

# DESIGNING AN ALGORITHM AND A COMPUTER APPLICATION FOR RECORDING A TEAM'S PROGRESS IN COMPLETING A DISSERTATION WORK

RĂDULESCU Nicoleta-Cristina, SAVU Tom

Faculty of Industrial Engineering and Robotics, Study program: Applied Computer Science in Industrial Engineering, University POLITEHNICA of Bucharest, Year of Study: 4, e-mail: cristina.radulescu2000@gmail.com

*SUMMARY: The Dissertation Team Record software application allows users to enter, modify and view information required for dissertation management. It is a software application that runs in a web browser, which extracts or enters the necessary information into text files. It allows the user to perform actions such as: inserting a table for each dissertation topic, adding students to the tables related to the chosen topics, editing the name of the topic or the coordinator and removing students from the tables with the submission of the dissertation, withdrawing from the master's program or their expulsion.*

*KEYWORDS: software application, web browser, VI.*

## 1. Introduction

Management web applications are software applications that run in a web browser and allow the user to enter, modify and access data in an efficient and structured way. Their purpose is to simplify and automate certain processes as well as to centralize and store data [1]. The Dissertation Record Web Application is a software application for teachers that allows them to enter and modify data as well as find and view the information they need right on a web browser. The home page of the software application (Figure 1) provides an overview of the existing dissertation topics and the student teams that have opted for that topic. Also, each theme name is accompanied by the name of the leading teaching staff. In addition to these, the main page also has buttons to help add or edit an already existing topic, buttons to allow students to be edited and/or moved to other text files, and a button to set examination boards.



The screenshot shows the home page of the software application. At the top left is a logo with the text '1941' and 'U.A.B.'. At the top right are two buttons: 'Adaugare tema' and 'Comisii'. The main heading is 'Teme disertatie Dept.TCM'. Below this, there are three sections, each representing a dissertation topic. Each section has a header row with 'Student', 'Specializare', 'Data sustinere', and 'Arhivare'. The first section is for 'tema23' by 'Dumitrescu', showing a student 'Enescu' with specialization 'IPFP' and date '20.06'. The second section is for 'tema07' by 'Dumitrescu', showing a student 'ION' with specialization 'IACC' and date '20.06'. The third section is for 'tema1' by 'Florea Madalin', showing two students: 'Gabi' with specialization 'IAAC' and date '20.06', and 'Popescu' with specialization 'FIIR' and date '20.06'. Each student entry has a set of action buttons: 'Promovat', 'Stergere', 'Mutare', and 'Editare'.

Student	Specializare	Data sustinere	Arhivare
Enescu	IPFP	20.06	Promovat Stergere Mutare Editare
ION	IACC	20.06	Promovat Stergere Mutare Editare
Gabi	IAAC	20.06	Promovat Stergere Mutare Editare
Popescu	FIIR	20.06	Promovat Stergere Mutare Editare

Fig. 1 The home page of the software application [2]

The graphical interface of the application was created using the HTML programming language. To improve the appearance of web pages, css files were also used, which were inserted into the

documents with the html extension using the "link" element in the "head" section [3]. LabView was used to find, update, insert and delete data. All this data is stored in three text files. (*Echipe.txt*, *Studenti.txt*, *Persoane.txt*, *Promovat.txt*, *Ștergere.txt*) The Tab delimiter was used in all three text files. The file *Team.txt* contains the theme ID, the teacher ID and the name of the theme. The ID of the teaching staff can also be found in the *Teachers Profesori.txt* file next to the name of the teaching staff, and the ID of the topic can be found in the *Studenti.txt* file together with the student's specialization and his name. This structure of the text files was chosen in order to be able to interconnect them and allow the LabView software to retrieve and process data from all three files simultaneously. In the file *Passed.txt* are moved the students who submitted the dissertation and passed it, and in the file *Delete.txt* are transferred the students who either did not submit the dissertation or did not pass it. When these buttons are accessed, the respective student is deleted from the *Students.txt* file and added to one of the two files (*Promovat.txt*, *Ștergere.txt*).

## 2. Current status

In order to create the software application for the record of the dissertation topics, it was established that nine VIs are needed that are connected to each other according to the diagram shown in figure 2. Also in the diagram, the VIs that were realized are marked with yellow until the current stage of the work.

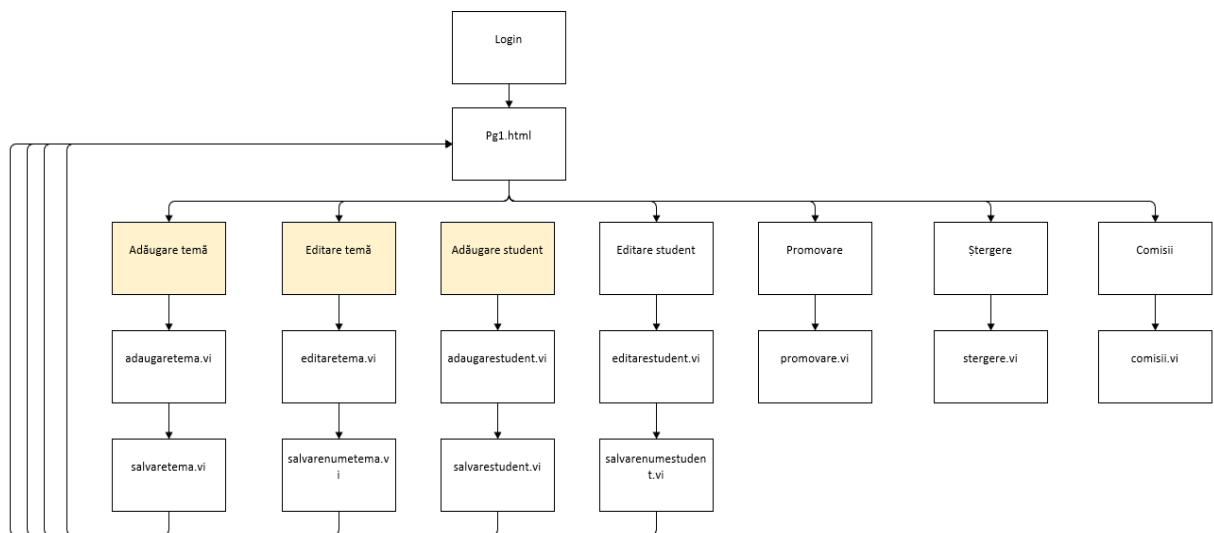


Fig 2. The diagram

The VI used to start the application is „*Adăugaretema.vi*” (Figure 3) and is a VI that is based on the html page named "Pg1.html". Its role is to insert a table for each assigned topic as well as to populate it with the students who will perform the respective topic. In order to achieve this, the html page named "Pg1.html" was divided into four smaller html pages. The first includes the introduction, namely: the logo, the title and the two buttons for adding the theme and commissions that were displayed in the browser by reading the page and displaying the "*Pg1\_introducere.html*" page with the commands "Read from Text File" and "Write response.vi". It was done this way because this entry only needs to appear once and not every time a new theme ID is found in the text file "*Echipe.txt*" and a new table is added. In the FOR structure, the rest of the html pages that make up the big page "pg1.html" as well as the text files "*Echipe.txt*", "*Persoane.txt*" and "*Studenti.txt*" were read. The text files were read with the "Read Delimited Spreadsheet.vi" function. The required columns were extracted from text files using the commands "Index Array", "Search 1D Array" to get the index of the line on which the searched text is located and "Index Array" to search on the previously found line and the value located on another

column. The values thus obtained were used to replace the predefined values in the html page ("##ID##", "##Tema##", "##Indr##").

This was done by using the "Search and Replace String" function. After running the program, it was noticed that the lines containing the theme ID, the manager and the name of the theme appeared in the browser by correlating the data read from the text files. In order to insert one table at a time and populate it with the related students, a new FOR loop was introduced in the "Studenti.vi" program together with a CASE type structure inside the first FOR loop. Inside the CASE structure, two "Search and replace" functions were used to replace the tags "##Stud##" and "##Spec##" with the values extracted from the text files. The values thus obtained were concatenated with the "Concatenate strings" function and sent to "Write Respinse.vi" to be displayed in the web page.

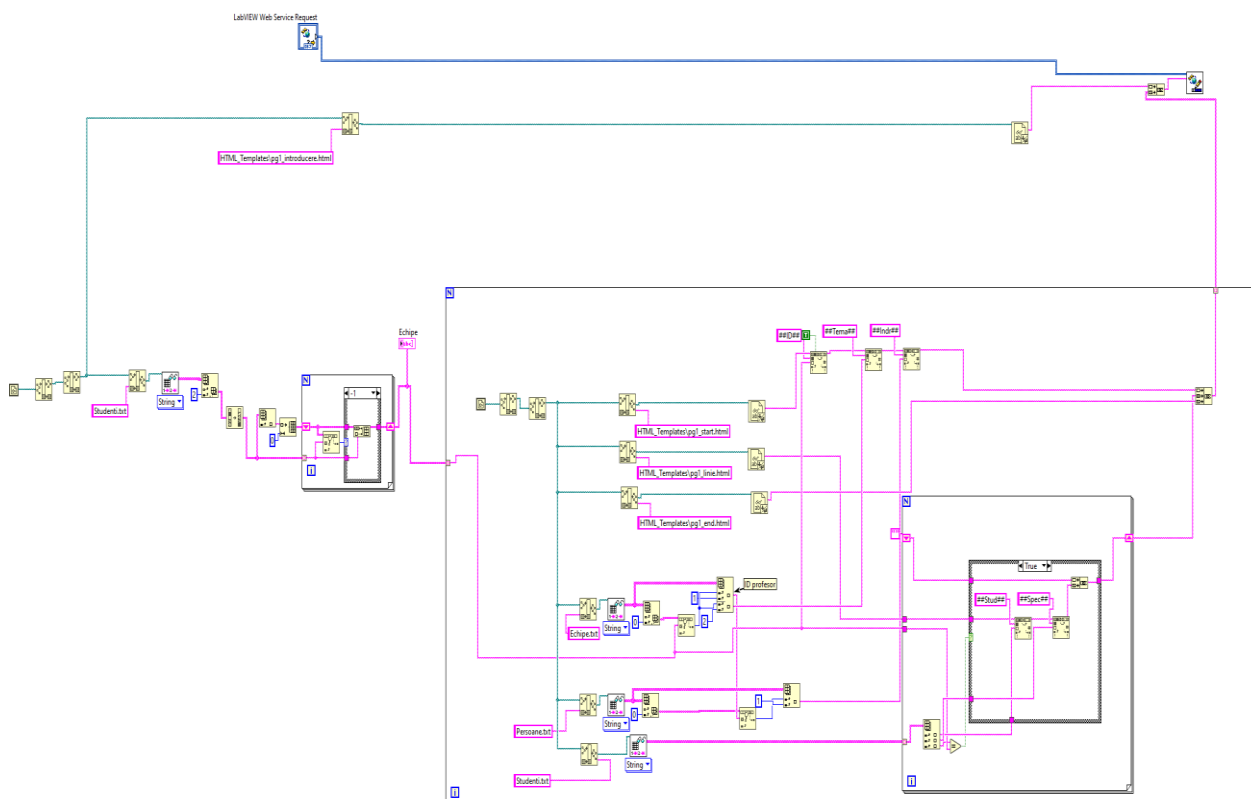


Fig. 3. Adaugaretema.vi [4]

At the next stage, the VI was made for adding a new theme. In the HTML page „*Adaugaretema.HTML*” fields were created in which the user will enter the ID of the theme and its name, as well as a list box from which the user can choose the leader of the team that will create the theme. For this I created a VI called "adaugarebeltema.vi" which I set to the POST method to be able to insert the data read from the web page into the text file "Echipe.txt". Figure 4 shows the functions that were used to design the algorithm for adding a new theme, namely: "Current VI's Path", "Strip Path", "Build Path" and "Read from Text File" to display in web page HTML document. The "Echipe.txt" file was also read with these functions. With the "Read Form Data.vi" function, where a constant was created in which the tag indicated in the HTML page was inserted, the data entered by the user in the web page in the three fields (ID, theme name and leader) were read. This data was entered into the text file with the "Write to Text File" function. In order for the entered data to respect the file format, functions were also used to insert the tab delimiter after each written text, as well as a function to move to the next line every time a new theme is entered. In order for all the data taken from the web page to be inserted into the text file, functions such as: "Open/Create/Replace File", "Set File Position", and "Close File" were used.

The VI for adding students was created similarly to the one for adding assignments. To enter a student, the user must select from the list box the names of the student he wants to add, select his degree and enter the ID of the topic he wants to add. To create this VI, the html page "*adaugastudenti.html*" was read, the data entered by the user in the web page with the same functions that are presented in figure 4 were taken and entered in the text file "*Studenti.txt*" using the same functions as in the Add Theme VI.

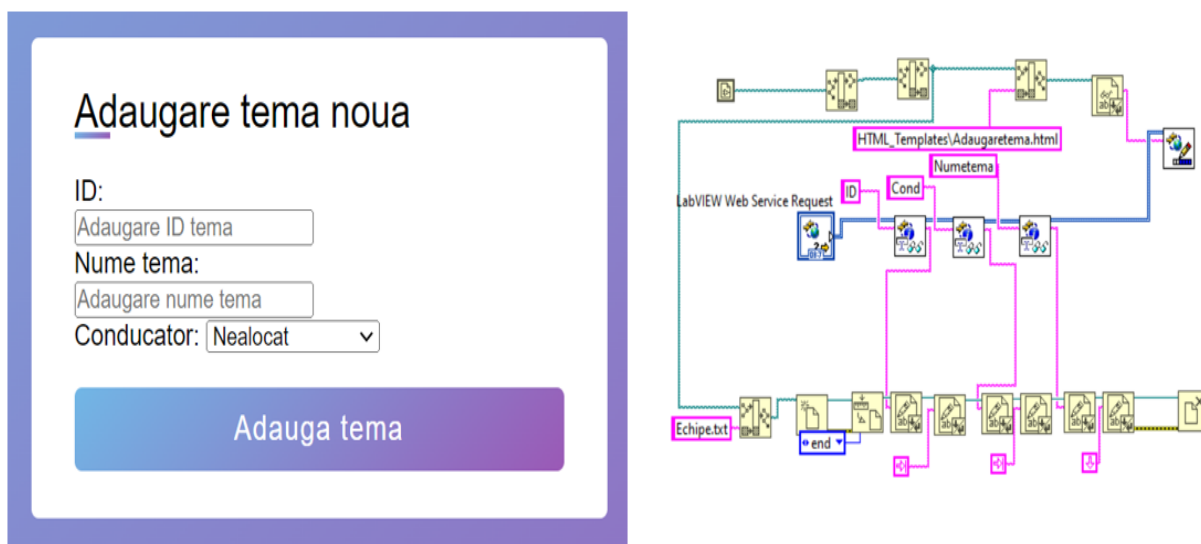


Fig. 4. The functions that were used to design the algorithm for adding a new theme [4]

### 3. Conclusions

In the future, more VIs will be added to the web service that will be able to add the correct topic of dissertation work to the right student to the file with the students who took the bachelor's degree or to the file with the students who withdrew from the master's degree. Another VI that will be created will be the one for editing students, a VI in which it will be possible to modify its topic and enter the date on which the dissertation exam was taken. There will also be a page where review boards can be assigned, as well as a VI to allow changing the topic name or tutor.

### 4. References

- [1].: <https://cmevo.com/ro/blog/ce-este-o-aplicatie-web/>
- [2].: [https://www.w3schools.com/html/html\\_entities.asp](https://www.w3schools.com/html/html_entities.asp)
- [3].: [https://www.w3schools.com/css/css\\_howto.asp](https://www.w3schools.com/css/css_howto.asp)
- [4].: [https://www.ni.com/docs/en-US/bundle/labview/page/glang/programming\\_pal.html](https://www.ni.com/docs/en-US/bundle/labview/page/glang/programming_pal.html)