Journal of Applied Business Information Systems

http://www.jabis.ro

Security optimization of a distributed application with free access to resources

Mihai DOINEA*

* Informatics Economic Department, Academy of Economic Studies, Bucharest, Romania

Abstract: *The paper aims to identified and use the main security features for a distributed applications and optimize the overall security level. The distributed framework with all its characteristics is presented and the optimization objectives are defined according to the framework particularity. A security architecture is built emphasizing the components that participate to the protection of the distributed system. Each one of the security control implemented into the application is described, highlighting an optimization process for the entire security level.*

Keywords: *distributed applications, security, resources, free access, optimization, methods.*

1. Distributed framework

The application for daily calorie consumption is meant to give help to individuals who want to regularly check their calorie schedule, building for them menus based on the intensity effort, daily set it.

The Secure Distributed Application for Calorie Optimization framework, following referred as SDACO, is built on several layers having a three-tier distributed architecture [1]. The three layers are represented by:

- presentation layer- is given by the interface of the application designed to help users resolve their needs;
- database layer - is represented by a Oracle Database Scheme which stores all the application data;
- application layer - contains all the distributed application main code composed of C# classes.

Each one of the aforementioned layers combines resources in order to give trustful and reliable information for users, helping them to identify their current bodybuilding conformation and providing a list of possible menus dynamically configured based on the everyday level of effort.

The application's framework having the three layers mentioned above is presented in figure 1, [1]:

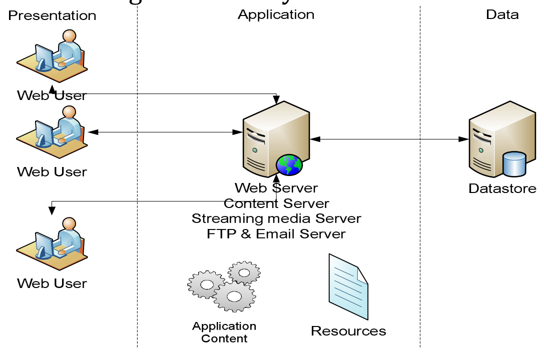


Figure 1. *The Three-Tier Web Based Distributed Application*

The presentation layer is described by the user's interface in which it can be entered the input data like:

- height and weight, expressed either in imperial or metric units;
- age in years;
- sex which will make the difference between several bodybuilders metrics;
- activity effort type expressed by a list of five different levels, each one giving examples of several activities for it.

The graph for the SDACO application with free access to resources is depicted in the figure 2, highlighting the possible connections between each application component. Users can access the following functions in a one way informational flow with the possibility to come back in different region of the process

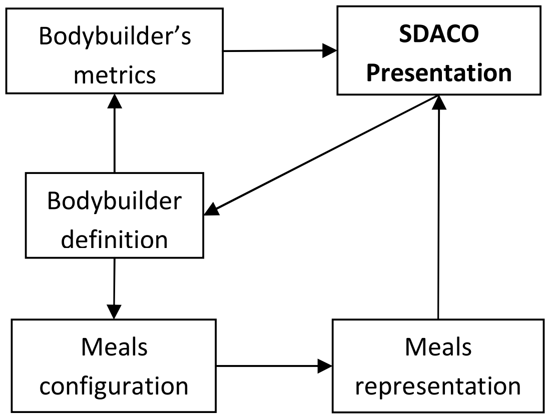
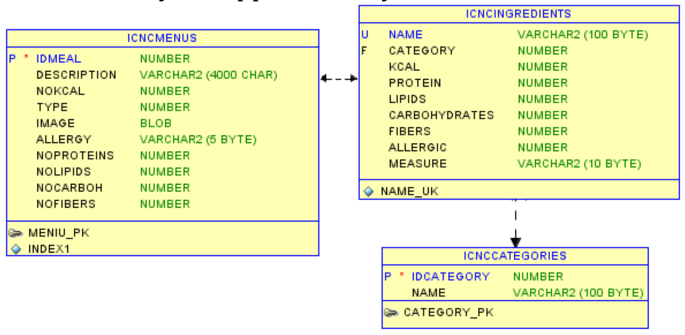


Figure 2. *Presentation Layer for SDACO*

The application layer is defined by the totality of the server-side and client-side source code which intermediates between user's actions and the database layer. The application layer is written using different programming and description languages which are cooperating in achieving the higher level of efficiency and performance, such as:

- C Sharp, C# implemented in the .Net Framework as a managed code language running under a CLR, Common Language Runtime virtual machine;
- AJAX for efficiently achieve all the asynchronous procedures defined in the application; AJAX stands for Asynchronous JavaScript and XML;
- JavaScript for successfully implementing client-side functions;
- HTML, a mark-up language over which ASPX is implemented;
- CSS, Cascading Style Sheets for managing the application's layout.

In the figure 3, it is represented a partial internal structure of the database, which can store this information being processed later by the application layer:



ICNCMENUS		
P *	IDMEAL	NUMBER
	DESCRIPTION	VARCHAR2 (4000 CHAR)
	NOKCAL	NUMBER
	TYPE	NUMBER
	IMAGE	BLOB
	ALLERGY	VARCHAR2 (5 BYTE)
	NOPROTEINS	NUMBER
	NOLIPIDS	NUMBER
	NOCARBOH	NUMBER
	NOFIBERS	NUMBER
	MENIU_PK	
	INDEX1	

ICNCINGREDIENTS		
U	NAME	VARCHAR2 (100 BYTE)
F	CATEGORY	NUMBER
	KCAL	NUMBER
	PROTEIN	NUMBER
	LIPIDS	NUMBER
	CARBOHYDRATES	NUMBER
	FIBERS	NUMBER
	ALLERGIC	NUMBER
	MEASURE	VARCHAR2 (10 BYTE)
	NAME_UK	

ICNCCATEGORIES		
P *	IDCATEGORY	NUMBER
	NAME	VARCHAR2 (100 BYTE)
	CATEGORY_PK	

Figure 3. *Tables description and relations*

The many to many relation presented in figure 3 between menus and ingredients table is divided into two one to many relation using a physical table stored on HDD which is uniquely assigned to menus table through ***description*** field, containing a list of ingredients uniquely identified in ingredients table through ***name*** field.

The application layer gives free access to the following functions for SDACO framework, with no means of recording the activity undertaken by users in the application:

- calculation of the following bodybuilding metrics **B**ody **M**ass **I**ndex, **B**asal **M**etabolic **R**ate, **F**at **M**ass **I**ndex, **L**ean **B**ody **M**ass and BMI classification level, revealing the status of a person: underweight, normal, overweight or obese, as calculate in [2,3]:
- calories consumption based on the aforementioned values and the level of effort intensity introduced by a user;
- menu configuration with three meals per day and a total amount of calories equal to the necessity level prior determined.

The application uses the following structure of object oriented classes, as presented in figure 4.

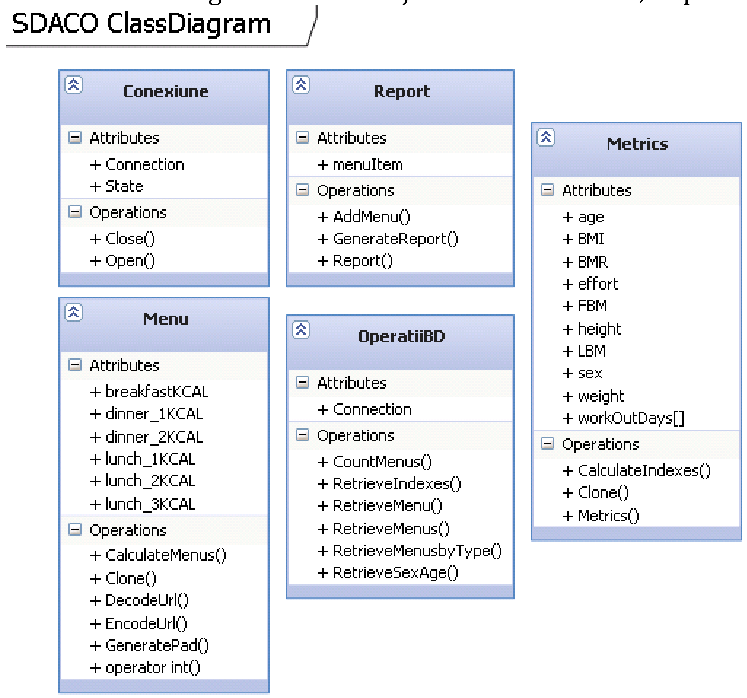
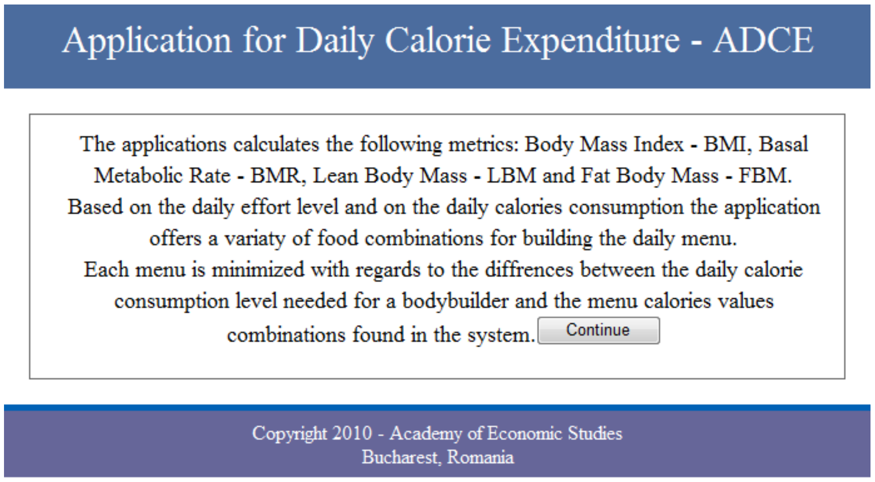


Figure 4. *SDACO Class Diagram Solution*

The classes presented in figure 4 provide all the results needed in order to have full and free access to the functions of the SDACO framework.

2. SDACO presentation

The application interface is easy to use and intuitive, easily giving access to all its functions, without any need of assistance for inexperienced users. The figure 5 is the main frame for the application which, as presented in figure 2, the graph associated with the application structure, gives access to the following functionalities.


Application for Daily Calorie Expenditure - ADCE

The applications calculates the following metrics: Body Mass Index - BMI, Basal Metabolic Rate - BMR, Lean Body Mass - LBM and Fat Body Mass - FBM.
Based on the daily effort level and on the daily calories consumption the application offers a variaty of food combinations for building the daily menu.
Each menu is minimized with regards to the diffrences between the daily calorie consumption level needed for a bodybuilder and the menu calories values combinations found in the system.

Copyright 2010 - Academy of Economic Studies
Bucharest, Romania

Figure 5. *SDACO main frame*

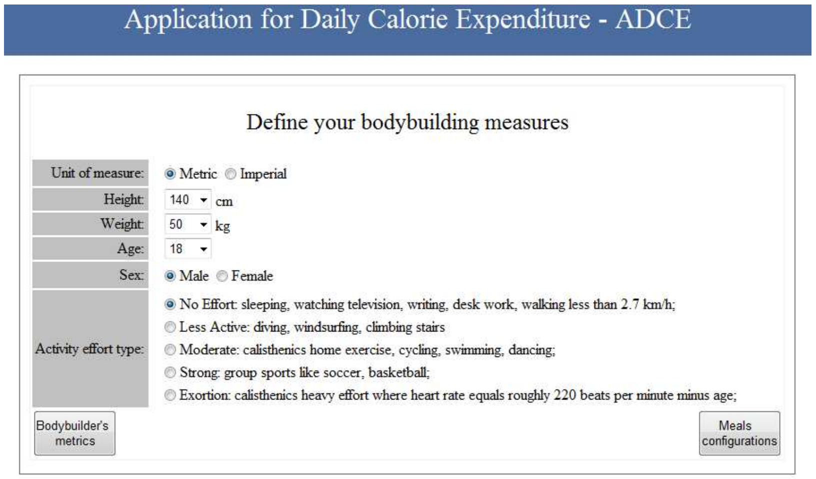
The second step of the application is the input interface, consisting in choosing the unit of measurement, metric or imperial, and several inputs like weight (Kg/lb) and height (cm/in) along with user's age and sex.

The effort correlation along with the daily calorie consumption is a complex bound, as presented in [4], in which many types of effort should be evaluated:

- physical effort is one of the most researched areas. Scientist, physicians, sportsmen and other interested parties had developed a whole system of indicators. Trying to get the best out of the human body, they measured calories consumption in human regular activities and training exercises and classified them by effort intensity;
- neurological effort has its word in the process of daily calorie consumption. One of the best practice examples is the student session period when majority students change their way of life for a couple of weeks. In this time, intensive cognitive effort is made which will request a high amount of proteins, which are the main resource for the brain to function at full capacity. Doing so, using more proteins as fuel for the thinking machine, will alter the established equilibrium, resulting in less food per capita per day. At the end of the students session period they will fill a change in their weight status in plus or in minus;
- emotional effort is also a direct factor which affects the daily calorie consumption and one of great importance. Emotional effort can lead people in or out of medical problems by influencing their meals.

The activity effort must also be entered, by choosing from a list of five options, each one specifying a level of intensity together with some examples of activities: no effort - sleeping, watching television, writing, desk work, walking less than 2.7 km/h; less active - diving, windsurfing or climbing stairs; moderate - calisthenics home exercise, cycling, swimming or dancing; strong - group sports like soccer, basketball; exhaustion - calisthenics heavy effort where heart rate equals roughly 220 beats per minute minus age. Those types of exercises are correlated to the level of effort that the body, as in [5].

In figure 6 is presented the frame of data input which uses the classes mentioned in the SDACO Class Diagram to provide the implemented functionalities.


Application for Daily Calorie Expenditure - ADCE

Define your bodybuilding measures

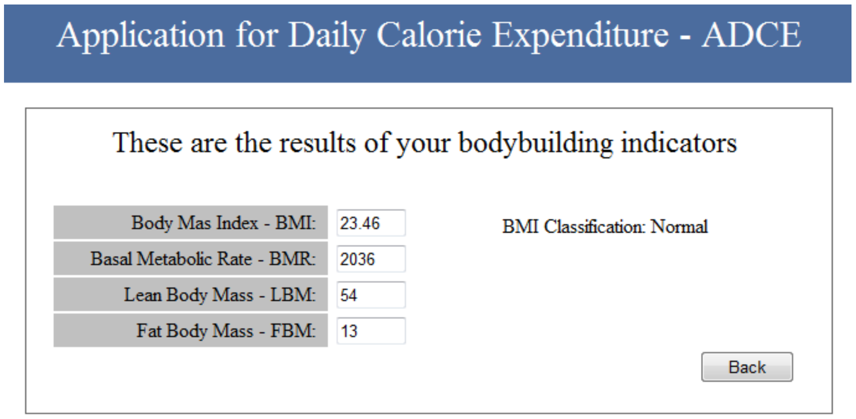
Unit of measure:	[✓ Metric] Imperial
Height:	140 cm
Weight:	50 kg
Age:	18
Sex:	[✓ Male] Female
Activity effort type:	[✓ No Effort: sleeping, watching television, writing, desk work, walking less than 2.7 km/h;] Less Active: diving, windsurfing, climbing stairs; Moderate: calisthenics home exercise, cycling, swimming, dancing; Strong: group sports like soccer, basketball; Exortion: calisthenics heavy effort where heart rate equals roughly 220 beats per minute minus age;

Figure 6. *SDACO input interface*

Based on the data entered in the data input interface, the application gives access to the bodybuilder's metrics interface in which he can see his bodybuilder's values for:

- BMI – Body Mass Index, is a statistical index which compares a person's weight and height; this index is relevant in estimating a healthy weight;
- BMR – Basal Metabolic Rate is the amount of energy expended while at rest in a neutrally temperate environment; this release of energy in this state is sufficient only for the functioning of the vital organs;
- FMI – Fat Mass Index is the amount of fat reported to the muscular mass and the total weight of a person;
- LBM – Lean Body Mass represents mass of the body without fats;
- BMI classification status.

The figure 7 depicts the bodybuilding indicators and the bodybuilder's classification based on the BMI value.


Application for Daily Calorie Expenditure - ADCE

These are the results of your bodybuilding indicators

Body Mas Index - BMI:	23.46
Basal Metabolic Rate - BMR:	2036
Lean Body Mass - LBM:	54
Fat Body Mass - FBM:	13

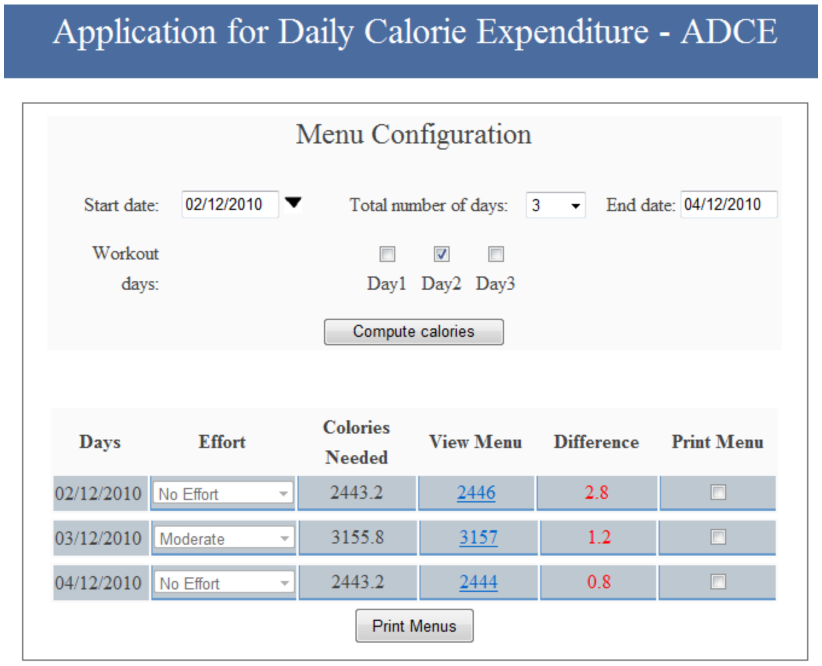
BMI Classification: Normal

Figure 7. *Bodybuilding indicators interface*

The following frame is accessed also from the data input interface, just like the previous one, and here users can customize their time interval for each they want to get a menu combination.

The time interval must be smaller than 7 days. The users can select the start date and the number of days for which they will get the meals combinations. The final date is automatically adjusted based on the input date and the total number of days.

For the entire time interval, a user can select which are the workouts days of that period in which he will exercise according to the effort intensity specifications given prior to this phase, on the data input interface. The other remaining days will be taken into consideration as being days in which the users is not exercising, and the meals combinations offered will be calculated with no effort activities parameter, as depicted in figure 8.



Application for Daily Calorie Expenditure - ADCE

Menu Configuration

Start date: 02/12/2010 Total number of days: 3 End date: 04/12/2010

Workout days: Day1 ☐ Day2 ☑ Day3 ☐

Compute calories

Days	Effort	Colories Needed	View Menu	Difference	Print Menu
02/12/2010	No Effort	2443.2	2446	2.8	☐
03/12/2010	Moderate	3155.8	3157	1.2	☐
04/12/2010	No Effort	2443.2	2444	0.8	☐

Print Menus

Figure 8. *Menu configuration interface*

After the menus were constructed they can be printed and the daily calorie intake is charted to show how the bodybuilder was actually evolving with his daily calorie consumption, as presented in figure 9.

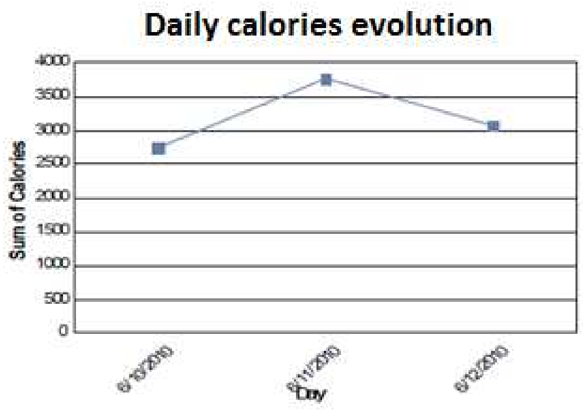
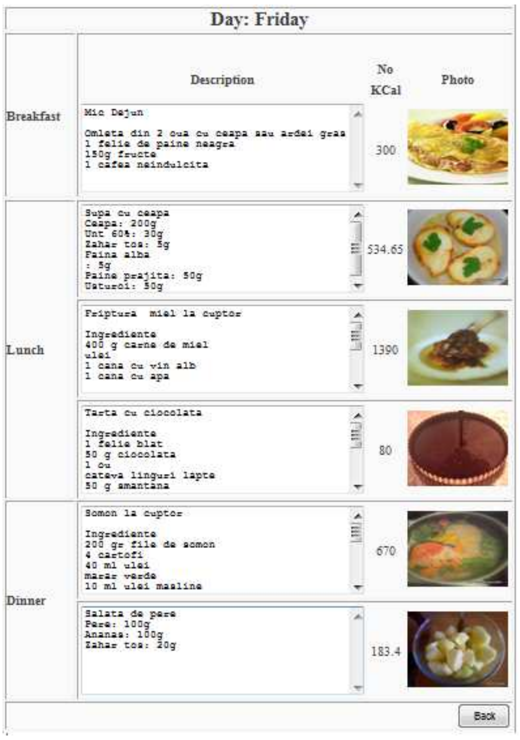


Figure 9. *Calorie Fluctuation Chart*

Each daily meal can also be viewed individually, the content being mapped in categories according to the three daily meals: breakfast, lunch and dinner, figure 10. The user can push the button ***Back*** for he to return to the main frame of the distributed application.



Day: Friday

	Description	No KCal	Photo
Breakfast	Mic Dejun Omleta din 2 oua cu ceapa sau ardei gras 1 felie de paine neagra 150g fructe 1 cafea neindulcita	300	
Lunch	Supa cu ceapa Ceapa: 200g Unt 60%: 30g Zahar tos: 5g Faina alba : 5g Paine prajita: 50g Usturoi: 50g	534.65	
	Friptura miel la cuptor Ingrediente 400 g carne de miel ulei 1 cana cu vin alb 1 cana cu apa	1390	
	Tarta cu ciocolata Ingrediente 1 felie blat 50 g ciocolata 1 ou cateva linguri lapte 50 g smantana	80	
Dinner	Somon la cuptor Ingrediente 200 gr file de somon 4 cartofi 40 ml ulei marar verde 10 ml ulei masline	670	
	Salata de pere Pere: 100g Ananas: 100g Zahar tos: 20g	183.4	

Figure 10. *Meal description interface*

The data provided for a single meal, give bodybuilders info about the meal content, meal total number of calories and a suggestive picture of the food.

3. Optimization assumptions

Security optimization for the SDACO framework is described as a process of improvement of the general security level, especially if the distributed application is offering its resources free to a non homogenous mass of users. The application must avoid any resource tampering or any undesired interruption of its functionalities.

The main characteristics that need to be constantly assured are the availability and functionality. Under no circumstances the application mustn't cease operation whatever the users are doing and entering into the system.

The optimization is directed towards the following concepts of improvement, which will help the distributed application to preserve its resources under unpredicted behaviour:

- validation of input data;
- protection of sensitive information;
- simplifying the communication between users and distributed framework.

Input data validation must be conducted to lower the possible errors coming from wrong data introduction. This limitation in necessary, but in the same time can't be abused because of the negative implications brought upon the usability factor.

Figure 11 is revealing a balance which must be achieved between the total limitations of the distributed applications interface and the usability factor perceived by users at a normal rate.

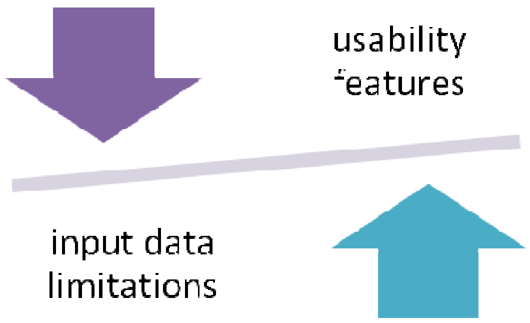


Figure 11. *SDACO restrictions balance*

Following we will discuss another optimization aspect given by the degree of vulnerable information criteria, giving example on the SDACO implementation.

The first optimization criterion is deducted from the aspect of communication. In the process of communication between different parts of application, important and sometimes vulnerable information is vehiculated. The process of communication, must be, in this sense, well managed, taking care that there's no possibility to gain control over the resources using sensitive data, skipped to end-users in the communication process.

Some of the ways of accessing data from the communication process between application components is by checking the following objects for any sensitive material:

- URL's – the internet addresses used to navigate through the application; sensitive information can be revealed in this region as part of the web submission process of a form through the GET method;
- Cookies – temporally client-side files used for storing different setting data for the web based distributed application.

In this sense, an optimization in terms of minimizing the value of the indicator ***D**egree of **S**ensitive **D**ata*, DSD is required [6]:

$$DSD = 1 - \frac{TSDP(Kb)}{TSD(Kb)}$$

where:

TSDP – amount of sensitive data which is protected by user's access;

TSD – amount of total sensitive data vehiculated by the application.

The DSD indicator takes values between $[0; 1)$ in the following scenarios:

- *DSD = 0*, when all sensitive information vehiculated in the application is protected;
- $\lim_{TSDP\to 0} DSD = 1$ because $\lim_{TSDP\to 0} \frac{TSDP}{TSD} = 0$ and $DSD = 1 - 0 \Longrightarrow DSD = 1.$

The limitation of sensitive data to end-user is a major factor of impact upon the vulnerability of a distributed application.

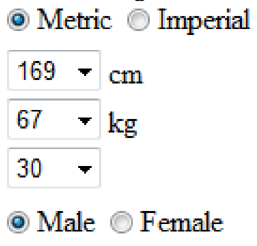
4. Security optimization methods

The security methods used for achieving optimization are described by standard or developed user controls and functions to counteract the unpredictable behaviour of bodybuilding users who are willing to test the resources putted at their disposal.

An important aspect is given by the input data interface which is the main gate of attacking a distributed application.

The first step was to limit the introduction of free data by users and to force them to choose from predefined values. For doing this controls like ***Drop Down List*** or ***Radio List Buttons*** were used, as shown in figure 12. In order to do so, the weight and height were restricted to specific intervals, as follows:

- height – values between 140 and 220 for metric units or 55 and 87 for imperial units;
- weight – values from 50 to 200 for metric units and 110 to 441 for imperial units;
- age – values from 18 to 99 years;
- sex – a pair of radio buttons with the two possible values;
- activity effort type – radio buttons describing the effort intensity level.



[✓ Metric] Imperial

169 cm

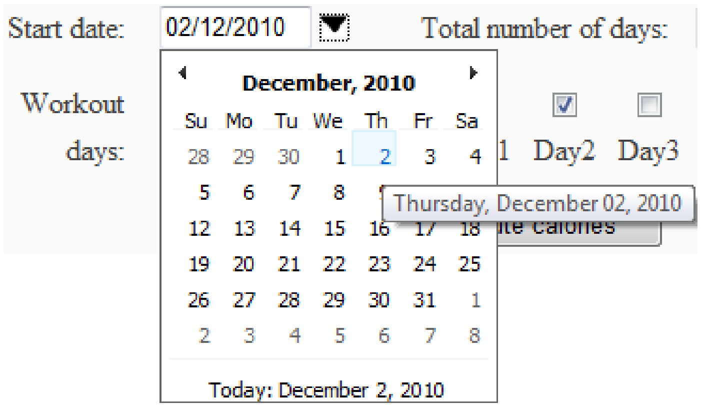
67 kg

30

[✓ Male] Female

Fig. 12 – *Restriction controls for SDACO*

The introduction of the date specific for the beginning period of workouts is made through an AJAX calendar control picker which limits the introduction of invalid date periods, figure 13.



Start date: 02/12/2010 Total number of days:

Workout days:

December, 2010

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	1	2	3	4
5	6	7	8			
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Thursday, December 02, 2010

Today: December 2, 2010

Figure 13. *AJAX calendar picker*

But the validation of the date beside the fact that is a correct value must be also be done at a conceptual level, meaning that the date chosen as the first date of the total time interval for which the bodybuilder is configuring his meals, must be greater or equal to the current date. For this a client-side function is used to validate the date chosen from the calendar picker. This function is described below.

```
function checkDate(sender, args) {
   var today = (new Date()).setHours(0, 0, 0, 0);
   var selected = sender._selectedDate.setHours(0, 0, 0, 0);
   if (selected < today) {
    alert("You cannot select a day earlier than today!");
    sender._selectedDate = new Date();
    sender._textbox.set_Value(sender._selectedDate.format(sender._format));
    changedateOut(Date.UTC(sender._selectedDate),false);
   }
   else{
          changedateOut(selected, true);
   }
}
```

Also, the controls used to display the start and end date must be disabled for inputting other values than the ones selected or calculated. For doing this the following function was used as a client-side script which together with the previous function disables the edit operation of those two displaying controls.

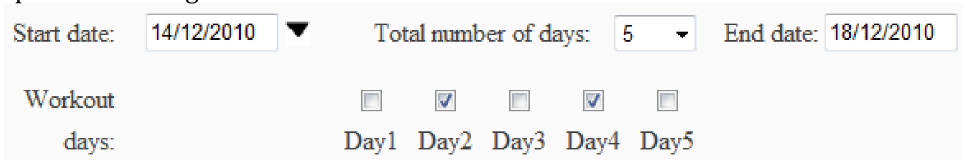
```
function isValidKey(e){
        var charCode = e.keyCode || e.which;
        if (charCode == 8 || charCode == 46)
        return false;
        return true;
}
```

The end date is calculated based on the input date and the number of total workout days selected from a drop down list control with a maximum number of days equal to seven, as seen in figure 14.

The script for validating the end date was automatically register on the server and passed to the page dynamically. The procedure through which the script was written to the page from managed code is described below.

```
protected void RegisterScript(){
  ClientScriptManager cs = Page.ClientScript;
  Type scriptType = this.GetType();
  StringBuilder scriptText = new StringBuilder();
  scriptText.Append("<script type=\"text/javascript\">{body function}</script>");
  if (!cs.IsClientScriptBlockRegistered(scriptType, "clientScript"))
   cs.RegisterClientScriptBlock(scriptType, "clientScript", scriptText.ToString());
}
```

The workout days after they were established, by means of a check box list control they can be selected as presented in figure 14.


Start date: 14/12/2010 Total number of days: 5 End date: 18/12/2010

Workout days: Day1 [ ] Day2 [✓] Day3 [ ] Day4 [✓] Day5 [ ]

Figure 14. *Date and workout days' selection*

The client-side validation of data inputted is not enough. A double side validation must be conducted in order to avoid data tampering which may occur along the communication channel. A simple interception of data from the client-side is enough to alter the values and pass wrong or modified parameters to the server which, without knowing that the transfer was intercepted, will try to convey the results, also wrong, to the end user without knowledge of any alteration.

This can be avoided by using a server validation of any data inputted on the client side. For example, the values for start and end date inputted on the client and validated by the local JavaScript functions, are also validated using a simple ***try*** and ***catch block*** which, in case of an error – an interception and modification of date values along the communication channel – will return a custom error object which will be passed to a upper level.

```
try{
        bodybuilder.DateIn = DateTime.Parse(txtDateIn.Text);
    bodybuilder.DateOut = DateTime.Parse(txtDateOut.Text);
}
catch (DataException dex){
  return new error(dex);
}
```

Another way of optimizing the security level is by protecting undisclosed information which might be accessed or accidentally viewed by users. An implementation of this optimization process was made in the application by encrypting the URL syntax presented above with an open source encryption algorithm found to be most efficient in [6].

If in the process of transferring information about the daily menus for providing the users the final results, the following syntax, *S1*, for the URL is used by the application:

S1=http://www.webserver/pathASPXfile?param1=value1&param2=value2&...&param7=value7, where:

- webserver – icnc.ase.ro;
- pathASPXfile – application file path of the requested resource;
- $param_i, i = \overline{1,7}$ - the parameters' names for sending menu ID's for the next resource;
- $value_i, i = \overline{1,7}$ – the parameters' values consisting in menu ID's and day ID.

Having these types of information at its disposal, a user can actually see throughout the whole sets of menus, just by modifying the parameters values and submitting again a valid request to the web server.

The security optimization process takes place under the following step procedure:

A. identifying the parameters' values and creating the *S1* URL syntax;

B. encrypting *S1* into *S2* by means of the encryption function $OTP(S1) = S2$;

C. creating the new URL, *S3* from *S2*, based on the URL Base64 Encoding representation for being able to be sent over the internet;

D. the host page gets the URL S3 syntax from the request object of the page and decodes it from the Base64 representation, transforming it into S2 encrypted state;

E. from the encrypted S2 state, the mirror encryption operation takes place, as the OTP are symmetric class algorithms, and the final S1 syntax is decoded;

F. the S1 syntax will serve to successfully retrieve all the menu ID's and day ID for showing the end results.

The final URL syntax which travels the network and is transparent to the user is like *S3=http://www.icnc.ase.ro/User/MenuDetail.aspx?pass=ARaQQtorYdYiL7wHrQusDs4Do8WrtQ2*.

Table 1, shows an improvement of the *DSD* indicator due to growth of the *TSDP* value, and eventually an optimization of the total security level.

Table 1. DSD correlations

Without Security	**Relation**	**With Security**
$TSDP_0$ (Kb)	<	$TSDP_1$(Kb)
$\frac{TSDP_0(Kb)}{TSD(Kb)}$	<	$\frac{TSDP_1(Kb)}{TSD(Kb)}$
$1-\frac{TSDP_0(Kb)}{TSD(Kb)}$	>	$1-\frac{TSDP_1(Kb)}{TSD(Kb)}$

From table 1 is resulting $DSD_1 < DSD_0$, meaning an optimization of the level of security by hiding a greater part of the undisclosed data.

5. Conclusions

The distributed application with free access to resources wants to offer full functionality to users highlighting the necessity of security implementation in order that users who are entering and using it, can't jeopardize the availability, reliability and any other quality characteristic of the distributed applications and in the same time enjoying the full functionality.

The implemented security components are meant to restrict normal user not to undertake actions that could compromise the application but on the other hand, the security components are not meant to lower the application's usability.

References

[1] C. Boja and M. Doinea, "Security assessment of a web distributed application," *Informatica Economica Journal*, vol. 14, No.1, 2010, pp. 152 – 162, ISSN 1453-1305

[2] Wikipedia Online, Body mass index, [Online], Available at: http://en.wikipedia.org/wiki/Body_mass_index

[3] M. Doinea, "Distributed Application for Calories Optimization," *Body Building Journal*, Vol. 1, No. 1, 2010, pp. 83-93, ISSN 2066-8007

[4] J.E. Blundell, J. Cooling and N.A. King, "Differences in postprandial responses to fat and carbohydrates loads in habitual high and low fat consumers (phenotypes)," *British Journal of Nutrition*, No. 88, 2002, pp. 125-132, ISSN 0007-1145

[5] I. Ivan, S. Vinturis, "Effort-nutrition correlation," *Body Building Journal*, Vol. 1, No. 1, 2009, pp. 94-101, ISSN 2066-8007

[6] Doinea, M. – "Security optimization of a distributed application for calculating daily calories consumption," *Journal of Information Systems & Operations Management*, Vol. 4, No. 1, 2010, pp. 12-22, ISSN 1843-4711

Mihai DOINEA has a master diploma in Informatics Security (2006). He is also a lecturer assistant and he teaches data structures and advanced programming languages at the Academy of Economic Studies. He published more than 40 articles in collaboration or as single author. His research interests are given as follows: informatics security, distributed applications, optimization criteria, databases, artificial intelligence, information management, security policies, mobile devices, networking and wireless communication.

Acknowledgements

This article is a result of the project „Doctoral Programme and PhD Students in the education research and innovation triangle". This project is co funded by European Social Fund through The Sectorial Operational Program for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies (project no. 7832, "Doctoral Programme and PhD Students in the education research and innovation triangle, DOCECI").