Sinteza
Međunarodna naučna konferencija Univerziteta Singidunum

Impact of Internet on Business Activities in Serbia and Worldwide
Uticaj Interneta na poslovanje u Srbiji i svetu

DOI: 10.15308/SINTEZA-2014-969-974

DEVELOPMENT OF AN APPLICATION FOR AUTOMATIZATION OF WAREHOUSE OPERATIONS

Zoran Nešić[1], Nebojša Denić[2], Miroslav Radojičić[1], Jasmina Vesić Vasović[1]

[1]University of Kragujevac, Faculty of Technical Sciences, Čačak, Serbia

[2]Alfa University, Faculty of Information Technology, Belgrade, Serbia

Abstract:
In this paper is presented an approach for development of the application for automation of warehouse operations. The paper presents the most important elements of this issue, which are based on the analysis of the relational database model, development of software solutions, analysis of information system. The paper represents a methodological review of the computer support for automation of warehouse operations.

Key words:
Application Software, Warehouse Operations, Information Systems.

INTRODUCTION

Under the economically conditions of the the global crisis, companies today must rapidly adapt to changes and new situation, which appear on the market. In this regard, investment in information technology (IT) is no longer an investment in the future but the present. The development of the IT sector provides the development of the whole society, as in the current conditions of particular importance to business systems and enterprises in Serbia. In one of many typical companies in Serbia for the sale and distribution of automobile spare parts, "Simonida-Gracanica", is explored the possibility of modernizing the business through a methodological review of computer support for automation of warehouse business. The emphasis is placed on improving the business and achieving positive business results.

The problem of warehouse business is considered by many authors, which to this segment gives great importance. One approach to integrated consideration of a warehouse business Van Den Berg (2007) analyzing warehouse management systems and effective warehouse management [1]. Richards (2011) analyze all problems of warehouse operations in terms of improving efficiency and minimizing costs [2]. Jiffry (2012) emphasis placed on consideration of improvement of space utilization [3]. However, application of information technology and computer support also represents a significant element of the warehouse operation improvement [4] [6].

Modern aspects of improvement warehouse operations include a number of different directions:

- Warehouse redesign [7]
- Warehouse simplification and automation [8]
- Warehouse distribution system [9]
- Building of warehouses information [10]
- Warehouse operations planning system [11]
- Integrated framework for warehouse management [12]

In this paper, the aspect of improving warehouse operations is placed on the application of information technology for the automation of operations [13]-[15].

RELATIONAL DATABASE MODEL

In order to develop a software application it is performed a detailed analysis of business processes in the enterprise. By the analysis of the needs in this company was established that there is an obvious need, in addition to records of data about cars, customers and their transactions, for the realization of applications to control inventory of spare parts in the warehouse. For the creation a database was used MICROSOFT SQL SERVER MANAGEMENT STUDIO EXPRESS 2008 R2.

In the database Cars, will be used four tables for operation with records of equipment, which are: article, customers, input, exit. The relational database model is shown in Fig. 1.

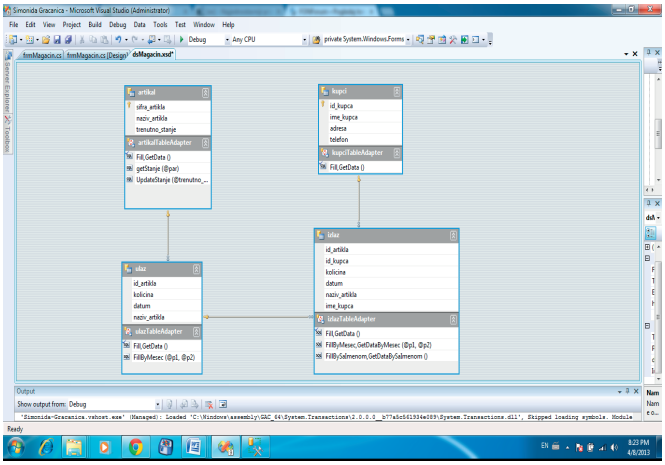
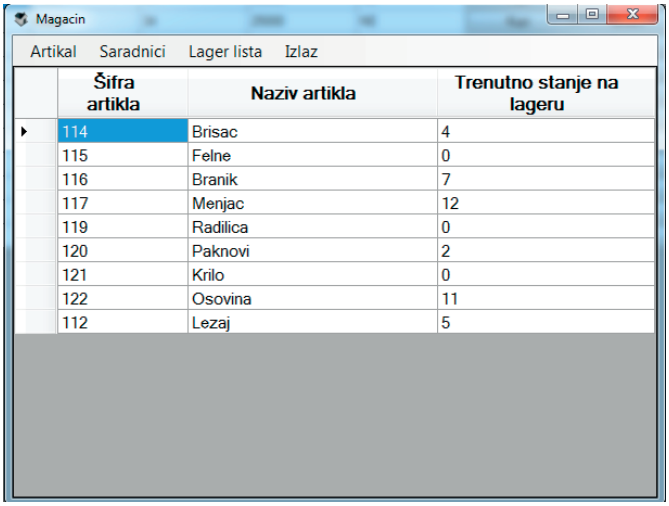


Fig. 1. The relational database model



Šifra artikla	Naziv artikla	Trenutno stanje na lageru
114	Brisac	4
115	Felne	0
116	Branik	7
117	Menjac	12
119	Radilica	0
120	Paknovi	2
121	Krilo	0
122	Osovina	11
112	Lezaj	5

Fig. 2. The initial form of application solution

The tables have the following structure:

article – The table that contains information about the items with fields:

- id article, the identification number of product, type *string*, and also the primary key of the table
- Product Name, some article name, type string
- current state, the quantitative stock items in a warehouse, type float

customers – The table containing information about customers with fields:

- id customer, identification number of customer, type integer, primary key of the table
- customer name
- address
- phone

input – The table containing the information about the date of arrival of received goods and the amount, with fields:

- id item, external key of the table
- amount, the amount of items which enters the warehouse
- date, the date of arrival of a particular item

exit - The table containing the information about the delivery of goods to customers or distributors, with fields:

- id item, external key of the table
- id customer, identification of the customer
- amount, the amount of items coming out of the warehouse
- date, the date of arrival of a particular item

REVIEW OF THE SOFTWARE SOLUTION

The information solution which is implemented in the "Microsoft Visual Studio 2008" program allows the user to perform basic interaction with the database. These are the following options:

- Record, edit and delete data about spare parts
- Record, edit and delete data about clients, customers, distributors
- Recording of the inputs and outputs of goods from the warehouse

A key element in the development of the application is a client application for connection to the databaseThe application is required to add *Dataset* which establishes a connection to the database. By activating *->Add->New Item->DataSet* is added the *DataSet*, which is later converted into standardized, by dragging the desired database from the Server Explorer window.

In the scheme of "*DataSet*" are added methods for each table that already contain SQL queries, with parameters depending on the table and which, if necessary, a call without having to define each tname the SQL statement in the program.

The application consists of a number of forms:

- *Warehouse* - a form gives the user the ability to review all spare parts in the database and to open any other application forms. This functionality is implemented using a "*DataGridView*" control
- *Article Record* - a form gives the user the ability to add new items to the base
- *Article Editing* - a form provides the user the ability to modify the data of an item
- *Article Deletion* - a form gives the user the option of deleting items
- *Customer Input* - a form gives the user the option of adding a new customer in the database
- *Customers Edition* - a form gives the user the ability to edit customer data
- *Customers Deletion* - a form gives the user the option of deleting in customer database
- *Customer List* - a form gives the user the ability to view a list of all customers
- *Sale Check* - a form gives the user access to the list of delivered items, quantity, customer, date, monthly filtering and deletion of selected items from the list
- *Date of entry* - a form gives the user access to the list of submitted items, quantity, date, monthly filtering and deletion of selected items from the list
- *Adding* - a form provides the user select the desired item and add it to the balance in a warehouse, at the selected amount

- *Sale* - a form gives the user the ability to input of data delivery and data entry about customer, product, and quantity per unit

Form Warehouse

When starting the program calls the constructor form frmWarehouse in which the method is called:

this.articleTableAdapter.Fill(this.dsWarehouse.article);

which is used to fill a certain table of the Data Set (in this case the article-) by data from the database. The main control on the form is "*DataGridView*" which use the above-mentioned table "*DataSet*" as a data source. Fig. 2 shows the initial form of the application solution.

In addition to control "*datagridview*" on the main form is used and control "*menustrip*" as a connection with other forms of the application. The control menustrip by activating a menu item, opens the required form, in the case of form "*frmInputArticle*" the code is following:

frmInputArticle frm = newfrmInputArticle();

frm.Show();

The first command creates a new instance of the class *frmInputArticle* and the next is showing in the focus.

When starting specific items on the menu are invited a pre-defined methods that create objects, e.g. certain forms for Input Article, the method is as follows:

```
privatevoid InputToolStripMenuItem_Click(object sender, EventArgs e)
 {
frmInputArticle frm = newfrmInputArticle();
 frm.Show();
 frm.FormClosing += newFormClosingEventHandler(frm_FormClosing);
 }
```

The last command adds EventHandler *FormClosing* which is common to all forms and is used to update the data from the database in the "DataSet" after closing each form:

```
void frm_FormClosing(object sender, FormClosingEventArgs e)
 {
this.articleTableAdapter.Fill(this.dsWarehouse.article);
 }
```

Article Record

Form *FrmInputArticle.cs* displayed when is activated menu item, "*Article*" and then "*Input*". Form contains two controls "*Textbox*", and control "*Button*", and field for Input code of *Article* and the name of *Article*, as well as the option which call the appropriate method. The method checks whether the conditions are met, or of is the entered text in both fields for *Input*. After that, verifies if code already exists in the database. It is checked in catch block:

```
catch (Exception ex)
 {
if (ex.Message.StartsWith("Violation of PRIMARY KEY constraint 'PK_article'")) { MessageBox.Show("Code already exists in the database "); return; }
MessageBox.Show(ex.Message.ToString()),
 }
```

There are also checks to see if there might be another type of exception and has to notify the appropriate user by MessageBox.

Article Edition

The form contains two text fields for input, a "*combobox*" and a button. The value that appears in this case is a name of the *Article*, and "*ValueMember*" and the value of the field is a code of *Article*. In this way is later passed parameter to SQL query for making changes to a specific *Article*.

After input and activation of "Edit article" performs a method *UpdateArticle* of a *TableAdapter*. To it is passed the values of text fields as parameters and „*ValueMember*" value of "*ComboBox*", in order to reach a single Article that we want to change.

SQL query contained in the method *articleTableAdapter.Update*:

```
UPDATE article
SET code_Article = @code_Article, name_Article = @name_Article, current_Status = @current_Status
WHERE (code_Article = @Original_code_Article)
```

Article Deletion

The form contains only one ComboBox and the button. ComboBox contains all values of article names. As in the previous form, after selecting a specific Article and activate the "Delete article" shall be called SQL command by the *articleTableAdapter*:

```
DELETE FROM article
WHERE (code_Article = @Original_code_Article)
```

„@Original_code_Article" in this case is a parameter which receives a value from the field "ValueMember" of control „ComboBox1", or transmission of code of Article that shall be deleted.

Customer Input

Form Input customer / associate contains three text fields and the button for input. During the activation of the input is checked input data. If this condition is satisfied, calls a method Insert which is part of *customersTableAdapter* and forward it the parameters given in the text fields. The content of the SQL command is as follows:

```
INSERT INTO customers
 (name_customer, address, phone)
VALUES (@name_customer,@address,@phone)
```

If in the catch block is caught the exception, writes an

error message. The variable p, type integer,. is introduced as the value returned by the command *customersTableAdapter. Insert.* Refers to the number of rows in the database that have been added. In case of a successful input into the database, this value will be 1 and by checking the values we know that the input is completed. This is signaled by the MessageBox with the text "Successful input of associates. \ N Do you want to continue with input of associates?". If the answer is positive, from the text boxes are cleared previously entered values and provides the possibility of entering the data on a new associate. If the answer is negative a form closes.

Customers Edition

The form contains control "*ComboBox1*", three text fields for input and the option to confirm the changes. In the control "*ComboBox1*" is contained a list of all employees and allows the selection of one whose data should be changed. After entering all the changed data and activation of the "Edit of associates", it is calling a method "*customersTableAdapter. Update*" to which parameters are passed from the form. Command returns the number of rows modified in the database. If the value is "1" it means that the update was successful. After that the user is informed by the appropriate "*Messagebox*". The call of a command „*customersTableAdapter. Update*" is performed in try-catch block, for processing of possible exceptions that can occur most often if the application has a problem at connecting to the database. Message on the type of exception is also written in the control "*MessageBox*".

Customers Deletion

The form contains "*ComboBox1*" in which is selected an associate whose data is to be deleted from the database. It is calling the following command:

customersTableAdapter.Delete(int.Parse(comboBox1.SelectedValue.ToString()))

Predefined text of SQL statement, contained in the preceding command, is:

DELETE FROM customers

WHERE (id_customer = @Original_id_customer)

After changing the value of the control, it is calling EventHandler for the event „OnValueChanged" with the method "fill()":

```
privatevoid fill() {
DateTname d = dateTnamePicker1.Value;
int f = 0;
 f = d.Month + 1;
if (f == 13) f = 1;
DateTname d1 = newDateTname(d.Year, d.Month, 1);
DateTname d2 = newDateTname(d.Year, f, 1);
this.exitTableAdapter.FillByMonth(dsWarehouse.exit, d1, d2),
 }
```

The value of control "*DateTnamePicker*" takes the variable d, type *DateTname,* which subsequently is used to filter the data in the table input, which satisfy the given condition.

The control "datagridview" provides detailed information after filtering by the command:

this.exitTableAdapter.FillByMonth(dsWarehouse.exit, d1, d2);

The parameters of this method are presented in Table Dataset *dsWarehouse.exit* which is filled with data from the database and the first and last day of the specified month.

The current form provides an opportunity for the user to delete the selected row in the control, by activating the option "Delete the selected records." Code which follows takes the data from the selected row and parse them into the corresponding inputs of predefined command for deleting. The data on the input of a given amount is then subtracted from the value of "status" of the table "article" in the database, by *articleTableAdapter* and associated command "*UpdateStatus*".

```
string idArticle=(dataGridView1.SelectedCells[0].Value.ToString());
string name = (dataGridView1.SelectedCells[1].Value.ToString());
float amount = float.Parse(dataGridView1.SelectedCells[2].Value.ToString());
string article = dataGridView1.SelectedCells[3].Value.ToString();
DateTname d = DateTname.Parse(dataGridView1.SelectedCells[4].Value.ToString());
int idcustomer=int.Parse(dataGridView1.SelectedCells[5].Value.ToString());
if (exitTableAdapter.Delete(idArticle, idcustomer, amount , d.Date) == 1)
 {
float tren = (float)articleTableAdapter.getStatus(idArticle);
 articleTableAdapter.UpdateStatus(tren + amount , idArticle);
MessageBox.Show("Successful deletion "); fill() ;
 }
```

After a successful deletion are automatically update and display the data in the control "datagridview" and writes the message on successful deletion.

Date of entry

The form used to display information about the items entered monthly is shown in Figure 3

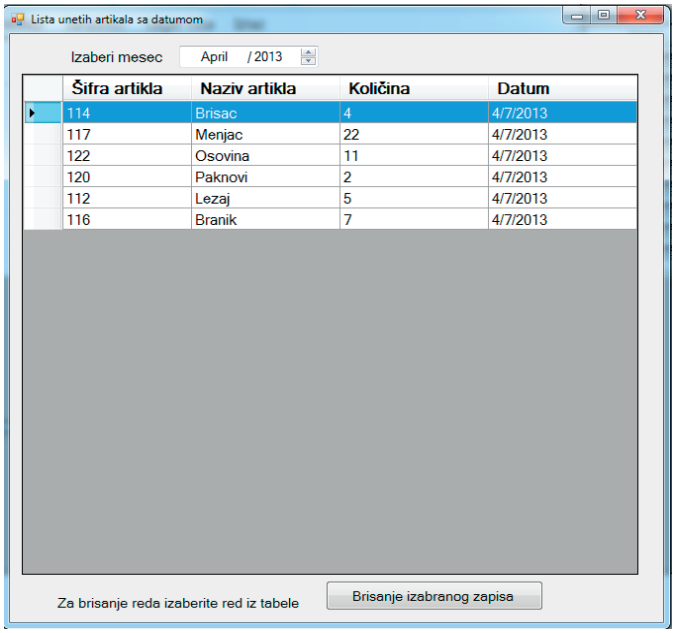


Fig. 3. The data of articles

This form contains controls "*DateTnamePicker*", "*DataGridView*" and the button "delete selected records". In the control "*DateTnamePicker*" is selected Month and year in which want to review the entered article. After changing the value of the control, it is calling EventHandler for the event "OnValueChanged" with the method "*fill()*":

```
privatevoid fill()
 {
DateTname d = dateTnamePicker1.Value;
int f = 0;
 f = d.Month + 1;
if (f == 13) f = 1;
DateTname d1 = newDateTname(d.Year, d.Month, 1);
DateTname d2 = newDateTname(d.Year, f, 1);
this.inputTableAdapter.FillByMonth(dsWarehouse.
input, d1, d2);
 }
```

The value of control "*DateTnamePicker*" takes the variable d, type DateTname, which later is used for filtering the data in the table Input, which satisfy a given condition.

The control "datagridview" provides detailed information after filtering by the command

```
this.inputTableAdapter.FillByMonth(dsWarehouse.
input, d1, d2);
```

The parameters of this method are DataSet table dsWarehouse.input, which is filled with the data from the database and the first and last day of the specified month.

The current form provides an opportunity for to the user and to delete the selected row in the control by activating the option "Delete the selected records." Code which follows takes the data from the selected row, parse them into appropriate input values of the predefined command for deleting. The data of the input, of a given amount, is then subtracted from the value of "Status" of the table "Article" in the database, by *articleTableAdapter* and associated command "UpdateStatus".

```
string idArticle = (dataGridView1.SelectedCells[0].
Value.ToString());
float amount = float.Parse(dataGridView1.Selected-
Cells[2].Value.ToString());
DateTname d = DateTname.Parse(dataGridView1.Se-
lectedCells[3].Value.ToString());
if (inputTableAdapter.Delete(idArticle, amount ,
d.Date) == 1)
 {
float tren=(float)articleTableAdapter.
getStatus(idArticle);
 articleTableAdapter.UpdateStatus(tren - amount ,
idArticle);
MessageBox.Show("Uspešno brisanje"); fill();
 }
```

After a successful deletion, is automatically update and display the data in the control "*datagridview*" and writes the message on successful deletion.

ADDING

The form provides the user selection of desired article and addition to the Status in Warehouse at the selected amount.

The form is called by activating options *LagerLista->Dodavanje* or by the function key F5. On form there are three controls: "*combobox*", "*textbox*" and a control of type "*button*". The data source of the control "combobox" is the table article, through which is selected the desired code of article, for update. It is used two controls of "TableAdapter ": to access the data from the table "Article" and update the field "current_Status", and the other is used to add a new record to the table" input. After selection of article code, below the text "Article name" it is written the corresponding name of article from the database. By starting the option "Add to stock list", by the command "*articleTableAdapter.GetStatus*" is itaked the the status of requested article and stored in a variable *t_Status*, type *float*. On that value is added the contents of text of the *txtAmount*, previously converted to type float. Finally, the changed value of the variable is again entered into the database, as well as a new record in the table "Input" with all the values.

SALE

The form provides a selection of article the user, input of quantity and customer who will make takeover of a given amount of selected article. The form is called by activating option *StockList-> Sale* or by the function key F8. The form consists of the following controls: two "*Combobox*", the one "*textbox*" and a control of type "*button*". The data source of the control "*cmbCustomer*" is table Customer by which is selected customer code. This value is input to the table "exit" with article code and quantity. The data source of the control "*cmbArticle*" is the table *Article*, through which is selected the desired code of article. Then the value of the field "Status" in the table *Article* is reduced by the entered amount .

After selection of article code, below the text "Article name" is written the corresponding name of article from the database. After activating the option "DELIVER", by the command "*articleTableAdapter.GetStatus*", is itaked the status of requested article and stored in a variable t_ Status, type *float*. On that value is added the text of the *txtAmount*, previously converted to the type *float*. Finally, the changed value of the variable is again entered into the database, as well as a new record in the table "exit" with all related values.

CONCLUSION

The advantage of the present application for support to warehouse operations of the company "Simonida" Gracanica is that the process of entering, editing, deleting and searching is very simple and provides quick and easy use of the program. The whole concept of realization of operations in the company for the sale and distribution of automobile spare parts and equipment "Simonida" Gracanica, by applying above presented tools and procedures ensuring the adequacy, reliability, efficiency, accuracy and specificity.

By using the advantages provided by the Visual Studio 2008, in a simple and easy way can be created a dynamic application, which has excellent interaction with the end user. Based on the above presented, we can see that this software and IT solution can be used by all of the organizational structure of a company of any profile, not only commercial enterprises, but also production and other. It can be seen that the software package is widely conceived that each user can successfully use it in their domain of work. As already mentioned, the user interface has been carefully designed and easy to use. Most importantly, every user used to work in Office environment should be easy to get used to working in a new environment and quickly become familiar with all the options which this software package provides.

Acknowledgment

Research presented in this paper was supported by Ministry of Education and Science of the Republic of Serbia, Grant III-44010, Title: Intelligent Systems for Software Product Development and Business Support based on Models.

REFERENCES

[1] Van Den Berg J. P. (2007), Integral Warehouse Management: The Next Generation in Transparency, Collaboration and Warehouse Management Systems, Management Outlook, cop., Utrecht.

[2] Richards G. (2011), Warehouse Management: A Complete Guide to Improving Efficiency and Minimizing Costs in the Modern Warehouse, Kogan Page Publishers, London.

[3] Jiffry S. A. (2012), Effective Warehouse Management: Improvement of Space Utilization & Trace-ability by Adopting a Floating Location Warehouse, LAP Lambert Academic Publishing.

[4] Hompel M. T. and T. Schmidt (2006), Warehouse Management: Automation and Organisation of Warehouse and Order Picking Systems, Springer, Berlin ; New York.

[5] Obal P. D. (2007), Premier List of Warehousing Software and Warehouse Management Systems, Industrial Data & Information, Tulsa, Okla.

[6] Obal P. (2007), Selecting Warehouse Software from WMS and ERP Providers, Industrial Data & Information, Tulsa, Okla.

[7] R. Buil, M. A. Piera, "Warehouse redesign to satisfy tight supply chain management constraints", Transactions on Information Science and Applications, Vol. 5, No. 3, pp. 286-291, March 2008.

[8] B. He et al., "BIwTL: a business information warehouse toolkit and language for warehousing simplification and automation", Proceedings of the 2007 ACM SIGMOD international conference on Management of data SIGMOD '07, Beijing, China, 11-14. June 2007, pp. 1041-1052.

[9] L. B. Schwarz, J. E. Ward and X. Zhai, "On the Interactions Between Routing and Inventory-Management Policies in a One-Warehouse N-Retailer Distribution System", Manufacturing & Service Operations Management, Vol. 8, No. 3, pp. 253-272, Summer 2006.

[10] A. Laha, "On the issues of building information warehouses", Proceedings of the Third Annual ACM Bangalore Conference COMPUTE '10, New York, 2010, Article No. 2.

[11] T. C. Poon et al., "A real-tname warehouse operations planning system for small batch replenishment problems in production environment", Expert Systems with Applications: An International Journal, Vol. 38, No. 7, pp. 8524-8537, July, 2011.

[12] P. Ahluwalia and A. Ramachandran, "An integrated framework for warehouse management using wireless sensor networks and decision support systems", Proceedings of the 1st International Conference on Wireless Technologies for Humanitarian Relief ACWR '11, Kerala, India, 18-21 December 2011, pp. 23-27.

[13] Ray A. K. and T. Acharya (2004), Information Technology: Principles and Applications, Prentice-Hall of India, New Delhi.

[14] Kirk A. (2009), Information Technology, Ferguson, New York.

[15] Rajaraman V. (2003), Introduction to Information Technology, Prentice-Hall of India, New Delhi.