Texila International Journal of Academic Research
Volume 4, Issue 2, Dec 2017

Contractor Information Management System (CIMS) – Import Module

Article by Wais Meeran
MSc in Information Technology, Texila American University, Afghanistan
E-mail: wmeeran@texilaconnect.com, waismeeran@gmail.com

Abstract

X-organization is looking for a solution to enables the X-client to identify which contractors are being used in Afghanistan and for what purpose (contracts). The contractor should be defined as business / enterprise / organization / or individual that will commission or employ to perform a specific job for a specific period through a contract with the X-Organization. Furthermore, the application should provide details on where the contracts are as well as the value, and to indicate the performance of the contract / contractor per different clients.

Based on strong research, the Application has been developed in Drupal. After some time, the organization decided to add more features in the application to satisfy their clients for example bulk import, grouping specific fields and a map showing geographical information.

Previously X-organization clients were adding each contractor separately which was quite time consuming, but recently they installed a new module called "Feeds Import" to facilitate their client's data import process. This new module allows clients to import, a list of their contractor/contracts information just by importing from a CSV file.

By implementing the above features the X-organization faced another challenge. Each CSV file had to be imported several times to accomplish the bulk import process and we later explain in more details in the next sections of this paper why the clients had to import the same file several times?

Now, the organization is looking for a solution to eliminate the need for importing one content type multiple times. So, the purpose of this document is to design and develop a module through which the import process is made dynamic, flexible and user friendly.

Keywords: *Drupal, import module, Contractor Information Management System, Feeds Import, Field collection, bulk import.*

Introduction

Before we explain the needs of this module, we need to explain the Contractor Information Management System. The purpose of this purpose-built and designed system is to define a business / enterprise / organization / or individual that will commission or employ to perform a specific job for a specific period through a contract with the X-Organization. Furthermore, the application is providing details on where the contracts are, as well as the contracts monetary values, and to indicate and assess the performance of the contract / contractor per different clients.

The Contractor Information Management System is designed as an electronic, inexpensive, fast, transparent, dynamic, secure, reliable and multipurpose.

The current multipurpose CIMS main features are listed below:

- Satisfies all agencies requirements.
- Enables a client to query the details of contractors and contracts (client's agency contracts) from anywhere using an Internet connection.
- Designed to be very user friendly. Any registered user with a valid login code can access the CIMS from anywhere with an Internet connection. Nominated agency focal points will receive a login code and training on how to add Contractor and Contracts information.
- Can be only accessed through the agencies specific users.

DOI: 10.21522/TIJAR.2014.04.02.Art013
ISSN: 2520-3088

- Allows the user to identify what contracts have been undertaken, where, for what services/works and for how much value.
- Designed in a manner which allows the agencies to add multiple fields and some fields having multiple values.
- Registered users are able to check how many contracts a specific contractor has ongoing with the X-Organization (search results dependent upon user access level). The capacity of the contractor is therefore able to be assessed to determine whether the contract is likely to be fulfilled at a satisfactory standard (technically or administratively) and prevent over-extension of the contactors resources.
- Serves as the common platform for submitting requests for monitoring and reporting missions at risk.
- Enables organizations to be informed about total expenditures on contracts throughout Afghanistan and can drill down to identify each region and district, as well as by type of contract (works, services, goods, lease, etc.).
- Defines financial thresholds for each Contractor. The system is able to aggregate the total contracts awarded and when the threshold is surpassed, it is flagged. This will then lead to a collective review (and possibly audit) of the contractor/partner to increase collective due diligence on contractor/partner's engagement.
- Enables comments to be registered about contracts and contractors to ensure greater information sharing (read functionality dependent upon user access level).
- Enables users to see the performance indicator of a contractor (Satisfactory, Not Satisfactory, and Suspended) and, if necessary, can request more information from the relevant agency.
- The system has various levels of access to restrict stored data usage and processing:
- Level 1: Data Entry (X-Organization Staff Only -Access to specific agency data).
- Level 2: Supervisory (Approved Focal Points) – Access to data from all agencies, including performance and total contractor value.
- Level 3: Manager (Head of Agency or Designate – as approved by the Head of X-Organization) - Access to all of the above including specific contract values.
- Level 4: Administrator (X-Organization development unit) – Access to all data and administration of the system.

Research problem

Currently some content types need to be imported multiple times. For instance, if we want to import contractor then we must import it 5 times in different "feeds importer" (Contractor_import_1, Contractor_import_2, Contractor_import_3, Contractor_import_4, and finally contractor_import_5). And the reason behind this process is that location and field collection modules will create separate tables for each field collection in the database and it creates a link with content type through foreign key. So, feed importers module neither allows us to map both content types and field collection, nor to map multiple field collection in a single feed importer process. Due to this fact, we need to have multiple feed importers[1].

The purpose of this document is to design and develop a customized module for Contractor Information Management System which is developed based on Drupal. So, this module eliminates the need for importing one content type multiple times. The new customized module interface is designed to be dynamic, flexible and easy to scale. The new module will address most of the CIMS users concerns in regard to feeds import feature forcing the users to do repetitive imports of same data, static features usage and time consuming processes.

The current "Feeds import" module needs to be transformed to a modern, electronic, inexpensive, fast, transparent, dynamic, secure, reliable, multipurpose module for contractor information management system (CIMS). The current multipurpose module challenges are listed below:

The module should be design to be very user friendly.

[1] It is very important to know that first the content type should be imported and then we will be able to import its field collections (because field collection need the primary key of the content type in order to be imported successfully.)

- Allow administrator to select their bulk importers.
- Allow users to import the contractors in series.
- The module should work for all types of fields collection.

The administrator should be able to apply security policies.

- The module should be able to show the report messages once at the end of bulk import.
- The module should be accessed only through the agencies specific users.
- The module should be designed to allow the administrator only to add unlimited bulk importers.

Proposed solution

The following are two proposed solutions for this problem:

1. Customizing the existing feeds import module.
2. Developing a custom module in top of feeds import module.

First solution will work, but it is not recommended by Drupal community since it is called hacking of module. By hacking the module, the X-organization will face problems in future so we cannot go with this solution.

Second solution will be recommended if no solution on the marketplace is available today that can meet at least 80% of our requirements. As we explored the marketplace, we couldn't find any existing solution for solving this problem, hence we will have to go with this second solution. The solution which is proposed for this bulk importing process, is to develop a custom module in top of feeds import module to eliminate the need for importing one content type multiple times.

The custom module will trigger whenever any content type has been imported through feeds import module. If the content type has dependencies, then it will immediately start to import its dependencies after the completion of the first import (content type). For instance, Contractor has five dependencies which are Contractor_import_1, Contractor_import_2, Contractor_import_3, Contractor_import_4, and finally contractor_import_5, so this module will automatically start to import all its dependences in series. Let me put this another way, normally when user import contractor_import_1 after completion of import process the system will show message (failure/success with report download link) to user, but now with the help of this module system will work as following.

When the user tries to import "contractor_import_1", if all the information related to contractor_import_1 is saved into database without error, then before it displays the success message to user, it will check whether this contractor has dependencies or not if it has, then it will start to import its dependencies and once all the dependencies import is finished then it will complete the importing process and display the message (failure/success with report download link) to user. Otherwise, if some error is displayed and the data couldn't be save to database then system will not start to import its dependencies and it will immediately display the failure message.

Success factors

Looking at the scope of this problem, knowledge of several computer science subjects are required to propose a workable solution. We will mainly need knowledge and concepts of object-oriented programming, algorithms and data structures, database design, HTML/CSS and software engineering courses.

Object-oriented programming knowledge is required because our capabilities can be massively augmented by machines. While being able to simply ask the machine for "what you need" gives you some results, knowing how to clearly tell it "what to do" can give you more personalized results and make any automatable tasks easier to automate. Object-oriented programming has great advantages over other programming styles such as code reuse and recycling, encapsulation, design benefits and etc. The most important thing is that Drupal is completely developed based on object-oriented programming principles [8] and [9].

Algorithm and Data structure knowledge and concepts are required in order to write proper programming statements that can be easily coded in reduced number of code lines. Algorithm tells exactly how to do

DOI: 10.21522/TIJAR.2014.04.02.Art013
ISSN: 2520-3088

something to reach our goal or to achieve the required output or to accomplish our task. Programming is all about applying logic and reason to solving technical /programming problems. When you solve the problem, you arrive at an algorithm. Your algorithm is the solution to your problem. Let me put this another way: Your algorithm is the solution to your problem. If you don't have an algorithm, you haven't solved your programming problem. An algorithm is the series of logical instructions for the computer that you translate into the programming language of your choice, whether that be Java, Python, C++, Go, Ruby or PHP [1] and [7].

When we code our solution, we may indeed lean on the work of other people. We may use libraries and frameworks. This is why code reusability is so important. Nobody writes software from scratch. We build on top of an existing foundation. However, the set of unsolved programming problems in the universe is infinite. When we write an application, it is just one of many programming problems. The importance of algorithms is that they constitute all of programming techniques. Data structures is the best way of storing and organizing information in a computer so that it could be retrieved and used most productively and efficiently. Data structures are key to designing efficient algorithms.

Database design and development concepts are required to store and retrieve data in a convenient and efficient manner. Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information. In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts by unauthorized users. If data is to be shared among several users, the system must avoid possible anomalous results [1].

HTML/CSS skills and knowledge is required in order to design the interfaces or layout of the web pages. The CSS will define styles for our documents, including the design, layout and variations in display for different devices and screen sizes [13].

Software engineering which is a set of methodologies, techniques and tools which helps us to build high quality software that works and fits our budget based with the consideration of organizational requirements. It is an important and critical discipline, concerned with cost effective software development. It is based on a systematic approach that uses appropriate tools and techniques, operates under specific development constraints, and most importantly, follows a process with the consideration of the user and business requirements. As a result, a good programmer should follow and use the software engineering principles along with algorithm and data structure knowledge to develop a robust and high performance application [11].

Limitations

Our proposed solution is developed to improve the CIMS features of and existing system for "Data Import" process. The developed module is dependent on the CIMS existing modules to work properly and efficiently. However, the developed module could be customized to any organization's requirement with little efforts using its dynamic features. The solution is limited to the CIMS developed using Drupal, PHP programming language and feeds import module. The data flow in the system is customized and automated to meet the x-organization and its agencies requirements to improve their day to day activities and tasks.

Goals and achievements

Our goal is to design and develop a module to support and make it easier for those Drupal users who have multiple feeds importer (for any reasons) for its one content type, which forces the user to import one file (CSV) several times in order to accomplish the bulk upload. We aim to eliminate the needs for importing one content type multiple times. The module is designed to be dynamic, flexible and easy to use. The newly developed and customized module will possess the following main features:

- Can be used for other databases which are having the same senior for instance, the Contractor Information Management System has been configured as multisite in different countries (having same source code, but different databases schemas).
- Designed to be very user friendly.
- Allows the administrator to select their bulk importers.

- Allows users to import the contractors in series.
- The module will work for all content types which have dependencies (field collection, map and etc.).
- Enables the administrator to apply security policies.
- Displays the report messages ONCE at the end of bulk import.
- Can be only accessed through the agencies specific users.
- Designed in a manner which allows the administrator to add unlimited bulk importers.
- Works for all Drupal users who are using field collection and feeds import together.

Method

In order to solve the stated problem and answer the research questions we will have to develop a module with the capability to provide the required features based on the organizational needs. We consider the following best software engineering practices and use some tools to improve software development phases, introduced the required features, and to support software development tasks in general.

Tools usage and automation processes are fundamental components in software engineering. And they are essential for improving productivity, efficiency, and effectiveness of our activities in the software development process. Hence, we have used several tools in this research such as Integrated Development Environment (IDE) eclipse, Version Control System Git and UML Diagraming tools such as Signavio.

Software process model

Process modelling is the best way to visualize a business process and to make the process much more understandable for technical and non-technical audience. The software development process has to go through a cycle in order to be delivered reliability, effectively and efficiently. Therefore, choosing the right lifecycle model is of fundamental importance. First, we describe the software lifecycle or software processes.

We have a several software process models for instance: waterfall process, spiral process, evolutionary prototyping process, rational unified process and agile process. Since, this application has been already developed based on Scrum Agile development method, so we will continue with is software development model. Agile process is one of the best models which is very useful for small and medium-size software development projects.

Agile is a group of software development methods that are based on highly iterative and incremental development. In particular, I'm going to discuss Test Driven Development or TDD.

This involves the iteration of three main phases. In the first phase, we mark as red, we write test cases that represents our requirements, and for which we have not written code yet. Therefore, they will fail, obviously. So, we're in this sort of red or fail phase.

From this phases, we move to another phase, in which after we write enough code to make the test cases pass. We have a set of test cases that are all passing. Therefore, we can consider this as the green phase. We had enough code to make the test cases pass because the test cases represented our requirements. We have just written enough code to satisfy our requirements.

When we do this over time the structure of the code deteriorates, because we keep adding pieces. So, that's why we have the first step, which is refactoring. In this step, we modify the code, and we will talk about refactoring extensively later in this paper. We modify the code to make it more readable and maintainable. In general, we modify to improve the design of the code. After this phase, we will go back to write more test cases for new requirements and to write code that makes these test cases pass, and this process continues. So, we'll continue to iterate among these phases.

Also, in this case, we will talk about agile software processes. In particular, about extreme programming, or XP, and Scrum in more details.

I would like to go with Scrum, Agile process or TDD (Test-driven development) because, agile process is one of the best model which is very useful for small and medium software development projects.

Agile has the following two processes, which will be discussed one by one.

DOI: 10.21522/TIJAR.2014.04.02.Art013
ISSN: 2520-3088

XP (extreme programming)

Kent Beck describes the XP as “XP is a lightweight methodology for small to medium sized teams developing software in the face of vague or rapidly changing requirement”. So as a result, XP is a lightweight, humanistic, discipline and all about software development.

Scrum

Scrum is a lightweight process framework for agile development and the most popular one.

Process framework means that it follows a particular set of practices to deliver high-quality software and to meet the requirement of stakeholders based on iteration. And lightweight means that keeps the task as small as possible.

There are three main kinds of actors in scrum development model.

- Product owner (customer): is mainly responsible for the product backlog, where the Product backlog is basically the list of things that have to be done, the backlog is in fact for the project. And that is analogous to the user stories to be realized in XP. So, what the product owner does is to clearly express these backlogs items and to also order them by value or weight, so they can be prioritized.
- Team: The team is responsible for delivering shippable increments to estimate the amount of development effort required for the backlog items. It is normally self-organized and consists of four to nine people, and it is what you would consider normally as the main development team in a project. In other words, in Scrum development model there is no need for overall team leader, who decides which person will do which task or how a problem will be solved. These are issues that are decided by the team as a whole.
- Scrum Master: is the person who’s is responsible for overall Scrum process, so he or she has to remove obstacles, facilitate events, helps communications, and so on. So, you can see the Scrum master as sort of a manager or the person who's got oversight, or the supervisor of the Scrum process.

Software phases

Software processes are normally characterized by several phases, which is called the software phases, and only one of these phases are mainly focused on coding. The other phases are meant to support other parts of software development process. [10]

1. Requirements engineering
2. Design (high level structure)
3. Implementation (write code)
4. Verification and validation
5. Maintenance

Requirement engineering

Based on the above scenario (problem scope) or business requirements, the below system requirement should be achieved:

- The module should be designed to be very user friendly;
- The module should allow administrator to select their bulk importers;
- The module should allow users to import the contractors in series;
- The module should work for all types of fields collection as well as other modules;
- The administrator should be able to apply security policies (in this custom module);
- The module should be able to show the report messages once at the end of bulk import;
- The module should be accessed only through the agencies’ specific users;
- The module should be designed in a manner which allow the administrator to add unlimited bulk importers;

The module should eliminate the need for importing the same CSV file several times;

Design

This view shows all the main components of a CIMS upload module for X-Organization. It makes it easier for developer to understand the processes, the components, dependencies and critically of the system. Based on the design the developer can easily test the module.

The below process activity diagram is designed in Signavio. This tool makes the job of developers much easier. As shown in Figure .

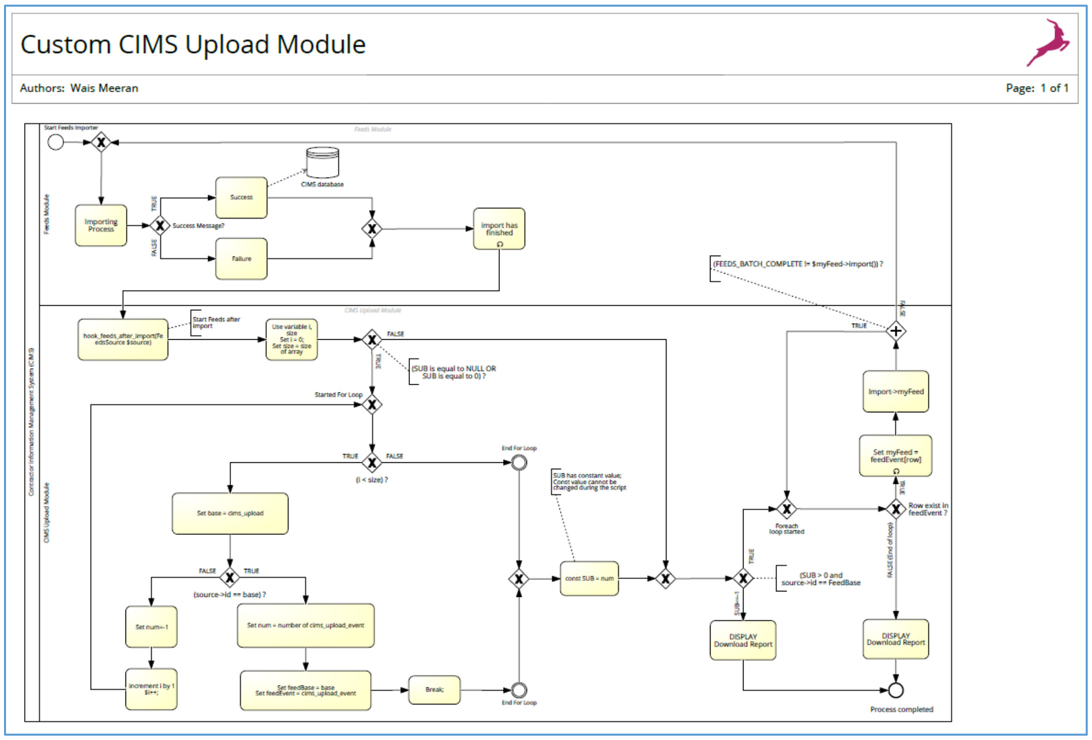

Custom CIMS Upload Module
Authors: Wais Meeran
Page: 1 of 1
Contractor Information Management System (CIMS)
Feeds Module
Start Feeds Importer
Importing Process
Success Message?
TRUE
FALSE
Success
Failure
CIMS database
import has finished
(FEEDS_BATCH_COMPLETE != $myFeed->import()) ?
CIMS Upload Module
hook_feeds_after_import(FeedsSource $source)
Start Feeds after import
Use variable i, size Set i = 0; Set size = size of array
(SUB is equal to NULL OR SUB is equal to 0) ?
Started For Loop
(i < size) ?
End For Loop
Set base = cims_upload
(source->id == base) ?
Set num=-1
Set num = number of cims_upload_event
increment i by 1 $i++;
Set feedBase = base Set feedEvent = cims_upload_event
Break;
SUB has constant value; Const value cannot be changed during the script
const SUB = num
(SUB > 0 and source->id == FeedBase
SUB==-1
DISPLAY Download Report
Foreach loop started
Row exist in feedEvent ?
FALSE (End of loop)
Set myFeed = feedEvent[row]
Import->myFeed
Process completed

Figure 1. This view shows all the main process of CIMS upload module

Implementation

PHP Language

We need to select a programming language for our server side. And according to our requirements PHP is the best choice since Drupal is based on PHP.

Backend process

This module has one main function and seven auxiliary functions2.

The auxiliary functions are: hook_help(), hook_permission(), hook_menu(), hook_form(), add_remove_ajax_callback(), find_feeds_by_id() and list_feeds(). The description of each functions is described in a module code as inline comments.

The one main function (hook_feeds_after_import()) has been used in order to accomplish the task[6].

Function hook_form()

As per the organizational requirement we need to develop a dynamic field. Dynamic fields must allow the administrator to add unlimited bulk importer and bulk importer events furthermore, the dynamic fields should be adding at the tail of the list, but should be deleted from anywhere in the list or vice versa. The

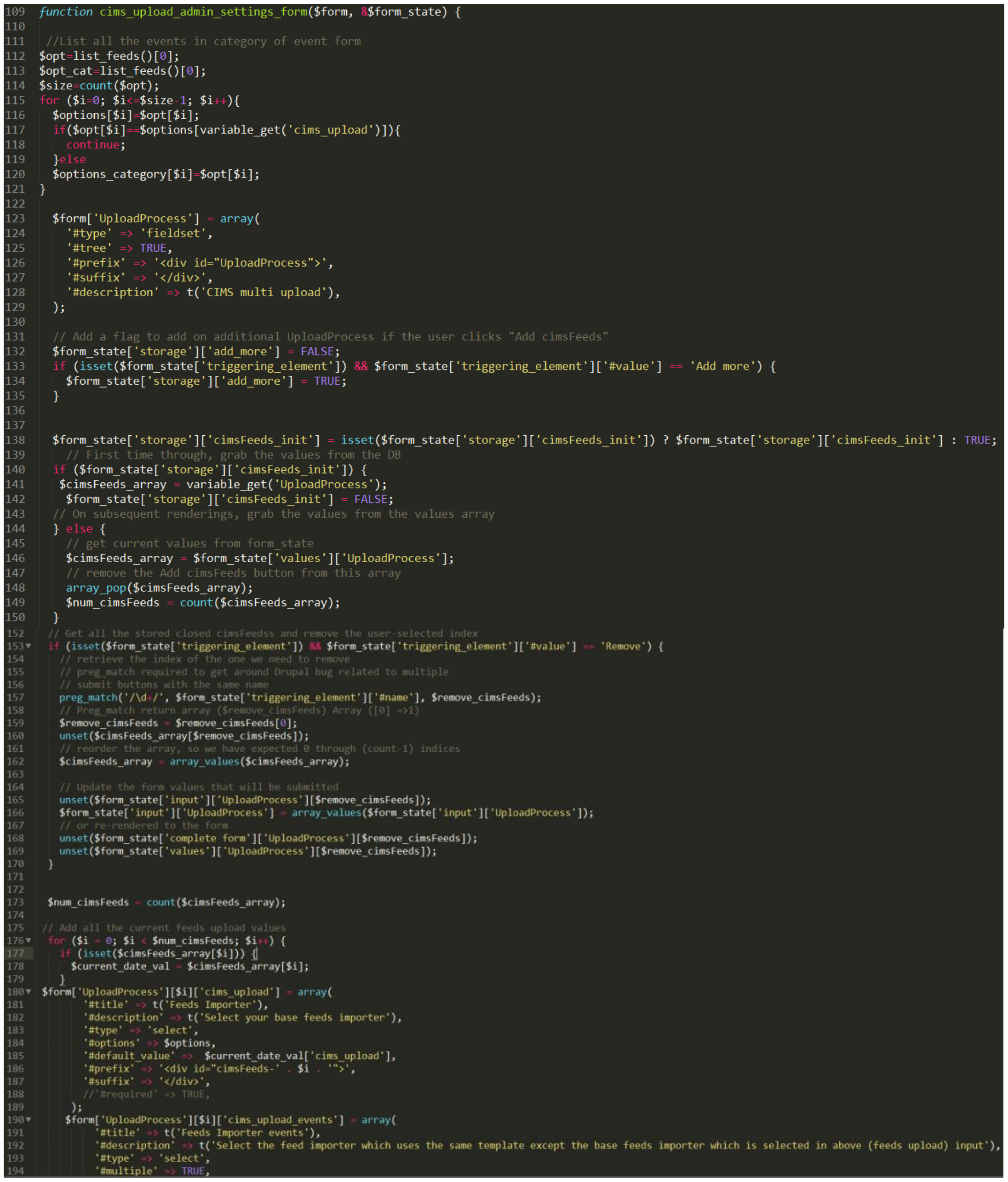
[2] To develop a custom module, I have used the following source [3].

DOI: 10.21522/TIJAR.2014.04.02.Art013
ISSN: 2520-3088

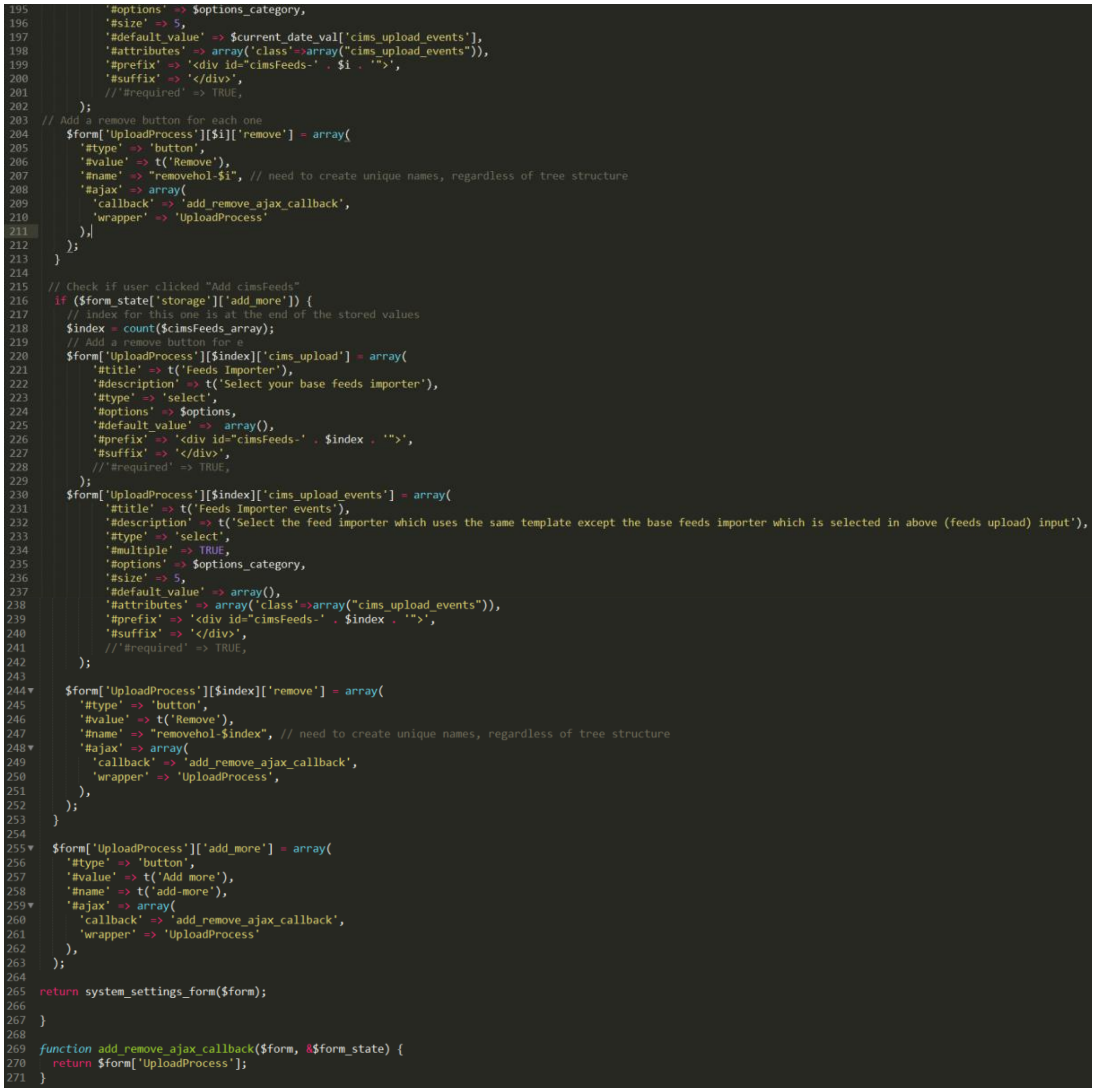
Dynamic form has implemented by using the following sources [4], [5], [12] and [14]. Figure is the complete code of dynamic field.

To implement the dynamic field, I have considered array data structure.

An array data structure is an ordered map in PHP. A map is a type that associates values to keys and this type is optimized for several different uses, it means that it can be treated as an array, list, hash table dictionary collection, stack, queue and more [2].



```
function cims_upload_admin_settings_form($form, &$form_state) {

 //List all the events in category of event form
$opt=list_feeds()[0];
$opt_cat=list_feeds()[0];
$size=count($opt);
for ($i=0; $i<=$size-1; $i++){
  $options[$i]=$opt[$i];
  if($opt[$i]==$options[variable_get('cims_upload')]){
    continue;
  }else
  $options_category[$i]=$opt[$i];
}

  $form['UploadProcess'] = array(
    '#type' => 'fieldset',
    '#tree' => TRUE,
    '#prefix' => '<div id="UploadProcess">',
    '#suffix' => '</div>',
    '#description' => t('CIMS multi upload'),
  );

  // Add a flag to add on additional UploadProcess if the user clicks "Add cimsFeeds"
  $form_state['storage']['add_more'] = FALSE;
  if (isset($form_state['triggering_element']) && $form_state['triggering_element']['#value'] == 'Add more') {
    $form_state['storage']['add_more'] = TRUE;
  }


  $form_state['storage']['cimsFeeds_init'] = isset($form_state['storage']['cimsFeeds_init']) ? $form_state['storage']['cimsFeeds_init'] : TRUE;
    // First time through, grab the values from the DB
  if ($form_state['storage']['cimsFeeds_init']) {
   $cimsFeeds_array = variable_get('UploadProcess');
    $form_state['storage']['cimsFeeds_init'] = FALSE;
  // On subsequent renderings, grab the values from the values array
  } else {
    // get current values from form_state
    $cimsFeeds_array = $form_state['values']['UploadProcess'];
    // remove the Add cimsFeeds button from this array
    array_pop($cimsFeeds_array);
    $num_cimsFeeds = count($cimsFeeds_array);
  }
 // Get all the stored closed cimsFeedss and remove the user-selected index
 if (isset($form_state['triggering_element']) && $form_state['triggering_element']['#value'] == 'Remove') {
   // retrieve the index of the one we need to remove
   // preg_match required to get around Drupal bug related to multiple
   // submit buttons with the same name
   preg_match('/\d+/', $form_state['triggering_element']['#name'], $remove_cimsFeeds);
   // Preg_match return array ($remove_cimsFeeds) Array ([0] =>1)
   $remove_cimsFeeds = $remove_cimsFeeds[0];
   unset($cimsFeeds_array[$remove_cimsFeeds]);
   // reorder the array, so we have expected 0 through (count-1) indices
   $cimsFeeds_array = array_values($cimsFeeds_array);

   // Update the form values that will be submitted
   unset($form_state['input']['UploadProcess'][$remove_cimsFeeds]);
   $form_state['input']['UploadProcess'] = array_values($form_state['input']['UploadProcess']);
   // or re-rendered to the form
   unset($form_state['complete form']['UploadProcess'][$remove_cimsFeeds]);
   unset($form_state['values']['UploadProcess'][$remove_cimsFeeds]);
 }


 $num_cimsFeeds = count($cimsFeeds_array);

// Add all the current feeds upload values
 for ($i = 0; $i < $num_cimsFeeds; $i++) {
   if (isset($cimsFeeds_array[$i])) {
     $current_date_val = $cimsFeeds_array[$i];
   }
$form['UploadProcess'][$i]['cims_upload'] = array(
         '#title' => t('Feeds Importer'),
         '#description' => t('Select your base feeds importer'),
         '#type' => 'select',
         '#options' => $options,
         '#default_value' =>  $current_date_val['cims_upload'],
         '#prefix' => '<div id="cimsFeeds-' . $i . '">',
         '#suffix' => '</div>',
         //'#required' => TRUE,
     );
    $form['UploadProcess'][$i]['cims_upload_events'] = array(
          '#title' => t('Feeds Importer events'),
          '#description' => t('Select the feed importer which uses the same template except the base feeds importer which is selected in above (feeds upload) input'),
          '#type' => 'select',
          '#multiple' => TRUE,
```

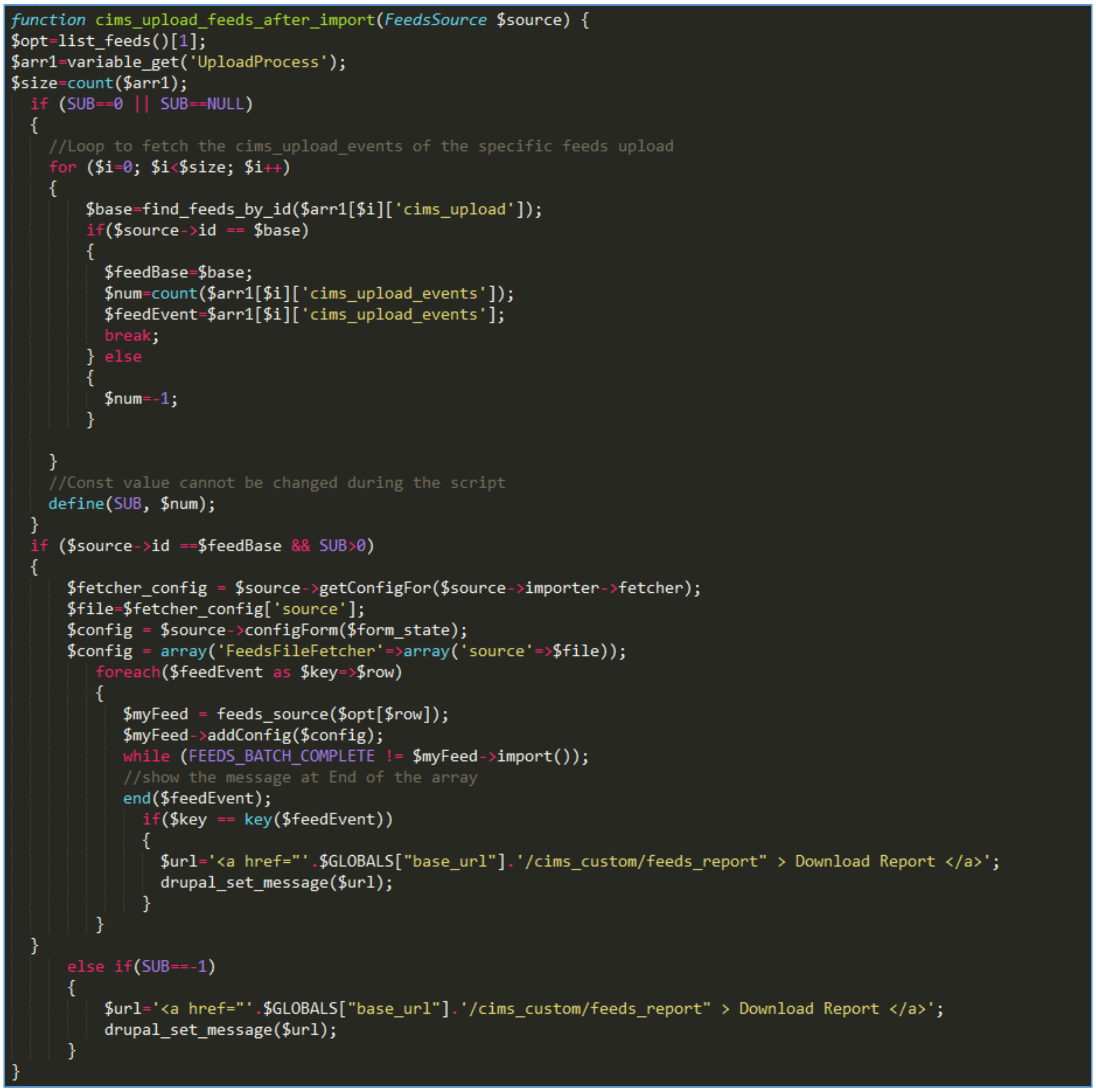


```php
            '#options' => $options_category,
            '#size' => 5,
            '#default_value' => $current_date_val['cims_upload_events'],
            '#attributes' => array('class'=>array("cims_upload_events")),
            '#prefix' => '<div id="cimsFeeds-' . $i . '">',
            '#suffix' => '</div>',
            //'#required' => TRUE,
      );
// Add a remove button for each one
    $form['UploadProcess'][$i]['remove'] = array(
      '#type' => 'button',
      '#value' => t('Remove'),
      '#name' => "removehol-$i", // need to create unique names, regardless of tree structure
      '#ajax' => array(
        'callback' => 'add_remove_ajax_callback',
        'wrapper' => 'UploadProcess'
      ),
    );
  }

 // Check if user clicked "Add cimsFeeds"
  if ($form_state['storage']['add_more']) {
    // index for this one is at the end of the stored values
    $index = count($cimsFeeds_array);
    // Add a remove button for e
    $form['UploadProcess'][$index]['cims_upload'] = array(
        '#title' => t('Feeds Importer'),
        '#description' => t('Select your base feeds importer'),
        '#type' => 'select',
        '#options' => $options,
        '#default_value' =>  array(),
        '#prefix' => '<div id="cimsFeeds-' . $index . '">',
        '#suffix' => '</div>',
        //'#required' => TRUE,
      );
    $form['UploadProcess'][$index]['cims_upload_events'] = array(
          '#title' => t('Feeds Importer events'),
          '#description' => t('Select the feed importer which uses the same template except the base feeds importer which is selected in above (feeds upload) input'),
          '#type' => 'select',
          '#multiple' => TRUE,
          '#options' => $options_category,
          '#size' => 5,
          '#default_value' => array(),
          '#attributes' => array('class'=>array("cims_upload_events")),
          '#prefix' => '<div id="cimsFeeds-' . $index . '">',
          '#suffix' => '</div>',
          //'#required' => TRUE,
      );

    $form['UploadProcess'][$index]['remove'] = array(
      '#type' => 'button',
      '#value' => t('Remove'),
      '#name' => "removehol-$index", // need to create unique names, regardless of tree structure
      '#ajax' => array(
        'callback' => 'add_remove_ajax_callback',
        'wrapper' => 'UploadProcess',
      ),
    );
  }

  $form['UploadProcess']['add_more'] = array(
    '#type' => 'button',
    '#value' => t('Add more'),
    '#name' => t('add-more'),
    '#ajax' => array(
      'callback' => 'add_remove_ajax_callback',
      'wrapper' => 'UploadProcess'
    ),
  );

return system_settings_form($form);

}

function add_remove_ajax_callback($form, &$form_state) {
  return $form['UploadProcess'];
}
```

Figure 2. The view shows the compete code of dynamic field in drupal

Function hook_feeds_after_import()
This function is used when module requires to perform an operation after a feed source has been parsed. Figure 3 is the complete code which runs after feed source has been parsed.

DOI: 10.21522/TIJAR.2014.04.02.Art013
ISSN: 2520-3088



```
function cims_upload_feeds_after_import(FeedsSource $source) {
$opt=list_feeds()[1];
$arr1=variable_get('UploadProcess');
$size=count($arr1);
  if (SUB==0 || SUB==NULL)
  {
    //Loop to fetch the cims_upload_events of the specific feeds upload
    for ($i=0; $i<$size; $i++)
    {
        $base=find_feeds_by_id($arr1[$i]['cims_upload']);
        if($source->id == $base)
        {
          $feedBase=$base;
          $num=count($arr1[$i]['cims_upload_events']);
          $feedEvent=$arr1[$i]['cims_upload_events'];
          break;
        } else
        {
          $num=-1;
        }

    }
    //Const value cannot be changed during the script
    define(SUB, $num);
  }
  if ($source->id ==$feedBase && SUB>0)
  {
      $fetcher_config = $source->getConfigFor($source->importer->fetcher);
      $file=$fetcher_config['source'];
      $config = $source->configForm($form_state);
      $config = array('FeedsFileFetcher'=>array('source'=>$file));
        foreach($feedEvent as $key=>$row)
        {
          $myFeed = feeds_source($opt[$row]);
          $myFeed->addConfig($config);
          while (FEEDS_BATCH_COMPLETE != $myFeed->import());
          //show the message at End of the array
          end($feedEvent);
            if($key == key($feedEvent))
            {
              $url='<a href="'.$GLOBALS["base_url"].'/cims_custom/feeds_report" > Download Report </a>';
              drupal_set_message($url);
            }
        }
  }
      else if(SUB==-1)
      {
        $url='<a href="'.$GLOBALS["base_url"].'/cims_custom/feeds_report" > Download Report </a>';
        drupal_set_message($url);
      }
}
```

Figure 3. The compete code which runs after feed source has been parsed

White-box testing

White box testing is the kind of testing that we perform when we open up the box. It means that we look inside the program, and we actually test it based on its code.

Based on automatic testing (branch and condition coverage) in Signavio our code coverage is 100%.

Maintenance

Maintenance is a fundamental activity in software life cycle.

Below are the reasons why software maintenance is a necessary phase in software development.

- Bug report
- Feature request
- Environment change

- Development organizations perform three kinds of maintenance activities.
- Corrective maintenance: to eliminate problem with code
- Perfective maintenance: to accommodate feature request (improve the software)
- Adaptive maintenance: to take care of the environment changes

This module is part of perfective maintenance since X-organization wants us to add feature in the existence application.

In each one of the above maintenance activities, the iteration will be done. It means that again we will need to identify the requirement for any of the above maintenance activities which includes design, implementation, verification and validation and finally we will be able to release the product/ module.

Results

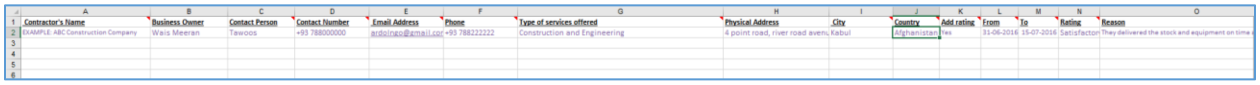
While developing this module, I have considered the knowledge and concepts of different computer science subjects for example: object oriented programming, algorithms, data structure, database design, HTML/CSS and Software Engineering best practices. As a result, a user friendly and extendable module which eliminates the need for importing one content type multiple times has been developed.

Bulk import of contractor/contracts

Prepare two CSV Files – one for Contractors and one for Contracts using the template provided (Below details 'Contractor' Tab). Table 1. Shows the contractors template for parsing into CIMS database, Table 2. Shows the contracts template for parsing into CIMS database.

Contractors

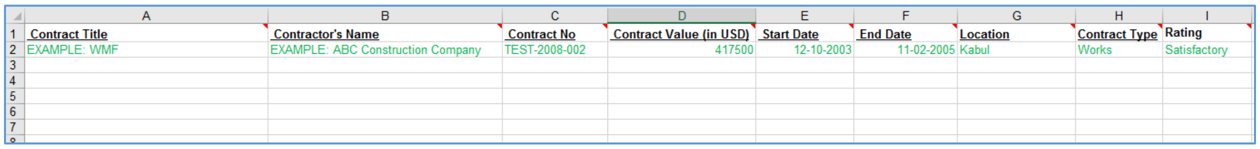
Table 1. Shows the contractors template for parsing into CIMS database



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Contractor's Name	Business Owner	Contact Person	Contact Number	Email Address	Phone	Type of services offered	Physical Address	City	Country	Add rating	From	To	Rating	Reason
2	EXAMPLE: ABC Construction Company	Wais Meeran	Tawoos	+93 788000000	ardoingo@gmail.cor	+93 788222222	Construction and Engineering	4 point road, river road avenu	Kabul	Afghanistan	Yes	31-06-2016	15-07-2016	Satisfactor	They delivered the stock and equipment on time
3															
4															
5															
6															

Contracts

Table 2. Shows the contracts template for parsing into CIMS database



	A	B	C	D	E	F	G	H	I
1	Contract Title	Contractor's Name	Contract No	Contract Value (in USD)	Start Date	End Date	Location	Contract Type	Rating
2	EXAMPLE: WMF	EXAMPLE: ABC Construction Company	TEST-2008-002	417500	12-10-2003	11-02-2005	Kabul	Works	Satisfactory
3									
4									
5									
6									
7									

The Column under the CONTRACTS sheet for 'Contractor's Name' is to ensure that when the data is uploaded, the database links the relevant Contracts to the Contractor.

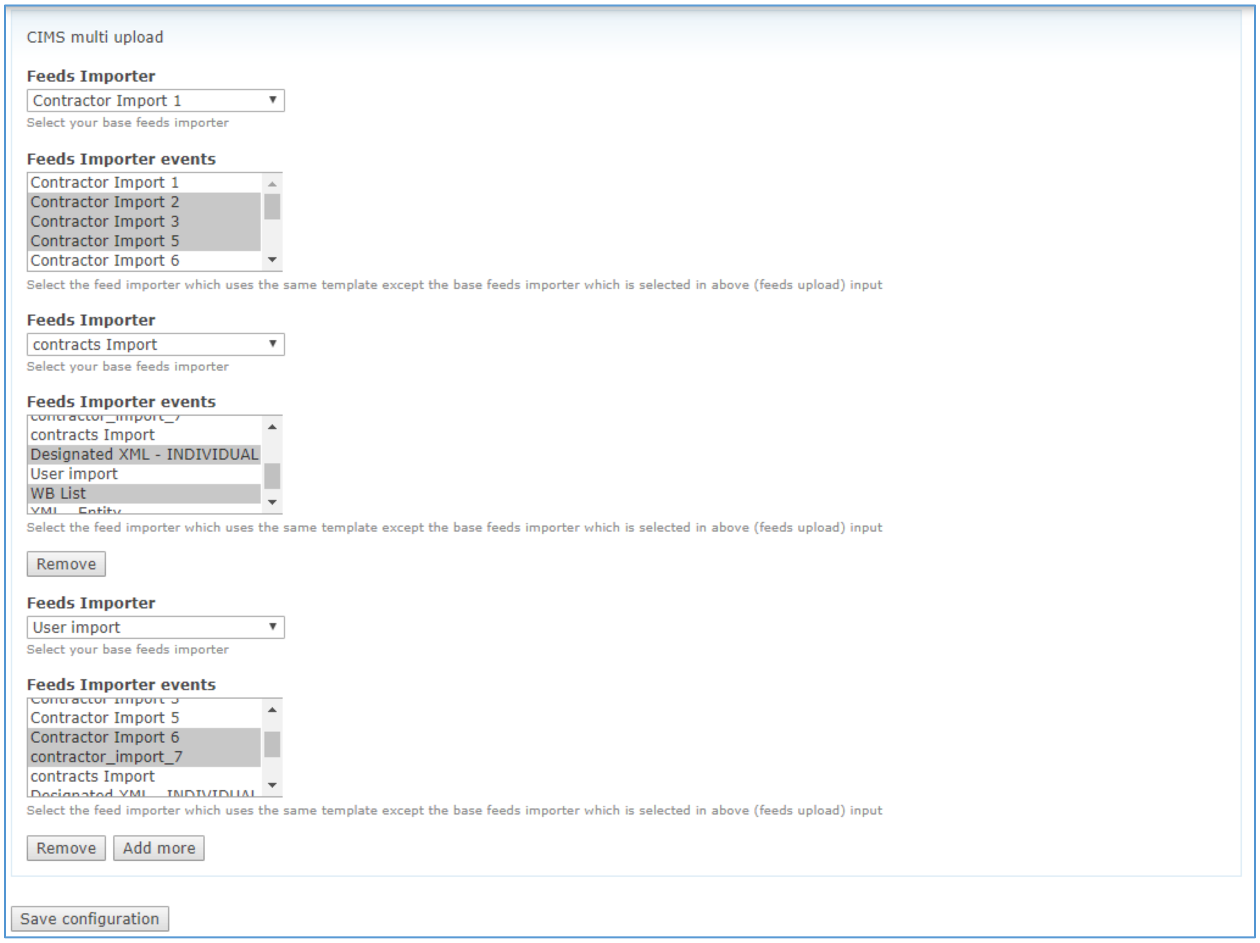
Therefore, it is important to ensure that the spelling you use for the Contractor's name is identical on both excel sheets.

It is only necessary to list the Contractors once in the CONTRACTOR sheet (i.e. you do not need to duplicate the details on the Contractor sheet if you have more than one contract.).

Administrator view

Configuration setting of bulk import. Figure shows the configurations and settings of the CIMS-import module

DOI: 10.21522/TIJAR.2014.04.02.Art013
ISSN: 2520-3088



CIMS multi upload

Feeds Importer
Contractor Import 1
Select your base feeds importer

Feeds Importer events
Contractor Import 1
Contractor Import 2
Contractor Import 3
Contractor Import 5
Contractor Import 6
Select the feed importer which uses the same template except the base feeds importer which is selected in above (feeds upload) input

Feeds Importer
contracts Import
Select your base feeds importer

Feeds Importer events
contracts Import
Designated XML - INDIVIDUAL
User import
WB List
Select the feed importer which uses the same template except the base feeds importer which is selected in above (feeds upload) input

Remove

Feeds Importer
User import
Select your base feeds importer

Feeds Importer events
Contractor Import 5
Contractor Import 6
contractor_import_7
contracts Import
Select the feed importer which uses the same template except the base feeds importer which is selected in above (feeds upload) input

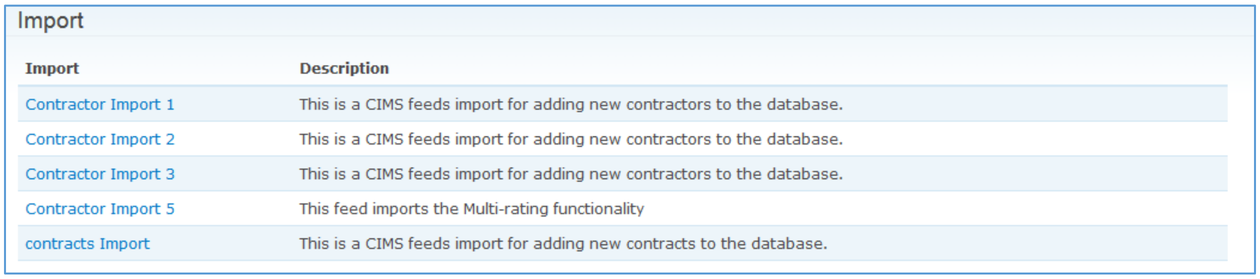
Remove Add more

Save configuration

Figure 4. This views shows the configurations and settings of the CIMS-import module

Importing process

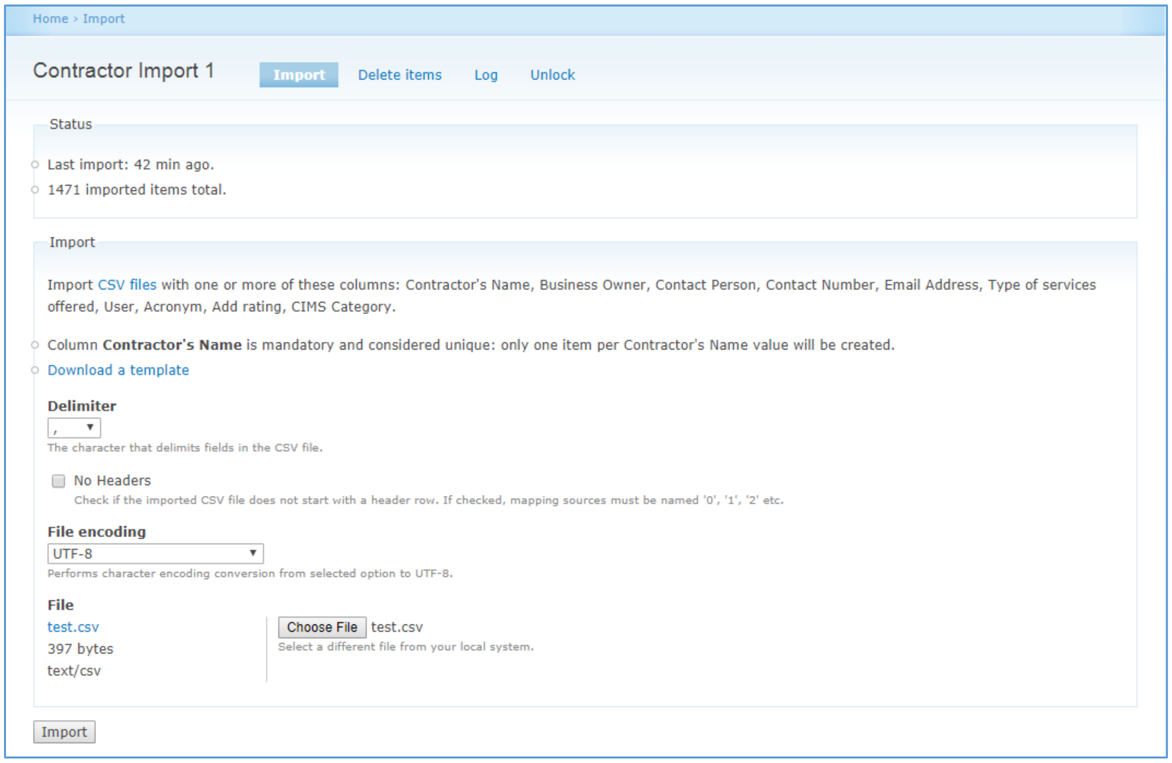
Previously, each feed import need had to be imported separately. For instance, user needs to import contractor import 1, contractor import 2, contractor import 3 and finally contractor import 5 furthermore, this process had to be done in order. But now by developing this new module, there is no need to import each feed import separately, the user only needs to import contractor import 1 and rest will be managed by the module as shown in below screenshots.



Import

Import	Description
Contractor Import 1	This is a CIMS feeds import for adding new contractors to the database.
Contractor Import 2	This is a CIMS feeds import for adding new contractors to the database.
Contractor Import 3	This is a CIMS feeds import for adding new contractors to the database.
Contractor Import 5	This feed imports the Multi-rating functionality
contracts Import	This is a CIMS feeds import for adding new contracts to the database.

Figure 5. This view shows the list of feeds importers



Status

- Last import: 42 min ago.
- 1471 imported items total.

Import

Import CSV files with one or more of these columns: Contractor's Name, Business Owner, Contact Person, Contact Number, Email Address, Type of services offered, User, Acronym, Add rating, CIMS Category.

- Column **Contractor's Name** is mandatory and considered unique: only one item per Contractor's Name value will be created.
- Download a template

Delimiter
,
The character that delimits fields in the CSV file.

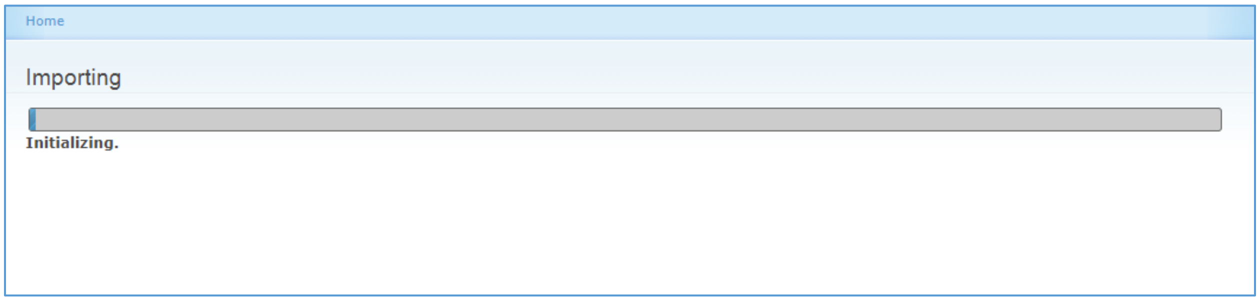
No Headers
Check if the imported CSV file does not start with a header row. If checked, mapping sources must be named '0', '1', '2' etc.

File encoding
UTF-8
Performs character encoding conversion from selected option to UTF-8.

File
test.csv
397 bytes
text/csv

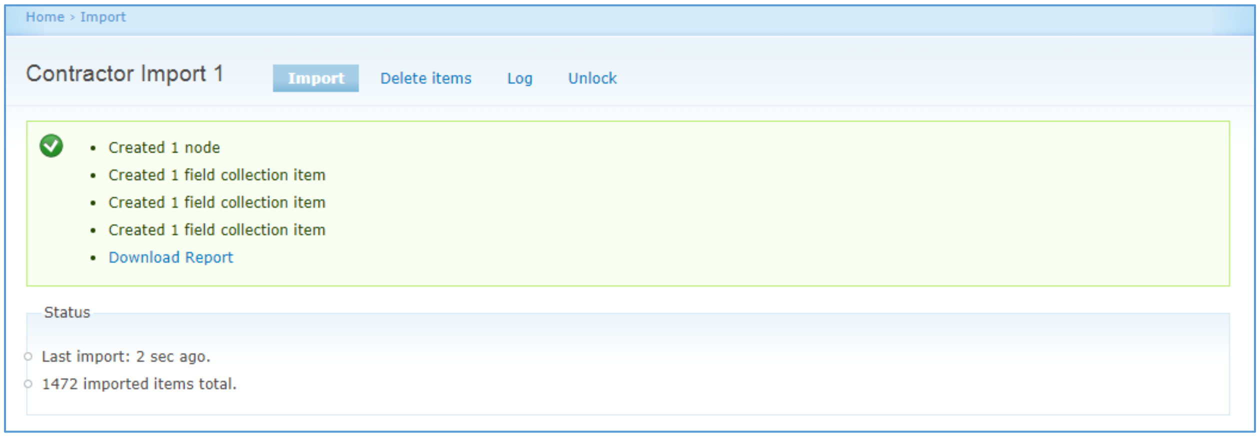
Choose File test.csv
Select a different file from your local system.

Figure 6. Shows the page where user can select a CSV file to import the contractor import 1



Initializing.

Figure 7. Shows the process of importing contractor import 1 with its dependencies



- Created 1 node
- Created 1 field collection item
- Created 1 field collection item
- Created 1 field collection item
- Download Report

Status

- Last import: 2 sec ago.
- 1472 imported items total.

Figure 8. Shows the result of imported contractors' import 1 with its dependencies.

DOI: 10.21522/TIJAR.2014.04.02.Art013
ISSN: 2520-3088

Discussion

The module is designed and developed from scratch. To the best of my knowledge, no one has worked on solving this problem previously. The purpose of this module is to support either X-organization or Drupal community who are using similar products or dealing with similar business processes.

Previously, X-organization had many problems with feeds import, location and field collection. For example, it was very time consuming and unprofessional to import one file several times and the clients had to import the files in order otherwise and error would have been produced. Furthermore, it created limitation in importing process. In other words, the more information we wanted to import, the more feeds import was required and that was making it a difficult job for the users to import one file 10 to 15 times. But by developing this module we do not need to worry about these complicated and time consuming processes since is managed by the custom module that I have developed to solve this problem.

Conclusion

In this paper, we analyzed X-Organization problem and collected the business requirements. We used software development method and other tools to design, implement, test, validate and deliver a customized module for an existing application. Luckily, when the proposed solution to problem was presented to X-Organization, they agreed to implement it after several rounds of testing in their UAT environment and finally it became part of their production system. As a result, the developed module is actively being used and complimented by X-Organization clients.

This module has been developed as an assignment and because of limited time and space I could not concentrate and present other parts of the overall application. Furthermore, enterprise application cannot be developed by a single person. Enterprise application development process needs multiple development teams and each team works separately on different part of the application and contribute to the entire software development lifecycle process.

References

[1] Abraham Silberschatz, Henry F. Korth, s. Sudarshan, (2006). Database System Concepts 6th.

[2] "Arrays in PHP", [Online] Available at: http://php.net/manual/en/language.types.array.php, [Accessed 20-Jul-2017].

[3] "Creating custom modules", [Online] Available at: https://www.drupal.org/docs/7/creating-custom-modules, [Accessed 22-Jul-2017].

[4] "Dynamic Fields", [Online] Available at: https://www.formget.com/how-to-dynamically-add-and-remove-form-fields-using-javascript/ , [Accessed 20-Feb-2017].

[5] "Drupal Form", [Online] Available at: https://www.sitepoint.com/understanding-forms-drupal/, [Accessed 20-Jun-2017]

[6] "Feeds", [Online] Available at:
http://www.drupalcontrib.org/api/drupal/contributions%21feeds%21feeds.api.php/7, [Accessed 15-Apr-2017].

[7] Mark Allen Weiss, (1999) Data Structures and Algorithm Analysis in C++, fourth Edition, Florida International University Indianapolis.

[8] "Object Oriented Programming", Paul and Harvey Deitel, [2011] C++: How to Program 8th edition

[9] "Object Oriented Programming", [Online] Available at:
https://www.cs.drexel.edu/~introcs/Fa15/notes/06.1_OOP/Advantages.html?CurrentSlide=3 , [Accessed 19-Feb-2017].

[10] "Programing with PHP", [Online] available at: https://www.w3schools.com/php/ , [Accessed 1-Feb-2017].

[11] "Software Engineering", [Online] Available at:
https://www.tutorialspoint.com/software_engineering/software_engineering_pdf_version.htm, [Accessed 10-May-2017].

[12] "Triggering element", [Online] Available at: https://www.drupal.org/node/2546700, [Accessed 18-Jun-2017].

[13] "Using HTML 5 and CSS", [Online] Available at: http://www.w3schools.com/ , [Accessed 10-Feb-2017]

[14] “Unique #name”, [Online] Available at: https://www.drupal.org/node/1342066, [Accessed 15-Jun-2017].