3.0, will be based on a number of improvements, such as: a new form for presenting content; a more interactive and powerful desktop-like user interface; a range of intelligent agents and a set of existing concepts from various non-web systems. Authors like Rego *et al.* [19] agree, stating that e-learning 3.0 will be at least at four key concepts: distributed computing, extended smart mobile and desktop technology, collaborative intelligent, 3D visualization and interaction.

Clearly e-learning has become an important part of our educational life and there are different web-based technologies that have been developed to support the user in the learning process [20]. Previous learning methods were restricted to access and assimilation of knowledge, so a web-based system is a valuable support for face-to-face communication as well as a way of transmitting the learning material to enhance the learner's own studies, supporting a higher degree of individualization [20-21]. These web-based systems are based in a set of common features. They are learner driven considering the learners as the main body; they provide nonlinear learning concepts stating that learners can adjust their learning process at any moment; they also deploy a set of effective ways for continue education and lifelong education, with the aim of reducing a large number of unnecessary duplication of work [22].

Web-based learning systems and the self-concept of e-learning have been used in different niches, not only in universities or schools but also in organizations. In the higher educational domain, for instance, learning technology plays critical role to allow learners to learn online were by they can learn even though of different geographical location with their instructor [17]. In the other hand in the organization domain, web-based learning systems are being

used to provide to their members with an opportunity to choose learning contents according to their need [23], and also to improve efficiency and costs reduction [18].

Most of today's e-learning systems are based on a constructionist pedagogical theory, this theory claims that individual learners construct mental models to understand the world around them. Such systems focus on enhancing the interaction of the learner and the instructor via the availability of various resources and activities, where the instructor avails various resources to the learner during the running period of the course. Mostly, there is little support for the instructor while creating the learning materials. Most of these systems lack an easy-to-use navigation mechanism of existing resources, so much effort is duplicated [20]. There are also other problems associated with e-learning types of implementations such as the lack of ability to stimulate learner participation until the end of the course [2]. This fact is the main reason why some of today's forms of e-learning are more oriented towards communication and collaboration between learners and instructors. Other problem is the level of adoption of various instruments being very different for each system [24], mainly due to limitations in the learning platform or event due to the poor bridge between the platform and the concepts in the learning process. Bergstedt *et al.* [21], Rahman and Sahibuddin [17] consider that formatting is tightly coupled with the content where there is no reuse of the learning content and also the lack of promoting effective social interaction in online learning. Learning platforms run their own learning objects and cannot be exchanged with another LMS, due to the lack of a reliable sharing mechanism among these standards such that results in an isolated e-learning platform. There are other problems that slow down the development of a new range of solutions. Current e-learning systems deny the creation of mental models and ways where the user is more comfortable and familiarized with the host system, so a major problem is the time that the learner takes to understand how to use the e-learning system, where much of the effort is applied in the wrong place. Learners and instructors need to spend more time with their learning contents and not the system itself.

2.3 E-Learning Platforms and Technologies

Over the past decade, an increasing number of schools and businesses have setup e-learning platforms for their learners and employees for the purpose of efficiency. These platforms have evolved rapidly and partially adapted to the reality of the current state of learning in different environments, however, a few of them are successful [25].

There are different e-learning technologies in use that state how actual learning will take place depending on the environment in which they are implemented [14]. They include learning management systems (LMS), content management systems (CMS), learning content management system (LCMS), TV, CD-ROM, virtual worlds, and collaborative technologies. E-learning technologies are the main component of an e-learning platform [14], however there

are some concepts like learning objects and content standards that must be defined to complement and support the definition of e-learning technologies.

In one hand we have learning object (LO), where in programming terminology, an object is a minimal unit or module of a program, standing by itself and including code plus data. In the field of computer programming objects reusability contributes to increasing the software development productivity. So, learning objects are productive in that they are, used as many times as they are need even in other program. Learning objects are any entity, digital or non-digital, which can be used re-used or referenced during technology-supported learning and they include multimedia content, instructional content, learning objectives, instructional software, persons, and organizations or events referenced during technology supported learning [26]. The LO notion has become so popular that the institute of electrical and electronic engineers (IEEE) formed the learning technology standards committee (LTSC) to develop accredited technical standards, recommended practices, and guides for learning technologies, that includes, software components, tools, technologies, and design methods that facilitate development, deployment, maintenance and interoperation of computer-based education and training components.

In the other hand we have content standards where they cover all aspects of e-learning and distance education from representation, packaging, and deployment of learning objects (Figure 4). These standards are important because they establish high-level principles for organizing learning resources for development of e-learning platforms. They also regulate interoperability between applications and improve the reuse of learning objects, in other words, the meeting between learning objects and e-learning standards are shareable with another learning management system [27-25]. There are numerous standards launched to support e-learning environments, like the sharable content object reference model (SCORM), learning object metadata (IEEE LTSC LOM), and IMS Common Cartridge [25].

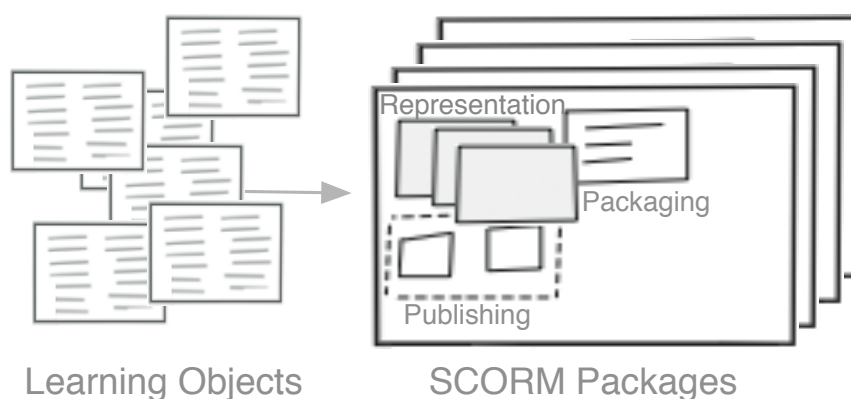


Figure 4 - Learning Objects and SCORM Packages.

SCORM is the most commonly used standard and it was created by advanced distributed learning initiative (ADL) composed by a set of technical standards for e-learning software products. SCORM tells programmers how to write their code so that it can “play well” with other e-learning software [28]. This has a great impact in the reduction of the cost of content integration (Figure 5). SCORM may also be consider as a criterion for e-learning that can increase e-learning capacity by making the content to have reusability, interoperability, accessibility, durability, and affordability functions. SCORM has been developed in version 1.1, the earliest, then updated to version 1.2, and the latest, versions 1.3 know as SCORM 2004.

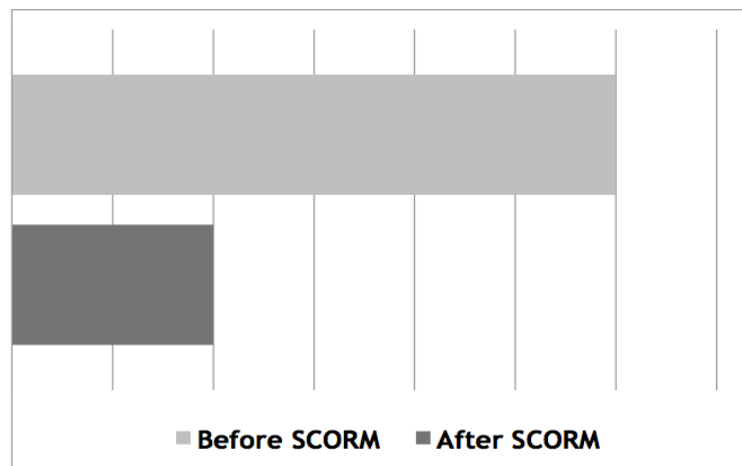


Figure 5 - Cost of Content Integration before and after SCORM.

The construction of a perfect platform is the key objective in our time to provide new ways of content delivery and better ways to provide a better and a more efficient learning experience. An e-learning management platform should be equipped with basic functions of educational administration, education resources, curriculum management, collaborative management, evaluation management, and aids tools [29]. Jun and Xiao [23] consider that there are five levels to conquer the objective of reaching a perfect learning platform. The general communication level where learners and instructors can correspond with each other about the schedule arrangements and contents of e-meetings; the online reference level that consists in the offer of external links; the tests, assessments and surveying level that outlines that learners are able to make self-assessment; the distribution of computer-based training level where learners can download self-needed information to personal computers and receive real-time training, and finally the delivery of multimedia level that describes the communications with the learners are made through videos, animations, and studios. In addition to the levels described above, Jun and Xiao [23] also consider other functions that focus on the way of transmitting knowledge where learners can have a direct access to other learners and more extensive internet databases, optional ways of communication among learners such as chatting and topic discussion and abilities for instructors to give assignments and feedbacks [14]. Generally speaking, a good learning environment involves an authoring

system, an learning management system, and a content management system to create, execute, and store teaching content [25].

The primary learning management system or better computer assisted instruction system was introduced in 1960 by university of Illionois and was called programmed logic for automated teaching operation (PLATO) [30]. However over the years other learning platforms have been developed, such as: Moodle, Blackboard, Sakai, Docebo, SABA LMS, Dokeos, Claroline, ILIAS, OLAT, Joomla LMS, and many others. Some of these LMS platforms are commonly and successfully used in e-learning. They aim at supporting instructors in creating and managing online courses and also provide them with a great variety of features with can be included in the course such as learning material, quizzes, discussion forums, and assignments [31]. So, an LMS is a standard-based system developed to provide online learning services for learners, instructors, tutors, and administrators. These systems are responsible for the management of learning procedures, learning content, and communication with other solutions [32, 14]. Learning management systems play to deliver the desired learning objects when and as need to track the learner's reactions and responses [25]. LMS also provides a collection of tools and functions to support teaching in the learning process, usually including course management tools, online groups chat and discussion, homework collections, grading, and course evaluation [33].

In the other hand CMS like Docebo, and Dokeos aim to supply all the needed tools to assure content of a certain quality and deliver the content in regard to the needs and preferences of the individual consumer, offering division of content, assets management, workflow management, users management, roles, ability to import/export content, content syndication, individualization of content, version management, and content archiving which supports the editorial process [21]. Essentially a content management system provide tools to create content using, or not, a common standard like SCORM. Finally, LCMS outlines an integrated system functionally combining LMS, with CMS. That is, it has the contents development and management functions of a CMS as well as the learning management functions of an LMS. As definition a learning content management system is a platform on witch contents are developed, stored, and managed as learning objects [26].

2.3.1 Open-source E-Learning Platforms

Open-source e-learning platforms are the first choice, since there are visible limitations in the economic and technical support fields. There are many excellent open-source e-learning platforms since open-source software was brought into the field of education [29]. We describe some of these platforms in the following paragraphs.

One of the most influential open source platform is a solution called modular object-oriented dynamic learning environment (Moodle). Moodle is known as a course management system

and sometimes we also know it as a learning management system. This platform provides educators tools to create a course web site and it is used all over the world by more of 400,000 registered users. This platform has features that allow it to scale up very large deployments and hundreds of thousands of learners, yet it can also be used for a primary school or an education hobbyist. Many institutions use it as their platform to conduct fully online courses, while some use it simply to augment face-to-face courses. As mentioned Moodle provides a vast range of features such as: user management, open courses, support to customizable performance scales, forums, wikis, tasks, assignments management, polls/quizzes, and also support for variant types of content [34].

Another platform in the field of open-source is ATutor. ATutor is a open source learning content management system and a social networking environment designed with accessibility and adaptability in mid. ATutor packs a range of features that parse by common management of users, tests and capability of store scores, forums, chats, pools, reading lists, content management, course management, backup management, and learner's profile management [35].

Sakai is an open source learning management system used over one-third of the top 100 universities in the world, colleges, and schools [36]. Sakai is an enterprise-ready collaboration and courseware management platform that provides users with a suite of learning, portfolio, library, and project tools. The Sakai platform is divided into sites, roles, permissions and operations. A site is a navigation destination a place for end users to collaborate with a particular set of users on a particular set of entities. Sites have a selection of tools, each one configured for the site, and a page and tool layout. A role is a set of permissions, and roles exist within the context of a site. Permissions in Sakai are tied to roles, and permissions are the descriptions about a certain role can access a certain object. Finally, operations are actions that describe what you can do [36].

Docebo is an open source learning content management system used in corporate and higher education markets. The platform supports multi-languages, modules, chats, wikis, forums, SCORM 1.2/2004 standards, videoconference, competence modules, customizable reports, business intelligence, and HR software interfaces [34].

Claroline is an open source collaborative e-learning platform. It allows hundreds of organization worldwide like universities, schools, companies, and associations to create/administer courses and collaboration spaces through the web. This solution is used in more than 80 countries and is available in a variant range of languages. This platform is organized following instructor's pedagogical experience and needs. Claroline is organized around the concept of spaces associated to a course or a pedagogical activity, where each course space provides a list of tools enabling the instructor to: write a course description,

publish documents in any format, administer public and private forums, develop, propose assignments to be handed online, see the statistics of users activity, and use wikis to write collaborative documents [34].

ILIAS is an open source web-based learning content management system that supports the management of learning content, including compliance SCORM 2004, collaboration tools, communication, evaluation, assessment tools, course and group management, repository with role-based access control, standards compliance (LOM, SCORM 1.2, SCORM 2004, IMS-QTI, AICC), chats, forums, exercises, tests, podcasting, Google maps support, and several methods of authentication (LDAP, CAS, Radius, SOAP) [34].

Online learning and training (OLAT) is an open source learning management system that has been developed by the University of Zurich, Switzerland. OLAT provides the ability to create e-learning courses, encapsulating collaborative tools to learner's groups, and the ability to execute electronic assessments [37]. OLAT main features are: wide range of supported standards, authoring tools for HTML and IMCS CP, flexible and can be used intuitively, expandability of the system and course configuration (using a flexible management interface).

One last open source platform is Dokeos. Dokeos is one of the largest and most recognized companies dedicated to open source learning management systems. Its main product, called also Dokeos, is a SCORM compliant open source learning suite used by multinational companies, federal administrations and universities. Dokeos main features are: SCORM learning path authoring, templates-based document production, tests, conversion tools, LDAP and OpenID authentication system, gradebook, reservations, user sessions, and multi-language support [34].

2.3.2 Commercial E-Learning Platforms

There is a number of commercial e-learning platforms that are raising the bar in the quantity and quality of provided learning services. One of the most influential is the Blackboard platform developed by the company Blackboard Inc.

BlackBoard is a commercial Web-based learning system used by schools and universities for giving instructions to learners, for online interaction, and for educational assessment. This platform is used by more than 70 percent of the U.S colleges and universities. As of June 2006, the Blackboard empire includes over 12 million users in over 60 countries. Products are offered in 12 languages to over 2,200 learning institutions and contain more than 2,500 supplements from educational publishers [38]. Blackboard is a platform build to enhance teaching methods and the learning process of learners providing a range of features such as: whiteboards, virtual class rooms, tests, assignments management, chat, notifications, user

management, tasks management, content management, staff information, wimba support (wimba is an enabled provider for virtual class rooms), course management, and a content editor.

Joomla LMS is a commercial component for Joomla content management system and is based on a two-layer architecture, the server and the client side. Joomla LMS is a high-end e-learning piece of software to manage online content and administer e-learning processes. Joomla LMS main features passes through the existence of a number of features and tools such as: homework manager, links, learning path tools, SCORM support, attendance management, gradebook, online quizzes, documents repositories, Dropbox and mailbox support, embedded forum, chat and conference, multi-language interface, user roles, and multiples interfaces [39].

SABA LMS consists in a comprehensive solution to address strategic goals of learning organizations including a range of features such as: certification management (supported by wizards), configurable business rules, automatic equivalence management, course and certification versioning, meta-repository, Intelligent support for learning objects, life-cycle existence in the content, simulations tools, blogs, and others collaborative features [40].

2.4 Personal Learning Environments

Indubitably the rapid growth of the Internet has paved the way for e-learning to become an important formal learning facility for educational institutes and enterprises [41]. The existence of a qualitatively different era of the Web, as evidenced by most Web applications and websites that have emerged and transpired over recent years contributed to that. The new/old Web 2.0 era and the technologies beneath it provide both demands and challenges for information technology (IT) educators to reform traditional Web development courses [42]. The rapid advancement of Web technologies has created ever-increasing demands from both learners and employers for instruction in the techniques necessary to develop and manage systems. In this scenario learning management systems, and others, represent today's preferred technologies for learning, unfortunately they show poor adaptability to each users needs and a poor aggregation of web 2.0 services. Personal learning environment (PLE) are the troops to win this battle, merging a wide range of web 2.0 services and actuality answering the needs of users. Gradually, PLEs are gaining ground over traditional LMS, because they are a way to overcome the common LMS limitations such as the inability of fostering a lifelong learning perspective, taking into account individual user needs [43-44].

Personal learning environments (PLEs) originally surfaced in 1998 when media lab in Helsinki released the first future learning environment (FLE) a web-based learning environment designed to support learner and group centered work that concentrates on creating and developing expressions of knowledge. This implementation has established the definition and

concept of personal learning environment. As shown in Figure 6 a PLE can be defined as a new approach of reusing existent technologies where there is an association of existing sub-concepts like lifelong learning, collaborative study, capacity to support heterogeneous sources and informal learning, social learning, and a number of learning styles. The main idea is that the institution does not always need to choose the interface and tools for the learner instead they are the personal choice of the learner [45]. Harmelen [46] consider that a PLE is a type of e-learning system, that is structured on a model of e-learning itself rather than a model of the institution, and that PLEs are concerned with the coordination of the connections made by the learner with units and agents across a wide range of systems.

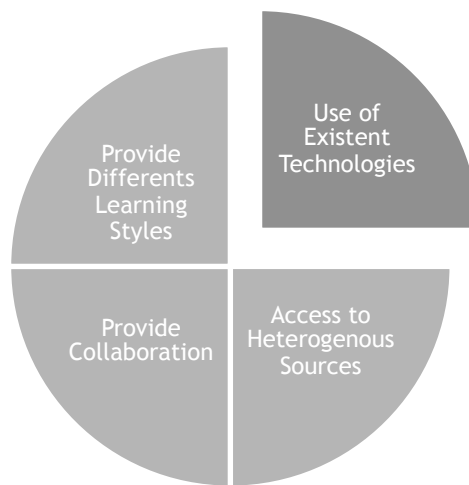


Figure 6 - Illustration of the Personal Learning Environment concepts.

Other authors like Mikroyannidis [43], Henri and Charlier [47], agree that a PLE is not a software application but a concept where individual learners can configure the group of learning resources and applications in such a way that they achieve the best outcome for their personal learning goals and knowledge management. Learners can also monitor their learning progress, with different widgets, and integrate both personal and professional interests within the PLE. According to Cerbo and Doderio [44] a PLE can also be viewed as a portfolio system, allowing the owner to maintain her repository of contents, and selectively share it as needed, exposing the owner's interests and allowing automated selective search of digital contributions. There are numerous platforms based on these concepts, existing a clear fusion between content management systems, learning management systems and personal learning environments. In the following paragraphs are presented some of theses platforms.

DIEL is a LMS/PLE platform, built on top of Moodle. The main objective of DIEL is to enhance the user's e-learning experience with a highly interactive, appealing and flexible 3D graphical environment, which exploits the concept of social translucence [44]. DIEL implements a metaphor based on rooms, containing objects that represent learning resources and where

user avatars may approach an object in order to exploit it in order to trigger a corresponding action [44].

Colloquia is a PLE platform mediated by an email server, that provides support for a conversational and activity based model of learning. Colloquia maintains information about people, resources, and tasks. Instructors set up activities and sub-activities at different levels of granularity and allocate people, resources, and tasks to those activities. There is a distributed locus of control in that learners who are assigned to activities may create sub-activities. Communication between users happens in the context of different activities, so there are special conversation types for different patterns of interaction between learners and instructors [46].

The Manchester PLE Framework constructed was a spin-off PLE development from Bodington VLE. The initial starting point was developed and extended by a team at the University of Manchester [46]. This framework could be instantiated as a PLE, in this scheme a PLE is considered a desktop or a laptop based client that is capable of both networked and stand-alone use [48].

PLEXus is a personal learning environment based on the semantic technology of topic maps. A semantic-based navigation in e-learning will enable variation, differentiation and individualization, which are important pedagogical factors in the development of a PLE. This platform provides a learner interface allowing customized views of learning objects and learning activities [49].

Other traditional platforms like Blackboard are slowly evolving into the field of personalization and contextualization, in the aspect of the user and not the hosting institution, retaining the importance of shifting the learning aspect to the learner and his own way to learn and improve his success. However, despite the emergence of a wide range of new and refreshing concepts associated with personal learning environments and attempts to evolve previous platforms, there is also a set of issues that we need to identify. Motivational aspects of learners and learning communities using PLEs need to be understood more comprehensively and supported accordingly if the PLEs are meant to cater for learning methods and needs; the social software tools inherent support of learning communities can be fruitful, but if not carefully implemented in a formal learning context the individual aspect of learning is lost; it also must be understood that choosing a distributed or an integrated PLE solution has different implications; control exists in the institution and not in the learner, so is needed to shift the control from the institution to the learner; learners' inability of combining formal and informal learning knowledge and existence of issues in the integration of applications, services, and solutions [50]. As shown in Figure 7 a set of guidelines

were proposed to fight these issues, with the objective of developing an enabled personal learning environment system [49].

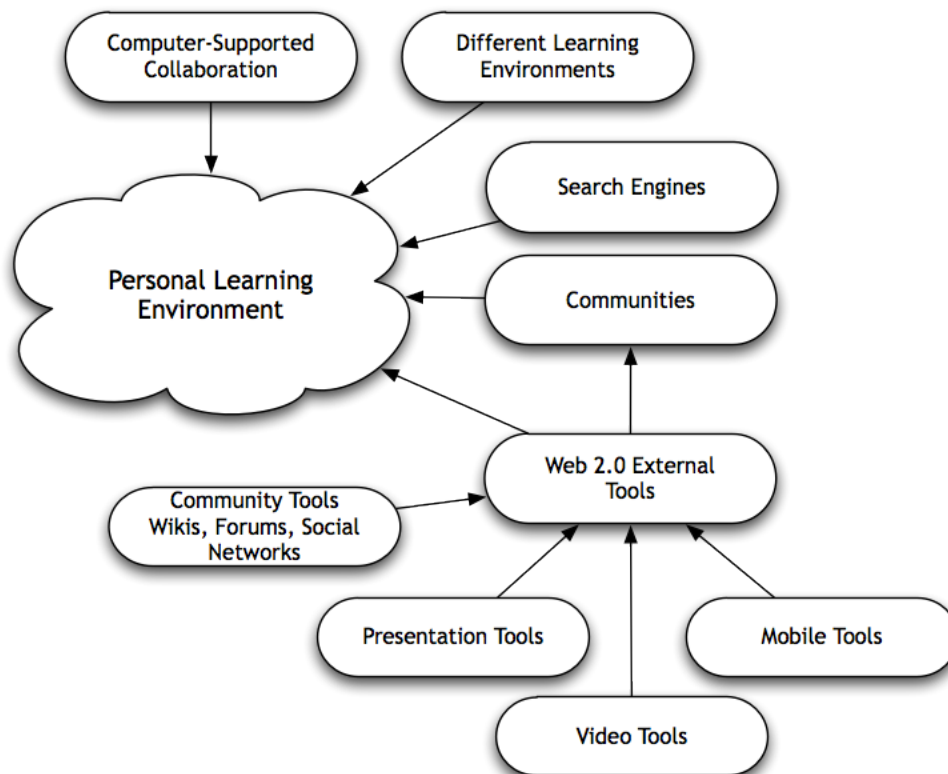


Figure 7 - Guidelines for an Efficient Personal Learning Environment.

The computer-supported collaboration guideline details various forms of cooperation such as document sharing, different media elements for various means, and a number of dialogues using different synchronous and asynchronous ways. The different learning environments guideline outlines that each environment stores his dialogues and messages necessary for further learning reflection by the learner. In addition, providing a number of configurations options to allow a control of individual components of the learning environment. The capacity to offer material guideline details the option to offer material for others to review, resume, link materials, share, study, and get feedback from peers [48]. The support to different communities guidelines state that each learner can join more than one community and contributing in their own way, using a number of Web 2.0 services, where each community has a number of members (learners), who have the same or similar interest. The search engines guideline details that learners can search learning material on the Internet or in the internal repository of the platform, however this search need to be as smart as possibility to adapt the searched material to the needs and profile of the current learner [51]. Furthermore, adding a suggestion engine provides a number of hints, taking into account the user's profile. Additionally to these guidelines an innovative and friendly user interface must be considered in order to allow customized views of learning objects, learning activities, and

different tools inside the environment. Employing well-know user interfaces patterns is necessary to reduce the learning curve of how to use the solution [49].

2.5 Rich Internet Applications

In desktop context, operating system (OS) is an important subject, involving many concepts and mechanisms, which are dynamic and abstract. In such a subject, providing the abstraction and design paradigms to enable users to fully understand the theoretical principles involved is difficult. The study of operating systems involves gaining an understanding of both the responsibilities of an operating system, and the algorithm and data structures used in implementing these responsibilities. The graphic user interface (GUI) presented in the OS, as we know it today, incorporates several ideas about how human beings should interact with computers. One is that icons (pictures) are easier to understand than plain text and that the window component of GUI, allows the user to have different parts of the screen devoted to different activities [52].

In a Web context the complexity of tasks performed through Web applications is increasing, in particular when high levels of interaction client-side processing, and multimedia capacities have to be provided. In this context traditional HTTP-HTML Web applications are showing their limits and developers are building the future of the Web using rich internet application (RIA) technologies, defined as Web applications with many additional features. RIA offer online and offline capabilities, sophisticated user interfaces, and the possibility to store and process data directly on the client side [53-54].

As shown in Figure 8, RIA applications are using a set of concepts and ideas that are present in multiple environments, like applications that allows concepts that improve the user experience and increase the familiarity of how to use the Web system. However, implementing this kind of ideas, in effective user interfaces is difficult, as they must cleverly combine hypermedia-like interface of “conventional” Web software, with the kind of interface functionality we usually find in desktop applications (with drag and drop, popup-information and diverse interface effects). To make matters worse, these applications must also deal with a myriad of concerns which comprise multiple requirements both functional and non-functional and which usually crosscut each other [55]. The permanent beta state of rich internet applications complicate things further: new interface widgets or interaction style are constantly introduced, checked to asses users acceptance, and then either becoming core components or eliminated.

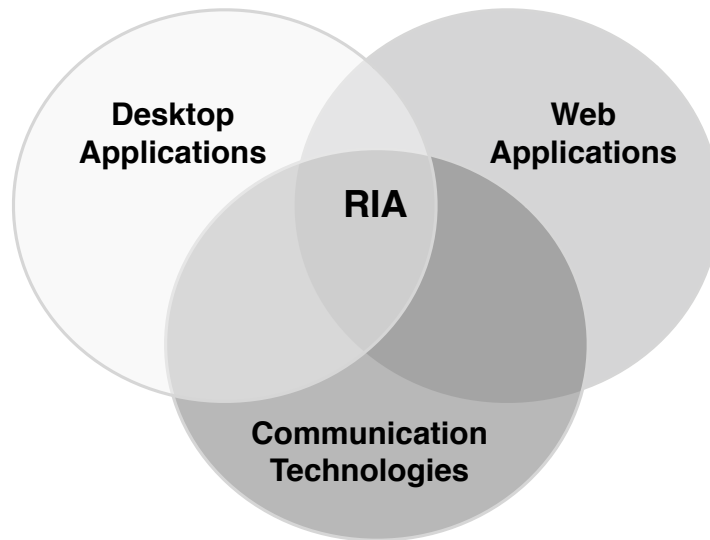


Figure 8 - Illustration of Rich Internet Applications concepts.

There are countless frameworks, APIs, and SDKs that allow the development of rich internet applications, in the following paragraphs we present some of these frameworks, APIs, or SDKs.

Tersus is an open source visual programming language that has emerged from real life needs to become a general-purpose software development platform with the aim of building rich web applications. This solution provides a unified language for modeling user interfaces, client-side behavior and server-side processing, is mainly used by individuals and enterprises to develop web applications as well as cross-device mobile applications [56].

FLEX is a open source application framework that allow development of mobile applications for iOS, Android, BlackBerry, as well traditional applications for browser and desktop using the same programming model, tool, and codebase [57].

ASP.NET AJAX Control Toolkit is an open source project built on top of the Microsoft ASP.NET AJAX framework. It is a joint effort between Microsoft and the ASP.NET AJAX community that provides a powerful infrastructure to write reusable, customizable and extensible ASP.NET AJAX extenders and controls, as well a rich array set of controls that can be used of the box to create an interactive and rich web experience. The AJAX Control Toolkit contains more than 30 controls allowing the development of easily create rich, interactive web pages [58].

Dojo toolkit is an open source modular framework designed to ease the development of cross-platforms and provide rich enabled user interface environments, the Dojo toolkit also provides a set of grids, charts and other features, only possible due to its powerful APIs [59].

Chapter 3 - Requirements Analysis

This chapter describes the requirements analysis, also called requirements engineering consisting in the process of determining user expectations for a new or modified product. This process of requirements analysis was started before the development of version 1.0 and reanalyzed in the pre-development of version 2.0. In Figure 9 is showed the development model adopted for this project.

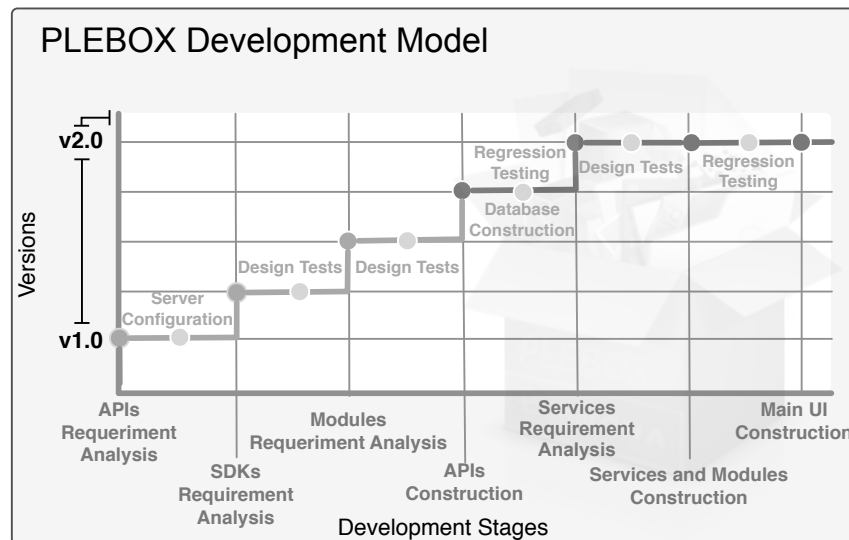


Figure 9 - Adopted Development Model.

3.1 User Cases Diagrams

Uses cases diagrams is a methodology used to identify, clarify and organize system requirements, composed by a set of possible interactions sequences between systems and users in a particular environment. In the following sections we present the main user case diagram and its sub-diagrams, additionally, we also present the main class diagram of each particular section.

3.1.1 APIs/Services and SDKs

The major components, either in version 1.0 or version 2.0 of the platform, are the APIs, services, and SDKs that enables the possibility to create components (modules or features). Therefore, developers have at their disposal a set of methods delivered in the form of APIs, services and SDKs, allowing the management of various aspects of the platform. Should be noted that besides the existence of learning APIs/services there is also a set of non-learning APIs/Services that are intended to administer the platform. The main user case diagram is shown in Figure 10. Basically this diagram reflects the interaction between the platform, databases and developers with a set of features, services, APIs and SDKs. In the following paragraphs we describe each one of the users cases that are encapsulated into the main diagram.

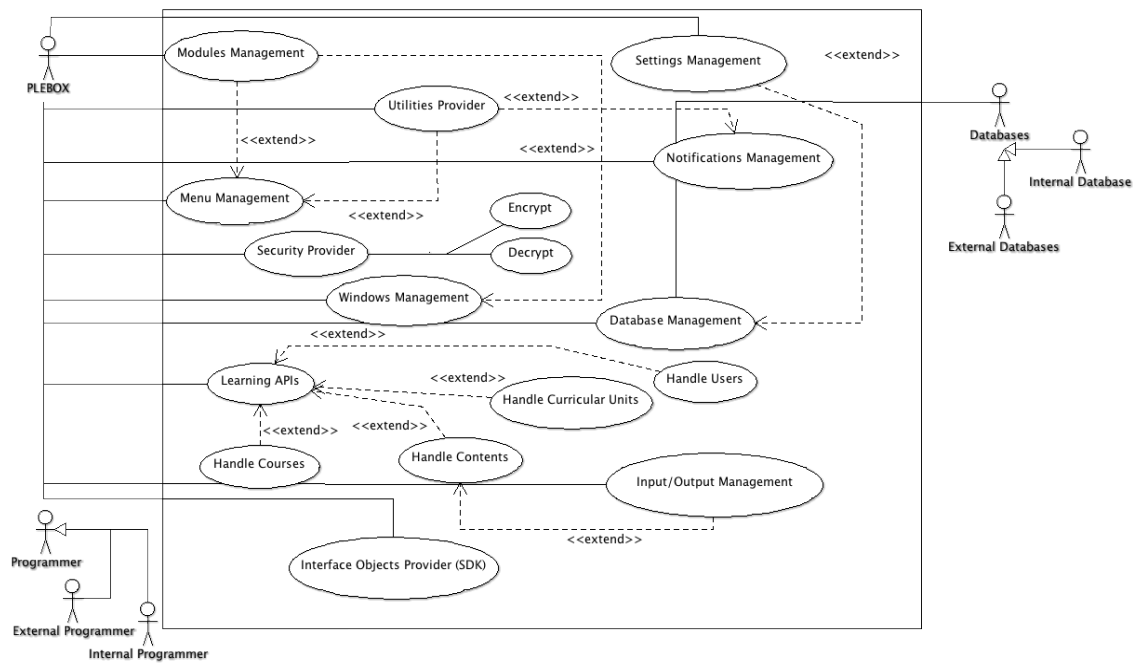


Figure 10 - Main User Case diagram.

The modules management user case diagram (Figure 11) describes a service that enables the installation of modules into the platform, manage permissions, manage icons and manage ways to initialize modules. The menu management user case diagram (Figure 12) describes a feature that is inside the user interface to retrieve a list of content according to the current user's permissions. The windows management user case diagram (Figure 13) aims to describe the service to manage users' desktop, incorporating the ability to register and manage windows common characteristics. The database management user case diagram (Figure 14) describes an internal API to manage common operations to be executed into a relational database management system.

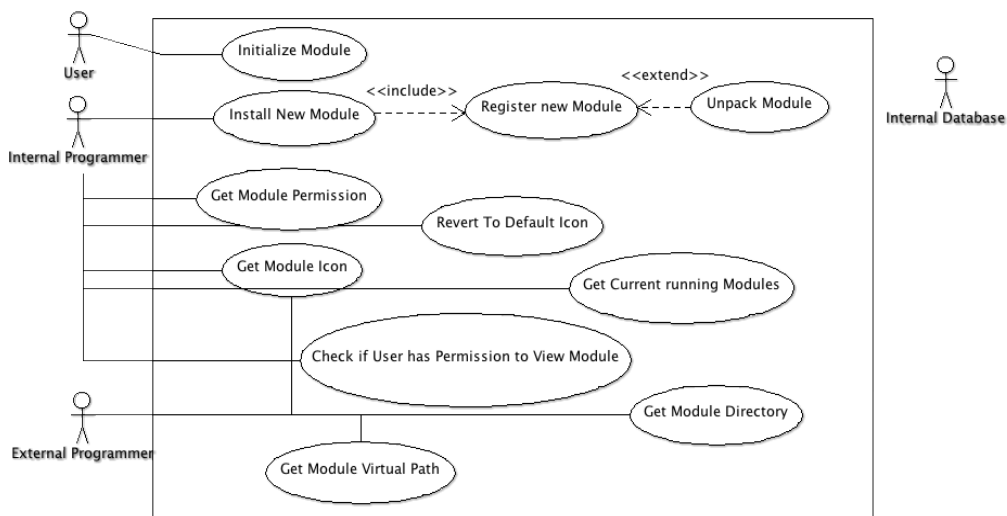


Figure 11 - Modules Management User Case Diagram.

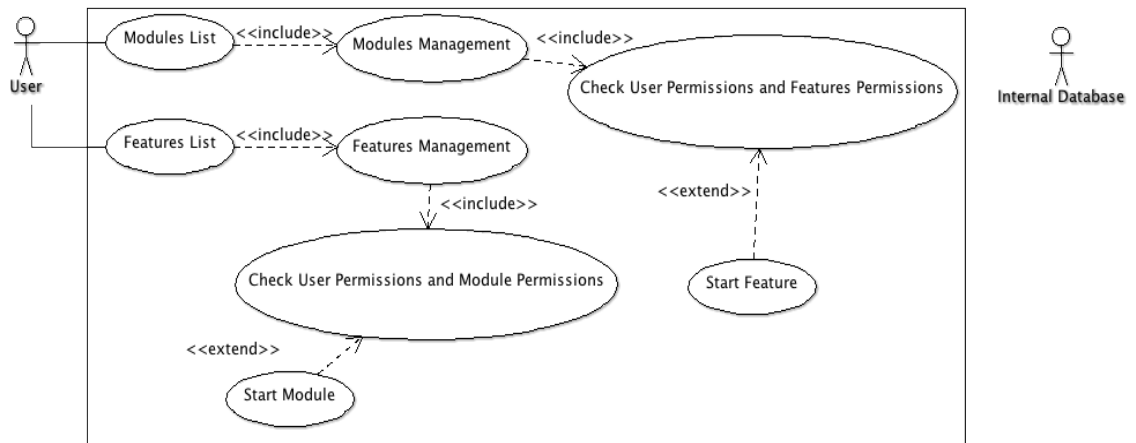


Figure 12 - Menu Management User Case Diagram.

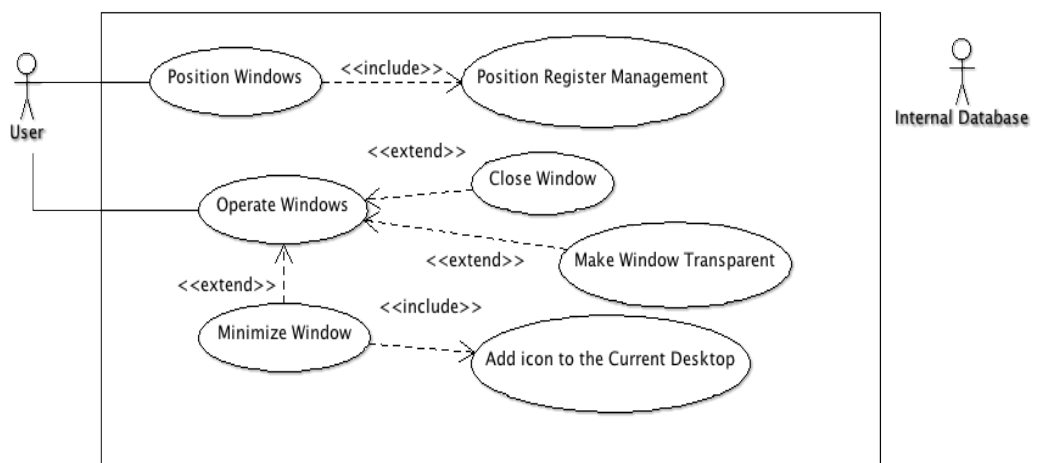


Figure 13 - Windows Management User Case Diagram.

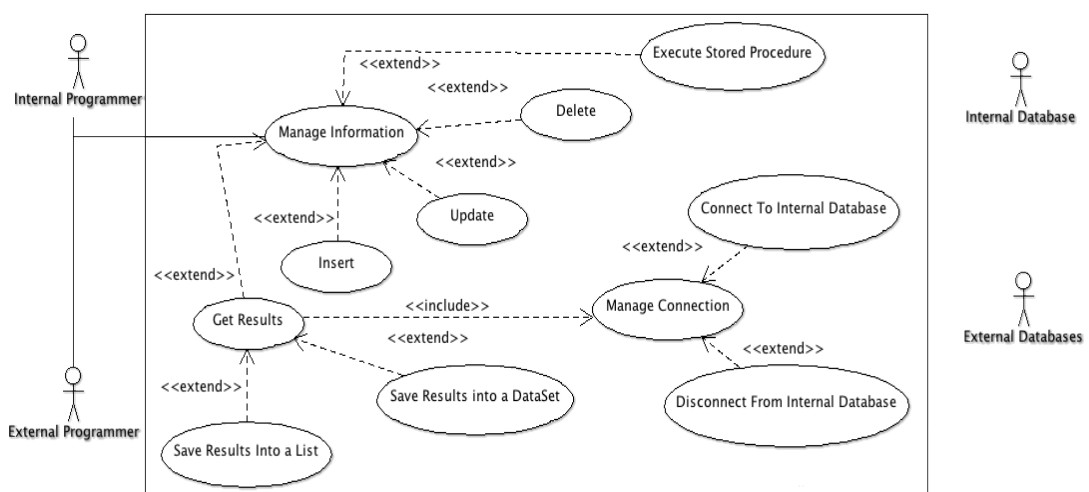


Figure 14 - Database Management User Case Diagram.

The notifications management user case diagram (Figure 15) describes the external service to manage notifications, which includes the ability to send notifications to users (whether they be instructors or learners) to groups and also to curricular units. The interface objects provider user case diagram (Figure 16) describes the PLEBOX SDK, which has a set of interface objects that can be used to expand or build new contents for the box. The settings management user case diagram (Figure 17) describes an internal API for configuring versions, groups, and databases to be used by the platform.

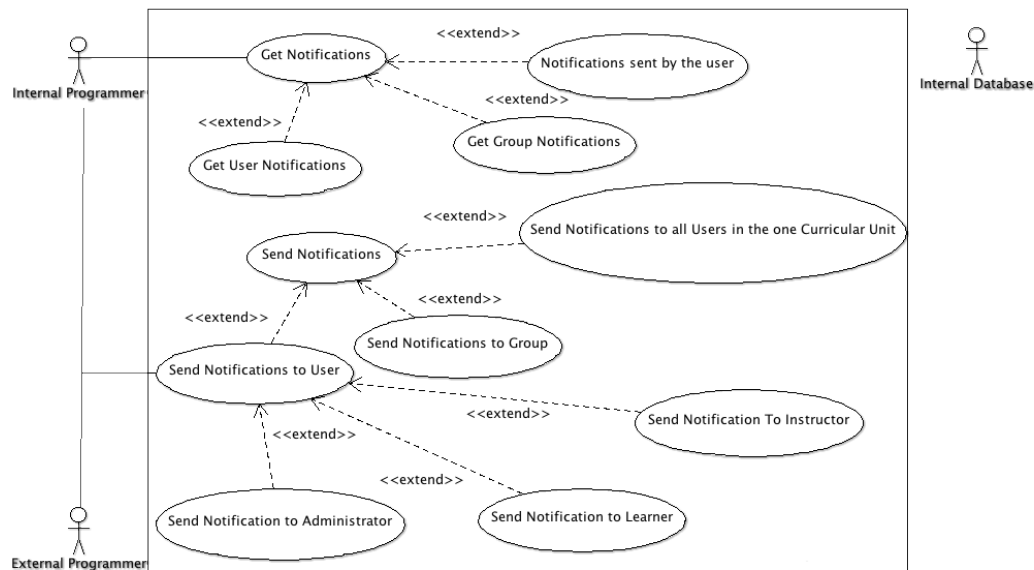


Figure 15 - Notifications Management User Case Diagram.

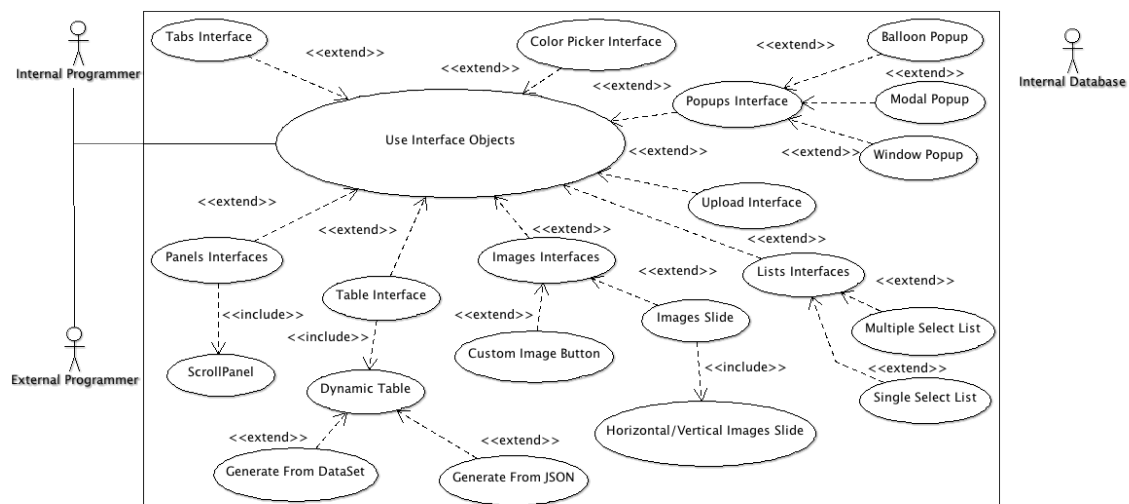


Figure 16 - Interface Objects Provider User Case Diagram.

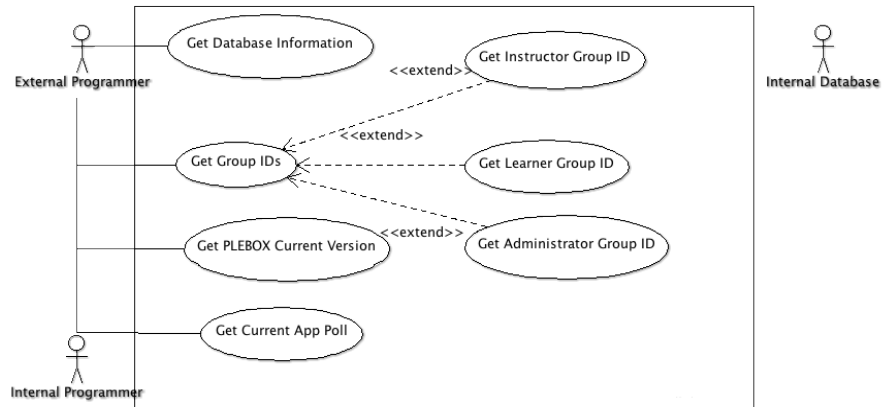


Figure 17 - Settings Management User Case Diagram.

The utilities provider user case diagram (Figure 18) describes an internal API that provides a useful set of development utilities. The input/output management user case diagram (Figure 19) describes an internal API that provides a set of operations to manage files and directories that are present in a server. The handler users user case diagram (Figure 20) describes a service that allows the management of users, including associations operations between users, courses and curricular units. Additionally, this diagram details a set of verification operations for associations and permissions.

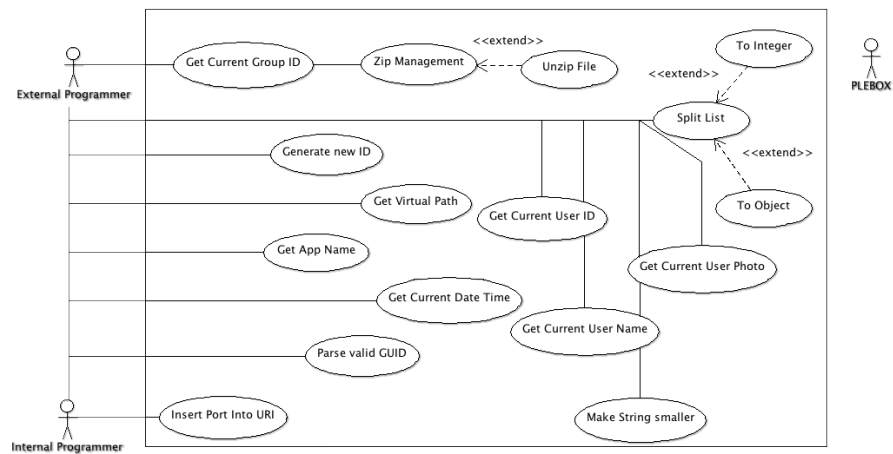


Figure 18 - Utilities Provider User Case Diagram.

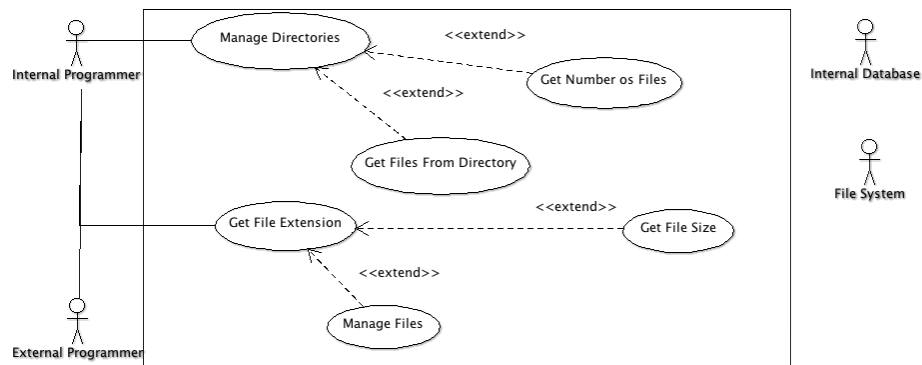


Figure 19 - Input/Output Management User Case Diagram.



Figure 20 - Handle Users User Case Diagram.

The handle courses user case diagram (Figure 21) describes a service to manage courses and their respective characteristics. This management includes the ability to know which users are associated with a particular course, the ability to manage different types of courses and also the ability to obtain information about a particular course, for example knowing which curricular units is associated with one course. The handle curricular units user case diagram (Figure 22) describes a service to manage curricular units, including the ability to obtain association information between users and curricular units. The handle repositories user case diagram (Figure 23) details a service for managing the register of local and remote repositories. The handle contents user case diagram (Figure 24) outlines a service for content management and availability. In addition, this diagram outlines the management of standards, types and the management of assignments that allows the possibility to assign content to users or curricular units.

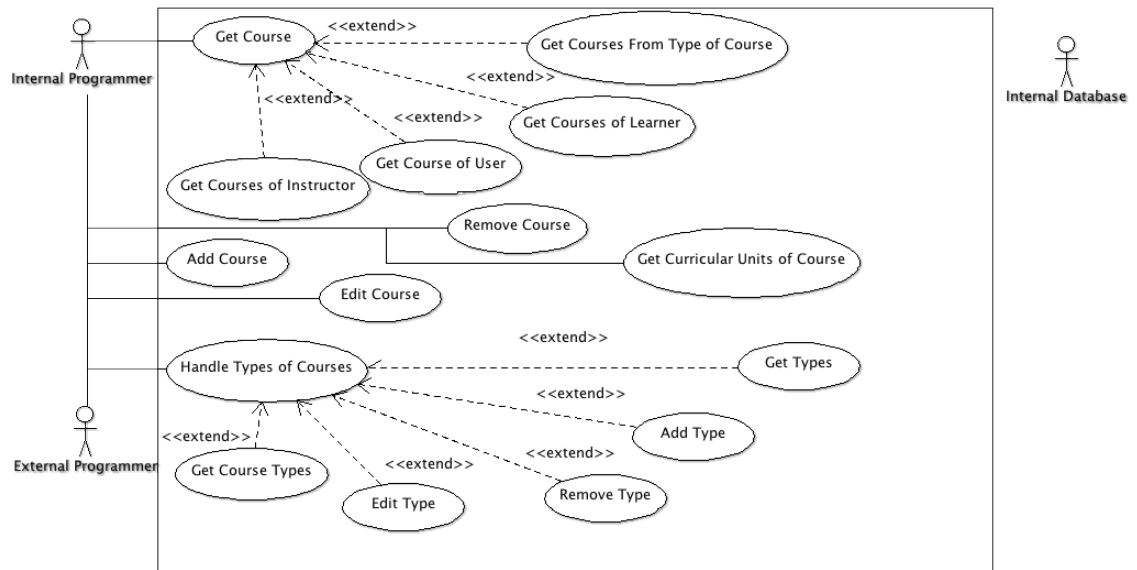


Figure 21 - Handle Courses User Case Diagram.

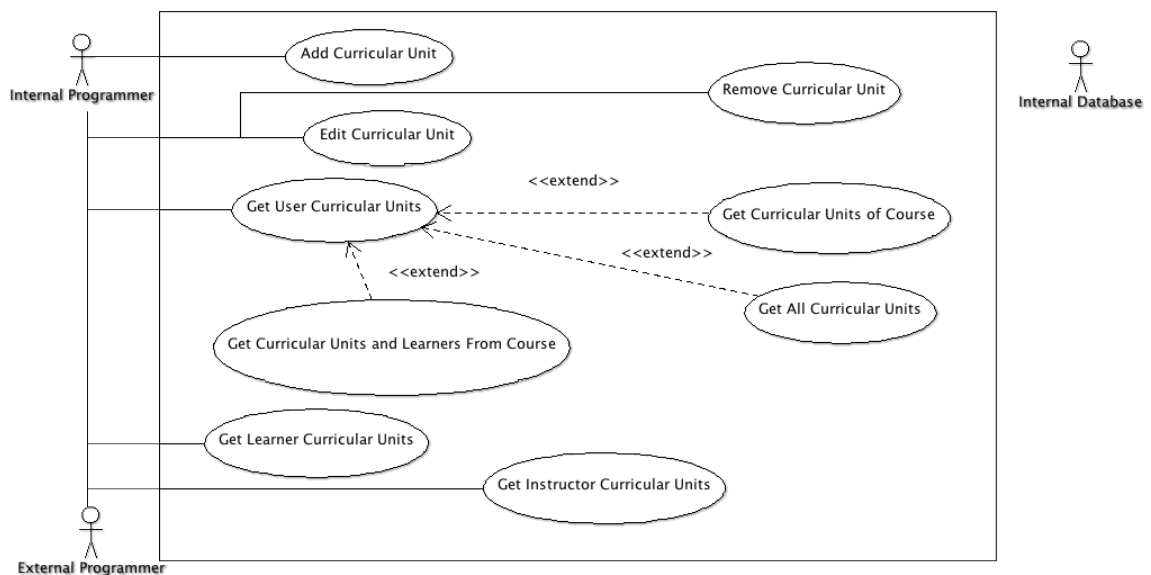


Figure 22 - Handle Curricular Units User Case Diagram.

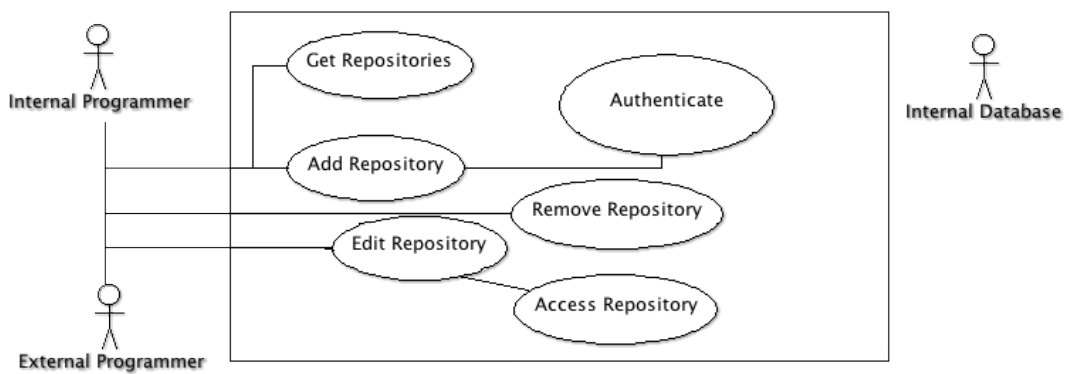


Figure 23 - Handle Repositories User Case Diagram.

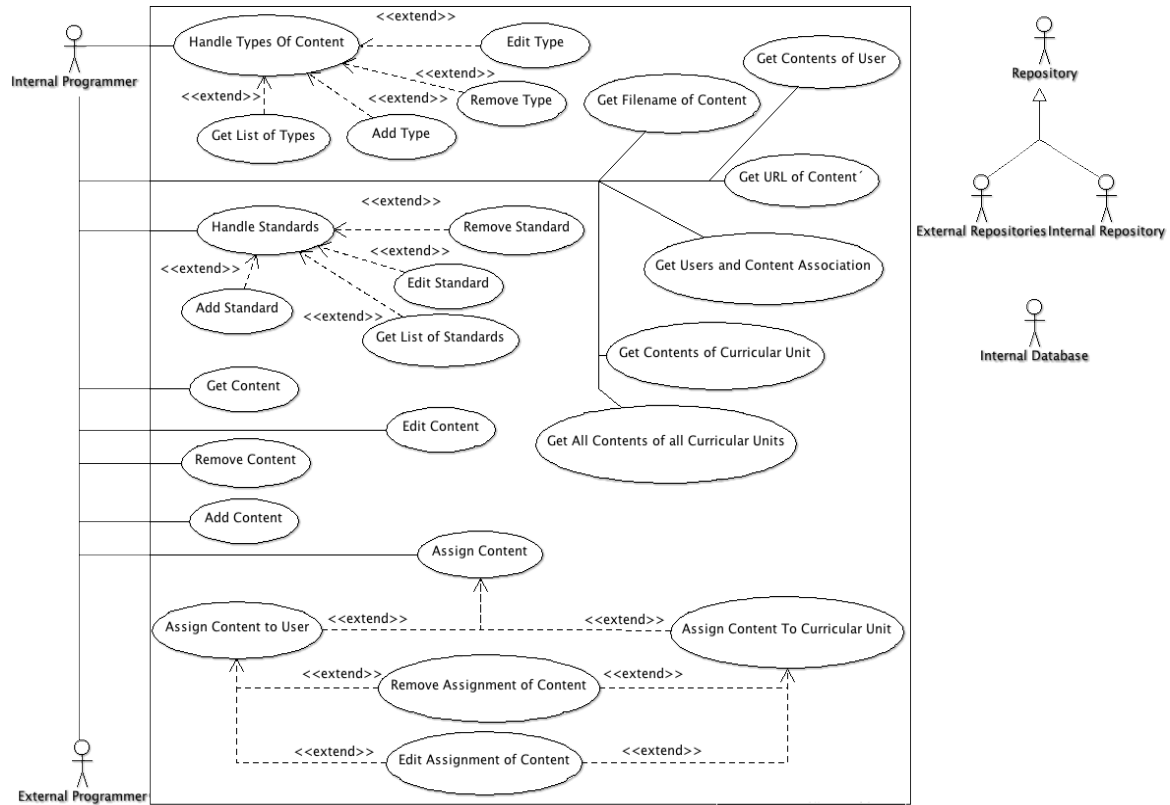


Figure 24 - Handles Contents User Case Diagram.

3.1.2 Non-Learning Modules and Widgets

Non-Learning Modules adds content management and simple interactions to the platform. We analyzed the requirements for two modules and one widget: the manage modules user case diagram details an easy way to install modules into the box (Figure 25), the manage notification user case diagram (Figure 26) outlines the management of notifications and the profile widget user case diagram details the possibility to edit information related to the current user profile (Figure 27).

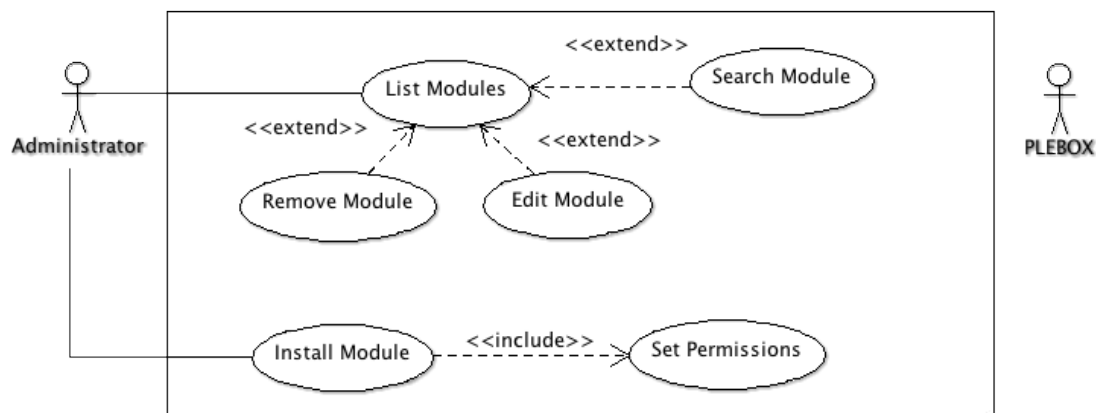


Figure 25 - Manage Modules User Case Diagram.

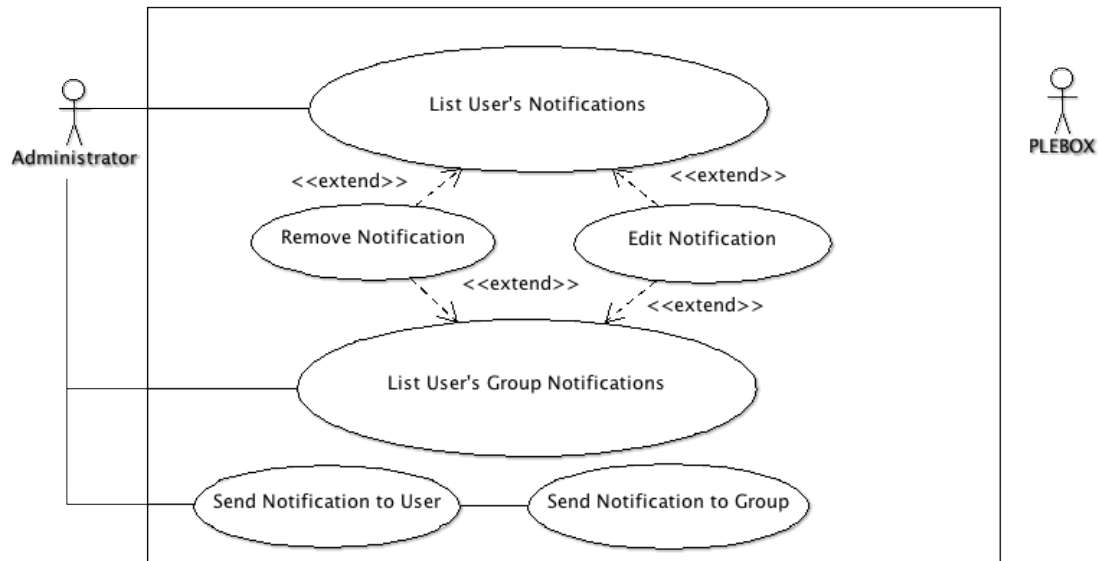


Figure 26 - Manage Notifications User Case Diagram.

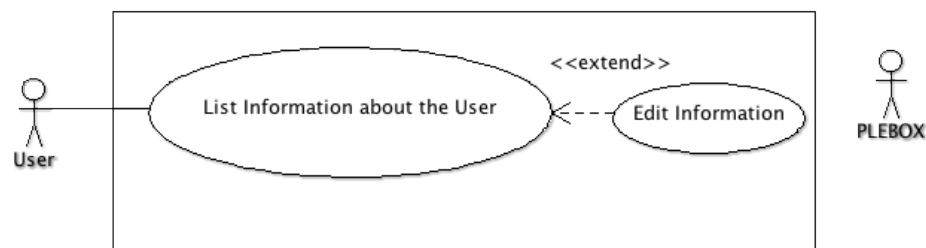


Figure 27 - Profile User Case Diagram.

3.1.3 Learning Modules

The learning modules are a set of user interfaces that aggregate the functionalities presented in the learning services. All these modules were designed as demonstration modules with the aim of showing the learning services/APIs and SDKs capabilities. The main user case diagram is shown in Figure 28. This diagram details the mode of operation of all learning modules and its relationships with different groups of users. In the following paragraphs we describe each user case encapsulated into the main diagram, representing different modules.

The manage content user case diagram (Figure 29) details a module aimed to be used by administrators, where is possible to manage the entire set of content. This diagram outlines the possibility to manage content (list, add, edit, remove) and also to manage associations between content, users, groups and curricular units. The manage content standards user case diagram (Figure 30) details a module to be used by administrators, describing the usage of a set of services to manage content standards (list, add, edit, and remove).

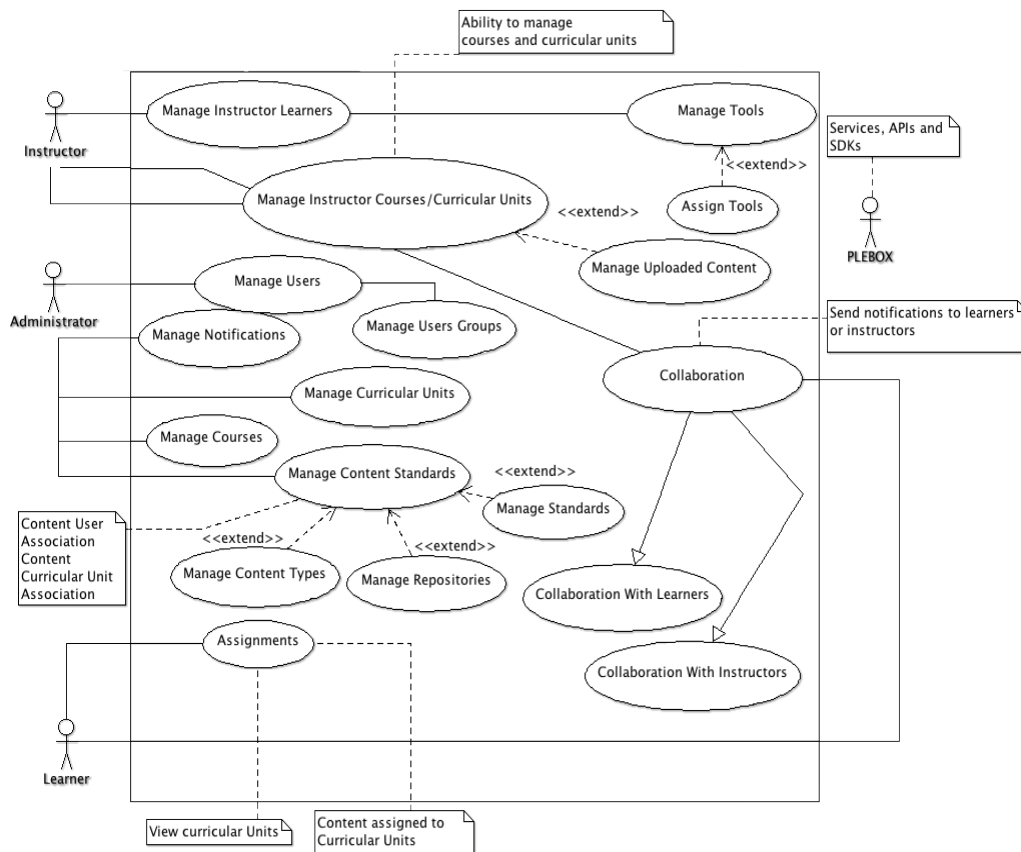


Figure 28 - Modules User Case Main Diagram.

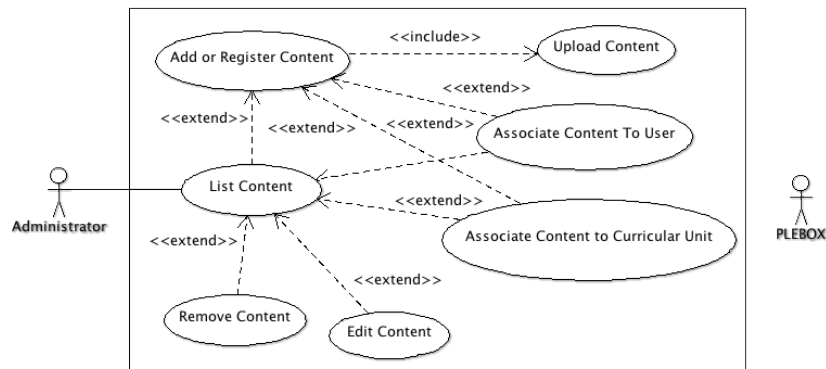


Figure 29 - Manage Content User Case Diagram.

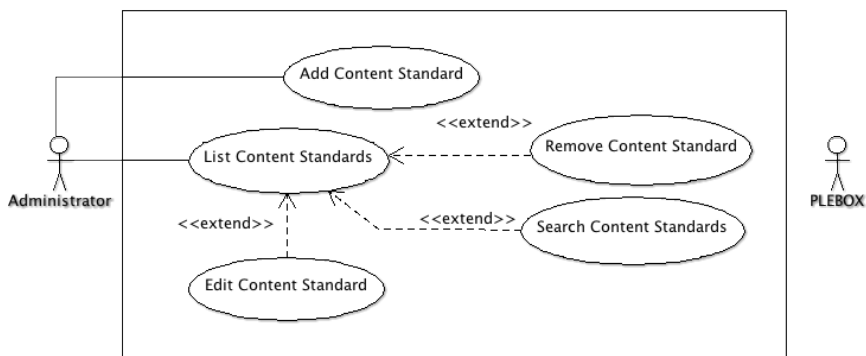


Figure 30 - Manage Content Standards User Case Diagram.

The manage content types user case diagram (Figure 31) details a module to be used by administrators, describing the use of a set of services to manage (list, add, edit, and remove) types of content. The manage courses user case diagram (Figure 32) details a module to be used by administrators in order to manage courses (list, search, add, edit, and remove), including features to manage associations between types and courses. The manage curricular units user case diagram (Figure 33) outlines a module to be used by administrators in order to manage curricular units (list, search, add, edit, and remove), including an association management to assign curricular units to courses.

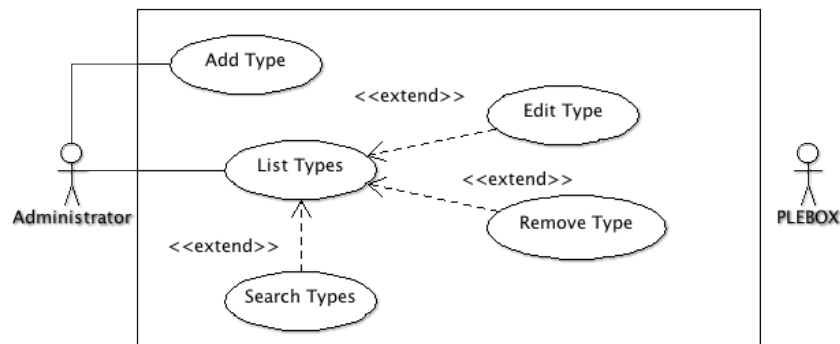


Figure 31 - Manage Content Types User Case Diagram.

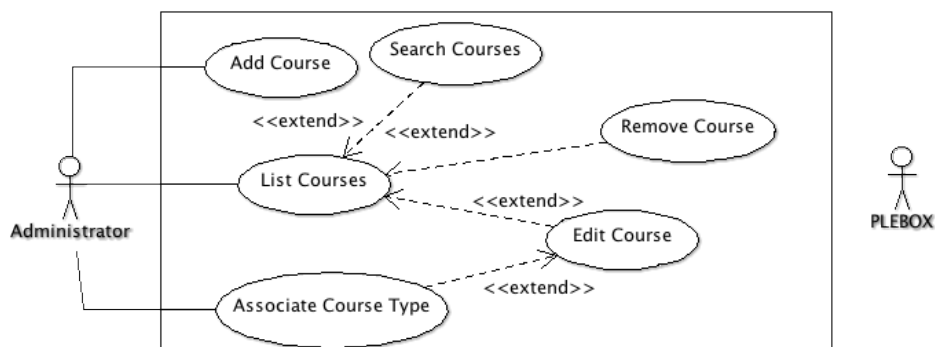


Figure 32 - Manage Courses User Case Diagram.

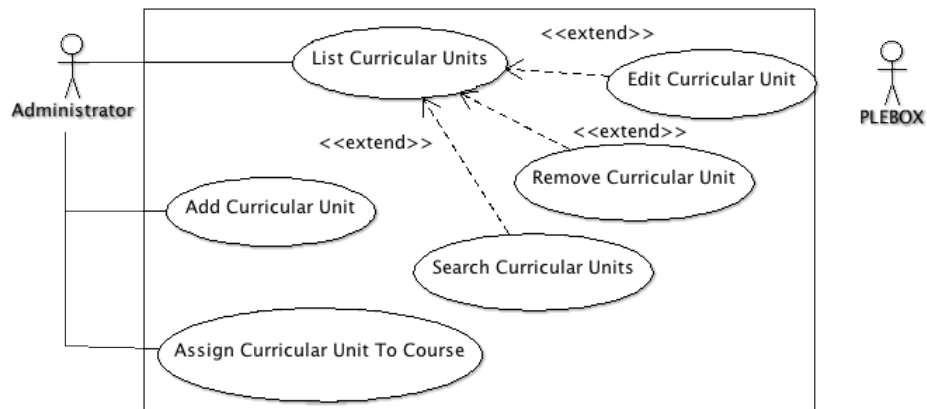


Figure 33 - Manage Curricular Units User Case Diagram.

The manage courses and curricular units user case diagram (Figure 34) details a module to be used by instructors where it is possible to management content related with his courses and respective curricular units. This management details that an instructor has the ability to add content to a set of curricular units. The manage learners user case diagram (Figure 35) details a module to be used by instructors with the aim of accessing and list, information about his learners.

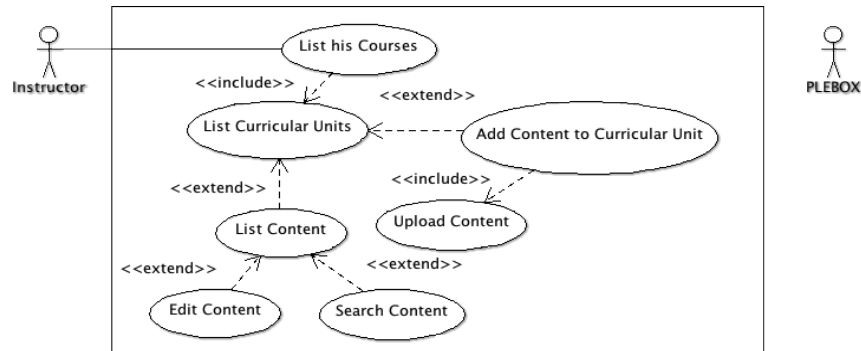


Figure 34 - Manage Courses and Curricular Units User Case Diagram.

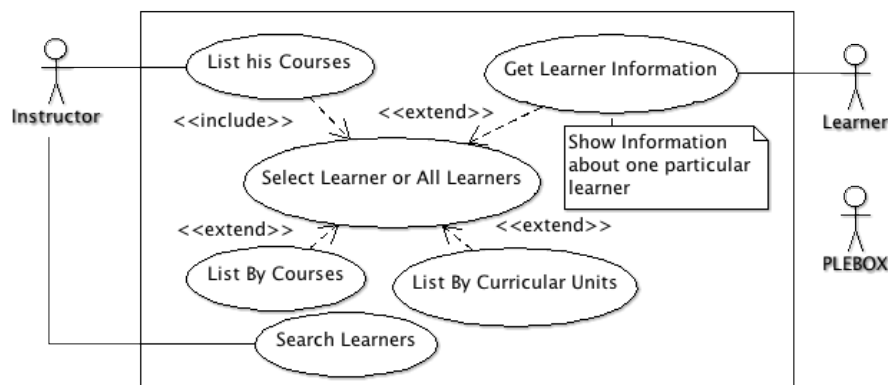


Figure 35 - Manage Learners User Case Diagram.

The manage notifications user case diagram (Figure 36) outlines a module to be used by administrators where it is possible to manage notifications and different ways to send notifications to groups or users.

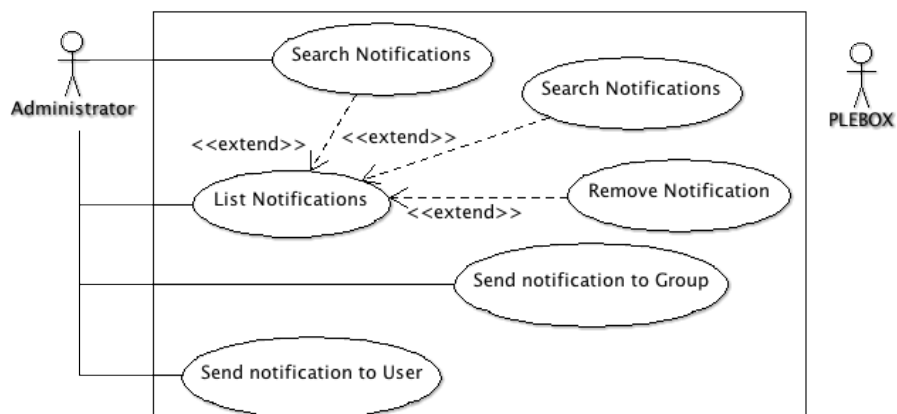


Figure 36 - Manage Notifications User Case Diagram.

The manage repositories user case diagram (Figure 37) details a module to be used by administrators, describing a set of operations to manage internal and external repositories. The manage users groups diagram (Figure 38) details a module to be used by administrators where it is possible to manage (list, search, add, edit, and remove) users groups (administrators, learners, or instructors).

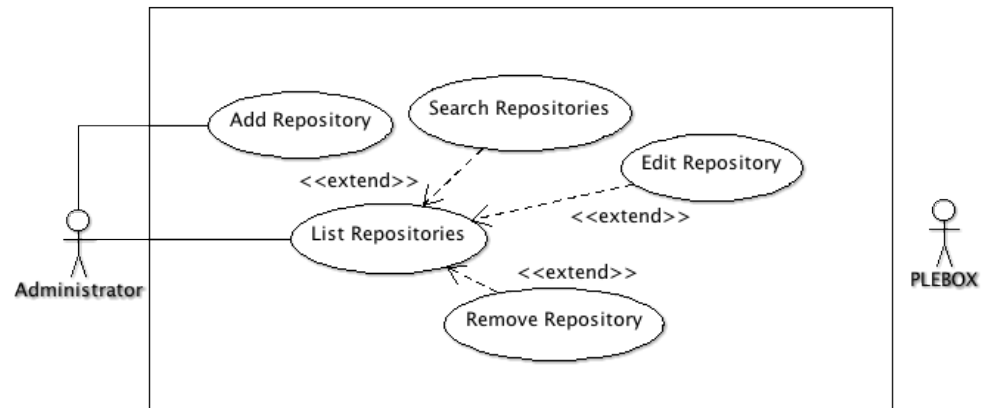


Figure 37 - Manage Repositories User Case Diagram.

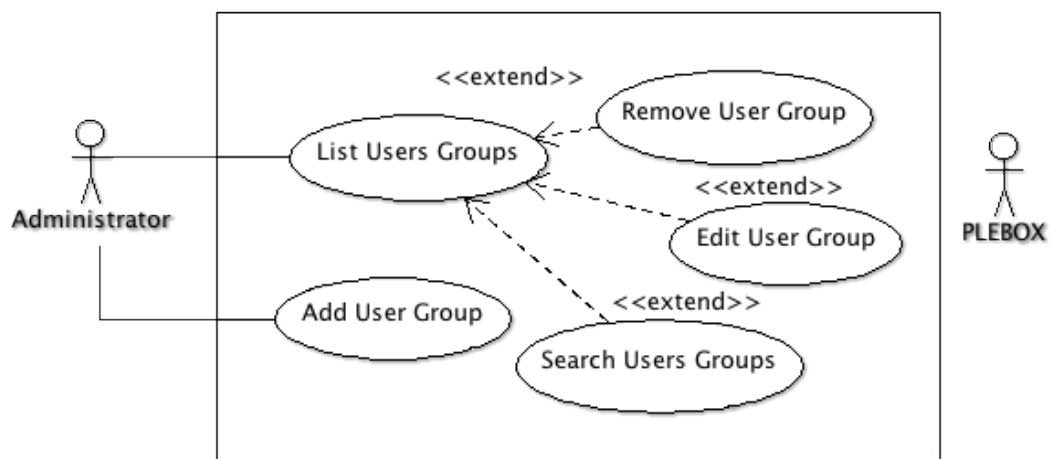


Figure 38 - Manage Users Groups Module User Case Diagram.

The manage users user case diagram (Figure 39) details a complex module to be used by administrators where it is possible to manage a set of users, including the management of any possible associations between users, courses, and curricular units. The collaboration with learners user case diagram (Figure 40) and the collaboration with instructors user case diagram (Figure 41) details a set of two modules to be used by learners and instructors. Both diagrams detail ways to receive and send notifications to learners or instructors of a particular curricular unit. The assignments user case diagram (Figure 42) outlines a module to be used by learners, describing the usage of a set of services to manage contents (execute, view, and resume).

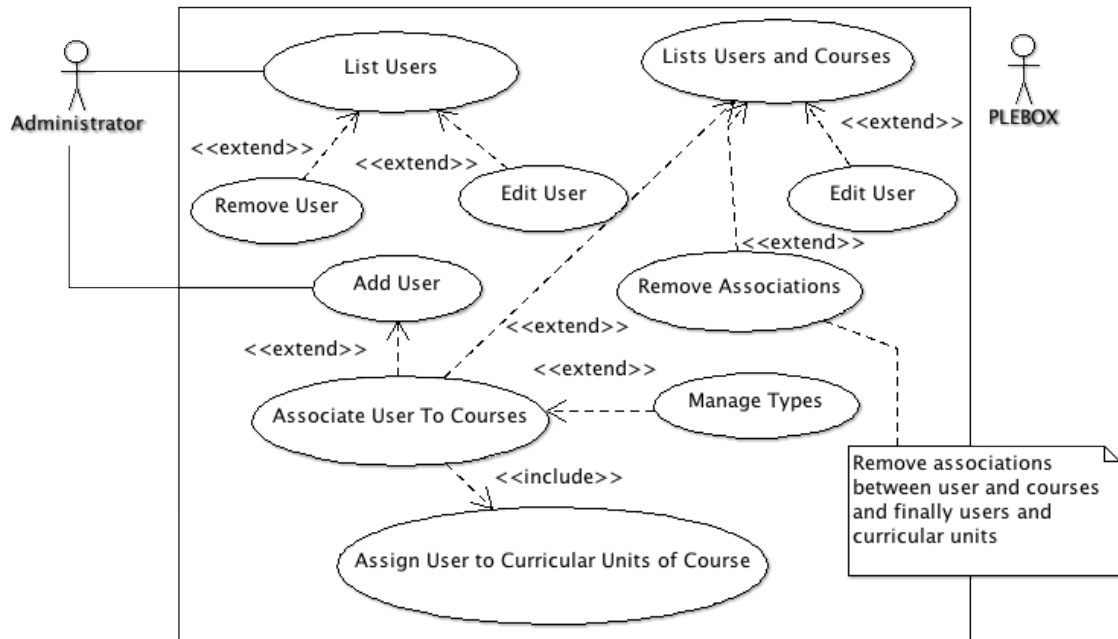


Figure 39 - Manage Users Module User Case Diagram.

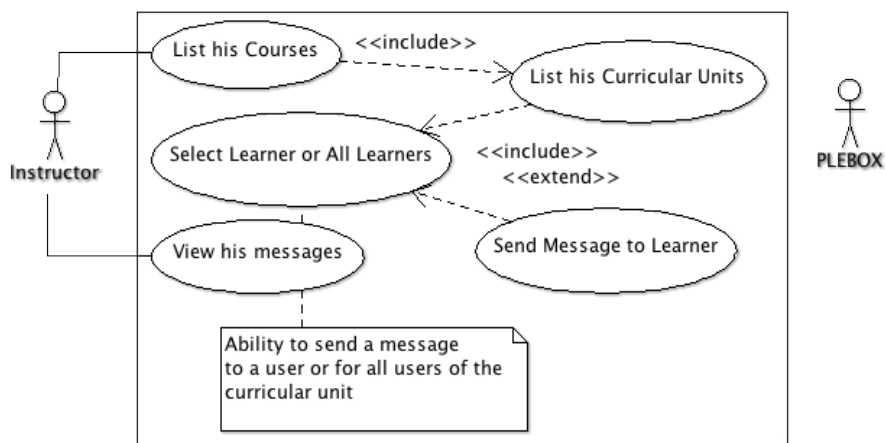


Figure 40 - Collaboration With Learners User Case Diagram.

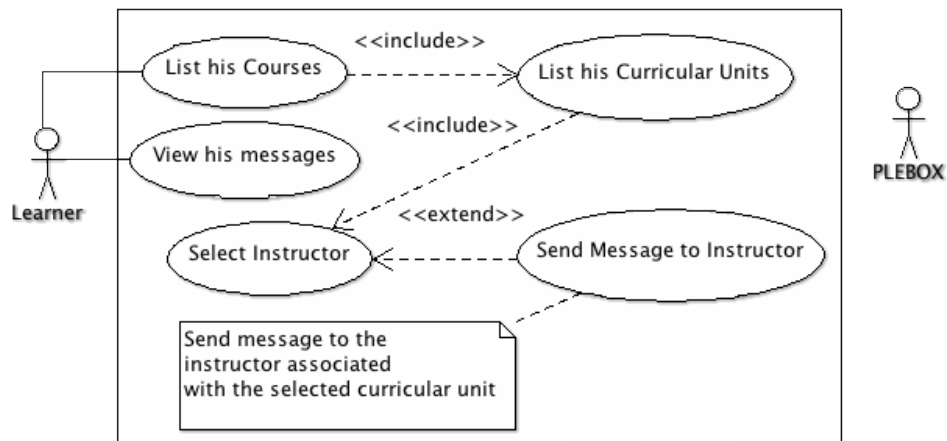


Figure 41 - Collaboration With Instructors User Case Diagram.

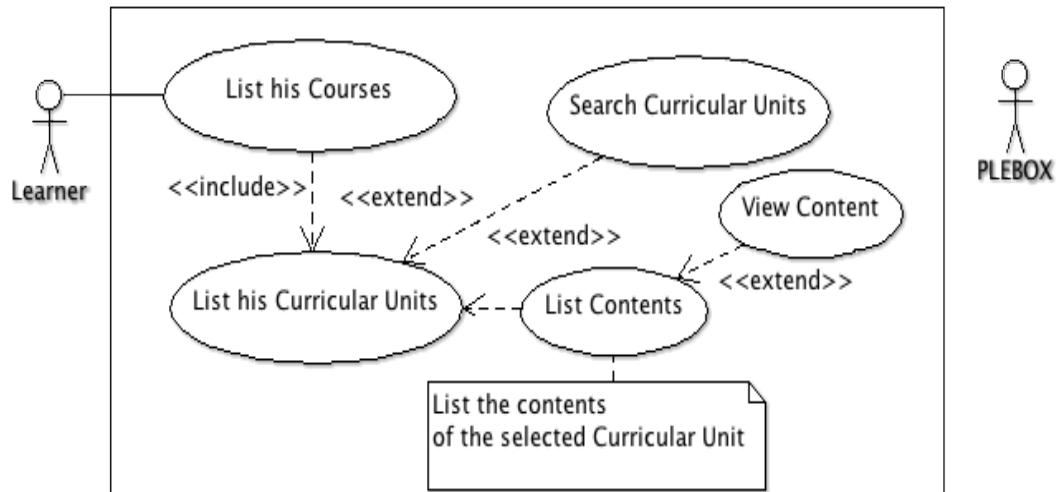


Figure 42 - Assignments Module User Case Diagram.

3.2 The Relational Structured Database

A relational database is a collection of data items organized as a set of formally-described tables from which data can be accessed or reassembled in many different ways without having to reorganize the databases tables.

4.3.1 Entity Relationship Model

In software engineering an entity relationship model (ER) is a representation of abstract and conceptual data. In other words the entity relationship model is a high-level data model composed by entities and entities-relationships that is useful in the development of a conceptual design for a database. When a relational database needs to be designed is necessary to construct and draw an ER before the beginning of application development process. Thus this model provides an abstraction of operation principles of the application.

Our model, presented in Appendix 1, is divided into two major zones. The core zone containing all essential features for modules, notifications, user management, and the learning management system zone containing support for courses, curricular units, and content. In the following paragraphs we thoroughly explain the entities in each zone and its relationships.

A number of important entities and relationship tables compose the core zone. The *PLEBOX_Module* represents and abstraction of a module and is related to the entities *PLEBOX_Permission* and *PLEBOX_User*, that respectively represents an abstraction of a permission and a user. The relationships between entities are made through a set of relational tables. The *PLEBOX_Permission_Module* details that one or more permission can be associated with one or more modules. On the other hand, the *PLEBOX_Desktop* relationship table details that one or more modules can belong to a desktop of one or more users. These

relationships makes possible the existence of different desktops composed by modules and environment settings, on the other hand, modules are divided to specific groups of users through the use of permissions.

As previous stated, the *PLEBOX_User* represents an abstraction of a user. Additionally this entity is related to the entities *PLEBOX_Group* and *PLEBOX_Notifications*, which respectively represent an abstraction of a group and a notification. The *PLEBOX_User_Group* relationship table details that one or more users can belong to one or more groups, and the *PLEBOX_User_Notification* relationship table details that one or more notifications can be sent to one or more users. Furthermore, the *PLEBOX_Group_Notification* relationship table details that one or more notifications can be sent to one or more groups of users. These relationships allow the platform to encapsulate users into groups (e.g. administrators, instructors, learners, and guests) and also allow the existence of core structures to exchange messages/notifications between users and groups.

Several entities and relationship tables compose the learning management system zone. The *PLEBOX_Course* that represents an abstraction of a course linked to the entity *PLEBOX_Course_Type* was designed to outline the description that one type of course can belong to several courses but a course does not belong to various types. The *PLEBOX_Course* entity is also associated with the entities *PLEBOX_User*, and *PLEBOX_CurricularUnit* that respectively represents an abstraction of a user and a curricular unit. The relationships between courses, users, and curricular units are made through a set of relational tables. The *PLEBOX_Course_User* relationship table details that one or more users can be enrolled with one or more courses, the *PLEBOX_Course_CurricularUnit* and the *PLEBOX_User_CurricularUnit* relationship tables details that one or more courses/users can be sent to one or more courses/curricular units. As main contribution these relationships allow the platform to enroll users into courses and curricular units.

On the other side we have the *PLEBOX_Content* representing an abstraction of content. This entity is related to the entities *PLEBOX_Repository*, *PLEBOX_Standard*, *PLEBOX_CurricularUnit*, *PLEBOX_User*, and *PLEBOX_Type*. The *PLEBOX_CurricularUnit_Content* relationship table details that one or more contents can be enrolled with one or more curricular units, the *PLEBOX_User_Content* relationship table details that one or more contents can be assigned to one or more users. In addition, a group of entities were designed to assist the characterization of content. The *PLEBOX_Repository* entity outlines that one repository can belong to several contents however a content cannot belong to various repositories; the *PLEBOX_Standard* describes that one standard can belong to several contents however a content does not belong to various standards. Finally, the *PLEBOX_Type* entity aims to detail that one type can belong to several contents but a single content does not belong to various

types. As main contribution these relationships allow the platform to assign learning content to users and curricular units.

3.3 Class Diagrams

A class diagram is a blueprint that describes the structure of a system by showing the systems classes, their attributes, operations, and relationships that will later be deployed as services, APIs, and SDKs. In the following paragraphs we present a set of class diagrams that details the core functionalities of the platform.

The access and utilities components class diagram (Figure 43) details a set of operations for managing the access to a database and a number of security and utilities operations to provide common operations to upper layers.

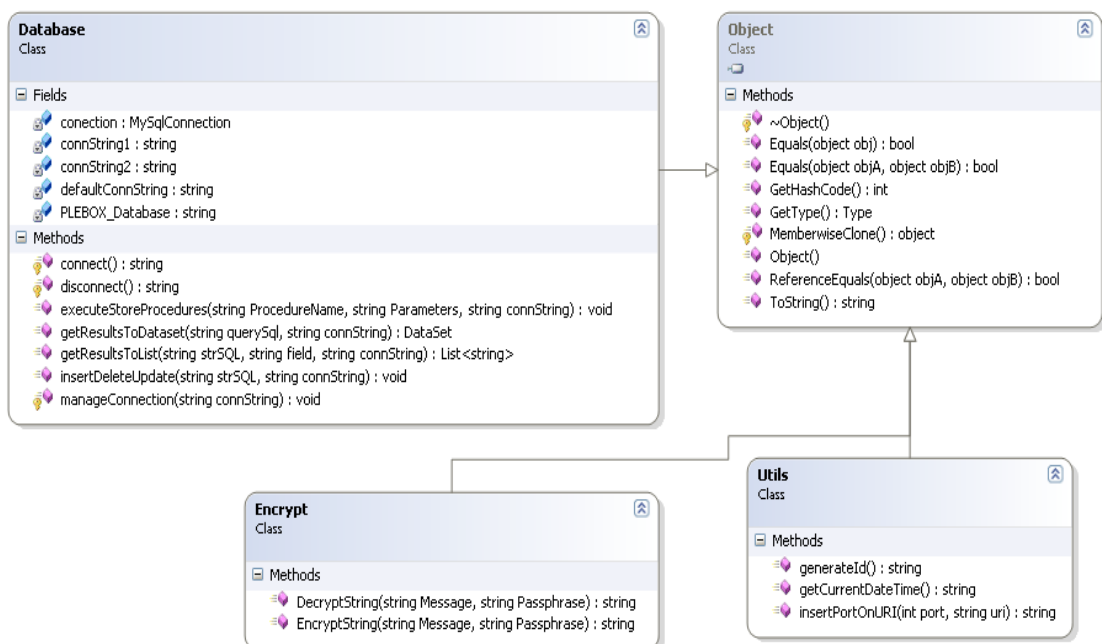


Figure 43 - Access and Utilities Components Class Diagram.

The desktop components class diagram (Figure 44) outlines the structure and operations to manage user's desktop and a set of associated modules that can be encapsulated into windows. The users class diagram (Figure 45) details the structure and operations to manage users, also providing ways to allow the combination between users and groups. The content class diagram (Figure 46) outlines the structure and operations to manage the content and associated characteristics, like standards, types, and repositories. The courses class diagram (Figure 47) details a set of features that allows the management of courses, also providing an association management that link courses or curricular units to users. The curricular units class diagram (Figure 48) outlines a set of operations to manage curricular units and associations between users and courses.

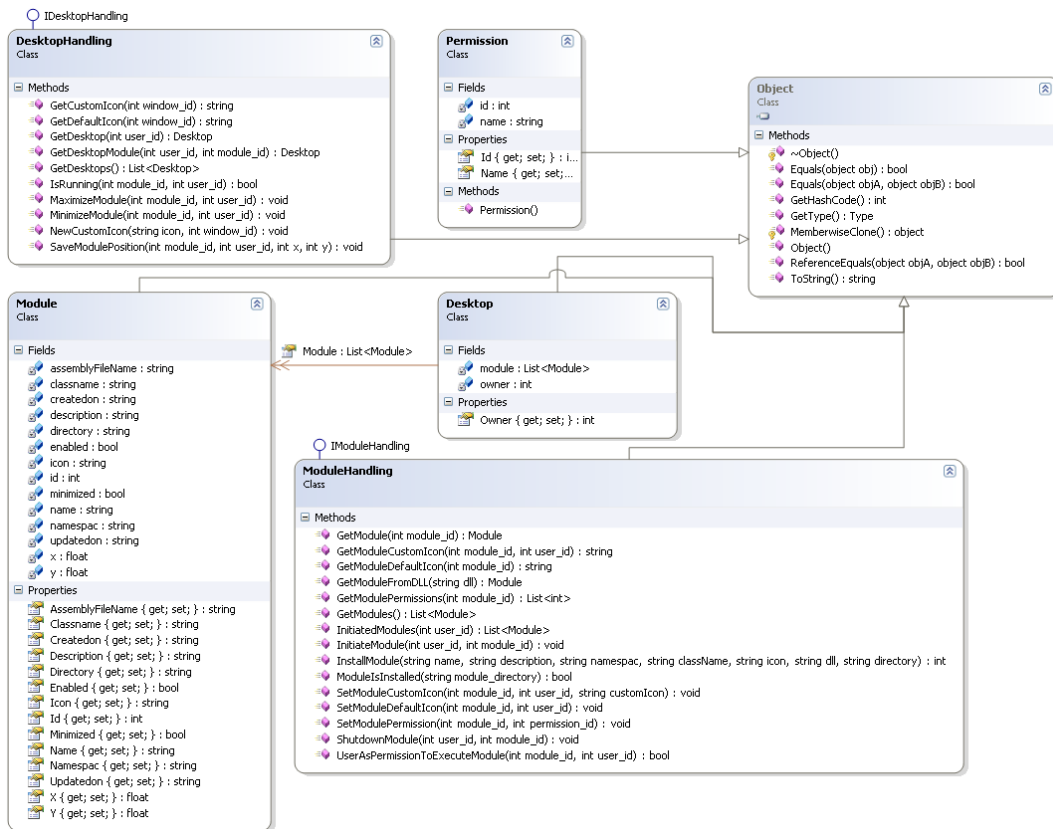


Figure 44 - Desktop Components Class Diagram.

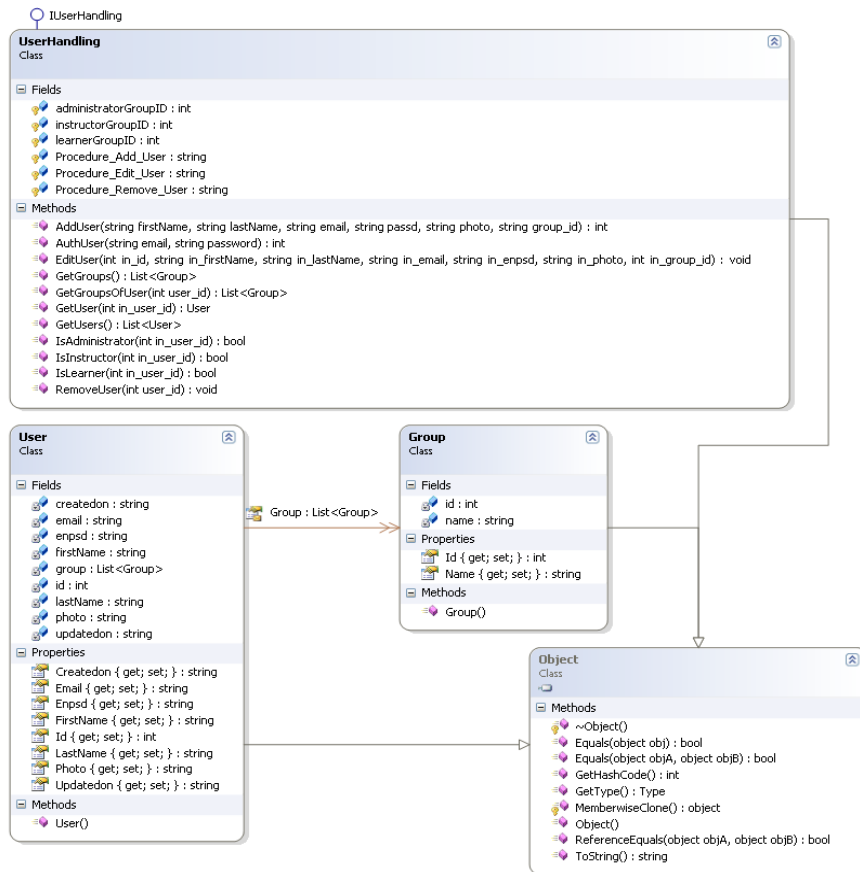


Figure 45 - User Class Diagram.

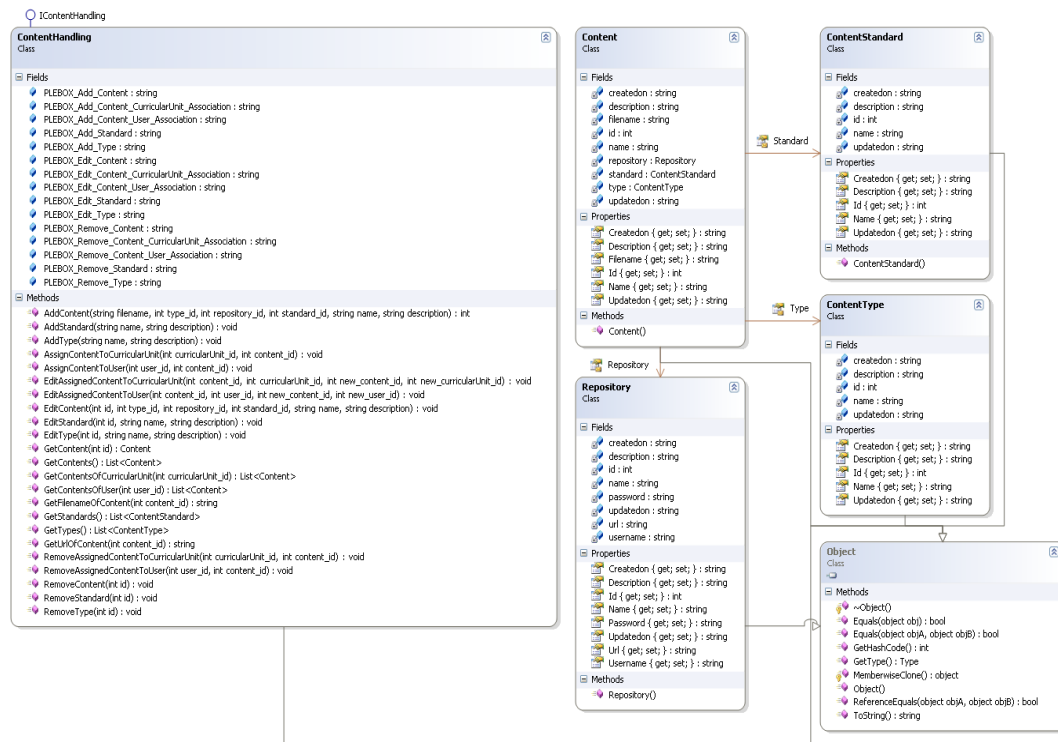


Figure 46 - Content Class Diagram.

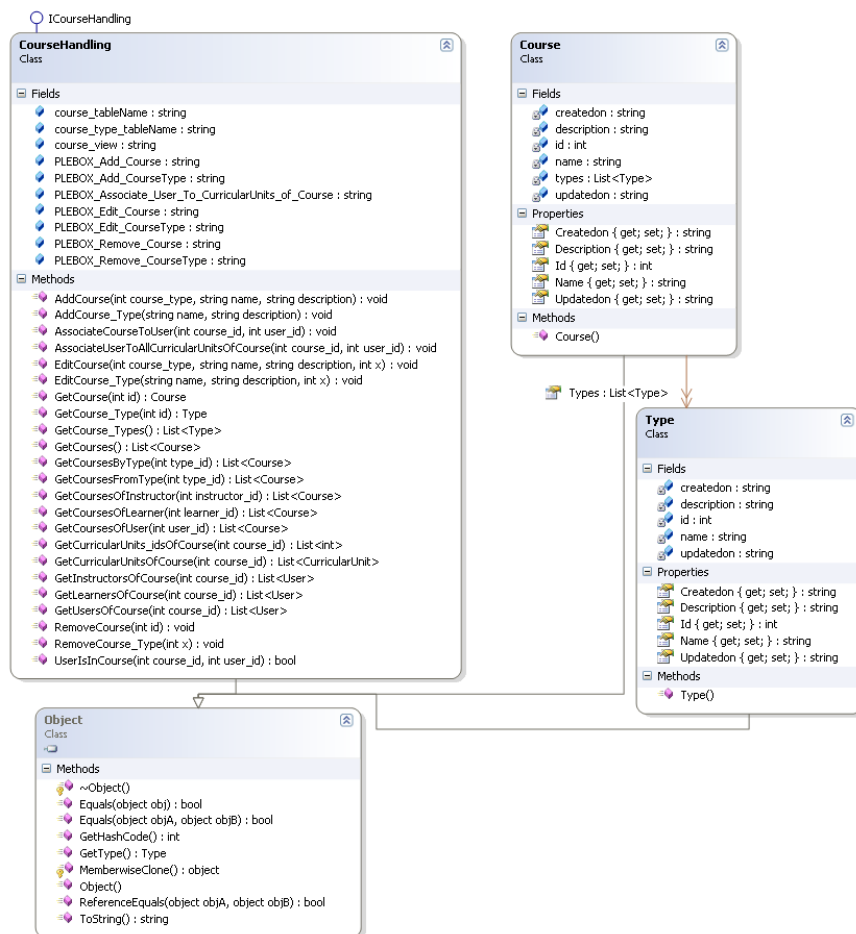


Figure 47 - Courses Class Diagram.

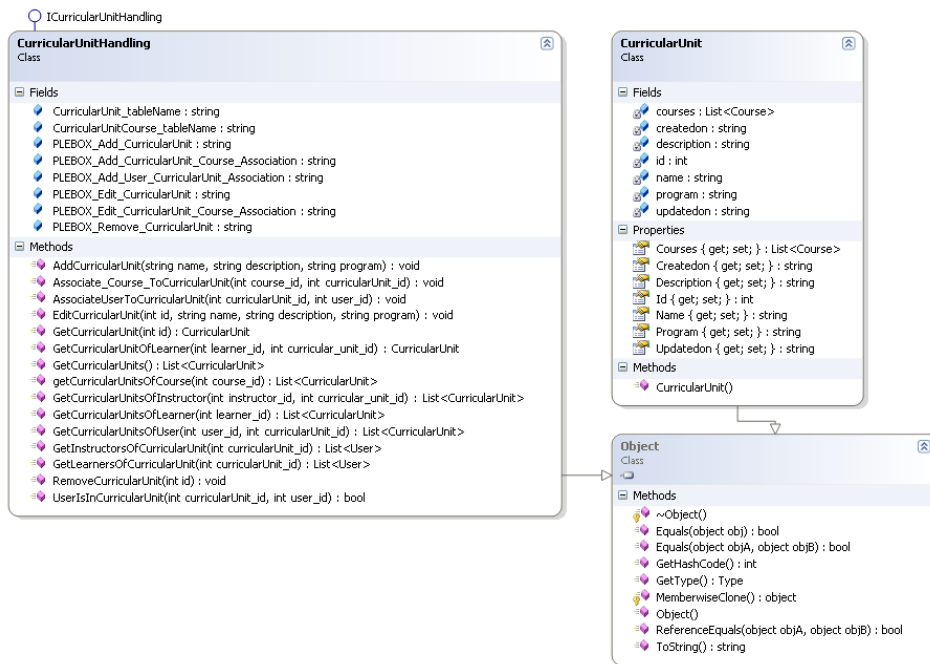


Figure 48 - Curricular Units Class Diagram.

The notifications class diagram (Figure 49) details a range of features to manage notifications, which includes ways to retrieve users and groups notifications and a simplified way to send notifications to users, groups or even to learners/instructors of a selected curricular unit. The repositories class diagram (Figure 50) outlines a set of operations to register local or remote repositories. This will enable the existence of content aggregation in one single place.

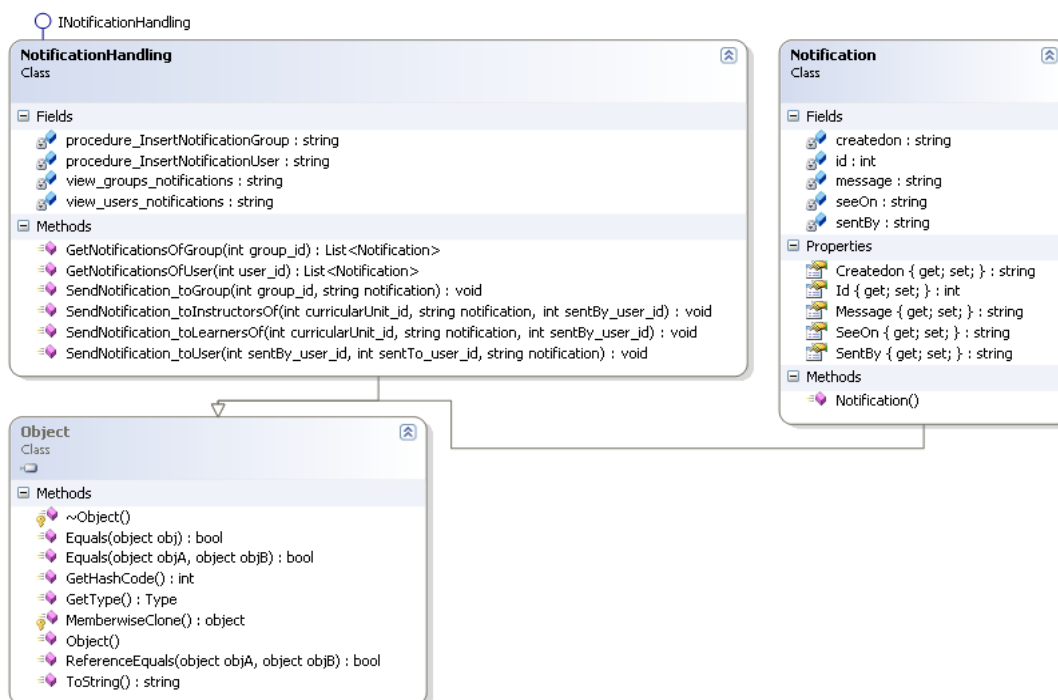


Figure 49 - Notifications Class Diagram.

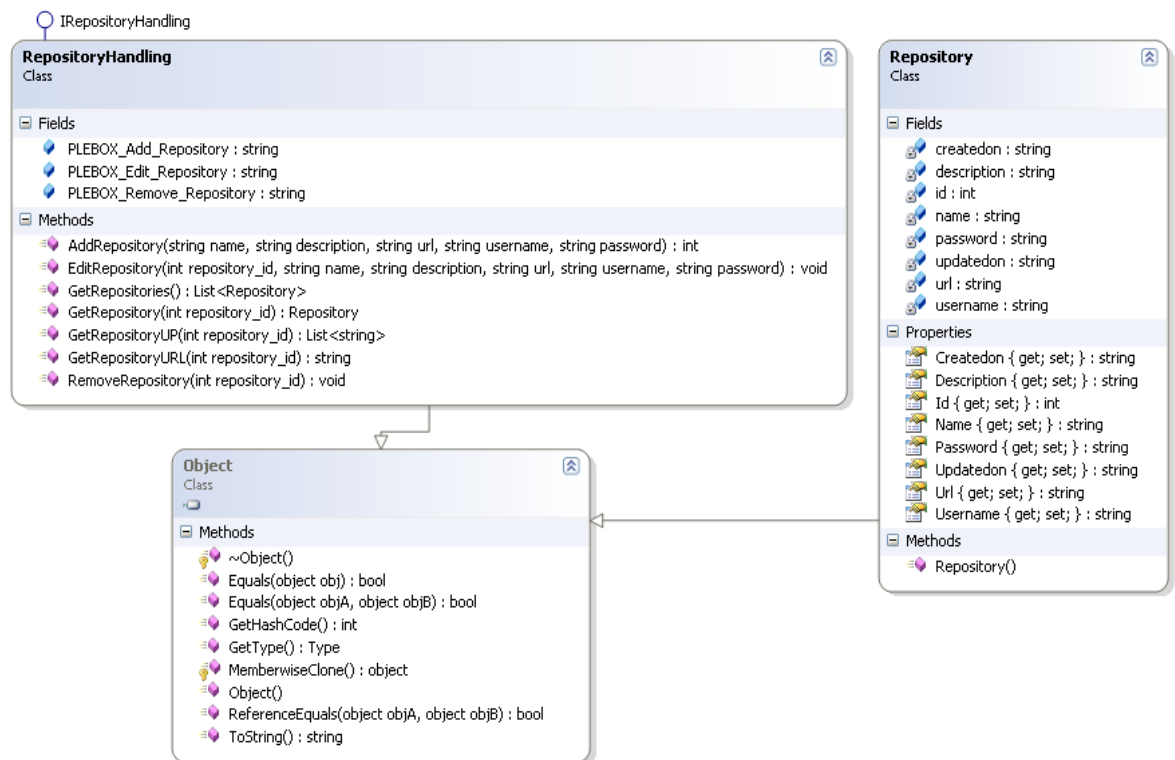


Figure 50 - Repositories Class Diagram.

Chapter 4 - PLEBOX Technologies

This chapter outlines a description of the PLEBOX platform, developed versions, resources, architectures, views, stored procedures and a detailed description of different development scenarios. Furthermore this chapter provides a list of necessary steps to start developing a module for the platform.

4.1 Introduction

Personal learning environment box (PLEBOX) is an e-learning platform that combines features from learning management systems, personal learning systems and rich Internet applications. The box concept means that within it there is a set of modules, features, and tools that can be added or used into different user environments or desktops (Figure 51).



Figure 51 - PLEBOX Logo Illustration.

This platform contributes to modular and expandable software, supporting a simplified interaction with users. Moreover, it provides, a single learning environment by integrating all required tools. As a result, the user is focused on the learning environment, without the need to use external tools. With PLEBOX, the concept of e-learning platform is stretched into the concept of rich internet applications and remote desktops, by enabling learning tools installation, courses management, curricular unit management, windows type interaction, drag and drop features, content assignment, and communication layers [6]. PLEBOX modules are just applications, that can be installed on the box, which may or many not be developed by others, with the aid of a set of APIs/services and SDKs. Furthermore, PLEBOX features are divided into type one features and type two features. Type one features are applications with the aim of managing the box and provide a user interface to manage users, roles, and modules. Type two features, also called widgets, are mini applications that exist within the user's desktop and deliver a simple set of functions.

The initial prototype, version 0.1, was developed on top of the SharePoint framework and used a learning management system called SharePoint learning kit (SLK), which provides basic learning management systems operations. However, due to issues and limitations related to

the SharePoint framework and SLK we decided to move the platform to a stand-alone environment. The new version, 1.0, provides a more powerful learning and non-learning APIs, SDKs, and a high rate of stability that was not possible when it was running on top of the SharePoint framework. Nevertheless, version 1.0 was insufficient for some requirements, such as the inability to incorporate his internal APIs in a new or existing external environment. With this in mind, we start the development of a new version, version 2.0, based on REST services.

4.2 Resources

A number of resources were used to develop and support the platform. In the following sub-sections we present these resources.

4.2.1 Windows Server 2008 R2

Windows Server 2008 R2 is a server operating system manufactured by Microsoft. It was released on July 22, 2009 and launched on October 22, 2009. This operating system was built on top of Windows NT 6.1, the same core operating system used with the client-oriented Windows 7.

This version includes new functions for active directory management (this active directory is a structure used on Microsoft Windows computers and servers to store information about, networks users and domains), virtualization tools, and native support for several processors. Additionally, it also includes a new Internet informational services (IIS) version, version 7.

4.2.2 Internet Information Services 7

The Internet information services is a software included in Windows Server 2008 R2 and other Windows product families, which provides security-enhanced, easy-to-manage web servers for developing and reliably hosting web applications, and web services. This enables organizations to deliver rich, web-based experiences. Internet information services supports HTTP, HTTPS, FTP, FTPS, SMTP, and NNTP.

The architecture of IIS is modular, where modules, also called extensions, can be removed or added individually so that only modules required for specific functionality have to be installed. IIS also includes native modules as part of the full installation. Examples of those modules can be, security modules, content modules, caching modules, login modules, and a number of diagnostics modules.

4.2.3 Visual Studio 2010

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. Is used to developed console, graphical user interface applications, web applications, web sites, and web services in both native code together with managed code for all platforms supported by Windows, Windows Mobile, .NET framework, and Microsoft Silverlight.

Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The integrated debugger allows debugging in a source-level or in a machine-level. There are others built-in tools including forms designer for building GUI applications, web designer, class designer, and database schema designer. Microsoft Visual Studio is expandable, because accepts plug-ins that enhance the functionality at almost every level.

This IDE supports different programming languages such as Visual C++, VB.NET, C#, and F#. However, it also supports other languages like Python, M, and Ruby. Visual Studio enables the support to a vast number of web languages and meta-languages, such as XML/XSLT, HTML/XHTML, JavaScript, and CSS.

Microsoft Visual Studio is available in several editions: standard, professional, tools for office, and team system. The use of one version over another depends exclusively on the target area, however the professional edition includes virtually all features of the other editions [60].

4.2.4 Relational Database Management System - MySQL

MySQL is a relational database management system (RDBMS) based on structured query language (SQL) released in January 1998 that allows the creation of relational database structures on a Web-server, in order to store data or automate procedures. MySQL runs in all platforms, including Linux, UNIX, Mac OS, and Windows. It is fully multi-threaded using kernel threads and provides application program interfaces (APIs) for many programming languages, including C#, C, Eiffel, Java, Perl, PHP, Python, and TCL.

MySQL is used in a wide range of applications, including data warehouse, e-commerce, e-learning, web databases, login applications, and distributed applications. It is also increasingly embedded in third-party software and other technologies.

The software XAMPP can be used to easy deploy and install a MySQL server, this software is an open source cross-platform Web server solution stack package, consisting of the Apache HTTP server, MySQL, and a set of common interpreters.

4.2.5 ASP.NET and the Microsoft .NET Framework

The .NET framework is an environment for building, deploying, and running Web applications or Web services. The .NET framework consists of three major parts: programming languages (C#, Visual Basic, and J#); server technologies and client technologies (ASP.NET and Windows forms); development environments (Visual Studio .NET and Visual Web Developer).

ASP.NET is a Microsoft server-side Web technology that is compiled into the .NET framework. ASP.NET takes an object-oriented programming approach to Web page execution. This powerful server side scripting technology enables scripts (embedded in Web pages) to be executed by an Internet server like IIS. ASP.NET is the next generation ASP, but it's not an upgraded version, instead is an entirely new technology for server-side scripting written from the ground up.

4.2.6 REST and SOAP Web Services

As a new distributed computing technology, Web services are more and more often used for interoperation among heterogeneous platforms [61]. Web Services provide a conceptual foundation and a technology infrastructure for service-oriented computing. It allows program written in different languages on different platforms to communicate with each other in a standard-based way. They are considered as reusable software components over the Internet, not limited to current environments, instead they can be deployed into every software-system that is Web service-aware [62-63].

The world wide web consortium (W3C) defines Web service as a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface description in a machine-processable format and those others systems interact with the Web service in a manner prescribed by its description using SOAP-based Web services or REST-based Web services [64]. Simple object access protocol (SOAP) is a protocol specification for exchanging data within a network. This type of Web services have little relation to the Web, where applications access and invoke the functional components located on remote machines through the APIs over the network, where they also convert data from platform-specific formats into XML format.

In the other hand, representational state transfer (REST) is an architecture style for designing networked applications. REST services are based on URLs and HTTPs methods [65], where the idea is that, rather than using complex mechanism such as COBRA, RCP, or SOAP to connect between machines, simple HTTP is used to make calls to create data, post data, read data, or delete data between machines. REST is a lightweight Web service architecture style, its implementation and operation is less marked than SOAP and it is more concise. It can be completed through the HTTP protocol and by using cache it can also improve the response speed performance and efficiency, which are all better than SOAP protocol. This means, that

the REST architecture can be used in a more vast range of target devices and environments, such as the mobile environment.

4.2.7 Windows Communication Foundation 4.0

The windows communication foundation (WCF) is an API in the .NET framework to build connected service-oriented Web or non-Web applications. It is a rich technology foundation designed to build distributed service-oriented applications (Figure 52). WCF supports SOAP and REST Web services because is designed to use service oriented architecture principles to support distributed computing where services have remote consumers. Clients can consume multiple services and theses can be consumed by multiple clients, where they are not coupled with each other.

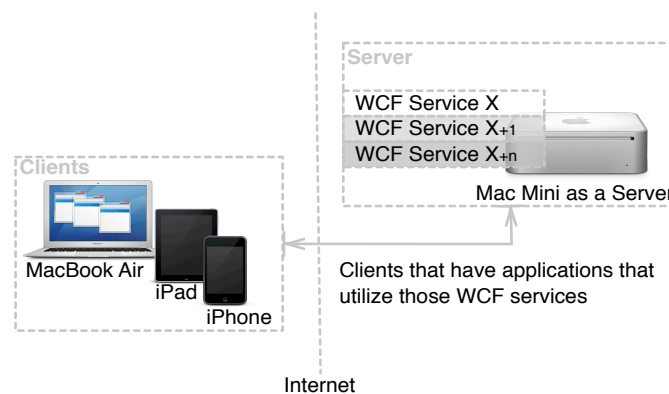


Figure 52 - WCF Services Basic Concept.

WCF uses two main concepts, endpoints, and behaviors. Endpoints enable clients to connect to a WCF where each service exposes its contract via one or more endpoints. Endpoints have an address and binding properties that specify how the data is transferred or exposed as XML or JSON. Behaviors are types that modify or extend a service or a client functionally. This means that developers can create custom processing operations, transformation and inspection of messages that are sent or received. Some common examples of behavior use are: control of whether metadata is published, provide security features to a service, possibility of parameter validation, message validation, and invocation of additional operations when messages are received or sent.

4.2.8 JavaScript Libraries

A JavaScript library is a repository of pre-written JavaScript that allows an easy way to develop JavaScript based Web-applications, in particularly for AJAX and other Web-centric technologies. JQuery is a free and open-source cross-browser JavaScript library designed to simplify the client-side scripting of HTML, handling events, manipulate styles, preform animations, and add Ajax interactions to web pages. JQuery was released on January 2006 and it is used by over 49% of the 10,000 most visited websites. Because of JQuery's architecture, others developers can create custom plugins to extend the functionalities of the

library. There are many plugins currently available, such as: Uniform, PicNet, JScrollPanel, DataTables, Tippy, Farbtastic, and File Style.

4.3 Database Views and Stored Procedures

A number of views, and stored procedures were developed to support the platform. In the following sub-sections we present these views and stored procedures.

4.3.1 Database Views

In the database field, a view consists of a stored query accessible as a virtual table that is sorted and displayed in a particular way. In our solution we have a set of fifteen views divided into three key conceptual zones: users, contents, and courses (Figure 53). In the following paragraphs we describe each of the zones and their respective views.

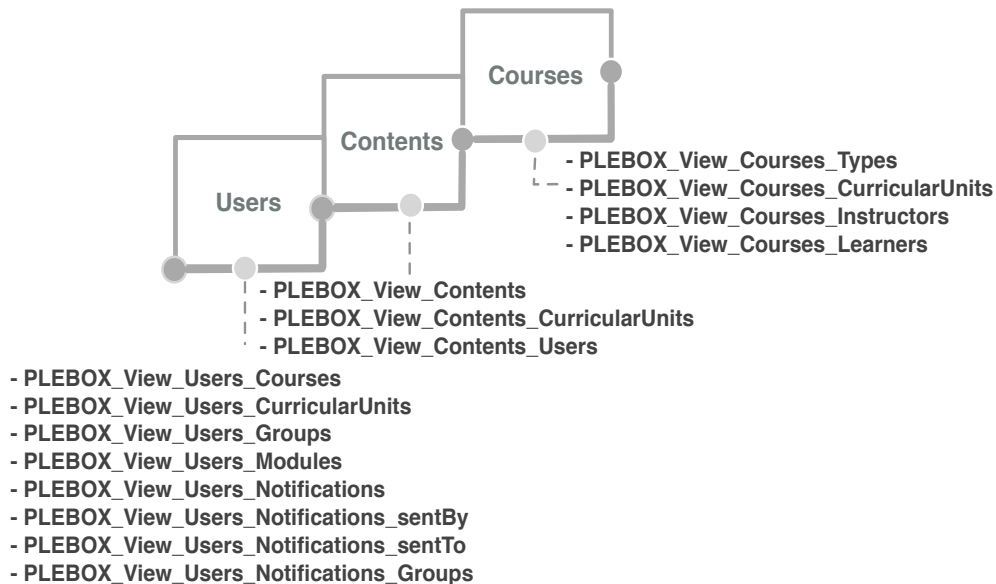


Figure 53 - Conceptual Zones and Respective Views.

Firstly in the courses zone we have the following views: *PLEBOX_View_Courses_Types*, *PLEBOX_View_Courses_CurricularUnits*, *PLEBOX_View_Courses_Instructors*, and *PLEBOX_View_Courses_Learners*. The first view is intended to show the courses and their respective types, the second view aims to bring together the relationships between the courses and curricular units. The last two views are intended to show the instructors and learners that are related to a set of courses.

Secondly in the contents zone we have the following views: *PLEBOX_View_Contents*, *PLEBOX_View_Contents_CurricularUnits* and *PLEBOX_View_Contents_Users*. The first view is intended to show the content and related standards and repositories, the second view has the objective of showing the contents related to a set of curricular unit, the third, and last view, aims to show the content that is related to the users.

Thirdly in the users zone we have the following views: *PLEBOX_View_Users_Courses*, *PLEBOX_View_Users_CurricularUnits*, *PLEBOX_View_Users_Groups*, *PLEBOX_View_Users_Modules*, *PLEBOX_View_Users_Notifications*, *PLEBOX_View_Users_Notifications_sentBy*, *PLEBOX_View_Users_SentTo* and *PLEBOX_View_Users_Notifications_Groups*. The first and second view are intended to show the relationships between users and courses or curricular units, the third view has the main objective to show the relationships between users and groups presents in the platform. The fourth view aims to show which modules are running in the desktop of several users. Finally the fifth, sixth, seventh, and eighth views are intended to show the relationships between notifications, users, and groups. This means that is possible to know which group or specific user a notification was sent or received.

4.3.2 Stored Procedures

A stored procedure is a function that is inside a database, available to applications that access and operates a relational database system. In our solution we have a group of stored procedures divided into nine key conceptual zones: desktops, users, standards, repositories, notifications, contents, courses, curricular units, and the associations zone (Figure 54). In the following paragraphs we describe each of the zones and their respective stored procedures.

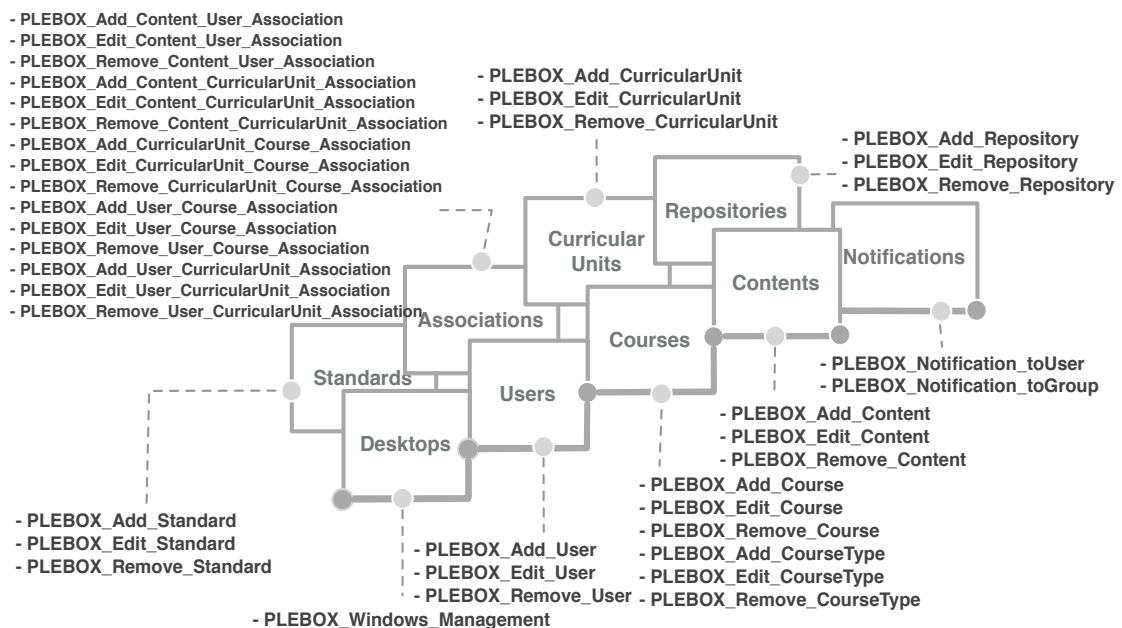


Figure 54 - Conceptual Zones and Respective Stored Procedures.

The primary zone is the desktops zone, this zone contains the stored procedure *PLEBOX_Windows_Management* that allows the possibility of managing desktop windows, this means that using this store procedure we can save the position (x,y) of a set of modules that are present in the user's desktop.

The users zone aims primarily to manage users (add, edit, and remove). This zone is composed by a set of stored procedures: *PLEBOX_Add_User*, which has the core functionality to add users to the platform, *PLEBOX_Edit_User* that aims to edit users of the platform and finally the *PLEBOX_Remove_User* that as the objective of remove a user of the platform.

The standards zone aims primary to manage standards (add, edit, and remove). This zone is composed by a set of stored procedures: *PLEBOX_Add_Standard*, *PLEBOX_Edit_Standard*, *PLEBOX_Remove_Standard* that respectively are aimed to add, edit, and remove standards.

The repositories zone aims primary to manage repositories (add, edit, and remove). A range of stored procedures composes this zone: *PLEBOX_Add_Repository*, *PLEBOX_Edit_Repository*, *PLEBOX_Remove_Repository* these procedures have the function to add, edit, and remove repositories.

The notifications zone aims primary to manage notifications. This zone is composed by a set of stored procedures: *PLEBOX_Notification_toUser*, *PLEBOX_Notification_toGroup* intended to send notifications to users or groups.

The contents zone aims primary to manage the contents that are present in one local or remote repository. This zone has the following stored procedures: *PLEBOX_Add_Content*, *PLEBOX_Edit_Content*, *PLEBOX_Remove_Content* that respectively are aimed to add, edit, and remove contents.

The courses zone aims primary to manage courses (add, edit, and remove). This zone has the following stored procedures: *PLEBOX_Add_Course*, *PLEBOX_Edit_Course*, *PLEBOX_Remove_Course* providing functions to enable the possibility to add, edit, and remove courses. This zone also contains the stored procedures, *PLEBOX_Add_CourseType*, *PLEBOX_Edit_CourseType*, and *PLEBOX_Remove_CourseType* that respectively are aimed to add, edit, and remove types of courses.

The curricular units zone aims primary to manage curricular units. This zone has the following stored procedures: *PLEBOX_Add_CurricularUnit*, *PLEBOX_Edit_CurricularUnit*, *PLEBOX_Remove_CurricularUnit* that respectively are aimed to add, edit, and remove curricular units.

The association's zone is the last zone that aims primary to manage associations between users, courses, curricular units, and contents. In the context of users and contents there are three main procedures *PLEBOX_Add_Content_User_Association*, *PLEBOX_Edit_Content_User_Association*, and *PLEBOX_Remove_Content_User_Association* aimed to add, edit, and remove associations between contents and users. On the other hand, we have the possibility of managing the association between the content and a set of curricular units, using the

stores procedures *PLEBOX_Add_Content_CurricularUnit_Association*, *PLEBOX_Edit_Content_CurricularUnit_Association*, and *PLEBOX_Remove_Content_CurricularUnit_Association*. In the context of courses and curricular units there is the possibility to manage the association between curricular units and courses, using the stored procedures: *PLEBOX_Add_CurricularUnit_Course_Association*, *PLEBOX_Edit_CurricularUnit_Course_Association*, and *PLEBOX_Remove_CurricularUnit_Course_Association*. Furthermore in the context of the user is also possible to manage associations between users and curricular units and also users and courses using the stored procedures *PLEBOX_Add_user_Course_Association*, *PLEBOX_Edit_User_Course_Association*, *PLEBOX_Remove_User_Course_Association*, *PLEBOX_Add_User_CurricularUnit_Association*, *PLEBOX_Edit_User_CurricularUnit_Association*, and *PLEBOX_Remove_User_CurricularUnit_Associations*. This means that is possible to add a user to a curricular unit, even if it is not in the course to which the curricular unit is associated.

4.4 Architectures

Software architecture is often described as the organization or structure of a system, while the system represents a collection of components that accomplish a specific function or a set of functions [66].

4.4.1 Version 1.0

In figure 55 is showed the architecture of version 1.0 composed by the core layer and twenty-one main components. Each component operates with others performing a specific function and can be considered as APIs, since any programmer can use them. In the other hand, layers represents a group of components. In the following paragraphs we present each of these components, which are encapsulated into the core layer.

The exceptions handling component is responsible for managing errors and provide exception handling, which if untreated can be catastrophic to the system. This component is mostly used by the database and IO component. The IO component provides basic input/output operations, taking into account that there is specific number of folders present on the server with read and writes permissions. The database component is highly used across the system and is responsible for managing database connections, providing read, write, and other common operations. The notification component provides operations for communication to and between users. This component is so flexible that can be used in a learning context, providing a number of operations to send notifications to a group of learners, instructors, and associated curricular units. The desktop component provides desktop oriented operations, this means that each module can be executed in a particular desktop, and its definitions are maintained over time, even if the user logoff. In addition this component provides operations for desktop personalization and maintenance. The menu component provides operations to create and assemble a list of modules and features that are allowed to the current user. This component is closely linked to the desktop component.

The SDK components provide a set of graphical objects that allows a rapid development of modules and features. These graphical objects can be windows, popups, dynamic tables, tabs, panels, and lists. The learning components provide a number of functions to manage users, contents, courses, and curricular units. Developers can use these components to construct learning modules and features. The features component and modules component provide a set of APIs to cluster a range of functions to manage modules and features. These components are closely linked to the SDK's components and also with the desktop component.

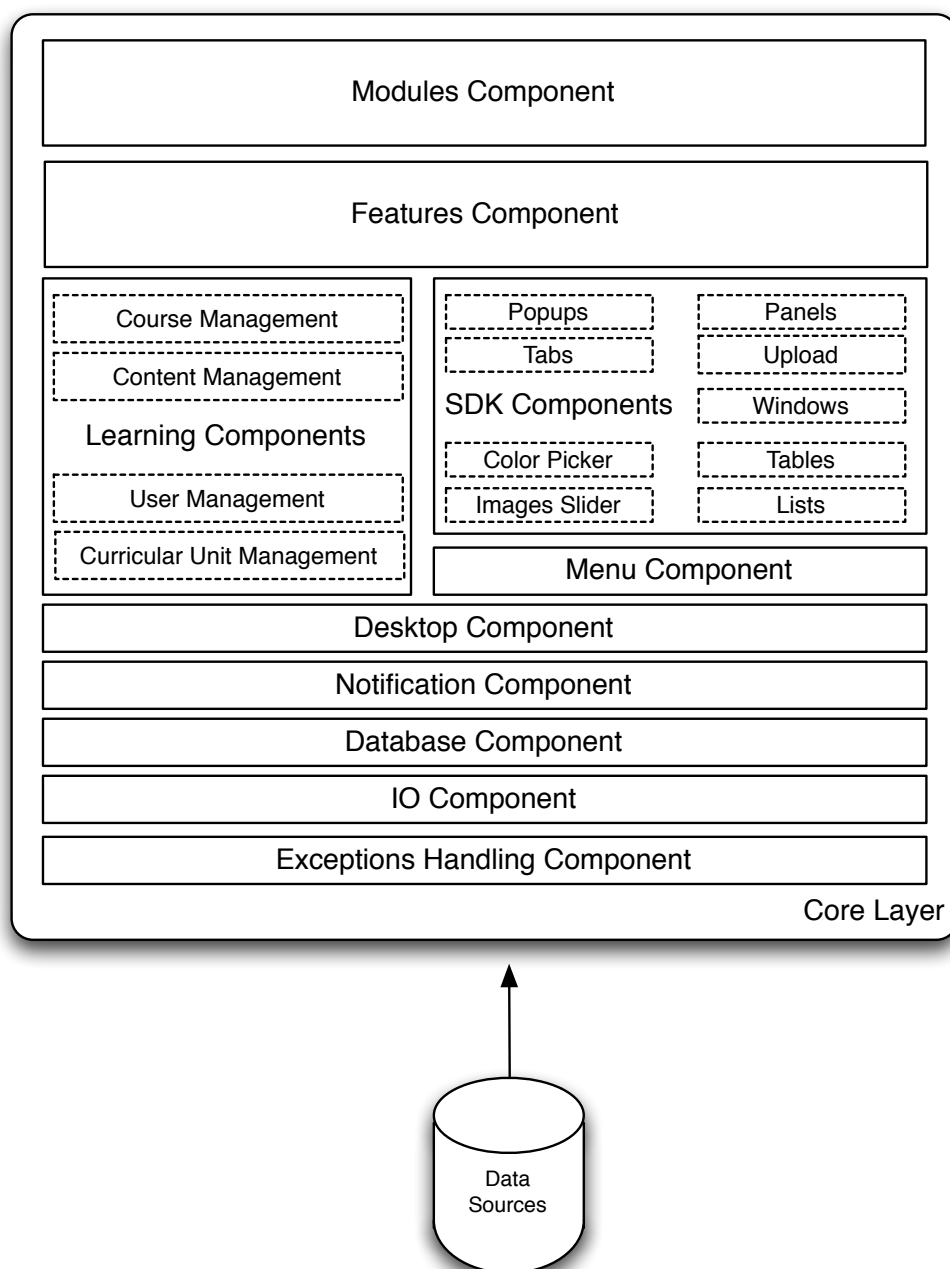


Figure 55 - Architecture of PLEBOX V1.0.

4.4.2 Version 2.0

Nowadays, information technology is moving in the direction of client/server computing where multiple platforms of computing equipment are linked together on the Web, allowing users to make the best out of each piece of hardware [67]. In version 1.0 of the platform we present concepts like modules, desktops, and different types of features. Furthermore it was been made the distinction of different types of features. In the next version, 2.0, the concept of module and type one features evolved and it became clear that these two concepts should be united. The major difference in the development of type one features is that they are no longer produced “inside the box”, instead they are now produced in the same way that modules are, regarding type two features, these are now called widgets.

Finally, the great improvement and main difference between version 1.0 and version 2.0 is the division into logical layers of data, making possible the disaggregation of components, user interfaces, services, and SDKs. Thus, providing the possibility to deploy a number of services into external platforms and environments. Figure 56 details a variation of the three-tier/layer architecture adapted for the 2.0 version of the PLEBOX platform, composed by 4, layers and a set of inline components.

A three-tier application is one where the logical decomposition of the application is reflected both at the software and hardware level. A three-tier application has front-end clients (e.g. browsers), middle-tier application servers (e.g. Web servers), and back-end database servers. Clients interact with users and send requests to application servers on behalf of users. Application servers process these requests and update the databases according to the business logic of the application. In addition to this, our architecture present a new tier/layer with the aim of distribute the business logic through Web services [68]. In the following paragraphs we present each layer/tier.

The first layer is the data layer responsible for managing connections and executions of commands against the data source [66]. This layer is not aware of other layers and is composed by a set of components. The security component ensures a set of security operations on passwords, transactions, and operations. The exceptions handling component is responsible of managing errors and provide exception management and lastly the IO component and the database component is responsible for managing database connections, providing read, write, and other common operations.

The second layer is the business layer aimed to retain the business logic, validations, and calculations related with the data provided by the data layer. The business layer is composed by a set of entities and related operations that accept and return simple or complex data structures [66]. This layer contains a set of components, such as learning components, which provides operations to manage users, contents, courses, repositories, and curricular units.

The objects handling component provides a representation of entities and related operations for important elements of the system like modules, desktops, users, and repositories. The notification component, which provides communication functions across the platform. The menu component providing operations to create and assemble a list of modules and features that are allowed to the current user and finally the desktop component is aimed to provide a set of desktop oriented operations. The desktop component also provides a number of operations to manage modules and widgets, providing ways to administer and install different modules and widgets into a set of environments, taking into account permissions and purposes.

The third layer is the services layer aimed to define the service interface and translator components that securely transfer data formats between this layer and external data contracts. Basically we deployed the business logic components as REST Web services, exposing a number of learning and administrations functions, that were initially sealed “inside the box”.

The fourth, and last layer, is the presentation layer aimed to detach user interaction logic and user interface components. This component is the user interface that aggregates all the services provided by the service layer [66]. In this layer are also detailed two usage scenarios. Scenario x that comprises the PLEBOX’s desktop-like user interface composed by modules and widgets powered by a set of APIs, SDKs, and services. Scenario y that outlines the use of PLEBOX’s learning and administration services inside an external platform or environment, that has nothing to do with the PLEBOX desktop-like environment. Additionally, in a scenario x is possible to use a number of SDK’s interface objects to help developers to create and populate interface functions into desktop-like modules and widgets.

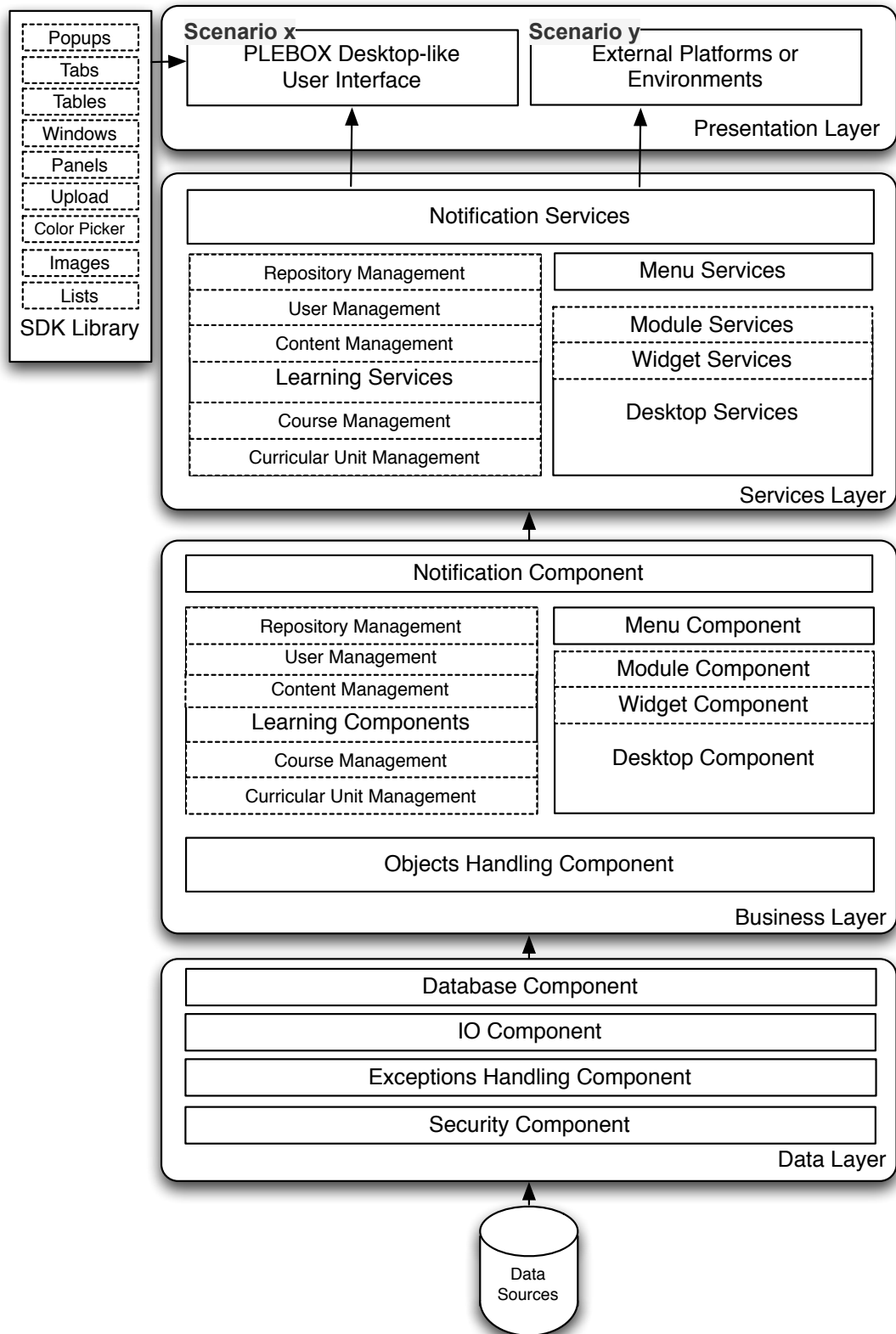


Figure 56 - Architecture of PLEBOX V2.0.

4.4 Server and Client Architectures

In terms of hardware architecture, we have a set of important parts that enables a better management of different environments. At UBI we have an Apple's X server containing a set of virtual machines that allows the installation of multiples environments (Figure 57).

The virtual machine concept enables an isolation of resources, and provides a way of running multiples operation systems into one unique machine. A virtual machine behaves exactly like a physical computer and contains it own virtual CPU, RAM, hard disk, and network interface cards. The virtualization of network interfaces allows the communication between the virtual machine and the outside world, since the virtual machine behaves like a physical device present in the same network of the host server.

Considering this architecture, client devices can perform roles of development (third-party developers can publish and develop new modules) or perform end user roles (learners, instructors, and administrators can use the platform to learn, teach, or administer).

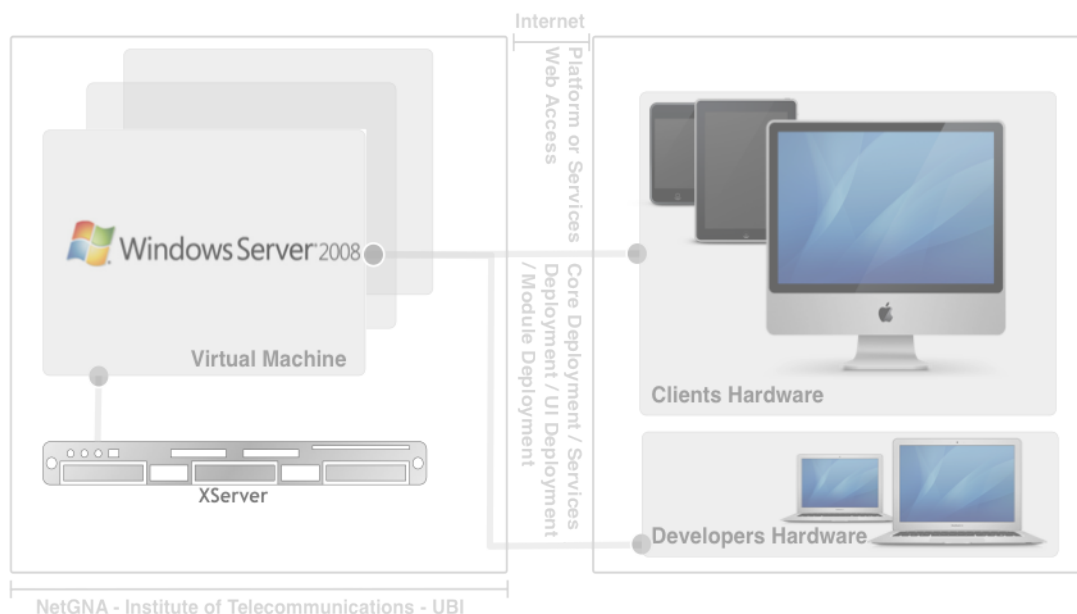


Figure 57 - Server Hardware Architecture.

4.5 Internal and External Development Scenarios

The platform supports methods for internal and external development scenarios. This means that in an internal development scenario a programmer can develop and improve each architectural layer, using the appropriated visual studio solution and the appropriated web publish profile. In the other hand, in a scenario of external development, the programmer has at his disposal a set of APIs, services, SDKs, and Visual Studio templates to develop and publish new modules to the box (Figure 58). In addition, programmers have at their dispose a wiki with practical examples of how to use the PLEBOX SDK, services, and internal APIs.

As stated early modules are just application, that can be transparently installed and developed, for an external development scenario developers can use a set of procedures to construct and developed modules:

- Create a PLEBOX module using the provided VS 2010 template;
- Copy the provided internal PLEBOX.dll file into the bin folder of the solution;
- Reference the PLEBOX library file;
- Create a new publish profile (Figure 59);
- Publish the solution to the PLEBOX server.

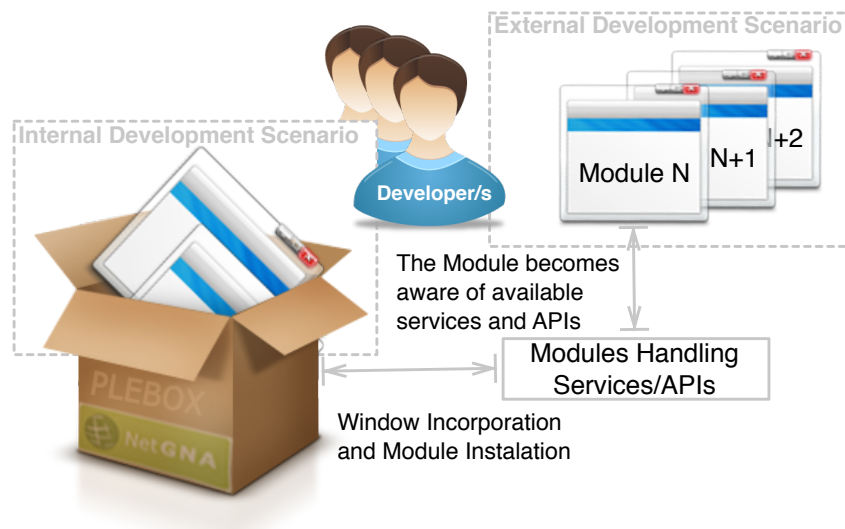


Figure 58 - Internal and External Development Scenarios.

The screenshot shows the 'Publish Web' dialog box in Visual Studio. The 'Publish profile' is set to 'Module Deploy to pleboxdev'. The 'Publish' section shows 'Build configuration: Debug', 'Publish method: Web Deploy', 'Service URL: https://193.136.67.227:2231/MsDeploy.axd', and 'Site/application: Default Web/PLEBOXdev/Modules/ManageUsers/'. The 'Credentials' section shows 'Allow untrusted certificate' checked, 'User name: Administrator', and 'Password: [masked]'. The 'Save password' checkbox is also checked.

Figure 59 - Publish Profile.

Chapter 5 - PLEBOX Components Design

This chapter details an insight into the concepts that drove the construction of a set of PLEBOXs components (user interfaces, services, APIs, and SDKs). The PLEBOX desktop-like user interface (Figure 60) contains a set of components that allows user management, module encapsulation, interface objects deployment, learning-related content management, and other common learning and non-learning operations.

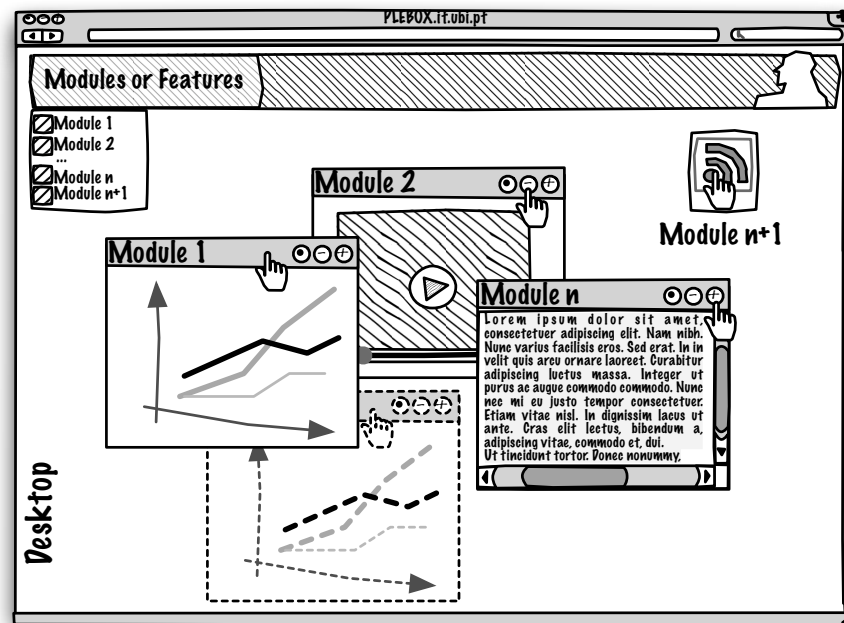


Figure 60 - Concept Sketch of the PLEBOX's Desktop

5.1 Non-Learning Components

The PLEBOX desktop-like user interface is composed by a number of major APIs and non-learning components supported by the database, its views and procedures. These components are intended to create an innovative personal environment.

The first component of the PLEBOX's desktop is the user management component aimed to manage users that are inside the box. This includes a set of operations to authenticate, list, add, remove, and edit users and all its specificities. Additionally this component supplies detailed information about each user, which includes profile information, enrolled curricular units, and enrolled courses.

The second and third component is the desktop and the menu component. These components enable the management of desktops for different users, taking into account that each desktop is characterized by a set of modules and different environment settings. The desktop management includes operations to determinate which module is running in one specific user, ability to store x/y coordinates, maximize/minimize windows, manage modules permissions,

add, amend or change the current icon. Additionally it provides operations to administer different modules, features, or widgets into a set of environments, taking into account permissions and purposes. Moreover the menu component is aimed to build a list of modules, descriptions and initialization options to provide clickable ways to start modules into a particular desktop, taking into account the current user's permissions. The fourth component is the notification that aims to provide exchange of messages between users and groups.

On top of this, a number of internal components or internal APIs provide a set of common operations. The database internal component enables a simple way, using the SQL language, to establish a connection between the client and a MySQL database in order to execute, list, and manage a number of platform-related information. Additionally, the security internal and the utility internal components, respectively, allow the encryption of data using simple encryption algorithms, and a set of operation for URL manipulation, ID management, data and time management, string and list manipulation, and other useful operations.

5.2 Learning Components

The PLEBOX desktop-like user interface is composed by a number of major APIs and learning components driven by the necessity of learning operations. Additionally these components can be used not only in the PLEBOX's desktop-like user interface but also into a number of different platforms and environments (Figure 61).

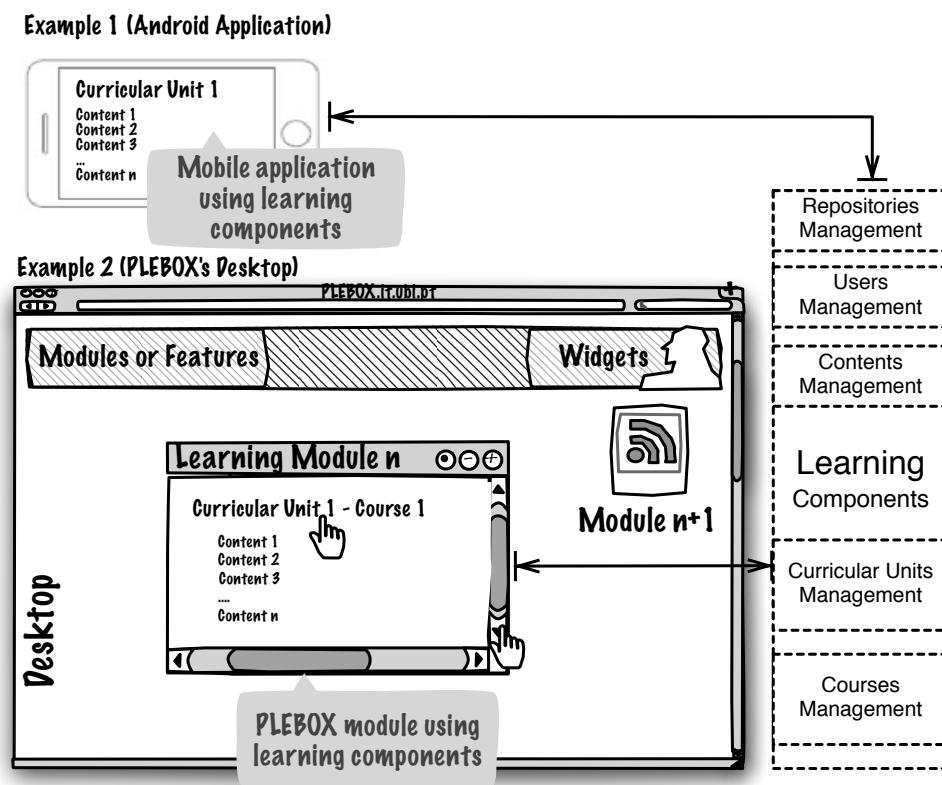


Figure 61 - Concept Sketch of the PLEBOX's Learning Components.

The course management and curricular unit management component enables course management and curricular unit management, also providing features to manage course types, and associations between users, courses, and curricular units. The repository management component is aimed to provide a set of operations to edit, retrieve, remove, and add a set of learning content repositories that can exist locally or remotely.

The content management components aimed to supply a way to manage content (list, add, edit, and remove), and its characteristics, like types and standards. Additionally, this component provides operations for management content assignment - this means that we can assign content to a user, curricular unit, or course. As build objective, the entire database structure allow the existence of content even if it is not associated with a curricular unit or course.

The final component is the notification component, outlined as an extension of the non-learning notification component that allows the exchange of text messages between users and groups. This extension allows the exchange of messages not only between users and groups but also between learners and instructors enrolled in a particular curricular unit and course.

5.4 Software Development Kit Component

The software development kit component allows a rapid development of modules, features, or widgets in an internal development scenario. This is possible using the dynamic-link library (DLL) provided by the PLEBOX platform. The SDK component is composed by a set of interface objects employed to provide a common set of required user interfaces features for Web applications (Figure 62).

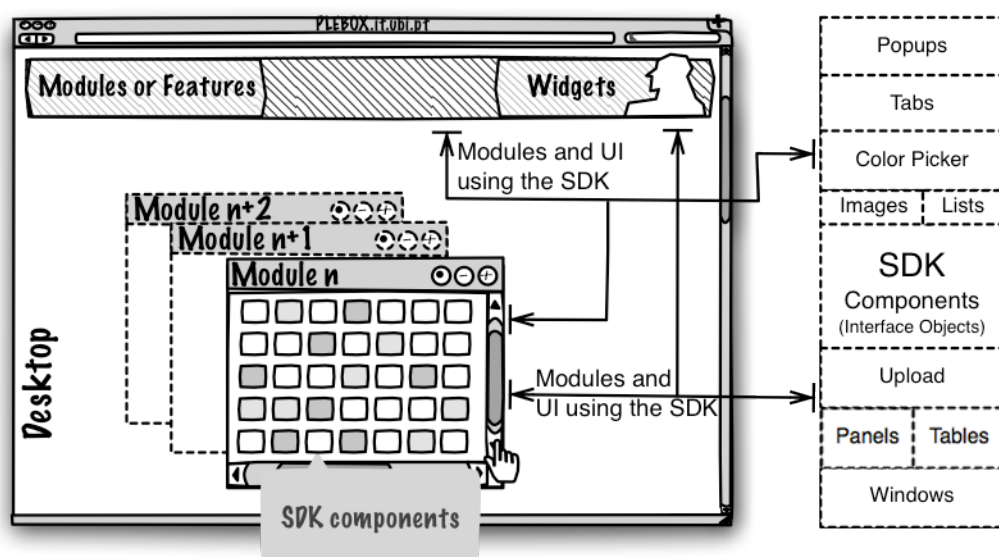


Figure 62 - Concept Sketch of the PLEBOX's Software Development Kit.

The window interface object component (Figure 63), considered the most complex object interface in the PLEBOX's SDK aimed to provide a simple way of components encapsulation. This object includes drag and drop functions, animations, and window management features making the persistence of data possible by using the window and the desktop component.

The dynamic table interface object component (Figure 64) is intended to allow the creation of tables in an easy and fast way using data sources. This interface also gives the possibility to encapsulate other common or SDK's interface objects to control and manage table items. On the other hand the scroll panel interface object component provides a scroll zone, which may incorporate a set of interface objects.

Other two components are the upload and the list interface object, respectively shown in Figure 65 and 66. These two components allow the creation of dropdown lists in an easy and customizable way and also the deployment of an asynchronous upload control to manage and upload files to a specific repository.

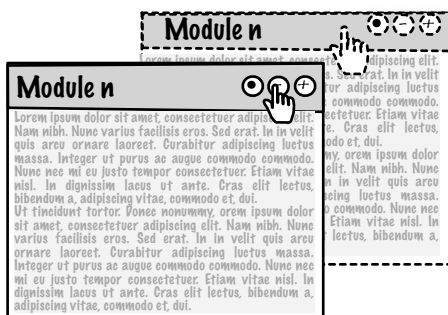


Figure 63 - Window Interface Object.

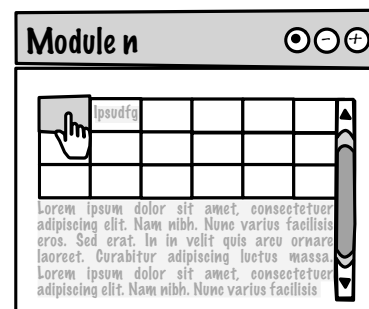


Figure 64 - Dynamic Table Interface Object.

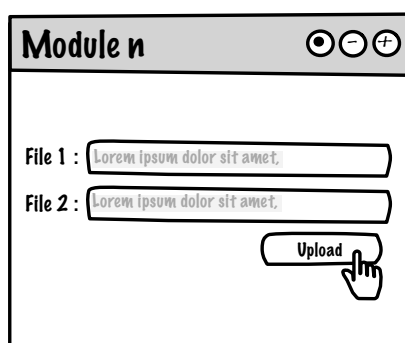


Figure 65 - Upload Interface Object.

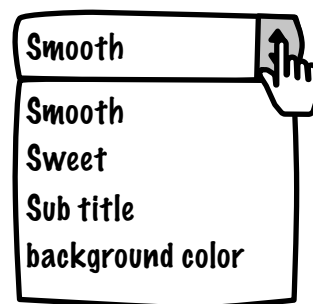


Figure 66 - List Interface Object.

The popup interface object component is divided into sub-objects that allow the control and management of modal windows. The balloon sub-object displays a popup balloon inherent to a specific object (Figure 67). In the other hand the modal window sub-object displays a window when an event is triggered (Figure 68). The tab interface object and the color picker interface object components provide a tab-based interface in order to improve user usability (Figure 69) and a simple interface for color management (figure 70).

The last component, called images interface object is divided into sub-objects that allow the creation of image buttons and images slides. The button image sub-object is a simple clickable image that can trigger an event (Figure 71). The last sub-object is the images slide (Figure 72). Theses slides allow the creation of scroll (horizontal/vertical, or both) areas containing images, where each image can be associated with a particular event trigger by the user.

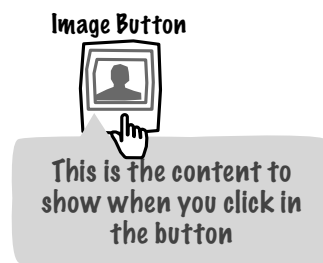


Figure 67 - Balloon Sub-Object.

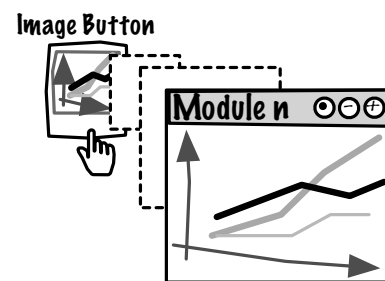


Figure 68 - Modal Window Sub-Object.

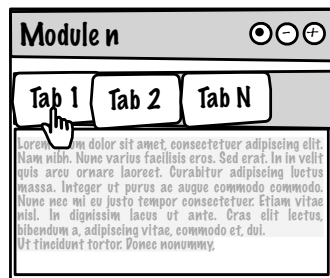


Figure 69 - Tab Interface Object.

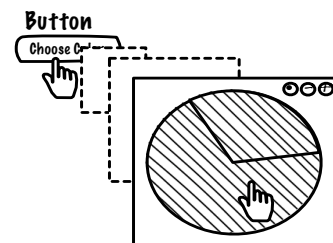


Figure 70 - Color Picker Interface Object.

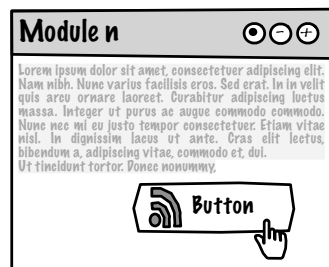


Figure 71 - Button Image Sub-object.

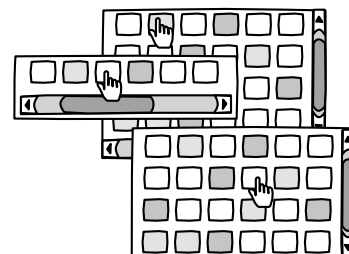


Figure 72 - Images Slide Sub-object.

Chapter 6 - System Demonstration and Validation

This chapter outlines the demonstration and validation of the platform, composed by the demonstration of the main interface and its elements, learning and non-learning modules and also the demonstration of a range of SDK's interface objects. Additionally this chapter demonstrates a set of learning services running “outside of the box”.

6.1 Platform Usage

In this section we present a set of user interfaces, which we describe as platform usage. These user interfaces were designed to consume current PLEBOX APIs, services, and SDKs that exist in the version 1.0 and subsequently in version 2.0 of the platform. In the following sub sections we present the main PLEBOX user interface, the modules management feature, a set of oriented-learning modules and finally a number of examples to demonstrate the usage of the PLEBOX SDK.

6.1.1 PLEBOX User Interface Main Environment Elements

Either using the version 1.0 or 2.0 of the platform is possible to create a user interface that enables the existence of multiple working environments. This means that each user has its own desktop, with its modules and respective settings. All this is possible because developers have at their disposal a set of non-learning and learning APIs, services, and SDKs that allows the creation, deployment, and usage of these and others user interface scenarios. In the following paragraphs we describe every element presented in the PLEBOX user interface main environment.

The authentication user interface component is shown in Figure 73. This element enables the possibility to authenticate users and provide an access to different working environments. In this element a number of login services and APIs were used in order to populate working functions to authenticate, validate, and redirect the user.

The desktop user interface element appears after the authentication has been made and aims to present a learning environment (Figures 74 and 75) composed by a set of sub-elements like widgets (Figures 76, 77, 78, and 79), menus, and modules. These sub-elements were created using the PLEBOX SDK and registered through the use of APIs and services according to the current user. As shown in Figure 75, this means that each user has its own learning environment, where all the modules settings are saved (module name, description, icons, position, window theme, and minimization/maximization options)

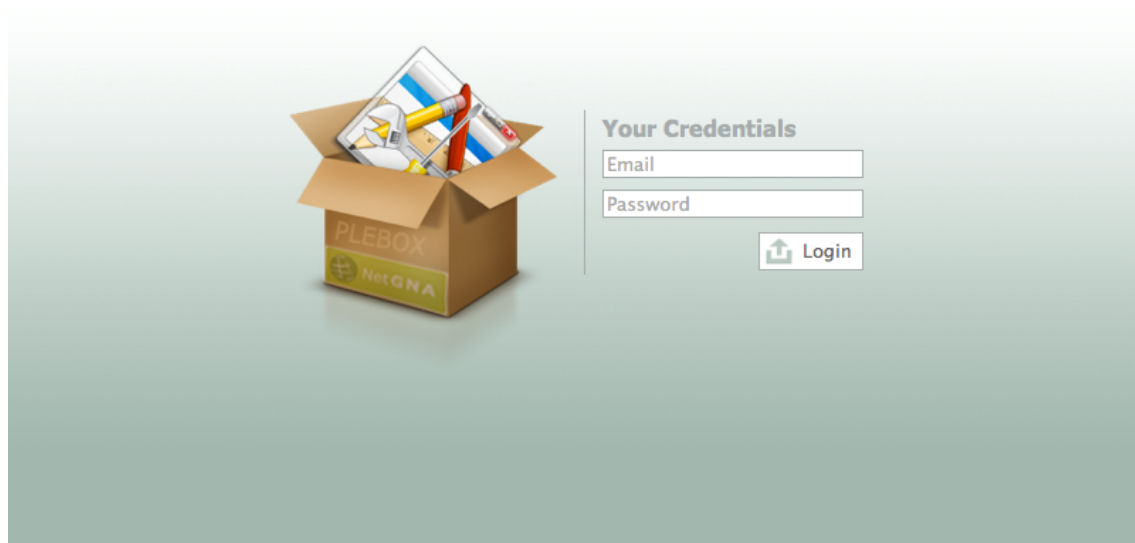


Figure 73 - Authentication User Interface Element.

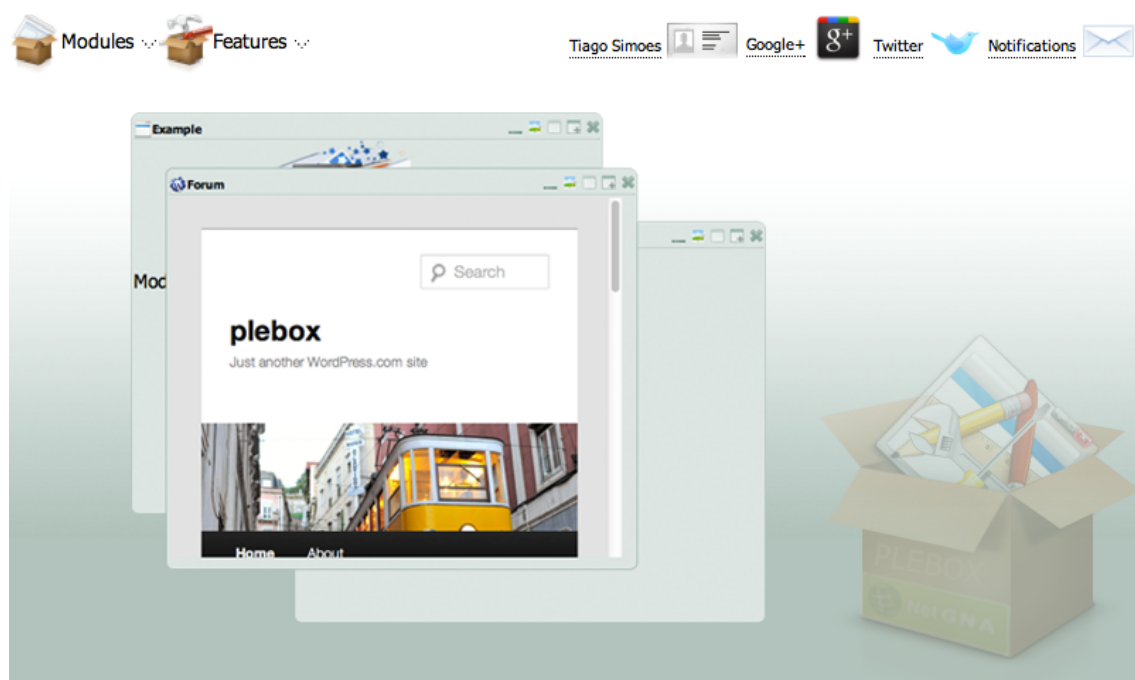


Figure 74 - Desktop User Interface Element.




Figure 75 - User Custom Desktop.


Tiago Simões 

User Profile

Name
 Last Name
 Email
 New Password
 Verification


Figure 76 - User Profile Widget.

Google+ 


Tiago Simões

 In 14 people's circles
 Latest updates

Figure 77 - Google+ Widget.

Twitter 




Microsoft
 Microsoft VIDEO: @WindowsPhone and @Nokia launch the Lumia 800 in Denmark with a bang: bit.ly/zyPx6h
 2 hours ago · reply · retweet · favorite
 Microsoft Must Have Games are back for @WindowsPhone. Check out the five new titles coming out over the next five weeks: bit.ly/wQ1JRp
 15 hours ago · reply · retweet · favorite
 Microsoft 10 ways to tune up your PC: bit.ly/wDLsFn
 18 hours ago · reply · retweet · favorite
 MSFTNext Come meet the funniest man in Microsoft Research bit.ly/Ai4Qgq
 yesterday · reply · retweet · favorite

Figure 78 - Twitter Widget.

Notifications 

User Notifications
No notifications!

Group Notifications

Notification 6	17-01-2012 14:01:00
Notification 5	17-01-2012 14:00:00
Notification 4	16-01-2012 15:50:31
Notification 3	16-01-2012 15:50:30
Notification 2	16-01-2012 15:50:29

Figure 79 - Notifications Widget.

6.1.2 Modules Management

Using the provided services and APIs a user interface was built in order to manage a number of modules that exist inside the box. In addition, this user interface uses the PLEBOX services/API to install and deploy modules. In figure 80 is shown the user interface, for administrators to manage all platform modules.

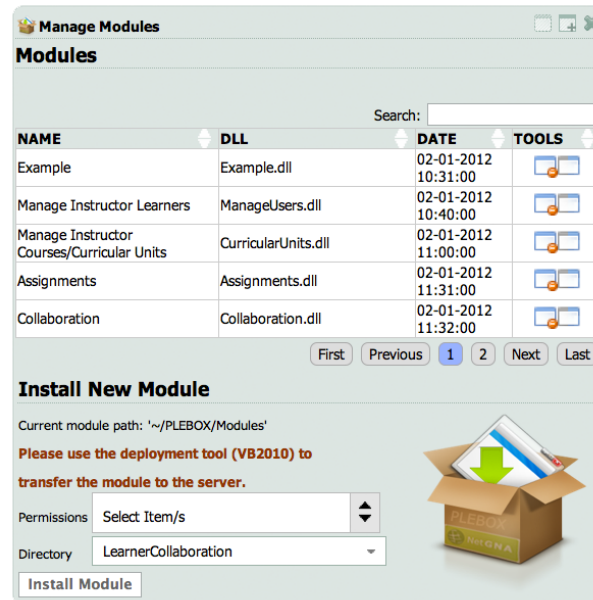


Figure 80 - Manage Modules.

6.1.3 Software Development Kit

The PLEBOX SDK, as spoken in the previous chapter, presents a set of interface objects that aim to reduce the development time of modules. In the following paragraphs we outline each individual interface object present in the SDK. As support for the developers, there is a wiki that helps developers with practical examples of how to use these interface objects.

The window interface object (Figure 81) provides a set of functions to manage windows, namely different ways to set icons (Figure 82), maximize (Figure 83) and minimize options (Figure 84), and also a set of transparency options (Figure 85). Shown in Figure 81 is an example of an encapsulated module into a window, which only has the function of displaying the message “Hello World”.

The dynamic table interface object is shown in Figure 86, this object provides a set of ways to create and deploy a group of tables and related operations such as lists, records search (Figure 87), custom CSS style management, paging, and records control (Figure 88). This is possible using a simple set of PLEBOX tags and the usage of minor code to build and deploy a custom dynamic table.

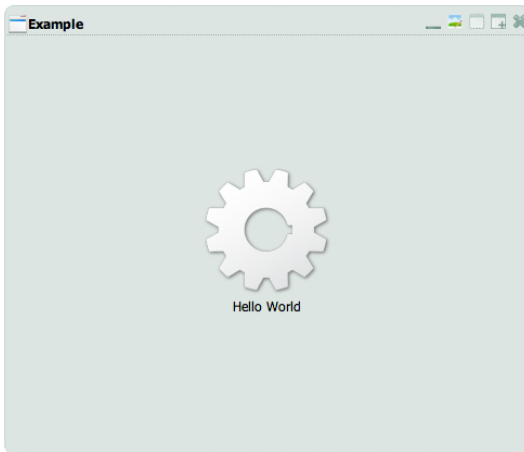


Figure 81 - Window Interface Object.

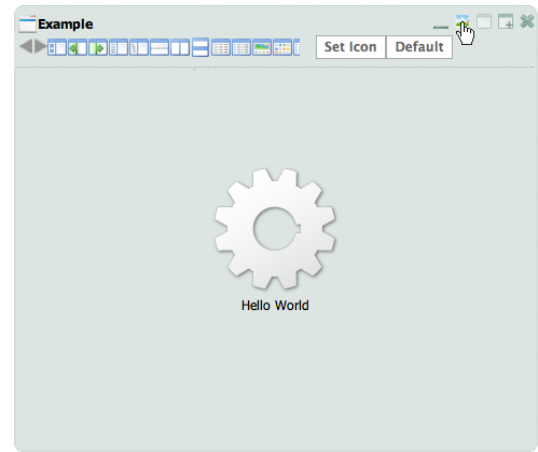


Figure 82 - Change the Icon of the Window.

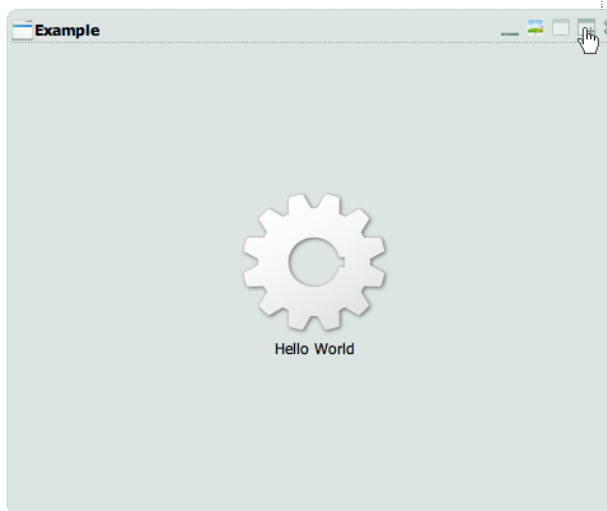


Figure 83 - Maximized Window.



Figure 84 - Window as Icon.

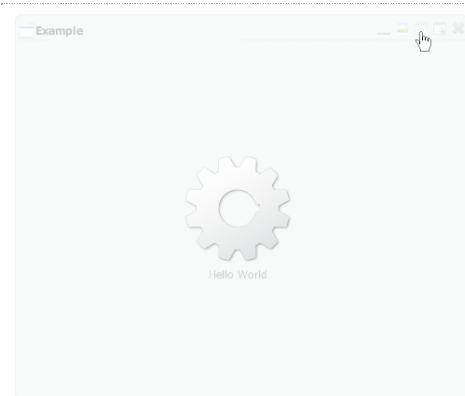


Figure 85 - Make the Window Transparent.

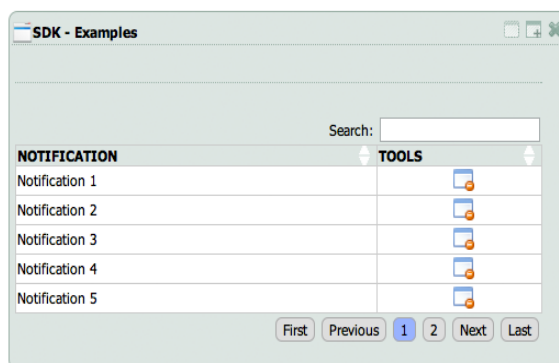


Figure 86 - Dynamic Table Interface Object.

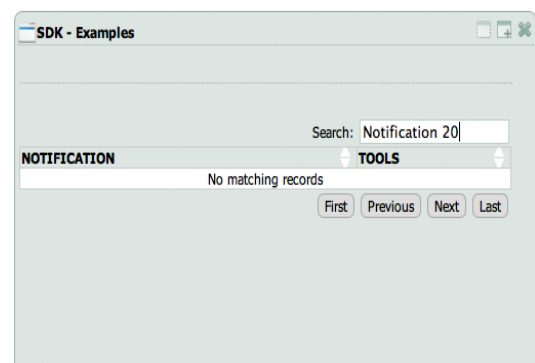


Figure 87 - Records Search.

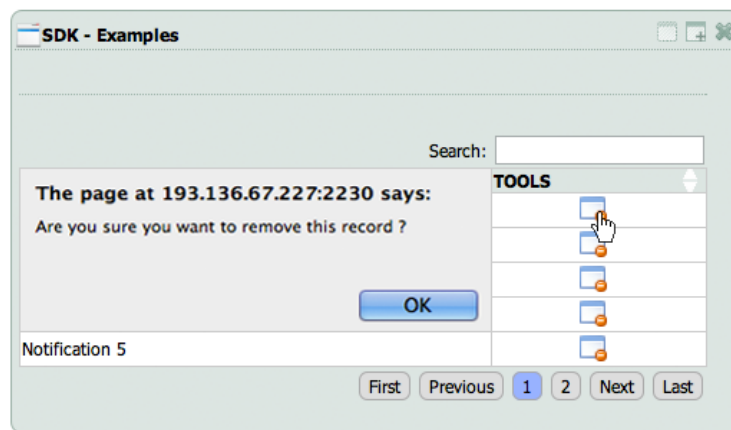


Figure 88 - Records Control.

The list interface object is shown in Figure 89 and enables an easy way to deploy a selectable customizable and manageable list. On the other hand, the upload interface object provides an interface to allow an asynchronous upload to a set of directories (Figure 90). To make all this possible developers have at their disposal a set of tags and respective parameters that allows the control and customization of these interfaces.

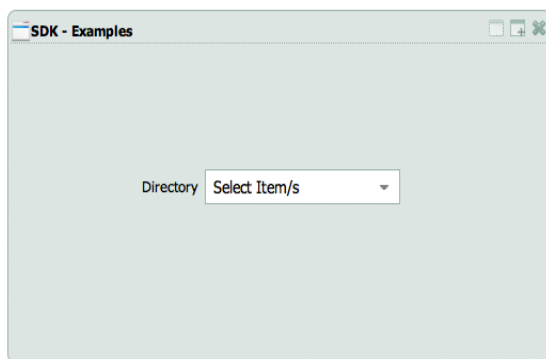


Figure 89 - List Interface Object.

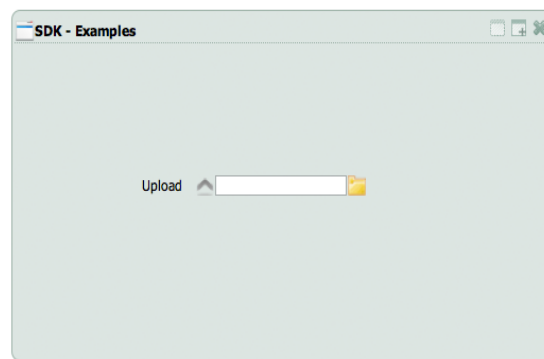


Figure 90 - Upload Interface Object.

The popup interface objects are shown in Figures 91 and 92, they are aimed to provide graphical ways of interaction with the user of the platform. The sub-object balloon and the modal window allow developers to encapsulate new functionalities to an adjacent window. One of the many usage scenarios for theses sub-object is shown in Figure 93, where in this example we have a table and we want a new set of windows to edit a particular record of the table. These objects can be used in different scenarios, like a notifications list, or even a confirmation popup to allow or deny the execution of a particular task.

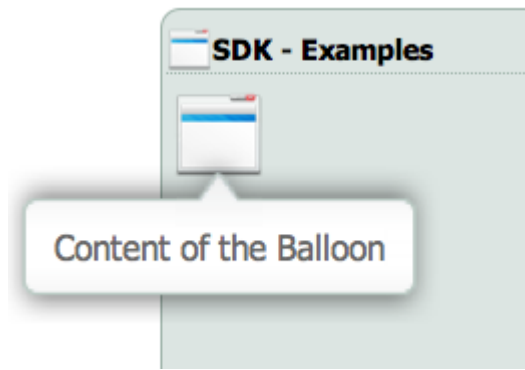


Figure 91 - Sub-Object Balloon.

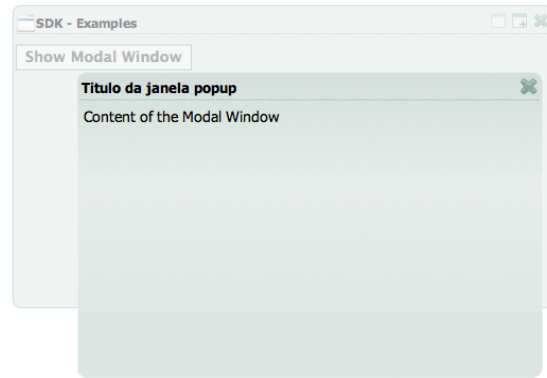


Figure 92 - Sub-Object Modal Window.

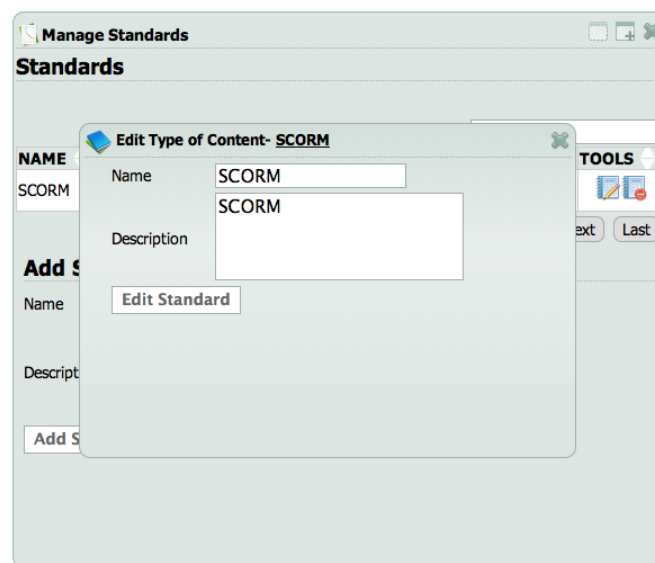


Figure 93 - Popup Interface Objects Usage Example.

The tab interface object and the color picker interface object are respectively shown in Figures 94 and 95. The color picker interface object is a simple utility that developers use to provide color management. Additionally, the tab interface object provides ways to separate content into tabs in an easy and transparent way. This interface also allows the encapsulation of common .NET controls such labels or textboxes (Figure 94, 95) and even PLEBOX SDK objects like the color picker interface object (Figure 96).

The panel interface object, shown in Figure 97, allows developers to encapsulate controls within a scroll zone. On the other hand we have the images interface object that has two sub-objects: the images slider interface sub-object, shown in Figure 98 (that uses the scroll panel to enable a list of images), and the image button interface sub-object aimed to provide a custom clickable image button (Figure 99).



Figure 94 - Tab Interface Object - Label Encapsulated.

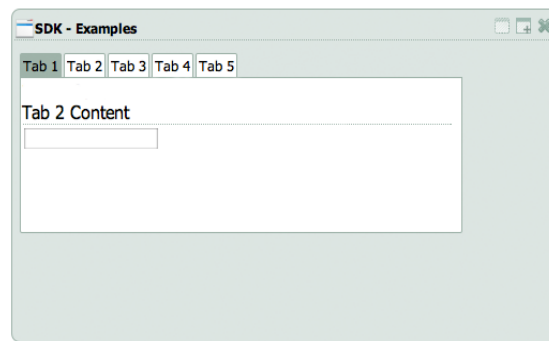


Figure 95 - Tab Interface Object - Label and Textbox Encapsulated.

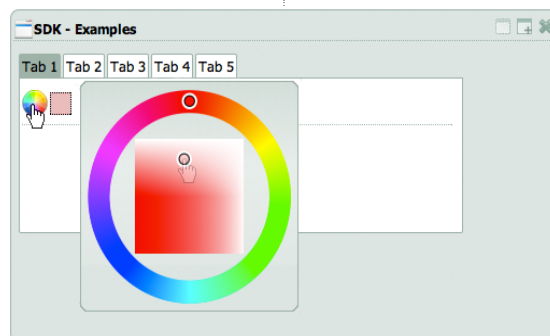


Figure 96 - Color Picker Interface Object Inside the Tab Interface Object.

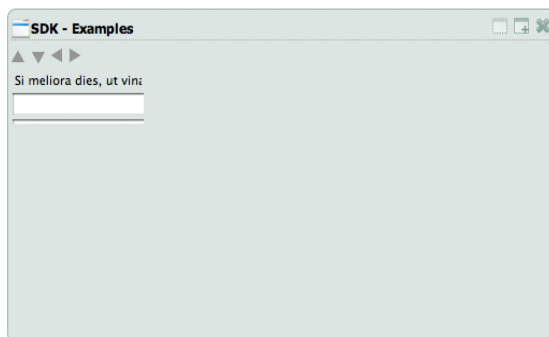


Figure 97 - Panel Interface Object.

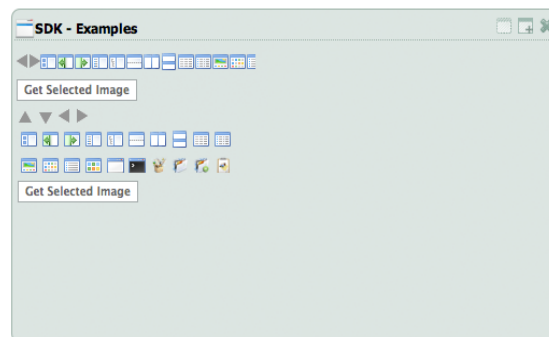


Figure 98 - Images Slider Interface Sub-Object.

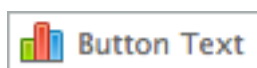


Figure 99 - Image Button Interface Sub-Object.

6.1.4 Learning Modules

In order to demonstrate learning working examples, a set of modules have been developed. These modules use the PLEBOX SDK and also the learning APIs/services with the aim of providing a set of user interfaces to manage learning features. In the following paragraphs we present each learning module and their target groups for which they are addressed,

considering that these modules have been developed only for demonstration proposes using every developed API, SDK, and service.

For users who belong to the administrator group the following modules were developed: manage users, manage courses, manage curricular units, manage content, manage content standards, manage types, manage repositories, and the module manage notifications.

The manage users learning module allows operations to list users (Figure 100), add users (Figure 101), edit users (Figure 102), and remove users. In addition to this, is possible to associated users to specific courses and also edit those associations (Figure 103). This module uses a set of learning services and the SDK's interface objects: window, tab, popup, dynamic table, and also the list interface object.

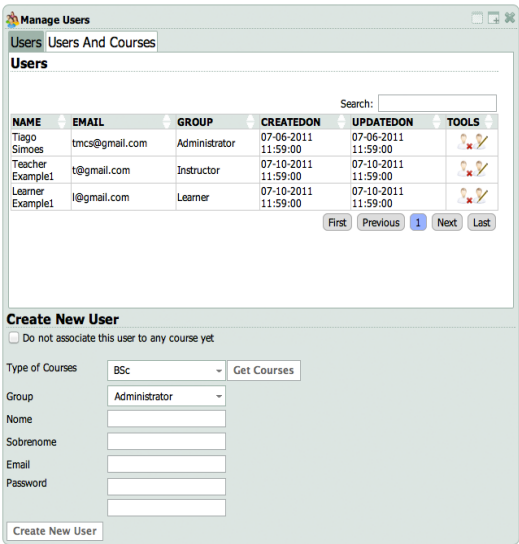


Figure 100 - List Users.

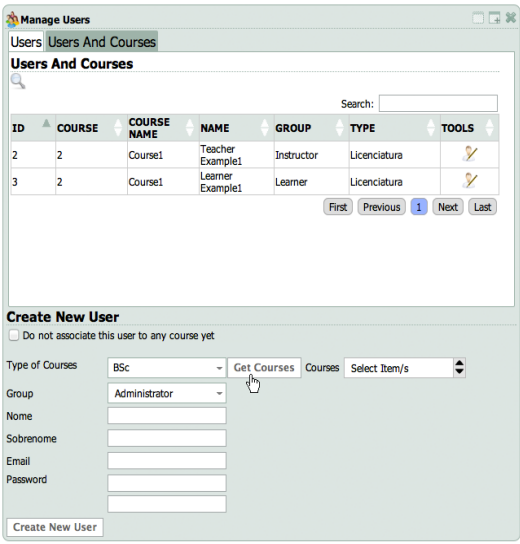


Figure 101 - Add User.

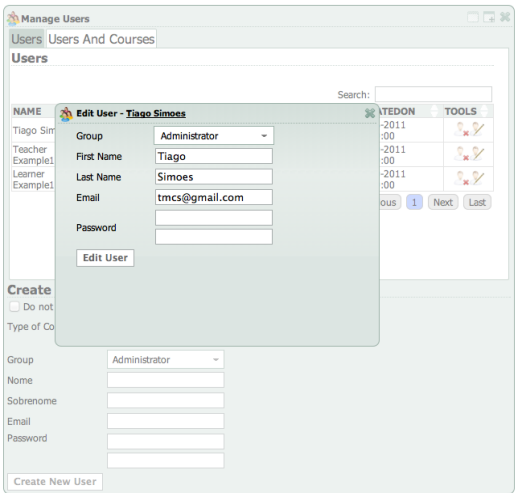


Figure 102 - Edit User Information.

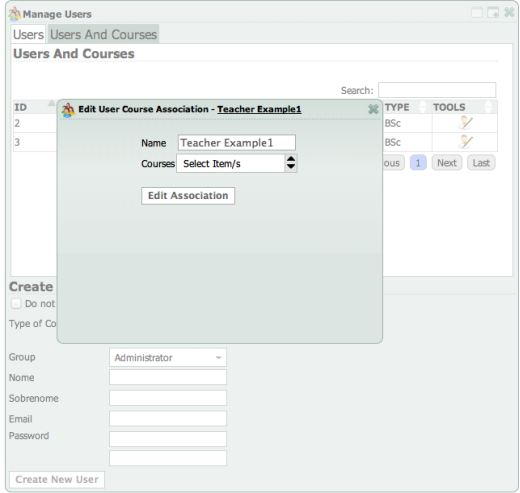


Figure 103 - Edit User Associations.

The manage courses learning module, which is shown in Figure 104, aims to provide a user interface to list, add, edit, and remove courses or types (Figure 105). This module is supported by the learning service course management and also by a number of interface objects.

The manage curricular units learning module (Figure 106) details a learning module with the objective of listing, adding, editing (Figure 107), and removing curricular units from the platform. This module uses the curricular unit service learning and also a number of interface objects present in our SDK, such as: the window interface object, the tab interface object, the popup interface object, the dynamic table, and the list interface object.

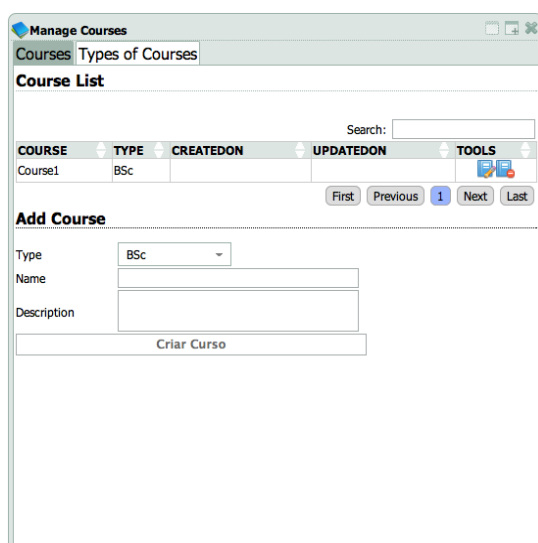


Figure 104 - Manage Courses.

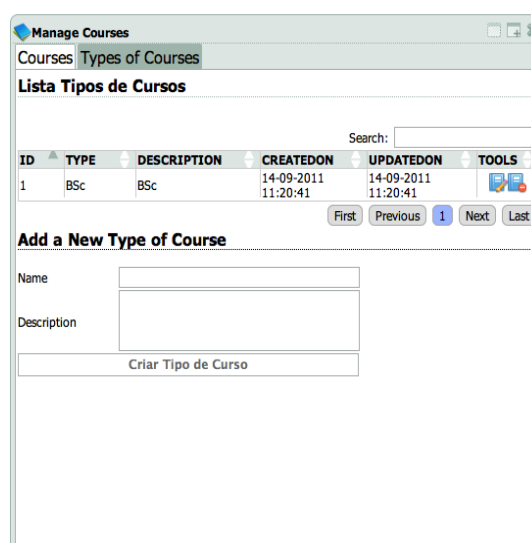


Figure 105 - Manage Types of Courses.

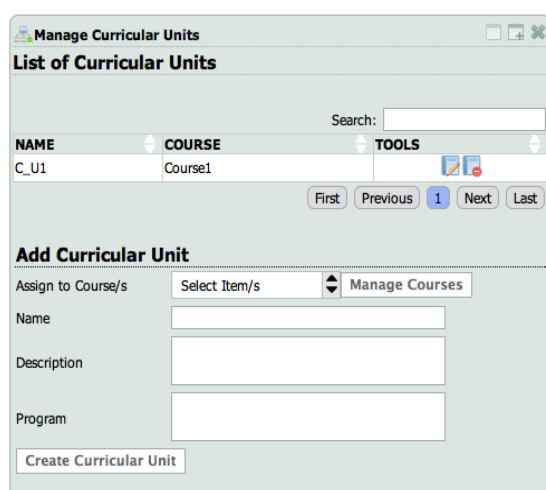


Figure 106 - Manage Curricular Units.

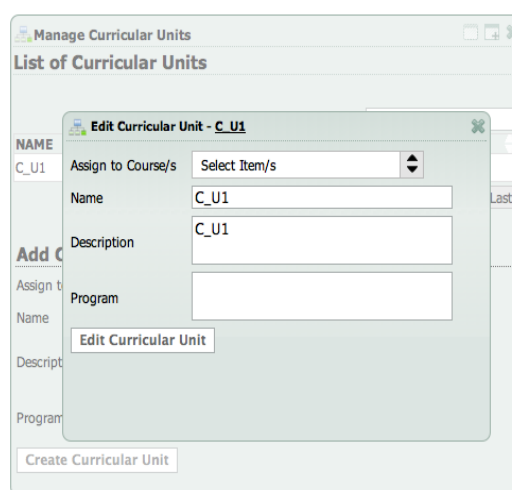


Figure 107 - Edit Curricular Unit.

The manage content learning module (Figure 108) was developed to demonstrate the capabilities of the content management learning service that administer content and related associations. This module allows the assignment of content to users or curricular units (Figure 109 and 110). Additionally, this module also delivers a set of listing options and operations to manage content (add, edit, and remove).

The manage content standards is a learning modules aimed to manage a list of standards used by the content. As shown in Figures 111 and 112 this management includes operations to list, add, edit, and remove content. This module is supported by the use of a learning service called content management and also a set of SDK interfaces objects.

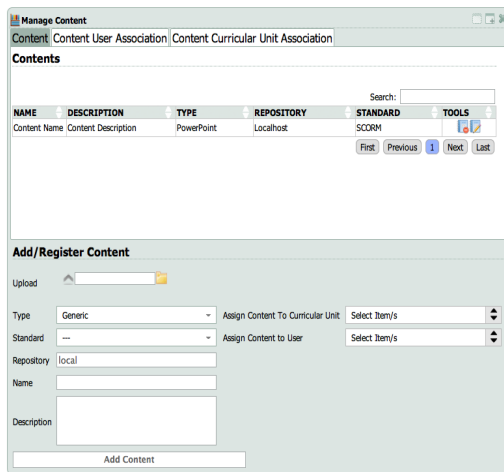


Figure 108 - Manage Content.

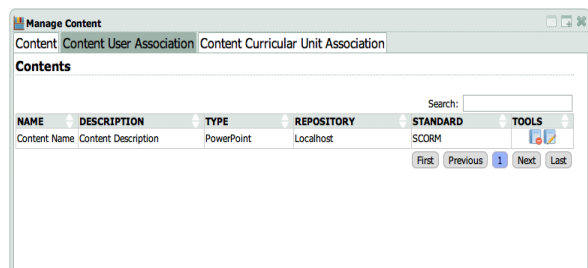


Figure 109 - Manage Content User Association.

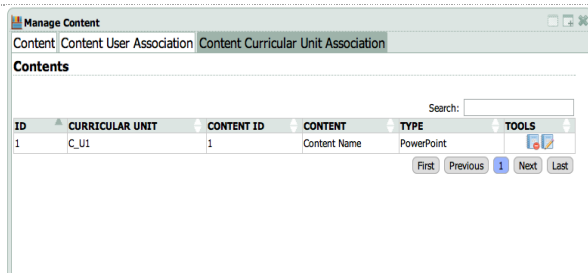


Figure 110 - Manage Content Curricular Unit Association.

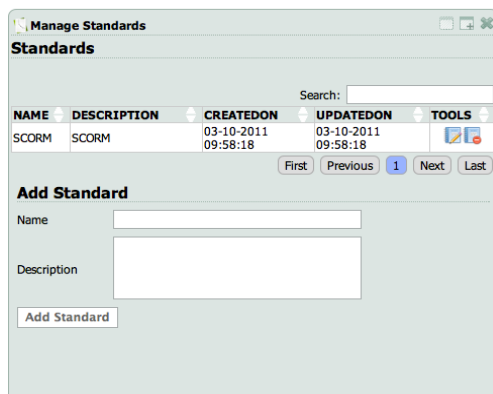


Figure 111 - Manage Content Standards.

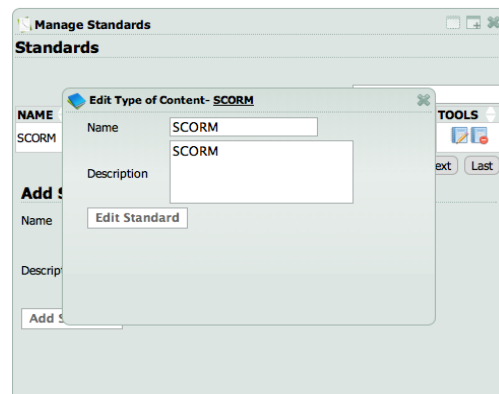


Figure 112 - Edit Standards.

The manage types is a learning module that details the management of types of courses. As shown in Figures 113 and 114, this module provides operations to list, add, edit, and remove types of courses. This module uses the content management learning service and also a number of SDK interface objects.

The manage notifications that is shown in Figure 115, is a learning module aimed to manage a set of notifications, enabling the possibility to sent notifications to users or groups. This module uses the learning service to handle notifications, addition to this it also uses the PLEBOX SDK interfaces such as: the window interface object, the tab interface object, the popup interface object, the dynamic table, and finally the list interface object.

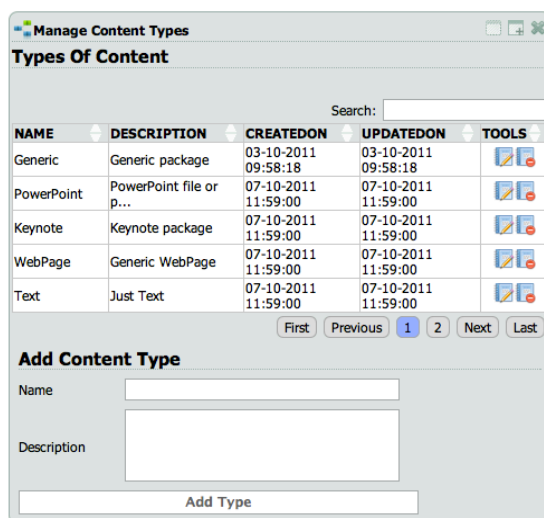


Figure 113 - Manage Types.

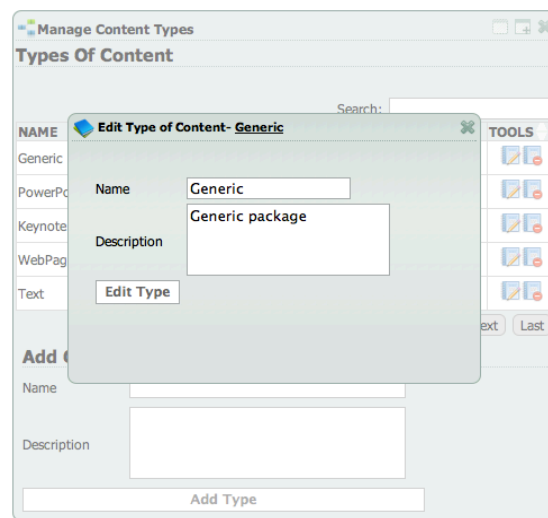


Figure 114 - Edit Type.

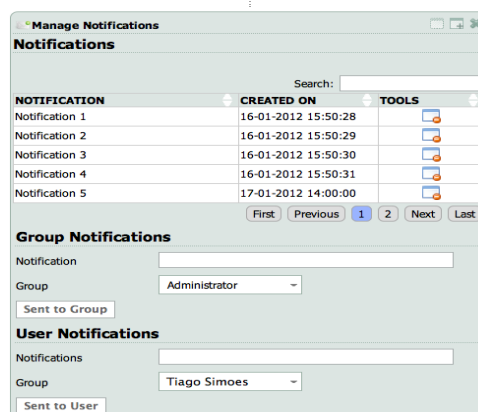


Figure 115 - Manage Notifications.

The manage repositories learning module was developed with the aim of managing and register information about local or external repositories (Figures 116 and 117). This module uses the repositories management learning service and also our SDK for deploying a user-friendly user interface.

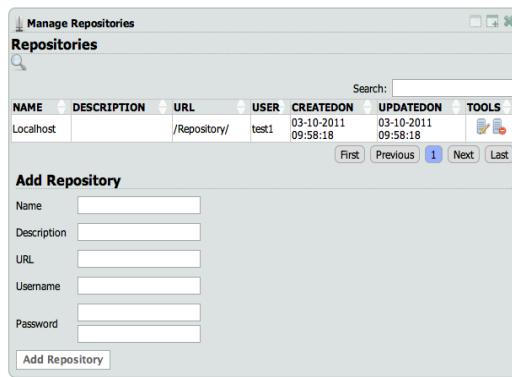


Figure 116 - Manage Repositories.

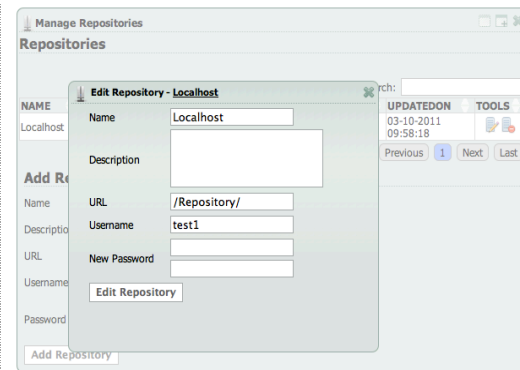


Figure 117 - Edit Repository.

Another group of modules are those that are directed for users belonging to the instructors group. For this users the following modules were developed: manage instructor courses and curricular units, manage learners, and the instructor's collaboration module.

The manage courses and curricular units learning module (Figure 118) was developed with the aim of providing a set of operations to assign (Figure 119), list (Figure 120), view (Figure 121), edit (Figure 122), and remove content from a set of curricular units. This module is supported by the use of a learning service called content management and also a set of SDK interface objects.

The manage learners learning module (Figure 123) aims to show information about learners that are attending a set of instructor's curricular units (Figure 124). This module uses a set of learning services, in way to know what users are enrolled into the instructor's curricular units.

The instructor's collaboration learning module (Figure 125) aims to provide a set of operations to communicate with learners that are enrolled with a set of curricular units that the instructor is teaching. This module provides an interface to list instructor's notifications and also an interface to send notifications to one or more users that are enrolled with a particular curricular unit (Figure 126).

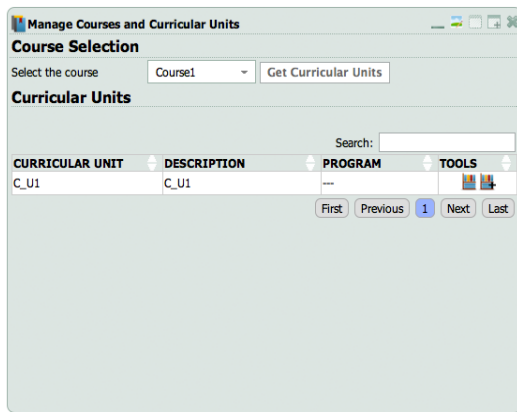


Figure 118 - Manage Courses and Curricular Units.

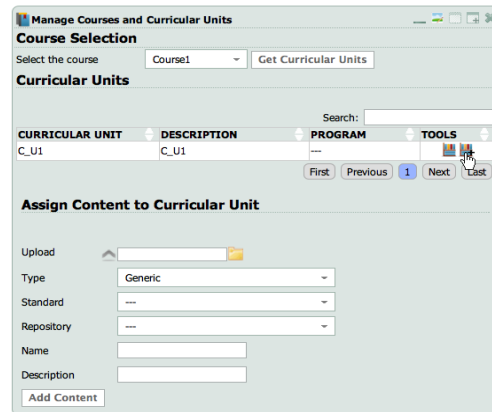


Figure 119 - Assign Content to Curricular Unit.

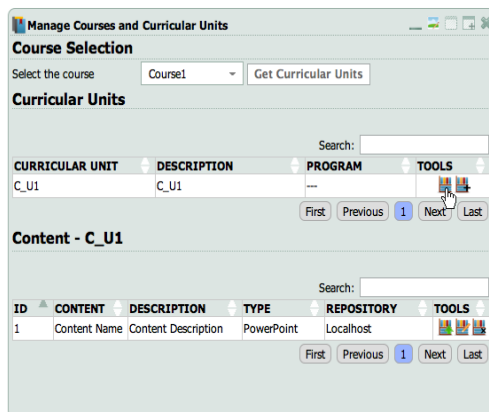


Figure 120 - View Contents of the Selected Curricular Unit.

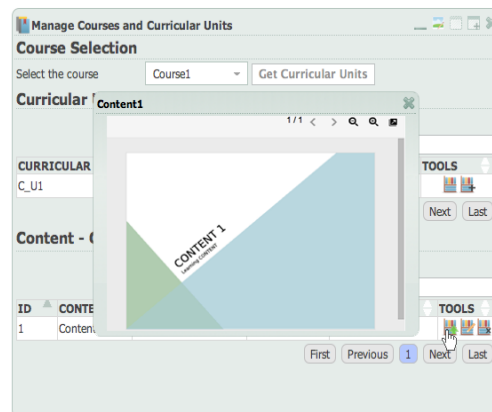


Figure 121 - View Content.

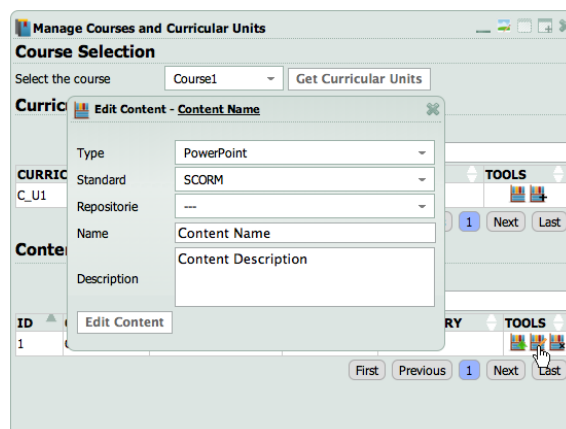


Figure 122 - Edit Content.

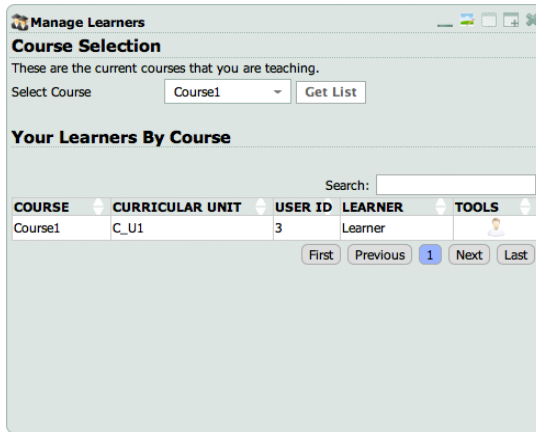


Figure 123 - Manage Learners.

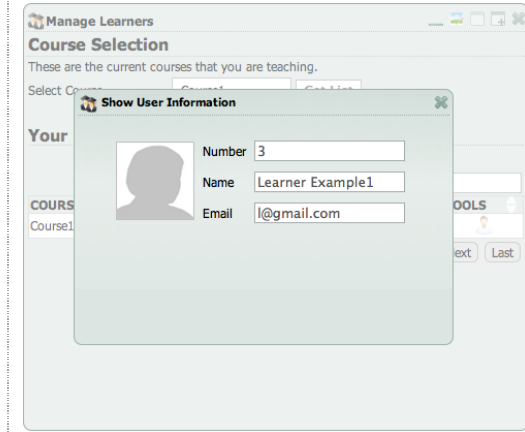


Figure 124 - View Learner Information.

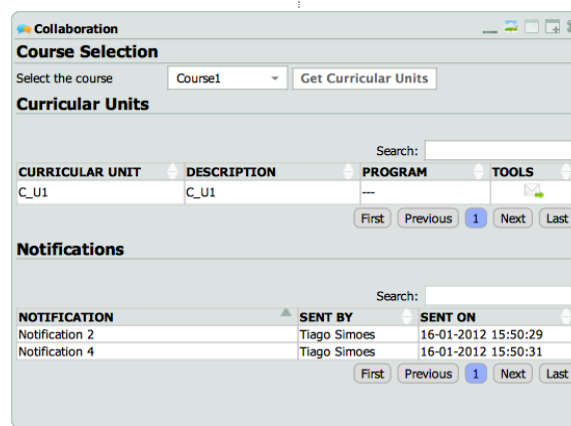


Figure 125 - Instructor's Collaboration.

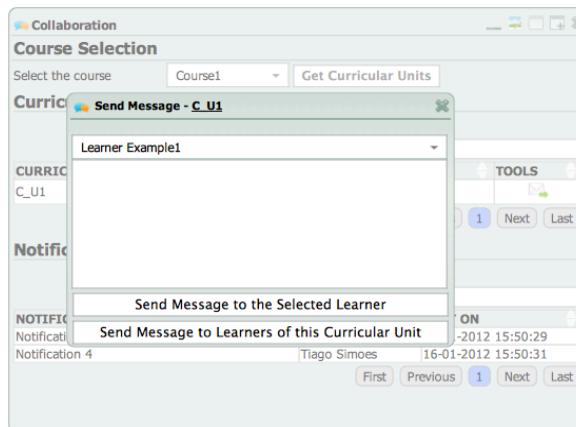


Figure 126 - Send Notification(s) to Learner(s).

Two modules were developed for the learners group, the assignments learning module and the learner's collaboration module.

The assignments learning module (Figure 127) was developed with the aim of deploying a set of operations to list and perform a set of assigned contents/tasks that are associated with a

set of curricular units (Figure 128). This module uses the content management learning service and interface objects of the PLEBOX SDK to provide a user-friendly interface.

The last module is the learner's collaboration module (Figure 129) that aims to provide a set of operations to communicate with instructors that are enrolled with a set of curricular units that the learner is attending. This module provides an interface to list learner's notifications and also an interface to send notifications to instructors, which are enrolled with a selected curricular unit (Figure 130).

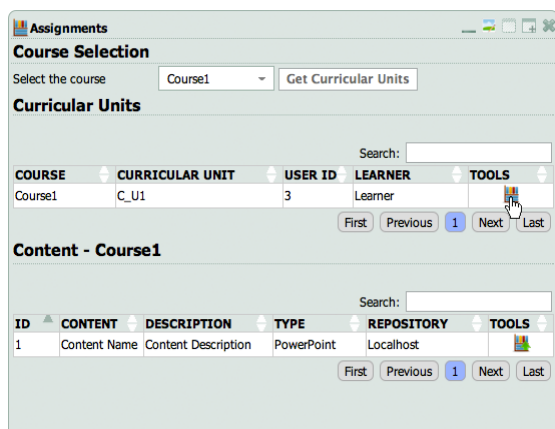


Figure 127 - Assignments.

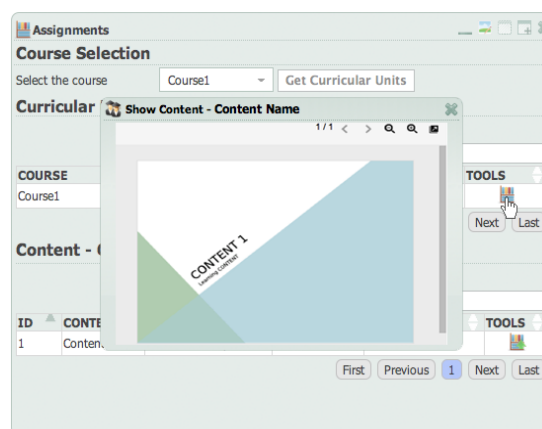


Figure 128 - Perform Learning Task.

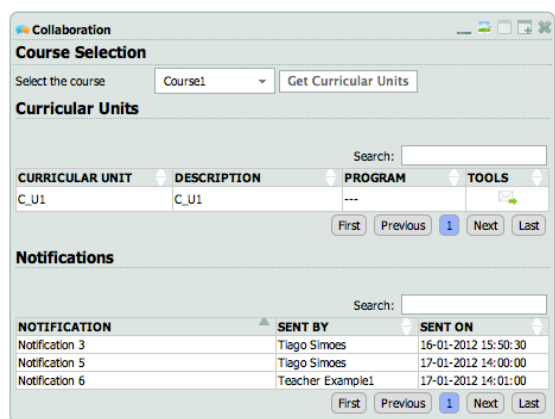


Figure 129 - Learner's Collaboration.

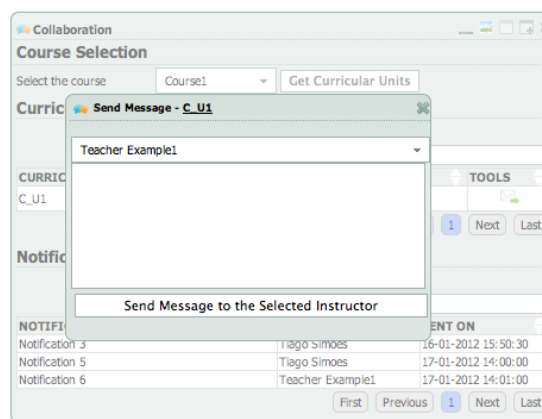


Figure 130 - Send Notification(s) to Instructor.

6.2 Aggregation Examples

In this section we present a set of examples to demonstrate that our services can be integrated, using different technologies, into multiple environments. These examples not represent in any a way a finished product instead they are intended to show the level of abstraction of our services.

6.2.1 Example 1 - ASP.NET Web Application

The first example was developed using primarily a set of .NET technologies and has the purpose of showing how simple it is to use a PLEBOX service (e.g. the module service). This .NET Web application example, shown in Figure 131, has the primary function of providing a list of previously register modules, which can be used by a user that was previously defined in the application code.

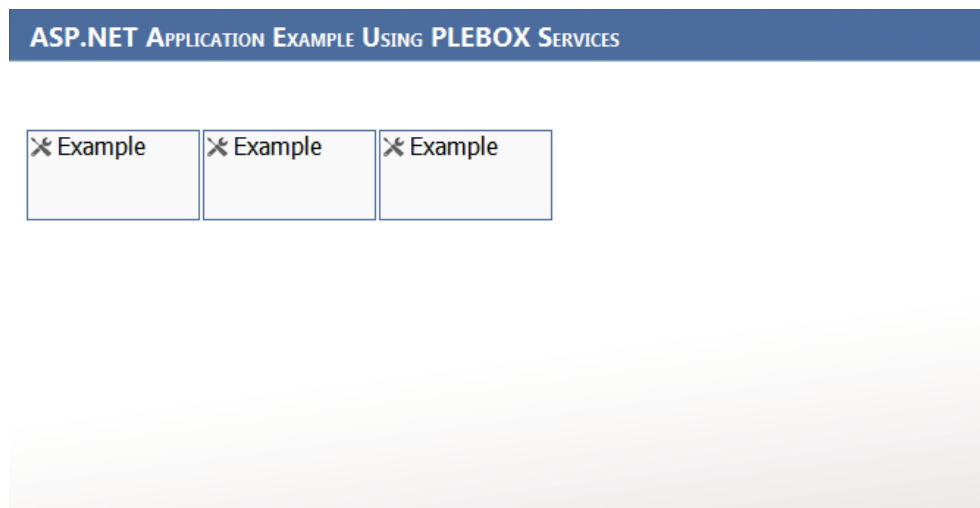


Figure 131 - Example 1 (ASP.NET Web Application).

6.2.2 Example 2 - Web-Based JavaScript Application

The second example was developed using JavaScript and HTML technologies. This example, shown in Figure 132, is characterized by the use of learning services, such as: the user management services for authenticating the user (Figure 133), the module services to manage modules of the current user, and the menu services to allow the creation of a clickable list to provide ways to initialize modules into the environment (Figure 134).

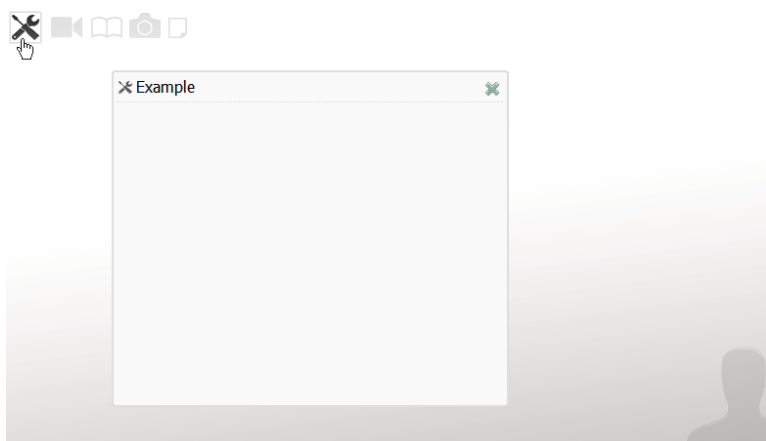


Figure 132 - Example 2 (Web-Based JavaScript Application).

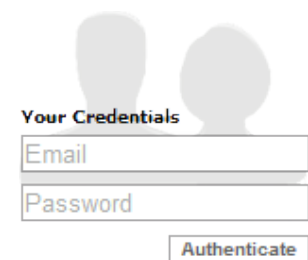


Figure 133 - Authentication.



Figure 134 - Modules List.

6.2.3 Example 3 - Android Application

The third example was developed for the Android OS, with the aim of providing a list of contents to be performed by learners. This Android application, shown in Figure 135, is called Assignments and aims to demonstrate a set of PLEBOX learning services being used in a mobile environment. This example uses the user learning services for authenticating the user, (Figure 136), and the content learning services to receive a list of contents that were assigned to the user (Figure 137). In addition, and if the device supports the source of the assignment, the learner will be able to perform or view the selected content (Figure 138).



Figure 135 - Example 3
(Android Application).



Figure 136 - Mobile Login.



Figure 137 - Assignments List.



Figure 138 - Viewing Content.

Chapter 7 - Conclusions and Future Work

This dissertation described the proposed personal learning environment box (PLEBOX) platform and the creation, extension, and inclusion of a number of learning services into the platform. In addition, this dissertation described the work performed to create a range of solutions to support the development of modules and internal components. These solutions were proposed on the sense of solving problems encountered in the related literature about the topic.

Chapter one presented the focus of the dissertation including a perspective of the e-learning topic and related issues. The objectives and main contributions were presented, and the document organization was also included.

Chapter two addressed the related work about e-learning, e-learning platforms, personal learning environments, and rich Internet applications. This chapter introduced the origins of the Web and, therefore, the concept of e-learning. The evolution of e-learning, its importance and related problems in the literature were discussed. A set of definitions to explain and cluster a range of definitions applied to a number of different learning solutions, including related concepts like learning standards and learning objects were identified. Additionally, a number of commercial and open source learning platforms were identified in order to include a discussion about main features of e-learning solutions. Finally, the concept of personal learning environments, problems, platforms, rich Internet applications, and RIA frameworks were included.

Chapter three explored the requirements analysis to determinate the features and interactions of the proposed system. This chapter details the user cases for a range of proposed PLEBOX services, APIs, SDK, and learning and non-learning modules. Additionally, the database ER diagram and the class diagram of the core components of the system were included, detailing the foundations of the learning and non-learning services/APIs provided by the platform.

Chapter four introduced the PLEBOX platform. A set of PLEBOX operations principles, including a number of contributions to developers and users were demonstrated, that includes a set of tools that allow the creation of learning and management modules. Additionally, the main characteristic of each version was described, including a set of problems that led the platform to new versions. Finally, used technologies, database views, stored procedures, architectures, and developed scenarios detailing the exposure of concepts that allow the deployment of features and components were presented.

Chapter five described the ideas that drove the development of a number of services, APIs, and SDKs. In this chapter a number of mockup ideas behind the construction of a range of

learning and non-learning APIs, services, and SDKs were presented. These components can be used not only inside the internal environment of the platform but also outside of the PLEBOX environment.

Chapter six presented the demonstration and validation of the platform, modules, and development resources. The platform usage, namely the PLEBOX user interface main components, the modules management, the SDK interface objects and the learning modules were demonstrated. Finally, three case studies were demonstrated, with the objective to show a number of learning services working in an external environment.

E-learning has been evolving due to necessity of encapsulate a set of ideas and technologies that have been emerging in the context of Web 2.0. In addition, the appearance of new learning ideologies (e.g. learner-centered learning approaches) has led to the need of providing new models embedded into a new platform. These ideas led to the fact that many of the current available systems are inflexible and outdated, and even those who are already using new learner-centered learning approaches ideologies and technologies continue to include them, as patches, into learning solutions that were previously based on old technologies and principles.

The PLEBOX platform was created from scratch bearing in mind these new ideologies and Web 2.0 technologies. This platform creates a diverse set of desktop-like learning environments, providing learning and administration modules. The PLEBOX can be seen as a model for service usage in and outside of the platform, meaning that is possible to use a set of PLEBOX learning services into an internal or external environment (e.g. into a mobile application). Finally, a group of internal APIs and SDKs helps developers to construct a group of must-have learning and administration tools.

The proposed objectives, including the design and creation of a range of application programming interfaces (APIs) for module development, and also the design and implementation of range of building components that form the foundations of an LMS platform were largely overcome. In addition, a number of improvements were created, namely a set of features and services developed in a service-driven model providing the ability to incorporate a set of learning features into external environments. Besides all these add-ons, a number of SDK interface objects, for modules and internal components, were developed. These interface objects are aimed to reduce the time expended in the developed of common user interfaces, like tables, popups, buttons, and other major interface tools. Finally, a main user interface, a range of learning/administration modules, and three aggregation examples were constructed in order to demonstrate the capabilities of our services, APIs, and SDKs. Exhaustive experiments were driven with success and it is ready for real use.

As a suggestion for future work the current SDK version of the PLEBOX platform should be improved with more controls and re-analyzed to be independent of the PLEBOX main user interface. Additionally, the design and development of a set of services in order to provide interactions with available social networks and cloud storage platforms is suggested. Finally, the development of modules for others external environments (e.g. Moodle) using a set of PLEBOX administration and learning services may be proposed.

References

- [1] K. Chassie, "The allure of e-learning", in IEEE Potentials, vol. 21 (3), pp. 33-35, August 2002.
- [2] J. Rodrigues and F. Sabino, "Enhancing e-learning experience with online social networks", in IET Communications, vol. 5 (8), pp. 1147-1154, May 2011
- [3] D. Zhang, J. L. Zhao, L. Zhou and J. F. Nunamker Junior, "Can e-learning replace classroom learning?", in Communications of the ACM, vol. 47 (5), pp. 75-79, May 2004.
- [4] M. Llamas-Nistal, M. Caeiro-Rodriguez and M. Castro, "Use of E-Learning Functionalities and Standards: The Spanish Case", in IEEE Transactions on Education, vol. 54 (4), pp. 540-549, November 2011.
- [5] S. Rapuano, and F. Zoino, "A Learning Management System Including Laboratory Experiments on Measurement Instrumentation", in 2005 IEEE Instrumentation and Measurement Technology, Ottawa, Canada, May 16-19, 2005, pp. 1227-1232.
- [6] H. Zhao, L. Yang and Y. Wang, "The personal learning environment (PLE) based on web 2.0", in 2010 IEEE 2nd Symposium on Web Society (SWS), Beijing, China, August 16-17, 2010, pp. 22-25.
- [7] J.H. Liu, B.S. Huang and M. Chao, "The design of learning object authoring tool based on SCORM", in 5th IEEE International Conference on Advanced Learning Technologies, Santander, Spain, July 5-8, 2005, pp. 778-782.
- [8] J. Ma and D. Zhou, "Fuzzy set approach to the assessment of student-centered learning", in IEEE Transactions on Education, vol. 43 (2), pp. 237-241, May 2000.
- [9] D. Dagger, A. O'Connor, S. Lawless, E. Walsh and V.P. Wade, "Service-Oriented E-Learning Platforms: From Monolithic Systems to Flexible Services", in IEEE Internet Computing, vol. 11 (3), pp. 28-35, May-June 2007.
- [10] S. Murugesan, "Understanding Web 2.0", in IT Professional, vol. 9 (4), pp. 34-41, July-Aug 2007.

- [11] N. Capuano, A. Gaeta, A. Marengo, S. Miranda, F. Orciuoli and P. Ritrovato (2009), "Creation and Delivery of Complex Learning Experiences: the ELeGI Approach", in 2009 International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), Fukuoka, Japan, March 16-19, 2009, pp. 239-246.
- [12] S. M. Arachchi, M. Siriwardena, R. Madanayake, and K. Dias, "Identification of learning styles and learning domains in Sri Lanka in the development of e-learning content", in 2010 International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, September 29-30, 2010, pp. 50-55.
- [13] S.H.A Hamid and T.H. Chuan, "Designing Learning Styles Application of E-Learning System Using Learning Objects", in Conference Advances in Web Based Learning - ICWL, Penang, Malaysia, July 19-21, 2006, pp. 81-92.
- [14] T. M. C. Simões, J. J. P. C. Rodrigues, and I. de la Torre, "Personal Learning Environment Box (PLEBOX): A new approach to E-learning platforms", in Computer Applications in Engineering Education, 2011.
- [15] S. Wu, C. Song, H. Chen, and J. Zhan, "E-learning Teaching Method Research and Ideas", in 2010 International Conference on Web Information Systems and Mining (WISM), Sanya, China, October 23-24, 2010, pp. 140-143.
- [16] A. P. Oliveira, Ó. Mealha and C. Santos, "Visualization of Web Based e-Learning Activity", in 14th International Conference Information Visualization, London, United Kingdom, July 27-29, 2010, pp. 219-224.
- [17] N. A. Rahman and S. Sahibuddin, "Social interaction in e-learning: An overview", in 2010 International Symposium on Information Technology, Kuala Lumpur, Malaysia, June 15-17, 2010, pp. 1-4.
- [18] N. Zhang and H. Bao, "Research on E-learning with Digital Technology in Distance Education", in 2010 International Conference on e-Education, e-Business, e-Management and e-Learning, Sanya, China, January 22-24, 2010, pp. 299-302.
- [19] H. Rego, T. Moreira, F. Garcia and E. Morales, "Metadata and knowledge management driven web-based learning information system", in international Journal of Technology Enhanced Learning, vol. 1 (3), p. 215, 2009.

- [20] S. Eide, T. Kristensen and Y. Lamo, "A model for dynamic content based e-learning systems", in the 2008 Euro American Conference, New York, USA, September 10-12, 2008, pp. 1-8.
- [21] S. Bergstedt, S. Wiegrefe, J. Wittmann and D. Moller, "Content Management Systems and e-Learning-Systems - A Symbiosis?", in 3rd IEEE International Conference on Advanced Learning Technologies, Athens, Greece, July 9-11, 2003, pp. 155-159.
- [22] M. Iskander, "Analysis and Prospect on the E-Learning System Instructional Design", in International Symposium on Information Science and Engineering, Shanghai, China December 20-22, 2008, pp. 303-306.
- [23] W. Jun and L. Xiao, "The Levels and Realization of E-learning Platforms", in 1st International Workshop on Education Technology and Computer Science, Wuhan, China, March 7-8, 2009, pp. 234-238.
- [24] C. J. Costa and M. Aparicio, "Analysis of e-learning processes", in the 2011 Workshop on Open Source and Design of Communication, Lisbon, Portugal, July 11-11, 2011, p. 37.
- [25] Chun-Chia Wang, Wen-Chang Pai and N.Y Yen, "A sharable e-Learning platform based on Cloud computing", in 2011 3rd International Conference on Computer Research and Development (ICCRD), Shanghai, China, March 11-13, 2011, pp. 1-5.
- [26] Young-Kyun Baek, Hyun-Jung Cho and Bo-Kyeong Kim, "Uses of learning objects in a wireless Internet based learning system", in International Conference on Computers in Education (ICCE), Auckland, New Zealand, December 3-6, 2002, pp. 898-900.
- [27] V. Devedzic, J. Jovanovic and D. Gasevic, "The Pragmatics of Current E-Learning Standards", in IEEE Internet Computing, vol. 11 (3), pp. 19-27, May-June 2007.
- [28] SCORM. <http://scorm.com/scorm-explained/>, accessed on April 27, 2012.
- [29] J. Liu, H. Men and J. Han, "Comparative Study of Open-Source E-Learning Management Platform", in 2009 International Conference on Computational Intelligence and Software Engineering, Wuhan, China. December 11-13, 2009, pp. 1-4.

- [30] S. Kumar, A. K. Gankotiya and K. Dutta, "A comparative study of moodle with other e-learning systems", in 2011 3rd International Conference on Electronics Computer Technology (ICECT), Kanyakumari, India, April 8-10, 2011, pp. 414-418.
- [31] S. Graf, Kinshuk, and Tzu-Chien Liu, "Identifying Learning Styles in Learning Management Systems by Using Indications from Students Behaviour", in 2008 8th IEEE International Conference on Advanced Learning Technologies, Santander, Spain, July 1-5, 2008, pp. 482-486.
- [32] L. Wen and C. Jesshope, "A general learning management system based on schema-driven methodology", in IEEE International Conference on Advanced Learning Technologies, Joensuu, Finland, August 30 - September 1, 2004, pp. 633-635.
- [33] A. I. Albarrak, H. A. Aboalsamh and M. Abouzahra, "Evaluating learning management systems for University medical education", in 2010 International Conference on Education and Management Technology (ICEMT), Cairo, Egypt, November 2-4, 2010, pp. 672-677.
- [34] N. R. Reyes, P. V. Candeas, S. G. Galan, R. Viciano, F. Canadas and P. J. Reche, "Comparing open-source e-learning platforms from adaptivity point of view", in 2009 European Association for Education in Electrical and Information Engineering (EAEIE) Annual Conference, Valencia, Spain, June 22-24, 2009, pp. 1-6.
- [35] Q. Li, R. W. H. Lau, T. K. Shih and F. W. B. Li, "Technology supports for distributed and collaborative learning over the internet", in ACM Transactions on Internet Technology, vol. 8(2), pp. 1-24, February 2008.
- [36] H. Li, Q. Liu and Q. Wang, "The Research on Permission Management of the Open-source Sakai Platform", in 2009 International Conference on Industrial and Information Systems (IIS), Haikou, China, April 24-25, 2009, pp. 66-69.
- [37] S. Arnold and J. Fisler, "OLAT: The Swiss Open Source Learning Management System", in 2010 International Conference on e-Education, e-Business, e-Management and e-Learning, Sanya, China, January 22-24, 2010, pp. 632-636.
- [38] P. Bradford, M. Porciello and N. Balkon, "The Blackboard Learning System: The Be All and End All in Educational Instruction?", in Journal of Educational Technology Systems, vol. 35 (3), pp. 301-314, 2007.

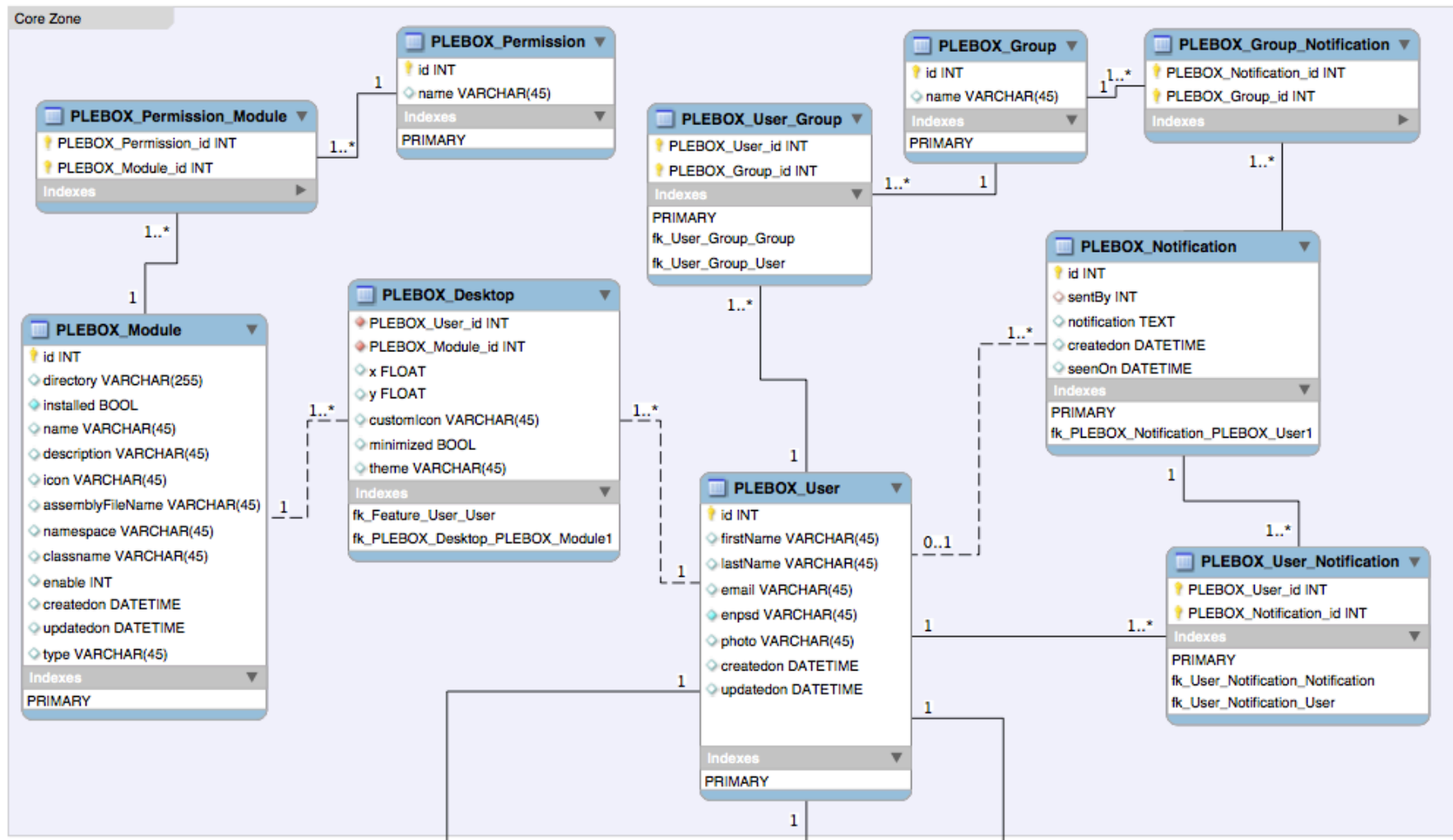
- [39] Joomla LMS. <http://www.joomlaims.com/>, accessed on April 28, 2012.
- [40] Saba LMS. <http://www.saba.com/>, accessed on April 28, 2012.
- [41] A. Soumplis, E. Chatzidaki, E. Koulocheri and M. Xenos, "Implementing an Open Personal Learning Environment", in 2011 15th Panhellenic Conference on Informatics (PCI), Kastonia, Greece, September 30-October 2, 2011, pp. 345-349.
- [42] Y. D. Wang and N. Zahadat, "Teaching web development in the web 2.0 era", in the 10th ACM conference on SIG-Information Technology Education (SIGITE), New York, USA, October 22-24, 2009, p. 80.
- [43] A. Mikroyannidis, "Supporting Self-Regulated Learning within a Personal Learning Environment: The OpenLearn Case Study", in 2011 11th IEEE International Conference on Advanced Learning Technologies (ICALT), Athens, Greece, July 6-8, 2011, pp. 607-608.
- [44] F. D. Cerbo and G. Doderio, "Bridging the Gap between PLE and LMS", in 2011 11th IEEE International Conference on Advanced Learning Technologies (ICALT), Athens, Greece, July 6-8, 2011, pp. 142-146.
- [45] S. J. Green, E. J. Pearson and S. Gkatzidou, "Formal specification of an adaptable personal learning environment using prolog", in the 1st ACM SIGMM international workshop on Media Studies and Implementations that Help Improving Access to Disabled Users, Beijing, China, October 19-24, 2009, pp. 29-38.
- [46] M. Van Harmelen, "Personal learning environments", in the 6th International Conference on Advanced Learning Technologies, Kerkrade, Netherlands, July 5-7, 2006, pp. 815-816.
- [47] F. Henri, B. Charlier, "Personal learning environment: A concept, an application, or a self-designed instrument?", in 2010 9th International Conference on Information Technology Based Higher Education and Training (ITHET), Cappadocia, Turkey, April 29-May 1, 2010, pp. 44-51.
- [48] M. Van Harmelen, "Design trajectories: four experiments in PLE implementation", in Interactive Learning Environments, pp. 16 (1), pp. 35-46, April 2008.

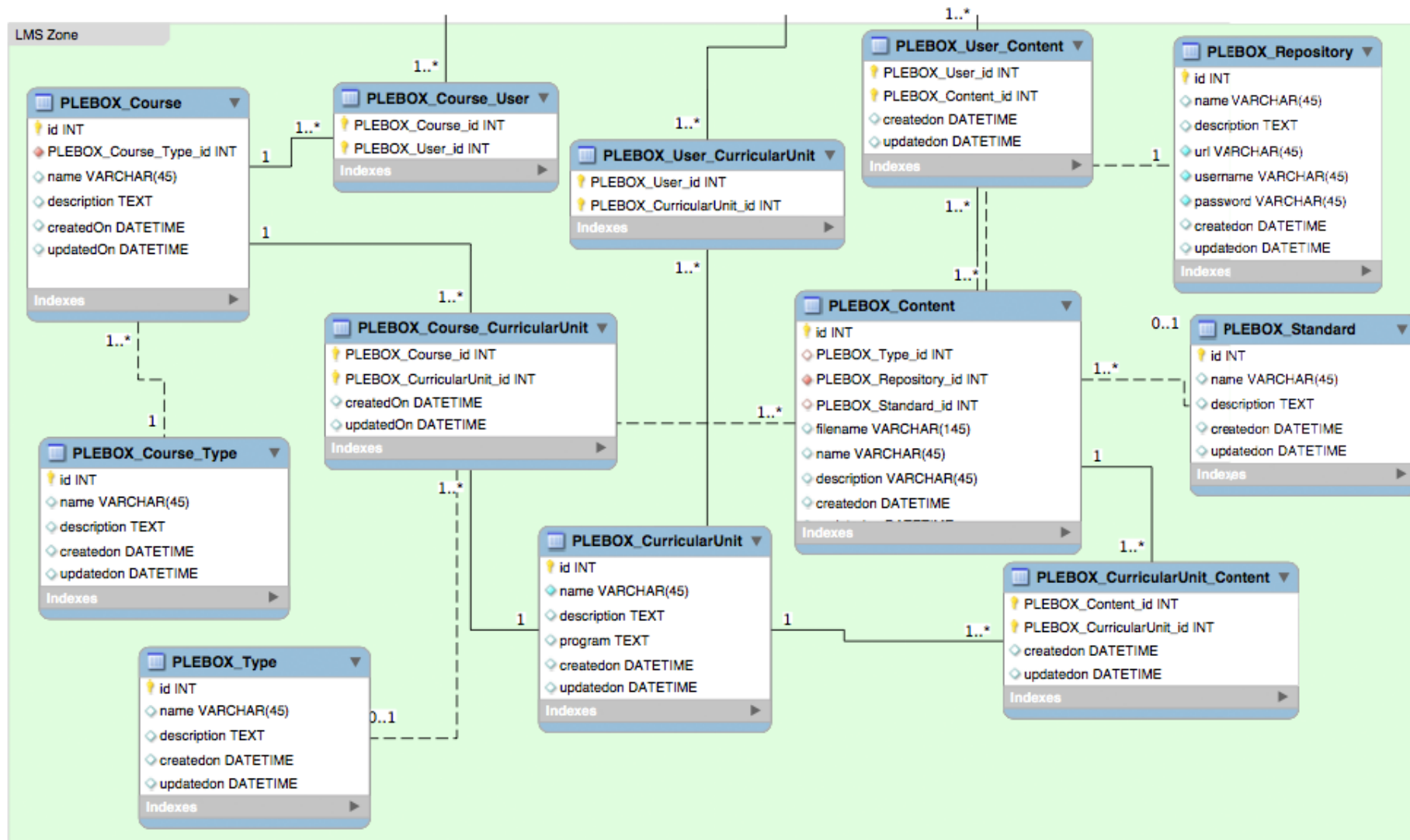
- [49] L. Kolas, A. Staupe, "The PLExus Prototype: A PLE realized as Topic Maps", in 7th IEEE International Conference on Advanced Learning Technologies (ICALT), Niigata, Japan, July 18-20, 2007, pp. 750-752.
- [50] A. Syvänen, J. Muukkonen and M. Sihvonen, "Are the open issues of social software-based personal learning environment practices being addressed?", in the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era, Tampere, Finland, September 30-October 2, 2009, pp. 142-148.
- [51] L. Jun and Y. Huiping, "Design of E-Learning2.0 Platform Based on Web 2.0", in 2010 2nd International Workshop on Education Technology and Computer Science, Wuhan, China, March 6-7, 2010, pp. 498-501.
- [52] J. Abbate, "Getting small: a short history of the personal computer", Proceedings of the IEEE, vol.87 (9), pp. 1695-1698, September 1999.
- [53] L. Potts, "Windows User Interfaces and Web User Interfaces: Convergence and Confusion in Online Experiences", in 2006 IEEE International Professional Communication Conference, New York, USA, October 23-25, 2006, pp. 22-26.
- [54] J. C. Preciado, M. Linaje, S. Comai and F. Sanchez-Figueroa, "Designing Rich Internet Applications with Web Engineering Methodologies", in 2007 9th IEEE International Workshop on Web Site Evolution, Paris, France, October 5-6, 2007, pp. 23-30.
- [55] M. Urbieto, G. Rossi, J. Ginzburg and D. Schwabe, "Designing the Interface of Rich Internet Applications", in 5th Latin American Web Congress (LA-WEB), Santiago, Chile, October 31-November 2, 2007, pp. 144-153.
- [56] O. Brandes, "Tersus visual programming platform", in 2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Pittsburgh, USA, September 18-22, 2011, pp. 245-246.
- [57] FLEX. www.adobe.com/products/flex.html, accessed on April 29, 2012.
- [58] AJAX Control Toolkit. <http://www.asp.net/ajaxlibrary/AjaxControlToolkitSampleSite/>, accessed on April 29, 2012.
- [59] Dojo Toolkit. <http://dojotoolkit.org/>, accessed on May 1, 2012.

- [60] Visual Studio. [http://msdn.microsoft.com/en-us/library/ms165079\(v=VS.80\).aspx](http://msdn.microsoft.com/en-us/library/ms165079(v=VS.80).aspx), accessed on May 2, 2012.
- [61] X. Feng, J. Shen and Y. Fan, "REST: An alternative to RPC for Web services architecture", in 2009 First International Conference on Future Information Networks (ICFIN), Beijing, China, October 14-17, 2009, pp. 7-10.
- [62] J. P. Thomas, M. Thomas, G. Ghinea, "Modeling of Web services flow", in IEEE International Conference on E-Commerce (CEC), Newport Beach, USA, June 24-27, 2003, pp. 391-398.
- [63] D. A. Menasce, "Composing Web Services: A QoS View", in IEEE Internet Computing, vol. 8 (6), pp. 88-90, November 2004.
- [64] L. Ribeiro, J. Barata and A. Colombo, "MAS and SOA: A Case Study Exploring Principles and Technologies to Support Self-Properties in Assembly Systems", in 2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems, Venice, Italy, October 20-24, 2008, pp. 192-197.
- [65] C. Fu, F. Belqasmi and R. Glitho, "RESTful web services for bridging presence service across technologies and domains: an early feasibility prototype", in IEEE Communications Magazine, vol. 48 (12), pp. 92-100, December 2010.
- [66] Microsoft Web Application Architecture Guide. <http://apparch.codeplex.com/releases/view/19802>, accessed on May 2, 2012.
- [67] J. Fong and R. Hui, "Application of middleware in the three tier client/server database design methodology", Journal of the Brazilian Computer Society, vol. 6 (1), July 1999.
- [68] S. Frolund and R. Guerraoui, "e-Transactions: end-to-end reliability for three-tier architectures", IEEE Transactions on Software Engineering, vol. 28 (4), pp. 378-395, April 2000.

Appendix 1

Entity-Relationship Data Model





Appendix 2

An Innovative Multi-Layer System Model to Improve Learning Processes

Orlando R. E. Pereira, Tiago M. C. Simões, Jorge E. F. Costa, and Joel J. P. C. Rodrigues, Senior Member, IEEE

An Innovative Multi-Layer System Model to Improve Learning Processes

Orlando R. E. Pereira, Tiago M. C. Simões, Jorge E. F. Costa, and
Joel J. P. C. Rodrigues, *Senior Member, IEEE*

Abstract— Learning processes and learning mechanisms are constantly evolving. The traditional classroom boards are replaced by Internet technologies, while the old handwritten notes are progressing to electronic notes. Through this evolution offered by new technologies, such as Web 2.0 and mobile and ubiquitous computing, learning agents need to adjust to this new reality. Distance, electronic, and mobile learning offer methods, technologies, and content, which decrease the limitations of traditional learning. Mobile learning is an extension of distance education, supported by mobile devices (e.g., smart-phones or tablets). It is an emerging learning model and process, which requires new forms of teaching, learning, contents, and dynamics between actors. This paper proposes an innovative system model for distant and mobile learning. The model considers four layers, supporting professors and student's activities, and learning and evaluation processes through innovative and intuitive user interfaces. The system is supported by several technologies including context and location awareness, pervasiveness, ubiquity, and intelligent agents. It tries to demonstrate how recent technologies can interact together to improve distant and mobile learning. The proposal is presented, demonstrated, and evaluated by over one hundred informatics engineering students. The results suggest that it is ready for deployment and evaluation in real case scenarios.

Index Terms—Distance Learning; Learning Technologies; Mobile Applications; Mobile Computing.

1 INTRODUCTION

Learning is a cognitive process owned by humans which enables the ability to retain new or transform existing knowledge. The ability involves the synthesis of several information types over a period of time and, as a result, a person should retain that behavior or knowledge. The learning process requires, at least, two types of agents, the person who will be instructed and the content learning itself. It can be achieved through two distinct learning methods, the traditional at the classroom or at distance. The traditional method involves a physical and direct interaction between the student and the professor in a classroom, or any other physical location. The distance learning breaks the physical connection between agents, and uses new technologies to achieve this goal [1].

Electronic learning (e-learning) provides an opportunity for students interacting electronically with each other as well as with their professors. It is directly supported by a technology called Learning Management Sys-

tems (LMS). LMS is a Web-based application that provides services for administration, documentation, training programs, and e-learning programs [2]. An LMS may be seen as a middleware between the student and a diversity of entities such as a professor, learning content, and evaluation process.

Through innovative technologies offered by recent mobile devices, new learning model called mobile learning (m-learning) have been coming. It offers people a new way of learning and an opportunity to learn anytime and anywhere. Under this approach, the physical restraints should not be seen as a real restriction, but rather a “user configuration”. M-learning can be achieved by the use of mobile devices, such as smart-phones or tablet computers. Due to the aforementioned specifications, m-learning is considered the next step of online learning by incorporating mobility as a primal requirement. Such devices offer an enormous range of technologies and potential to m-learning in terms of context and location awareness, ubiquity,

pervasiveness, personalization, and flexibility [3].

This paper presents an innovative learning system model to enhance learning processes. The system model comprises the following four main modules: *i*) the LMS service aggregator and provider, which can be directly used on the internal user interface or on any external supported devices, *ii*) the necessary algorithms for all data mining; such tools will supply the system active agents with central information, *iii*) a vast set of services aggregator that will provide abstraction of layers and data forwarding between internal and external system agents, and *iv*) the clients with the mandatory system requirements for system interaction. The system directly supports the learning process, professors, students, and the evaluation through innovative and intuitive user interfaces.

The remainder of the paper is organized as follows. Section 2 provides insights on available related work, while Section 3 presents the global system model focusing on its main components and inherent characteristics. Section 4 includes the performance evaluation and validation of the mobile application. Conclusions and future work are drawn in Section 5.

2 RELATED WORK

This section surfaces related approaches to LMS systems, focusing on its system model and its modularity (i.e., support for external modules, direct compatibility with mobile devices, existence of intelligent agents, and support with external platforms). Several projects with this purpose were proposed in recent years. The most relevant projects related with this topic will be reviewed, including some ongoing projects regarding mobile learning. For a better explanation of the projects, the authors split the related work in two categories. The first includes pure LMS systems, while the second addresses more diverse approaches for learning processes.

Blackboard is a well known LMS in the market since its services are used by over 9000 institutions in more than 60 countries [4]. A suite software provides solutions for various areas, such as, education, mobile, communication, and commerce. It considers the six following platforms: learn, transact, connect, mobile, collaborate, and analytics.

Each platform implies new module installation on server. Currently, it does not support intelligent agents, however, the support for native mobile devices and external platforms is robust.

SharePoint is a Web-based application platform created by Microsoft [5]. SharePoint is more than a simple LMS; it is a centralized resources platform for multiple Web applications supporting various technologies. It is typically associated with Web content, documents, and learning management systems. The SharePoint architecture considers four layers - foundation, project, service, and clients. Its nature implies a system highly modular (supporting only .NET modules). Currently, it does not support any kind of intelligent system agents or specific native mobile device interface.

Moodle is an open source course management system, also known as a LMS [6]. It has become very popular among educators around the world as a tool for creating online dynamic Web pages for their students. Its architecture is based on two layers - a server and a Web portal for clients. The server layer includes the following six modules: communication, productivity, student, administration, courses, and curriculum. Despite being a modular architecture, currently it does not support any external modules or intelligent agents.

MoMo (Mobile Moodle) is an extension of the Moodle platform and focuses on mobile devices [7]. It offers a new layer to the native system and provides the ability to implement m-learning scenarios with Moodle as a backend. Since MoMo uses Java technology at its foundation, it can be installed at any Java enabled device. Through this client users can access all the data anytime and anywhere. Since MoMo is an extension to Moodle, the Moodle administrators need to install the necessary MoMo extension on the server, which makes the compatible contents available for mobile users.

Joomla LMS is a commercial component for Joomla content management system [8]. As the aforementioned technologies, Joomla is based on a two-layer architecture, the server and the client. The server is based on PHP and MySQL technologies, while the client is a Web portal interface. The platform fully supports external content following the standard

sharable content object reference model (SCORM) format. However, it does not support intelligent agents or a native mobile device interface.

Regarding a second category of projects, the most relevant are the presented below. Samuel *et al.* [9] present the Information Gathering and Lesson Tool (IGLOO). It is a conceptual framework that evolved from the incorporation of pedagogical and technical aspects regarding learning and mobile learning. The goal of IGLOO development is to supply educators with an application that should run on mobile devices (independent of wireless connectivity) and could be used to facilitate pedagogical practices in both formal and informal learning scenarios. The system assumes a two-layer architecture, an administration system and a mobile application (supporting only Symbian OS). The administration application allows educators to set-up the quizzes. Therefore, the quizzes are sent to the student's mobile device by text message through GSM or Bluetooth.

Haitao *et al.* [10] proposed a mobile independent learning system that can detect a mobile device context and adapt it not only to the device but also to the learner itself. The system follows a two-layer approach, with a server and several clients. The devices (clients) can be the most diverse (from a computer to a mobile device), or any other devices that have the ability to access the network. The devices communicate between them through the server layer. When the server receives a request, it adjusts the answer parameters taking in account the client device, the bandwidth, among others.

Mahamad *et al.* [11] present a system to improve the education standard in Malaysia through the use of mobile devices. The system considers two layers (Server and Clients), and includes mobile quizzes and progress tracking. The project born with the objective to solve the problem identified when more than 35% of primary school students have failed the traditional Mathematic course. Currently, it only supports the Mathematic course and neither other modules nor a modular approach are proposed.

Teaching and learning anatomy and physiology are both aspects of healthcare education, and together are considered the most problematic area of the curriculum. Then,

White and Ousey [12] created an online tool that uses problem-based learning strategies to contextualize anatomy and physiology education, basing the learning process on real-world scenarios that will promote understanding and retention. Despite the system is not fully mobile it can take advantages of the new devices to fully access the information. The system is Web-based only and do not offer any external support.

Yu-mei *et al.* [13] propose a conceptual model that is supported by a framework for m-learning used in medical education. It includes multimedia, information interaction, management and monitoring modules, each one with specific characteristics and functionalities. The system is supposed to run on a PDA with Microsoft Windows Operating System (OS). Currently, it is under development and no experiments were performed yet.

Licea *et al.* [14] designed MADEE (Mobile Application Development and Execution Environment), a platform that supports the development of mobile and wireless information systems for mobile devices. It allows a student the development of easier and faster applications using conventional development tools. The architecture considers clients and object server layers. Each client can interact with the server executing specific orders, while the object server stores all the application data divided by object catalogs. Since the platform adjacent technology is Java, every Java enabled device can access to the system.

Several learning frameworks and LMS were analyzed, each one with specific and common characteristics. The authors argue that learning and mobile learning process technologies should evolve and adapt to the current available technologies and devices. In this sense, next section presents the proposal of a system model for achieving the aforementioned goal.

3 SYSTEM MODEL

The system model considers several layers. Each one includes different characteristics and specifications, although all of them cooperate for the system homogeneity. The use of a multi-layer approach allows services abstraction and thus a better specification of each one. The proposed system description

includes a server, middleware abstraction, and clients. The section starts with a description of the server side, including the PLEBOX platform and the core intelligence module [15]. Both are responsible for all the server structure and data processing. Next, a set of tools, including parsers to compilers and communication APIs are introduced. These components can be seen as a “man in the middle” between a server and clients. Clients include regular computers, smartphones, mobile devices, or even tablet computers. Figure 1 illustrates the proposed system model with its main components.

A detailed description of each module, its characteristics and specifications, highlighting the inherent innovation, is addressed.

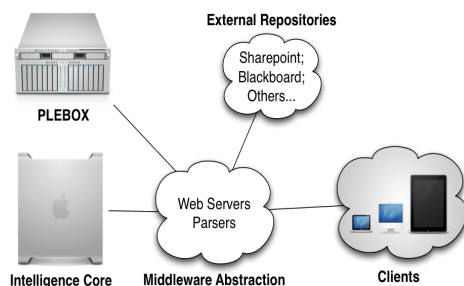


Fig. 1. Illustration of the system model.

3.1 PLEBOX

The PLEBOX platform is located on top of the .NET framework and uses MySQL and the Internet Information Server (IIS) for data storage and availability. Unlike all other platforms, the concept of the PLEBOX goes through the idea of a digital “box” that contains different elements making use of the Rich Internet Application (RIA) and operating systems (OS) concepts. It follows a personal learning environment approach with integration of modern tools provided by the so-called Web 2.0/3.0.

This platform provides a set of specific environments for each user. Each one contains several modules and features taking into account the inherent user settings and configurations. So, each user has a unique environment personalized by him/her. Figure 2 shows an image that illustrates a user environment. The user interface is totally dynamic and configurable providing a virtually unique environment.

A set of internal APIs enables various plat-

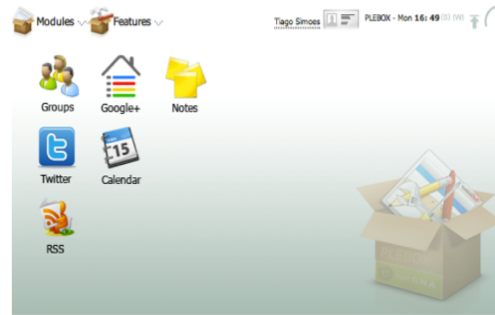


Fig. 2. Illustration of the PLEBOX user environment.

form characteristics, such as window system management, permissions management, modules or features management, database operations, and utilities. Furthermore, it supports especially common LMS main features like the creation of courses, contents, assignments, subjects, and repository queries. It also provides a development environment for libraries creation. Using the aforementioned APIs, a user can create or modify the available modules or features. The platform provides a software development kit (SDK) that enables the creation of objects, such as tables, lists, notifications, panels, among others, in an easy and comfortable way. The system is totally modular offering the possibility to expand and increase the size of the “box”, allowing the installation of new or modified applications easily. Each application can be created focusing a particular objective or organization.

Nowadays, learning platforms follow a set of outdated guidelines and rules for content management, presentation, and formalization. Those guidelines have become obsolete along the time and inappropriate to the students’ requirements. With PLEBOX, authors intend to introduce a new Web platform concept. The PLEBOX innovates on user interaction with the platform exercising concepts that only exist in non-Web-based systems, such as windows management, pure personalization of the environment, and ability to create and install external modules.

3.2 Core Intelligence

The core intelligence is the second component of the Server side and it directly supports the PLEBOX platform, providing the necessary tools for intelligent agents support in the system. This component provides intelligent mechanisms to achieve smart teaching

environments [16]. It can be seen as an innovative technology, which is based on cognitive and thinking computer science, integrating several artificial intelligence mechanisms on educational models.

The objective of this core intelligence module is the creation of a true and realistic cognitive learning model, which delivers specific and adaptive contents to students in a smoother and intuitive way. In order to achieve such objectives the following features were performed on this module: a specific intelligent environment that directly support the learning process enabling the automatic generation of a variety of problems and exercises; capacity to explain the same problems and teaching contents; automatically solve and generate understandable answers for all teaching content. This layer provides users an evolutionary learning system, able to predict user profile, characteristics, and learning style. Such classifications will provide a way to the system for characterizing the students in groups and treat each one individually and in a unique way. Based on the aforementioned classifications the system allocates resources and adapts to users autonomously. Each group of students or an individual student will have specific characteristics and those characteristics will be taken into account when contents will be delivered. Plus, the Intelligence Core will redirect specific content to each user or to a group of users. Thus, each group is able to learn using its own characteristics and metrics, providing a true personalized and focused learning process. Moreover, evolutionary monitoring will provide tools for constant monitoring users and all the inherent features, characteristics, and processes. If at a given time any user has a specific problem or difficulty, the system can adapt and helping him/her to pass through it. The system will act and react accordingly when possible situations may occur with students.

Using this set of characteristics, the Intelligence Core is adapted perfectly to the system and to all the users providing a truly intelligent learning mechanism.

3.3 Middleware Abstraction

The middleware abstraction provides a bidirectional and constant communication between a Server and Clients. This set of APIs supports the communication among system

agents. Despite being a transparent abstraction layer, it plays a key role in the system. It includes a modular middleware that provides the necessary tools for interconnection with external frameworks, APIs, architectures, and systems without loss of productivity, compatibility, or features, creating a broader computational system. Several specific APIs, such as database, learning, SDKs, among others, provide the needed and single tools for each system layer to communicate with each other.

The communication between layers and agents is performed through Web services (WS). In order to deliver the processing power to the corresponding services the system adopted the REST architecture by default. Specific PLEBOX WS will provide an easy way for users to authenticate on the LMS. Plus, it will provide access to the user characteristics and definitions. These characteristics are human-computer-interaction based. It will provide a tunneling for content gathering, processing, and delivering.

Intelligence Core WS will afford the necessary tools for intelligent agents to navigate over the system and process the inherent information. Everytime user requests information or the system delivered it, both WS work together to tune and adapt the inherent information.

External LMS API is a set of libraries that provides the system with a connection to external LMS systems such as Sharepoint, Blackboard, and others. Due to the fact of the system modularity this API provides a way to interact with external LMS and obtain all its contents in an easy and transparent way to the user and the system. The API provides methods for the internal system authenticating on external platforms, data retrieval, processing, and visualization.

External LMS WS will use the above-mentioned API's and process data accordingly and correctly deliver it to the PLEBOX layer for internal data storage. Each time a new data set arrives to the system it is stored internally for performance, security, and robustly purposes. Due to the fact the platform uses and connects to external platforms to get new or updated data it is crucial that data may be organized in a specific data format to achieve data unification. Plus, storing external data internally will benefit of all the intel-

ligent agents and raises the security mechanism to access the inherent data. The internal storage data format adopted follows the standard SCORM. Whenever the system retrieves data, despite the original format it will be stored using SCORM.

The specific WS client adds an intermediary layer for client authentication and data processing. It offers a transparent way to exchange and process data from/to the server layer.

Taking into account the afore-described characteristics and implementations the platform presents a robust and modular abstraction middleware that provides fully data exchange between all system agents in an almost transparent way.

3.4 Clients

The client layer is responsible for user and system interaction. Figure 3 presents an example of the user interface when he/she accesses his/her current enrolled curricular units. The user interface has three main sections that can be directly accessed: Main menu (left-side), Action-Bar (top) and the Central area.

The Main menu is used for directly access the main application menus like *Classes*, *Messages*, *Settings*, or *Assignments*. The Action-Bar is typically used for quick searches, social networks interaction and access to the application support system. The Central area displays the information. Every time a user changes the menu or do an action, it will be directly reflected on the central area. This area is bigger to take advantage of the Tablet computers screen size. Both Main menu and Action-Bar are always available during the user navigation. However, the Action-Bar is

dynamically modified depending the Main menu option and Central interface content.

Clients can range from a Web portal access to a smartphone, a mobile device or even a Tablet computer [17]. It is more than a simple user's interface (UI) since it uses specific and new approaches, such as a totally dynamic, interactive, adaptable, and personalized UI. It presents the information to users in an innovative way. If a mobile device or Tablet computer is available, this UI layer includes context awareness providing users technology that does small tasks in a transparent and ubiquitous way without directly interacting with the user. Moreover, users can automatically authenticate in certain defined locations, such as a university campus or a laboratory. The authentication can be performed though RFID, near field communication (NFC) [18], or through specific and unique tokens generated per each user device over a wireless network. On certain scenarios, the context awareness can provide specific information to users using external devices, such as an information monitor. If more than one user is available on a specific context the system can interact and create a collaboration scenario among them. Furthermore, the users can take the advantage of their context and use external resources available on that context, like printing a document or use a projector or LED display for a presentation.

Location awareness offers to users a possibility to receive contents based on their current location and network typology. Adaptable content should take into consideration not only the user preferences but also the type of user network and location. The detection of network boud rate will provide a way to receive specific contents and avoid scenarios where users have a slow connection to receive large files size. Using location awareness, the system guarantees the access to the right information data type. Figure 4 presents an example of a location aware scenario where a device detects its location and Internet connection settings. Using such settings the system adapts itself and delivers content accordingly.

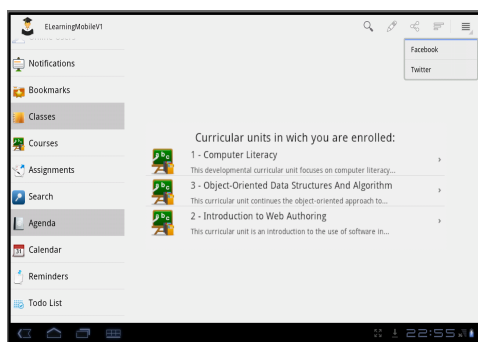


Fig. 3. Illustration of the mobile application user interface.

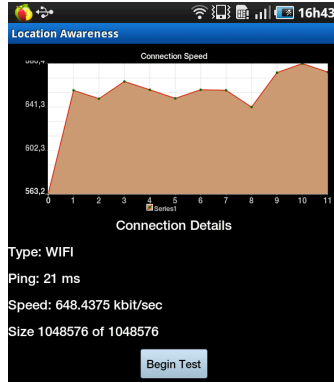


Fig. 4. Location awareness example.

Ubiquitous and wearable systems can provide specific information about physiological parameters. The conjunction of learning systems with such technologies offers a wide range of information that can be used to analyse and provide information to both system and users. For example, if a wearable system composed by sensors for temperature or heartbeat monitoring it is able to infer if a given user is stressed or not. Furthermore, it can study if that condition is responsible for his/her acts on the current system interaction. A large collection of wearable devices can be embedded on the system providing a robust and single learning experience. Since the system integrates intelligent agents and uses sensed information, it can focus on particular topics for specific users (personalized learning). Using this set of characteristics users are able to use a set of tools that provides an innovative mobile and intelligent learning environment. They can learn anytime, anywhere, and access the information that really matters since they learn in a personalised way. Due to the fact that mobile devices and Tablet computers market share are constantly evolving, a learning system that relies on those devices provides a strong option for those really want to learn and progress in a distant and mobile way.

4 PERFORMANCE EVALUATION AND VALIDATION

The authors performed experiments in order to evaluate and validate the proposed approach. The first main objective of this approach was the validation of the user's interface, its organization, user experience, and

inherent configurations. In order to get feedback from students and their interaction with the system, a prototype was deployed on a physical device and given to them for experiencing.

The experiment phase took place between January 2 and February 16, 2012. During this time interval more than 140 students of the University of Beira Interior (graduating in Informatics Engineering) were contacted to make the experience. From this universe, 117 correctly experienced the mobile application and filled correctly the survey questions.

The students used the application and experimented it on several scenarios (accessing and changing the user profile information; listing and accessing their current enrolled courses; listing and accessing their current assignments; accessing to a calendar style navigation interface, and changing several application settings). All the experiments followed a sequence of activities and took about 30 minutes each one, and reactions were gathered. At the end, a brief survey using the *Doodle* platform was filled in order to evaluate the initial user experience. The survey evaluated the easy to access the information, personal configurations, and initial system scalability. The questions may be found in Table I.

Name	Description
Q. 1	Is the overall performance adequate?
Q. 2	Is the application easy to use?
Q. 3	Is the application environment user friendly and intuitive?
Q. 4	Are the navigation options clear and consistent?
Q. 5	Is the UI organization adequate to the problem of learning?

Table I. List of the survey questions for the mobile application.

The results of the survey administration are available in Figure 5. As may be seen, at least half of the survey intervenients strongly agree that the overall application performance and the navigation options are consistent. Other big number of users (45%) tends to agree that the application is easy to use. A small number of students (16%) tend to disagree that application environment may

not be user friendly, intuitive, or the user interface may not be adequate for a learning environment (14%). These two questions are the ones that present more undecided students. It may occur due to the fact that users experimented a completely innovative application and a total different way to interact with contents.

Through this evaluation phase the authors can shape and adapt the development process in order to adapt applications to users' needs. Furthermore, during this evaluation phase, the server logs were also analyzed in order to anticipate and solve potential problems.

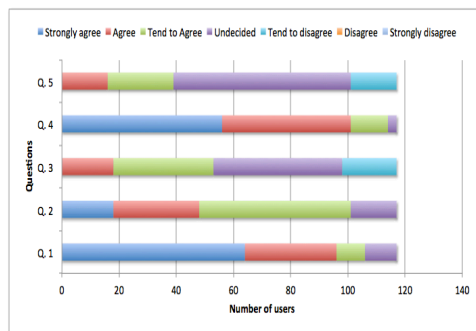


Fig. 5. Results of the system utility survey for students.

5 CONCLUSION AND FUTURE WORK

This paper is a first step towards a new and innovative model architecture that relies on recent technology to directly support learning and mobile learning environments. A four-layer system model and corresponding development on the PLEBOX platform was presented. Each layer is responsible for the LMS management, intelligent agents, abstraction middleware, and clients, respectively.

This paper presents an innovative mobile application and over one hundred students experienced it. The results obtained were very interesting and provided a good feedback about the application robustness, performance, and system interaction.

Despite the fact that the proposed model increases the implementation degree of complexity it can offer a brand new vision for the future of the learning process and scenarios. The model assembles several ideas that can be useful to update current learning models or even accomplish some that were previously impossible to achieve.

Currently, both students and professors have access to a wide range of learning technologies, however, new and innovative technologies can be used in conjunction to assist those approaches. Some learning technologies, characteristics, and restrictions currently available can be easily overcome in a neutral way by the aforementioned system model. Summarizing, the user will continuously access and use information anytime, anywhere using cutting edge technology of a wide range of computing areas.

In terms of future work the authors believe that learning processes, mechanisms, and agents will evolve to a new era, where technologies, learning materials, and people will directly interact in an almost natural way. So, the presented model may be adapted to cover all the aforementioned specifications. Plus, due to the constant evolution of mobile devices, operating systems, and technologies, the mobile application should be updated to upcoming technologies.

ACKNOWLEDGMENTS

This work was partially supported by the *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, by National Funding from the FCT – Fundação para a Ciência e a Tecnologia through the Pest-OE/EEI/LA0008/2011 Project, and by the E-Learning 3.0/5635 QREN Project.

REFERENCES

- [1] Z. Yazhen and L. Jian, "Application of 3G-based Mobile Learning in Teacher Training," in *Fourth International Conference on Information and Computing (ICIC 2011)*, 2011, pp. 27-29.
 - [2] I. A. Almrashdah, et al., "Distance learners acceptance of learning management system," in *6th International Conference on Advanced Information Management and Service (IMS 2010)*, 2010, pp. 304-309.
 - [3] K. Scott and R. Benlamri, "Context-Aware Services for Smart Learning Spaces," *IEEE Transactions on Learning Technologies*, vol. 3, pp. 214-227, 2010.
 - [4] A. Al-Badawi, "The Design of E-learning Services on the Blackboard," in *Fifth International Conference on Internet and Web Applications and Services (ICIW 2010)*, 2010, pp. 186-193.
- R. Zhijun, "Constructing Business Intelligence Solution with SharePoint Server 2007," in *International Conference on Electrical and Control Engineering (ICECE 2010)*, 2010, pp. 615-618.

- [6] S. Kumar, *et al.*, "A comparative study of moodle with other e-learning systems," in *3rd International Conference on Electronics Computer Technology* (ICECT 2011), 2011, pp. 414-418.
- [7] F. Xhafa, *et al.*, "Implementing a Mobile Campus Using MLE Moodle," *International Conference on in P2P, Parallel, Grid, Cloud and Internet Computing* (3PGCIC 2010), 2010, pp. 207-214.
- [8] X. Cao and Y. Wenhua, "Using Content Management System Joomla! to Build a Website for Research Institute Needs," in *International Conference on Management and Service Science* (MASS 2010), 2010, pp. 1-3.
- [9] O. O. Samuel, *et al.*, "Igloo: Mobile learning system to facilitate and support learners and educators," in *2nd International Conference on Adaptive Science Technology* (ICAST 2009), 2009, pp. 355-360.
- [10] P. Haitao, *et al.*, "The approach of device independence for mobile learning system," in *IEEE International Symposium on IT in Medicine Education* (ITIME 2009), vol. 1, August 14-16, 2009, pp. 575-578.
- [11] S. Mahamad, *et al.*, "M-Learning: A New Paradigm of Learning Mathematics in Malaysia," *CoRR*, vol. abs/1009.1170, 2010.
- [12] S. A. White and K. J. Ousey, "Teaching Anatomy and Physiology Online Using Problem-Based Learning," in *Second International Conference on Mobile, Hybrid, and On-Line Learning* (ELML 2010), 2010, pp. 23-28.
- [13] P. Yu-mei, *et al.*, "Learning Can Happen Anytime and Anywhere: The Application of M-learning in Medical Education," in *Second International Workshop on Education Technology and Computer Science* (ETCS 2010), vol. 2, 2010, pp. 508-511.
- [14] G. Licea, *et al.*, "Teaching mobile and wireless information systems development in engineering courses," *Computer Applications in Engineering Education*, vol. 18, pp. 261-268, 2010.
- [15] T. M. C. Simões, *et al.*, "Personal Learning Environment Box (PLEBOX): A new approach to E-learning platforms," *Computer Applications in Engineering Education*, 2011.
- [16] I. Magnisalis, *et al.*, "Adaptive and Intelligent Systems for Collaborative Learning Support: A Review of the Field," *IEEE Transactions on Learning Technologies*, vol. 4, pp. 5-20, 2011.
- [17] E. M. Il, *et al.*, "Tablet PC Technology for the Enhancement of Synchronous Distributed Education," *IEEE Transactions on Learning Technologies*, vol. 1, pp. 105-116, 2008.
- [18] R. Want, "Near Field Communication," *Pervasive Computing, IEEE*, vol. 10, pp. 4-7, 2011.

Orlando R. E. Pereira was born in Tortosendo, Portugal, in 1982. He received BSc graduation in 2008 in Informatics Engineering from University of Beira Interior, Portugal and MSc degree in Informatics Engineering – Networks and Multimedia from University of Beira Interior, Portugal in 2010. He is a PhD student in University of Beira Interior, Portugal. He is also

an invited assistant professor of the Informatics Engineering Department at the Superior School of Technology of the Polytechnic Institute of Castelo Branco, Portugal. He is the research group leader of Ubiquitous Computing on iTech-ON, Lda., Portugal. His current research areas are Mobile Computing and Mobile Learning.

Tiago M. C. Simões is an MSc student on Informatics Engineering at the University of Beira Interior. He received his BSc degree in Informatics Engineering from the University of Beira Interior in 2010. He is also a researcher at the University of Beira Interior on behalf of the eLearning 3.0 project. His current research areas focus on e-Learning technologies.

Jorge E. F. Costa is an MSc student on Informatics Engineering at the University of Beira Interior, Covilhã, Portugal. He received a BSc degree in Technologies and Information Systems from the University of Beira Interior in 2010. He is also a researcher at the NetGNA group, Instituto de Telecomunicações, Portugal on eLearning 3.0 project.

Joel J. P. C. Rodrigues is a professor in the Department of Informatics of the University of Beira Interior, Covilhã, Portugal, and researcher at the *Instituto de Telecomunicações*, Portugal. He received a PhD degree in informatics engineering, an MSc degree from the University of Beira Interior, and a five-year BSc degree (licentiate) in informatics engineering from the University of Coimbra, Portugal. His main research interests include sensor networks, e-health, e-learning, vehicular delay-tolerant networks, and mobile and ubiquitous computing. He is the leader of NetGNA Research Group (<http://netgna.it.ubi.pt>), the Vice-chair of the IEEE ComSoc Technical Committee on Communications Software, the Vice-Chair of the IEEE ComSoc Technical Committee on eHealth, and Member Representative of the IEEE Communications Society on the IEEE Biometrics Council. He is the editor-in-chief of the International Journal on E-Health and Medical Communications, the editor-in-chief of the Recent Patents on Telecommunications, and editorial board member of several journals. He has been general chair and TPC Chair of many international conferences. He is a member of many international TPCs and participated in several international conferences organization. He has authored or coauthored over 200 papers in refereed international journals and conferences, a book, and 2 patents. He had been awarded the Outstanding Leadership Award of IEEE GLOBECOM 2010 as CSSMA Symposium Co-Chair and several best papers awards. Prof. Rodrigues is a licensed professional engineer (as senior member), member of the Internet Society, an IARIA fellow, and a senior member of ACM and IEEE.