



Virtual Portal for Interaction and ICT Training for People with Disabilities

Integrated ViPi platform

Outcome No.		13	
Workpackage No.	4	Workpackage Title	Deployment of ViPi interactive community portal and e-learning platform
Authors		Kosmas Petridis, Nikolaos Kakardakos (HYPERTECH), Karel Van Isacker, Bernard Van Isacker (PhoenixKM)	
Status (F: final; D: draft; RD: revised draft):		F	
File Name:		D13 Integrated ViPi Platform	

The ViPi KA3 LLL project (511792-LLP-1-2010-1-GR-KA3-KA3NW) has been partially funded under the Lifelong Learning program, subprogramme KA3 ICT. This publication reflects the views only of the author(s), and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Education and Culture DG

Version History table

Version no.	Date	Dates and comments
0.1	13-03-2012	First draft version that contains the adopted technologies, installation procedure and open development issues of the integrated ViPi platform (Hypertech)
0.2	25-05-2012	Updated version that contains the full functionality of ViPi portal (Hypertech)
1.0	28-05-2012	Final revised version (PhoenixKM)

Table of Contents

Version History table	2
1 Introduction.....	6
2 ViPi Platform Aim & Objectives	7
3 ViPi Portal Functionality	9
3.1 Registering with ViPi.....	9
3.2 Profile Annotation	10
3.3 Browsing through ViPi Portal	10
3.4 The ViPi Learning Object Repository	15
3.5 Contributing to ViPi-Submitting New Content	15
3.6 Semantically-enabled Search	19
3.6.1 Conceptual Search.....	19
3.6.2 Semantic Navigation.....	19
3.7 Connection with Popular Social Networks	20
3.8 Printing Posts and Exporting to PDF.....	21
3.9 Using the ViPi Online Training Environment	21
4 Adopted Technologies & Installation Steps	22
4.1 Installation of Xampp	22
4.2 Installation & Deployment of WordPress.....	22
4.2.1 Enabling Permalinks	23
4.2.2 XML-RPC Services Export & Deployment	24
4.3 Installation & Deployment of the BuddyPress Plug-in	25
4.3.1 BuddyPress XML-RPC Services Export & Testing.....	26
4.4 Installation & Deployment of ATutor	27
4.5 Installation of Selected WordPress Plug-ins.....	28
4.5.1 WP-Post Ratings Plug-in	28
4.5.2 Multilingual Support.....	29
4.5.3 Facebook, Twitter & Google+ Social Widgets	31
4.5.4 Backup WordPress.....	31
4.5.5 Access Keys.....	32
4.5.6 Acronyms.....	33
4.5.7 All In One SEO Pack.....	34
4.5.8 Breadcrumb NavXT	34

4.5.9	Print Friendly and PDF	34
4.5.10	BuddyPress Sidebar	35
4.5.11	BuddyPress Moderation	35
4.5.12	Q and A – FAQ	36
4.5.13	Google Analyticator	37
4.5.14	Fast Secure Contact Form.....	37
4.5.15	Blog-in-Blog	38
4.5.16	Require Post Category	39
4.5.17	Search by Category.....	40
5	New Components Implemented inside ViPi.....	41
5.1	WordPress & ATutor Seamless Access	41
5.2	ViPi Semantic Content Management (VSCM) plug-in	41
5.2.1	Ontology Management	41
5.2.2	Administration Component.....	42
5.2.3	Annotation of Learning Objects (LOs)	44
5.2.4	Annotation of User Profiles	44
5.2.5	Conceptual Search.....	44
5.2.6	Semantic Navigation.....	44
5.2.7	XML-RPC Interface.....	44
6	Annex I: ViPi Semantic Content Management (VSCM) Module.....	47
7	Annex II: ViPi Semantic Content Management (VSCM) Ontology	48
8	Annex III: Evaluation of Additional WordPress Plug-ins.....	49
8.1	BuddyPress Mass Messaging.....	49
8.2	BuddyPress Like.....	49
8.3	BuddyPress Pending Activations	49
8.4	BuddyPress Verified	49
8.5	Dynamic Widgets.....	49
8.6	Menubar	49
8.7	Google XML Sitemaps v3 for qTranslate	49
8.8	qTranslate Extension	50
8.9	qTranslate META	50
8.10	BuddyPress and qTranslate	50
8.11	Language Bar Flags	50



Education and Culture DG

Lifelong Learning Programme

ViPi Project 511792-LLP-1-2010-1-GR-KA3-KA3NW

8.12	TDO Mini Forms.....	50
9	Annex IV: Evaluation of Accessibility Plug-ins	51

1 Introduction

The main purpose of the deliverable “D13-Integrated ViPi Platform” is to describe the functionality, the adopted technologies, as well as, the components used to launch the ViPi platform. Chapter 2 describes the ViPi portal aims & objectives. Chapter 3 presents the ViPi portal with all the integrated functionality. Chapter 4 describes the adopted technologies & the necessary installation steps that need be followed in order to deploy the integrated platform from scratch, whereas a solution is also provided to migrate an existing ViPi platform installation as well as backup the platform periodically. Chapter 5 is dedicated to the description of the new components & WordPress plug-ins that were developed during the WP4 development.

Finally, Annex I includes the ViPi Semantic Content Management (VSCM) approach, Annex II the corresponding VSCM ontology implemented, Annex III the evaluation performed on additional WordPress plug-ins, while Annex IV describes the extensive evaluation of accessibility plug-ins performed during the ViPi platform implementation.

2 ViPi Platform Aim & Objectives

A majority of Europe's people with disabilities are unemployed (ANED figures state over 60%). There are many reasons for this. Many people with disabilities have advanced skills but suffer significant barriers to employment. This is an important issue which must be addressed. For others, skill acquisition must be facilitated. ViPi aims to support and facilitate the acquisition of basic ICT skills for those people with disabilities who lack them.

ViPi aims to provide a “one-stop-shop” interactive portal & learning environment for ICT skills by delivering a comprehensive multilingual portal, with:

- an embedded multilingual social community (for VET centres, people with disabilities, ICT training centres, etc.), using the latest social media, facilitating interactive information sharing, interoperability, and collaboration, with access to all;
- a fully accessible (WCAG 2.0) multilingual Web 2.0 enabled online learning environment for ICT for people with disabilities and their trainers, with an interactive and extensive repository of interoperable SCORM compliant learning objects (LOs) that focus on basic ICT literacy to support the acquisition of core skills by people with disabilities, in order to facilitate their entry into or sustain their employment in, the regular labour market; LOs will be enriched with intuitive mobile and internet/PC based educational/serious games;

The integrated ViPi platform, which is described in the deliverable, will equally act as a “one-stop-shop” for trainer organizations to find and contribute LOs that they can integrate in existing learning environments and practices. Using Open Source Software, ViPi platform extensions can be easily integrated. The ViPi platform will thus be able to support a fully accessible and Open Source based pan-European learning network and community, bringing together key stakeholders and gatekeepers (VET, target groups, umbrella organizations), while offering a vast set of reusable (PC and Mobile) LOs, supported by Web 2.0 social services.

ViPi project envisages fulfilling the gap of accessible and flexible training, designed to meet the specific needs of people with disabilities, as well as the trainers themselves. ViPi project develops an interactive online platform where people with disabilities can access a wide variety of ICT training courses and serious games for acquiring ICT skills, while trainers are able to upload and download specific learning objects, using a semantically enriched environment to improve the searchability, thus increasing the relevance of the results obtained.

This will allow people with disabilities to grasp these core skills and become able to enter or sustain their employment in the regular labour market. The platform will be enriched with intuitive and accessible mobile and Internet/PC based educational/serious games. Additionally, also a mobile Android based social application linked with ViPi platform is also deployed.

The ViPi integrated platform is currently available in Dutch, English, Greek and Lithuanian languages.

It is obvious from the above objectives, that the ViPi platform adopts efficiently the concepts and technologies of Web 2.0, that is, it moves beyond the traditional one-way interaction of the Internet, providing collaborative and social networking facilities. The importance of this fact is twofold: i) The

two-way interaction capabilities of the Web 2.0 are employed also for the benefit of people with disabilities, increasing their quality of life and communication means, ii) The social networking technologies are employed in the training/education domain, showing the path to efficient technology adoption in domains other than the pure leisure communication.

Moving one step further, the ViPi project also explores the adoption of the Web 3.0 concepts. That is, the environment of the ViPi platform is semantically enriched, thus improve the search-ability and relevance of the results obtained. The ViPi platform is enhanced with semantic annotation and discovery of content, thus moving beyond the basic keyword search functionality and adopting at the extent possible the value of the Web 3.0.

The following sections provide further insight and detail the functionality and design considerations of the integrated ViPi platform.

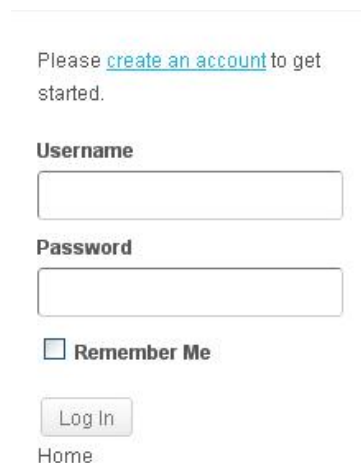
3 ViPi Portal Functionality

ViPi portal (beta demo version) is currently available through the following web page:

http://vipidev.dyndns.org:8080/sites/ViPi_portal/

3.1 Registering with ViPi

The registration of the ViPi portal is essential for every user that wants to have access to the ViPi content. This can be performed through the main page of the portal:



Please [create an account](#) to get started.

Username

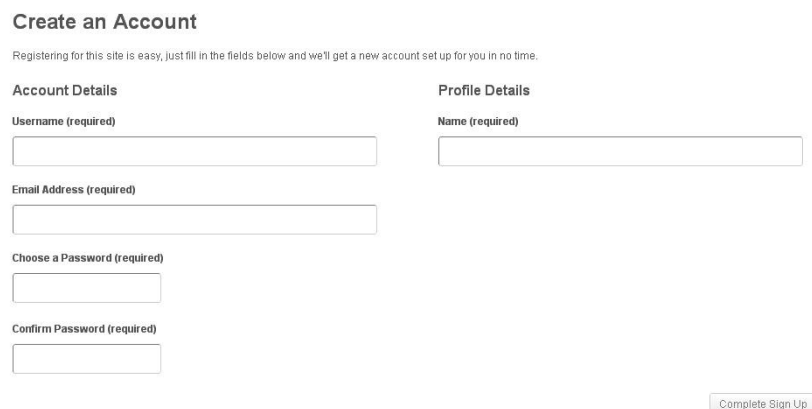
Password

☐ Remember Me

[Home](#)

Figure 1: The registration and login panel of ViPi portal

If the “Create an account” option is selected, then the user is redirected to the following page, where he/she is prompted to enter all required data. More specifically, username, password, email address and the name of the user are required for registering into ViPi portal.



Create an Account

Registering for this site is easy, just fill in the fields below and we'll get a new account set up for you in no time.

Account Details	Profile Details
Username (required)	Name (required)
<input type="text"/>	<input type="text"/>
Email Address (required)	
<input type="text"/>	
Choose a Password (required)	
<input type="password"/>	
Confirm Password (required)	
<input type="password"/>	

Figure 2: User fills in all required registration data

Once submitted, the user has to activate the registration by following the instructions that have been sent to his/her email.

3.2 Profile Annotation

One essential part of the registration process, is to provide a detailed profile with option to indicate interested fields that match user preferences. Interested fields are concepts (classes or instances) from the ViPi ontology that will be presented in a following section of the document.

The profile annotation of each user could be performed by dragging and dropping the desired and relevant concepts from left to the right. Once concepts are dropped into the “User annotations” tree, user profile is automatically annotated with the selected concept. The specific procedure will later allow the personalization of the navigation experience of the user within the ViPi platform.

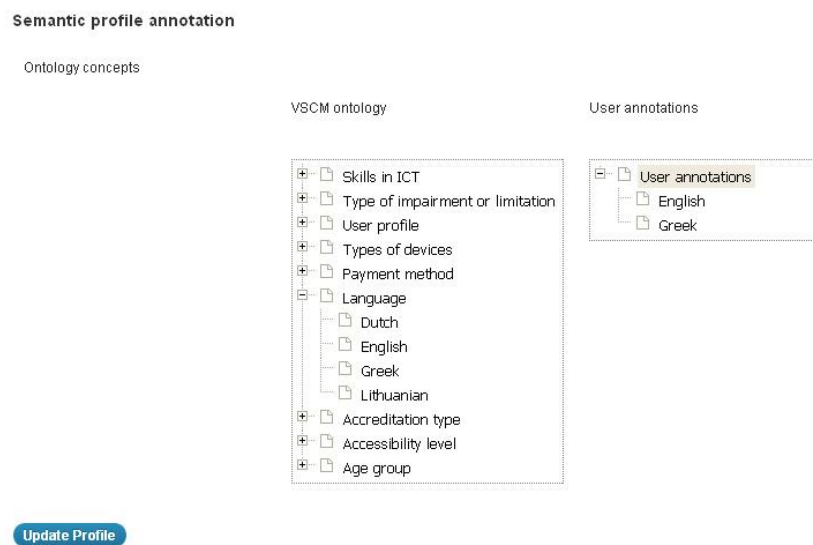


Figure 3: User profile annotation

3.3 Browsing through ViPi Portal

Once the user visits the portal, the following main web page appears.

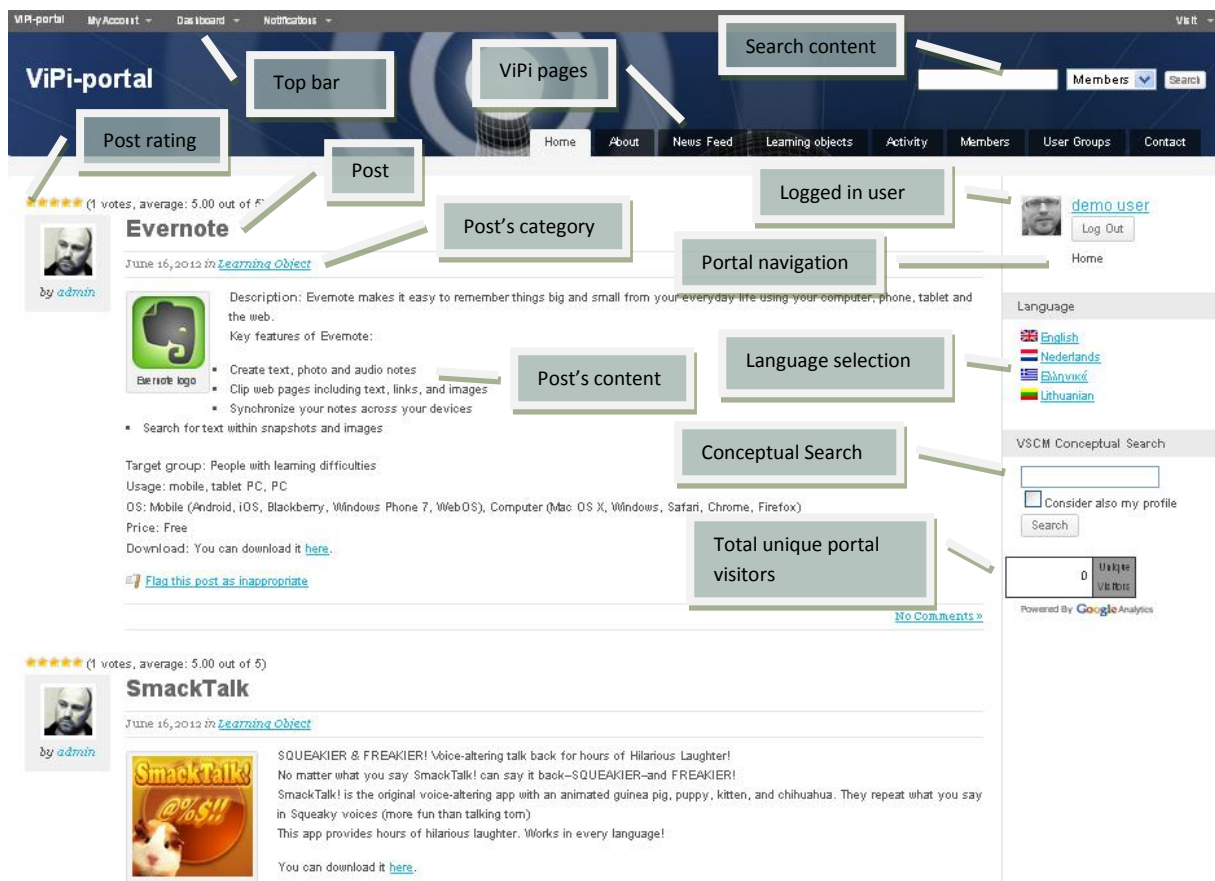


Figure 4: ViPi portal main page

The following sections appear in the main page of ViPi portal:

- **Top Bar:** Top bar contains a menu for easy access to logged in user account, management of posts, comments and notifications.

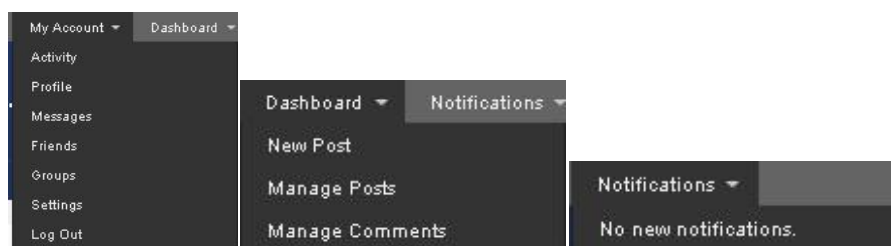


Figure 5: ViPi portal top bar functionality

In “My Account” menu, user has access to ViPi portal social network and general settings on the account and posts within the portal.

The ViPi platform features an integrated internal social network that allows providing social network capabilities between portal users. The social network includes the following selections:

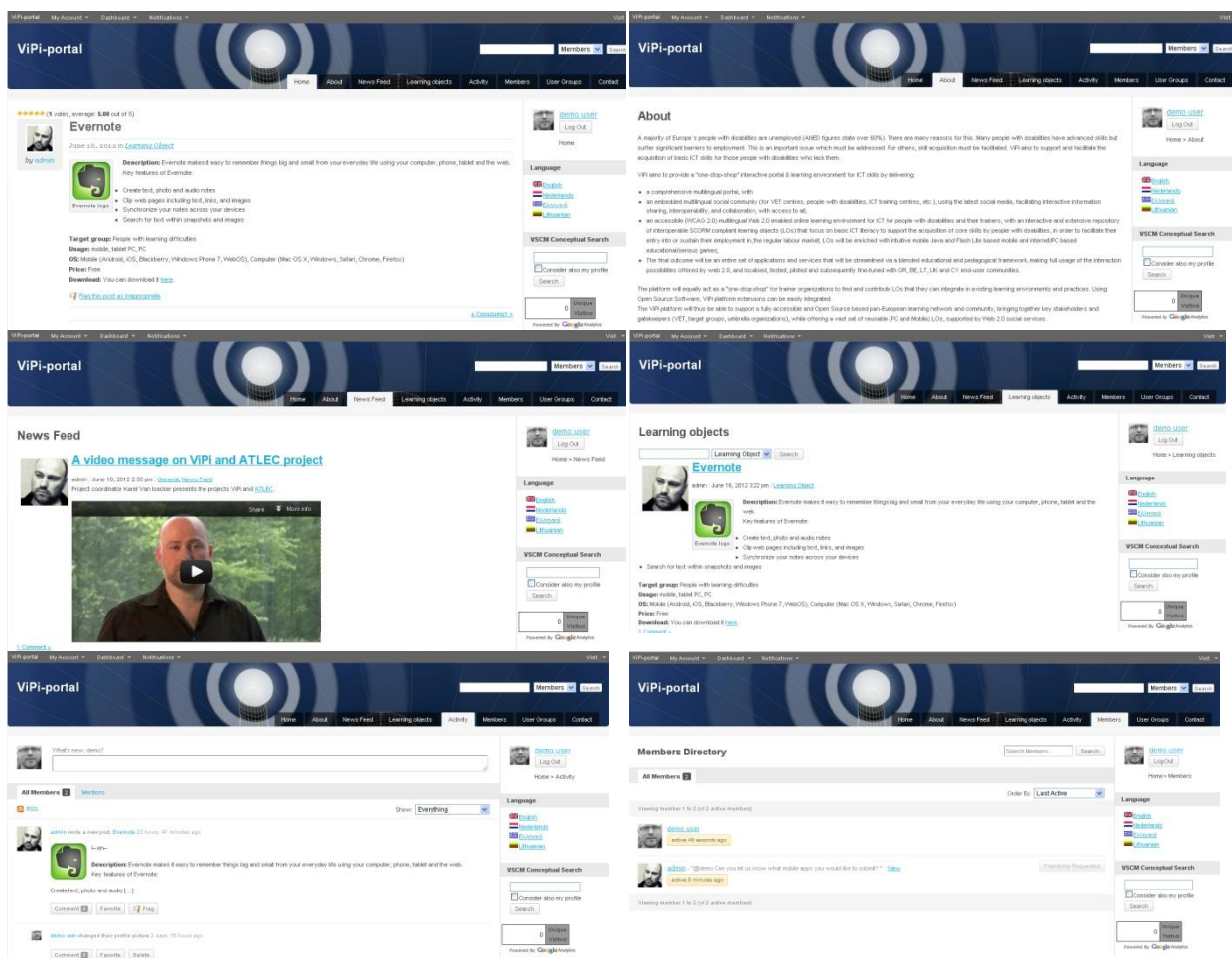
- Profiles for members
- Private message functionality

ViPi Project 511792-LLP-1-2010-1-GR-KA3-KA3NW

- Friend Networks
- Groups, photos and other user generated content
- Activity/Action streams of public actions on the portal

The “DashBoard” menu, provides the functionality to submit a new post to the portal and manage user’s posts and comments. Finally “Notifications” menu provides a notification list for the user regarding his interaction with other portal users.

- **ViPi pages:** ViPi pages provide a significant functionality for accessing main content of the portal. “Home page” is the default page that appears when a user logs in the portal and displays the most recent posts that are submitted to the platform by all users. “About” page gives a brief overview of the platform, while “News Feed” provides posts that are submitted to the “News” category. “Learning Objects” page lists all posts that are under the category “Learning Object” and compose the main content of ViPi platform. This page also contains a built in search functionality on top of the page for searching content within only Learning Objects. “Activity” page provides an overview of recent user’s activity on the portal, “Members” an overview of all registered users to the platform and “User Groups” an overview of users groups that are created within ViPi platform. Finally, “Contact” page provides a form to communicate with the portal administrator. A sample of these pages is displayed below.



The screenshots illustrate the ViPi portal's user interface across different sections:

- Home Page:** Displays a user profile (Evernote), a list of recent posts, and a search bar.
- About Page:** Provides information about the ViPi project, its goals, and the target audience.
- News Feed:** Shows a video message from a user, with details about the post and a comment section.
- Learning objects:** Displays a list of learning objects, including a video message and a search bar.
- Members Directory:** Shows a list of registered users, with a search bar and a filter for 'Last Active'.
- Search Results:** Displays a list of search results, including a video message and a search bar.

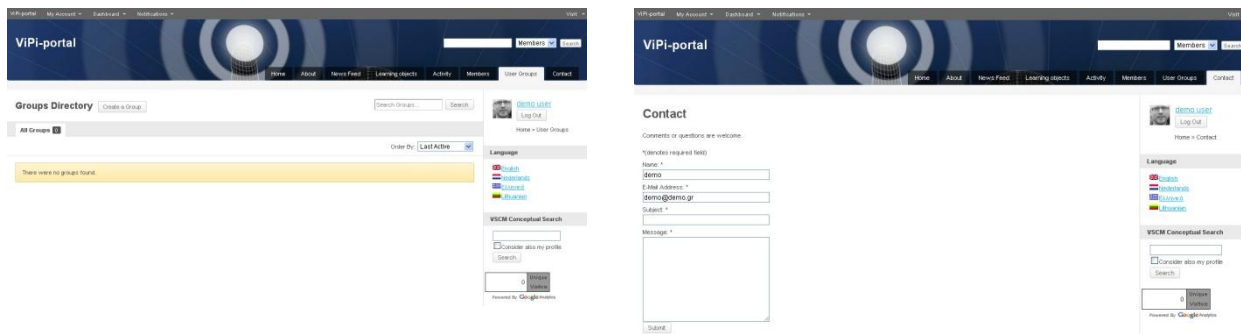


Figure 6: ViPi portal overview of pages

- **Search content:** Searching of content is available through the form presented in the following figure:



Figure 7: Searching of content within ViPi portal

User can search for registered users, groups or posts by using keywords. It should be noted that searching within posts exclude posts that are under the category “Learning Object”. Searching in Learning Objects and their content is available through the “Learning Objects” page. Search results are displayed in a list as follows:

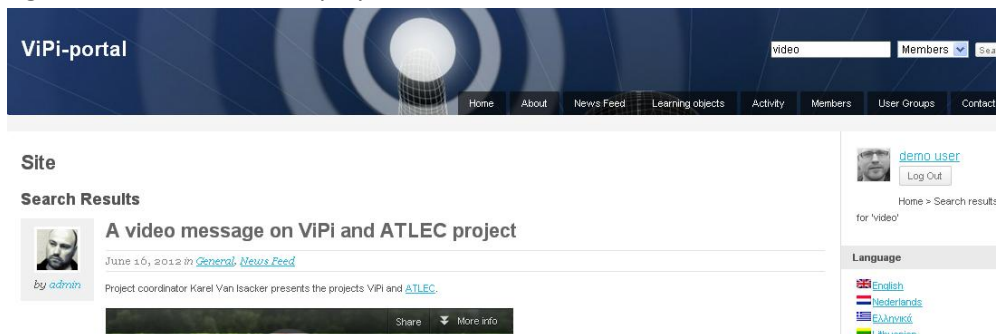


Figure 8: Example of search results

- **Language Selection:** Language selection enables multilingualism in ViPi portal by displaying all content in the selected language. Available languages that are supported are English, Dutch, Greek and Lithuanian. When a language is selected, e.g. Greek, then all portal content is displayed in Greek. If for example a content provider has not provided his post into a specific language, then appropriate messages appear indicating that this post is available in the language submitted.

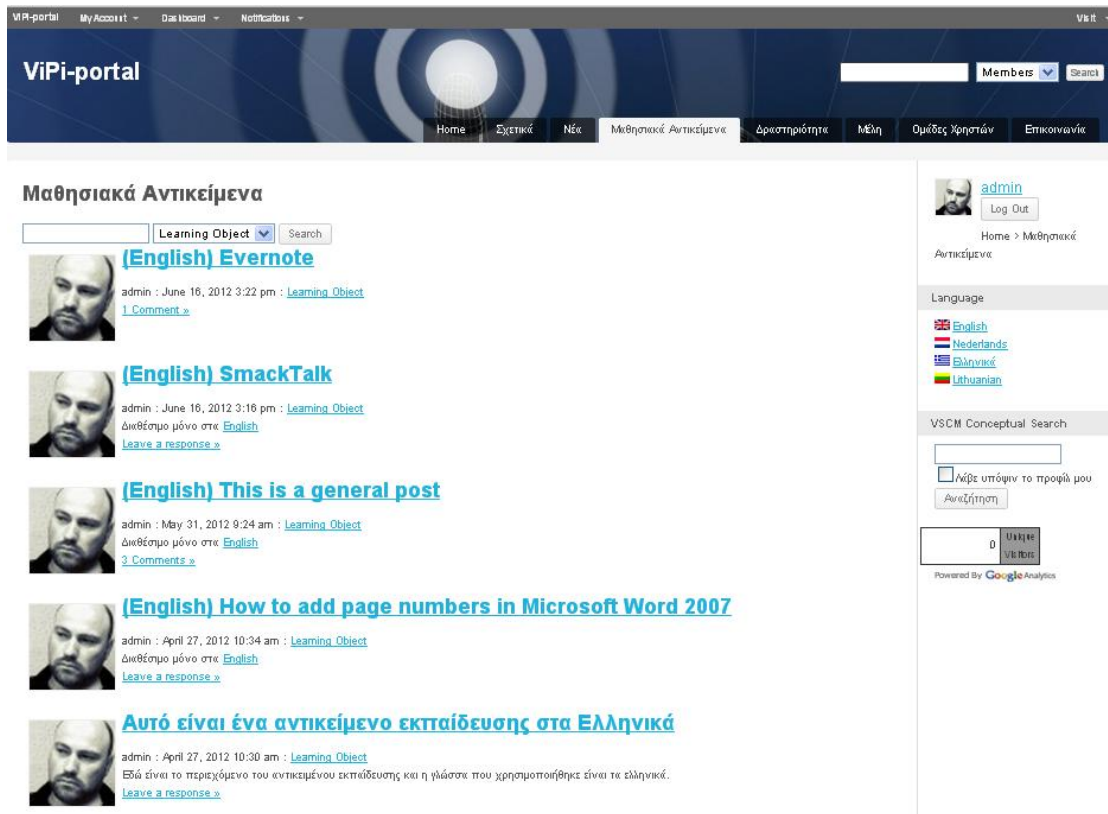


Figure 9: Example of Learning Objects page in Greek

- **Portal navigation:** Portal navigation shows the visitor's path to their current location.
- **Conceptual Search:** Provides semantic search capability within Learning Objects as it will be explained in detail in a later section.
- **Total unique portal visitors:** This widget provides the total number of unique visitors to ViPi portal.
- **Post:** All posts are displayed in the main page in descending order according to the date that they were submitted, excluding learning objects. Each post is consisted of a title, the content of post and its comments if available. Additionally, the category that a post belongs to appears after the date that it was submitted.
- **Post rating:** Users can rate each individual post using the relevant functionality located above the post title.



Figure 10: Rating of a Learning Object

The total number of votes along with the average rating score is displayed. Each individual user can rate a post only once. It should be noted that rating is only displayed and allowed on Learning Objects and not on every post of the platform.

3.4 The ViPi Learning Object Repository

An important part of the ViPi portal is the Learning Object (LO) repository that can be accessed through a dedicated Web page provided as an item in the central menu of the portal's homepage.

The repository will also be accessible through ViPi's mobile application (Android based). This repository strives to become a one-stop-shop of learning content created by trainers and other content providers, for the training of persons with disabilities, and providing trainers and other stakeholders with learning content in a variety of formats (URLs, courses, serious games, mobile games, etc.). Initially the repository will contain fully accessible training material for basic ICT skills, which was the application area of the ViPi project, however, the repository is implemented with built-in scalability to host content from any other area in the future and of any accessibility level.

Learning Objects is the terminology used in the ViPi portal to describe any piece of content that can be considered as content/material useful for the training of people. Therefore, LOs can span from structured courses, to small pieces of textual information, to multimedia files, to links to other resources, to serious and other games, etc.

The repository is implemented as multilingual, that is, the content can be provided in different and in more than one languages, as long as the content provider performed and uploaded the required translations.

The LOs are a specific category of content in the ViPi portal and, therefore, dedicated search functionality is available to the interested users to locate the content they look for. Details on how to use the search capabilities are given in a later section. LOs are stored in the repository as individual and well separated entities, with title, content and other metadata that support their discoverability. Furthermore, they are all linked to specific categories of skills (currently only basic ICT skills), as well as, to specific types of disabilities, devices, etc. The LOs are expected to be used by trainers in their teaching either using the ViPi curriculum or using their own curriculums and practices, but also by trainees that want to find material on specific topics and suggest these to their trainers or read on their own, possibly while taking the courses through the ViPi training environment.

Overall, the ViPi LOs repository is the hosting place for a vast number of training content entities/material and offers an easy-to-use search mechanism to facilitate the efficient discovery of what the user is exactly looking for.

3.5 Contributing to ViPi-Submitting New Content

Submission of new posts can be performed through the top bar of ViPi portal as shown in the following figure.

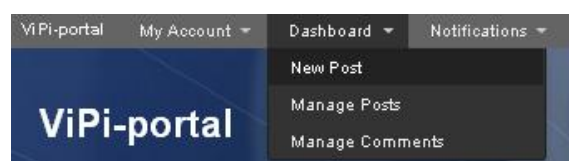


Figure 11: New post functionality

Once the user has selected to create a post a new page is opened in the user's dashboard containing a form that has to be filled in. First of all, the title of the new post should be entered in the preferred languages. Four distinct fields are displayed for the ViPi languages, and the user should enter the title of the post in the language(s) he/she intends to publish it.



Add New Post

Title (English)

Title (Nederlands)

Title (Ελληνικά)

Title (Lithuanian)

Figure 12: New post title in the four languages

The next step is to provide the main content of the post. The following form is used, where the user should fill in the content in the preferred languages as separated in distinct tab fields.

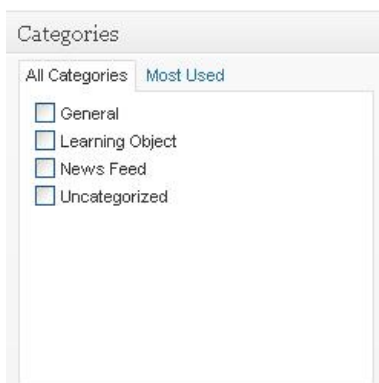


English Nederlands Ελληνικά Lithuanian Visual HTML

Figure 13: New post content in the four languages

Each content submission form includes text formatting options, as well as buttons for uploading images, files, videos etc. Presentation of HTML code is also supported by selecting the appropriate tab.

The next step is to select the category of the post. All available categories are listed in the box.



Categories

All Categories Most Used

☐ General

☐ Learning Object

☐ News Feed

☐ Uncategorized

Figure 14: Category selection when submitting a new post

Category selection is mandatory, so if no category is selected, and the user tries to submit the post for review, then the following popup message will appear.

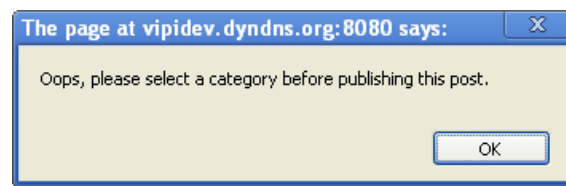


Figure 15: Category selection alert window

The category confirms the nature of the post. For example if “News Feed” is selected, then the post will be displayed under the “News Feed” page and should contain news information related to ViPi community. Accordingly, if the “Learning Object” category is selected then the content of this post should contain a new Learning Object, so this post will be displayed under the corresponding portal page. Multiple categories can be selected inheriting the properties of each category.

When the “Learning Object” category is selected, the following meta-box appears at the bottom of the page.

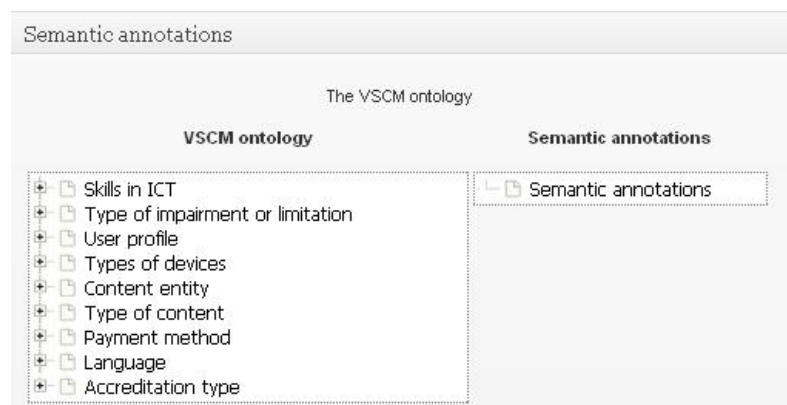


Figure 16: Semantic annotations meta-box

This meta-box provides the ability to semantically annotate the post using the ViPi ontology supported by the ViPi Semantic Content Management (VSCM) module, which will be presented in a later section. Two trees are then displayed: the VSCM ontology and the Semantic annotations tree. In the VSCM ontology tree, only the ontology concepts (classes and individuals) where the property value

<http://www.eurocyinnovations.com/ontologies/VSCM#forLOAnnotation>

is set to “YES” are being displayed. The tree is full expandable, and the annotation is performed by dragging concepts from the left tree and dropping them to the right one.

It should be noted that, instead of the real ontology concept name, a user-friendly text is displayed for each concept to assist the annotation procedure, translated into the user’s selected language (current supported languages are English, Dutch, Greek and Lithuanian).

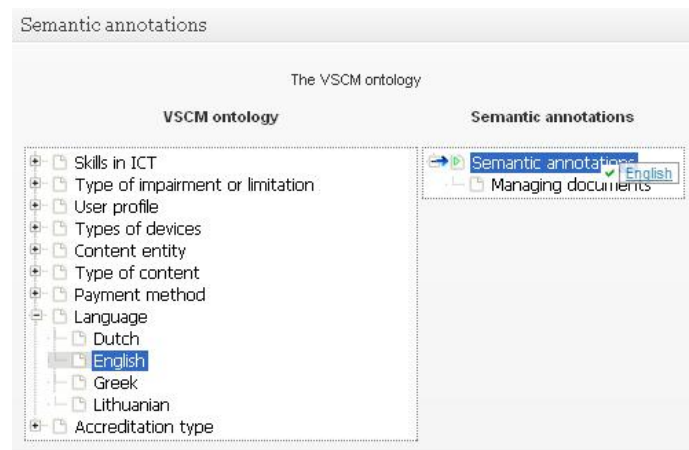


Figure 17: Drag and Drop functionality for annotating learning objects

Once a concept is dropped into the “Semantic annotations” tree, then the post is automatically annotated with this concept.

The user could then submit the post for review by the portal administrator.



Figure 18: Submitting a post for review

Alternatively, the user can preview the post, or save it for later usage. Once submitted, the status of the post is set to “Pending for review”, and the following entry appears in the administrator’s dashboard. If the administrator accepts the post, then it will be available through the ViPi portal to all users.

<input type="checkbox"/>	Title	Author	Categories	Tags	Languages	Date	SEO Title
<input type="checkbox"/>	test - Pending	demo	Learning Object	No Tags	English, Nederlands, Ελληνικά, Lithuanian	2012/06/19 Last Modified	

Figure 19: Pending posts in administrator dashboard

As already explained, the ViPi portal is the interface to the ViPi Learning Objects (LO) repository. The LO repository is expected to be enriched by training content to be contributed by persons external to the ViPi team. However, in order to maintain a high quality for the content in the repository, all potential training content providers are expected to follow the registration procedure offered on the ViPi portal. To further ensure the quality of the content, only approved and registered (on the ViPi platform) trainers are allowed to take the role of “Content Provider”.

3.6 Semantically-enabled Search

Simple search of content in ViPi portal (excluding Learning Objects) can be performed through the search functionality presented in Figure 7. Additionally, search within Learning Objects could be performed through the dedicated functionality integrated into “Learning Objects” page by using the two widgets of the ViPi platform that are presented in the next sections.

3.6.1 Conceptual Search

This widget performs semantic queries according to the text inserted, in order to retrieve annotated Learning Objects. The user could enter the desired keywords in the text box provided, and by selecting the “Search” button, the widget retrieves the annotated LOs and display them into a list.



Figure 20: The conceptual search component

It should be noted that conceptual search is independent of the language selected. The component can also provide more personalized results, by taking into account the profile annotations defined upon the registration of the user. As may be seen in the figure below, the user could select the “Consider also my profile” checkbox. Then, only LOs relevant to both the search keywords & the user profile will be displayed.

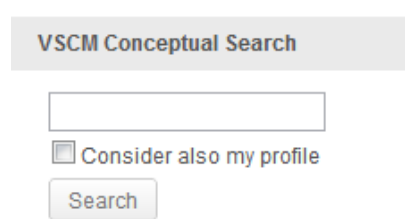


Figure 21: The “Consider also my profile” option

3.6.2 Semantic Navigation

The semantic navigation component provides a widget that allows navigation through the ViPi ontology.

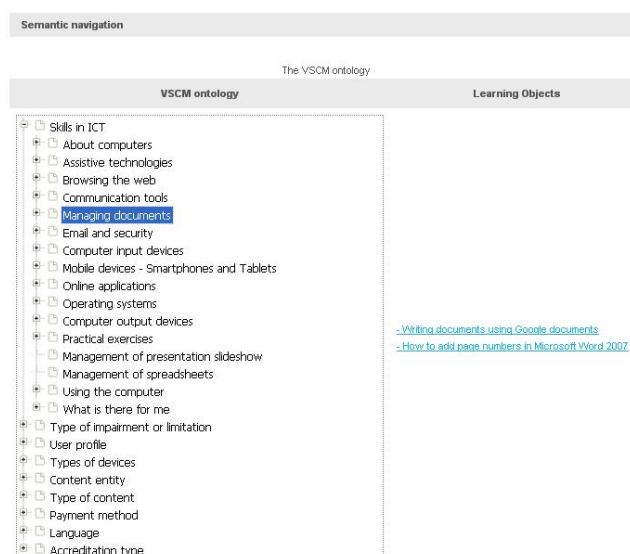


Figure 22: The semantic navigation component

By selecting classes or instances of the ontology tree, the user may view the posts annotated with this class. Annotated posts are displayed in the right column, named “Learning Objects” and are actually links to these LOs. Similar to the semantic annotation tree, user-friendly names are also used to present the ontology concepts, while the tree content is presented in the user selected language.

3.7 Connection with Popular Social Networks

Users could publish ViPi platform posts into three widely known social networks: Facebook, Twitter and Google+. In every post page, the following buttons appear:



Figure 23: Connection with social networks

When pressing one of the social network buttons, a login form appears as popup window where the user should enter his/her own credentials. If user has already signed in the same browser session, then the sign-in popup message is skipped and the post appears in the corresponding social network page.

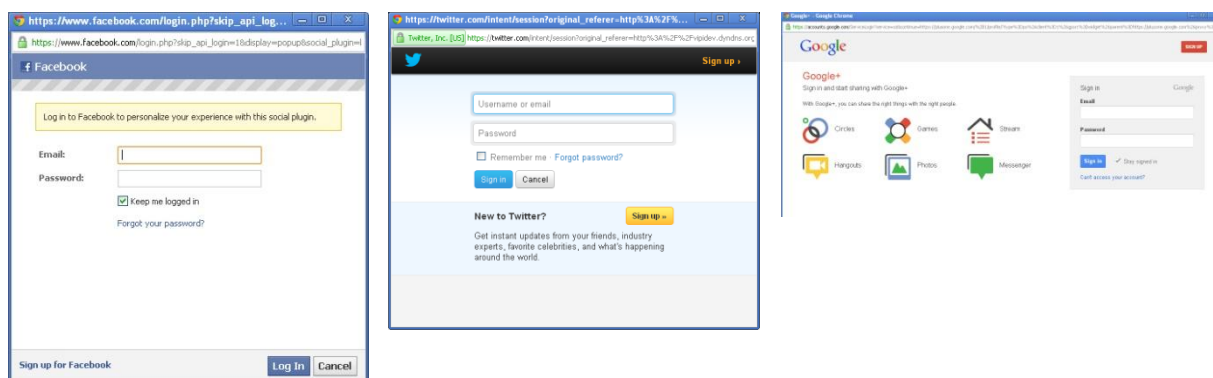


Figure 24: Social networks sign in pop-ups

3.8 Printing Posts and Exporting to PDF

Each post can be printed and exported to the PDF format by pressing the buttons appearing at the bottom of each post.



Figure 25: Printing and exporting to PDF functionality

Once the print button is pressed, the following pop-up window appears where the user can select either to print or export the selected post to the PDF format.

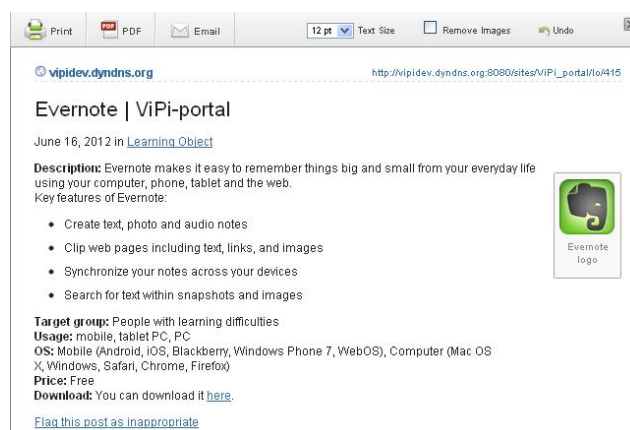


Figure 26: Printing and exporting to PDF pop-up window

3.9 Using the ViPi Online Training Environment

The core accessible training environment is offered through the ViPi platform, supported by ATutor. This learning management system was designed with accessibility as a priority. A wide range of features ensure assistive technology users can participate fully in learner, instructor, and administrative activities.

Once a user is logged into the ViPi platform, s/he can seamlessly access the training environment. The different roles a user can have in the learning platform are the following:

- Trainees
- Trainers
- Administrators (ViPi Staff)

4 Adopted Technologies & Installation Steps

The following subsections present the components used and the necessary steps that need to be performed in order to properly install and deploy the integrated ViPi platform, along with addressing the issues that may be faced during the installation of these components/plugin-ins.

It is noted that in the following text the word “localhost” should be replaced with the IP of the server where the main back-end platforms (WordPress, ATutor etc.) are going to be installed.

4.1 Installation of Xampp

Xampp is an easy to install **Apache** distribution containing the **MySQL** database management system, the **PHP** web scripting language and the **Perl** programming language. This distribution is used to host the backend, WordPress and ATutor, platforms, as well as the backend repository (MySQL database). The latest (currently 1.7.7) version of Xampp can be downloaded from the URL:

<http://www.apachefriends.org/en/Xampp.html>

Installation of Xaamp can be performed easily by following the wizard steps. Once finished, the MySQL and Apache Servers must be set up and running. It should be noted that Xaamp is only a recommendation due to its user friendly interface, thus any other recent MySQL or Apache implementations can be adopted.

4.2 Installation & Deployment of WordPress

Before proceeding with the installation of WordPress, a new database that will host the WordPress platform shall be created within the MySQL database system, e.g. “WordPress”, coupled with a database user, e.g. “WordPressUser”.

The next step is to download the latest version -currently 3.3.1 is used- of **WordPress** from the URL:

<http://WordPress.org/download/>

In order to install WordPress, the downloaded WordPress compressed file shall be uncompressed into the “**htdocs**” folder of Xampp installation (or into the appropriate folder according to the apache web server used). For setting up the WordPress instance, the file “**wp-config-sample.php**” has to be renamed to “**wp-config.php**” and the following entries have to be edited by using a text editor:

Editing Steps (wp-config.php)
define('DB_NAME', 'database_name_here');
define('DB_USER', 'username_here');
define('DB_PASSWORD', 'password_here');

Table 1: Editing of wp-config.php for first time installation of the WordPress

, where, adopting the example details given above, the text 'database_name_here' should be replaced with the name of the database WordPress will be installed (e.g. 'WordPress'), the

'username_here' the user name that has full access to 'database_name_here' (e.g. 'WordPressuser') and the 'password_here' with the password of the MySQL user.

Finally, in order to create the necessary tables in the database of WordPress, the entries of the URL: <http://localhost/WordPress/wp-admin/install.php> should be properly filled in. Once installation is completed, the portal is ready for usage and configuration.

4.2.1 Enabling Permalinks

Permalinks are the permanent URLs to the individual content entities of WordPress (posts, pages, categories, etc). A permalink is what another webpage will use to link to a page (or section) in the subject WordPress instance, or how a link could be sent in an e-mail message. Thus, it is very essential during the installation procedure to enable permalinks, under **Settings->Permalinks**.

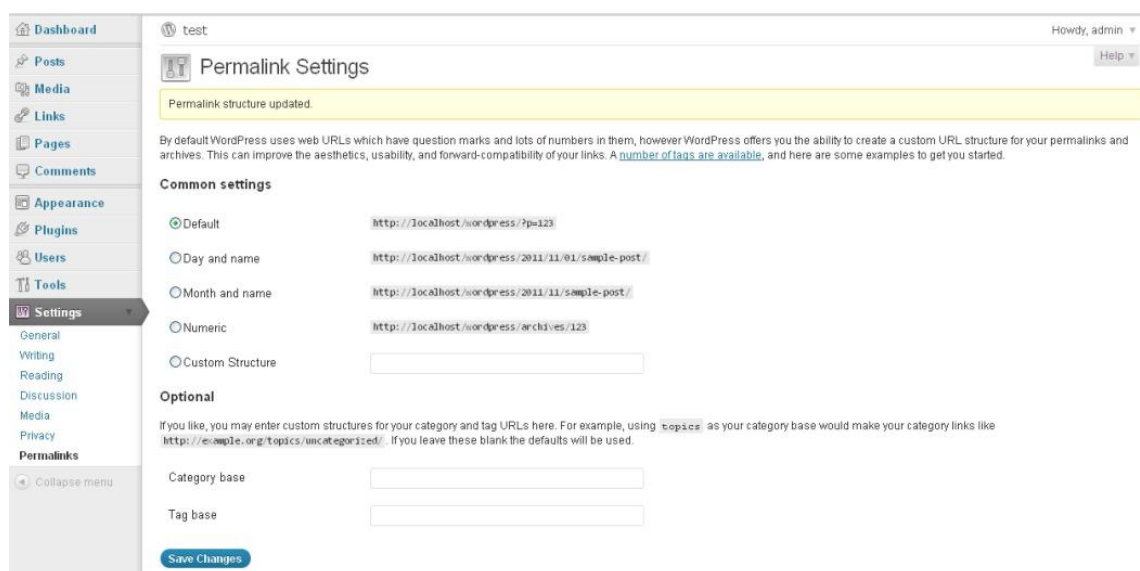


Figure 27: Enabling permalinks during the first time installation of the ViPi platform

The recommended ViPi portal permalink structure is:

`/category%/post_id%`

which should be placed into the **“Custom Structure”** field in the above form.

When a user submits a post, its permalink is displayed in the editor and could be used as a reference by other users.



Figure 28: Permalinks in a post message

4.2.2 XML-RPC Services Export & Deployment

One of the core functionalities supported by the WordPress platform is the exposing of several functionalities with XML-RPC services. In order to have several services exposed, the remote publishing feature should be enabled. In that way, the existing XML-RPC interfaces will be exposed whereas new interfaces could be implemented, allowing for publishing or accessing the ViPi platform data remotely (e.g. from a mobile device).

XML-RPC interfaces could be enabled via the administrator view in the following menu: **Settings->Writing->Remote Publishing**.

The XML-RPC methods currently supported by WordPress could be found in the URL that follows:

http://codex.wordpress.org/XML-RPC_wp

A variety of XML-RPC clients are available (most of them distributed as open source) so far for publishing remotely content in a WordPress installation. These include clients for PC/Mac terminals and mobile phones (Android, iPhone, Blackberry, Symbian, palmOS, J2ME). A list of clients are presented in

http://codex.wordpress.org/Weblog_Client

whereas, the **WordPress for Android** open source project was used to test the WordPress for Android mobile devices:

<http://Android.wordpress.org/>

This library was tested successfully to ensure the native WordPress & BuddyPress XML-RPC services deployment. Finally, it should be noted that the XML-RPC interface has been extended by the ViPi Semantic Content Management (VSCM) plug-in developed within WP4 (see Section 5.2.6), in order to expose additional functions to the mobile and other clients via the web and especially fit into the ViPi mobile application needs.

4.3 Installation & Deployment of the BuddyPress Plug-in

BuddyPress is a WordPress plug-in that offers social-networking functionality. More specifically, it supports creating profiles, posting messages, making connections, creating and interacting in groups, etc. The plug-in can be installed through the WordPress dashboard, as illustrated in Figure 29.

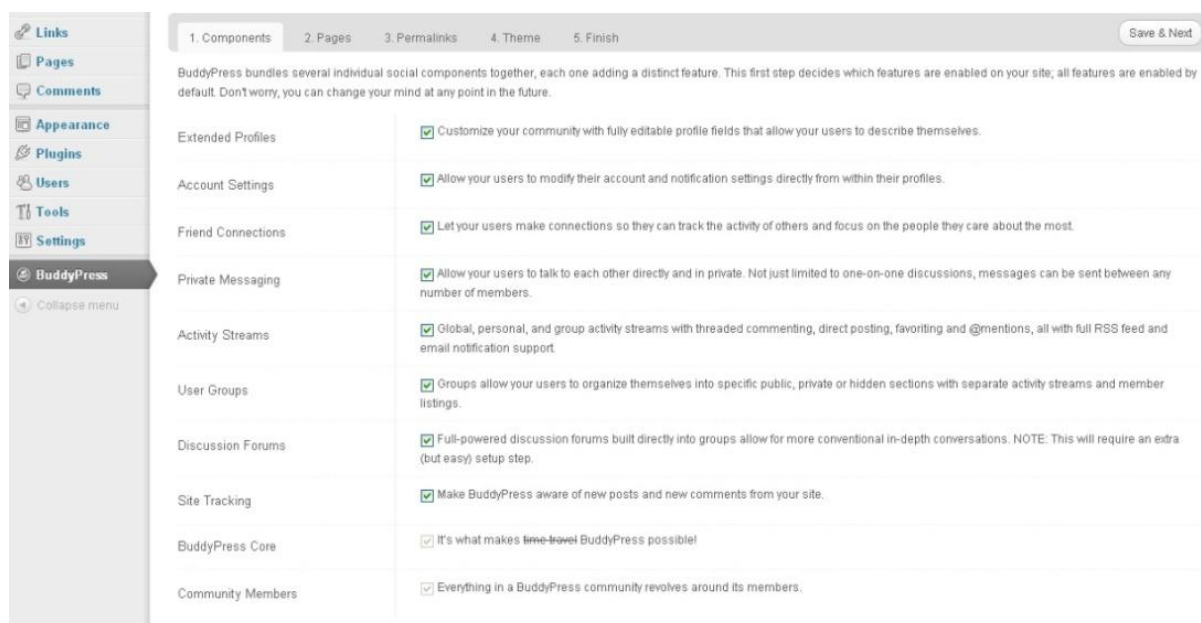


Figure 29: WordPress with BuddyPress as part of the ViPi platform installation procedure

In order to take advantage of the plug-in's full functionality, a BuddyPress compatible theme should be loaded. For this reason the BuddyPress default theme is recommended, but additional themes can be tested and activated as required. The major functionalities that have been installed are the following:

- Fully customizable profile for each user of the ViPi platform

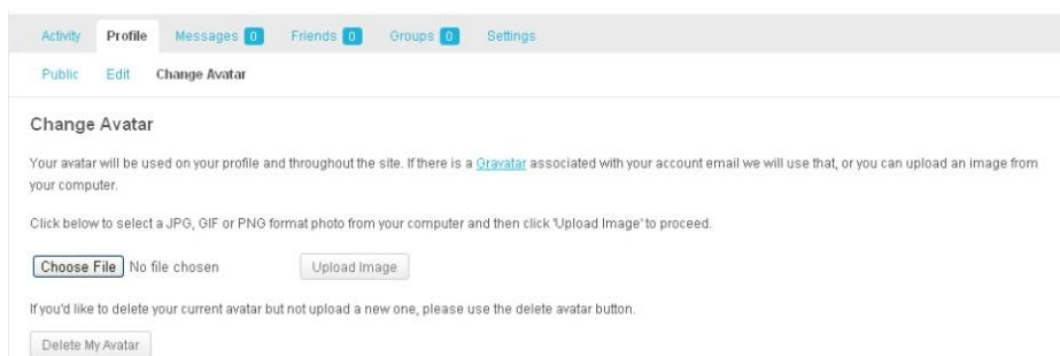


Figure 30: Customization of a registered user, fully synchronized with the ViPi WordPress database

- Private messaging between page users



Figure 31: Functionality of sending/receiving private messages within the ViPi platform

- Friends connections and Group activities



Figure 32: Adding friends and group-based activities within the social networking section of the ViPi platform

It should be noted that **BuddyPress user accounts** are fully synchronized with the **WordPress accounts**. BuddyPress uses its own tables within WordPress database for storing its data, e.g. **wp_bp_groups** table. During the installation procedure, no overlaps or integration issues are expected to be encountered.

4.3.1 BuddyPress XML-RPC Services Export & Testing

The BuddyPress plug-in provides a comprehensive XML-RPC interface, through the **BuddyPress XMLRPC – Receiver** plug-in. This can be downloaded from the following URL:

<http://wordpress.org/extend/plugins/buddypress-xmlrpc-receiver/>

The following functions are currently supported:

- **bp.verifyConnection:** checks if connection works
- **bp.getNotifications:** returns notifications like new messages, new friends, followers etc.
- **bp.updateProfileStatus:** updates the status of a profile
- **bp.getActivity:** gets various activity stream items
- **bp.updateExternalBlogPostStatus:** sends an activity stream update field under blogs
- **bp.deleteExternalBlogPostStatus:** deletes the activity update related to an already posted activity record
- **bp.getMyFriends:** returns the list of friends
- **bp.getMyFollowers:** returns the list of followers
- **bp.getMyFollowing:** returns the list of following
- **bp.getMyGroups:** returns a list of groups

As already mentioned, the BuddyPress XML-RPC services have been enabled and successfully tested/validated with desktop and mobile clients (see Section 4.2.2).

4.4 Installation & Deployment of ATutor

The **ATutor** Learning management tool can be downloaded from the following URL:

<http://ATutor.ca/ATutor/download.php>

The downloaded compressed file should be extracted into the “**htdocs**” folder of Xampp. By pointing a web browser to the URL:

<http://localhost/ATutor>

the user is prompted to follow the procedure to install ATutor, as illustrated in Figure 33 to Figure 35:



Figure 33: First time installation of the ATutor learning platform



Figure 34: Successful installation of ATutor as part of the ViPi platform

Following the instructions provided to the installation wizard, the user has to provide specific details concerning the ATutor database storage:

*Database Hostname: Hostname of the database server. Default: localhost	<input type="text" value="localhost"/>
*Database Port: The port to the database server. Default: 3306	<input type="text" value="3306"/>
*Database Username: The username to the database server.	<input type="text" value="wordpress"/>
*Database Password: The password to the database server.	<input type="text" value="wordpress"/>
*Database Name: The name of the database to use. It will be created if it does not exist. Default: atutor	<input type="text" value="atutor"/>
? Table Prefix: The prefix to add to table names to avoid conflicts with existing tables. Default: AT_	<input type="text" value="AT_"/>

Figure 35: Information details for the ATutor backend installation (database installation)

By providing the necessary details for the administrator and the user, the installation is completed successfully.

It is recommended to set the installation folder of ATutor inside the WordPress installation folder, to allow the **Backup WordPress** plug-in, presented later in Section 4.5.4, perform backup to ATutor files also. For the same reason, it is also recommended to use the WordPress database to install the ATutor database (Figure 35).

4.5 Installation of Selected WordPress Plug-ins

The current section provides information about the various available WordPress plug-ins that have been installed and deployed in the ViPi platform to extend its functionality, improve the user experience and increase accessibility of the available features. All the plug-ins included, have been installed and tested and the results are described below.

4.5.1 WP-Post Ratings Plug-in

This plug-in is used for rating posts, pages or comments.

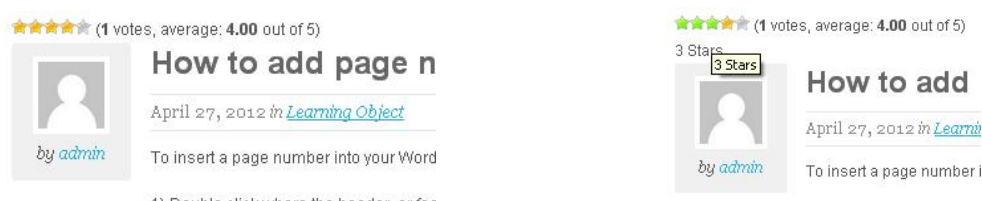


Figure 36: Rating plug-in installed as part of the ViPi platform

The average rating along with the total number of votes is displayed above the post title. To rate a post, the user has to move the mouse above the stars displayed and click on the desired star rating. The only limitations imposed, is that rating is only allowed to posts that are under the category “Learning Objects” and individual users can rate each LO one time only.

Use within the ViPi platform:

- Rating of LOs by users (people with disabilities and/or trainers). This will enable other users to choose specific LOs over others covering the same or similar subject.

4.5.2 Multilingual Support

Multilinguality in ViPi portal is ensured with the use of the **qTranslate** plug-in. It supports a variety of languages as can be seen in Figure 37 .
















Flag	Name	Action		
	Deutsch	Disable	Edit	Delete
	English	Disable	Edit	Default
	中文	Disable	Edit	Delete
	suomi	Enable	Edit	Delete
	Français	Enable	Edit	Delete
	Nederlands	Enable	Edit	Delete
	Svenska	Enable	Edit	Delete
	Italiano	Enable	Edit	Delete
	Română	Enable	Edit	Delete
	Magyar	Enable	Edit	Delete
	日本語	Enable	Edit	Delete
	Español	Enable	Edit	Delete
	Tiếng Việt	Enable	Edit	Delete
	العربية	Enable	Edit	Delete
	Português	Enable	Edit	Delete
Flag	Name	Action		

Figure 37: Management of supported languages within the qTranslate plug-in

Since ViPi portal main languages are English, Dutch, Greek and Lithuanian, some actions have to be performed for configuring this plug-in successfully. English language is by default enabled, and Dutch can be easily enabled by the menu provided in the above figure. Greek and Lithuanian languages are not contained into the default qTranslate installation, thus have to be inserted manually. For this reason, the following fields have to be filled in:



<p>Add Language</p> <p>Language Code <input type="text" value="el"/> <small>2-Letter ISO Language Code for the Language you want to insert. (Example: en)</small></p> <p>Flag <input type="text" value="gr.png"/>  <small>Choose the corresponding country flag for language. (Example: gb.png)</small></p> <p>Name <input type="text" value="Ελληνικά"/> <small>The Name of the language, which will be displayed on the site. (Example: English)</small></p> <p>Locale <input type="text" value="el_EL"/> <small>PHP and Wordpress Locale for the language. (Example: en_US) You will need to install the .mo file for this language.</small></p> <p>Date Format <input type="text" value="%A %B %e %q, %Y"/> <small>Depending on your Date / Time Conversion Mode, you can either enter a strtime (use %q for day suffix (st,nd,rd,th)) or date format. This field is optional. (Example: %A %B %e %q, %Y)</small></p> <p>Time Format <input type="text" value="%l:%M %p"/> <small>Depending on your Date / Time Conversion Mode, you can either enter a strtime or date format. This field is optional. (Example: %l:%M %p)</small></p> <p>Not Available Message <input type="text" value="Διέθεσιμο μόνο στα: %LANG;, : και %l"/> <small>Message to display if post is not available in the requested language. (Example: Sorry, this entry is only available in %LANG;, : and %l.) %LANG;<normal_seperator><last_seperator>% generates a list of languages seperated by <normal_seperator> except for the last one, where <last_seperator> will be used instead.</small></p> <p><input type="button" value="Add Language »"/></p>	<p>Add Language</p> <p>Language Code <input type="text" value="lt"/> <small>2-Letter ISO Language Code for the Language you want to insert. (Example: en)</small></p> <p>Flag <input type="text" value="lt.png"/>  <small>Choose the corresponding country flag for language. (Example: gb.png)</small></p> <p>Name <input type="text" value="Lithuanian"/> <small>The Name of the language, which will be displayed on the site. (Example: English)</small></p> <p>Locale <input type="text" value="lt_LT"/> <small>PHP and Wordpress Locale for the language. (Example: en_US) You will need to install the .mo file for this language.</small></p> <p>Date Format <input type="text" value="%A %B %e %q, %Y"/> <small>Depending on your Date / Time Conversion Mode, you can either enter a strtime (use %q for day suffix (st,nd,rd,th)) or date format. This field is optional. (Example: %A %B %e %q, %Y)</small></p> <p>Time Format <input type="text" value="%l:%M %p"/> <small>Depending on your Date / Time Conversion Mode, you can either enter a strtime or date format. This field is optional. (Example: %l:%M %p)</small></p> <p>Not Available Message <input type="text" value="Not available in Lithuanian"/> <small>Message to display if post is not available in the requested language. (Example: Sorry, this entry is only available in %LANG;, : and %l.) %LANG;<normal_seperator><last_seperator>% generates a list of languages seperated by <normal_seperator> except for the last one, where <last_seperator> will be used instead.</small></p> <p><input type="button" value="Add Language »"/></p>
---	--

Figure 38: Insertion of Greek and Lithuanian languages into ViPi portal

Once ready, the necessary dictionary files have to be downloaded and inserted into WordPress installation. The URL for downloading dictionary files for qtranslate is:

<http://svn.automattic.com/wordpress-i18n/>

The downloaded .mo file for Greek language has to be copied to:

ViPi portal installation folder/wp-content/plugins/qtranslate/langs

with name "qtranslate-el_EL.mo".

When the plug-in is installed and properly configured, any post can be written in the available languages and published accordingly. It should be noted that the plug-in does not translate the content automatically, but it is up to the content providers to fill-in the translations needed.

The plug-in is coupled with a widget that enables the end-users to choose the language to be displayed. In case the translation of a content entity is not available (has not been provided by content providers) in the selected language, the default one is presented (which is usually the English one).

Use within the ViPi platform:

- Provide full support of all ViPi main languages, that is, Dutch, English, Greek, and Lithuanian. The initial content of the platform will be translated and populated in advance in all 4 languages. However, it will be the content providers' responsibility to translate any LOs (or other content entities) to be published at later stage.

4.5.3 Facebook, Twitter & Google+ Social Widgets

This plug-in allows for publishing content entities to social networks like **Facebook**, **Twitter** and **Google+**. Publishing content to social networks is allowed only for posts.




Figure 39: Social network integration in posts

Use within the ViPi platform:

- Publish specific type of content directly on selected social networks upon publication in the ViPi platform.

4.5.4 Backup WordPress

This plug-in is useful for creating backup and migrating the WordPress installation into another location. Automatic backup scheduling is defined every day. Additionally, backups on demand can be performed easily from **Tools->Backups** in the administrator's Dashboard.



Manage Backups

[Back Up Now](#) [Settings](#)

✓ Your **database & files** will be automatically backed up **once daily**. The next backup will occur at **11:00 pm** on **June 13, 2012** and be saved to **C:/xampp_1_7_7/htdocs/sites/viPi_portal/wp-content/backups**.

✓ Your site is **51 MB**. Backups will be compressed and should be smaller than this.

Settings

You can define **constants** in your **wp-config.php** to control some settings. A full list of **constants** can be found in the [help panel](#). Defined settings will not be editable below.

Automatic Backups: ☒ Backup my site automatically.
☐ No automatic backups.

Frequency of backups: Automatic backups will occur **Daily**

What to Backup: Backup my **database & files**

Number of backups: The last **10** backups will be stored on the server.

Email backups: A copy of the backup file will be emailed to this address. Disabled if left blank.

Excludes:
A comma separated list of file and directory paths that you do **not** want to backup.
e.g. file.php, /directory/, /directory/file.jpg

[Save Changes](#)

Figure 40: ViPi portal backup settings page

The backup process produces a **.zip** file within the **wp-content/backups** folder. In order to migrate the whole platform into another server, the compressed file must be unzipped into a proper folder. Additionally, the data of the MySQL database must be inserted. This can be performed with a database management tool (like **phpMyAdmin**). The file **database_wordpress.sql** created within this zip, must be imported to enable the database contain all the information of the previous host. Note that **utf-8** should be selected as character set of the sql file during the database import.

If the new database is migrated into a new one with different name, then the appropriate fields in **wp-config.php** should be modified accordingly.

Use within the ViPi platform:

- Supports the portability of the installed back-end components (WordPress, BuddyPress, ATutor, etc.).

4.5.5 Access Keys

This plug-in allows adding Access Keys to Category and Page navigation menus to make the website keyboard accessible. The Access Keys plug-in can be set up from **Posts->Access Keys** menu.

Current Access Keys

Pages:
Categories:

Current Access Keys

Pages:

Access Key for Activate 1

Access Key for Activity 2

Access Key for Forums 3

Access Key for Groups 4

Access Key for Members 5

Access Key for Register 6

Access Key for Sample Page 7

Categories:

Access Key for Uncategorized 8

Save

Figure 41: Access Key support for pages contained in the fully integrated ViPi platform

The administrator should assign a number to each access key as shown in Figure 41. Then by pressing Alt+Shift+ a number, the corresponding page will open, according to administrator's settings. The full list of keyboard key combinations for the most popular web browsers can be found here:

<http://anthologyoi.com/WordPress/plugin-ins/WordPress-access-keys.html>

Use within the ViPi platform:

- The plug-in can be used to activate access keys for the main menu items and other main items of the ViPi portal. Its extended use must be avoided in order not to introduce unnecessary complexity for the end-users.

4.5.6 Acronyms

This WordPress plug-in allows maintaining a list of acronyms from within the WordPress management interface. The instances of defined acronyms are automatically replaced in content entities and comments with the full HTML acronym tag, e.g.

`<acronym title="Hypertext Markup Language">HTML</acronym>.`

Add Acronym

Acronym
HTML
This is the acronym or abbreviation as it appears in your blog posts.

Full
Hypertext Markup Language
The full text is the expanded version of the abbreviation as it will appear in a tooltip displayed when a user hovers his/her mouse cursor over the abbreviation.

Add Acronym

Delete

Displaying 1-15 of 133

Acronym	Full
<input type="checkbox"/> AFAIK	As far as I know
<input type="checkbox"/> AIM	AOL Instant Messenger
<input type="checkbox"/> AJAX	Asynchronous JavaScript and XML
<input type="checkbox"/> AOL	America Online
<input type="checkbox"/> API	Application Programming Interface
<input type="checkbox"/> ASAP	As soon as possible

Figure 42: Using the acronym plug-in as part of the deployed ViPi platform

Then, if a content entity includes the acronym, a popup will be displayed showing the full text, when a user moves the mouse over the abbreviation.

Use within the ViPi platform:

- The plug-in can be used to enable acronyms for key words/phrases to be used within the ViPi platform. This will improve the user's navigation experience.
- The content providers that create content in specific languages, need to check whether acronyms already exist, and if not, create the necessary ones to use in their content entities. It is noted that the translations of acronyms in different languages, should be treated as separate individual acronyms.

4.5.7 All In One SEO Pack

This plug-in optimizes the WordPress blog for Search Engines (Search Engine Optimization).

Use within the ViPi platform:

- Although the ViPi portal does not have any pure marketing flavour, its appearance in results of search engines is essential to improve its popularity. That is, the users will not necessarily remember the URL to visit the ViPi platform but can use meaningful keywords to discover the portal through search engines.

4.5.8 Breadcrumb NavXT

This plug-in adds a breadcrumb navigation showing the visitor's path to their current location.



Figure 43: Breadcrumb NavXT plug-in functionality

Use within the ViPi platform:

- The plug-in was activated within the ViPi platform to improve the navigation experience of the end-users.

4.5.9 Print Friendly and PDF

This plug-in adds icons for allowing printing and transforming to PDF of a post and its comments. These icons are displayed at the bottom of each post and a popup windows appear when selected to provide the aforementioned functionality.



Figure 44: Adding printing functionality to the ViPi platform

Use within the ViPi platform:

- The print option is enabled for all pages of the ViPi platform.

4.5.10 BuddyPress Sidebar

This plug-in allows having multiple BuddyPress sidebars in a web site. It can be configured from the BuddyPress options and it extends the default WordPress sidebars area. For example, in the following figure, the Home, Activity, Members, Groups and Blogs sidebars have been added.

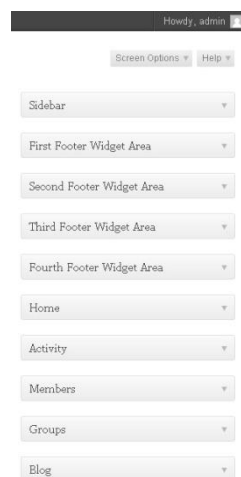


Figure 45: Adding five new sidebars

Use within the ViPi platform:

- The plug-in was activated within the ViPi platform to improve the navigation experience of the end-users.

4.5.11 BuddyPress Moderation

This plug-in adds buttons for reporting posts as inappropriate and gives a convenient way to moderators to view reports. When installed, the following button will appear under each post:



Figure 46: Moderation of posts

All the reports are summarized in the moderation page under BuddyPress menu.

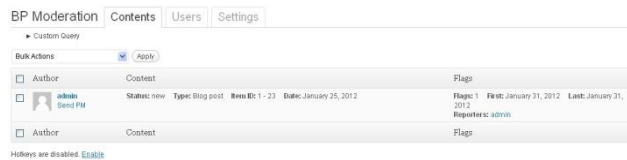


Figure 47: List of reported posts

Use within the ViPi platform:

- The plug-in was activated to provide a way of administering inappropriate content in the web page.

4.5.12 Q and A – FAQ

It allows insertion of FAQs into posts. This can be performed from the dashboard menu of WordPress.



Figure 48: Adding a new FAQ

It also allows the categorization of FAQs, so that the administrator could add the list of FAQs contained in a category to a post.



Figure 49: Managing FAQ categories and FAQs in a post

The insertion of a FAQ in a post could be achieved by using the shortcode **[qa]** into a post, or **[qacat="nameOfCategory"]** in order to display FAQs of a specific category.

Use within the ViPi platform:

- The plug-in was activated within the ViPi platform to provide useful help to users by answering their most common questions through a dedicated portal page. The questions and answers will be grouped in proper categories to improve the end-user navigation experience.

4.5.13 Google Analyticator

This plug-in enables **Google Analytics service** and includes widgets for displaying Analytics data. It provides statistical data about the Web site traffic by Google.

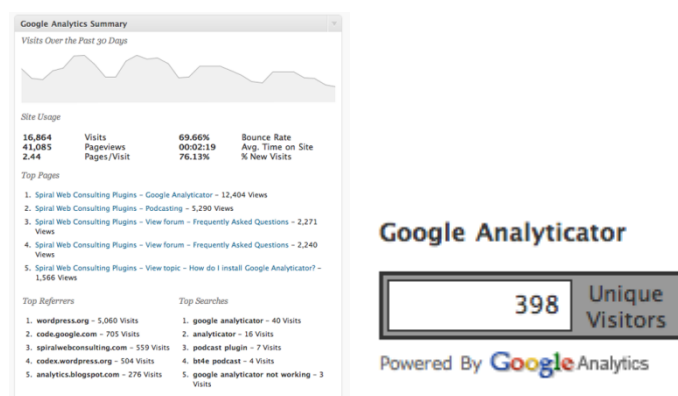


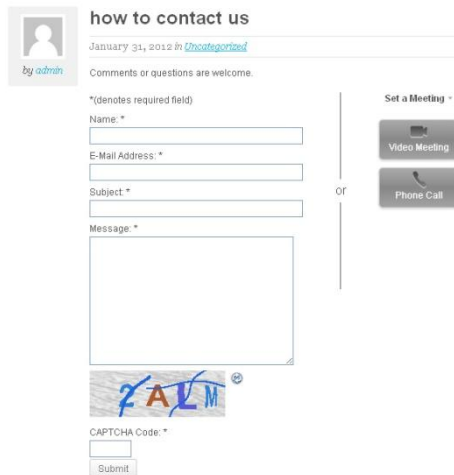
Figure 50: Examples of Google Analytics widgets.

Use within the ViPi platform:

- The plug-in was activated to provide statistical data related to the ViPi platform traffic, which will comprise useful information for the administrators of the platform, to plan the potential evolution or refinement of components.

4.5.14 Fast Secure Contact Form

This plug-in allows the administrator to easily create and add accessible contact forms to WordPress. The contact form allows users to send emails to a site's administrator, and also send a meeting request to talk over phone or video. An administration panel is present, where the webmaster can create and preview unlimited forms. In order to embed a contact form within a page or post, the creator needs to use the shortcode **[si-contact-form form='NumberOfForm']**, where **NumberOfForm** is the id of the form defined in the plug-in's settings page.



The screenshot shows a blog post titled "how to contact us" dated January 31, 2012, in the "Uncategorized" category. The post content includes a welcome message, a contact form with fields for Name, E-Mail Address, Subject, and Message, and a CAPTCHA. To the right of the form are buttons for "Set a Meeting", "Video Meeting", and "Phone Call".

Figure 51: Example of a contact form in a post.

Use within the ViPi platform:

- The plug-in was activated and specific contact form was created to provide an easy and accessible interface for end-users to contact the site's administrator and any other responsible person that will be deemed necessary.

4.5.15 Blog-in-Blog

This plug-in shows posts from a WordPress category on a page using shortcodes. For example, the "Learning Objects" page displays all posts under the category "Learning Object", by using the following shortcode:

```
[blog_in_blog category_slug='lo' num=5]
```

where "lo" is the slug of the category "Learning object".

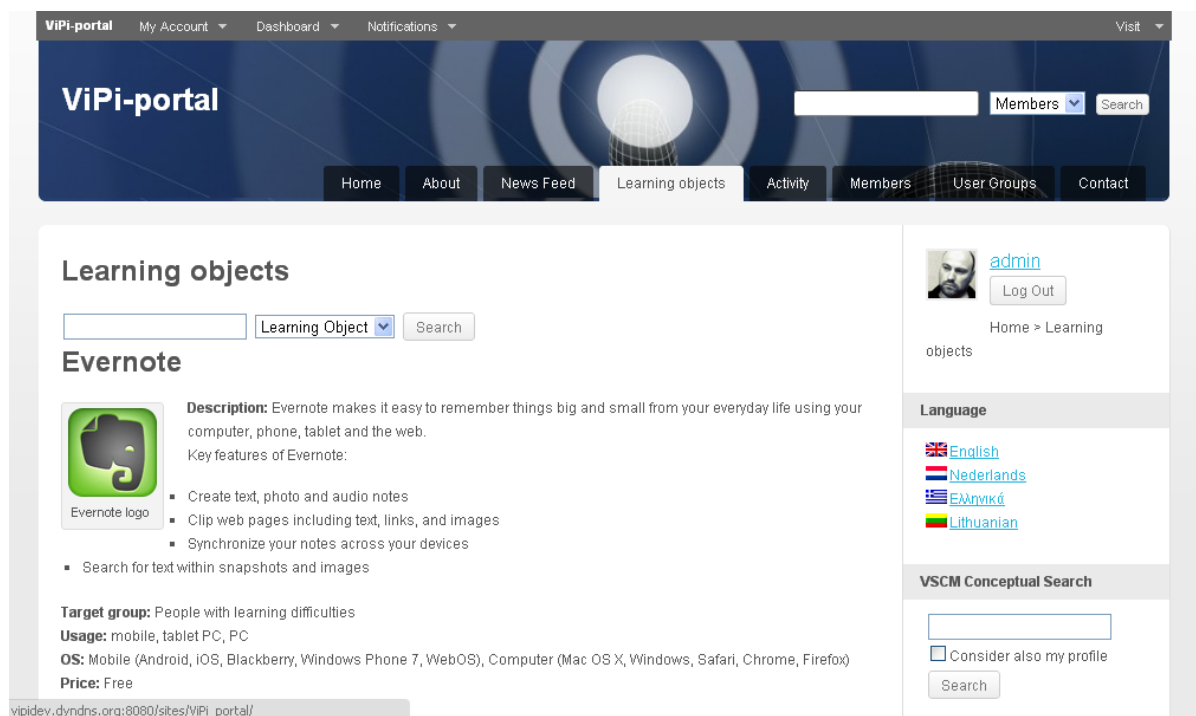


Figure 52: Example of a page that consists of specific category posts

Use within the ViPi platform:

- The plug-in is used to display posts of a specific category within a WordPress page.

4.5.16 Require Post Category

This plug-in ensures that every submitted post has an annotated category. If no category is selected when publishing a new post, then the following message will appear, prohibiting the user to upload post without selecting the category where it belongs.

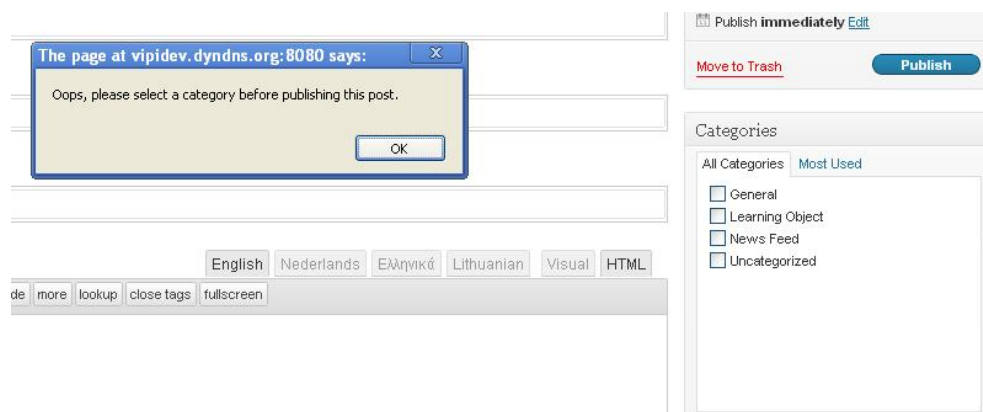


Figure 53: Category selection when publishing posts

Use within the ViPi platform:

- The plug-in is used to prohibit publishing posts that are not contained in any category.

4.5.17 Search by Category

This plug-in provides the ability to search in posts of a specific category.

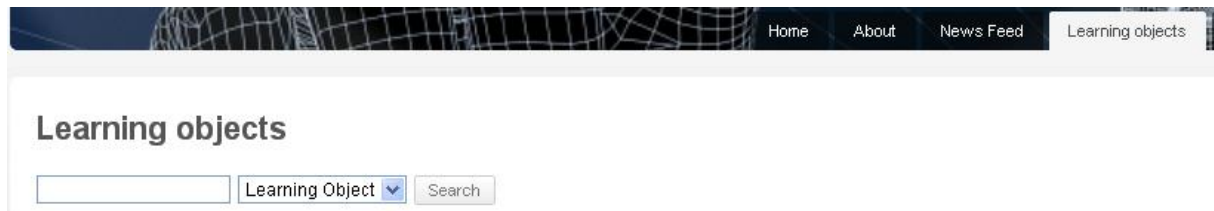


Figure 54: Searching of learning objects in a page

Use within the ViPi platform:

- The plug-in was activated to provide the search functionality within the “Learning Objects” category posts.

5 New Components Implemented inside ViPi

This section describes the components that have been designed and implemented in the framework of the project, to be part of the fully integrated ViPi platform. Any WordPress plug-ins that are going to be developed inside ViPi, are planned to be submitted to the **WordPress Plug-in Repository** as contribution of the ViPi consortium to the community.

The plug-ins will initially clarify the WordPress version(s) that they support. The strategy to be adopted by the ViPi consortium for the continuous support and maintenance of the plug-ins and the compatibility with newer WordPress versions, will be described in the exploitation plan of the ViPi project. In any case, since the plug-ins are made available to the WordPress community, support of newer versions can be offered by the WordPress community as well.

5.1 WordPress & ATutor Seamless Access

This implementation aims to support the necessary **single-sign-on** capability among the main ViPi platform components (WordPress, BuddyPress & ATutor). Once a user is logged into the ViPi platform through e.g. the WordPress portal, he/she is then able to access the associated social networking content (BuddyPress front-end) and also navigate in the ATutor eLearning environment without having to login again.

The single-sign-on implementation takes advantage of the unified access to the WordPress & ATutor database. Then, by overriding the login feature of the ViPi portal (WordPress back-end), the login to the eLearning environment (ATutor back-end) is performed in a transparent way and vice versa.

5.2 ViPi Semantic Content Management (VSCM) plug-in

The **ViPi Semantic Content Management (VSCM)** is a new WordPress plug-in implemented inside WP4 that enables the semantic annotation and discovery of LOs and user profiles within the ViPi platform, allows the implementation of semantic relations among these and advances the quality of navigation in the available LOs.

The plug-in implements the VSCM module approach adopted inside ViPi and especially during the WP4 development, which is described in detail in Section 6. Additionally, the VSCM plug-in takes advantage of the corresponding VSCM ontology created inside WP4 that is also detailed in Section 7.

The VSCM plug-in consists of the components presented in the following sections.

5.2.1 Ontology Management

The VSCM plug-in utilises the **Powl** framework and its underlying **RDF-API for PHP** for managing persistent ontologies. These frameworks can be found at the following addresses:

<http://sourceforge.net/projects/powl/>

<http://www4.wiwiw.fu-berlin.de/bizer/rdfapi/>

Both frameworks provide persistent management of ontologies in MySQL databases. For this reason, once the VSCM plug-in is installed for the first time, the following tables are created automatically in the WordPress database table:



Figure 55: MySQL tables for ontology management

These tables are used for storing the ontology models, namespaces, datasets and RDF triples that ontology is transformed to. It should be noted that these APIs have been properly adjusted to support UTF-8 text for storing the Greek and Lithuanian related ontology concepts. Additionally, querying within ontology is performed with the build in mechanism provided from these frameworks.

5.2.2 Administration Component

This component adds administration functionality within the WordPress backend platform, as shown in Figure 56.

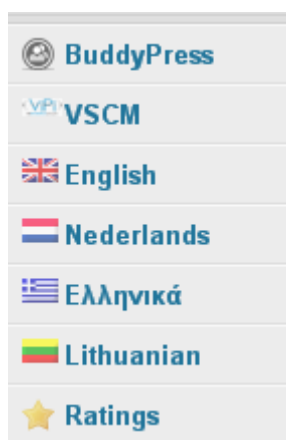


Figure 56: Plug-in preview within ViPi administrative view

The administration options are displayed in the VSCM menu of the administration panel that provides the necessary VSCM ontology management functionality

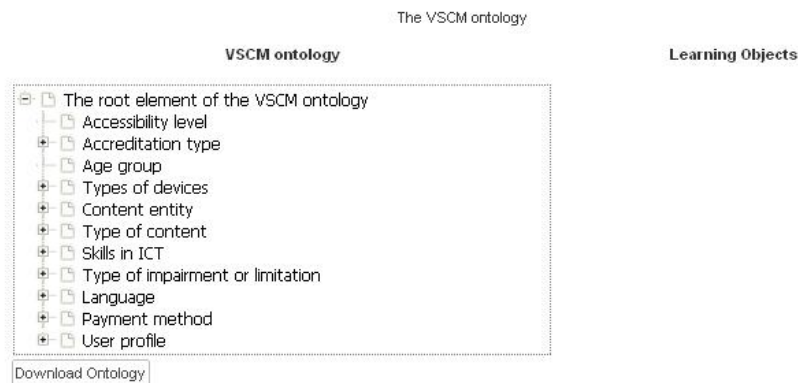


Figure 57: Ontology management in the administrator dashboard

When the plug-in is first time installed, the default ViPi ontology is automatically uploaded to the ontology repository. Thus, if the administrator navigates through the VSCM menu panel, he/she will be able to view all ontology classes/instances and its annotated Learning Objects. It should be noted that, instead of the real ontology concept name, a user-friendly text is displayed for each concept to assist the navigation procedure, translated into the user's selected language (current supported languages are English, Dutch, Greek and Lithuanian).

By moving the mouse cursor over a class/instance, the full URI of the class appears as a tool tip.

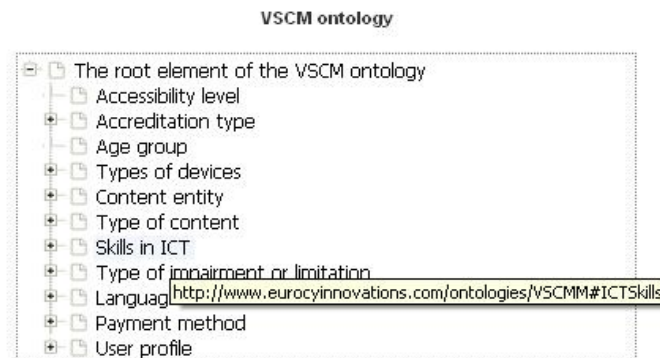
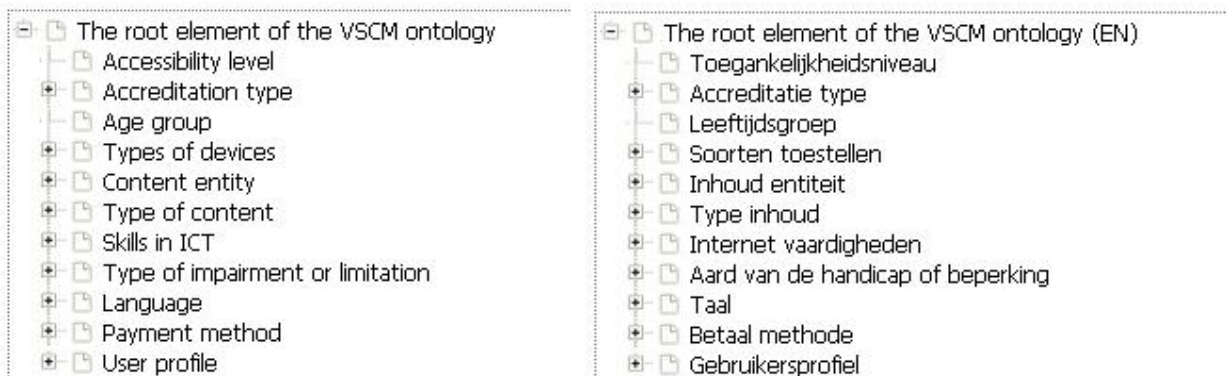


Figure 58: The default ViPi ontology tree structure and tooltip functionality

Switching to different languages can be performed by the default language selection of the administrator panel.



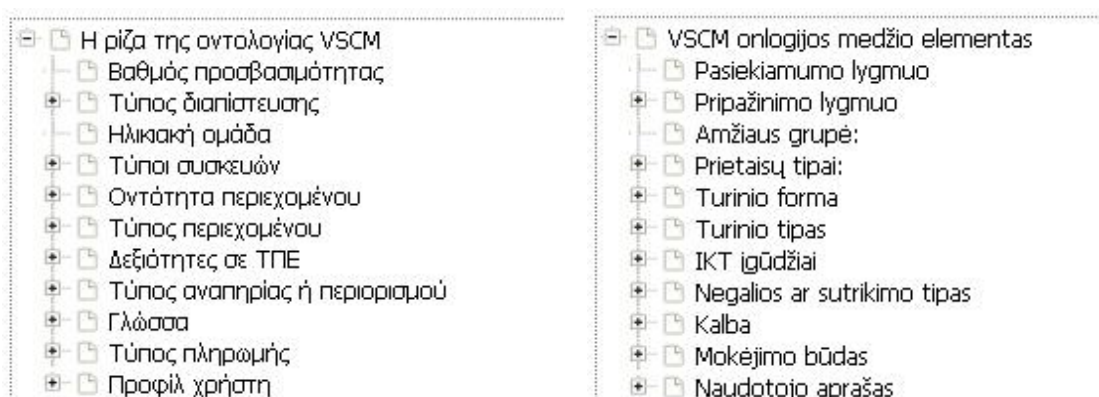


Figure 59: The default ViPi ontology in English, Dutch, Greek and Lithuanian languages respectively

Moreover, the button **“Download Ontology”** downloads the whole ontology from the database into a RDF file, named **“VSCM_Ontology.rdf”**.

5.2.3 Annotation of Learning Objects (LOs)

Annotation of LOs is an essential part of the ViPi portal and can be performed during the publishing of a new post belonging to the **“Learning Object”** category.

For this reason, the metabox containing the LO annotation functionality appears only when the **“Learning Object”** category is selected in the appropriate field. More details can be found in Section 3.5.

5.2.4 Annotation of User Profiles

Annotation of user profiles could be performed for each user registered in the ViPi platform. For this reason, a new entry on the WordPress profile page has been added, named **“Semantic profile annotation”**. Further details on the annotation procedure are presented in Section 3.2.

5.2.5 Conceptual Search

This widget performs semantic queries according to the text inserted, in order to retrieve annotated Learning Objects. The component functionality was presented in details in Section 3.6.1.

5.2.6 Semantic Navigation

The semantic navigation component provides a widget that allows navigation through the ontology. The component functionality was presented in details in Section 3.6.2

5.2.7 XML-RPC Interface

The VSCM plug-in also exports its functionality, by extending the WordPress XML-RPC interface, to enable access to ViPi portal content by the ViPi mobile application. For this purpose, the following functions have been developed & deployed.

XML-RPC Function	<i>String vscm.getOntology (boolean forLOAnnotation, boolean forUserAnnotation, String lang, String username, String password)</i>
Functionality	Returns the VSCM ontology together with the annotations (classes and instances) of the posted LOs in JSON format.
Arguments	<p>forLOAnnotation: A boolean parameter indicating whether the returned ontology will include the classes/instances where the property value</p> <p>http://www.eurocyinnovations.com/ontologies/VSCM#forLOAnnotation</p> <p>is set to "YES" (Actually includes the classes/instances used for the annotation of the LOs).</p> <p>forUserAnnotation: A boolean parameter indicating whether the returned ontology will include the classes/instances where the property value</p> <p>http://www.eurocyinnovations.com/ontologies/VSCM#forUserAnnotation</p> <p>is set to "YES" (Actually includes the classes/instances used for the annotation of user profiles).</p> <p>lang: The language in which the ontology will be returned. Currently, available options are: "en" for English, "nl" for Dutch, "el" for Greek and "lt" for Lithuanian.</p> <p>username: The username of the user that invokes the function. It is used for authentication purposes.</p> <p>password: The password of the user that invokes the function. It is used for authentication purposes.</p>
Return Value	<p>A String that contains the desired ontology in JSON format. A sample of the returned JSON format text follows:</p> <pre>[{ "title": "Skills in ICT", "uri": "http://www.eurocyinnovations.com/ontologies/VSCM#ICTSkills", "LOs": "", "children": [{ "title": "About computers", "uri": "http://www.eurocyinnovations.com/ontologies/VSCM#AboutComputers", "LOs": "134, 214", ... </pre> <p>In the above example, the</p> <p>http://www.eurocyinnovations.com/ontologies/VSCM#AboutComputers class was</p>

	used to annotate the LO posts with IDs 134 & 214.
XML-RPC Function	<i>String vscm.semanticSearch(String tags, boolean useProfileAnnotations, String username, String password)</i>
Functionality	Performs the conceptual search described in section 5.2.5 inside the posted LOs.
Parameters	<p>tags: A String including the search keywords.</p> <p>useProfileAnnotations: A boolean parameter indicating whether the search mechanism will take into account profile annotations in order to personalize the retrieved results according to user profile selections.</p> <p>username: The username of the user that invokes the function. It is used for authentication purposes.</p> <p>password: The password of the user that invokes the function. It is used for authentication purposes.</p>
Return Value	A list of String values, containing the IDs of the LO posts matching the query provided.

Table 2: XML-RPC interface of the VSCM plug-in

6 Annex I: ViPi Semantic Content Management (VSCM) Module

As already mentioned, the VSCM module enables the semantic annotation and discovery of LOs and user profiles within the ViPi platform and allows the implementation of semantic relations among these and advances the quality of navigation in the available LOs. The detailed VSCM approach adopted inside ViPi and especially during the WP4 development, which actually led to the creation of the VSCM plug-in (see Section 5.2), is described in a dedicated document delivered as an attachment to the current document.



7 Annex II: ViPi Semantic Content Management (VSCM) Ontology

The design of the VSCM module led to the creation of the corresponding VSCM ontology created inside WP4, which is also presented in a dedicated document delivered as an attachment to the current document.

8 Annex III: Evaluation of Additional WordPress Plug-ins

As part of the WP4 development procedure, several additional WordPress plug-ins have been tested. The evaluation results are shown below. In order to ensure necessity and proper functionality, extensive tests will be carried out during “WP6: Testing and Community building” & “WP7: Piloting and Community building” implementation to lead to the final selection of any additional plug-ins to be included in the integrated ViPi platform.

8.1 BuddyPress Mass Messaging

This plug-in allows sending messages to all BuddyPress users. It could be installed in ViPi platform if its functionality is considered necessary.

8.2 BuddyPress Like

This plug-in is yet not compatible with the WordPress version currently used (3.3.1).

8.3 BuddyPress Pending Activations

It allows bulk editing (manual activate, resend activation key, delete user) of pending user registrations on a single (non-multisite) WordPress installation of BuddyPress. For the time being, it does not seem to provide any significant functionality to the platform. However, it was installed in the latest WordPress version without any problem.

8.4 BuddyPress Verified

It allows the administrator to specify verified accounts or mark users as spam accounts. The applicability of this plug-in to the platform would be considered before its integration in the ViPi platform.

8.5 Dynamic Widgets

Dynamic Widgets provides control of the appearance of widgets in pages. It proved though that it does not provide any significant functionality to the platform.

8.6 Menubar

It provides single and multi-level menus for the WordPress site, styled with customizable menu templates. The applicability of this plug-in to the platform would be considered before its integration in the ViPi platform.

8.7 Google XML Sitemaps v3 for qTranslate

The plug-in generates a special XML sitemap, which will help search engines to better index the platform. This functionality was finally covered by All In One SEO Pack plug-in (See Section 4.5.7).

8.8 qTranslate Extension

This plug-in seems useful for supporting multilinguality on plug-ins, but any attempt to make it work inside ViPi platform proved unsuccessful (the plug-in is yet not compatible with the current version of WordPress used (3.3.1)).

8.9 qTranslate META

It allows to set multi-lingual META tags and a <title> override for the posts and pages. Its applicability in ViPi platform will be considered during WP6 & WP7 development.

8.10 BuddyPress and qTranslate

It adds drop down menu with qTranslate language to BuddyPress administration bar. Its applicability in ViPi platform will also be considered during WP6 & WP7 development.

8.11 Language Bar Flags

It displays a bar with configurable language flags to other language versions of the website. This plug-in seems to be useful since the ViPi platform will be maintained in many languages, but it is not yet compliant with the BuddyPress default theme.


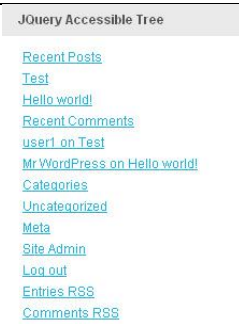
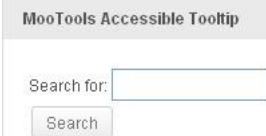
8.12 TDO Mini Forms

This plug-in allows to add highly customisable forms that work with Wordpress Theme to websites and allows non-registered users and/or subscribers (also configurable) to submit and edit posts and pages. Since the creation and submission of LOs will also be allowed by "content providers" outside the ViPi consortium, this plug-in could be used to hide from the end-user any irrelevant back-end parts that are usually displayed to platform administrators through the WordPress Dashboard.

9 Annex IV: Evaluation of Accessibility Plug-ins





As part of the development procedure, several plug-ins enabling accessibility and that were available in the WordPress plug-in Repository have been tested (mostly those developed within **Aegis & Accessible** FP7-ICT projects). The main outcome of this evaluation is that most of the plug-ins are outdated and considerable efforts should be made to get them work properly within the integrated ViPi platform. Hence, they will not be used. The evaluation results are provided below.

In order to ensure necessity and proper functionality, extensive tests will be carried out during “WP6: Testing and Community building” & “WP7: Piloting and Community building” implementation that would lead to the final selection of the plug-ins to be included in the integrated ViPi platform.

Plug-in Name	Description / URL	Screenshot	Comments / issues
MooTools Accessible Tooltip	An extension of the default search plug-in that uses the MooTools WAI-ARIA enabled tooltip (http://wordpress.org/extend/plugins/mootools-accessible-tooltip/)		-
JQuery Accessible Tree	A tree of a blog's recent posts, recent comments, categories and meta-info, that uses the JQuery WAI-ARIA enabled tree (http://wordpress.org/extend/plugins/jquery-accessible-tree/)		The layout is not displayed in a Tree like structure
JQuery Accessible Tooltip	An extension of the default search plug-in that uses the JQuery WAI-ARIA enabled tooltip (http://wordpress.org/extend/plugins/jquery-accessible-tooltip/)		-

Plug-in Name	Description / URL	Screenshot	Comments / issues
JQuery Accessible Tabs	A tab of the blog's recent posts, recent comments and archives that uses the JQuery WAI-ARIA enabled tab (http://wordpress.org/extend/plugins/jquery-accessible-tabs/)		The widget of this plug-in should be placed in the first footer widget area
JQuery Accessible Slider	A slider controlling the number of the blog's recent posts, that uses the JQuery WAI-ARIA enabled accordion and slider (http://wordpress.org/extend/plugins/jquery-accessible-slider/)		This plug-in does not work properly
JQuery Accessible Progressbar	A progress-bar showing up when the button group controlling the number of a blog's recent posts, recent comments and archives shown in an accordion is activated. It uses the JQuery WAI-ARIA enabled progress bar, accordion and button. (http://wordpress.org/extend/plugins/jquery-accessible-progressbar/)		This plug-in does not work properly
JQuery Accessible Menu	A menu bar of the blog's recent posts, recent comments, categories and meta. It uses the JQuery WAI-ARIA enabled menu bar. It's used in a wide widget area, like the "First Footer Widget Area" of the "Twenty Ten" theme (http://wordpress.org/extend/plugins/jquery-accessible-menu/)		Some visual improvements could be applied. It does not work on Google Chrome and auto complete feature does not

Plug-in Name	Description / URL	Screenshot	Comments / issues
			work properly
JQuery Accessible Dialog	<p>A dialog to add new tweets in a blog. It uses the JQuery WAI-ARIA enabled dialog.</p> <p>(http://wordpress.org/extend/plugins/jquery-accessible-dialog/)</p>		-
JQuery Accessible Carousel	<p>A carousel of the images in folder "wp-content/plugin-ins/JQueryAccessibleCarousel/lib/images" that uses the JQuery WAI-ARIA enabled carousel. All images (.jpg or .png or .gif) in the "wp-content/plugin-ins/JQueryAccessibleCarousel/lib/images" folder are first resized to 150x150 px in order to be used by the plug-in. Each image file name is used as the image's alt text (underscores "_" are replaced by spaces " ")</p> <p>(http://wordpress.org/extend/plugins/jquery-accessible-carousel/)</p>		This plug-in works smoothly but some visual adjustments have to be done in order to be compatible with the page theme
JQuery Accessible Autocomplete	<p>An extension to the default search plug-in that uses the JQuery WAI-ARIA enabled autocomplete</p> <p>(http://wordpress.org/extend/plugins/jquery-accessible-autocomplete/)</p>		-

Plug-in Name	Description / URL	Screenshot	Comments / issues
Fluid Accessible Sorting Grid	A sorting grid of the blog's recent posts, recent comments, categories and meta that uses the Fluid WAI-ARIA enabled sorting list (http://wordpress.org/extend/plugins/fluid-accessible-sorting-grid/)		The widget of this plug-in should be placed in the first footer widget area
Fluid Accessible Sorting List	A sorting list of the blog's recent posts, recent comments, categories and meta and uses the Fluid WAI-ARIA enabled sorting list (http://wordpress.org/extend/plugins/fluid-accessible-sorting-list/)		-
MooTools Accessible Tabpanel	A panel of tabs of the blog's recent posts, recent comments and archives that uses the MooTools WAI-ARIA enabled tabpanel (http://wordpress.org/extend/plugins/mootools-accessible-tabpanel/)		The widget of this plug-in should be placed in the first footer widget area
JQuery Accessible Checkbox	A checkbox controlling the number of the blog's recent posts, recent comments and archives shown in an accordion that uses the JQuery WAI-ARIA enabled accordion and checkbox (http://wordpress.org/extend/plugins/jquery-accessible-checkbox/)		This plug-in does not work properly

Plug-in Name	Description / URL	Screenshot	Comments / issues
MooTools Accessible Accordion	An accordion of the blog's recent posts, comments and archives that uses the MooTools WAI-ARIA enabled accordion (http://wordpress.org/extend/plugins/mootools-accessible-accordion/)		-
Fluid Accessible Progressbar	A progress-bar showing up when the button group controlling the number of the blog's recent posts, comments and categories shown in an accordion is activated and uses the Fluid WAI-ARIA enabled progress bar and the JQuery WAI-ARIA enabled accordion and button (http://wordpress.org/extend/plugins/fluid-accessible-progressbar/)		The widget of this plug-in should be placed in the first footer widget area
Fluid Accessible Sorting Portlet	(http://wordpress.org/extend/plugins/fluid-accessible-sorting-portlet/)	-	This plug-in could not be activated within WordPress (not valid header)
JQuery Accessible Accordion	An accordion of the blog's recent posts, comments and archives that uses the JQuery WAI-ARIA enabled accordion (http://wordpress.org/extend/plugins/jquery-accessible-accordion/)		-




Plug-in Name	Description / URL	Screenshot	Comments / issues
JQuery Accessible Button	WAI-ARIA enabled button plug-in for WordPress (http://wordpress.org/extend/plugins/jquery-accessible-button/)		This plug-in does not work properly

Table 3: Accessibility plug-ins tested inside the integrated ViPi platform