



# UNIVERSITÀ DI PISA

Scuola di Ingegneria

Corso di Laurea Triennale in Ingegneria Informatica

Progettazione e realizzazione su piattaforma Joomla  
di un modulo di reporting per la selezione di studenti  
stranieri nei Corsi di Laurea in Ing. Informatica

**Relatori:**

Prof. Mario G.C.A. Cimino

Prof.ssa Gigliola Vaglini

**Candidato:**

Marco Micera



# Indice

1. Specifiche.....	5
2. Sviluppo .....	6
2.1 Software, librerie e ambiente di sviluppo utilizzati.....	6
2.2 Struttura base del modulo.....	6
2.3 Struttura del file Excel di reporting .....	8
2.4 Descrizione del codice .....	9
2.4.1 Il metodo ajax() dell'oggetto jQuery .....	9
2.4.2 Download del modulo di reporting .....	9
2.4.3 Eliminazione dei dati riguardanti le richieste di ammissione.....	17
2.5 Modifica al modulo del form di ammissione.....	18
3. Installazione e integrazione del modulo nel sito web .....	19
3.1 Installazione.....	19
3.2 Integrazione nel sito web .....	21
4. Manuale d'uso .....	25
4.1 Apertura e chiusura delle iscrizioni .....	25
4.1.1 Apertura delle iscrizioni.....	25
4.1.2 Chiusura delle iscrizioni .....	28
4.2 Gestione delle candidature .....	30
4.2.1 Download delle candidature e selezione dei candidati.....	30
4.2.2 Eliminazione delle candidature precedenti.....	34
5. Conclusioni e possibili miglioramenti futuri .....	34
5.1 Conclusioni .....	34
5.2 Possibili miglioramenti futuri .....	35



# 1. Specifiche

Gli studenti stranieri che intendono accedere ai Corsi di Laurea Magistrale in Ingegneria Informatica (*Computer Engineering ed Embedded Computing Systems*) devono obbligatoriamente compilare un form presente sul sito ufficiale dei corsi <http://ce.iet.unipi.it>, nel quale sono richieste informazioni, oltre che di carattere generale, riguardanti la propria carriera universitaria e lavorativa.

APPLICATION FORM FOR MS IN  
COMPUTER ENGINEERING ADMISSION  
Please, read the [help](#).

Fields with "\*" (asterisk) are required.

\*First name:

\*Surname:

\*Gender:

Male

Female

Other

\*Date of Birth:

\*Month:  \*Day:  \*Year:

*Parte del form di ammissione*

I candidati hanno la possibilità di allegare file PDF (come documenti, curricula vitae, ecc.): essi vengono salvati su file system del web server, mentre le informazioni provenienti dal resto del form vengono inserite in un database.

Fino ad ora, il personale incaricato a svolgere la selezione dei candidati ha avuto il compito riportare tutti i dati su un apposito foglio Excel che permetteva loro di aggiungere ulteriori informazioni ricavate dalla visione dei documenti allegati e di effettuare l'operazione di selezione tra i vari candidati.

L'obiettivo del progetto è quello di automatizzare tale processo tramite la creazione di un modulo *Joomla!* di reporting in grado di generare un archivio contenente i file allegati dai candidati e un foglio Excel in grado di svolgere i compiti sopra riportati, automaticamente riempito con le informazioni dei candidati presenti sul database. Deve essere inoltre disponibile una procedura per l'eliminazione dei dati su database e file system: funzionalità utile dopo la fine delle selezioni dei candidati.

## 2. Sviluppo

### 2.1 Software, librerie e ambiente di sviluppo utilizzati

Per lo sviluppo di questo progetto è stato utilizzato il pacchetto open source XAMPP per la creazione di un web server locale, contenente PHP 5.6.24, Apache 2.4.23 e MySQL 5.6.30.

Come IDE è stato utilizzato Microsoft Visual Studio 14 con PHP Tools for Visual Studio 1.20.

Prima di iniziare lo sviluppo del modulo, è stata caricata una copia di backup del sito <http://ce.iet.unipi.it> sul web server locale ed è stato importato il dump del database di quest'ultimo.

È stato inoltre utilizzato il framework JavaScript jQuery e la libreria PHPEXcel, disponibile gratuitamente su GitHub all'indirizzo <https://github.com/PHPOffice/PHPEXcel>, che permette di operare su file Excel tramite linguaggio PHP. Entrambe queste risorse sono disponibili offline nella sub directory lib.

### 2.2 Struttura base del modulo

Per la creazione di un modulo *Joomla!* è stata seguita la guida sulla *Joomla! Documentation* disponibile su [https://docs.joomla.org/J3.x:Creating\\_a\\_simple\\_module/Developing\\_a\\_Basic\\_Module](https://docs.joomla.org/J3.x:Creating_a_simple_module/Developing_a_Basic_Module).

Come nome del modulo è stato scelto `mod_enrollments_report`, in quanto la convenzione adottata da *Joomla!* prevede il prefisso `mod_` all'inizio del nome di ogni modulo.

La struttura base del modulo *Joomla!* consiste in 4 file principali:

- `mod_enrollments_report.xml`

Contiene le informazioni sul modulo, tra cui i file che lo compongono, utili al *Joomla! installer* durante l'installazione. È stato inoltre aggiunto un file HTML vuoto per ogni directory, denominato `index.html`, che serve a non permetterne la navigazione.

Il contenuto di `mod_enrollments_report.xml` è riportato di seguito:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- From: docs.joomla.org/J3.x:Creating_a_simple_module/Developing_a_Basic_Module

This file contains information about the module.
It defines the files that need to be installed
by the Joomla! installer and specifies
configuration parameters for the module.
-->
<extension type="module" version="3.1.0" client="site" method="upgrade">
  <name>Reporting module for masters enrollments</name>
  <author>Marco Micera</author>
  <version>1.0.0</version>
  <description>Allows to download reports.</description>
  <files>
    <filename module="mod_enrollments_report">
      mod_enrollments_report.php
    </filename>
    <filename>helper.php</filename>
    <folder>config</folder>
    <folder>lib</folder>
    <folder>tmpl</folder>
    <folder>util</folder>
    <!-- Blank index files are included so that these directories cannot
        be browsed. -->
    <filename>index.html</filename>
  </files>
  <config>
  </config>
</extension>
```

- `mod_enrollments_report.php`

È l'entry point del modulo: in questo progetto ha come unico compito quello di includere il file di template `default.php` che contiene l'interfaccia grafica.

```
<?php
/* From: docs.joomla.org/J3.x:Creating_a_simple_module/Developing_a_Basic_Module
 *
 * This file is the main entry point for the module.
 * It includes the template which will display the module output.
 */

// Checks to make sure that this file is being included from the Joomla! application.
defined('_JEXEC') or die;

// Includes the template file 'tmpl/default.php'
require JModuleHelper::getLayoutPath('mod_enrollments_report');
```

- `helper.php`

Contiene la classe helper del modulo, usata in questo progetto solo per la comunicazione con il database. Possiede due semplici metodi per il recupero e la cancellazione di tutte le informazioni dalla tabella `g9f7i_student_form` del database `c17_multice`. Il codice è riportato di seguito:

```
<?php
/* From: docs.joomla.org/J3.x:Creating_a_simple_module/Developing_a_Basic_Module
 *
 * This file contains the helper class
 * which is used to do the actual work
 * in retrieving the information to be
 * displayed in the module (usually
 * from the database or some other source).
 */

class ModEnrollmentsReportHelper {
    public static function getData() {
        $db = JFactory::getDbo();
        $db->setQuery("select * from g9f7i_student_form");
        return $db->loadObjectList();
    }

    public static function deleteData() {
        $db = JFactory::getDbo();
        $db->setQuery("truncate table g9f7i_student_form");
        $db->execute();
    }
}
```

- `default.php`  
Consiste nel template del modulo, ovvero dell'interfaccia grafica attraverso la quale è possibile utilizzare le funzioni implementate. La struttura di tale interfaccia è ridotta al minimo per mantenerla semplice. Il modulo, una volta integrato e installato nel sito web ([come mostrato nel capitolo 3](#)), avrà il seguente aspetto:

## Enrollments report module

- > [Download admissions](#)
- > [Delete previous admissions](#)

*Interfaccia grafica del modulo di reporting*

L'aspetto fondamentale che permette il funzionamento dell'intero modulo si basa sull'invio di apposite richieste Ajax al web server ogni qual volta venga effettuato un clic su uno dei due link: ciò verrà approfondito nel [paragrafo 2.4 "Descrizione del codice"](#).

### 2.3 Struttura del file Excel di reporting

Il file Excel è composto dai seguenti fogli:

- `all_applications`  
Contiene l'elenco di tutti i partecipanti che hanno effettuato domanda.  
Per ogni partecipante ci sono tante colonne quante sono le informazioni necessarie per la selezione. Parte di queste informazioni vengono fornite grazie al form di ammissione, e il modulo si occupa di reperirle dal database e di inserirle in questo foglio.  
Altre informazioni sono invece solo ricavabili grazie alla visione dei documenti allegati forniti dai candidati.  
Per ogni partecipante, inoltre, è presente una colonna `valid` grazie alla quale è possibile scegliere se il candidato risulterà o no ammesso.
- `valid_applications`  
I candidati ammessi vengono riportati in questo foglio.  
A questo punto vengono visualizzate solo le informazioni più cruciali.
- `legenda`  
Contiene la legenda utile a interpretare correttamente il file Excel, come la legenda dei codici di status e la suddivisione dei valori di ranking delle università di provenienza dei candidati in varie categorie, con diversi pesi associati.



## 2.4 Descrizione del codice

Come già descritto nel precedente, l'entry point del modulo `mod_enrollments_report.php` provvede a includere il file di template `default.php` contenente l'interfaccia grafica.

In quest'ultimo file viene utilizzato il framework JavaScript jQuery, tramite la seguente inclusione:

```
<!-- jQuery -->
<script src="modules/mod_enrollments_report/lib/jquery-3.1.1.min.js"></script>
```

Vengono poi inseriti due link che permettono di richiamare le funzionalità descritte nel capitolo delle specifiche:

```
<h2>Enrollments report module</h2>
<ul>
  <li><a id="download" href="#">Download admissions</a></li>
  <li><a id="delete" href="#">Delete previous admissions</a></li>
</ul>
```

Gli id dei due tag HTML anchor permettono allo script jQuery incluso nello stesso file di effettuare le opportune richieste Ajax a seconda del link cliccato.

### 2.4.1 Il metodo `ajax()` dell'oggetto jQuery

jQuery mette a disposizione un metodo `ajax()` per poter effettuare richieste HTTP asincrone: la documentazione è disponibile su <http://api.jquery.com/jquery.ajax>. È stata utilizzata la versione con un unico parametro `settings`, che consiste in zero o più coppie chiave/valore che configurano la richiesta Ajax. In questo progetto sono state utilizzate solo due tipi di impostazioni:

- `url`: specifica l'URL al quale inviare la richiesta, in questo caso un file PHP.
- `success`: specifica la funzione da eseguire nel caso la richiesta sia stata soddisfatta con successo.

### 2.4.2 Download del modulo di reporting

Uno script jQuery in `default.php` si occupa della cattura dell'evento del clic e della richiesta Ajax corrispondente.

```
<script type="text/javascript" charset="utf-8">
  var utilDirectory = "../..../modules/mod_enrollments_report/util/";

  // Zip archive deleting delay
  var deleteAfter = 10 * 1000;

  $(
    $("#download").click(function() {
      jQuery.ajax({
        url: utilDirectory + "download.php",
        success: function(data) {
          // Downloads the zip file
          window.location.href = data;

          /* Deletes the generated zip archive
           after the user downloads it */
          setTimeout(function(){
            jQuery.ajax({
              url: utilDirectory + "deleteZipFile.php"
            });
          }, deleteAfter);
        }
      });
    });
  );
</script>
```

Il wrapper `$( ... )`; secondo la documentazione, è l'unica sintassi consigliata (dopo la versione 3.0) per consentire l'esecuzione di codice non appena il caricamento della pagina termina, in modo che il DOM sia in uno stato consistente, e quindi sicuro da manipolare.

Una volta effettuato il clic sul link "Download admissions", viene catturato l'evento da jQuery e viene inoltrata la richiesta al file `download.php` sul web server, nella sub directory `util`.

Questo script si occupa della costruzione dell'intero archivio contenente gli allegati e il file Excel, servendosi di classi di supporto, di seguito incluse a inizio script:

```
/* Creates the zip file containing
   the admissions data. */
require_once "../config/sheetConfig.php";
require_once "sheetManager.php";
require_once "../config/zipConfig.php";
require_once "zipManager.php";
```

Ciascuna delle due classi incluse fa uso di un file di configurazione (`sheetConfig.php` e `zipConfig.php`) nel quale vengono dichiarate alcune variabili necessarie per il funzionamento dell'applicazione.

```
// sheetConfig.php
<?php
    $sheetFileName = 'Reporting sheet.xlsx';
    $fileType = 'Excel2007';
    $sheetTemplateFile = '../sheet_template.xlsx';
    $firstRow = 2;
```

```
// sheetConfig.php
<?php
    $zipFileName = 'enrollments_report.zip';
    $attachmentsDirectory = '../../mod_student_form/uploads';
    $attachmentsFolderName = 'attachments';
```

In `download.php`, viene da subito creato il file Excel di reporting tramite la seguente istruzione:

```
$sheet = new SheetManager();
```

`SheetManager` è una classe dedicata alla gestione del file Excel: possiede un unico membro privato:

```
class SheetManager {
    private $sheet = null;
```

che consiste nell'oggetto che rappresenta il file Excel in elaborazione.

Il costruttore di questa classe provvede a effettuare una copia del file Excel di template per inserire successivamente tutte le informazioni prelevate dal database. Questo file sarà poi inserito nell'archivio zip che sarà disponibile al download.

```
function __construct() {
    // Copies the sheet template in order to modify it
    $this->copySheetTemplate();

    // Adjusts the sheet's style
    $this->setStyle();

    // Fills the all_applications worksheet
    $this->loadDatabaseData();

    // Saves Excel File
    global $fileType;
    global $sheetFileName;
    $sheetWriter = PHPExcel_IOFactory::createWriter($this->sheet, $fileType);
    $sheetWriter->save($sheetFileName);
}
```

Il primo metodo chiamato dal costruttore effettua una semplice copia del file Excel di template nella directory dello script. Tale file, una volta incluso nell'archivio, verrà eliminato. Tale metodo assegna inoltre all'unico membro privato sheet sopra descritto, il file Excel appena copiato.

```
function copySheetTemplate() {
    global $fileType;
    global $sheetTemplateFile;
    global $sheetFileName;

    // Copies the sheet template (single workbook version)
    if(!copy($sheetTemplateFile, $sheetFileName))
        exit("Sheet template not found.\n");

    $this->sheet = PHPExcel_IOFactory::createReader($fileType)->load(
        $sheetFileName
    );
}
```

Successivamente il costruttore assegna lo stile al foglio Excel. Per ora si tratta solo del livello di zoom, impostato al 70%, per consentirne una migliore visualizzazione:

```
function setStyle() {
    // Sets the zoom level on the applications worksheets
    $this->sheet->getSheetByName('all_applications')->getSheetView()->setZoomScale(70);
    $this->sheet->getSheetByName('valid_applications')->getSheetView()->setZoomScale(70);
}
```

Dopodiché vengono caricati tutti i dati del database e inseriti nel foglio Excel. A questo punto viene utilizzata la classe helper indicata dalla guida di Joomla!, tramite la seguente chiamata:

```
function loadDatabaseData() {
    global $firstRow;

    // Gets the data from the database
    $students = ModEnrollmentsReportHelper::getData();
}
```

Per consentire la chiamata di un metodo che utilizzi API Joomla!, non risulta sufficiente includere il file helper.php:

```
require_once "../helper.php";
```

Ma occorre importare le API Joomla!, in quanto il file in cui è contenuto tale classe (sheetManager.php) non rientra nella [struttura base di un modulo](#) (vedi paragrafo 2.2):

```
// Imports the Joomla library in order to use its API
// (JFactory object used in ModEnrollmentsReportHelper::getData())
define('_JEXEC', 1);
define('JPATH_BASE', realpath(dirname(__FILE__).'../../../../../'));
require_once(JPATH_BASE . '/includes/defines.php');
require_once(JPATH_BASE . '/includes/framework.php');
$mainframe = JFactory::getApplication('site');
```

Grazie a tale import, si evita di incorrere nell'errore "Class JFactory not found".

Il metodo chiamato è il seguente:

```
public static function getData() {
    $db = JFactory::getDbo();
    $db->setQuery("select * from g9f7i_student_form");
    return $db->loadObjectList();
}
```

Il vantaggio che ne deriva dall'utilizzo delle API *Joomla!* è visibile dalla prima riga di questo metodo: è possibile ottenere un oggetto (di tipo `jDatabase`) che rappresenta una connessione al database usato da *Joomla!* senza il bisogno di specificare nessun tipo di informazione.

Il metodo `getData()` quindi provvede a recuperare tutti i dati contenuti nella tabella interessata (`g9f7i_student_form`), e restituisce il risultato sotto forma di un array di result set.

Tale risultato viene quindi assegnato alla variabile `$students` nel metodo `loadDatabaseData()` della classe `SheetManager`.

```
function loadDatabaseData() {
    global $firstRow;

    // Gets the data from the database
    $students = ModEnrollmentsReportHelper::getData();
```

Viene selezionato quindi il primo foglio `all_applications`, nel quale inserire tutti i dati di tutti i candidati:

```
$allApplicationsWorksheet = $this->sheet->getSheetByName('all_applications');
```

Dopodiché viene eseguito un ciclo `for` nel quale si scorrono i result set appena ottenuti dal database:

```
for($i = 0; $i < count($students); ++$i) { ... }
```

Per ogni tupla del database (ognuna rappresentante un candidato), vengono effettuate molteplici chiamate al metodo `setCellValue()` reso disponibile dalla libreria *PHPExcel*, con le quali si inseriscono le informazioni nel primo foglio del file Excel. Di seguito, un esempio di chiamata di tale metodo che inserisce il cognome nella giusta cella del foglio di calcolo:

```
// Surname
$allApplicationsWorksheet->setCellValue(
    'F' . ($firstRow + $i + 1),
    $students[$i]->surname
);
```

Operazioni di questo tipo vengono effettuate un totale di 29 volte per ogni candidato.

L'ultima informazione detraibile dal database, ancora da inserire nel foglio Excel, riguarda le lingue straniere sconosciute. Il form delle ammissioni prevede un campo di testo apposito:

Foreign languages:

Il file Excel di template, utilizzato negli scorsi anni, prevede 7 gruppi di colonne dove poter inserire, per ogni lingua, la certificazione e altre informazioni deducibili dalla visione degli allegati.

Una parte del metodo `loadDatabaseData()` si occupa di inserire le lingue straniere conosciute nel file Excel prelevandole dal database, in cui sono salvate come un'unica stringa di testo.

Per fare ciò, occorre dividere la stringa di testo ottenendo un array di lingue conosciute (al più 7): ciò è svolto dalla funzione `preg_split()` che divide stringhe utilizzando espressioni regolari:

```
// Languages
/*  Creates an array of languages (taken from the "Foreign languages"
    text field filled by the enrolling student) by separating them
    with a regular expression.
        \s      =    white space
        +      =    one or more
*/
$languages = preg_split('/(\s|,|-)+/', $students[$i]->foreign_lang, 7);
```

L'unica altra informazione reperibile dal form è il punteggio ricevuto dalla certificazione di conoscenza della lingua inglese:

Please specify your  
English Certificate  
Score, if any:

Perciò, il metodo provvede a inserire anche questa informazione nel gruppo di colonne relativo alla lingua inglese. Come primo passo, vengono rimosse, dall'array di lingue `$languages`, tutte le occorrenze di "english" e/o "british", e viene inserita la lingua inglese come prima lingua, in qualsiasi caso.

```
/* Creates an array of all the occurrences of words beginning
   with 'eng' or 'bri', case insensitive. */
$keys = preg_grep('/^(eng|bri)/i', $languages);
// Set the internal pointer of an array to its first element
reset($keys);
// Removes all those occurrences from the languages array
do {
    unset($languages[key($keys)]);
} while(next($keys));
// Insert "English" as the first language
array_unshift($languages, "English");
```

In questo modo, nel caso in cui il candidato abbia inserito un livello di conoscenza di inglese, verrà visualizzato nel foglio Excel nel primo gruppo di colonne relativo alle lingue straniere conosciute (quello relativo alla lingua inglese); altrimenti, tale cella sarà vuota. Ciò viene svolto in:

```
// First language column
$langColumn = 57;
$allApplicationsWorksheet->setCellValueByColumnAndRow(
    $langColumn + 1,
    $firstRow + $i + 1,
    $students[$i]->english_score
);
```

Vengono poi inserite le lingue rimanenti:

```
foreach($languages as $language) {
    $allApplicationsWorksheet->setCellValueByColumnAndRow(
        $langColumn,
        $firstRow + $i + 1,
        $language
    );
    $langColumn += 6;
}
```

Di seguito, il risultato con alcuni candidati di prova (sono mostrate solo le colonne rilevanti per l'esempio):

Surname	First Name and Middle Name (if any)	language	written	language	language	language	language	language	language
White	Jack	English		French	Spanish				
Fernandez	Jenny	English	C2						
Borland	Wes	English	Motherton	Italian	French	Spanish	German		
Wellington	John	English	C2	Farsi	French				
Stevens	Michael	English	A1	Spanish					

Ottenuto prelevando le seguenti informazioni di prova dal database:

```
SELECT `surname`, `firstname`, `foreign_lang` FROM `g9f7i_student_form`
```

surname	firstname	foreign_lang
White	Jack	French, Spanish
Fernandez	Jenny	
Borland	Wes	Italian, French, Spanish, German
Wellington	John	English Farsi French
Stevens	Michael	Spanish english

Il costruttore della classe SheetManager termina la sua esecuzione salvando le modifiche effettuate al foglio di calcolo:

```
// Saves Excel File
global $fileType;
global $sheetFileName;
$sheetWriter = PHPExcel_IOFactory::createWriter($this->sheet, $fileType);
$sheetWriter->save($sheetFileName);
```

L'esecuzione torna allo script download.php, che provvede a creare l'archivio zip e ad aggiungervi il file Excel appena creato:

```
$zip = new ZipManager();
// Adds the Excel file to the archive
$zip->addFile($sheetFileName);
```

ZipManager è una classe creata per la gestione dell'archivio zip.

Similmente alla classe SheetManager, possiede un unico membro privato che rappresenta l'archivio:

```
class ZipManager {
    private $zip = null;
```

Inizializzato anch'esso dal costruttore:

```
function __construct() {
    global $zipFileName;

    $this->zip = new ZipArchive();

    // Creates the zip file
    if(!$this->zip->open($zipFileName, ZipArchive::CREATE))
        exit("Cannot create zip file.\n");
}
```

Lo script `download.php` aggiunge inoltre tutti gli allegati tramite la seguente istruzione:

```
$zip->addDirectory($attachmentsDirectory);
```

Il metodo è di seguito riportato:

```
public function addDirectory($directory) {
    global $attachmentsFolderName;

    // Returns the canonicalized absolute pathname
    $rootPath = realpath($directory);

    // Iterates through recursive iterators
    // (all node's children are a RecursiveIterator)
    $files = new RecursiveIteratorIterator(
        // Iterator that traverses through subfolders
        new RecursiveDirectoryIterator($rootPath)
    );

    foreach($files as $file) {
        // Skips directories (they would be added automatically)
        if(!$file->isDir()) {
            // Real source file path
            $filePath = $file->getRealPath();
            // Relative destination file path
            $relativePath = substr($filePath, strlen($rootPath) + 1);

            // Adds current file to archive
            $this->zip->addFile(
                $filePath,
                $attachmentsFolderName.'/'.$relativePath
            );
        }
    }
}
```

Tale metodo si basa sull'utilizzo dell'iteratore `RecursiveIteratorIterator`

(<http://php.net/manual/en/class.recursiveiteratoriterator.php>): PHP considera ogni nodo di un file system come con `RecursiveDirectoryIterator`: `RecursiveIteratorIterator` consente di iterare tre le sotto cartelle di `modules/mod_student_form/uploads`, directory (relativa alla directory root del sito) contenente tutti i PDF allegati.

L'iteratore considera tutti i nodi, ovvero anche le directory: saltando quest'ultime, è possibile mantenere la gerarchia delle directory durante l'inserimento dei file nell'archivio, indicando come secondo parametro della funzione `addFile()` il nome locale del file all'interno dell'archivio, comprensivo di percorso relativo.

Lo script `download.php` conclude la creazione dell'archivio con un semplice salvataggio:

```
$zip->saveZip();
```

Che consiste nella chiamata del metodo `close()` dell'oggetto `ZipArchive` in uso, condizione necessaria per creare l'archivio su file system, secondo la documentazione PHP:

```
public function saveZip() {
    $this->zip->close();
}
```

Viene infine eliminato il file Excel appena aggiunto all'archivio:

```
// Deletes the Excel file outside the zip archive
$sheet->deleteSheet($sheetFileName);
```

Lo script `download.php` termina restituendo l'indirizzo dell'archivio appena creato, relativo all'URL corrente del browser.

```
// Returns the zip file's directory
echo "../../modules/mod_enrollments_report/util/".$zipFileName;
```

Tale stringa viene passata alla funzione anonima contenuta nell'impostazione `success` dell'unico parametro del metodo `ajax()` dell'oggetto jQuery eseguito all'evento clic del link del download delle candidature.

```
success: function(data) {
    // Downloads the zip file
    window.location.href = data;

    /* Deletes the generated zip archive
       after the user downloads it */
    setTimeout(function(){
        jQuery.ajax({
            url: utilDirectory + "deleteZipFile.php"
        });
    }, deleteAfter);
}
```

La prima istruzione della funzione di successo provvede a far scaricare al browser l'archivio, impostando come destinazione l'URL del file zip, tramite la seguente istruzione:

```
// Downloads the zip file
window.location.href = data;
```

Provvede poi ad eliminare l'archivio dal file system del web server 10 secondi dopo l'inizio del download

```
// Zip archive deleting delay
var deleteAfter = 10 * 1000;
```

utilizzando il metodo `setTimeout()` che esegue la funzione passata come primo parametro dopo un numero di millisecondi specificato come secondo parametro:

```
/* Deletes the generated zip archive
   after the user downloads it */
setTimeout(function(){
    jQuery.ajax({
        url: utilDirectory + "deleteZipFile.php"
    });
}, deleteAfter);
```

Il compito dell'eliminazione dell'archivio dal file system del web server viene incaricato allo script `deleteZipFile.php` nella sotto cartella `util`:

```
<?php
    /* Deletes the generated zip archive
       after the user downloads it. */

    require_once "../config/zipConfig.php";
    require_once "zipManager.php";
    require_once "download.php";

    $zip->deleteZip($zipFileName); // Deletes the zip archive
```

che utilizza il metodo `deleteZip()` della classe `ZipManager`:



```

public function deleteZip($file) {
    if(!unlink($file))
        exit("Cannot delete zip archive.\n");
}

```

### 2.4.3 Eliminazione dei dati riguardanti le richieste di ammissione

Tale funzionalità può essere utile dopo aver svolto la selezione dei candidati.

Grazie allo stesso meccanismo, jQuery cattura l'evento del clic sul relativo link:

```

<script type="text/javascript" charset="utf-8">
    $(
        $("#delete").click(function() {
            jQuery.ajax({
                url: utilDirectory + "deletePreviousAdmissions.php"
            });
        });
    );
</script>

```

Effettua poi una richiesta Ajax allo script `deletePreviousAdmissions.php`, che provvede a chiamare il relativo metodo della classe helper indicata dalla guida *Joomla!*:

```

// Empties the student form table
ModEnrollmentsReportHelper::deleteData();

```

Tale metodo si limita ad eliminare il contenuto della tabella contenente i dati delle candidature:

```

public static function deleteData() {
    $db = JFactory::getDbo();
    $db->setQuery("truncate table g9f7i_student_form");
    $db->execute();
}

```

Come nel metodo `loadDatabaseData()` della classe `SheetManager`, occorre importare le API *Joomla!* per consentire ad un file non appartenente alla [struttura base di un modulo](#) di effettuare chiamate a metodi che utilizzano classi appartenenti alle API *Joomla!*.

```

require_once "../helper.php";

// Imports the Joomla library in order to use its API
// (JFactory object used in ModEnrollmentsReportHelper::deleteData())
define('_JEXEC', 1);
define('JPATH_BASE', realpath(dirname(__FILE__).'../../../../../'));
require_once(JPATH_BASE . '/includes/defines.php');
require_once(JPATH_BASE . '/includes/framework.php');
$mainframe = JFactory::getApplication('site');

```

Successivamente lo script provvede ad eliminare gli allegati delle candidature dal file system del web server:

```

$dir = '../mod_student_form/uploads/';
$it = new RecursiveDirectoryIterator($dir, RecursiveDirectoryIterator::SKIP_DOTS);
$files = new RecursiveIteratorIterator($it, RecursiveIteratorIterator::CHILD_FIRST);

foreach($files as $file) {
    if($file->isDir())
        rmdir($file->getRealPath());
    else
        unlink($file->getRealPath());
}

```

Anche in questo caso, si ricorre all'utilizzo dell'iteratore `RecursiveDirectoryIterator` che, come già spiegato nel [paragrafo 2.4.2](#), itera tra i nodi di un file system, che PHP rappresenta tramite iteratori `RecursiveDirectoryIterator`.

La costante `RecursiveDirectoryIterator::SKIP_DOTS` passata come secondo parametro al costruttore di `RecursiveDirectoryIterator` indica di saltare i file `.` e `..` delle directory, mentre la costante `RecursiveDirectoryIterator::CHILD_FIRST` passata come secondo parametro al costruttore di `RecursiveDirectoryIterator`, indica all'iteratore di cambiare l'ordine di attraversamento, visitando prima i figli, ovvero le sotto cartelle e il loro contenuto.

Occorre controllare se la variabile `$file` sia una directory o no in quanto la funzione per eliminare una directory è diversa dalla funzione per eliminare un file.

## 2.5 Modifica al modulo del form di ammissione

Il modulo del form di ammissione `mod_student_form` prevedeva un ulteriore salvataggio delle informazioni, già presenti su database, sul file system del web server, sotto forma di un semplice file txt. È stata quindi commentata la porzione di codice sotto riportata in `mod_student_form.php` (dalla riga 213 alla riga 217 incluse) per evitare tale comportamento: ciò diminuisce la ridondanza di informazioni sul file system del web server e soprattutto snellisce il contenuto dell'archivio scaricabile tramite il modulo di reporting delle candidature.

```
foreach ($a as &$value) {  
    file_put_contents('modules/mod_student_form/uploads/forms_content.txt', $value."\t" ,  
FILE_APPEND);  
}
```

```
file_put_contents('modules/mod_student_form/uploads/forms_content.txt', "\r\n" ,  
FILE_APPEND);
```

## 3. Installazione e integrazione del modulo nel sito web

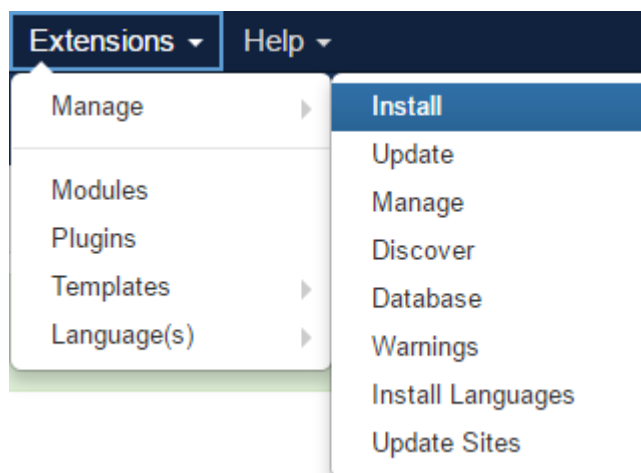
Di seguito viene riportata una piccola guida che indica, passo dopo passo, le operazioni da eseguire per installare il modulo in *Joomla!* e per integrarlo correttamente nel sito web dei corsi.

### 3.1 Installazione

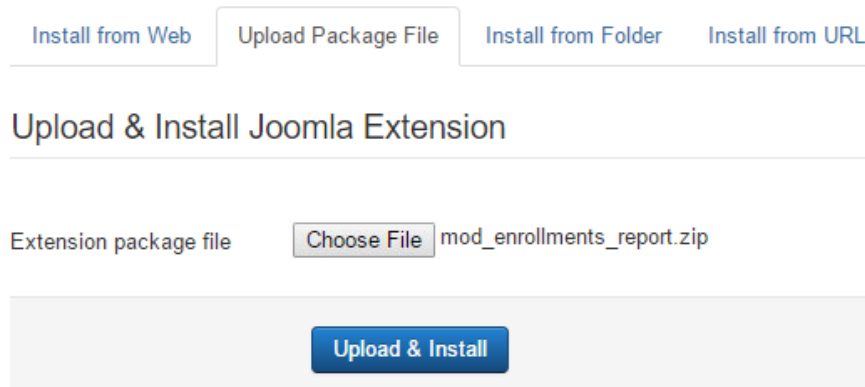
1. Effettuare l'accesso dal pannello amministrativo di *Joomla!* da <http://ce.iet.unipi.it/administrator/>



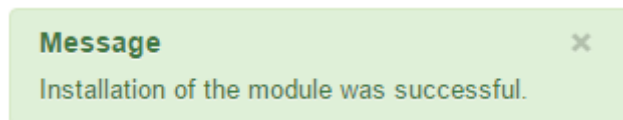
2. Installare il modulo cliccando, sulla barra dei menu posta in alto, *Extensions -> Manage -> Install*.



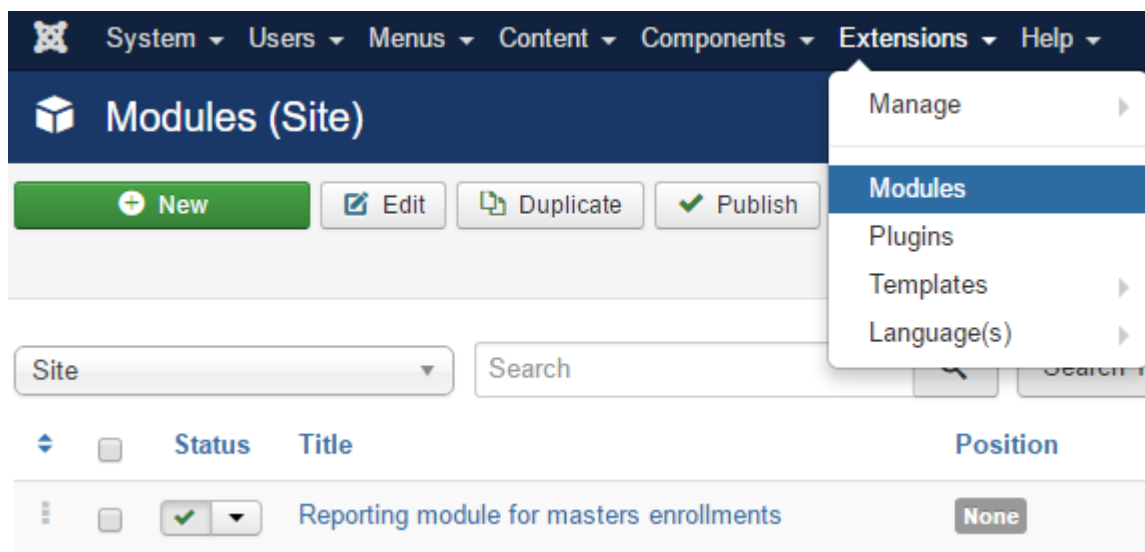
3. Selezionare la scheda *Upload Package File*, cliccare sul pulsante *Choose File* e selezionare il file *mod\_enrollments\_report.zip* contenente il modulo. Fare quindi clic sul pulsante *Upload & Install*.



Attendere finché *Joomla!* abbia completato l'installazione, mostrando il seguente messaggio di conferma.



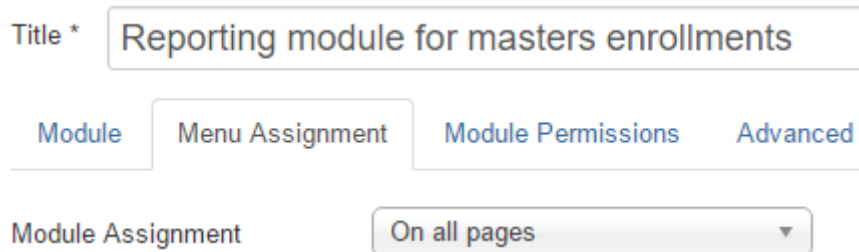
4. Aggiungere manualmente il file *sheet\_template.xlsx* reperibile dalla directory principale del file zip appena caricato (*mod\_enrollments\_report.zip*) sul web server, nella sub directory *modules/mod\_enrollments\_report* del sito, tramite un qualsiasi client FTP. Durante l'installazione di questo modulo, il *Joomla! installer* trasferisce solo i file con alcune estensioni, probabilmente per questioni di sicurezza. Tale file mancante è il template del foglio Excel che conterrà tutti i dati dei candidati.
5. Dalla barra dei menu posta in alto, selezionare *Extensions -> Modules*, e selezionare il modulo denominato "*Reporting module for masters enrollments*".



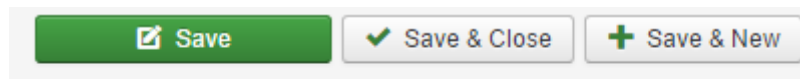
6. Nella scheda principale *Module*, assicurarsi che lo *Status* sia impostato su *Published*:



7. Selezionare la scheda *Menu Assignment* e impostare il campo *Module Assignment* su *On all pages*.



8. Cliccare sul pulsante *Save & Close* posto in alto.

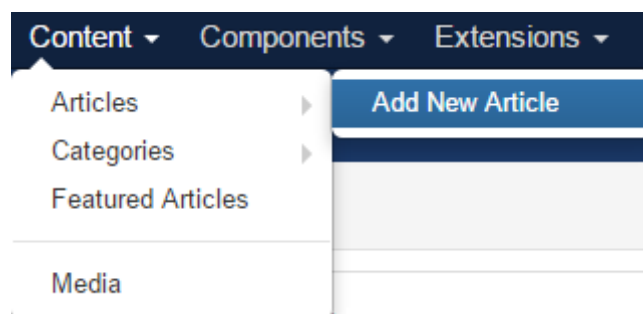


Title \* Reporting module for masters enrollments

### 3.2 Integrazione nel sito web

Una volta installato il modulo, è necessario integrarlo nel sito web per essere in grado di interagire con quest'ultimo. Il modulo è raggiungibile tramite una voce di menu, visibile solo ad utenti loggati sul sito con User Group appartenente all'Access Level *Special*, nella versione del sito in lingua inglese.

1. Come primo passo perciò occorre creare l'articolo contenente il modulo.  
Per fare ciò, selezionare dalla barra dei menu posta in alto nel pannello amministrativo *Content* -> *Articles* -> *Add New Article*.

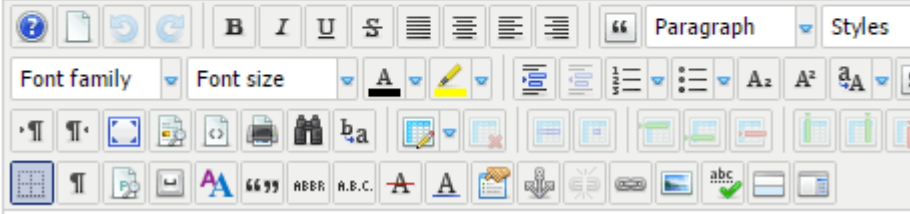


- Impostare il titolo dell'articolo in "Download admissions" e inserire la direttiva di caricamento {loadmodule mod\_enrollments\_report} del modulo nel contenuto dell'articolo:

Title \*

Content Images and links Options Publishing Associations

[Toggle Editor]



{loadmodule mod\_enrollments\_report}

- Impostare il campo Access su Special e il campo Language su English (UK):

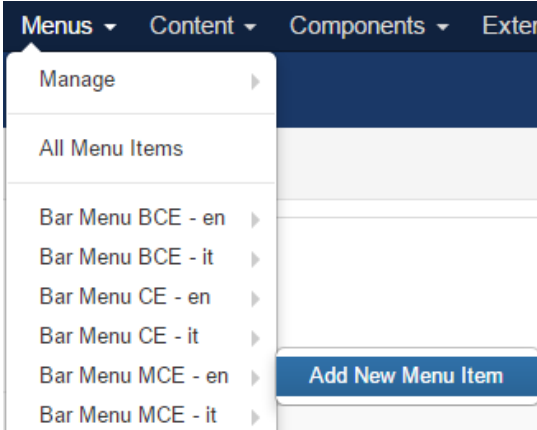
Access

Language

- Salvare l'articolo cliccando sul pulsante Save & Close posto in alto:

Title \*

- Occorre ora aggiungere la voce di menu grazie al quale è possibile raggiungere l'articolo appena creato. Per fare ciò, selezionare dalla barra dei menu posta in alto Menus -> Bar Menu MCE - en -> Add New Item Menu:



Menus -> Content -> Components -> Extension

- Manage
- All Menu Items
- Bar Menu BCE - en
- Bar Menu BCE - it
- Bar Menu CE - en
- Bar Menu CE - it
- Bar Menu MCE - en **Add New Menu Item**
- Bar Menu MCE - it

6. Impostare il titolo del menu in “*Download admissions*”:

Menu Title \*

7. Impostare il campo *Parent Item* su *Admission*, il campo *Access* su *Special* e il campo *Language* su *English (UK)*:

Parent Item

Access

Language

8. Fare clic sul pulsante *Select* del campo *Menu Item Type*

Menu Item Type \*

E selezionare *Articles* -> *Single Article*:

## Menu Item Type

### Articles

<b>Archived Articles</b>	Display all archived articles.
<b>Category Blog</b>	Displays article introductions in a single or multi-column layout.
<b>Category List</b>	Displays a list of articles in a category.
<b>Create Article</b>	Create a new article.
<b>Featured Articles</b>	Show all featured articles from one or multiple categories in a single or multi-column layout.
<b>List All Categories</b>	Shows a list of all the article categories within a category.
<b>Single Article</b>	Display a single article.

9. Comparirà ora, sotto il campo *Menu Item Type*, un altro campo denominato *Select Article*. Fare clic sul pulsante *Select*:

Select Article \*

Select an Article

Select

E selezionare l'articolo "*Download admissions*" creato precedentemente:

### Select or Change article

Search	Q	Search Tools ▾	Clear	ID descending ▾	20 ▾
Status	Title	Access	Language	Date	ID ▾
✓	Download admissions Category: Uncategorized	Public	🇬🇧 English (UK)	2016-12-11	182

Ordinare la lista per ID decrescente come in figura, per mostrare l'articolo appena creato in cima alla lista.

10. Salvare facendo clic sul pulsante *Save & Close* posto in alto.

Save

Save & Close

Menu Title \*

Download admissions

11. Ripetere i punti dal 5 al 10 per il menu *Bar Menu MECS – en*.



## 4. Manuale d'uso

### 4.1 Apertura e chiusura delle iscrizioni

Questa prima metà del manuale d'uso è rivolta al coordinamento didattico, e fornisce le minime operazioni fondamentali per aprire e chiudere le iscrizioni sul sito ufficiale dei corsi.

#### 4.1.1 Apertura delle iscrizioni

1. Effettuare l'accesso dal pannello amministrativo di Joomla! da <http://ce.iet.unipi.it/administrator/>



2. Occorre modificare i due articoli relativi alle ammissioni relative ai CdL Magistrale in Computer Engineering e in Embedded Computing Systems, raggiungibili tramite i due seguenti link:

- <http://ce.iet.unipi.it/index.php/en/admission-mce/application>
- <http://ce.iet.unipi.it/index.php/en/admission/requirements-2>

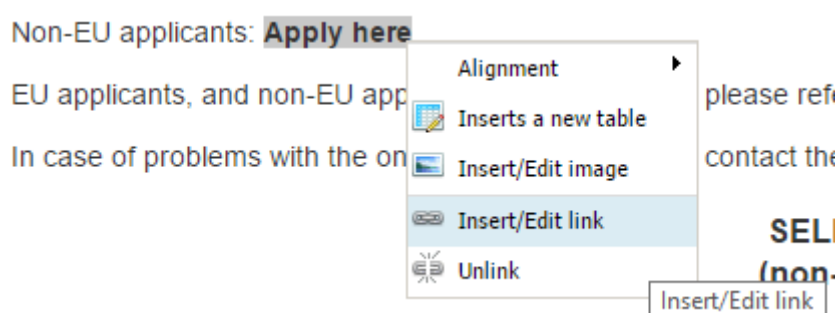
Per fare ciò, selezionare sulla barra dei menu posta in alto *Content* -> *Articles*



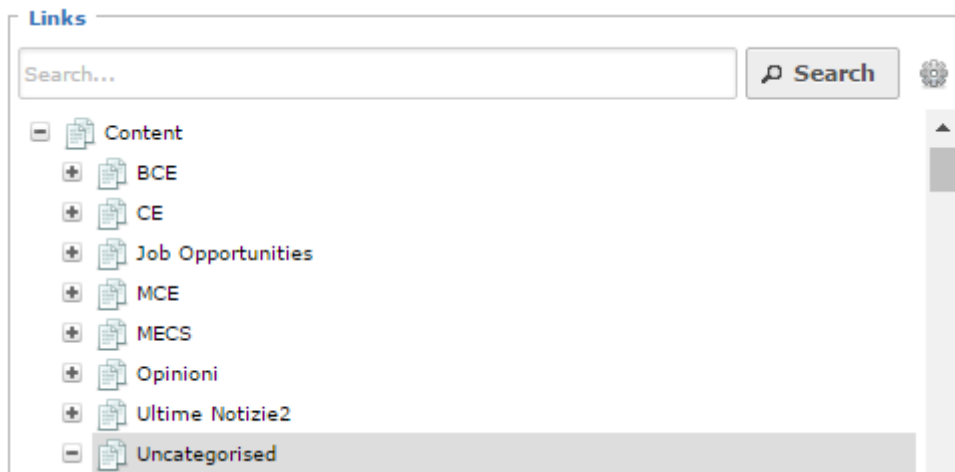
- Fare clic sull'articolo con ID 175: è possibile filtrare i risultati cercando la parola "registrazione" o "application", in modo da facilitarne la ricerca.

Status	Title	Access	Association	Author	Language	Date	Hits	ID
<input type="checkbox"/>	{lang it}Registrazione (a.a. 2016/2017){/lang} {lang en}Application (a.y. 2016/2017){/lang} (Alias: mce-application-2016) Category: MCE	Public		Super User	All	2016-04-29	593	175

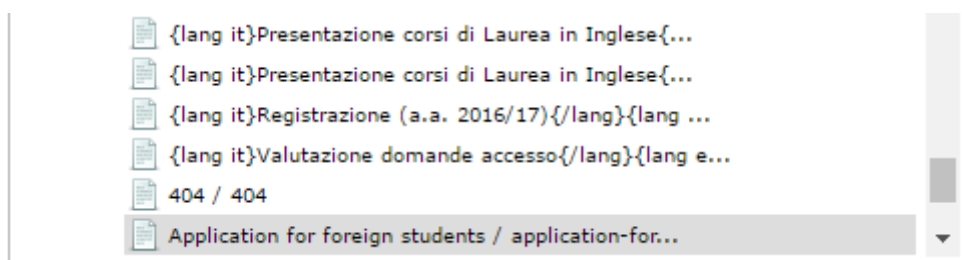
- Modificare la riga "Non-EU applicants: **Application closed**" in "Non-EU applicants: [Apply here](#)", rendendo "[Apply here](#)" un collegamento verso l'articolo contenente il form di ammissione. Per fare ciò, selezionare con il mouse la scritta "[Apply here](#)", fare clic con il tasto destro e selezionare *Insert/Edit link*



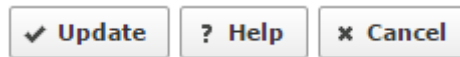
- Nel riquadro *Links*, selezionare *Content* -> *Uncategorized*



E successivamente selezionare l'articolo *Application for foreign students*:



Fare clic sul tasto *Update* in basso a destra per aggiornare il link:



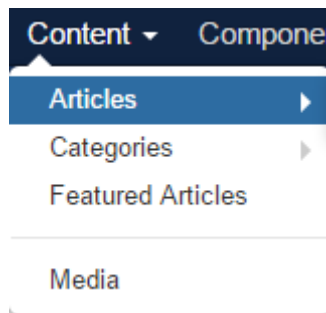
6. Fare clic su *Save & Close* in alto a sinistra per completare la modifica dell'articolo:



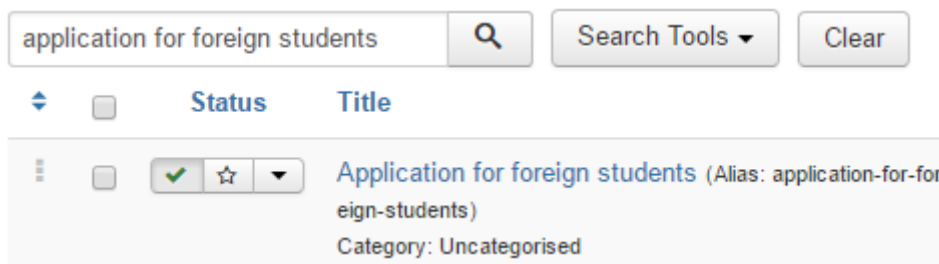
Title \*

7. Ripetere i punti dal 2 al 6 compresi, per l'articolo di ID 172, relativo alle ammissioni per il CdL Magistrale in Embedded Computing Systems.

8. Occorre ora rendere visibile l'articolo contenente il form di iscrizione. Per fare ciò, selezionare *Content -> Articles* dalla barra dei menu posta in alto:



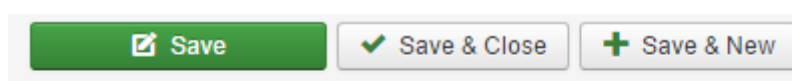
9. Fare clic sull'articolo *Application for foreign students*, raggiungibile cercando semplicemente il titolo nella barra di ricerca:



10. Impostare lo *Status* su *Published*:



11. Fare clic su *Save & Close* posto in alto a sinistra:



Title \*

#### 4.1.2 Chiusura delle iscrizioni

1. Effettuare l'accesso dal pannello amministrativo di Joomla! da <http://ce.iet.unipi.it/administrator/>



2. Occorre modificare i due articoli relativi alle ammissioni relative ai CdL Magistrale in Computer Engineering e in Embedded Computing Systems, raggiungibili tramite i due seguenti link:

- <http://ce.iet.unipi.it/index.php/en/admission-mce/application>
- <http://ce.iet.unipi.it/index.php/en/admission/requirements-2>

Per fare ciò, selezionare sulla barra dei menu posta in alto *Content* -> *Articles*:

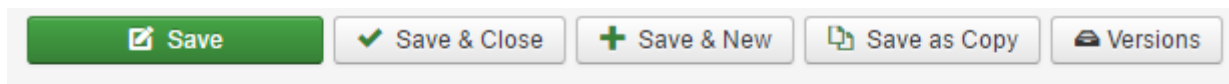


3. Fare clic sull'articolo con ID 175: è possibile filtrare i risultati cercando la parola "registrazione" o "application", in modo da facilitarne la ricerca.

Status	Title	Access	Association	Author	Language	Date	Hits	ID
<input checked="" type="checkbox"/>	{lang it}Registrazione (a.a. 2016/2017){/lang} {lang en}Application (a.y. 2016/2017){/lang} (Alias: mce-application-2016) Category: MCE	Public		Super User	All	2016-04-29	593	175

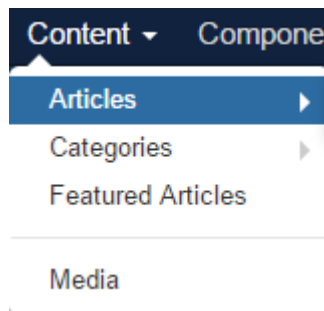
4. Modificare la riga "Non-EU applicants: [Apply here](#)" in "Non-EU applicants: **Application closed**". La scritta "**Application closed**" non deve più essere un link, in modo da non rendere più raggiungibile l'articolo contenente il form delle iscrizioni.

5. Fare clic su *Save & Close* in alto a sinistra per completare la modifica dell'articolo:

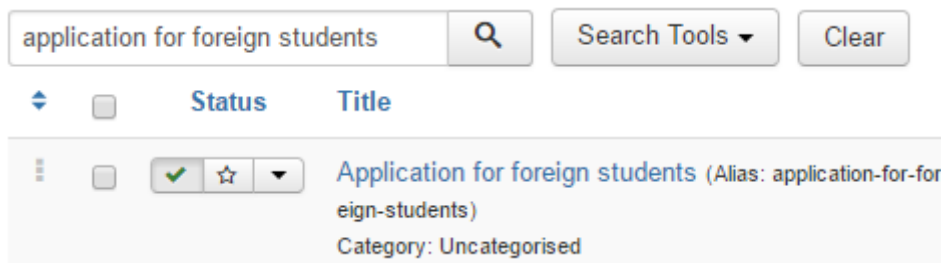


Title \*

6. L'articolo contenente il form delle iscrizioni potrebbe essere tuttavia rimanere accessibile da qualsiasi utente in possesso del link dell'articolo (ad esempio, uno studente che abbia salvato tale pagina tra i preferiti del browser). Per impedire che ciò accada, occorre cambiare lo stato di tale articolo. Per fare ciò, selezionare sulla barra dei menu posta in alto *Content -> Articles*:



7. Fare clic sull'articolo *Application for foreign students*, raggiungibile cercando semplicemente il titolo nella barra di ricerca:




	Status	Title
<input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Application for foreign students (Alias: application-for-for- eign-students) Category: Uncategorized

8. Impostare lo *Status* su *Unpublished*:

Status

9. Fare clic su *Save & Close* posto in alto a sinistra:



Title \*

## 4.2 Gestione delle candidature

Questa seconda metà del manuale d'uso è rivolta al personale incaricato alla selezione dei candidati.

### 4.2.1 Download delle candidature e selezione dei candidati

1. Effettuare il login tramite il modulo di login *Area riservata* presente in basso a sinistra su <http://localhost/ce-iet-unipi-it/index.php/it/> con delle credenziali relative ad un account con User Group appartenente all'Access Level *Special*:

**AREA RISERVATA**

Nome utente

Password

Accedi

› Nome utente dimenticato?

› Password dimenticata?

Gli User Group appartenenti a tale *Access Level* possono essere visualizzati e modificati selezionando, partendo dalla barra dei menu presente in alto, *Users -> Access Levels*

Users ▾ Menus ▾

Manage ▶

Groups ▶

Access Levels ▶

Search [ ] [Q] [Clear]

<input type="checkbox"/>	Level Name	User Groups Having Viewing Access
<input type="checkbox"/>	Public	Public
<input type="checkbox"/>	Registered	Manager, Registered, Super Users
<input type="checkbox"/>	Special	Author, Manager, Super Users

Da qui è possibile notare che al corrente i seguenti User Group ne fanno parte: *Author, Manager, Super User*.

2. Accertarsi di aver effettuato il login con successo:

**AREA RISERVATA**

Ciao Super User,

Esci

3. Selezionare la lingua inglese grazie al selettore di lingua posto in alto a sinistra:



### MAIN MENU

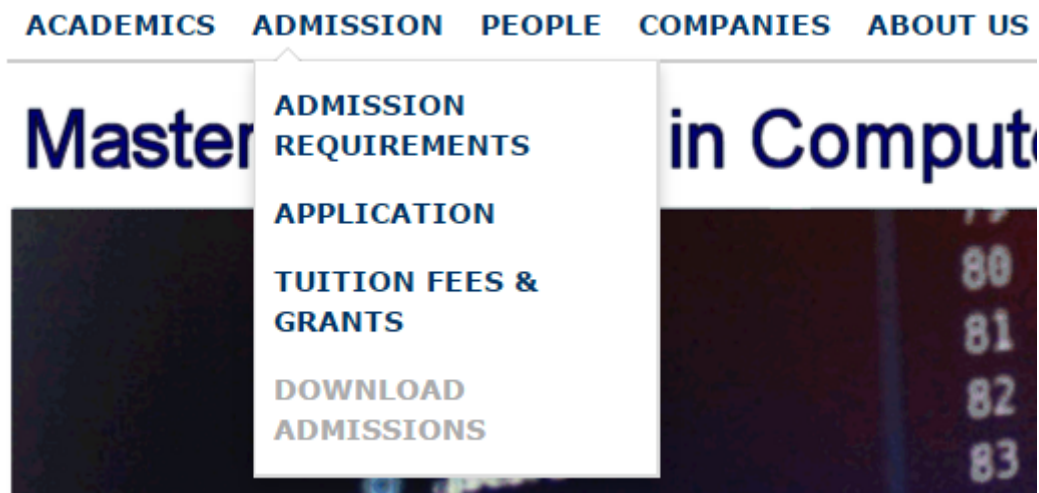
- » Home CE
- » UNIPI
- » DIPARTIMENTO

4. Selezionare il CdL Magistrale desiderato dall'area *Degree courses*, posto a sinistra:

### DEGREE COURSES

- » Bachelor of Science in Computer Engineering
- » Master of Science in Computer Engineering
- » Master of Science in Embedded Computing Systems

5. Dalla barra di navigazione presente in alto, selezionare *Admission* -> *Download admissions*:

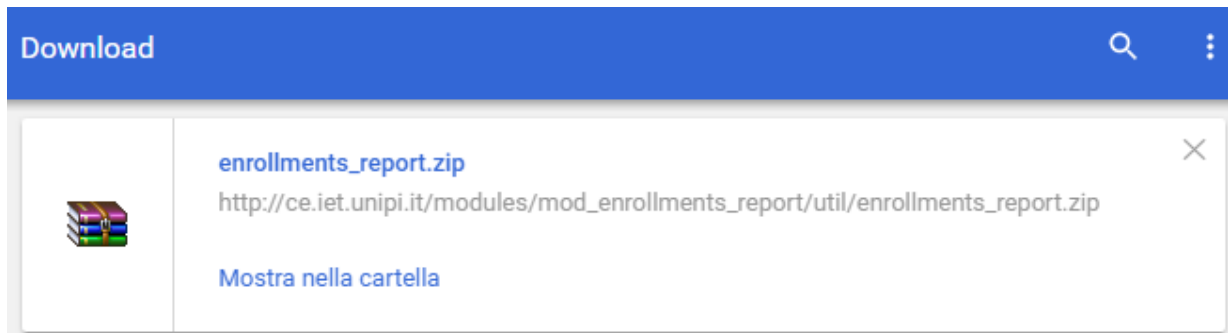


6. Per scaricare l'archivio contenente i PDF allegati dai candidati, e il file Excel di selezione, fare clic su *Download admissions*:

## Enrollments report module

- > Download admissions
- > Delete previous admissions

- Aspettare un paio di secondi affinché il browser scarichi l'archivio generato automaticamente (il browser mostrato per esempio in figura è Google Chrome):



- Una volta scompattato l'archivio, si ottiene la cartella contenente gli allegati attachments e il file Excel di selezione Reporting sheet.xlsx:

Nome	Tipo
attachments	Cartella di file
Reporting sheet.xlsx	Foglio di lavoro di Microsoft Excel

Gli allegati sono divisi per cartelle, una per ogni studente, seguendo il formato *Cognome\_Nome*, nello stesso modo in cui vengono salvati dal modulo del form delle ammissioni. Di seguito, un esempio di allegati alcuni candidati di prova:

Nome	Tipo
Borland_Wes	Cartella di file
Fernandez_Jenny	Cartella di file
Stevens_Michael	Cartella di file
Wellington_John	Cartella di file
White_Jack	Cartella di file

- Per procedere con la selezione dei candidati, aprire il foglio di calcolo Reporting sheet.xlsx: il primo foglio visualizzato, all\_applications, contiene tutte le informazioni di tutti i candidati che hanno compilato il form di iscrizione.

Tramite la colonna A è possibile ammettere dei candidati scrivendo TRUE nella riga relativa al candidato da ammettere:

	valid	risultato check	note	Fiscal Code (to be used for administrative reasons, see below for fiscal code generator)	Surname	First Name and Middle Name (if any)	Gender	Date of birth
2								
3	FALSE				White	Jack	M	1/5/1992
4	TRUE				Fernandez	Jenny	F	5/12/1992
5	FALSE				Borland	Wes	M	3/3/1984
6	TRUE				Wellington	John	M	13/4/1989
7	TRUE				Stevens	Michael	M	17/4/1990



I candidati selezionati verranno copiati automaticamente sul secondo foglio, denominato **valid\_applications**:

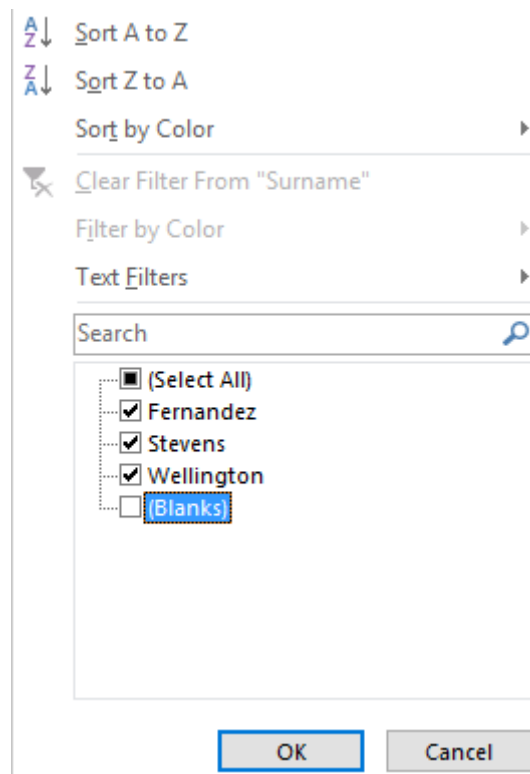
	note	Surname	First Name and Middle Name (if any)	Gender
2				
3				
4		Fernandez	Jenny	F
5				
6		Wellington	John	M
7		Stevens	Michael	M
8				
9				

all\_applications **valid\_applications** legenda (+)

Per mostrare i candidati ammessi in ordine alfabetico crescente, senza righe vuote intermedie, occorre fare clic sull'icona del filtro dei cognomi:



E assicurarsi che siano selezionati tutti i cognomi, tranne le stringhe vuote, individuate dall'elemento (Blanks) della lista:



Una volta applicato il filtro, occorre rieseguire questa operazione di filtraggio nel caso in cui si decida di ammettere altri candidati nel frattempo.

10. La legenda è consultabile nel terzo foglio, denominato **legenda**:

	B	C	D	E
3	<b>LEGENDA STATUS</b>			
4	0	AMMESSO		
5	1	AMMESSO SUB CONDICIO - TITOLO DA CONSEGUIRE		
6	2	AMMESSO - DA VALUTARE VALUTAZIONE DA PARTE DELLA COMMISSIONE L'AMMISSIBILITA' DEL TITOLO		
7	3	AMMESSO CON DOCUMENTAZIONE DIFFORME - DA VALUTARE VALUTAZIONE DA PARTE DELLA COMMISSIONE L'AMMISSIBILITA'		
8	4	NON AMMESSO - DOCUMENTAZIONE CARENTE		
9	5	NON AMMESSO - APPLICATION DOPPIA		
10	6	NON AMMESSO - CONSEGUIMENTO TITOLO OLTRE I TERMINI		
11				
12				
13	<b>ranking università scimago</b>		<b>categoria</b>	<b>pesi</b>
14	Ranking World Scimago (Normalised Impact+Only Higher Education) 1-1000		1	1
15	Ranking World Scimago (Normalised Impact+Only Higher Education) 1001-1500		2	0.9
16	Ranking World Scimago (Normalised Impact+Only Higher Education) 1501-2000		3	0.8
17	Ranking World Scimago (Normalised Impact+Only Higher Education) 2001-		4	0.7
18	European (geographic) + Turkish		5	1
19	non presenti		6	0.5

Si ricorda che è possibile modificare la struttura del file Excel modificando il template (`sheet_template.xlsx`) disponibile nella root del modulo nel web server. La sub directory del modulo è `modules/mod_enrollments_report`.

#### 4.2.2 Eliminazione delle candidature precedenti

1. Seguire i passi da 1 a 5 compresi del [paragrafo relativo al Download delle candidature](#).
2. Per eliminare le candidature precedenti, fare clic su *Delete previous admissions*:

## Enrollments report module

- > [Download admissions](#)
- > [Delete previous admissions](#)

La tabella interessata nel database verrà svuotata, e verranno eliminati tutti gli allegati presenti su file system del web server (in `modules/mod_student_form/uploads`).

## 5. Conclusioni e possibili miglioramenti futuri

### 5.1 Conclusioni

Gran parte della realizzazione di questo progetto non ha subito grossi intoppi: la documentazione PHP è molto ricca e dettagliata, lo stesso dicasi per la libreria PHPEXcel.

Ciò che ha richiesto maggior tempo è stata la creazione del file Excel che riportasse automaticamente nel secondo foglio i soli candidati selezionati nel primo.

La prima soluzione da me sviluppata consisteva in uno script VBA (Visual Basic for Applications), piuttosto lungo, associato all'evento clic su delle checkbox presenti al posto dei valori TRUE o FALSE attualmente presenti.

Tale soluzione non era scalabile in quanto Excel, non associando i controlli modulo (come le checkbox) alle celle, non permetteva di passare come parametro la riga alla quale la checkbox faceva riferimento

all'handler associato all'evento clic, costringendomi a replicare lo script tante volte quante erano le righe necessarie.

Una seconda soluzione prevedeva uno script VBA basato su un filtro avanzato: esso permette di copiare righe che rispettano certe condizioni in altre posizioni del file Excel.

I criteri per il filtro avanzato erano i seguenti:

```
all_applications!$A$2:$DK$230
all_applications!$EB$2:$EB$3
valid_applications!$B$2:$DK$230
```

Di seguito, lo script VBA realizzato:

```
Sub AdmittedFilter()
    Sheets("all_applications").Range("A2:DK230").AdvancedFilter _
        Action:=xlFilterCopy, _
        CriteriaRange:=Sheets("all_applications").Range("EB2:EB3"), _
        CopyToRange:=Sheets("valid_applications").Range("B2:DK230"), _
        Unique:=True
End Sub
```

Due sono i motivi che non hanno portato alla scelta di questa soluzione:

1. Il personale incaricato alla selezione dei candidati avrebbe dovuto inserire manualmente il filtro avanzato e lo script VBA nel file Excel in quanto la libreria PHPEXcel non supporta tali componenti.
2. Per effettuare la copia dei candidati ammessi, il personale avrebbe dovuto avviare la macro manualmente, in quanto PHPEXcel, non supportando i controlli modulo, vieta la creazione di file Excel contenenti pulsanti grazie ai quali è possibile avviare delle macro VBA.

La soluzione adottata utilizza quindi delle semplici formule IF che controllano il valore della rispettiva cella della colonna valid nel primo foglio all\_applications. Tale soluzione ha i seguenti difetti:

1. Il bisogno di applicare il filtro sui cognomi per riportare e aggiornare gli studenti ammessi nel secondo foglio, come mostrato nel manuale d'uso.
2. Il maggior tempo richiesto dalla libreria PHPEXcel per leggere un file con molte formule. Questo causa un tempo di attesa di circa 5 secondi prima che l'archivio venga scaricato.

## 5.2 Possibili miglioramenti futuri

Un aspetto migliorabile di questo modulo potrebbe consistere nell'inserimento del foglio fino ad ora utilizzato per le statistiche sui paesi di provenienza dei candidati.

Tale foglio non è stato aggiunto al file Excel di template in questo progetto in quanto la libreria PHPEXcel non è in grado di gestire grafici.