DESIGN OF AN ALGORITHM AND DEVELOPMENT OF AN ONLINE APPLICATION FOR MANAGING THE STRUCTURE OF PRODUCTS

PINTILIE Dănuț – Sebastian[1], GHEORGHIȚĂ Vlad[2]
[1]Faculty Industrial Engineering and Robotics, Specialization: IAII, Year of study: IV, e-mail: pintiliedanutsebastian@gmail.com
[2]Faculty Industrial Engineering and Robotics, TCM Department

ABSTRACT: The main concern of those involved in the manufacturing industry is to make products as correctly and quickly as possible and to use resources efficiently. Product management systems are the tool through which these requirements are met. These systems are so important because they provide the ability to manage and track all the information, applications and processes that define a product from the design phase until it is ready for market.

KEY WORDS: product, management, processes, web services, database.

1. Introduction

Product data management (PDM) systems typically manage product-related information such as geometry, engineering drawings, product specifications, CNC programs, analysis results, component tables, engineering change orders and more. PDM can also be seen as an integration tool that connects many different areas of product development, which ensures that the right information is available to the right person, at the right time and in the right form, across the enterprise. In this way, PDM improves communication and cooperation between different groups within the organisation, as well as between the organisation and customers. [1]

This paper focuses on the development of a PDM web application. A web application is software that performs a specific function using a web browser as a "client". Through the browser the client connects to processing functions that are located on a server. The typical structure of a web application is a 3-tier structure:

- Presentation Layer - The interface of the clinet, usually implemented using a browser-sensitive programming language, such as JavaScript or jQuery, combined with a markup language, such as HTML.
- Logical level - is the level where customer requests are received, processed and fulfilled. This layer differentiates a web application from a website. Client requests are transmitted via the HTTP communication protocol and are processed using algorithms in a dynamic development language such as PHP, ASP, ASP.NET and others.
- Data management layer - is represented by the database engine, such as MySQL, Oracle, etc., for managing data persistence and querying through appropriate tools, receiving and fulfilling DB read/write requests from the application logic [2]

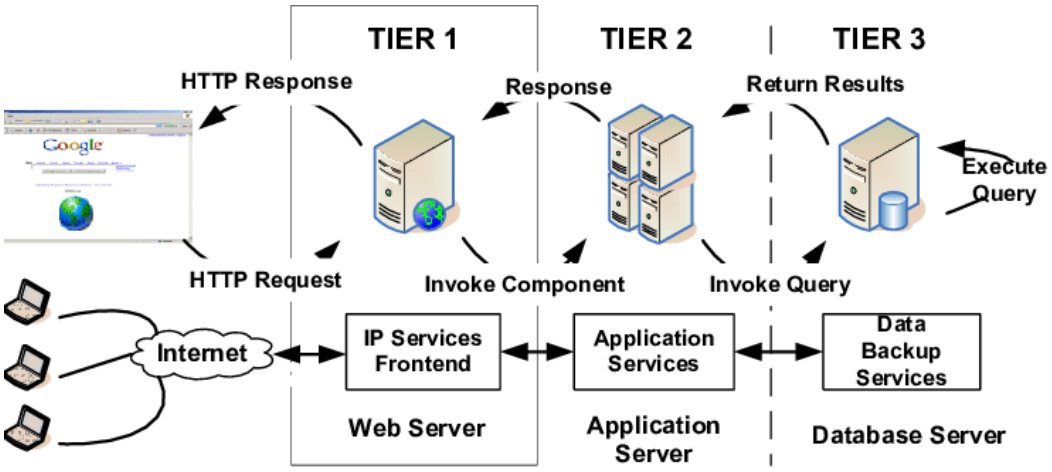

Figure 1. Typical structure of a web application [3]

The application in this paper aims to provide the possibility to manage the structure of products, in terms of their composition (BOM), the processes that products and their components undergo to reach their final state and the resources that are required to obtain the finished product. In order to fulfil these functionalities, we have structured the application on 3 levels, according to the typical structure. For the client interface we used the JavaScript programming language, together with the HTML markup language and the CSS formatting standard. For easier development of the interface we also used the Alpine.js framework which helps in easier development of dynamic elements in the web page, and for formatting elements we used the Tailwind CSS framework. The logic functions in the second level were realized using the LabVIEW (Laboratory Visual Instrumentation Engineering Workbench) programming environment. This is a well-known program for graphical instrumentation and coding (the graphical language is called "G") and is generally used for data acquisition, industrial automation or as control instrumentation. In this case, we used the Web Service module to be able to retrieve customer requests and process them further. Commands to the third level, the database represented in the MS Access application (MS Office package), were carried out using the Database functions in the Connectivity package of LabVIEW. These functions facilitate the connection to the database and the performance of database queries.[4]

2. Current status

At this moment, the application offers the possibility to query the database in order to display products to the user, add products to the database, define and display the composition and process structure of a product. These functionalities are accessible to the user through the web interface of the application. This interface has been built using HTML, JavaScript, CSS and the Tailwind CSS and Alpine.JS frameworks.

The form is divided into 3 sections: the general product information input area, the component input area and the process input area. The component and process input areas are initially defined with a single component or process definition area. To add more such zones, two functions have been defined in the JavaScript language, "adaugaComponenta()" and “adaugaProces()”.

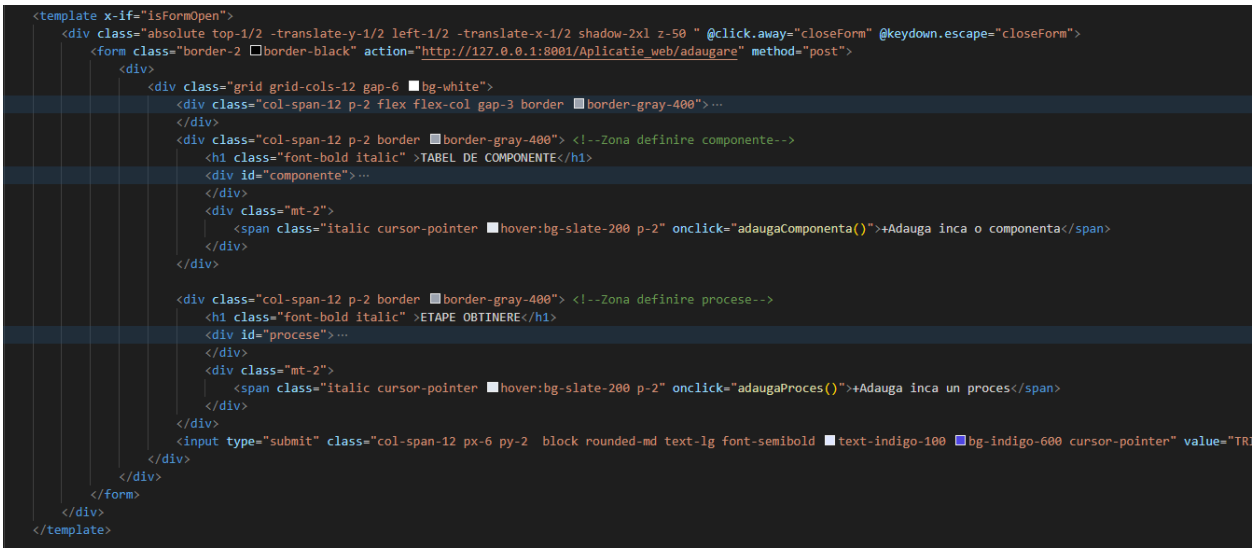

Figure 2. Definition of the form component

```
var b = 1;

function adaugaComponenta(){
    var n = parseInt(document.getElementById("nrcrt").innerHTML) + b;
    var i = document.createElement("div");
    i.className = document.getElementById('rand').className;
    var inputDenumire = document.createElement("input");
    inputDenumire.setAttribute("type","text");
    inputDenumire.setAttribute("id","denumireComponenta"+n);
    inputDenumire.setAttribute("name","denumireComponenta"+n);
    inputDenumire.className = "w-full h-full";
    var inputCantitate = document.createElement("input");
    inputCantitate.setAttribute("type","number");
    inputCantitate.setAttribute("id","cantitateComponenta"+n);
    inputCantitate.setAttribute("name","cantitateComponenta"+n);
    inputCantitate.className = "w-full h-full";
    var inputDisponibilitate = document.createElement("input");
    inputDisponibilitate.setAttribute("type","text");
    inputDisponibilitate.setAttribute("id","disponibilitateComponenta"+n);
    inputDisponibilitate.setAttribute("name","disponibilitateComponenta"+n);
    inputDisponibilitate.className = "w-full h-full";
    var id = document.createElement("div");
    var den = document.createElement("div");
    var cant = document.createElement("div");
    var disp = document.createElement("div");
    id.className = document.getElementById('nrcrt').className;
    den.className = document.getElementById('d').className;
    cant.className = document.getElementById('cantitate').className;
    disp.className = document.getElementById('disponibilitate').className;
    id.innerHTML = n;
    den.appendChild(inputDenumire);
    cant.appendChild(inputCantitate);
    disp.appendChild(inputDisponibilitate);
    i.appendChild(id);
    i.appendChild(den);
    i.appendChild(cant);
    i.appendChild(disp);
    document.getElementById('componente').appendChild(i)
    b++;
}
```

Figure 3. Defining the function "adaugaComponenta()"

This function creates, using the function "document.createElement", the „div” type element that defines the row for entering data. The same function is used to create the "input" add fields, which are assigned the

same attributes as the field originally defined with the help of ".setAttribute" function. With the function ".appendChild" the elements of type "input" are added to the element "row".

For the display and registration of the products I have developed algorithms in the LabVIEW programming environment, which I will present below.

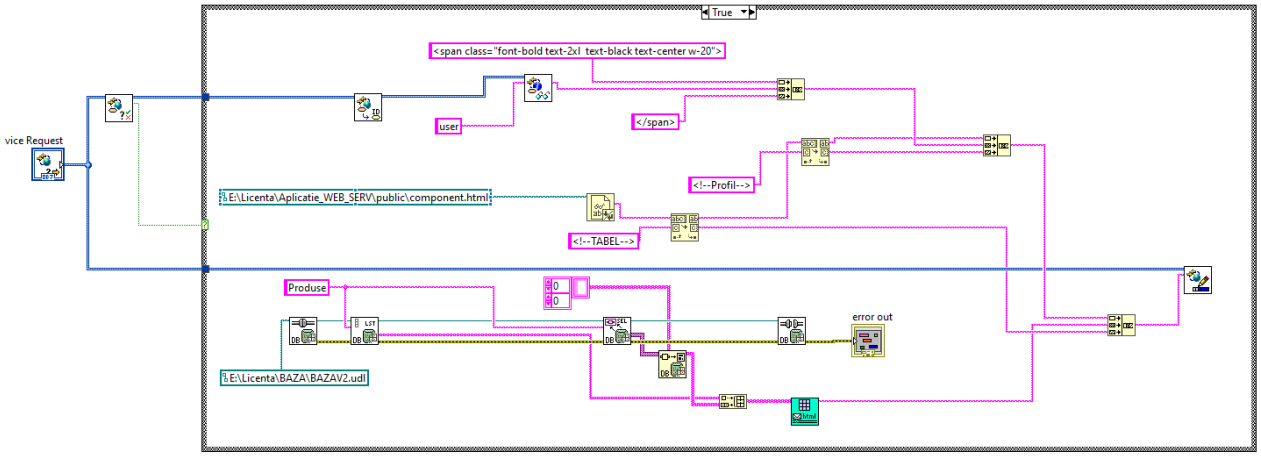

Figure 4. Algorithm for displaying products

The product display algorithm is accessible when the user has successfully passed the login page. This check is done with the function "Check If Session Exist.vi" if not, the user is redirected to the login page, if yes, the algorithm accesses the HTML file that it will process. In the section dedicated to the profile the name of the user will be added using the function "Read Session Variable.vi". At the same time, the algorithm accesses the database, from which it extracts the data from the "Produse" table, as well as its column names. The table thus formed is processed by the "Array2HTML" subprogram.

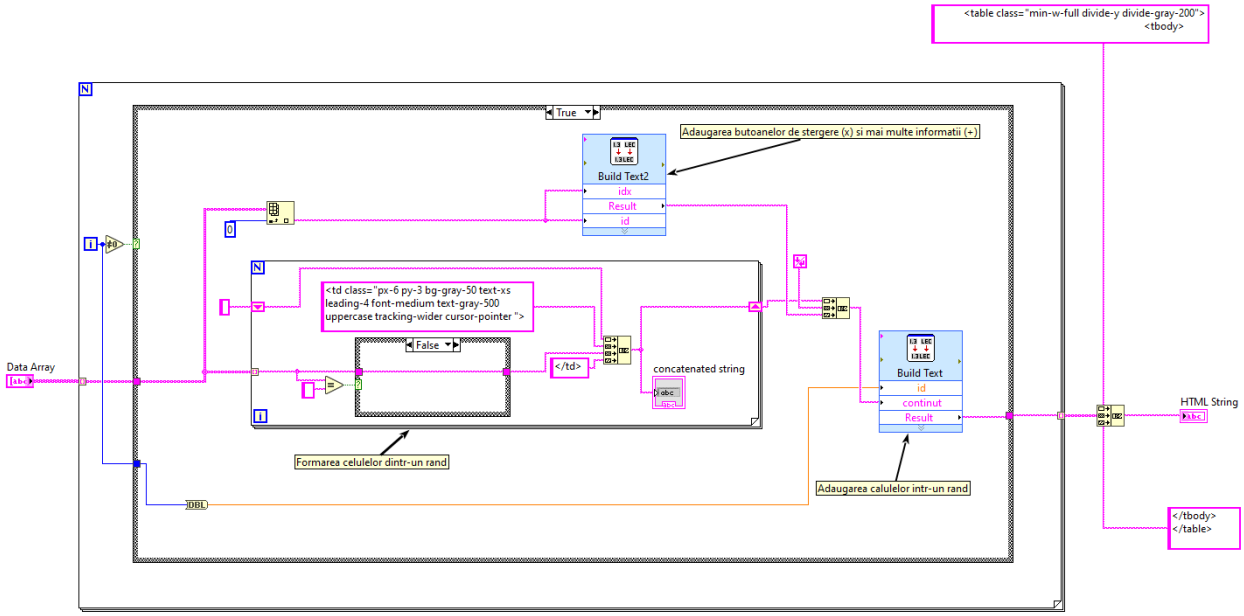

Figure 5. Subprogram for transforming an Array table in LabVIEW into an HTML table

In this algorithm, in order to access the table rows, the table is passed through a repetitive "For" structure. Then the first row is formatted to represent the head of the table. The rows are passed through another "for" structure to access the cells on each row. The data in these cells is added in html components of type "td". The array formed by the data of type "td" at the output of the second "for" is added in an html component of type "tr". The array formed at the output of the first "for" is added to an html component of type "table", and the text thus formed is sent to the main program. The main algorithm takes the text and inserts it, in the table section, into the text that is sent as a response via the "Write response.vi" function.

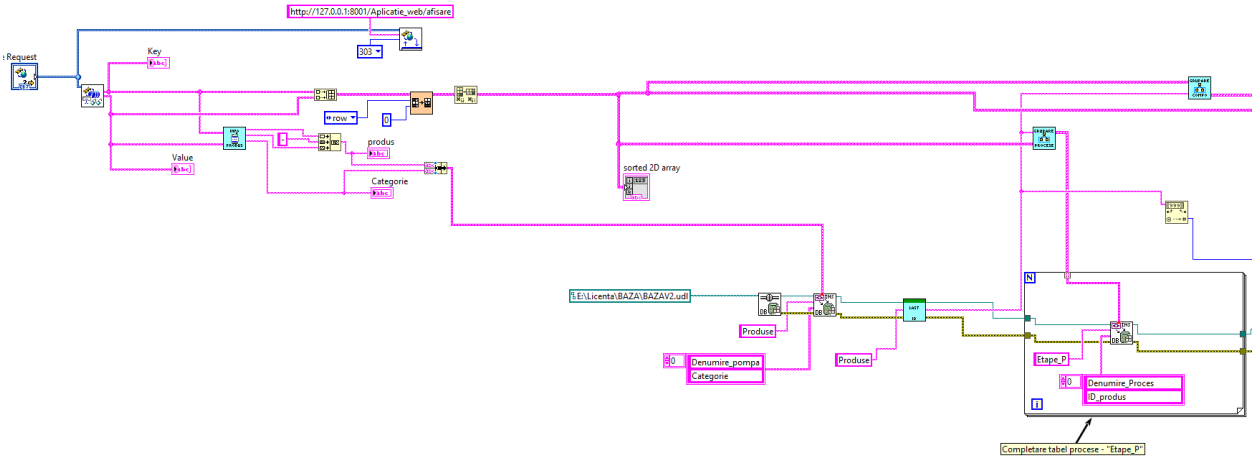

Figure 6. Data adding algorithm

The add product algorithm is called when the user submits the completed form with the required data. These are accessed using the "Read all form data.vi" function. The data is then grouped into arrays of clusters so that it can be added to the database. In order to be able to add the data to the tables, these arrays are traversed using a "for" structure, which is used to insert each data cluster into the corresponding table.

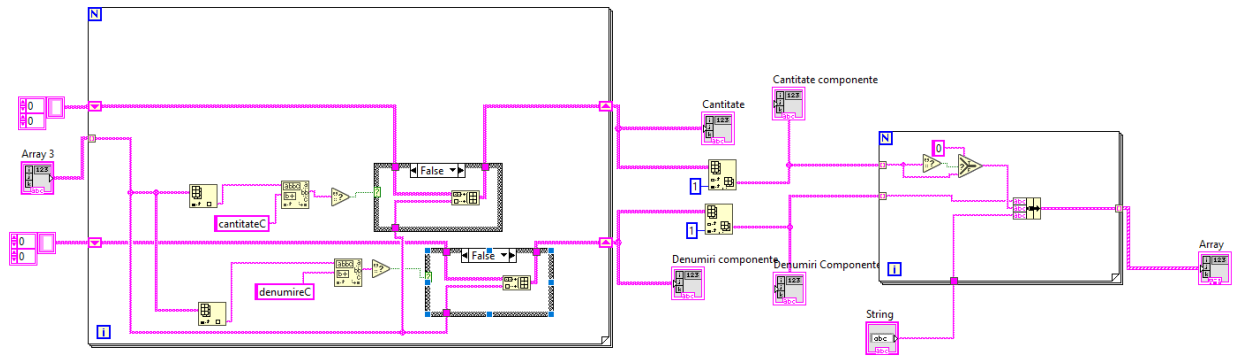

Figure 7. Subprogram for grouping component data

The grouping subprogram has as input data the array of all data in the form and the product id that is entered in the database. The algorithm selects only those data related to components, processes or resources, as needed, then the values of these data are grouped into clusters to be entered into the database.

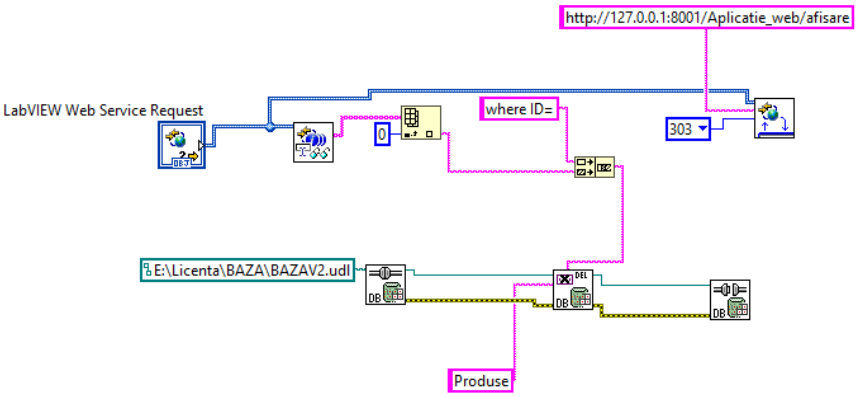

Figure 8. Algorithm for data removal

For deleting data from the database, when the delete form calls the function, the algorithm identifies the items by product id and passes the condition to the delete function. At the end of the operation the user is redirected to the main page of the application using the "Set HTTP Redirect.vi" function.

3. Conclusions and further developments

I believe that product management systems are an essential component in streamlining and automating production, processing and other processes. The importance of these systems is that they bring together information from all areas of product development.

In order to make the application as useful and powerful as possible, functionality to update product data within the application will be implemented in the coming period, as well as functionality to determine the availability of components and resources. Another option that can be added to this application is related to the possibility of accessing and uploading component working drawings, product overview drawings and appliance datasheets.

4. Bibliography

[1] X. W. Xu and T. Liu, "A web-enabled PDM system in a collaborative design environment," *Robotics and Computer-Integrated Manufacturing*, vol. 19, no. 4, pp. 315-328, Aug. 2003, doi: 10.1016/S0736-5845(02)00082-0.

[2] L. Alboaie and S.-C. Buraga, "Web services : basic concepts and implementations," 2006.

[3] T. Kgil and T. Mudge, "FlashCache: A NAND flash memory file cache for low power web servers," CASES 2006: International Conference on Compilers, Architecture and Synthesis for Embedded Systems, pp. 103-112, 2006, doi: 10.1145/1176760.1176774.

[4] John Essick, Hands-On Introduction to LabVIEW for Scientists and Engineers. Oxford University Press, 2018.