Corpus application manual

Content

1. Corpus User

Corpus User is web part of the whole *Corpus System.* It is designed for common users, who want to use tools for testing compression algorithms and who want to work, compare and create reports according to result data of each test. *Corpus User* web part is web GUI, which allows easy work with collected result data of all completed tests.

Corpus User starts with *Login page*, which is the first step to get into application. User must have confirmed access to application by administrator. Then all the features and useful tools for testing compression algorithms are enabled for user's use. (img. 1_1)
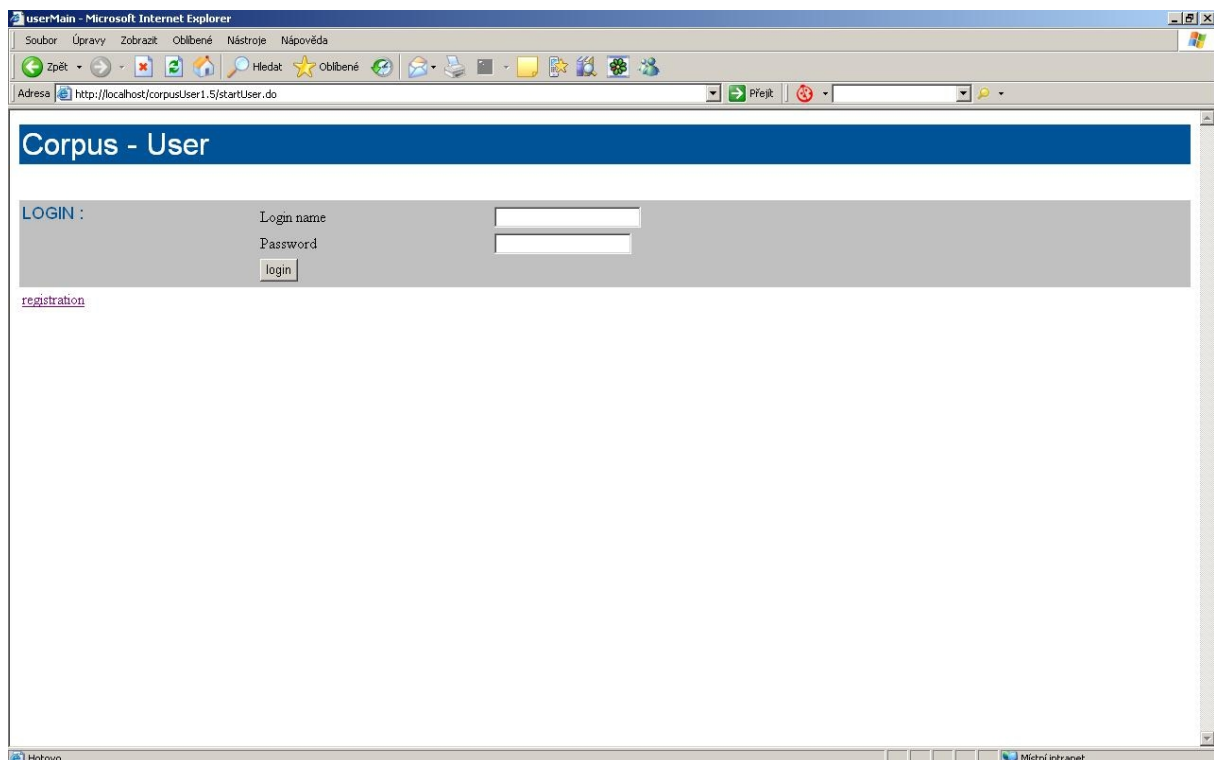
Img. 1_1

After entering the application, system shows *Main menu* page, which contains options:
Create compression test – tools for doing tests on compression algorithms
Statistic data comparison – tools for browsing through result data, compare them and making report tables, which contains data, specified by user
User tests results – tools for browsing through all completed or terminated compression tests
Parameter groups – tools for defining additional parameters, which can be associated with each test
(img. 1_2)
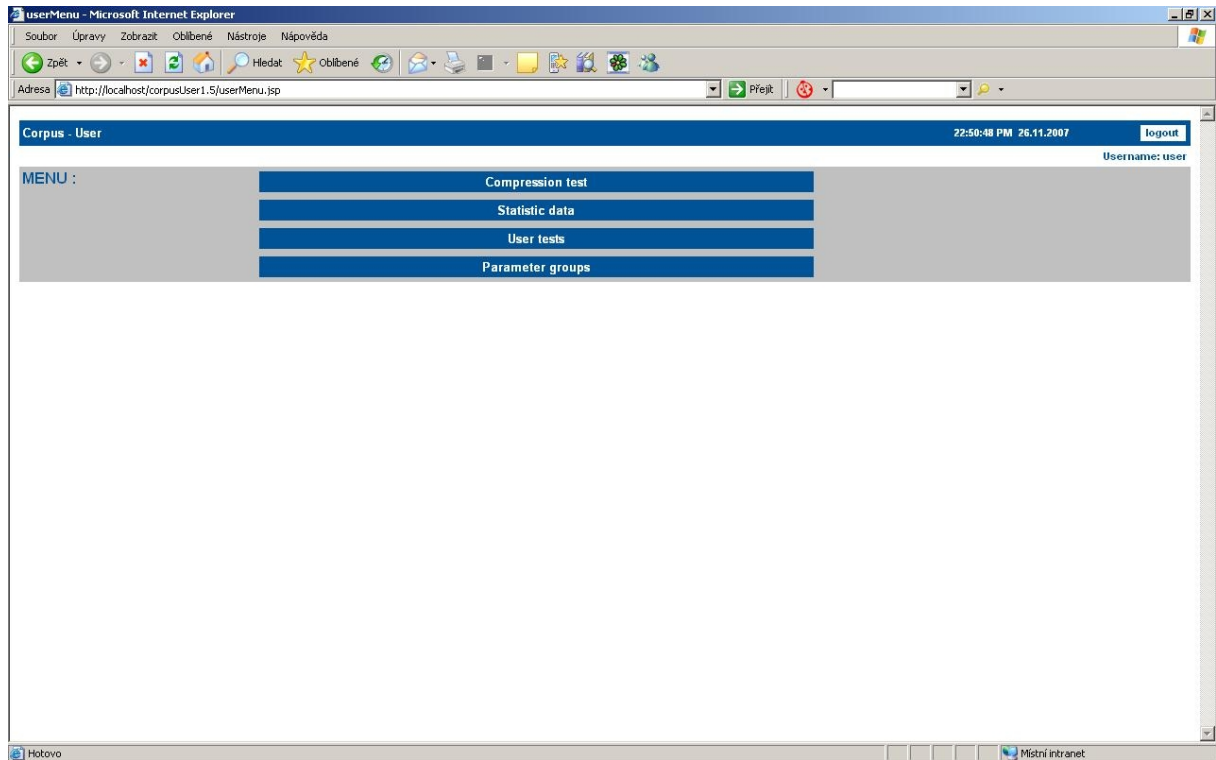
Img. 1_2

1.1. Create compression test

System functionality for testing reliability and efficiency of user's compression algorithm.

Firstly system shows page, where user can set up additional parameters of his test.
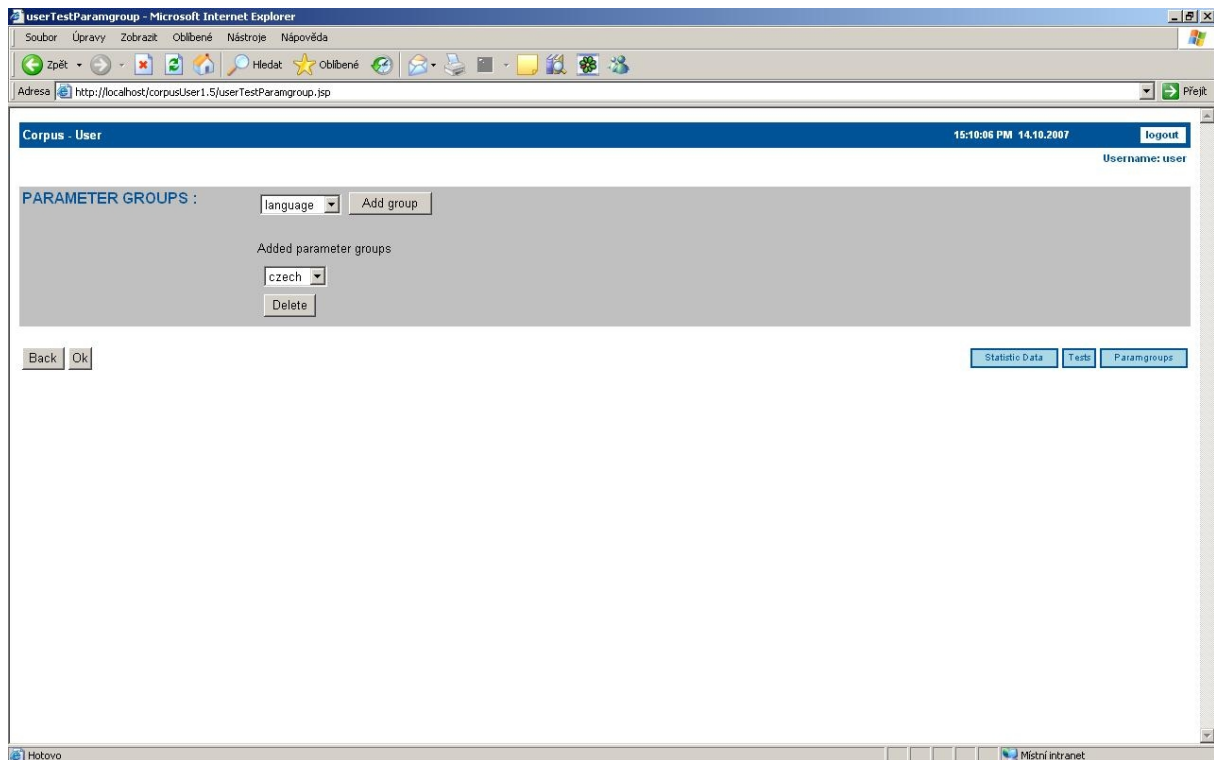
1.1.1. Parameter groups

Page *Parameter groups* allows user, to set up his own additional parameters, that he wants to associated with actual test. These parameters are only useful for creating special report, which uses user's additional parameters. User can continue to mandatory parameters without defining his additional parameters by pressing button *Ok*. (img. 1.1.1_1)
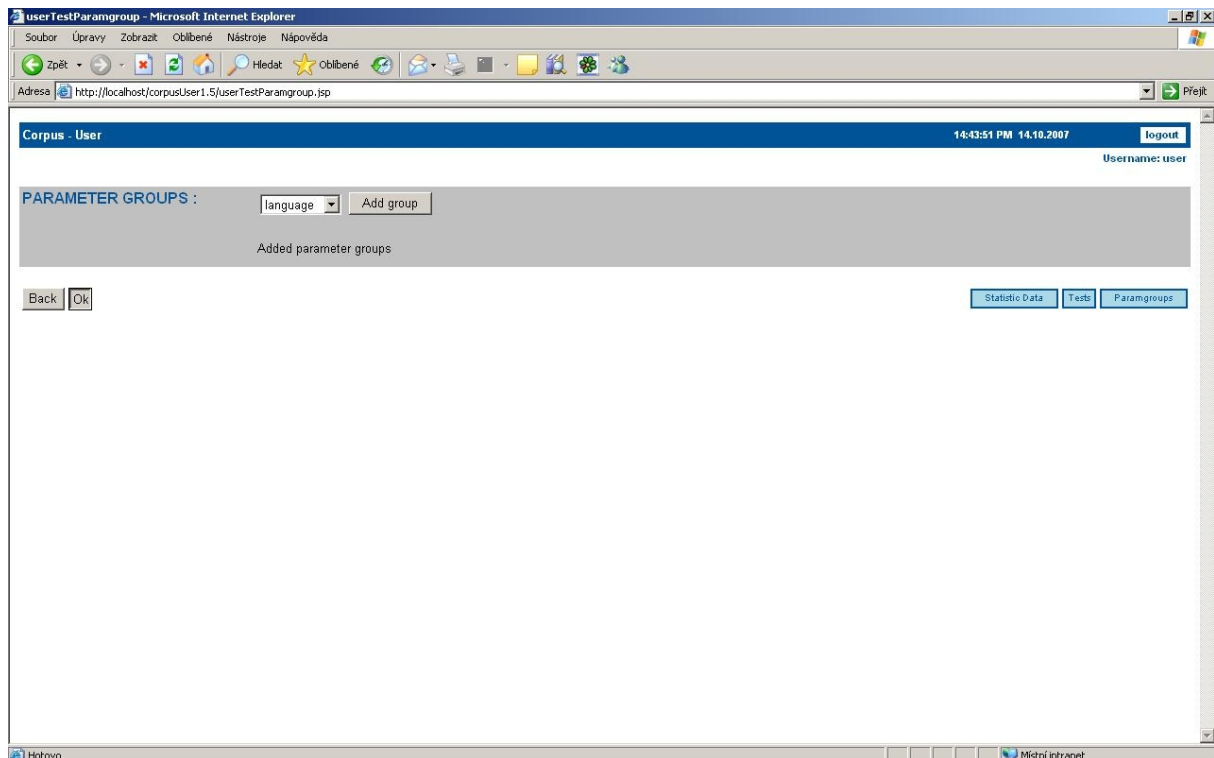
Img. 1.1.1_1

User can associate test only with parameters, which were already created. To create and define parameters, user must use system functionality *Parameter groups* manager. By default no additional parameters are added to test. If user wants to add new parameter to test, firstly he has to add parameter group, which contains his required parameter. User chooses name of parameter group from list of parameter groups and then press button *Add group*. (img. 1.1.1_2)

Img 1.1.1_2

After choosing group, user selects required parameter from list of parameters. By adding other groups, system allows to define more parameters, and also delete not suitable parameters by clicking on button *Delete* associated with not willing parameter. (img 1.1.1_3)
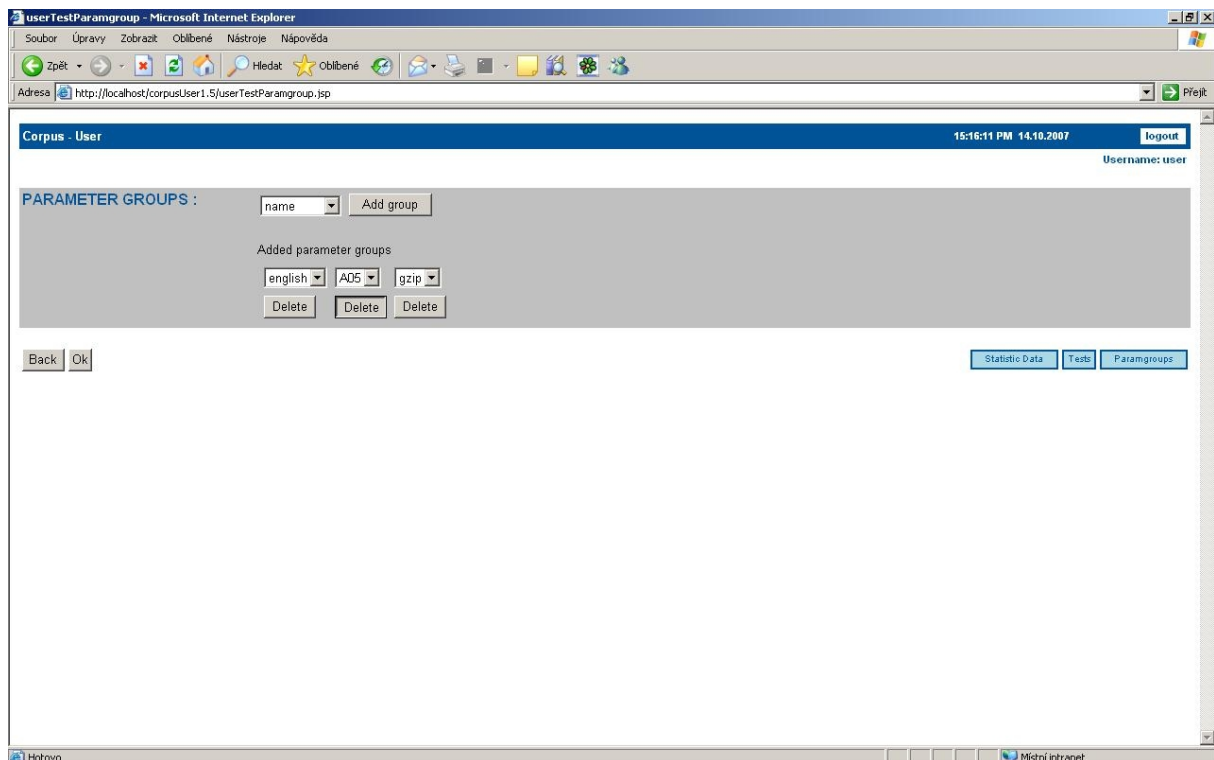

Img. 1.1.1_3

Again to continue into next step of test, user press button *Ok.*

1.1.2. Mandatory parameters

Page with mandatory parameters differs according to way, how user wants to upload his compression program, into two possibilities. If user has got his compression program only in one single file, he has to fill in *Test* input parameters. Group of *Test* input parameters are shown on page by default. (img 1.1.2_1)
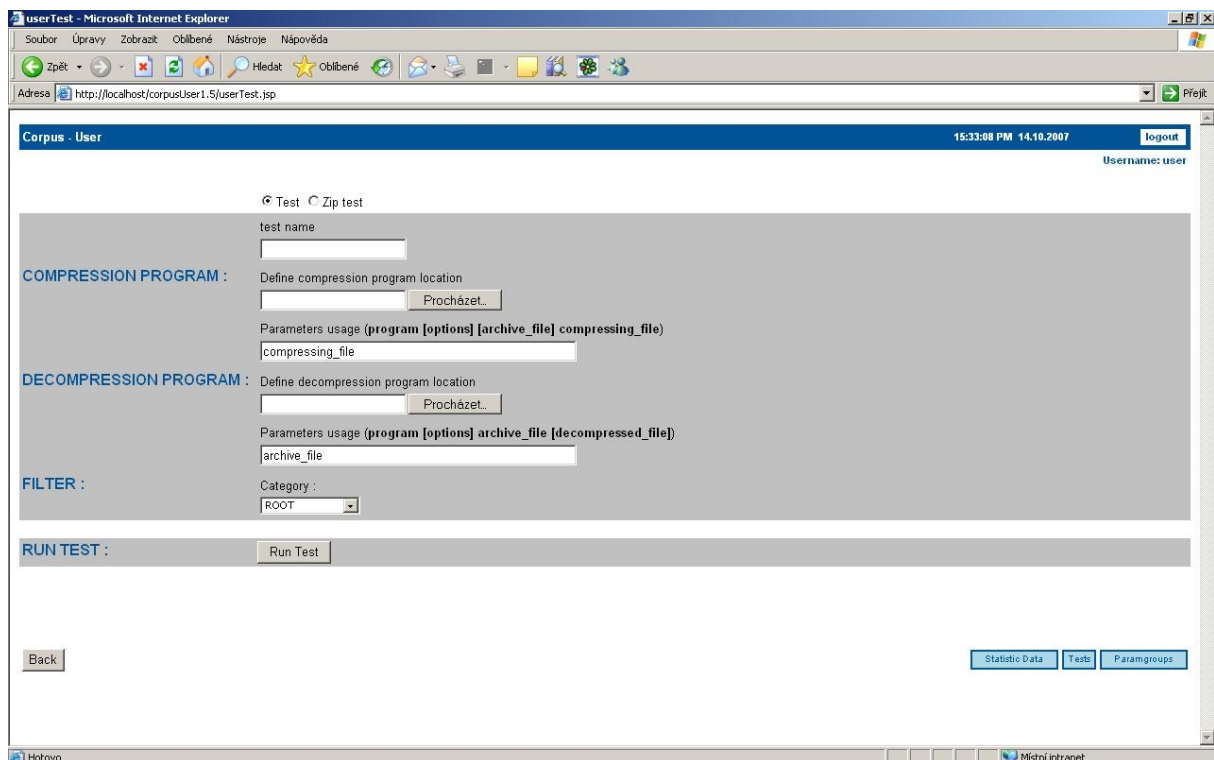
Img. 1.1.2_1

The another possibility, that user must use, is when his compression program uses some libraries, which are needed for correct execution of program, or his compression program is split into more files. Then user checks that he needs *Zip test* input parameters. (img. 1.1.2_2)
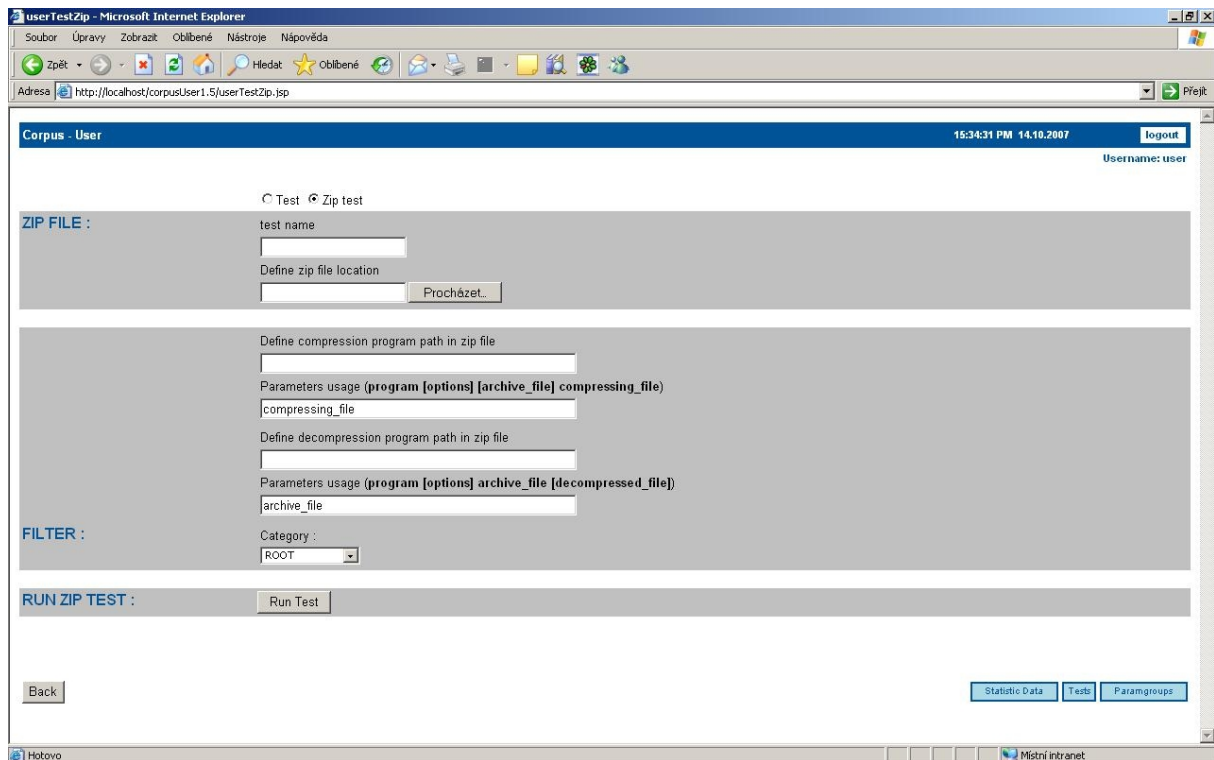
Img. 1.1.2_2

After pressing button *Run Test s*ystem validates all input parameters, if any error occurs, system prints all errors into screen. (img 1.1.2_3)
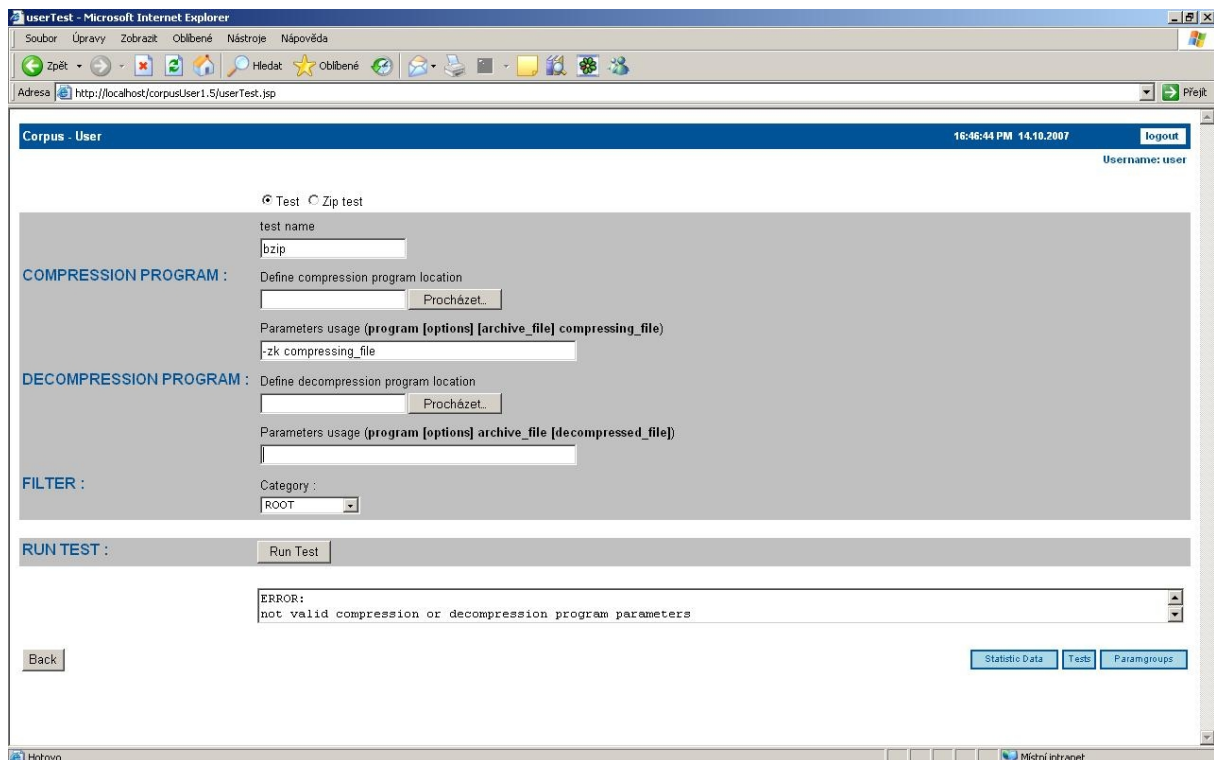
Img 1.1.2_3

If all parameters are valid, test starts, and system shows *Test status* page.

1.1.2.1. *Mandatory parameters – Test*

Input parameters:
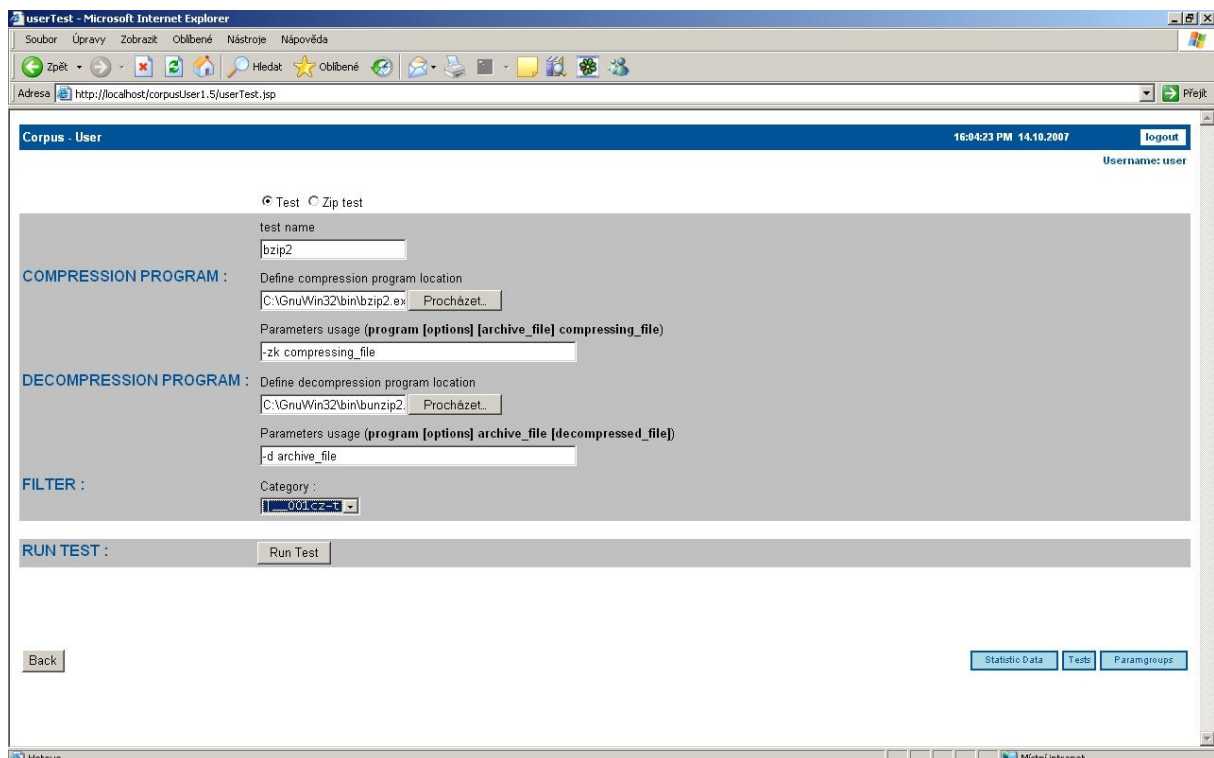*Test name* – name of test, max length of name is 20 chars

Compression program location – compression program file, and location on user's local disc

Compression program parameters – usage of program, value *compressing_file* must be used, represents input file to compress program, then user can fill in his own parameters (options), which are required for correct run of program, if program requires name of archive, then use default name for archive file – *archive_file*

Decompression program location – decompression program file, and location on user's local disc, if compression and decompression programs are the same, user must also define location of decompression program

Decompression program parameters – usage of program, value *archive_file* must be used, represents input file to decompress program, then user can fill in his own parameters (options), which are required for correct run of program, if program requires name of output file of decompression, then use default name *decompressed_file*

Category – user chooses category, part of corpus, on which he wants to run his test, corpus is shown as treeview
(img. 1.1.2.1_1)

Img. 1.1.2.1_1

1.1.2.2. Mandatory parameters – Zip Test

Input parameters:
Test name – name of test, max length of name is 20 chars

Zip file location – zip file location, and location on user's local disc, zip file contains all user files, which are necessary for correct run of compression or decompression

Compression program path – location of compression program in zip file

Compression program parameters – usage of program, value *compressing_file* must be used, represents input file to compress program, then user can fill in his own parameters (options), which are required for correct run of program, if program requires name of archive, then use default name for archive file – *archive_file*

Decompression program path – location of decompression program in zip file

Decompression program parameters – usage of program, value *archive_file* must be used, represents input file to decompress program, then user can fill in his own parameters (options), which are required for correct run of program, if program requires name of output file of decompression, then use default name *decompressed_file*
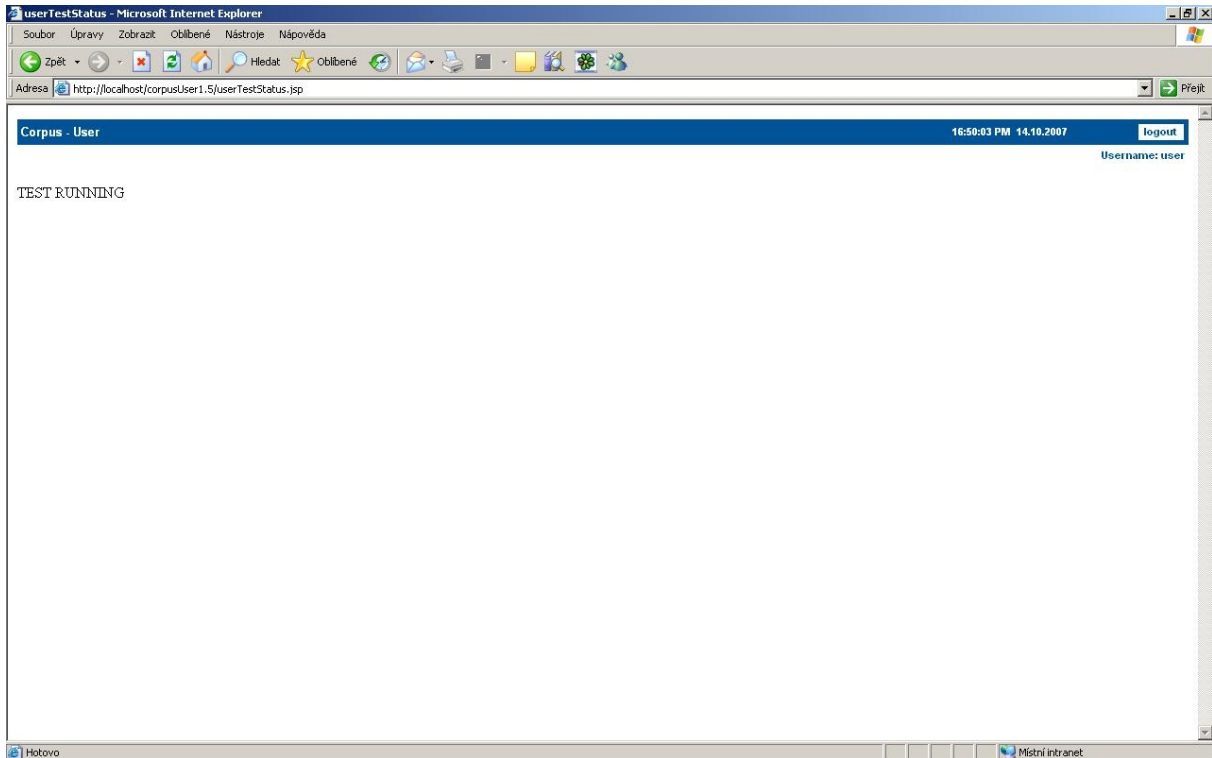
Category – user chooses category, part of corpus, on which he wants to run his test, corpus is shown as treeview
(img. 1.1.2.2_1)
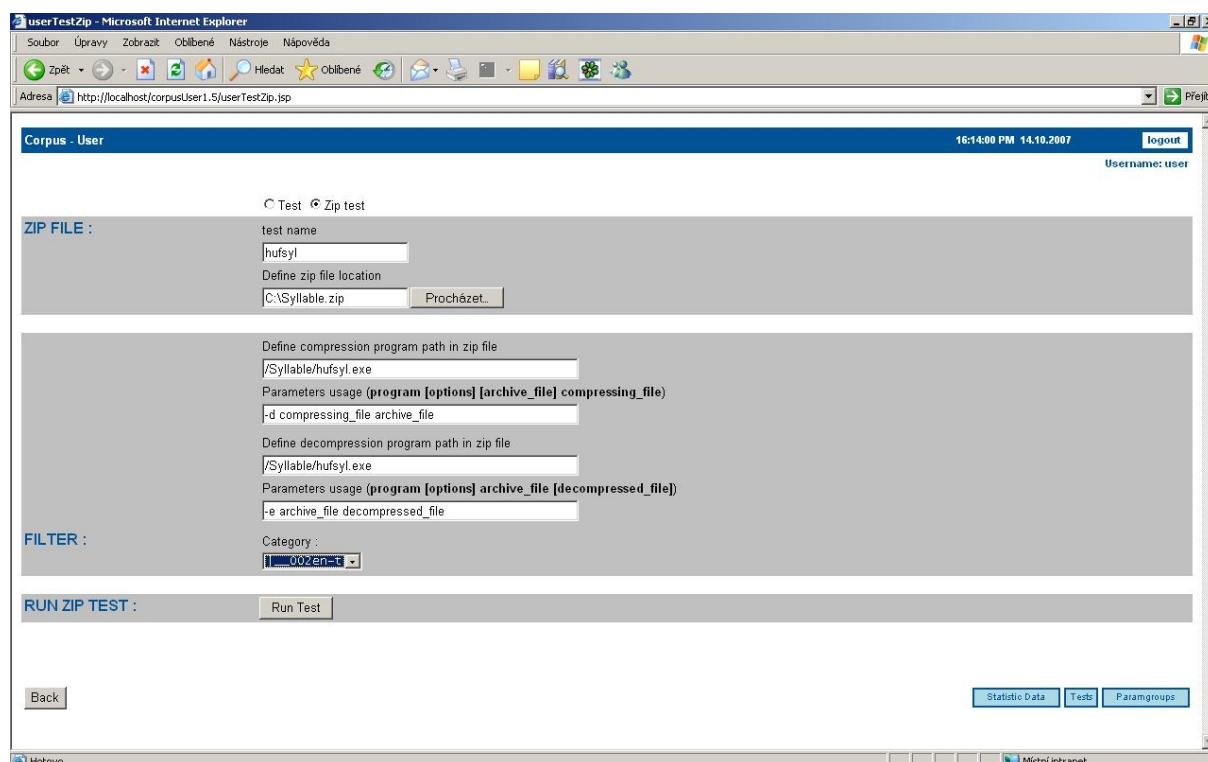
Img. 1.1.2.2_1

1.1.3. Test status

Page *Test status* is shown, while test is running. (img. 1.1.3_1)


Img. 1.1.3_1

1.1.4. Test result

When test is complete, doesn't matter if successfully or not, system displays *Test result* page. If any error occurs during execution of test, system prints report of all test errors (img 1.1.4_2), else test was succesfull (img. 1.1.4_1).

After pressing *Ok* button, system redirects page to first page – *Parameter groups*. User can start with another test.
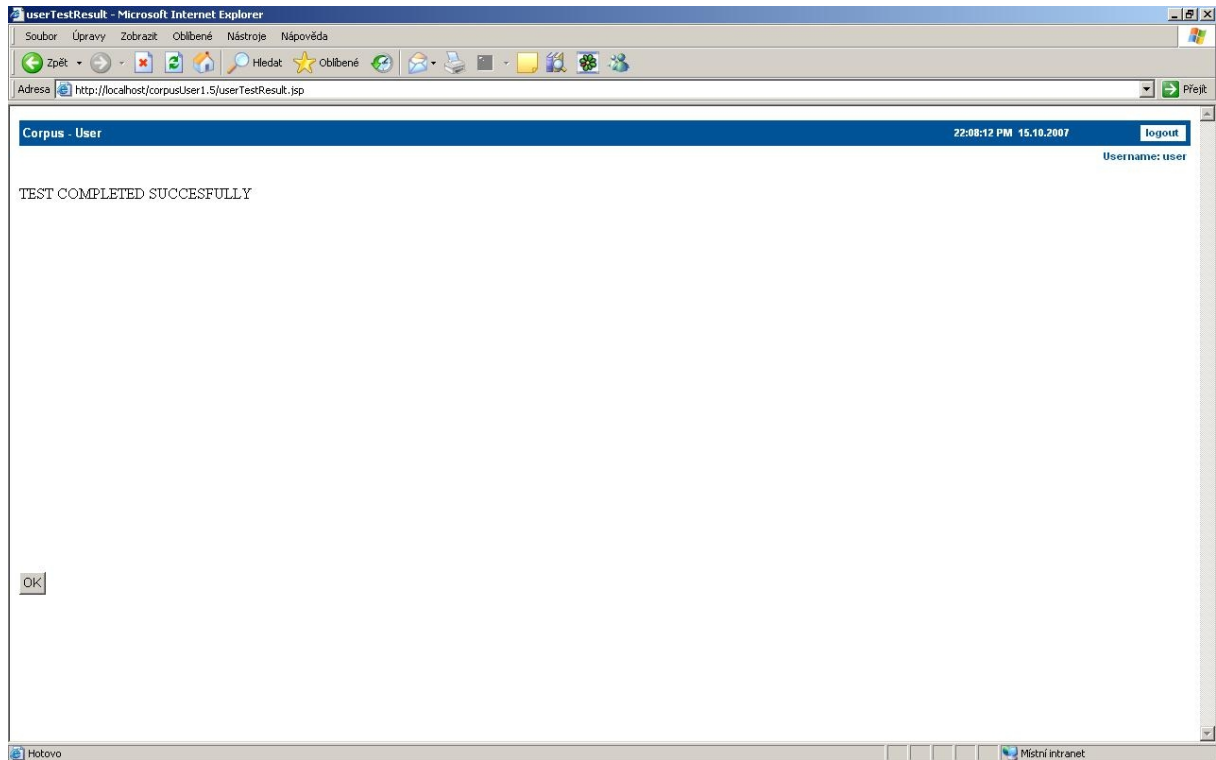
Img. 1.1.4_1
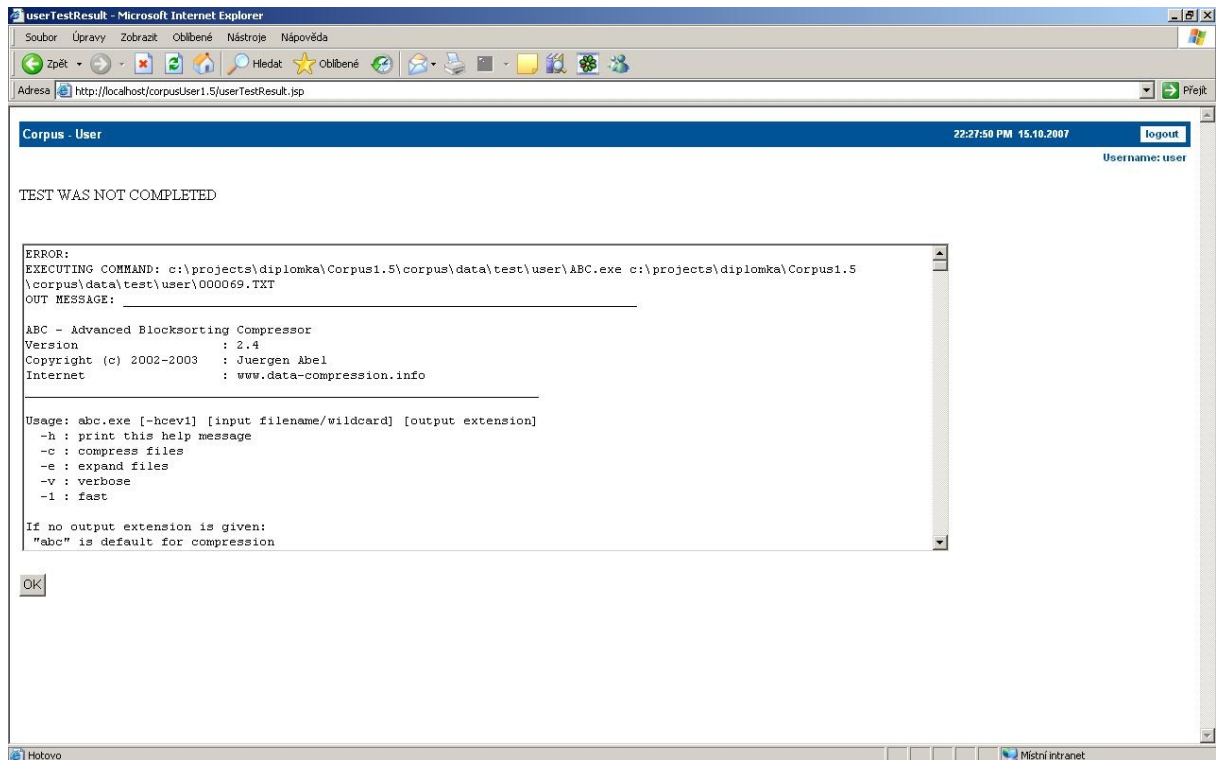
Img. 1.1.4_2

1.2. Statistic data comparison

System functionality for browsing and comparing result data of all tests. Only the data of successful tests are displayed.

Page with table containing basic info of all test results is displayed at first.

1.2.1. Basic info

In *Basic info* page, system displays table with all test results, which were done yet. Each row represents one test with result data. Test result data information contains:

Test ID – unique number of test, set up by system
Test name – name of test, set up by user
Corpus ID – unique number of *Corpus hierarchy*, number represents modification of *Corpus hierarchy*, on which was test done
Category – path in *Corpus hierarchy*, path represents subpart of hierarchy, on which was test done
Data size – average value of source data size, on which was test done, number represents average value of sizes of all corpus source files, which were used for test
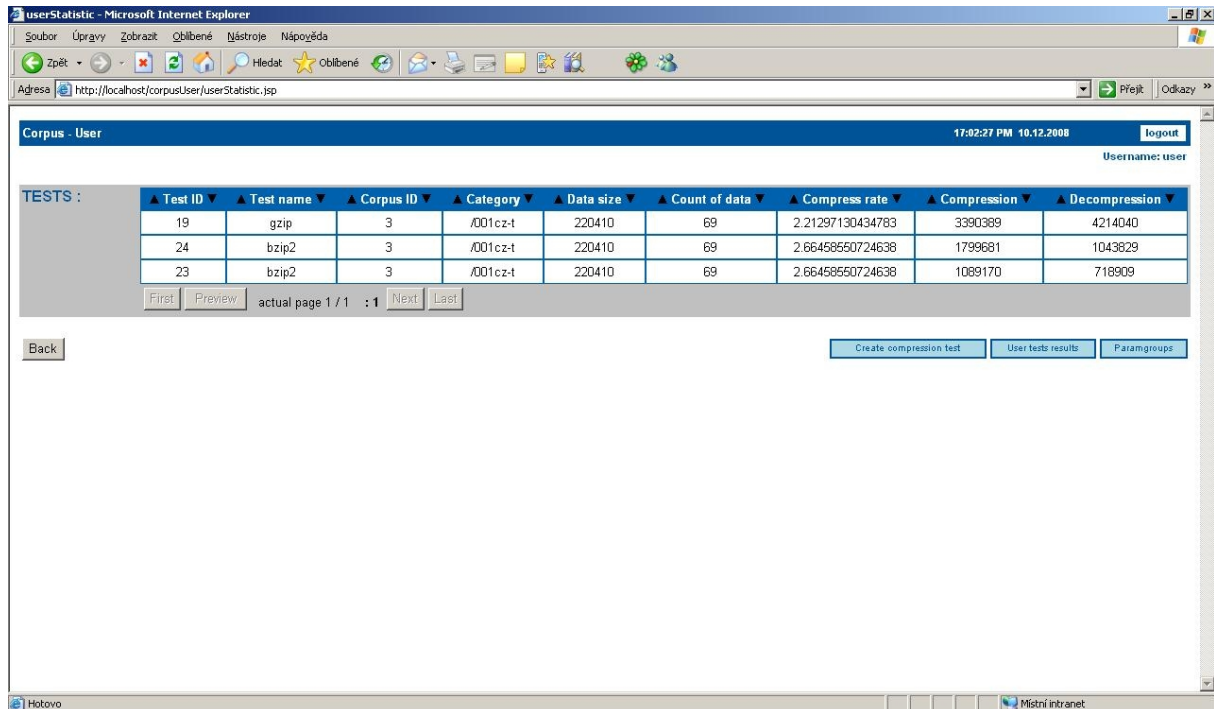*Count of data* – total sum of files, on which was test done
Compress rate – resulting compress rate value, this value represents average value of compress rates of all source files, compress rate value is rate of original source file and compression result file
Compression – resulting fastness value of compression, value represents average value of compression fastness values of all source files, compression fastness value is rate of source file size and compression duration (bytes per second)
Decompression – resulting fastness value of decompression, value represents average value of decompression fastness values of all source files, decompression fastness value is rate of compressed file size and decompression duration (bytes per second)

(img. 1.2.1_1)

Img. 1.2.1_1

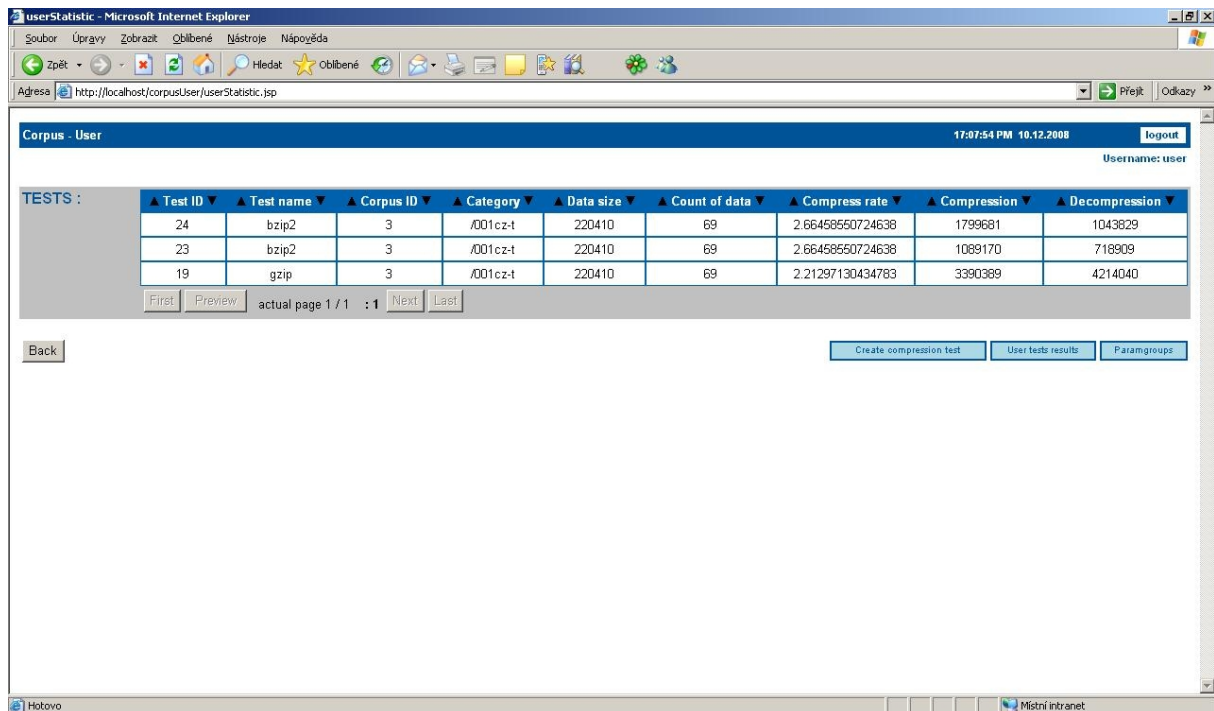User operations with each test result data (row):

Detail info – by clicking into chosen row, system displays table only with test result data, which were done on the same *Corpus hierarchy* version.

Table operations:

Sort test list according to selected column – by clicking on headline of column, user can sort list of tests by each column by ascending or descending order. Not using arrow up or down, system sorts list by ascending order by default. (img. 1.2.1_2)

Choose page – user can choose page, which he wants to display by:
 - pressing button *First* – system displays first page of tests
 - pressing button *Preview* – system displays preview page to actual page
 - pressing button *Next* – system displays next page to actual page
 - pressing button *Last* – system displays last page of table
 - clicking on number of page – system displays exact page

▲ Test ID ▼	▲ Test name ▼	▲ Corpus ID ▼	▲ Category ▼	▲ Data size ▼	▲ Count of data ▼	▲ Compress rate ▼	▲ Compression ▼	▲ Decompression ▼
24	bzip2	3	/001cz-t	220410	69	2.66458550724638	1799681	1043829
23	bzip2	3	/001cz-t	220410	69	2.66458550724638	1089170	718909
19	gzip	3	/001cz-t	220410	69	2.21297130434783	3390389	4214040

Img. 1.2.1_2 – sort desc by Test ID

1.2.2. Detail info

System displays table only with test result data, which were done on the same *Corpus hierarchy* version. Filter over *Corpus hierarchy* version is important to compare and create reports correctly. Table column information:

Test ID – unique number of test, set up by system
Test name – name of test, set up by user
Category – path in *Corpus hierarchy*, path represents subpart of hierarchy, on which was test done
Compress rate – resulting compress rate value, this value represents average value of compress rates of all source files, compress rate value is rate of original source file and compression result
Compression – resulting fastness value of compression, value represents average value of compression fastness values of all source files, compression fastness value is rate of source file size and compression duration (bytes per second)
*Decompression* – resulting fastness value of decompression, value represents average value of decompression fastness values of all source files, decompression fastness value is rate of compressed file size and decompression duration (bytes per second)

(img. 1.2.2_1)

15

Img. 1.2.2_1

User operations with each test result data (row):

Selecting tests – user can select tests (rows), with which he can work in next steps (img 1.2.2_2)

Compare by compress rate – by pressing button *Compare* under column *Compress rate*, system compares and sorts all selected test using statistical *Median method, Median method* uses *compress rate* values as input data

Compare by compression – by pressing button *Compare* under column *Compression*, system compares and sorts all selected test using statistical *Median method, Median method* uses *compression fastness* values as input data

Compare by decompression – by pressing button *Compare* under column *Decompression*, system compares and sorts all selected test using statistical *Median method, Median method* uses *decompression fastness* values as input data
(img 1.2.2_3)

Report – system displays page with GUI for creating reports

Report Paramgroups - system displays page with GUI for creating reports, which use additional test parameters

16

Table operations:

Sort test list according to selected column – by clicking on headline of column, user can sort list of tests by each column by ascending or descending order. Not using arrow up or down, system sorts list by ascending order by default. (img. 1.2.2_4)

Choose page – user can choose page, which he wants to display by:
- pressing button *First* – system displays first page of tests
- pressing button *Preview* – system displays preview page to actual page
- pressing button *Next* – system displays next page to actual page
- pressing button *Last* – system displays last page of table
- clicking on number of page – system displays exact page

(img. 1.2.2_5)

Img. 1.2.2_2
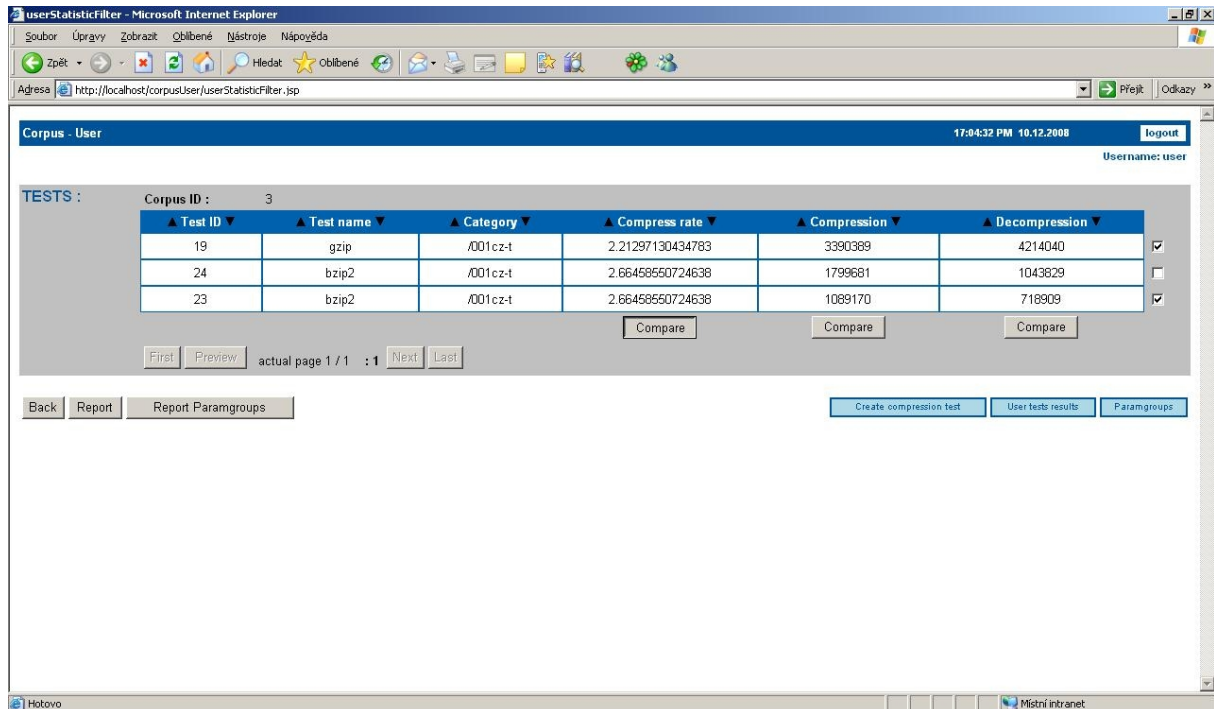
Img. 1.2.2_3

1.2.3. Compare

System displays table with sorted tests. Tests are sorted by specific value and system uses *Median test* to compare result data of tests. Tests are shown in ascending order.
(img. 1.2.3_1)

Table operations:

Choose page – user can choose page, which he wants to display by:
- pressing button *First* – system displays first page of tests
- pressing button *Preview* – system displays preview page to actual page
- pressing button *Next* – system displays next page to actual page
- pressing button *Last* – system displays last page of table
- clicking on number of page – system displays exact page
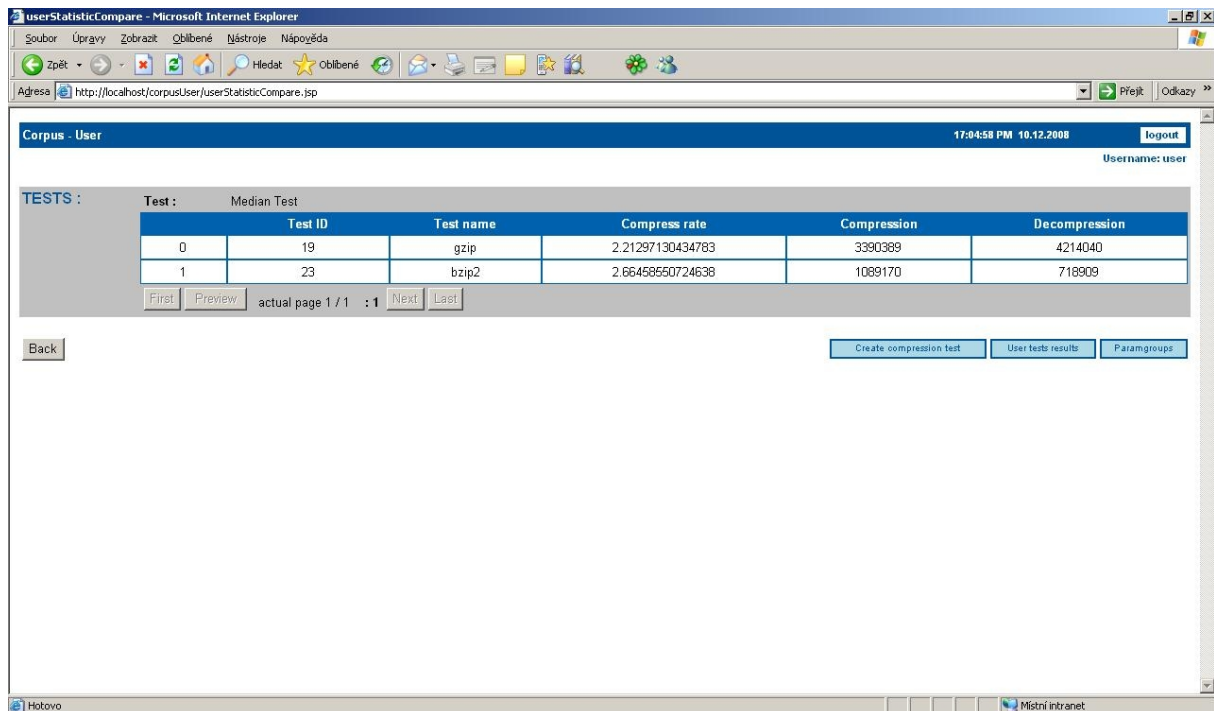
Img. 1.2.3_1

1.2.4. Report

System displays tools, which help user create own report from result data. (img. 1.2.4_1)

User operations for formatting table:

Displayed compression mode – user can choose two possibilities, how he see compression rate values, the first format is *Compression Ratio* and the second one is *Space Savings*. *Compression Ratio* represents rate of uncompressed size and compressed size. *Space Savings* noted as percentage value, is defined as the reduction in size relative to uncompressed size. (img. 1.2.4_2)

Count of decimals – user can choose how many decimals in compression rate values should be displayed (img. 1.2.4_3)

Choose columns – user can check columns, which he wants to have in report (img. 1.2.4_4)

Test Description column – user can split column *Test description* into more columns, so in report, there are painted more columns. User must use char '*|*' to split value into two columns.

19

After placing all split chars, system takes row with the highest count of split chars, and according to this line, system splits the whole column *Test description.* For example there are two rows, the first one has in *Test description* column value '*bzip*' and the second one '*gzip*'. Now, user wants to add some prefix description to actual name. So he adds split chars before each *Test description* column value. After his operation, there are two rows with values '|bzip' ans '|gzip'. So in report, there will be displayed two columns for test description, but the first split column is empty, because user does not defined prefix for any column:

	Test description	
	Split part 1	Split part 2
Row 1	--	bzip
Row 2	--	gzip

Now, user adds prefix value '*english*' into first row, so the *Test description* column has value '*english|bzip*', the second row is still empty:

	Test description	
	Split part 1	Split part 2
Row 1	english	bzip
Row 2	--	gzip

User can continue in splitting *Test description* column by using some postfix expressions and so on… (img. 1.2.4_5 – 1.2.4_6)

Img. 1.2.4_2

Img. 1.2.4_3

Img. 1.2.4_4

Img. 1.2.4_5

Img. 1.2.4_6

User operations for each test (row):

Check rows – user can choose, which tests (rows) he wants to have in report (img. 1.2.4_7)

Select row – user can use other tools for working with each row, but these tools are active only when at least one row is selected

Row Up – system moves selected row in front of previous row (img. 1.2.4_8)

Row Down – system moves selected row behind next row (img. 1.2.4_9)

Add Line – system adds double line behind selected row (img. 1.2.4.10)

Remove Line – system removes double line for selected row, if this double line exists

Functionality of adding and removing double lines into and from table, splits this table into more fragments. Each fragment has in each row max/min value, displayed as bold text. So if table has two fragments, each column has two max/min values.

23

Img. 1.2.4_7

Img. 1.2.4_8

Img. 1.2.4_9

25

Img. 1.2.4_10

Standard table operations:

Choose page – user can choose page, which he wants to display by:
- pressing button *First* – system displays first page of tests
- pressing button *Preview* – system displays preview page to actual page
- pressing button *Next* – system displays next page to actual page
- pressing button *Last* – system displays last page of table
- clicking on number of page – system displays exact page

If user is satisfied with format of table, he can generate report of this table into various file formats:

TeX report – by pressing button *Tex report*, system generates *TeX* source code of formatted table, and user can download and save this *TeX* source code into his local disc. Then he can use this downloaded source code into his *TeX* document. (img. 1.2.4_11 – 1.2.4_13)
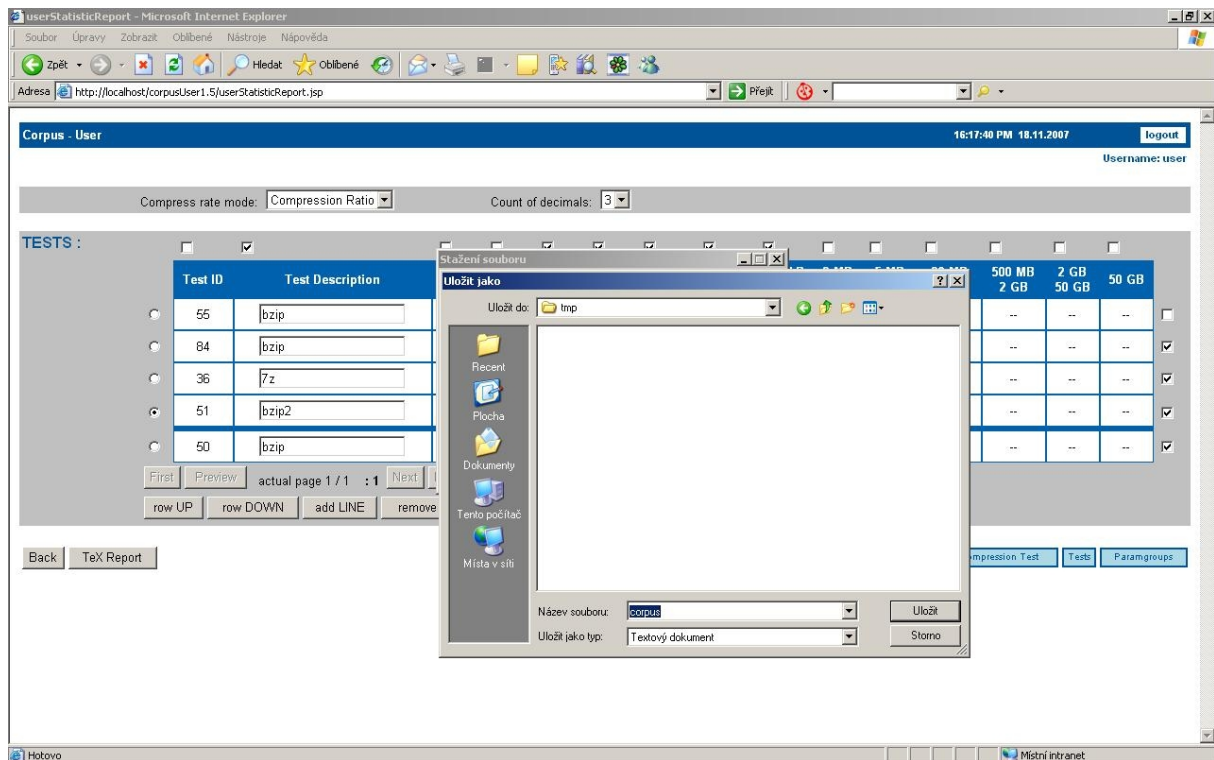
Img. 1.2.4_11

Img. 1.2.4_12 – TeX source code

| | 1 kB | 10 kB | 50 kB | 200 kB | 500 kB |
	10 kB	50 kB	200 kB	500 kB	2 MB
bzip	2.010	2.309	2.543	**2.564**	**2.581**
7z	**1.829**	**2.113**	**2.407**	2.572	2.700
bzip2	2.010	2.309	2.623	2.813	3.000
bzip	**2.010**	**2.309**	**2.623**	**2.813**	**3.000**
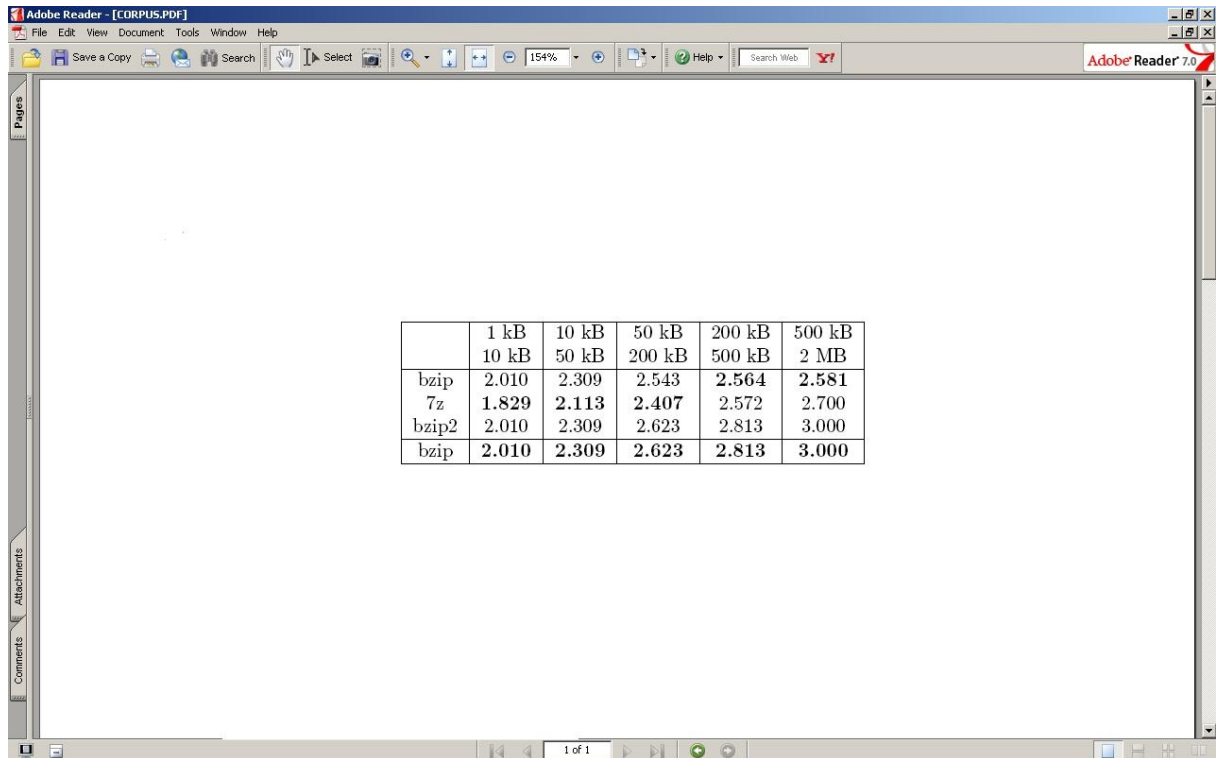
Img. 1.2.4_13 – TeX result in PDF file

R-Graph report – by pressing button *Graph*, system generates *R-Graph* source code of formatted table, and user can download and save this source code into his local disc. Then he can use this downloaded source code to generate graph from result values of tests and insert graph into his publications. (img. 1.2.4_14 – 1.2.4_16)
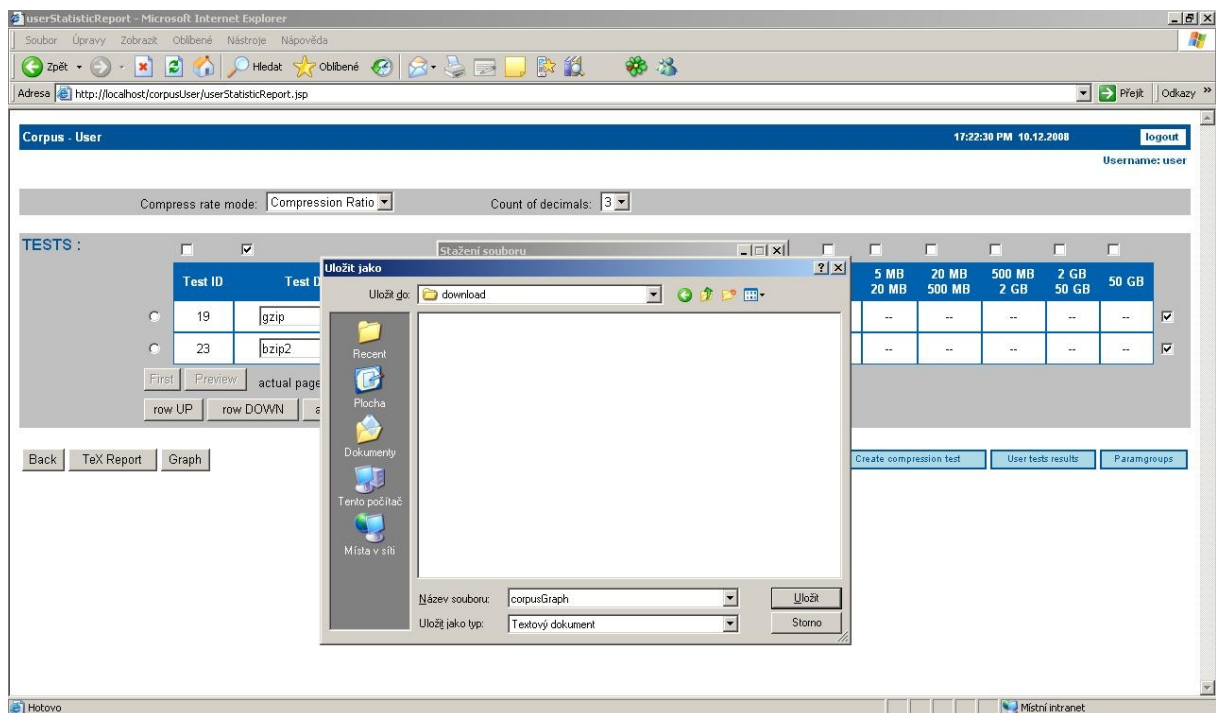
Img. 1.2.4_14 – R-Graph

```
dat <- data.frame(w=c("G02","G03","G04","G05","G06"),gzip=c(1.944,2.062,2.227,2.2
63,2.267),bzip2=c(2.010,2.309,2.623,2.813,3.000),nula=c(0,0,0,0,0),row.names=NULL
)
legenda1 <- c("gzip","bzip2")
par(usr=c(0,5,0,100))
par(cex=1.23)
par(lab=c(15,15,0))
par(lwd=2)
plot(nula ~ w, data=dat, main = "", xlab = "file sizes", ylab = "Compression
ratio in bpc", xlim = c(0.0,0.5), ylim = c(1.944,3.0), type="b", lty=1,pch=1)
lines(gzip ~ w, data=dat, type="b", lty=1, pch=0)
lines(bzip2 ~ w, data=dat, type="b", lty=2, pch=1)
legend(0.5, 2.995, lty = c(1,2), pch = c(0,1), legend = legenda1)
```

Img. 1.2.4_15 – R-Graph source code
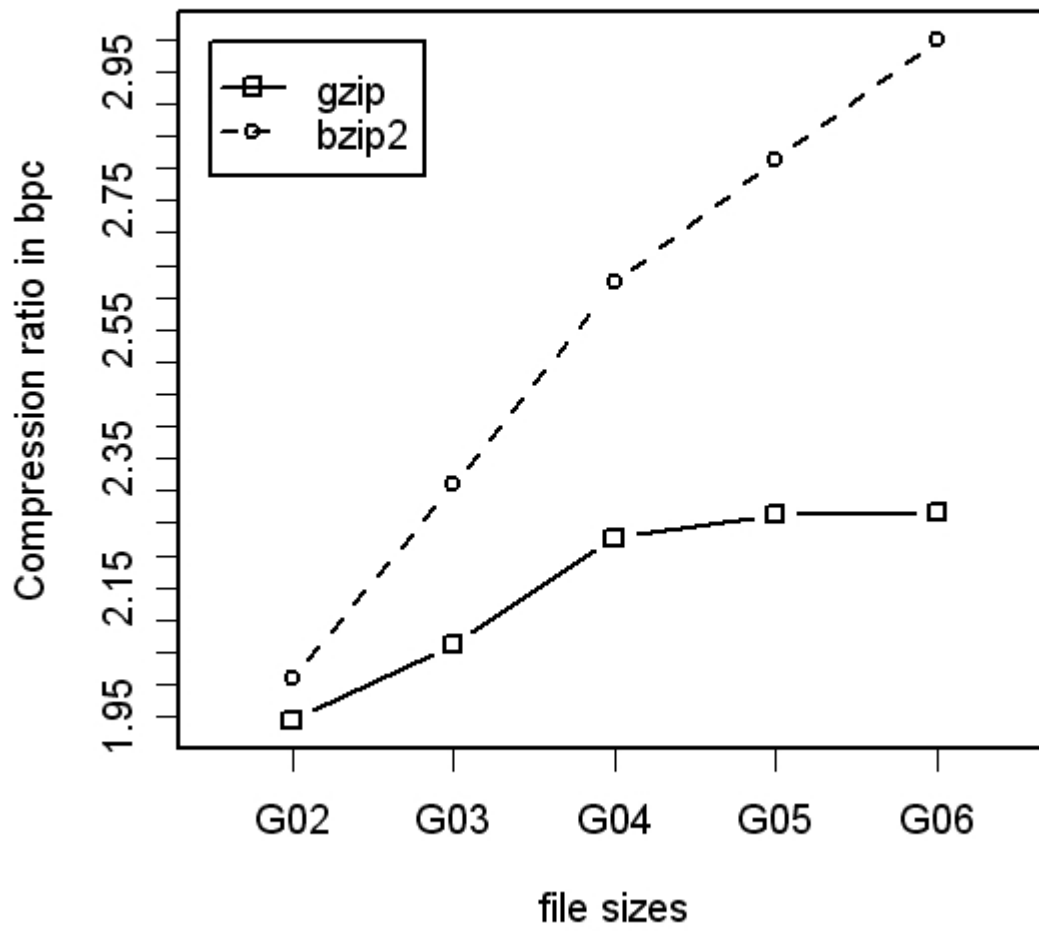
Img. 1.2.4_16 – R-Graph result

1.2.5. Report parameter groups

System displays tools, which help user to create own report from result data with additional parameters. (img. 1.2.5_1)
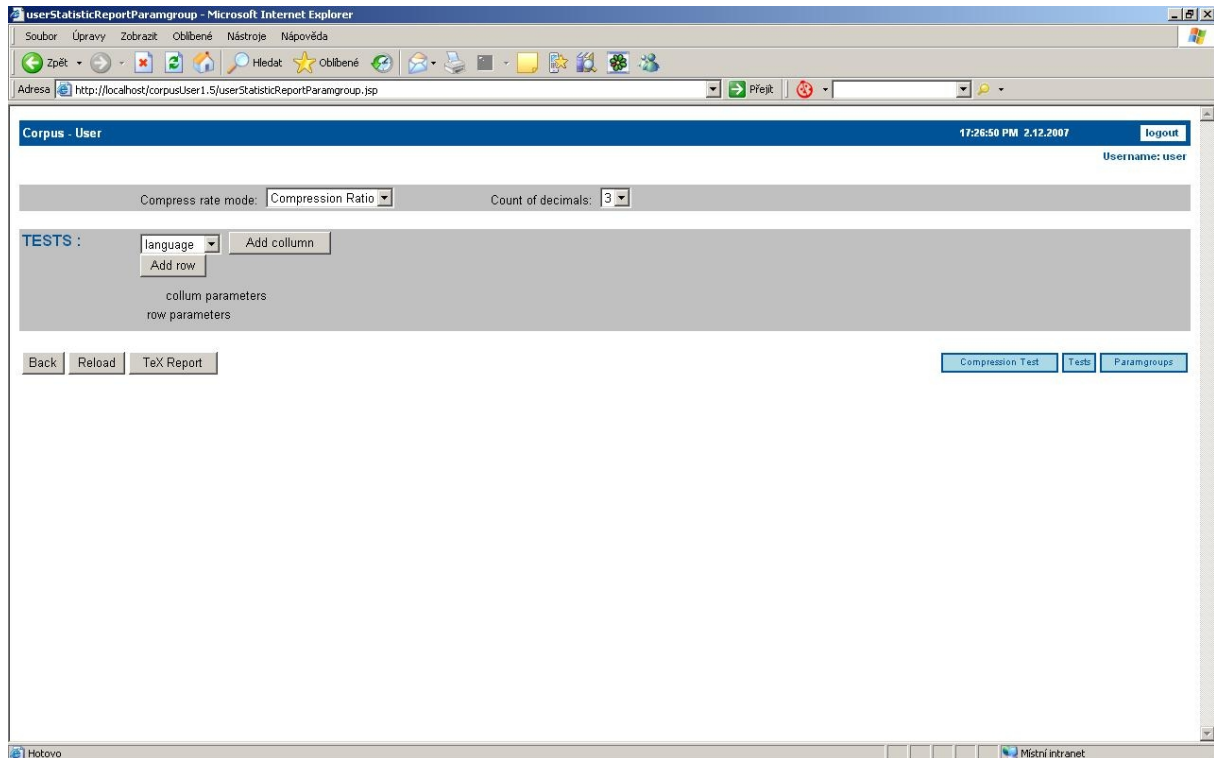
Img. 1.2.5_1

User operations for formatting report:

Displayed compression mode – user can choose two possibilities, how he see compression rate values, the first format is *Compression Ratio* and the second one is *Space Savings*. *Compression Ratio* represents rate of uncompressed size and compressed size. *Space Savings* noted as percentage value, is defined as the reduction in size relative to uncompressed size. (img. 1.2.5_2)

Count of decimals – user can choose how many decimals in compression rate values should be displayed (img. 1.2.5_3)
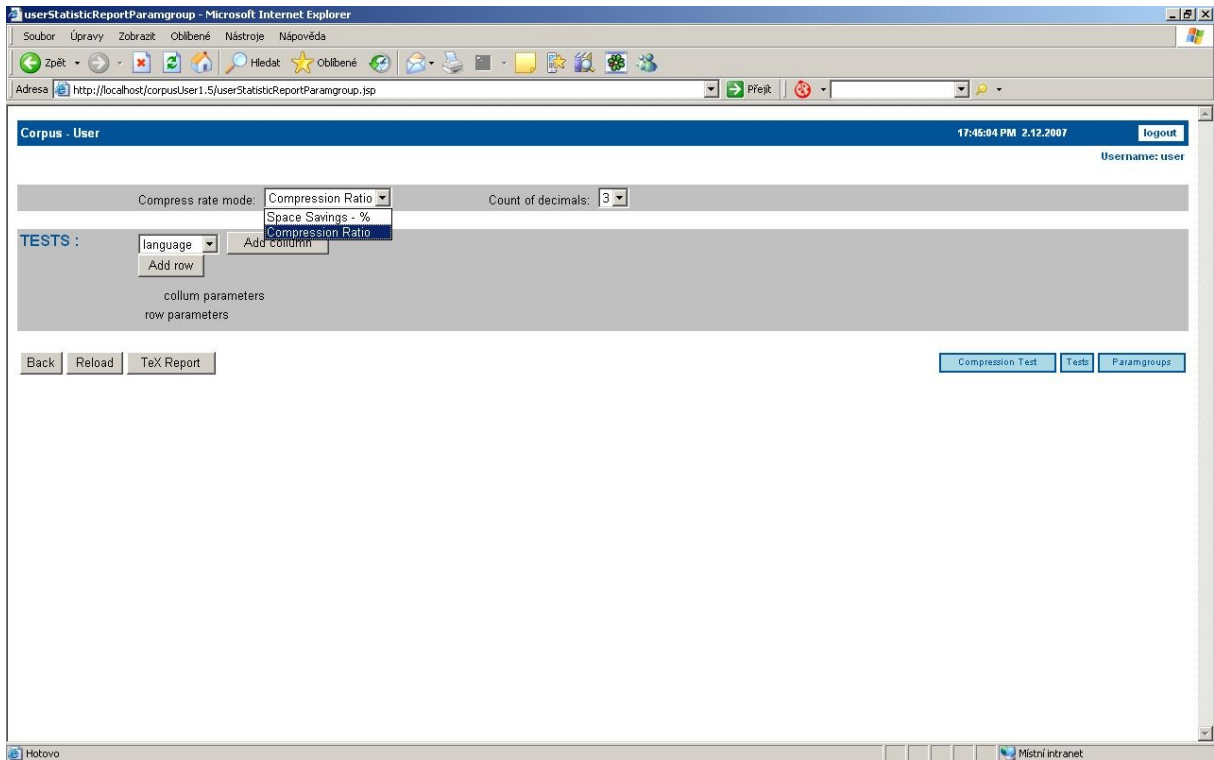
Img. 1.2.5_2
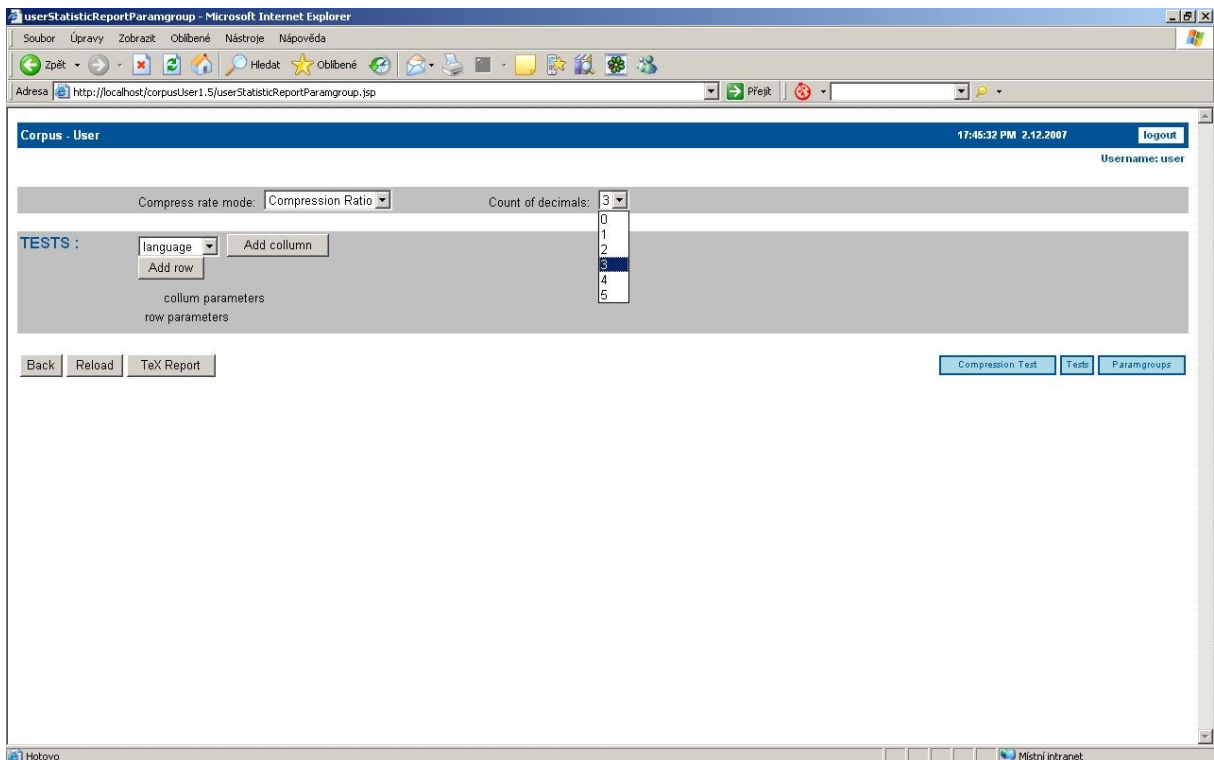
Img. 1.2.5_3

Table formatting report operations:

Add Column

User can choose one parameter group from list of parameter groups, then after pressing button *Add Column*, new column is added into report table, and this new column is associated with selected parameter groups. (img. 1.2.5_4)

System displays list of parameter values for added parameter group. First parameter value is taken from list. If user wants to specify other parameter value, he must choose it from the list. (img. 1.2.5_5)

User can remove column by pressing button *Delete* under the list of parameter values.

By associating column with specific parameter value, system takes for this column only test, which has got the same additional parameter in its preferences.
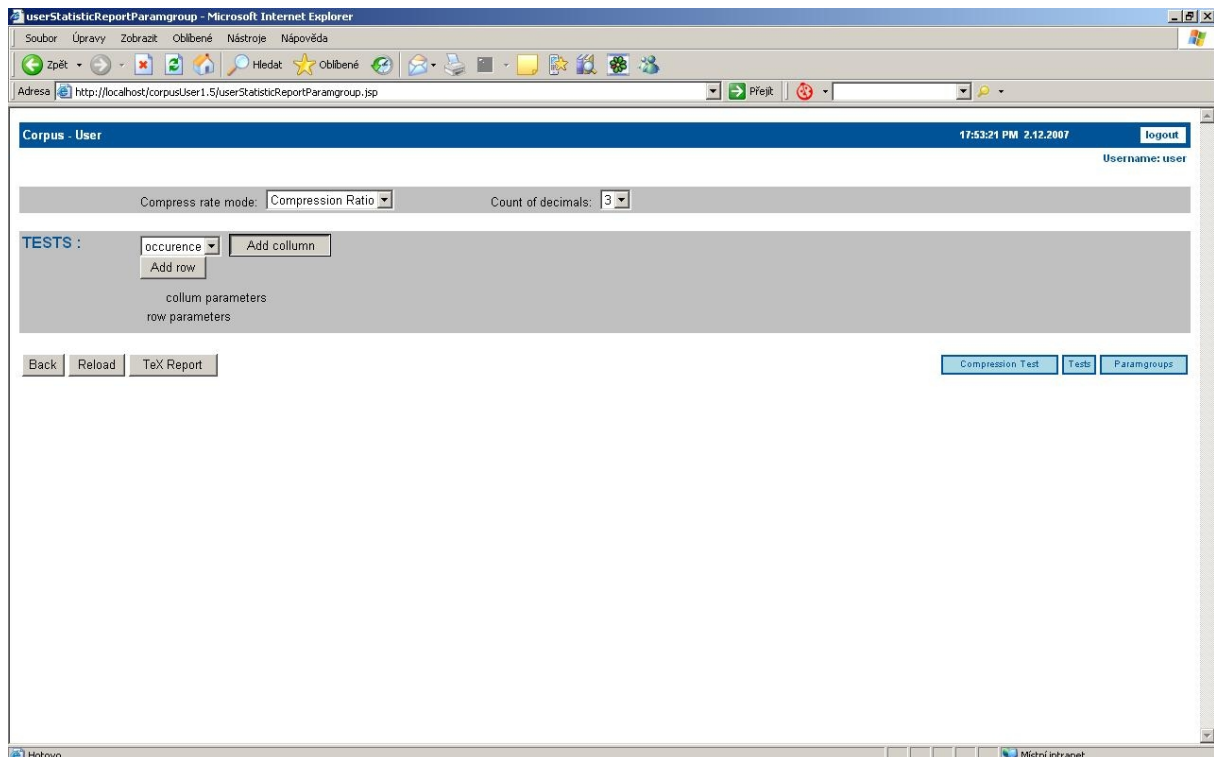
Img. 1.2.5_4
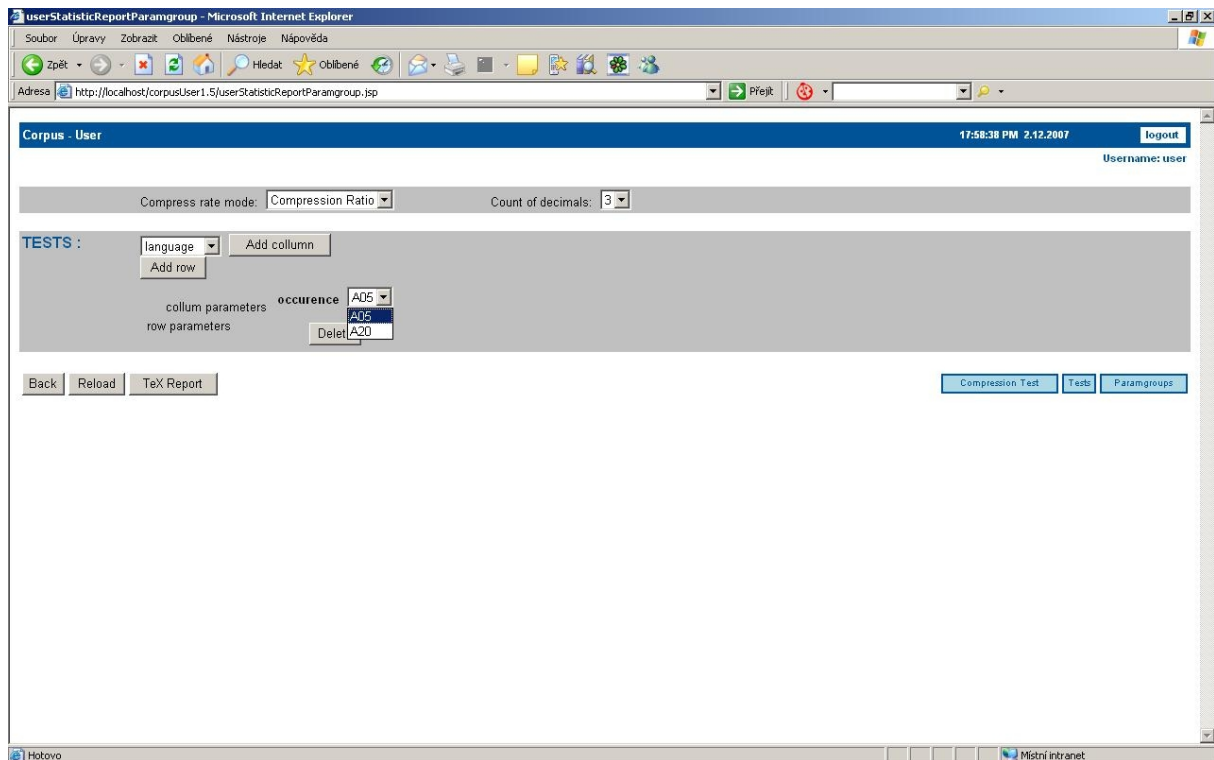
Img. 1.2.5_5

Add Row

User can choose one parameter group from list of parameter groups, then after pressing button *Add Row*, new row is added into report table, and this new row is associated with selected parameter groups. (img. 1.2.5_6)

System displays list of parameter values for added parameter group. First parameter value is taken from list. If user wants to specify other parameter value, he must choose it from the list. (img. 1.2.5_7)

User can remove row by pressing button *Delete* under the list of parameter values.

By associating row with specific parameter value, system takes for this row only test, which has got the same additional parameter in its preferences.

34
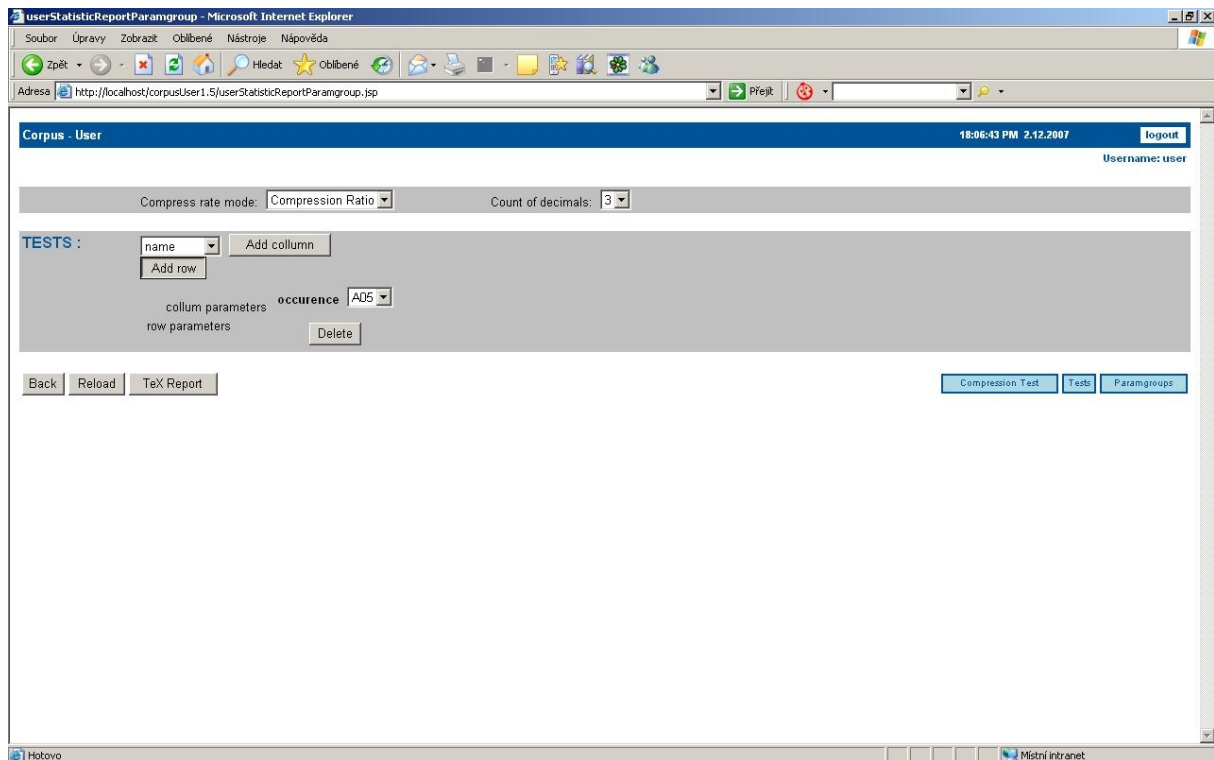
Img. 1.2.5_6

Img. 1.2.5_7

Reload

User can press button *Reload*, which runs process for loading result data, according to defined parameter values. If more tests are placed into one cell, because they have the same row parameter and column parameter, the result value for this cell is average value of their result data. If no test is associated with concrete cell, then system prints '—' into this cell.
(img. 1.2.5_8)
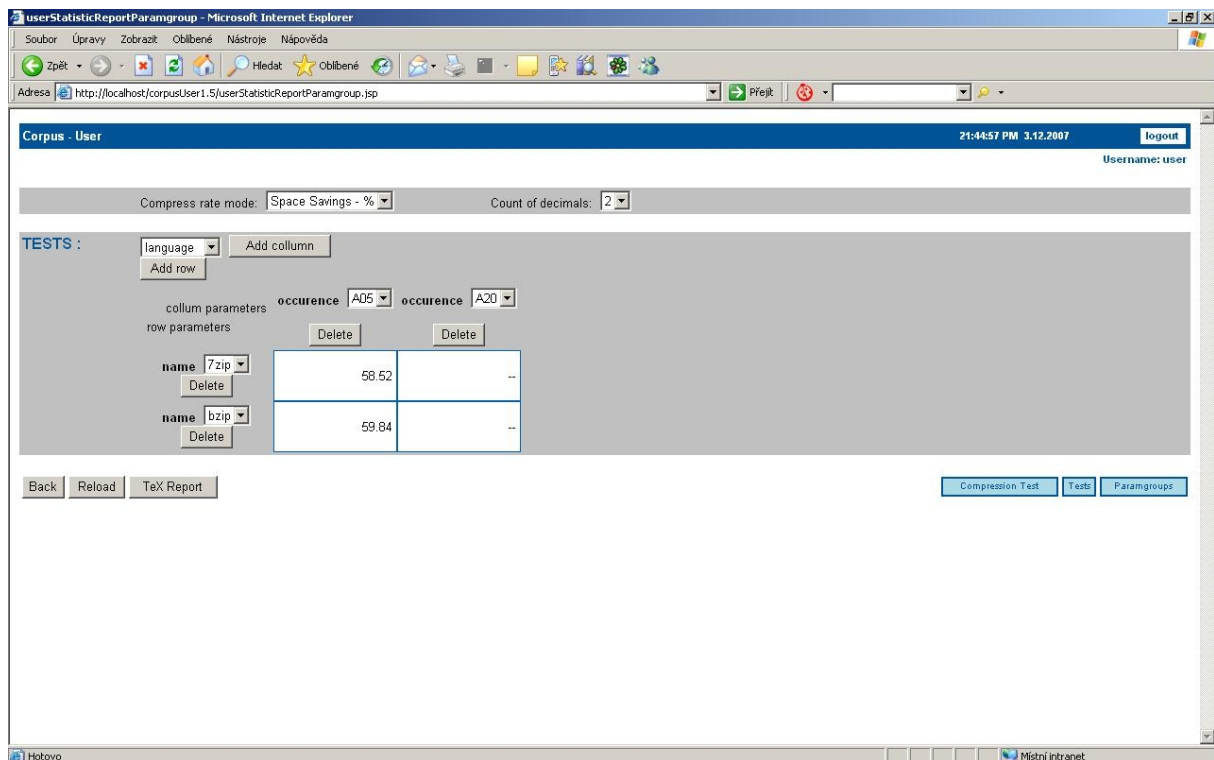
Img. 1.2.5_8

TeX Report

By pressing button *TeX report*, system generates *TeX* source code of formatted table, and user can download and save this *TeX* source code into his local disc. Then he can use this downloaded source code into his *TeX* document. (img. 1.2.5_9 – 1.2.5_11)
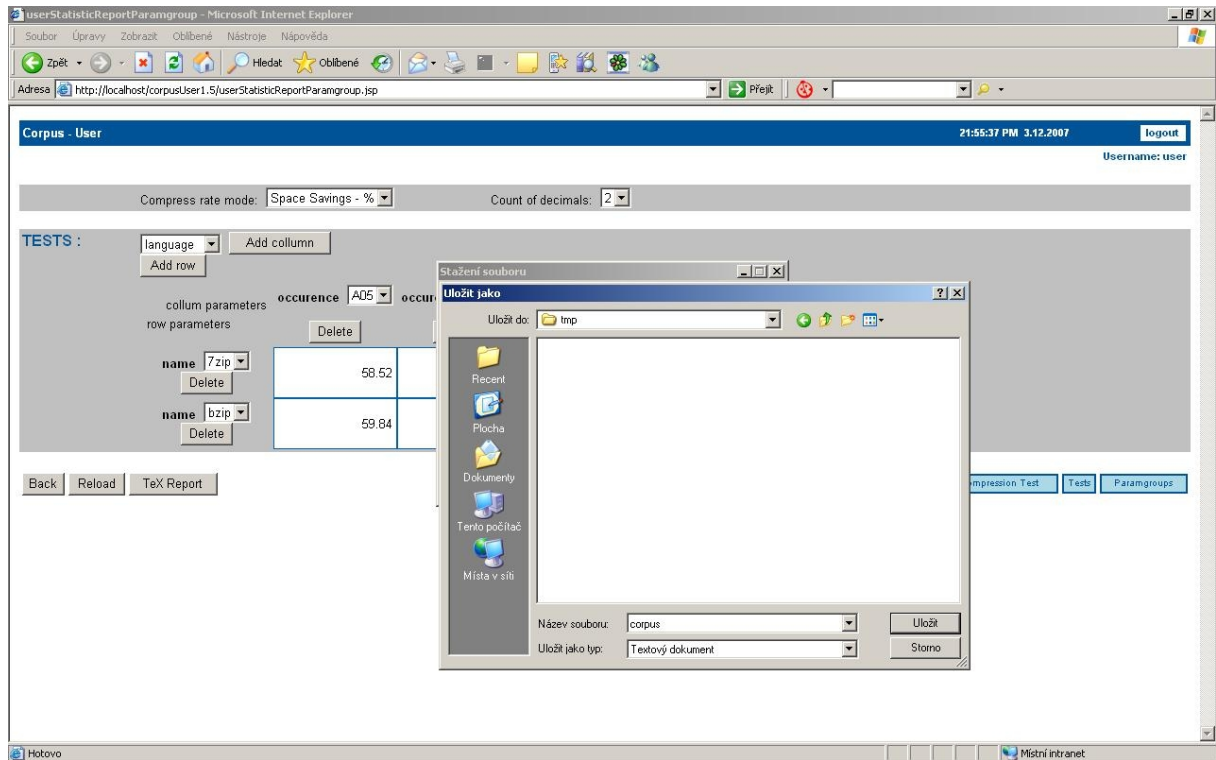
Img. 1.2.5_9

Img. 1.2.5_10 – TeX source code
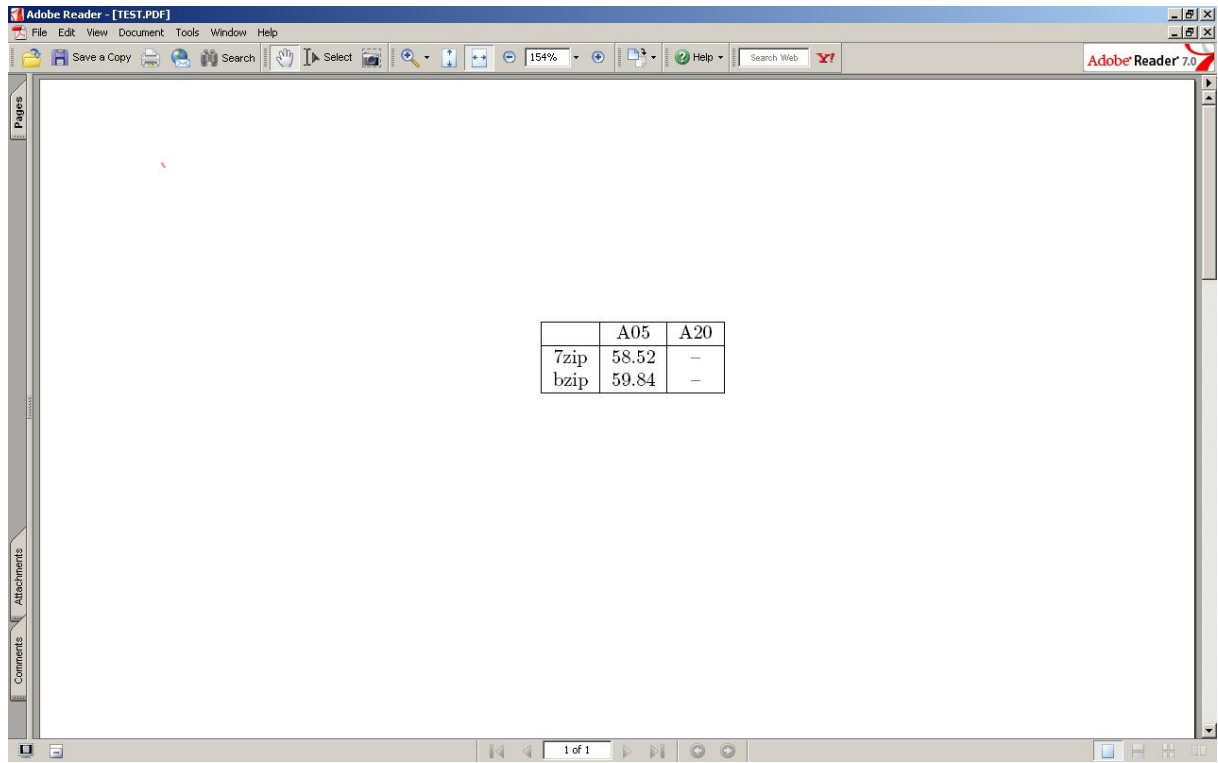
Img. 1.2.5_11 – TeX result in PDF file

1.3. User tests results

Functionality for browsing in list, which contains all useful information about each test of compression programs, which were processed yet.

Page with table containing basic info of all tests is displayed at first.

1.3.1. Basic info

In *Basic info* page, systems displays table with all tests, which were done yet. Each row represents one test. Test's information contains:

Test ID – unique number of test, set up by system
Test name – name of test, set up by user
Server ID – unique number of server, where the test was done, set up by system
Status – status of test result
Time – time when test start
Corpus ID – unique number of *Corpus hierarchy*, number represents modification of *Corpus hierarchy*, on which was test done
Category – path in *Corpus hierarchy*, path represents subpart of hierarchy, on which was test done

(img 1.3.1_1)

Img. 1.3.1_1

39

User operations with each test (row):

Download – by clicking this button, user can download compression program, on which was this test processed. All files, which were uploaded and required for test, are packed into zip file, and user downloads this zip file. To download compression program user must define destination on his local driver (by using popup window with file manager). (img. 1.3.1_2)

Delete – by clicking this button, user removes test from database, he removes it completely, that means with whole preview data, result data and files required for test. (img. 1.3.1_3)
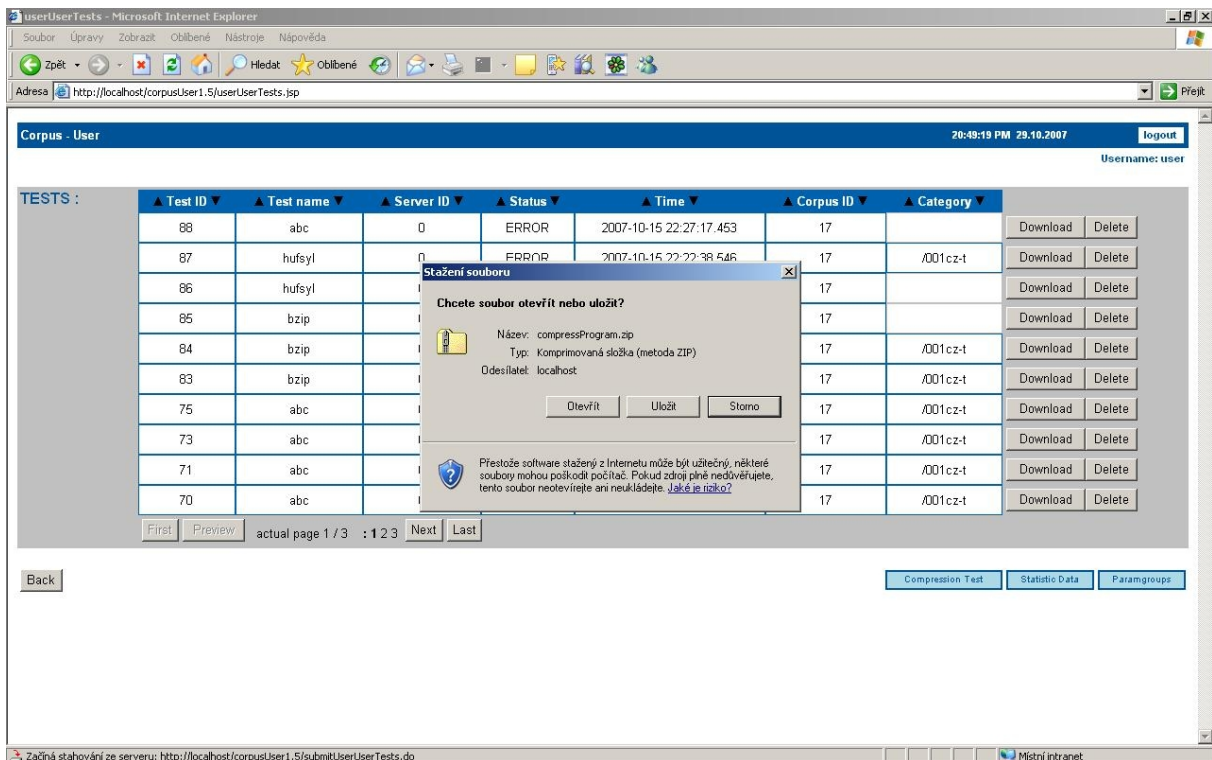
Img. 1.3.1_2
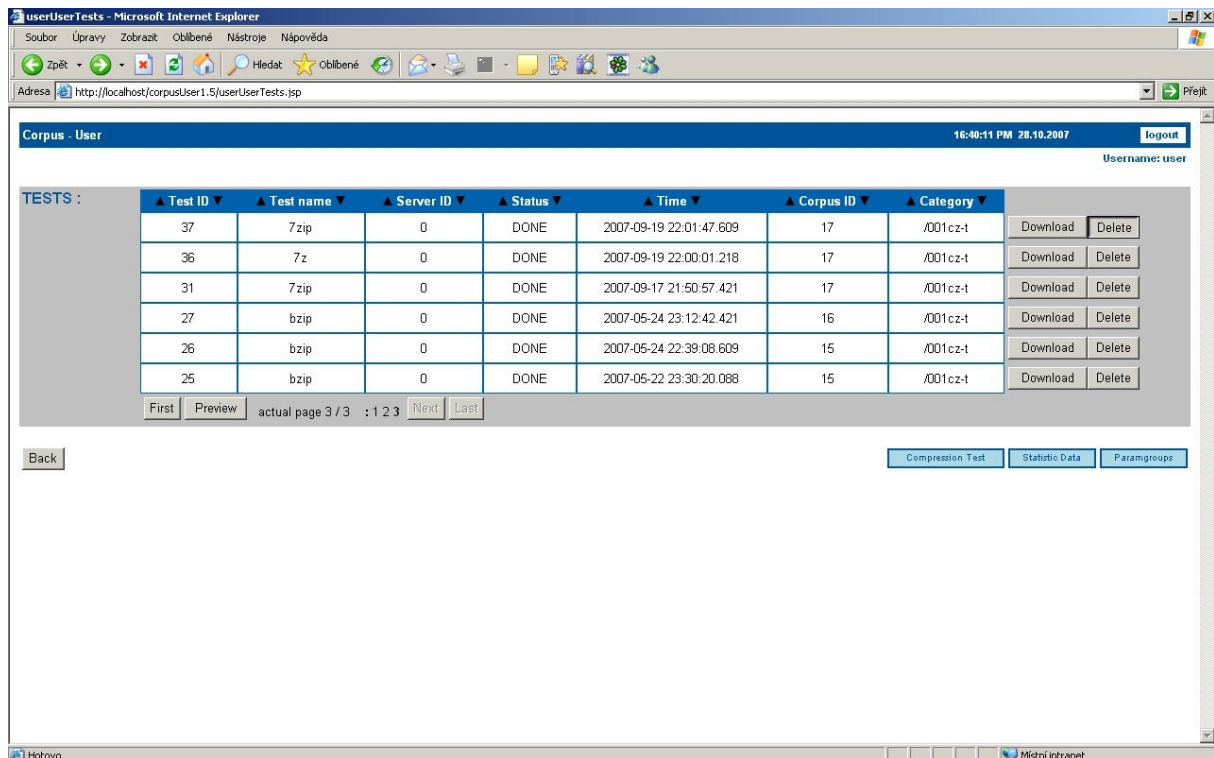
Img. 1.3.1_3

Table operations:

Sort test list according to selected column – by clicking on headline of column, user can sort list of tests by each column by ascending or descending order. Not using arrow up or down, system sorts list by ascending order by default. (img. 1.3.1_4 – img. 1.3.1_5)

Choose page – user can choose page, which he wants to display by:
   - pressing button *First* – system displays first page of tests
   - pressing button *Preview* – system displays preview page to actual page
   - pressing button *Next* – system displays next page to actual page
   - pressing button *Last* – system displays last page of table
   - clicking on number of page – system displays exact page

(img. 1.3.1_6 – img. 1.3.1_7)
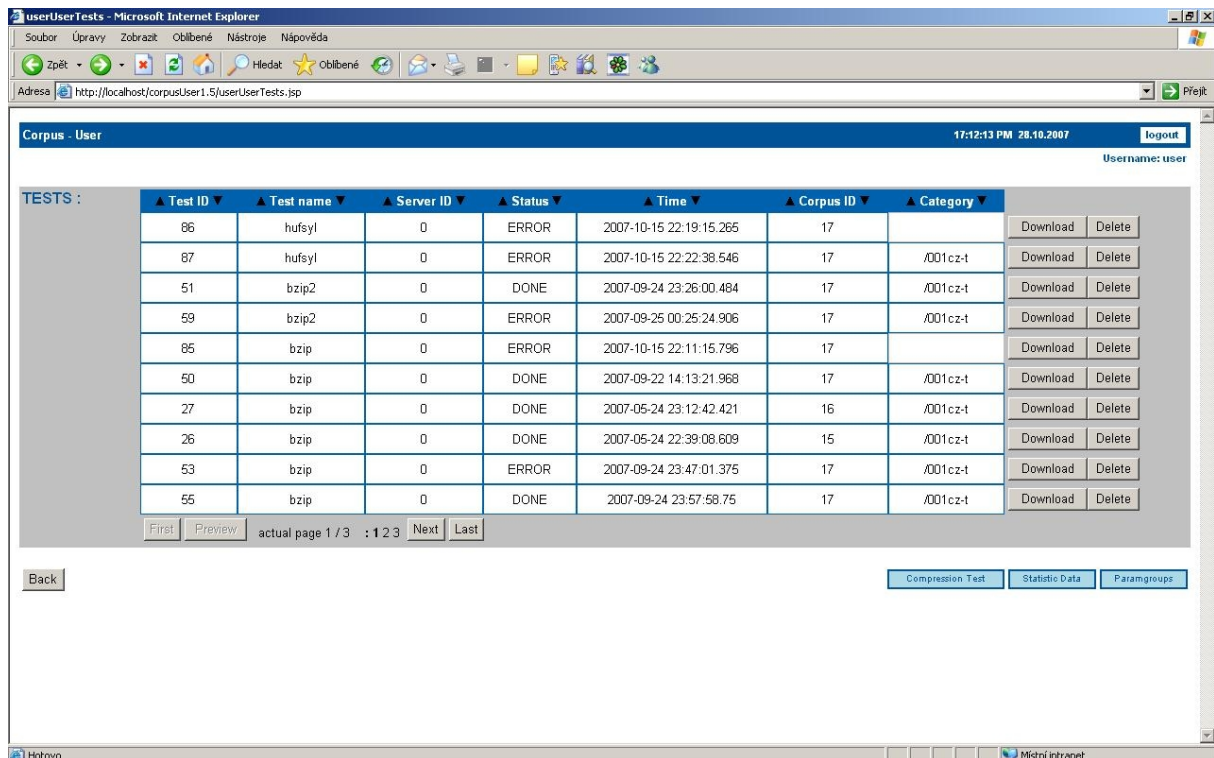
Img. 1.3.1_4 – sort desc by *Test name*
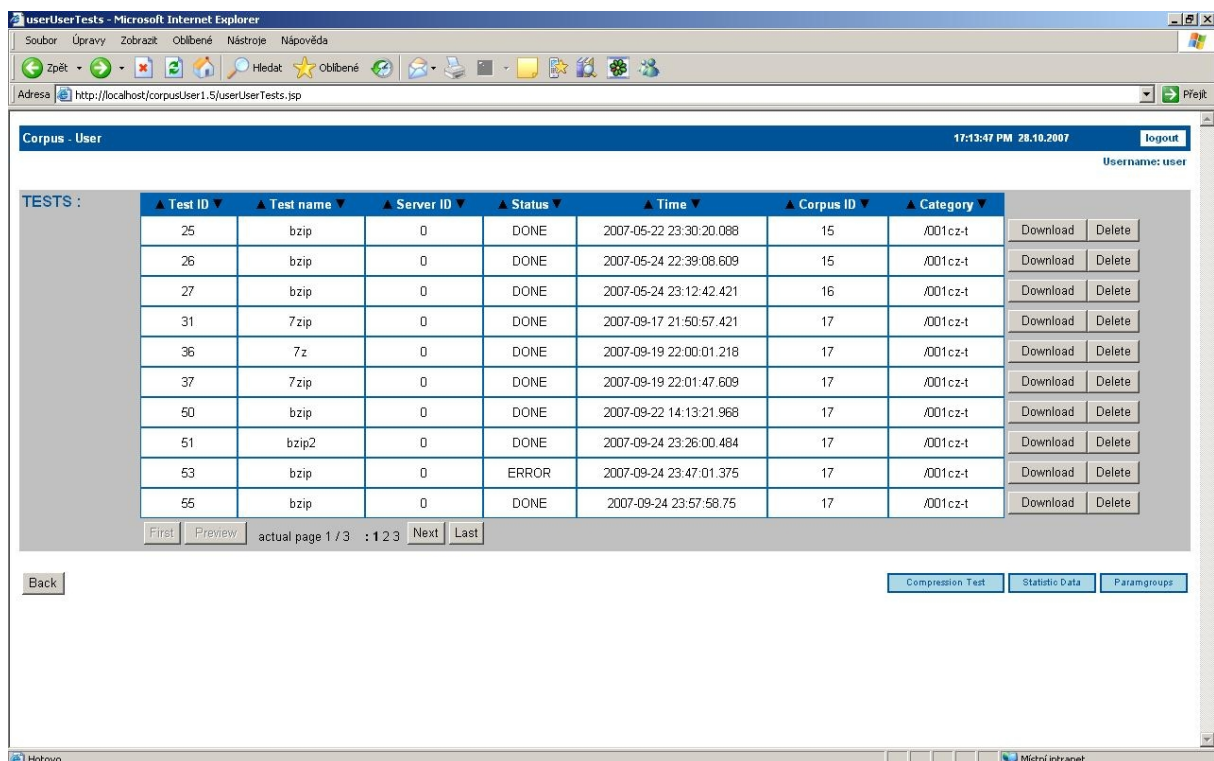
Img. 1.3.1_5 – sort by *Time*
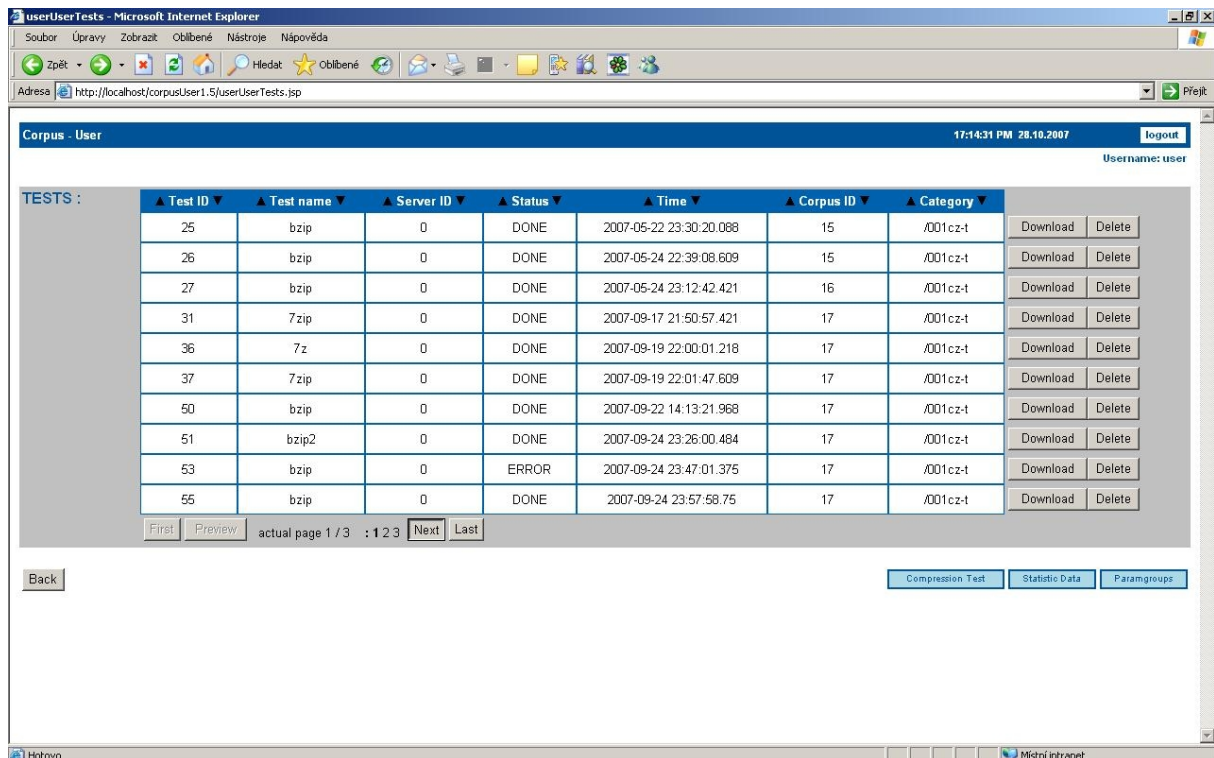
Img. 1.3.1_6 – next page
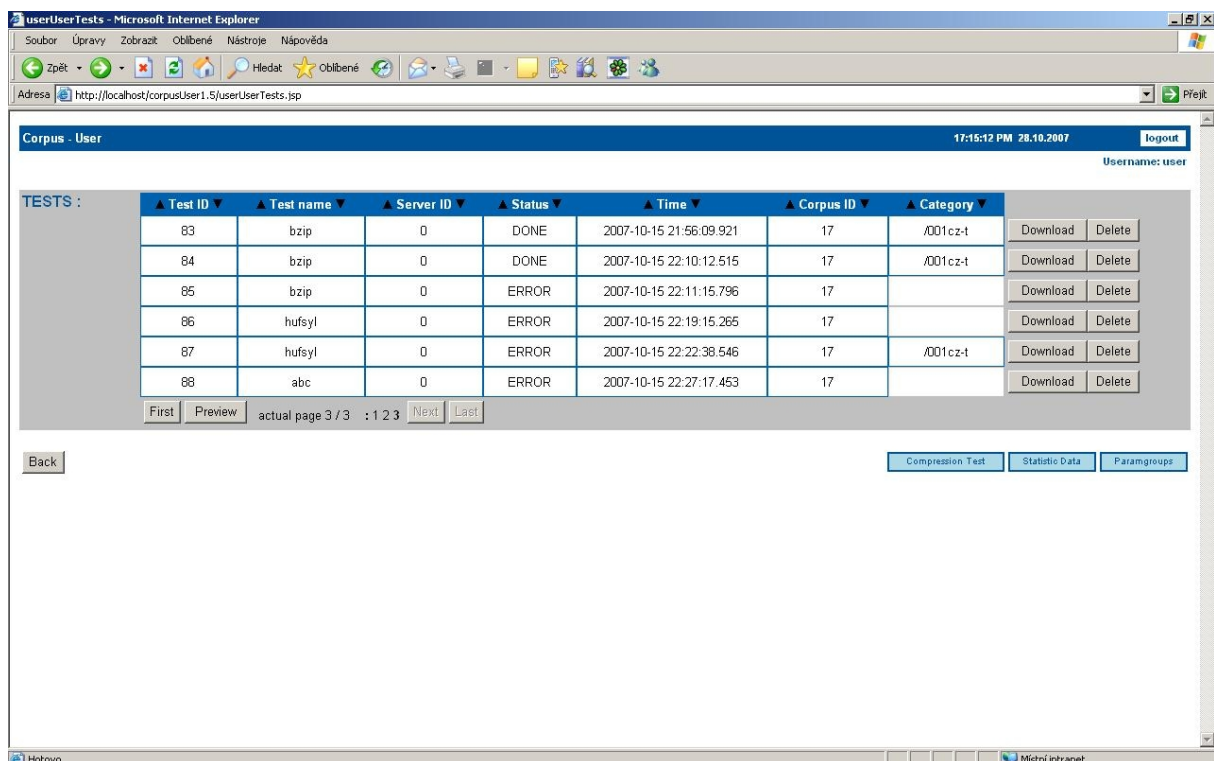
Img. 1.3.1_7 – exact page num. 3

1.4. Parameter groups manager

System functionality for defining user's additional parameter values of tests. This functionality is only useful for creating special reports, which user wants to create from his tests associated with his own additional parameters.

1.4.1. Parameter groups

System shows page *Parameter groups* with by user already defined *Paramgroups* and *Paramgroups values.* Each value belongs to only one group, so groups do not share parameter values. But more groups can contain value with the same name, but this occurrence does not mean, that these two values, from different groups, are the same for *Corpus* system representation.

On the page, there are displayed only values, which belong to selected group. To define new parameter value, user can add new value to already existing group, or he must set up group of values at first. (img. 1.4.1_1)

Img. 1.4.1_1

44
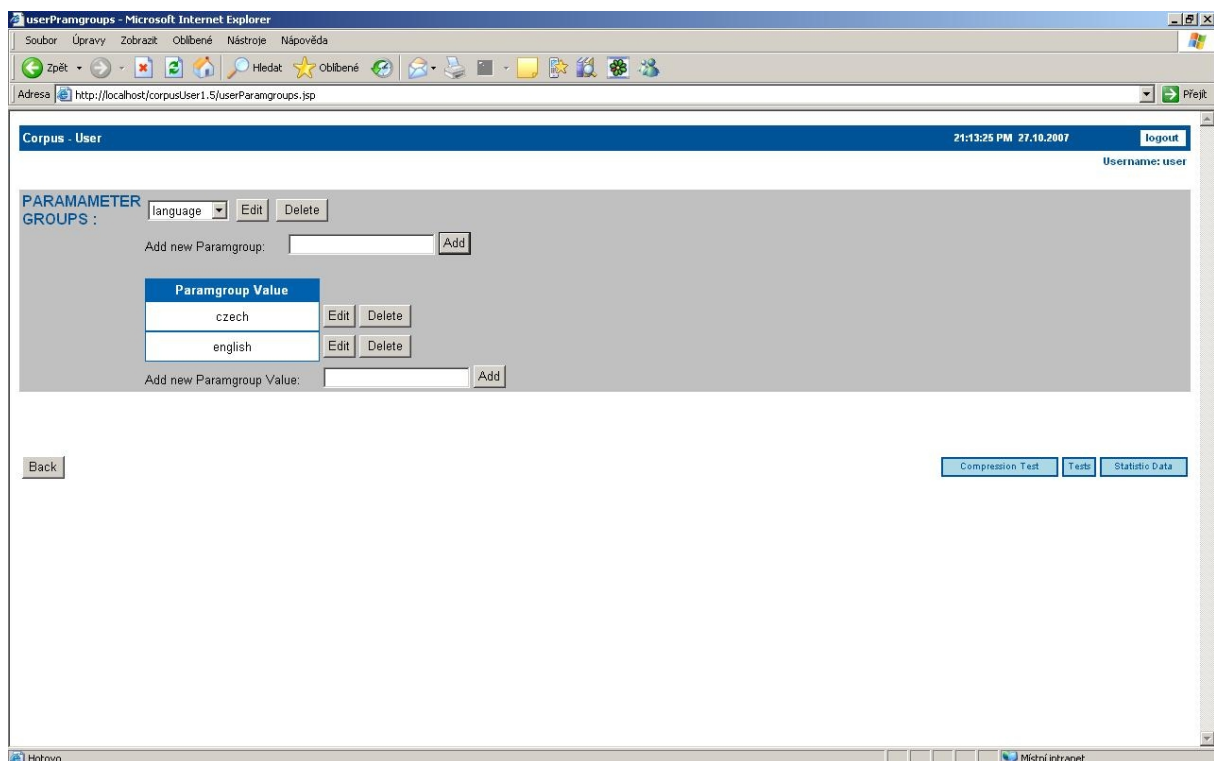
1.4.1.1. Adding new group

For adding new group of parameter values, user has to fill in name of new group into input field. Max length of group name is 20 chars. System does not allow more than one same name of group. Then press button *Add* next to the input field. (img 1.4.1.1_1)
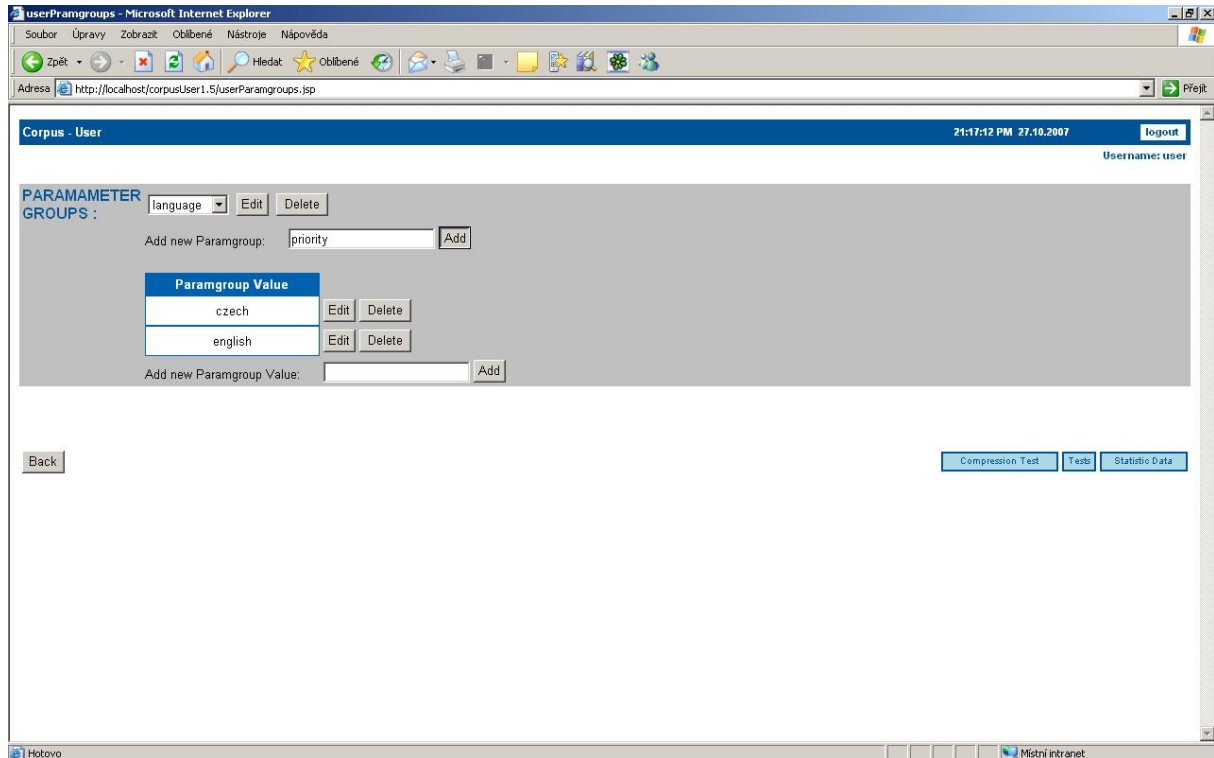
Img. 1.4.1.1_1

1.4.1.2. Edit group

User can only edit selected group, so he must choose group, which he wants to modify, from list of groups. This feature allows modification of the name of chosen group. Then, after pressing button *Edit*, placed next to select list of groups, system displays editable input field and for confirmation *Save* button. (img 1.4.1.2_1)

Img. 1.4.1.2_1

46
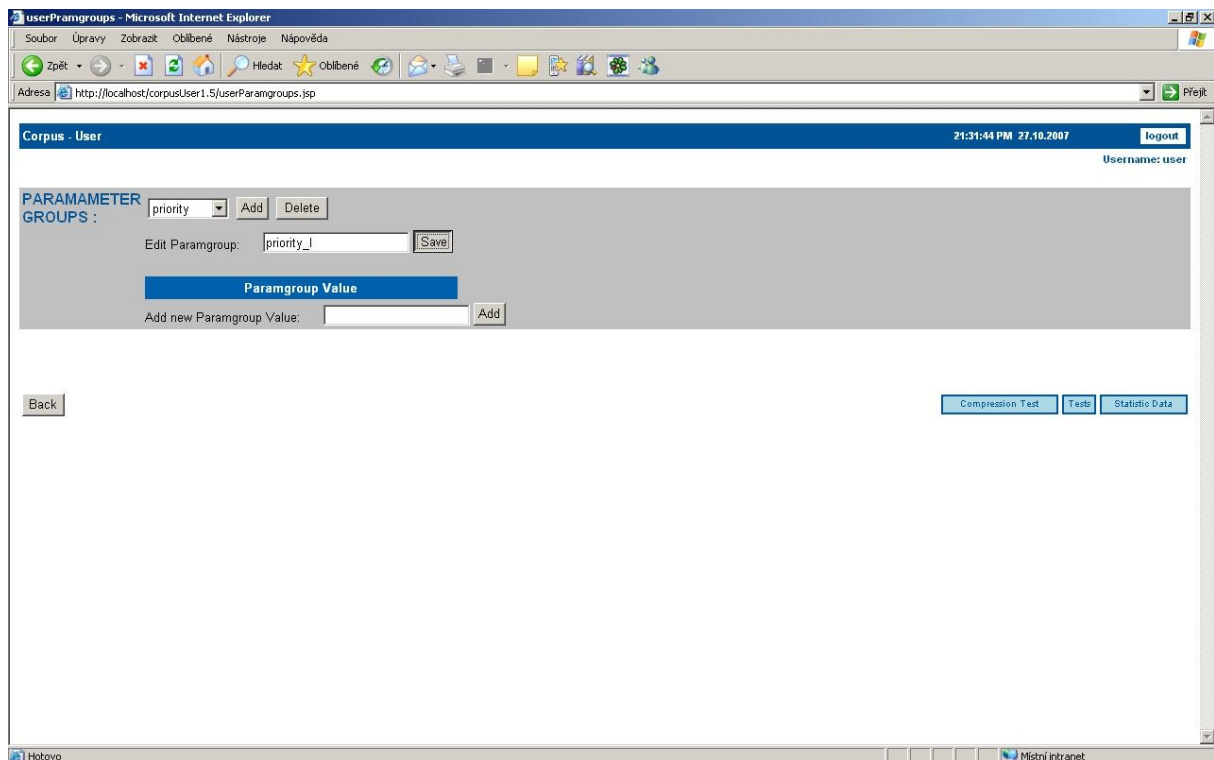
1.4.1.3. Remove group

For deleting parameter group, user must choose required group from list of groups and then after pressing button *Delete*, placed next to the select list of groups, chosen group is removed with all parameter values, which the removing group contains. (1.4.1.3_1)
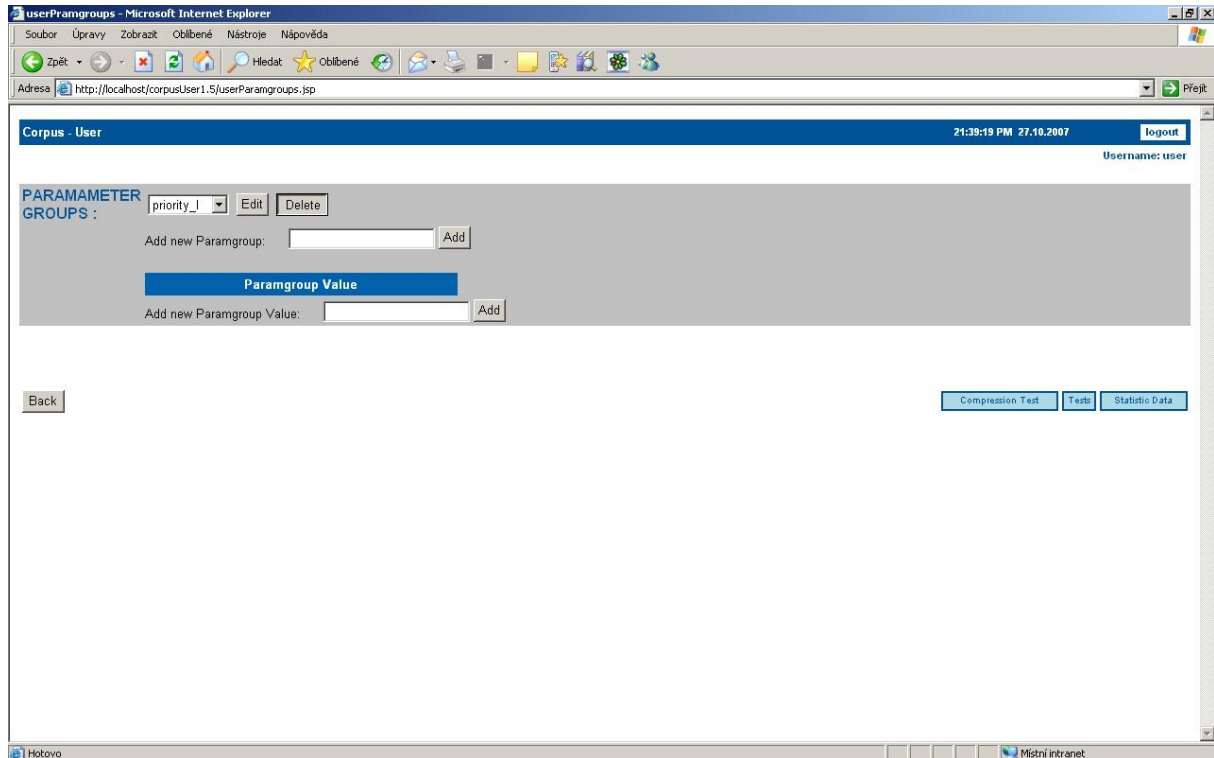
Img. 1.4.1.3_1

1.4.1.4. Adding parameter value

Parameter value operations are only possible, when at least one parameter group exists, because value can not exist single, without association to group. Values are added into groups, chosen by user. System does not allow more than one value, with the same name, for one group. Max value length is 20 chars. For adding new parameter value, user has to fill in name of new value into input field, displayed in table of parameter values for selected group. (img 1.4.1.4_1)
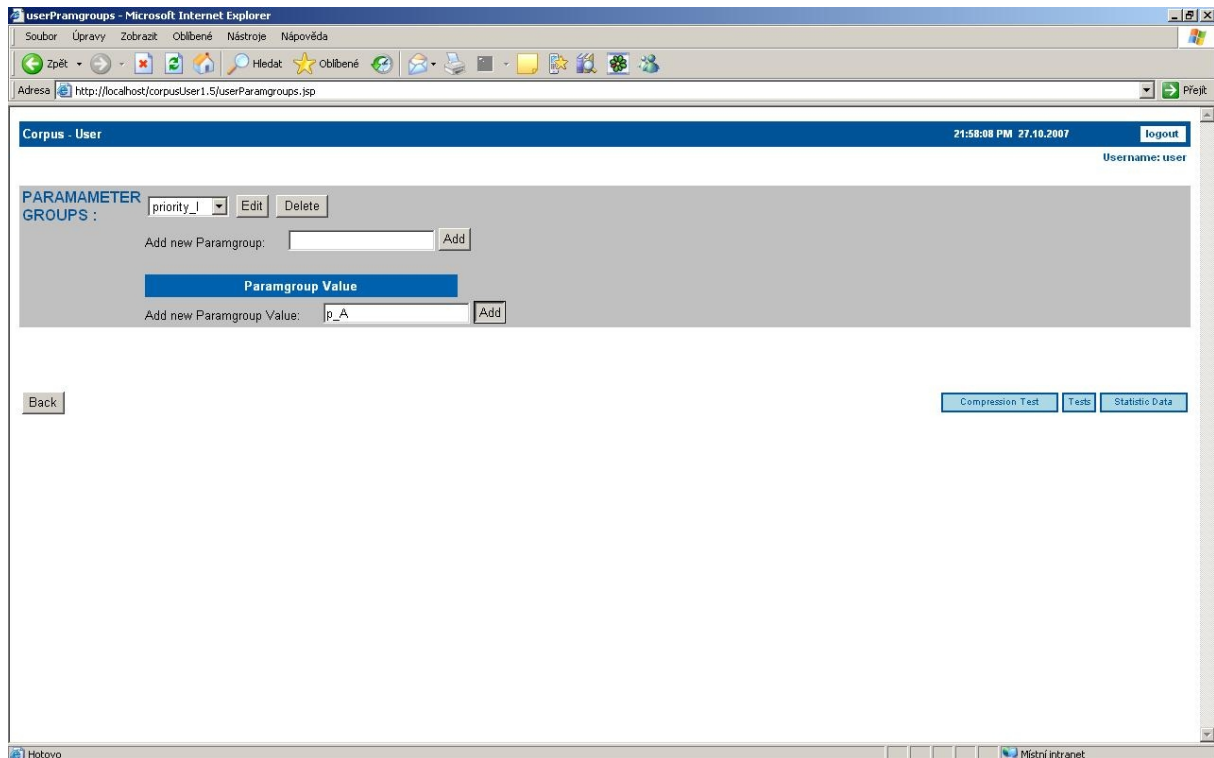
Img. 1.4.1.4_1

1.4.1.5. Edit value

User can only edit selected value, so he must choose value, which he wants to modify, from table of values. This feature allows modification of the name of chosen value. Then, after pressing button *Edit*, placed in the row, of chosen parameter value, system displays editable input field in the table and *Save* button for confirmation. (img 1.4.1.5_1)
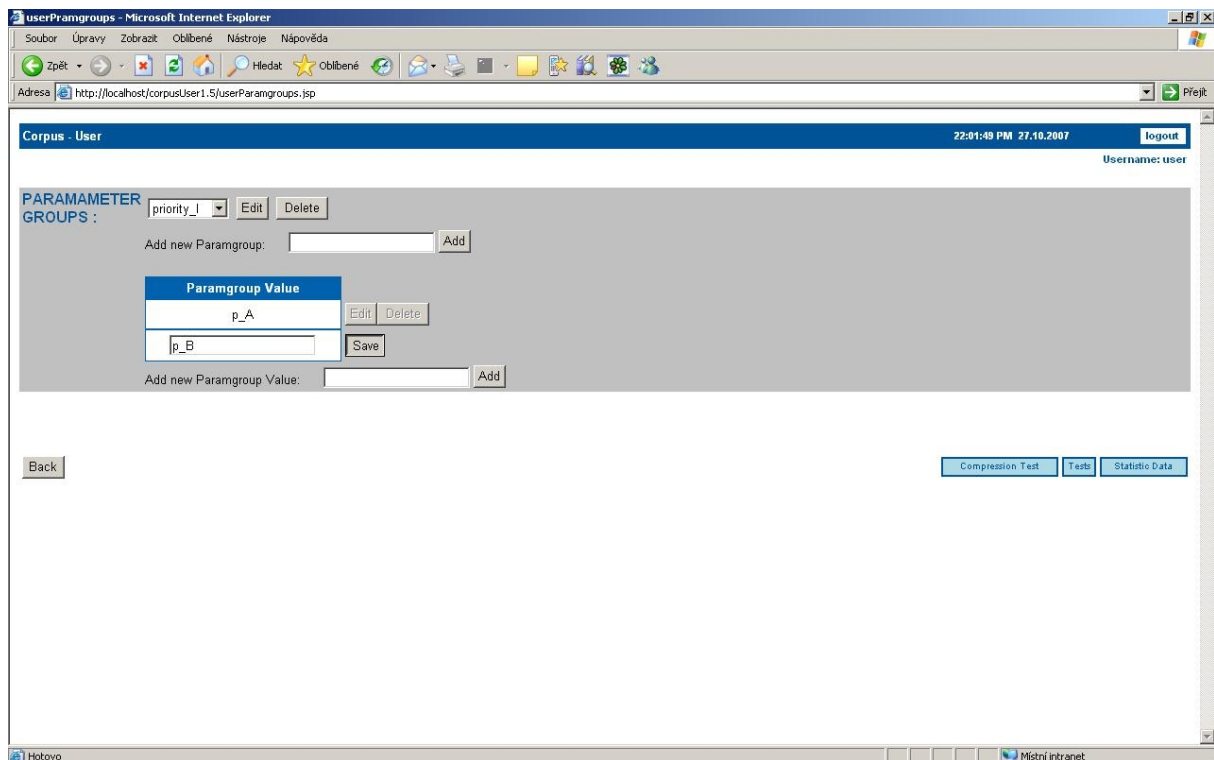
Img. 1.4.1.5_1

1.4.1.6. Remove value

For deleting parameter value, user must choose required value from table of values and then after pressing button *Delete*, placed in the row for selected value, chosen value is removed group. (1.4.1.6_1)
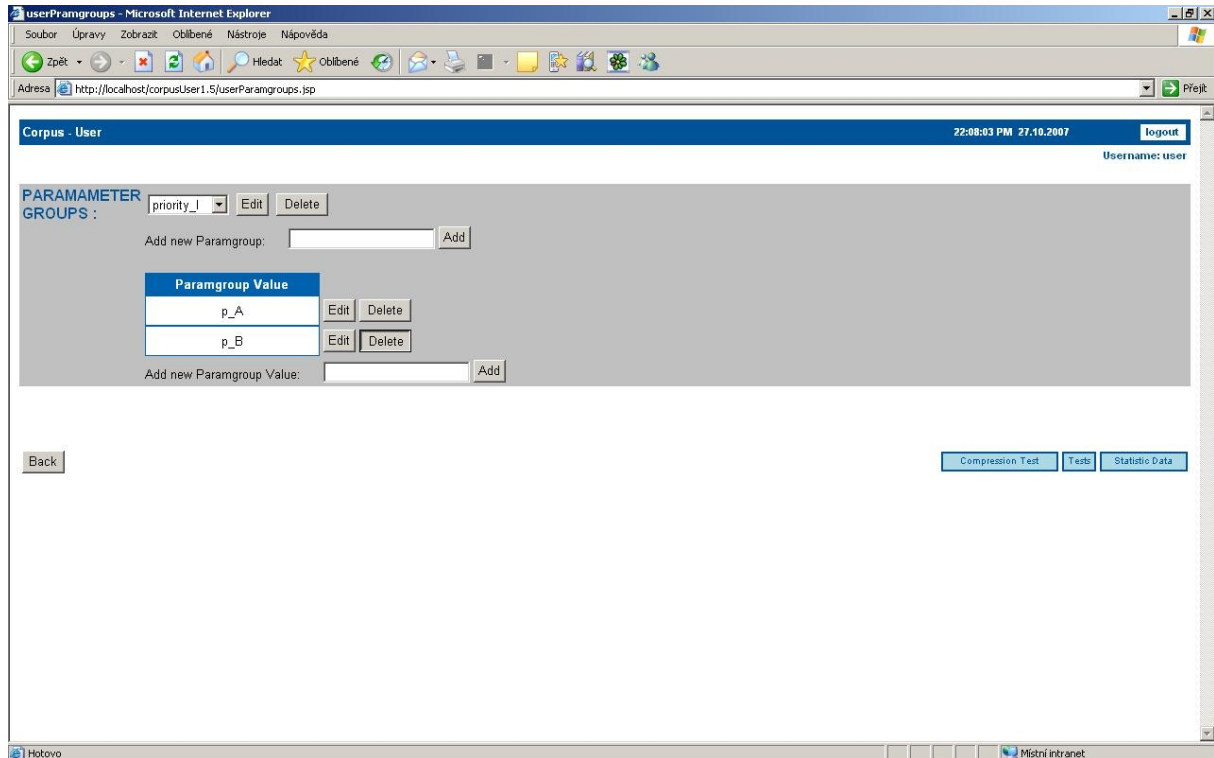
1.4.1.6_1

System validates value and group names, which are filled into input fields. If any error occurs, system prints error report.

1.5. Registration

System functionality, which helps user, to fill in the request form for getting access to *Corpus System.* (img 1.5_1)

User has to fill in some basic information:

Name – user's first name, this field is mandatory, allowed maximum 20 char-length string
Surname – user's surname, this field is mandatory, allowed maximum 20 char-length string
Login name – user's nick name, this field is mandatory, allowed max 20 char-length string

E-Mail – user's contact email, this field is mandatory, allowed max 20 char-length string
Telephone – user's telephone, this field is optional, allowed length maximum 20

Organization – name of user's organization, optional field, name of organization can have maximum 20 chars
Sex – user's sex
State – user's place of living, optional field, allowed max 20 char-length string

After filling in user's basic information, request form is sent to administrator of *Corpus System* and then, if request is applied, administrator will send email, that contains necessary data for login into *Corpus System*.

Img. 1.5_1

51

2. Corpus Admin

Corpus Admin is the second web part of the whole *Corpus System*. It is designed only for administrators, who want to use tools for creating *Corpus hierarchy,* who want to associate *Corpus hierarchy* with useful data and administrate user accounts. *Corpus Admin* web part is web GUI, which allows easy administrational work almost of the whole *Corpus System*.

Corpus Admin starts with *Login page*, which is the first step to get into the application. Admin must have confirmed access to the application. (img. 2_1)
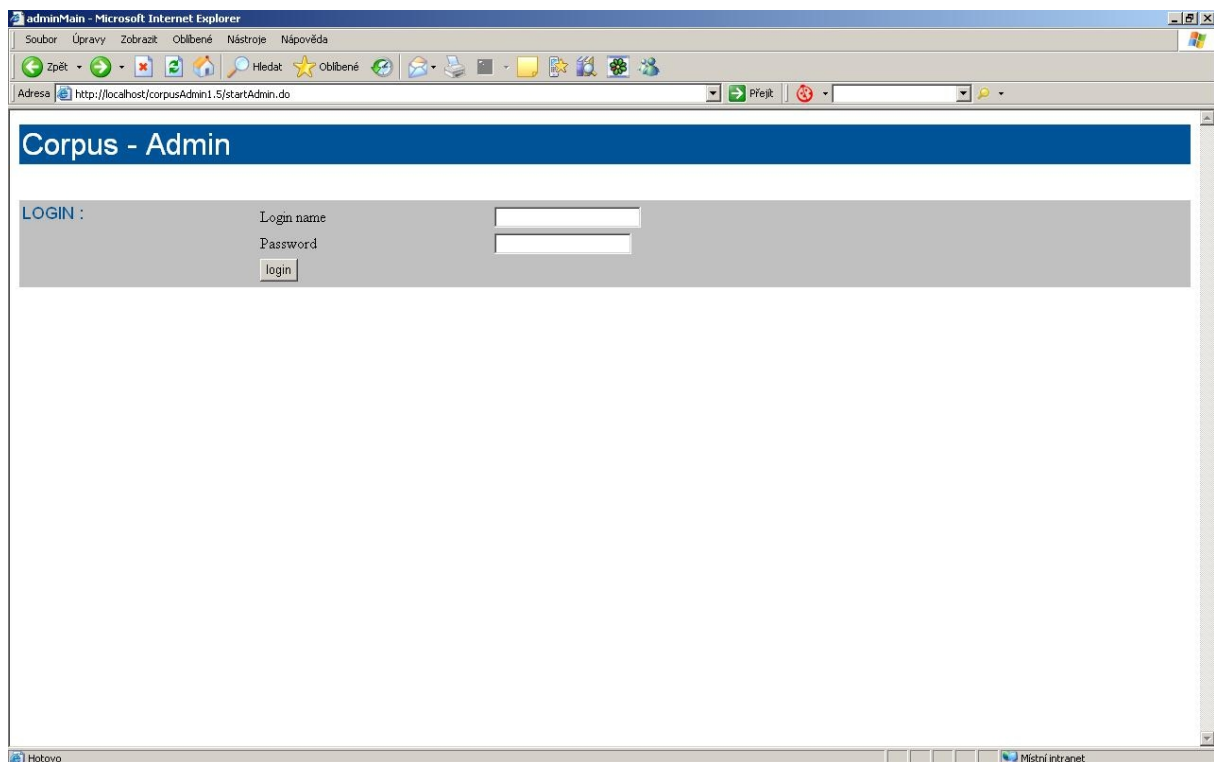
Img. 2_1

After entering the application, system shows *Main menu* page, which contains options:
Corpus hierarchy – tools for defining *Corpus hierarchy*
File / Data transfer – tools for browsing through file data, which are associated with specific part of *Corpus hierarchy*
Statistic data comparison – tools for browsing through result data of user's tests, compare them and making report tables, which contains data, specified by admin
User tests results – tools for browsing through all completed or terminated compression tests, which were done by user
Users – tools for browsing through all the user's accounts

Registration – tools for accepting or declining of any new user request for getting access into the system

Locks – tools for browsing through system locks, and their activation or deactivation

(img. 2_2)
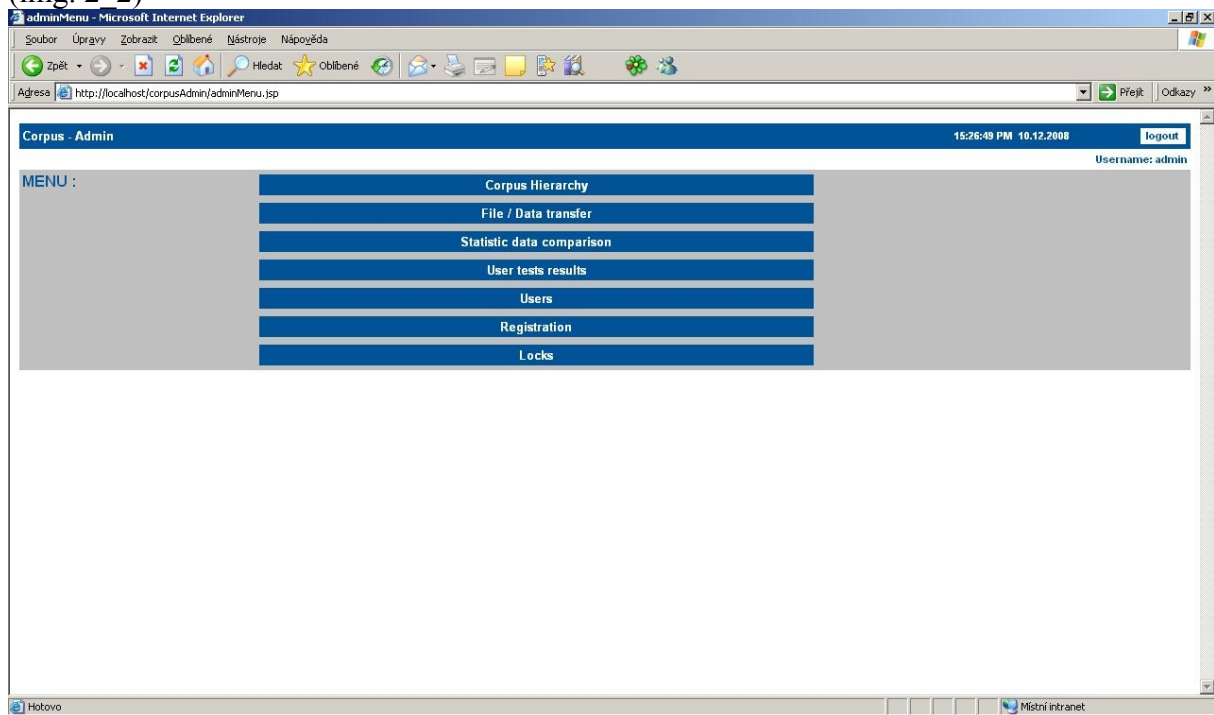
Img. 2_2

2.1. Corpus Hierarchy

System functionality for defining *Corpus hierarchy* structure.

Firstly system shows page with structure of the latest active *Corpus hierarchy.*

2.1.1. Corpus hierarchy structure

This page contains the latest active structure of *Corpus hierarchy*. The structure is designed in XML and has got tree structure - one root element (CORPUS_HIERARCHY), knot elements (CATEGORY) and on the lowest level, there are leaves (DATA). Leaves represent specific data associated with this hierarchy. (img. 2.1.1_1 – 2.1.1_2)

Admin operations with *Corpus hierarchy*:

Edit – admin can modify hierarchy structure, after pressing button *Edit hierarchy*, system displays page, where admin can directly modify hierarchy structure and save it

Download – admin downloads XML file with the whole hierarchy into his local disc, where he can also modify the hierarchy structure

Upload – admin can upload his hierarchy structure from his local disc into server, he must define the location of XML file on his local disc, after uploading hierarchy, system validates it and displays page, where he can save it that means, that uploaded hierarchy become active for the whole *Corpus system* (img. 2.1.1_3)
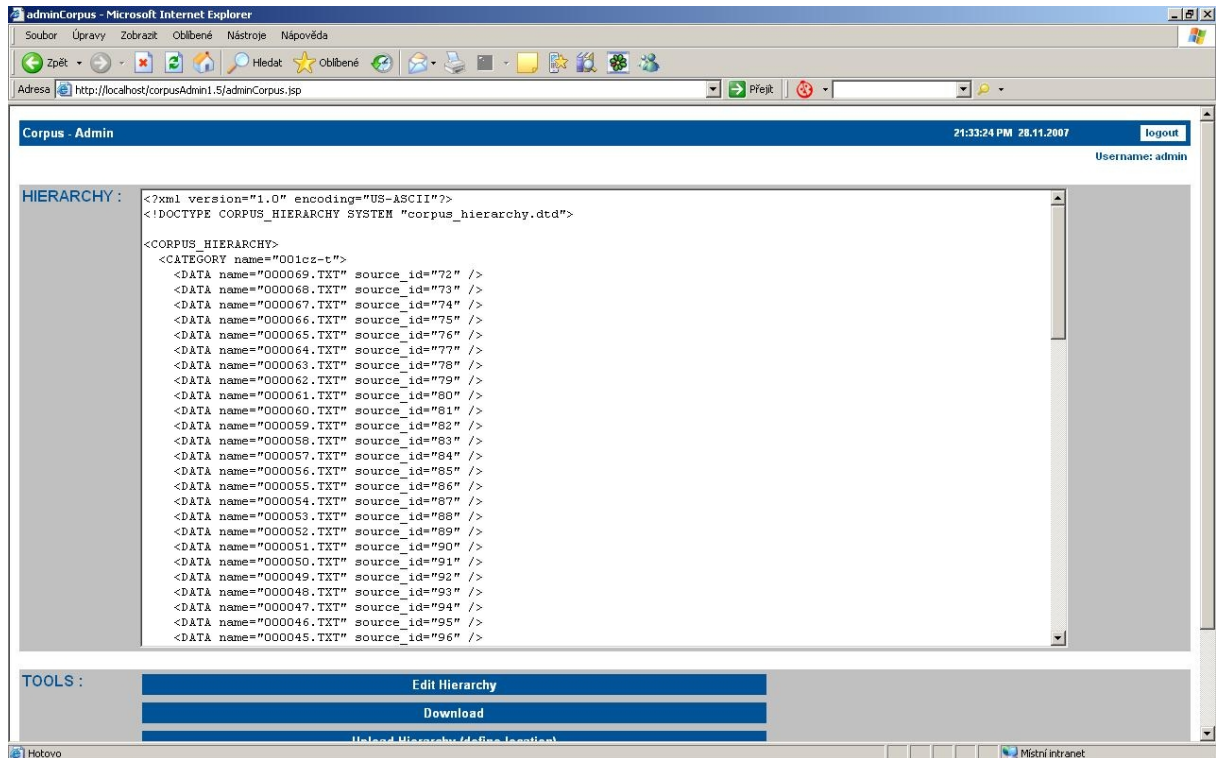
Img. 2.1.1_1
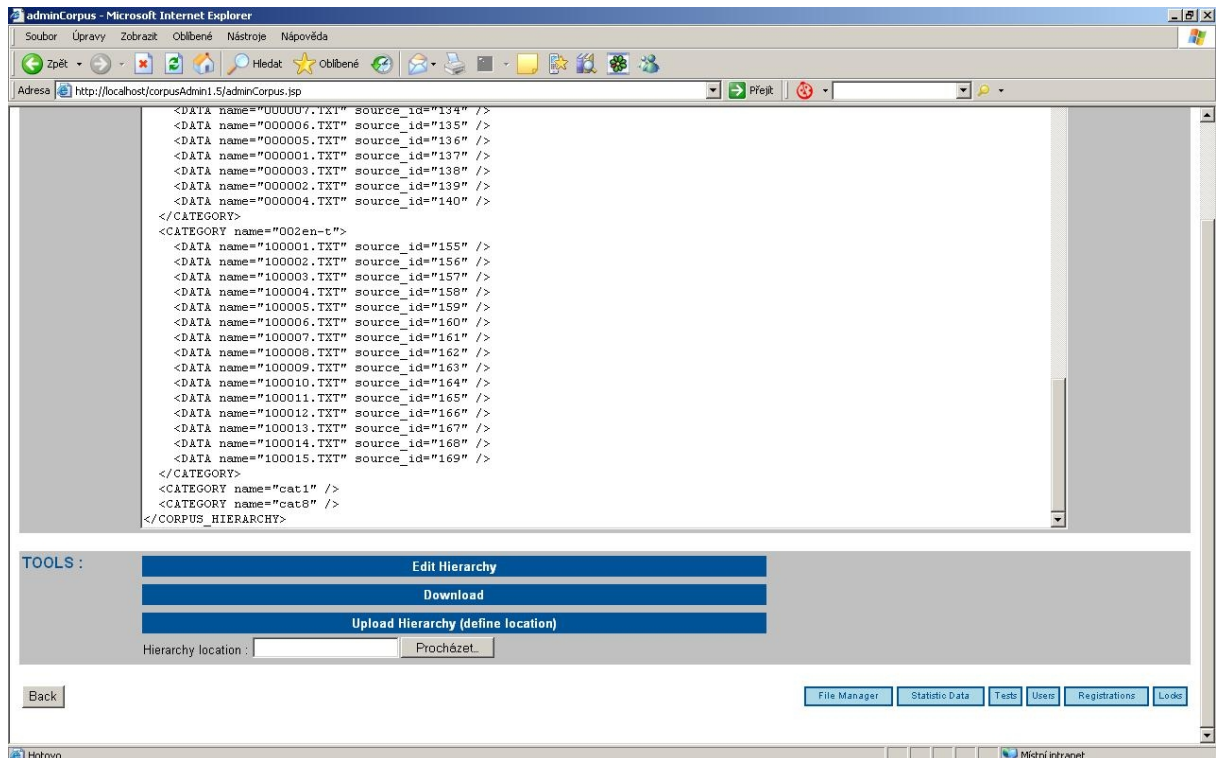
Img. 2.1.1_2
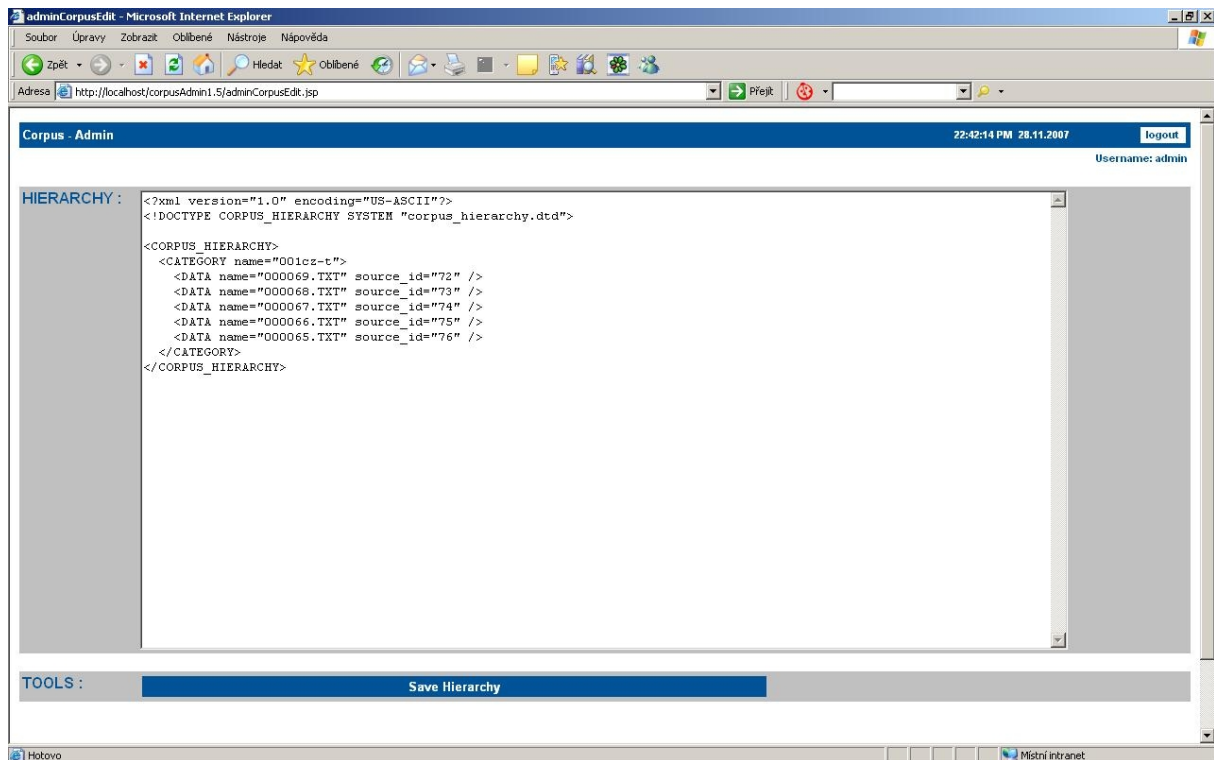
Img. 2.1.1_3 – uploaded corpus hierarchy

2.1.2. Edit hierarchy

System displays page, where admin can write directly into textarea, which contains hierarchy structure. This way of editing is useful for doing small changes in the structure. User confirms changes by pressing button *Save hierarchy*, this operation activates modified hierarchy for the whole *Corpus system*. (img. 2.1.2_1)
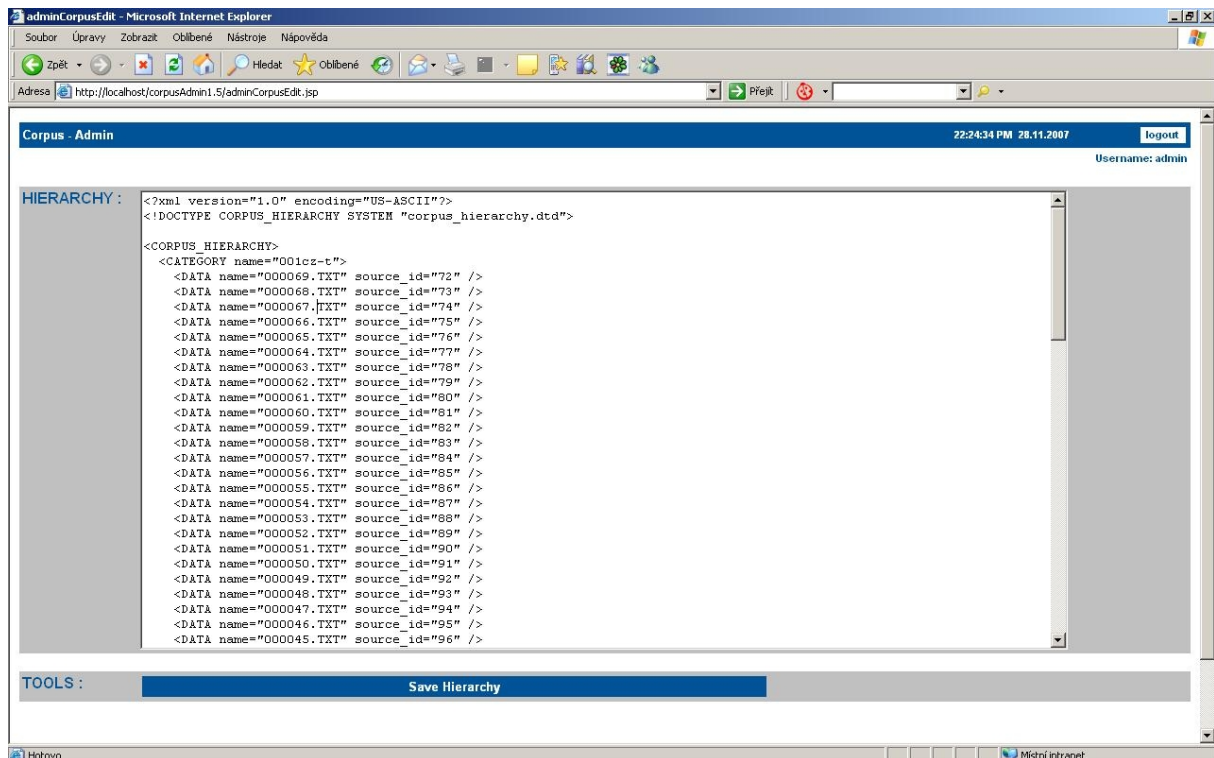
Img. 2.12_1

If any error occurs during validation of hierarchy, hierarchy is not confirmed and error report is displayed under the textarea of hierarchy structure.

2.2. File / Data transfer

System functionality for uploading, removing, moving real data for *Corpus*. This real data can be associated with specific part of *Corpus hierarchy*. The real data are used for compression tests.

System shows page, where user can manipulate with all real data. (img. 2.2_1)
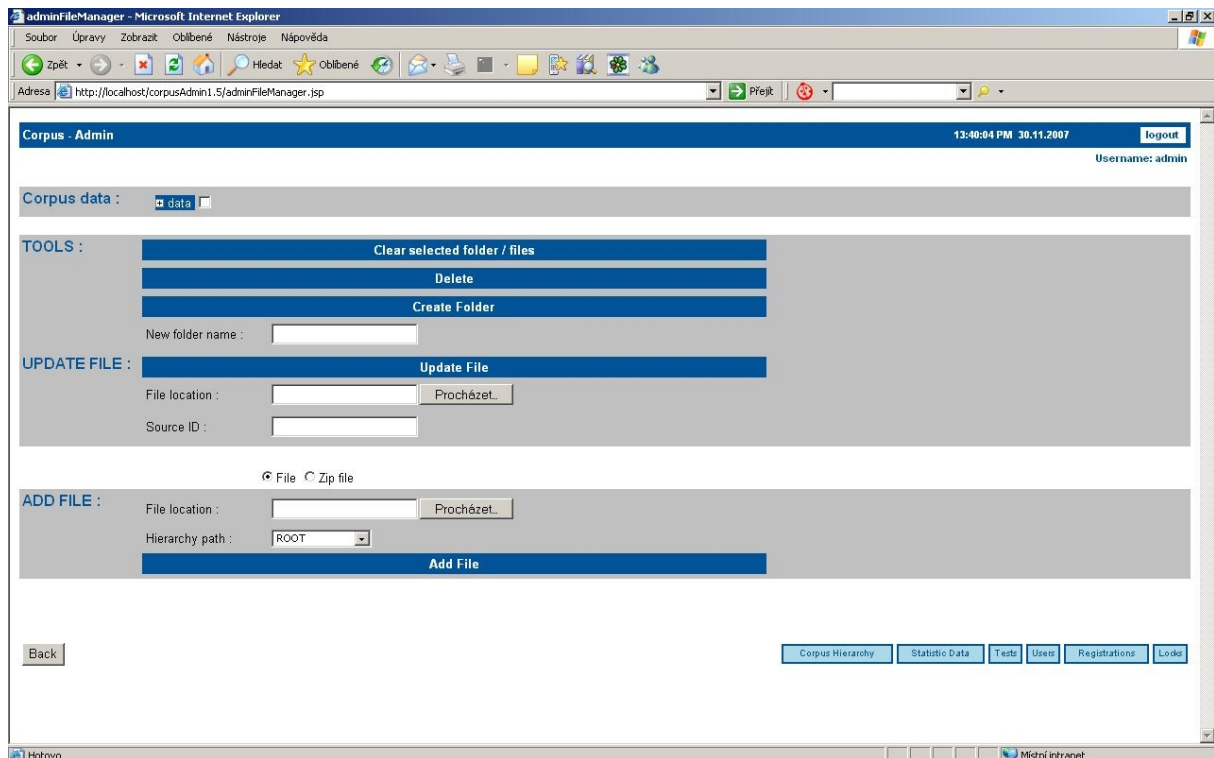
Img. 2.2_1

2.2.1. File manager

Page *File manager* allows admin to organize the file structure of real data on server local disc. System shows three sections, each with specific tools.

Corpus data – system displays file tree structure, how directories and files are organized on server disc. Admin can browse in this treeview, open directories and see the content of the directory. Admin can select and check directories or files, with which he can work.

Directories / Files tools – tools which are active just only on selected or checked directories or files:

> *Clear selected folder / files* – system unchecks selected directory, and also recursively all its subdirectories and files, which selected directory contains
>
> *Delete* – system removes all checked files and directories, which are child of selected directory, system also removes the whole content of each subdirectory
> *Create folder* – system creates new folder in selected directory, admin must define name of the new folder (img. 2.2.1_1)
>
> *Update file* – admin can replace existing data file, with new file, admin must define *Source ID* – which is identifier of replacing file, and then he must define location of the new file on his local disc (img. 2.2.1_2 – 2.2.1_3)
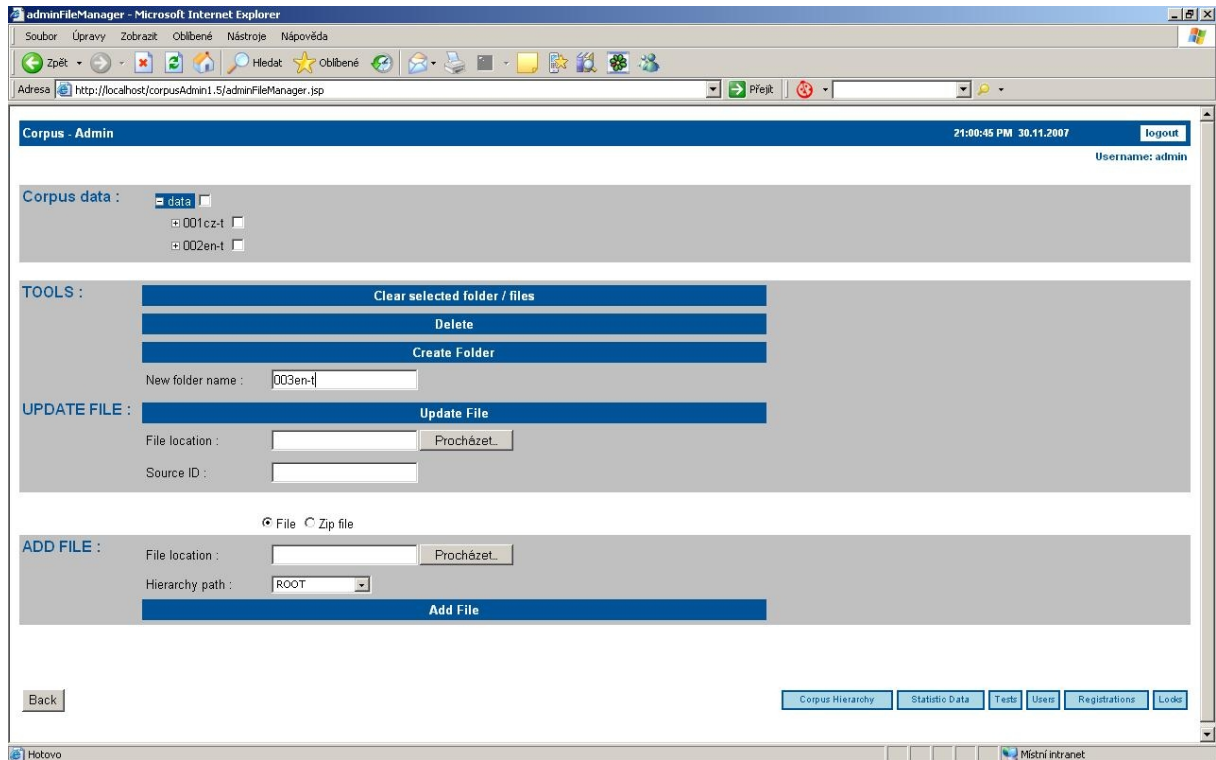
Img. 2.2.1_1
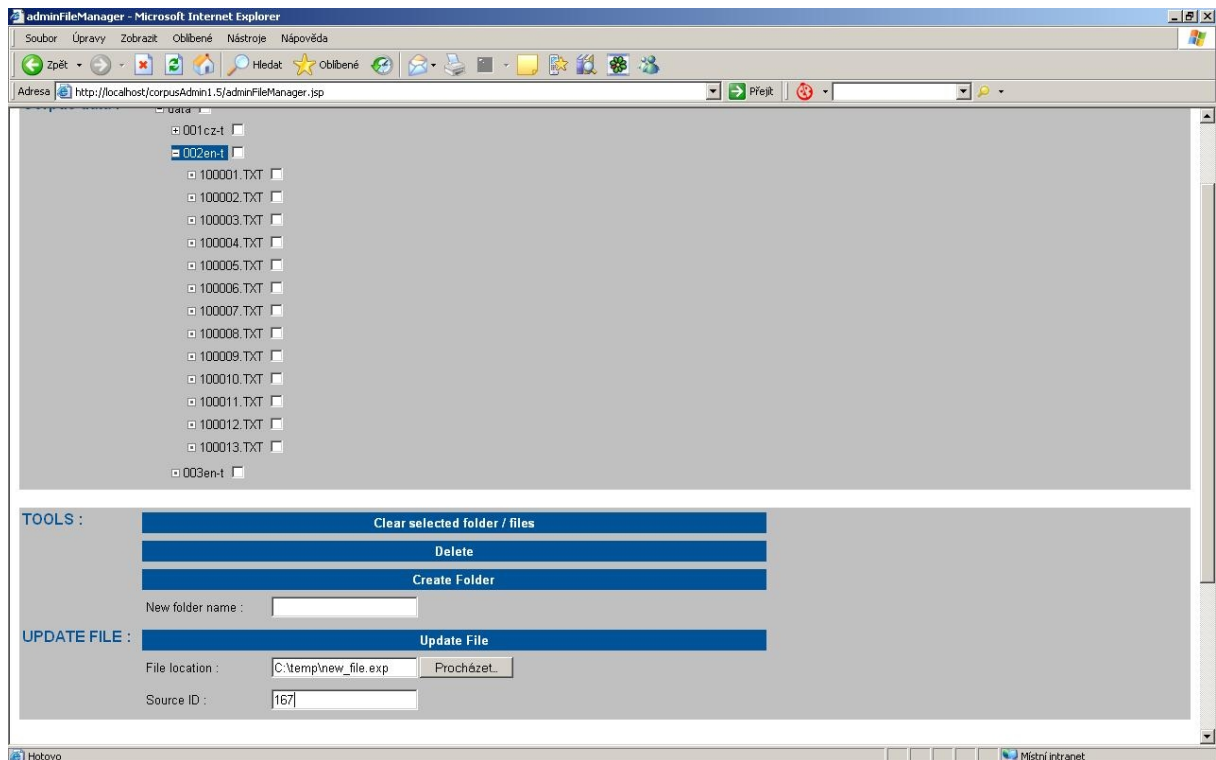
Img 2.2.1_2
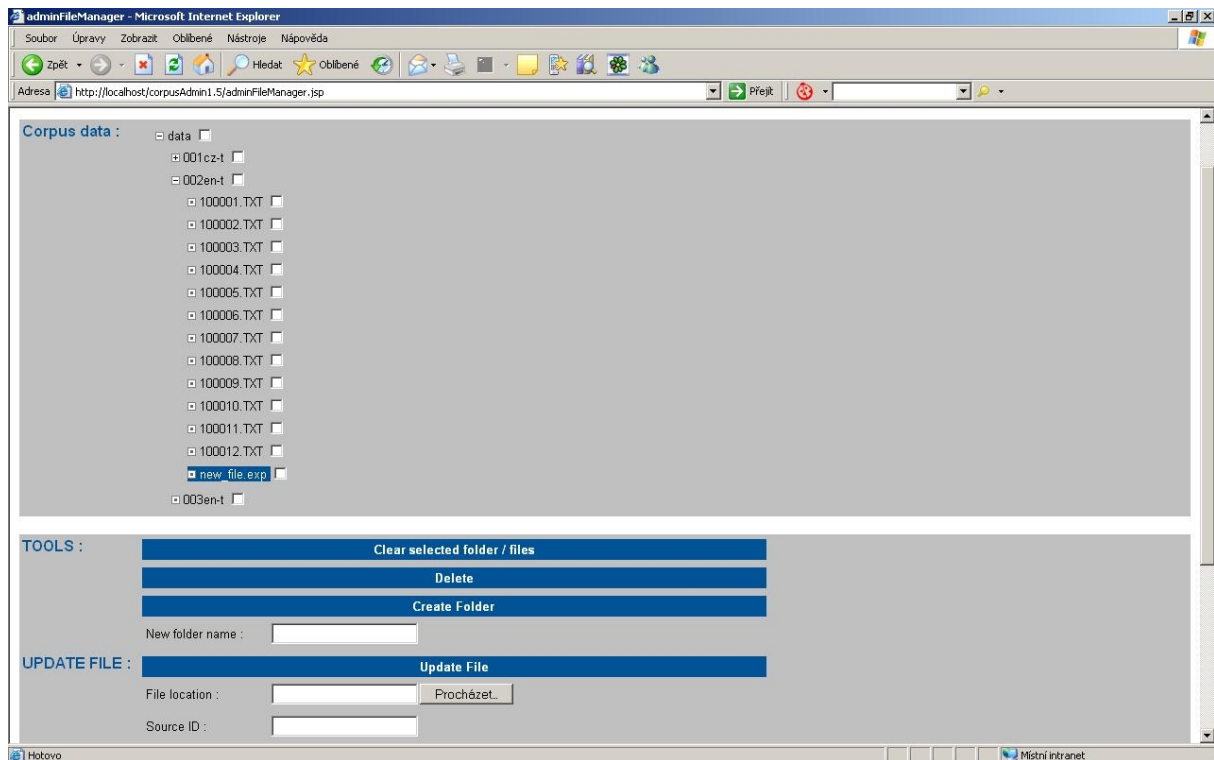
Img. 2.2.1_3

Add file – admin can add single file into specific part of *Corpus hierarchy*, or system allows him adding a group of files, but files have to by packed in one zip file, admin must define location of adding file (zip file) on his local disc, then he must also define *Hierarchy path*, which means part of *Corpus hierarchy*, where should be added file (zip file) placed

2.2.1.1. Delete files or directories

Admin must select in file treeview in which directory he wants to remove files or subdirectories, then he must check all files or directories, which should be removed. The checked files and directories have to be under the branch of selected directory.
(Img. 2.2.1.1_1)
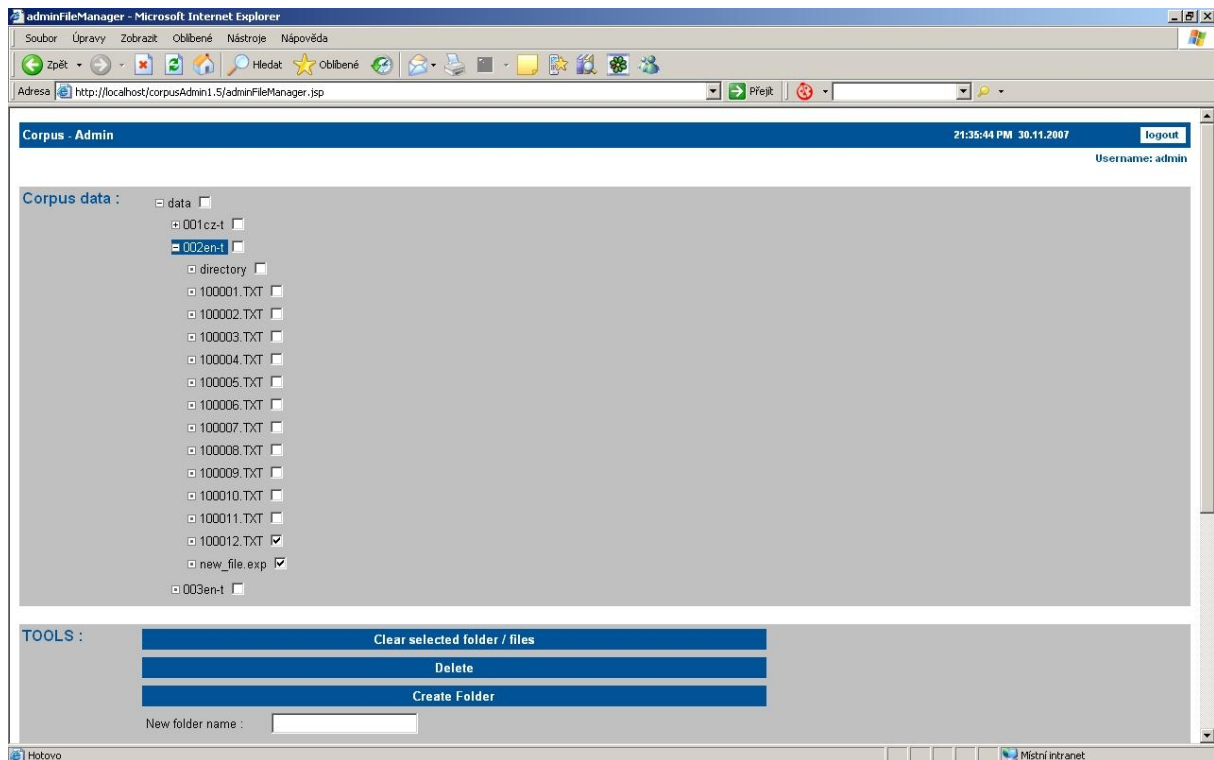
Img. 2.2.1.1_1

After pressing button *Delete*, system makes list of removing files or directories, and this list is displayed on next page. (img. 2.2.1.1_2)
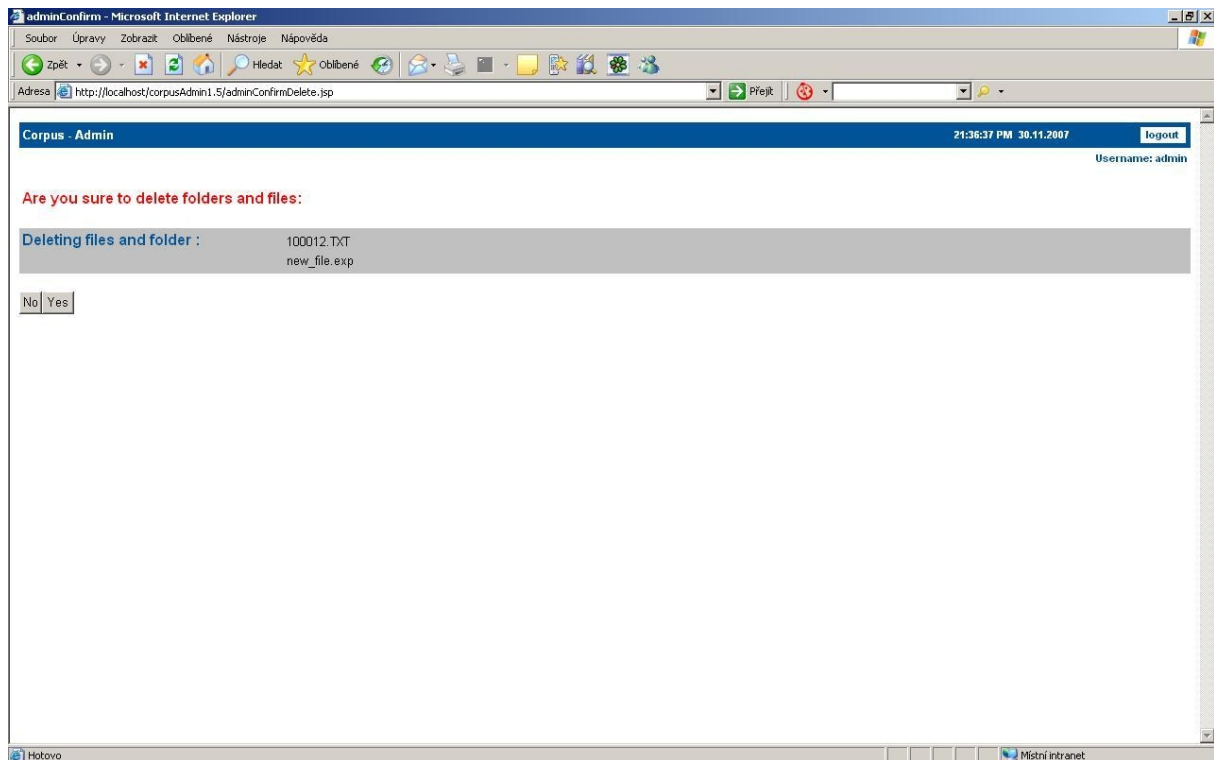
Img. 2.2.1.1_2

Admin discards delete action by pressing button *No*. After confirmation, system removes all checked files and directories from server file structure.

2.2.1.2 Add file

Admin can add single file or group of files from his local disc to server's file system. First he must choose one from two options *File* and *Zip file*. Option *File* allows adding only one file, the second option allows adding more files, but these files must be packed in one zip file, which will be uploaded to server. Zip file is then unpacked and system inserts all files from zip to specific part of *Corpus hierarchy*. After choosing one from these two options, admin must define file location of uploading single file or zip file. Last step, is to define *Hierarchy path*, which represents concrete part of *Corpus hierarchy*, into which the new data will be inserted. If admin uploads zip file, and the files have some directory structure in this zip file, the directory structure is also conserved on the server's file system. (img. 2.2.1.2_1)
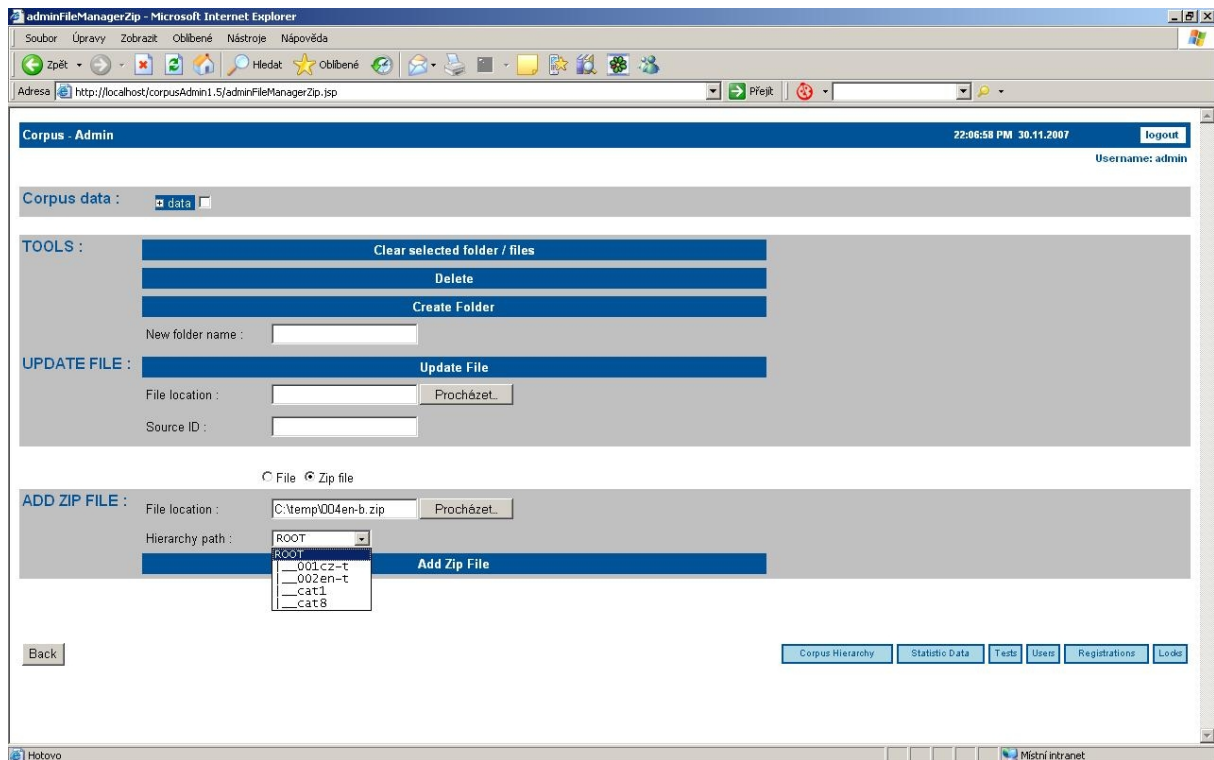
Img. 2.2.1.2_1

After pressing button *Add Zip File*, system checks if any of adding files already exist in *Corpus* data file system. Then system generates *Source ID* for each adding file. After all, list with adding files, their statuses and *Source IDs* is displayed on page. Admin can change status of each adding file to values :

ADD – only when adding file was not found on *Corpus* file system
OVERWRITE – when adding file was found on *Corpus* file system
NOT_OVERWRITE – when adding file was found on *Corpus* file system

(img. 2.2.1.2_2)
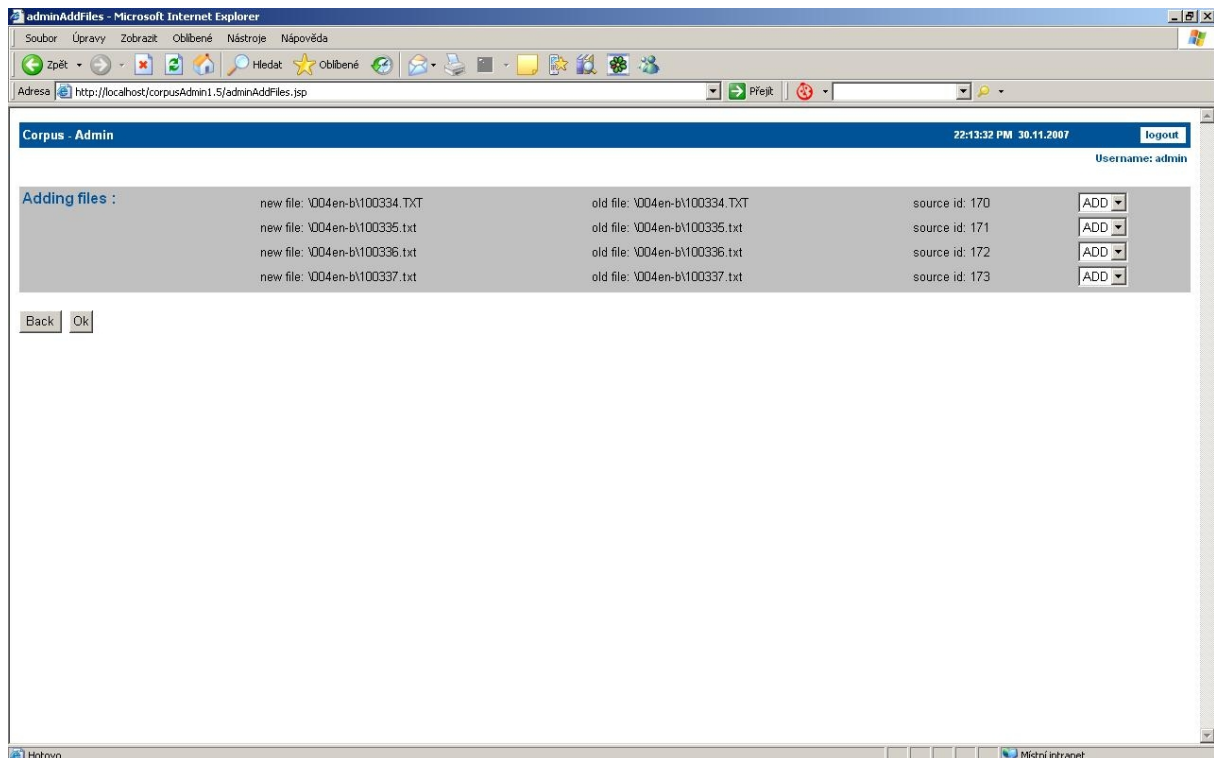
Img. 2.2.1.2_2

After confirmation, system adds new *Corpus hierarchy* part into selected part of hierarchy. And system also associates added files with hierarchy. The new structure of *Corpus hierarchy* is then displayed on page, where admin can this modified hierarchy confirm or discard. (img. 2.2.1.2_3)
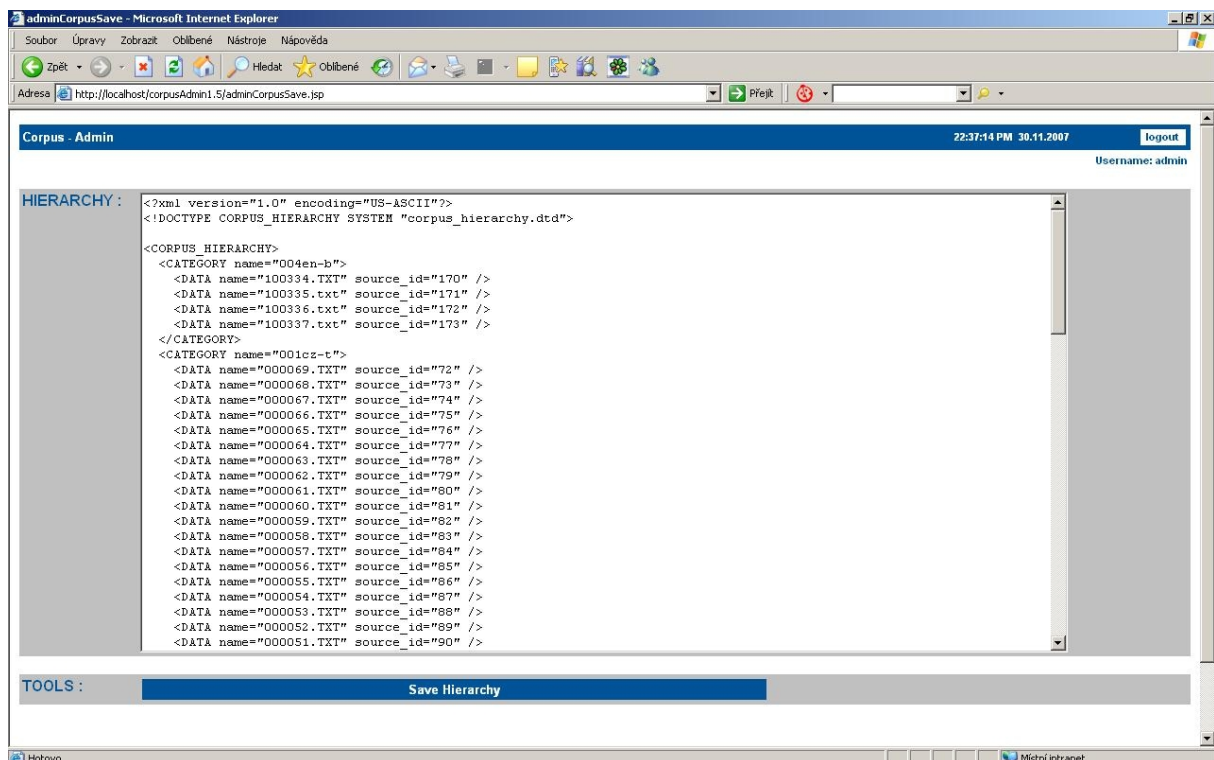
Img. 2.2.1.2_3

If admin discards modification of *Corpus hierarchy*, files were added into *Corpus* file system, but there were not associated with *Corpus hierarchy*, so this process must admin do next time in the future.

When the adding process is complete, system shows again first page of Data / File transfer. (img. 2.2.1.2_4)
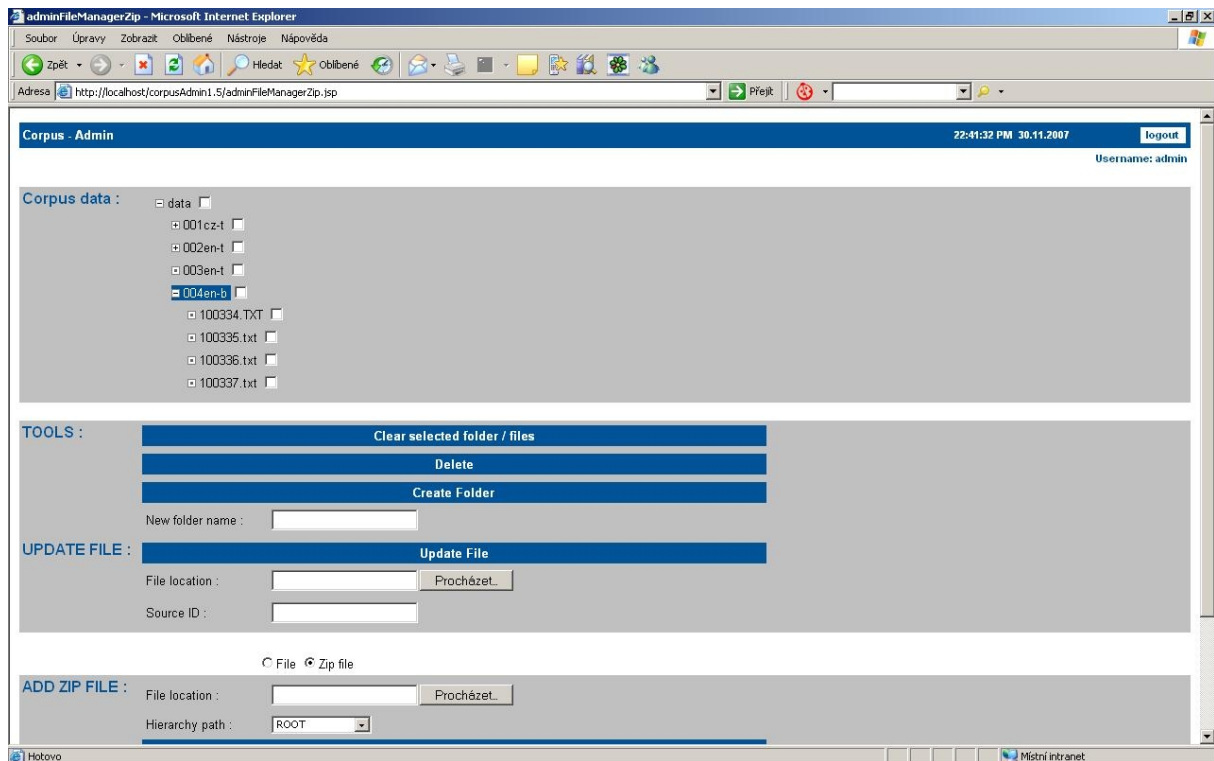
Img. 2.2.1.2_4

Processes with adding file or zip file use the same procedure.

2.3. Statistic data comparison

System functionality for browsing and comparing result data of all tests. Only the data of successful tests are displayed.

Page with table containing basic info of all test results is displayed at first.

2.3.1. Basic info

In *Basic info* page, system displays table with all test results, which were completed successfully. Each row represents one test with result data. Test result data information contains:

Test ID – unique number of test, set up by system
Test name – name of test, set up by user
Login name – name of user, who made this test
Corpus ID – unique number of *Corpus hierarchy*, number represents version of modification of *Corpus hierarchy*, on which was test done
Category – path in *Corpus hierarchy*, path represents subpart of hierarchy, on which was test done
Data size – average value of source data size, on which was test done, number represents average value of sizes of all corpus source files, which were used for test
Count of data – total sum of files, on which was test done
Compress rate – resulting compress rate value, this value represents average value of compress rates of all source files, compress rate value is rate of original source file and compression result file
Compression – resulting fastness value of compression, value represents average value of compression fastness values of all source files, compression fastness value is rate of source file size and compression duration (bytes per second)
Decompression – resulting fastness value of decompression, value represents average value of decompression fastness values of all source files, decompression fastness value is rate of compressed file size and decompression duration (byts per second)

(img. 2.3.1_1)

Img. 2.3.1_1

Admin operations with each test result data (row):

Detail info – by clicking into chosen row, system displays table only with test result data, which were done on the same *Corpus hierarchy* version.

Table operations:

Sort test list according to selected column – by clicking on headline of column, admin can sort list of tests by each column by ascending or descending order. Not using arrow up or down, system sorts list by ascending order by default

Choose page – admin can choose page, which he wants to display by:
- pressing button *First* – system displays first page of tests
- pressing button *Preview* – system displays preview page to actual page
- pressing button *Next* – system displays next page to actual page
- pressing button *Last* – system displays last page of table
- clicking on number of page – system displays exact page

2.3.2. Detail info

System displays table only with test result data, which were done on the same *Corpus hierarchy* version. Filter over *Corpus hierarchy* version is important to compare and create reports correctly. Table column information:

Test ID – unique number of test, set up by system

Test name – name of test, set up by user

Category – path in *Corpus hierarchy*, path represents subpart of hierarchy, on which was test done
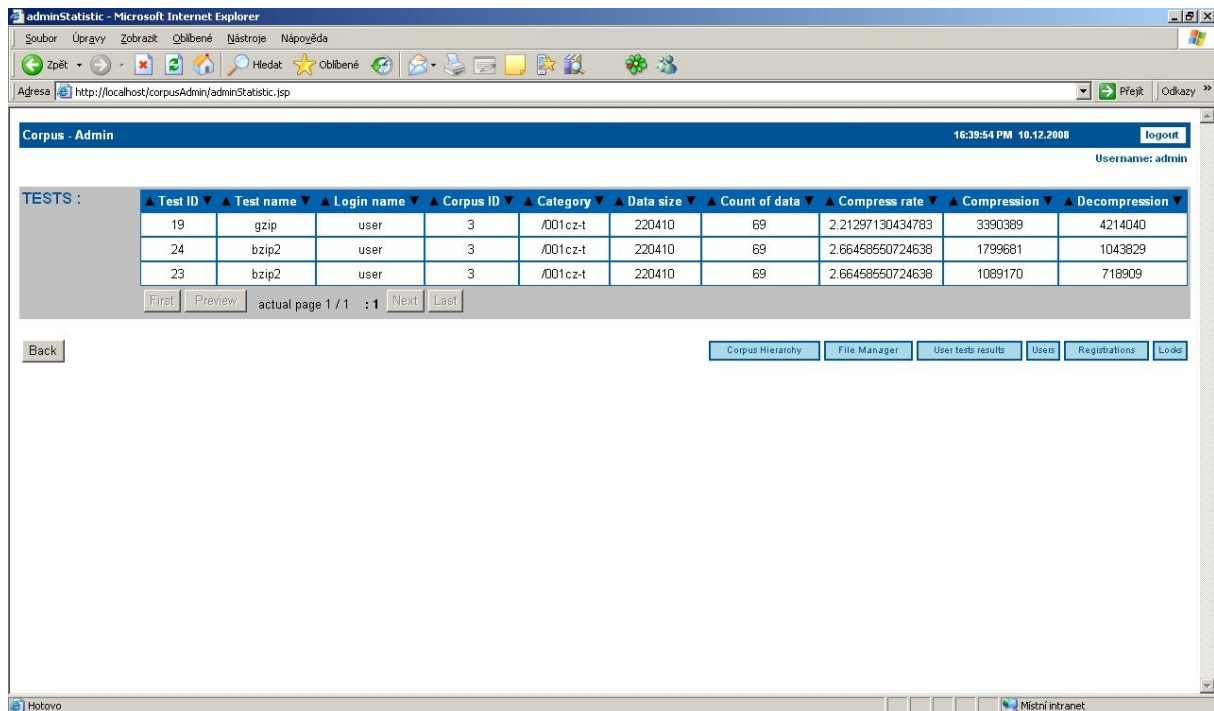
Compress rate – resulting compress rate value, this value represents average value of compress rates of all source files, compress rate value is rate of original source file and compression result

Compression – resulting fastness value of compression, value represents average value of compression fastness values of all source files, compression fastness value is rate of source file size and compression duration (bytes per second)

Decompression – resulting fastness value of decompression, value represents average value of decompression fastness values of all source files, decompression fastness value is rate of compressed file size and decompression duration (bytes per second)

(img. 2.3.2_1)

Img. 2.3.2_1

Admin operations with each test result data (row):

Selecting tests – admin can select tests (rows), with which he can work in next steps (img 2.3.2_2)

Compare by compress rate – by pressing button *Compare* under column *Compress rate*, system compares and sorts all selected test using statistical *Median method, Median method* uses *compress rate* values as input data

Compare by compression – by pressing button *Compare* under column *Compression*, system compares and sorts all selected test using statistical *Median method, Median method* uses *compress fastness* values as input data

Compare by decompression – by pressing button *Compare* under column *Decompression*, system compares and sorts all selected test using statistical *Median method, Median method* uses *decompress fastness* values as input data
(img 2.3.2_3)

Report – system displays page with GUI for creating reports

Report Paramgroups - system displays page with GUI for creating reports, which use additional test parameters

Table operations:

Sort test list according to selected column – by clicking on headline of column, admin can sort list of tests by each column by ascending or descending order. Not using arrow up or down, system sorts list by ascending order by default. (img. 1.2.2_4)

Choose page – admin can choose page, which he wants to display by:
- pressing button *First* – system displays first page of tests
- pressing button *Preview* – system displays preview page to actual page
- pressing button *Next* – system displays next page to actual page
- pressing button *Last* – system displays last page of table
- clicking on number of page – system displays exact page
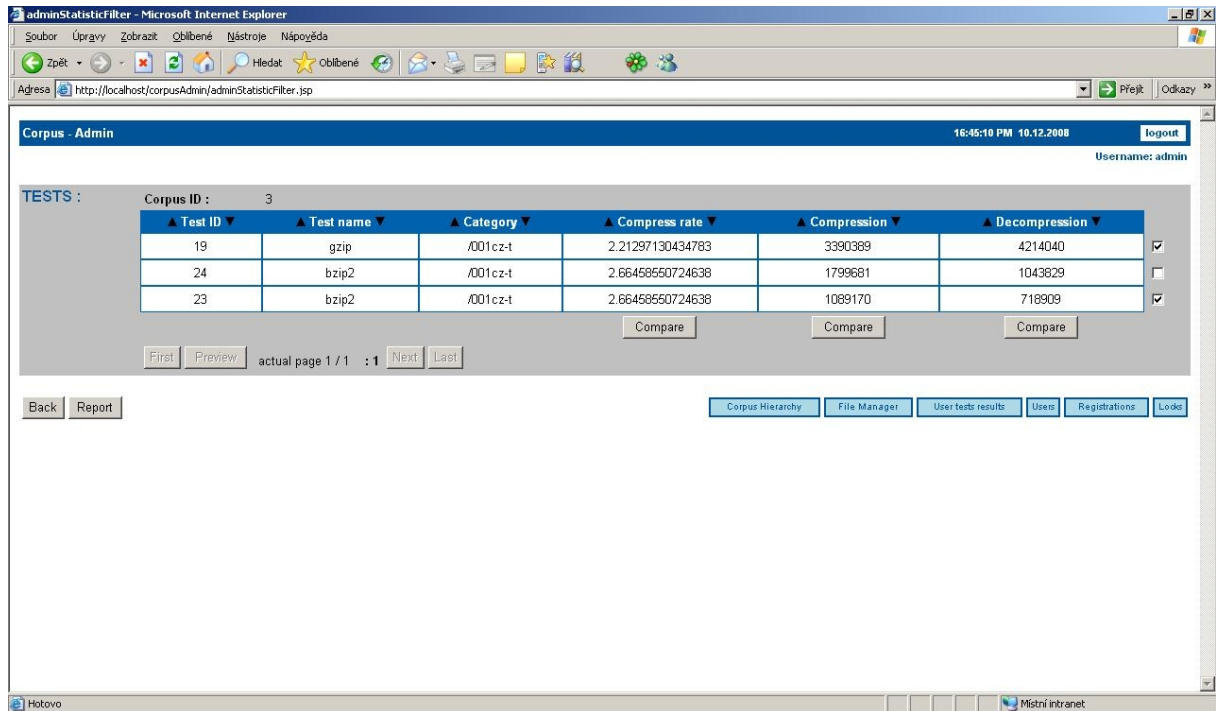
Img. 2.3.2_2
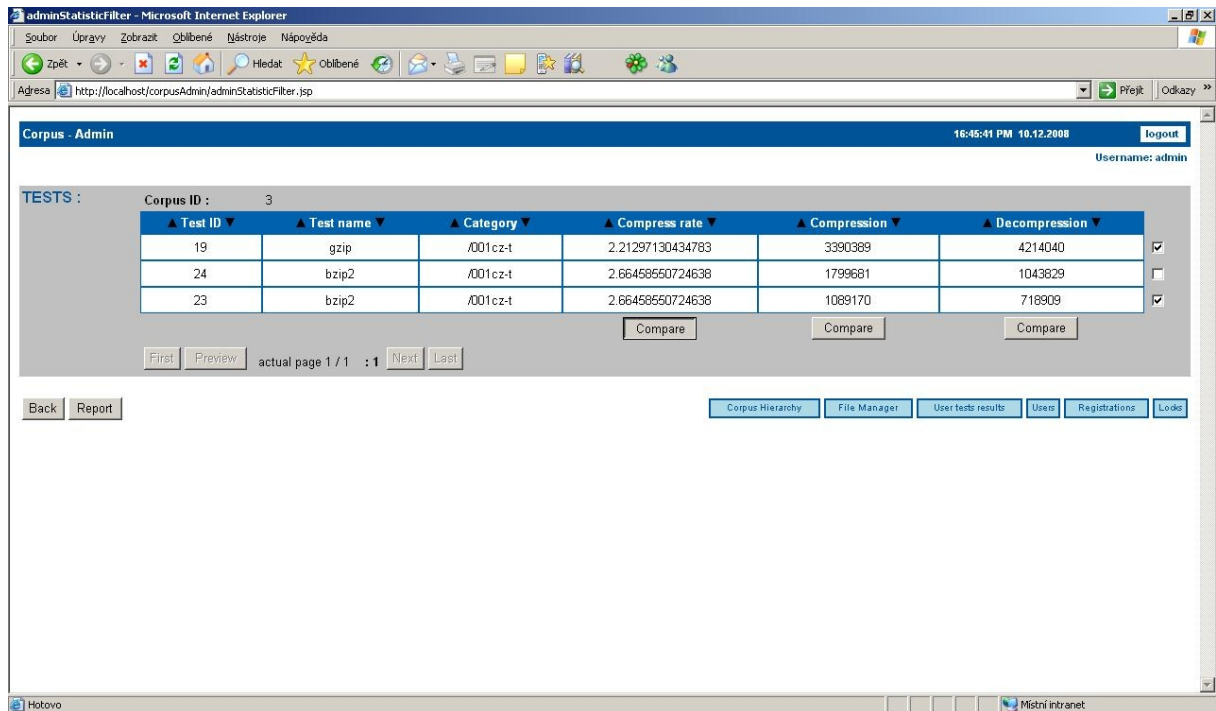
Img. 2.3.2_3

2.3.3. Compare

System displays table with sorted tests. Tests are sorted by specific value and system uses *Median test* to compare result data of tests. Tests are shown in ascending order.
(img. 2.3.3_1)

Table operations:

Choose page – admin can choose page, which he wants to display by:
- pressing button *First* – system displays first page of tests
- pressing button *Preview* – system displays preview page to actual page
- pressing button *Next* – system displays next page to actual page
- pressing button *Last* – system displays last page of table
- clicking on number of page – system displays exact page

Img. 2.3.3_1

2.3.4. Report

System displays tools, which help admin create own report from result data. (img. 2.3.4_1)

Admin operations for formatting table:

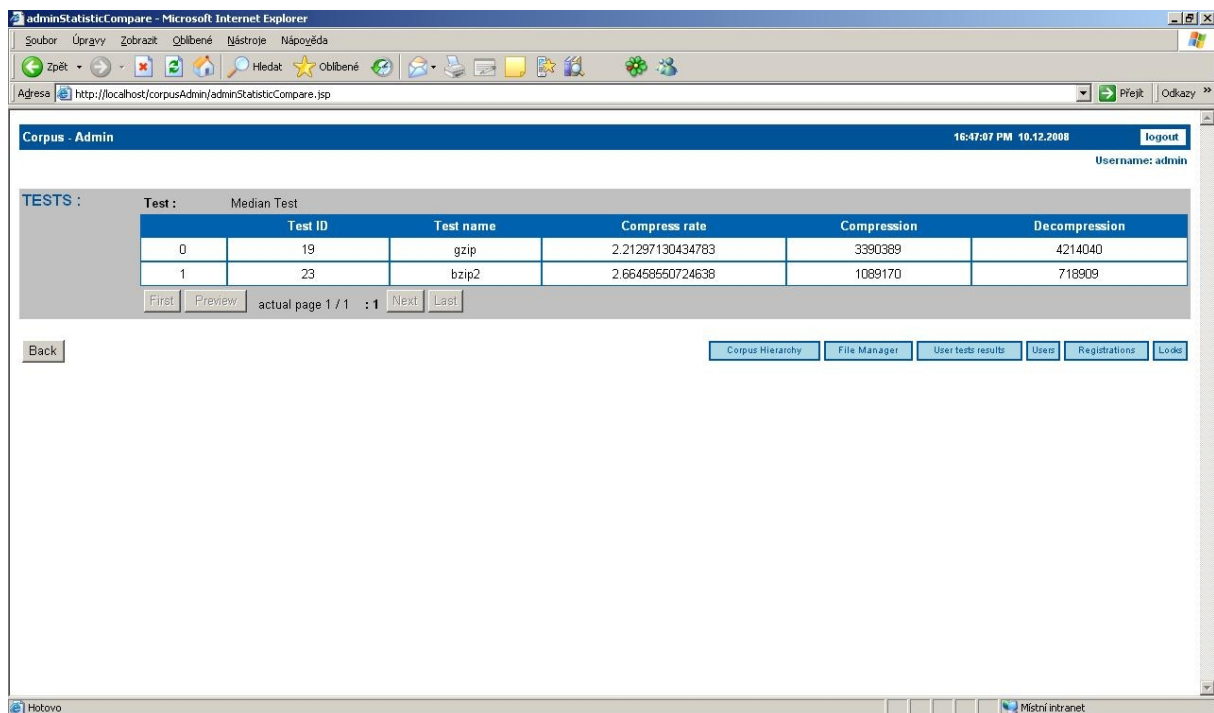*Displayed compression mode* – admin can choose two possibilities, how he see compression rate values, the first format is *Compression Ratio* and the second one is *Space Savings*. *Compression Ratio* represents rate of uncompressed size and compressed size. *Space Savings* noted as percentage value, is defined as the reduction in size relative to uncompressed size. (img. 2.3.4_2)

Count of decimals – admin can choose how many decimals in compression rate values should be displayed (img. 2.3.4_3)

Choose columns – admin can check columns, which he wants to have in report (img. 2.3.4_4)

Test Description column – admin can split column *Test description* into more columns, so in report, there are painted more columns. Admin must use char '*|*' to split value into two columns. After placing all split chars, system takes row with the highest count of split chars, and according to this line, system splits the whole column *Test description.* For example there are two rows, the first one has in *Test description* column value '*bzip*' and the second one '*gzip*'. Now, admin wants to add some prefix description to actual name. So he adds split chars before each *Test description* column value. After his operation, there are two rows with values '*|bzip*' ans '*|gzip*'. So in report, there will be displayed two columns for test description, but the first split column is empty, because admin does not defined prefix for any column:

```
            Test description
        Split part 1    Split part 2
Row 1       --              bzip
Row 2       --              gzip
```

Now, admin adds prefix value '*english*' into first row, so the *Test description* column has value '*english|bzip*', the second row is still empty:

```
            Test description
        Split part 1    Split part 2
Row 1   english             bzip
Row 2      --               gzip
```

Admin can continue in splitting *Test description* column by using some postfix expressions and so on… (img. 2.3.4_5)
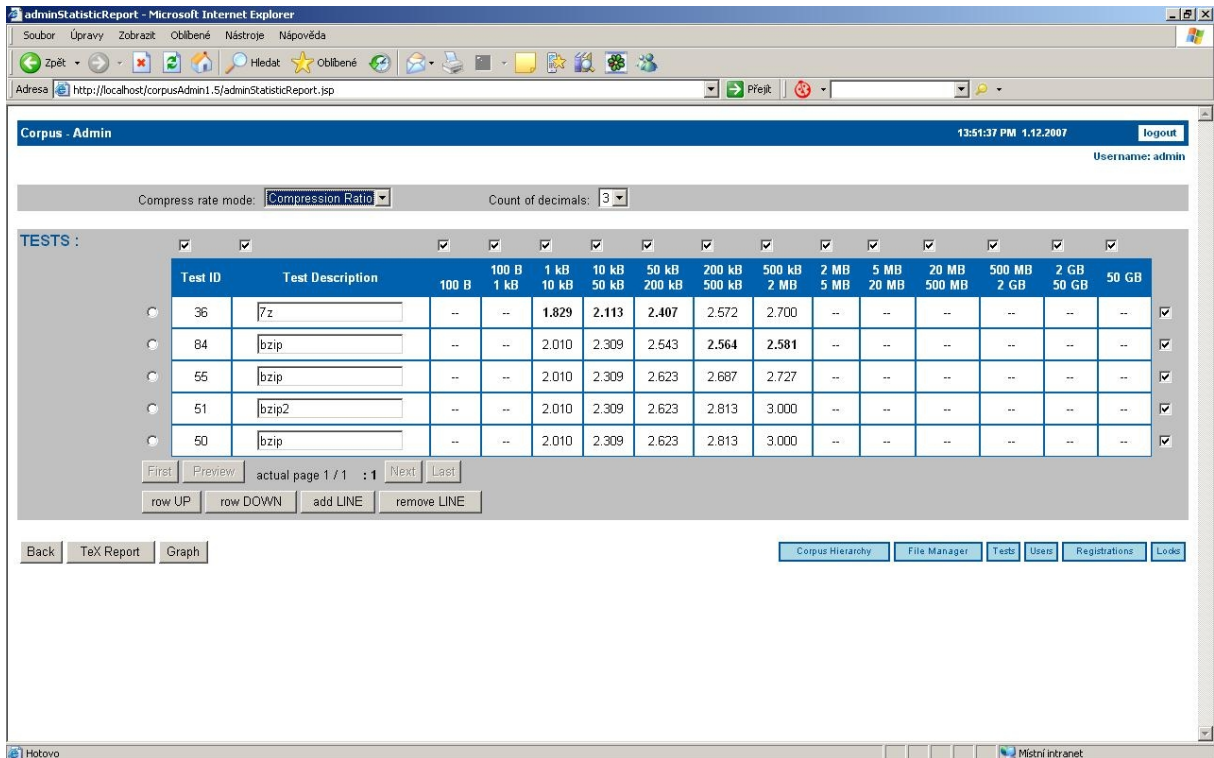
Img. 2.3.4_1
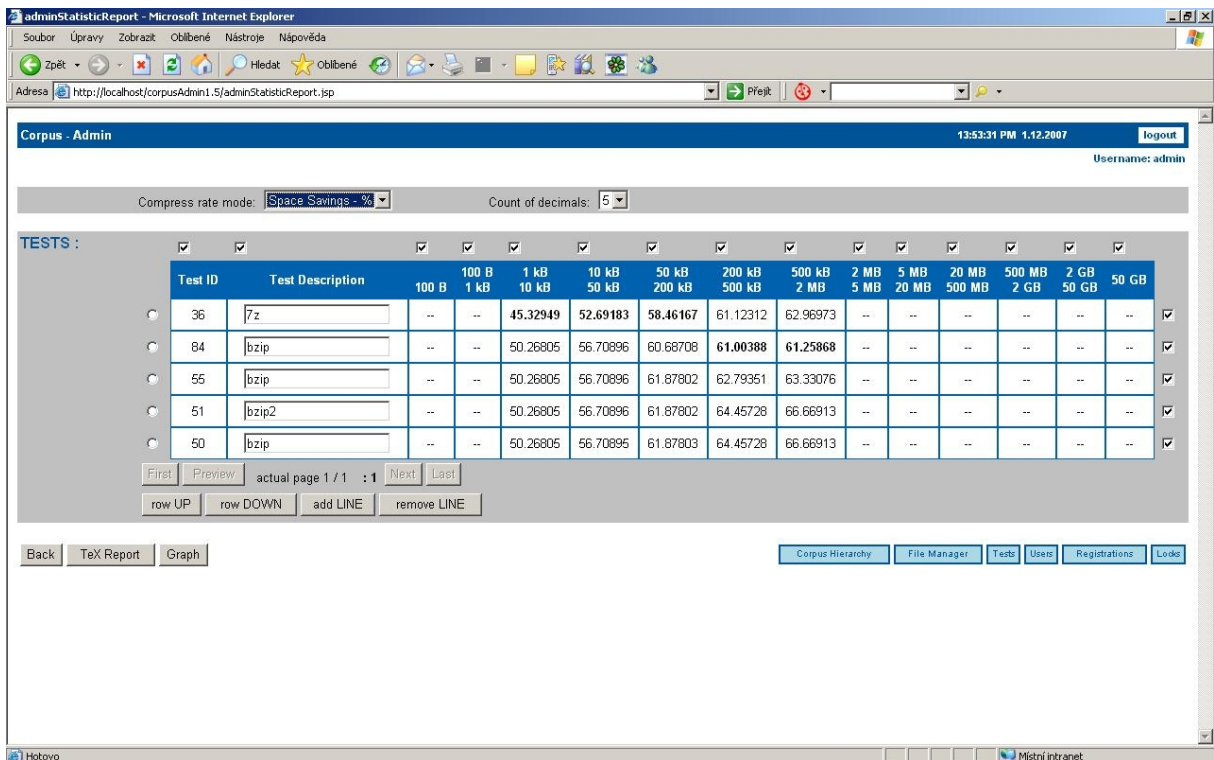
Img. 2.3.4_2

Img. 2.3.4_3
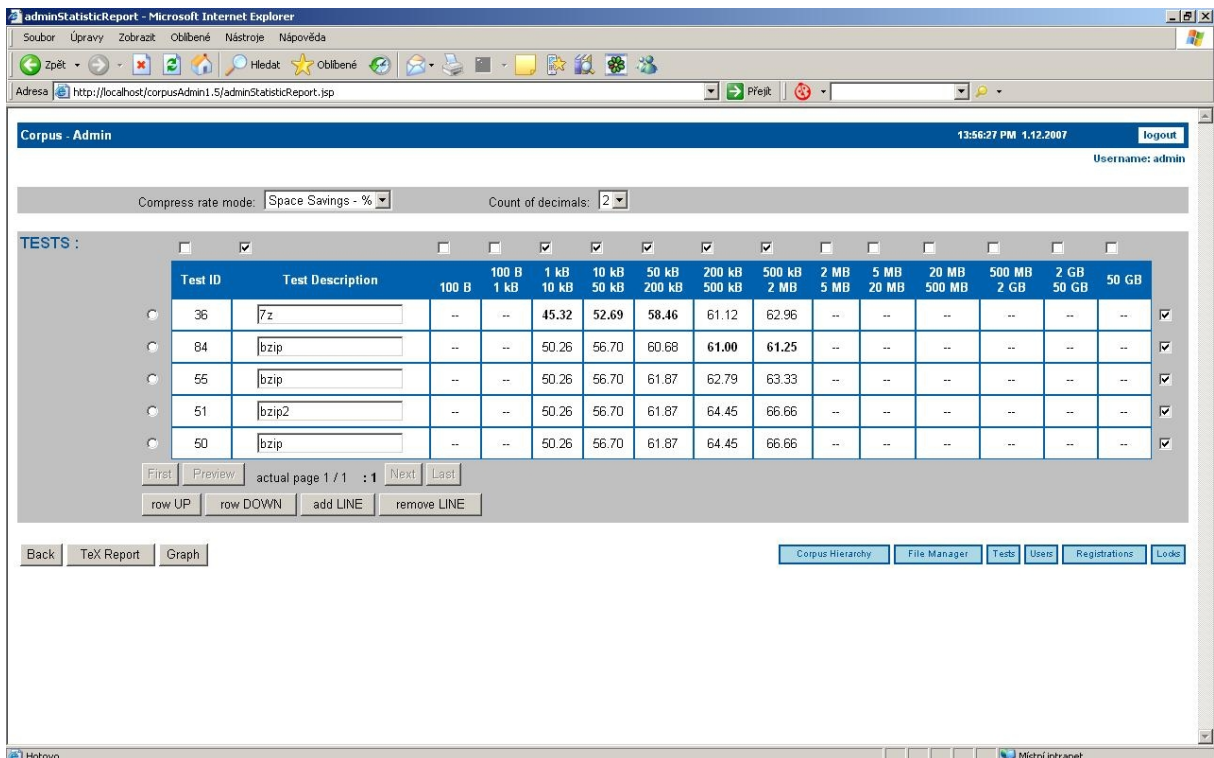
Img. 2.3.4_4
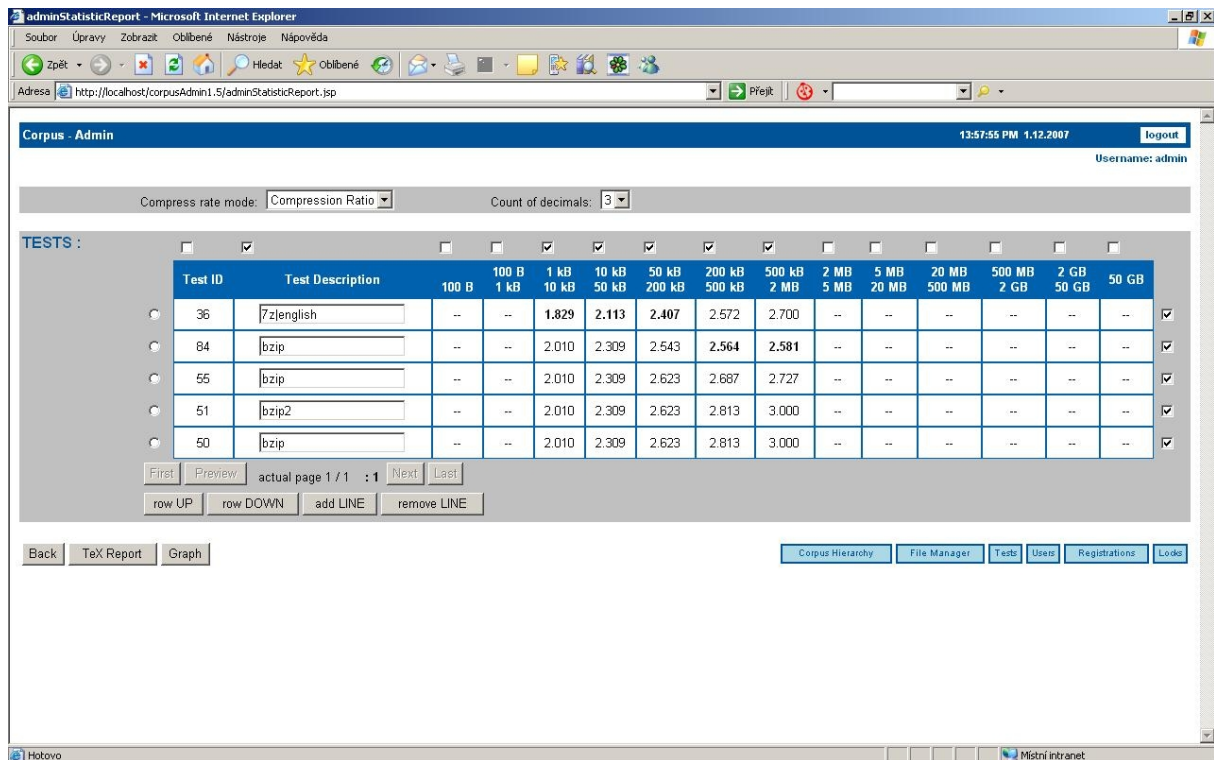
Img. 2.3.4_5

Admin operations for each test (row):

Check rows – admin can choose, which tests (rows) he wants to have in report

Select row – admin can use other tools for working with each row, but these tools are active only when at least one row is selected

Row Up – system moves selected row in front of previous row (img. 2.3.4_6)

Row Down – system moves selected row behind next row
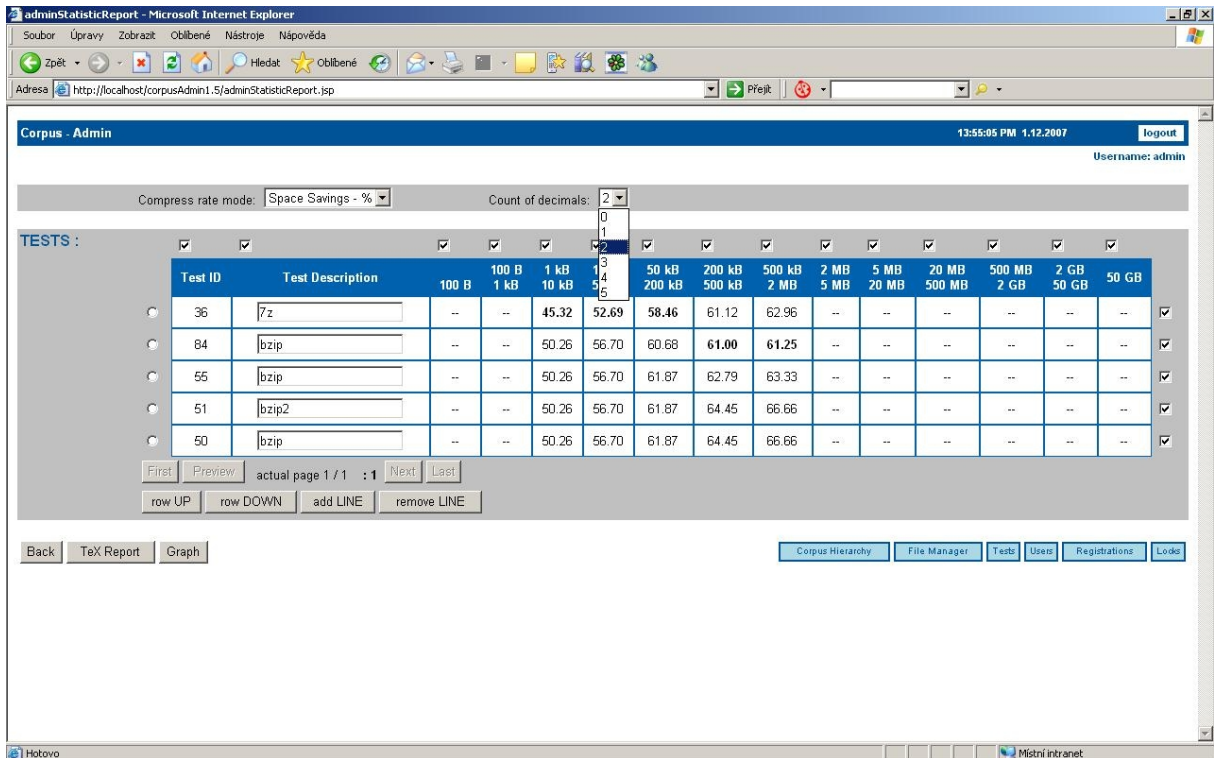
Add Line – system adds double line behind selected row (img. 2.3.4.7)

Remove Line – system removes double line for selected row, if this double line exists

Functionality of adding and removing double lines into and from table, splits this table into more fragments. Each fragment has in each row max/min value, displayed as bold text. So if table has two fragments, each column has two max/min values.
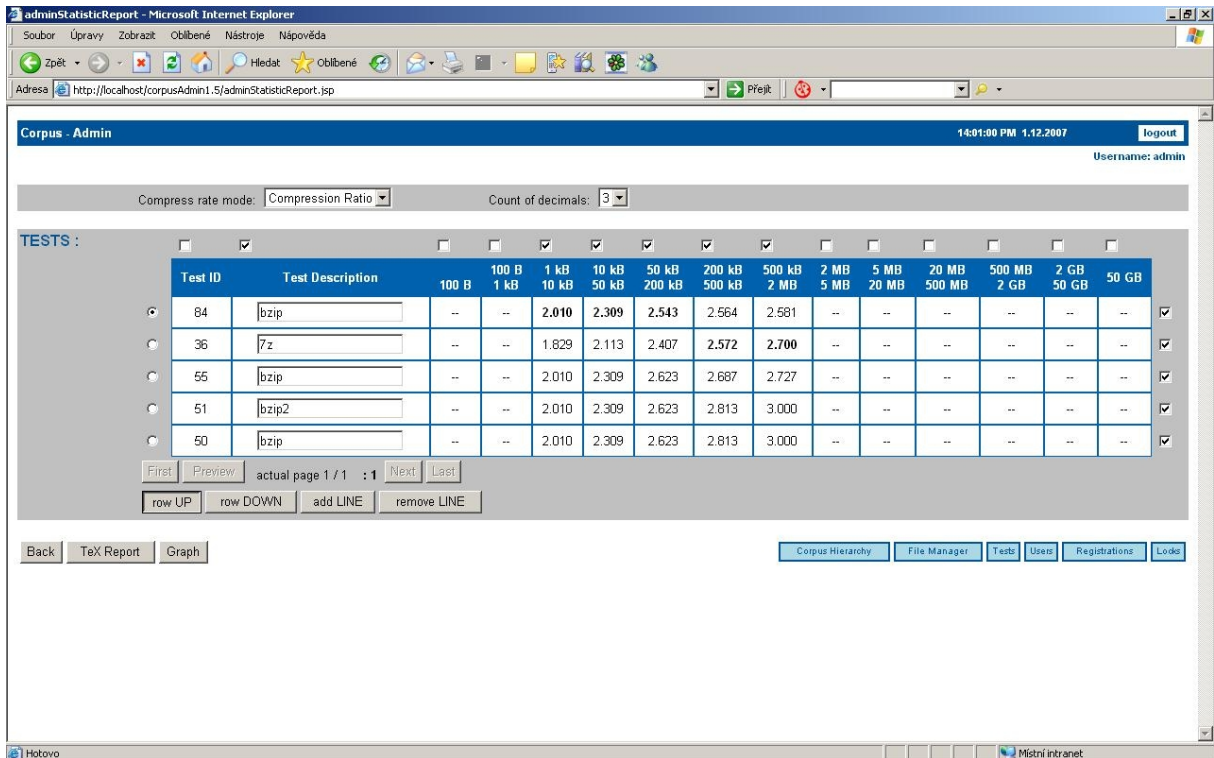
Img. 2.3.4_6

Img. 2.3.4_7

Standard table operations:

Choose page – admin can choose page, which he wants to display by:
- pressing button *First* – system displays first page of tests
- pressing button *Preview* – system displays preview page to actual page
- pressing button *Next* – system displays next page to actual page
- pressing button *Last* – system displays last page of table
- clicking on number of page – system displays exact page

If admin is satisfied with format of table, he can generate report of this table into various file formats:
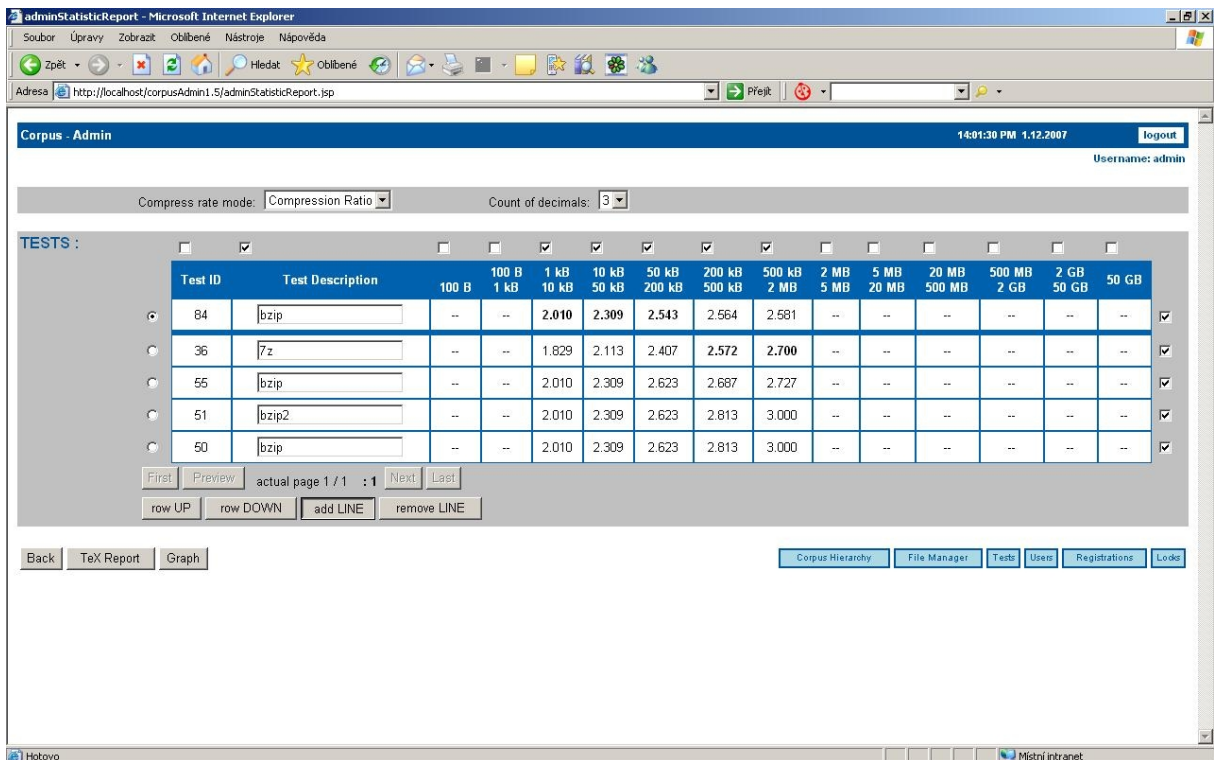
TeX report – by pressing button *Tex report*, system generates *TeX* source code of formatted table, and admin can download and save this *TeX* source code into his local disc. Then he can use this downloaded source code into his *TeX* document. (img. 2.3.4_8 – 2.3.4_10)
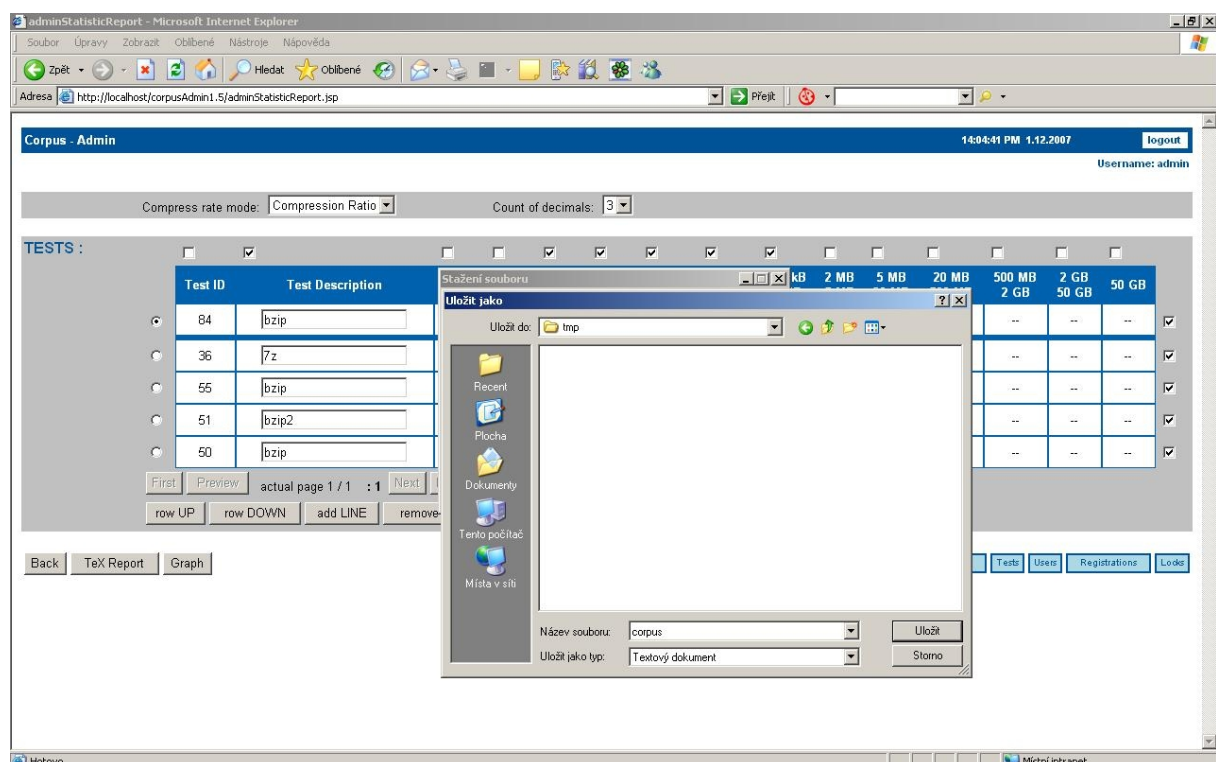
Img. 2.3.4_8

```
\begin{table}[h]
\begin{center}
\begin{tabular}{|c|c|c|c|c|c|}
\hline
 & 1 kB & 10 kB & 50 kB & 200 kB & 500 kB\\
 & 10 kB & 50 kB & 200 kB & 500 kB & 2 MB\\
\hline
bzip & \textbf{2.010} & \textbf{2.309} & \textbf{2.543} & 2.564 & 2.581\\
\hline
7z & 1.829 & 2.113 & 2.407 & \textbf{2.572} & \textbf{2.700}\\
bzip & 2.010 & 2.309 & 2.623 & 2.687 & 2.727\\
bzip2 & 2.010 & 2.309 & 2.623 & 2.813 & 3.000\\
bzip & 2.010 & 2.309 & 2.623 & 2.813 & 3.000\\
\hline
\end{tabular}
\end{center}
\end{table}
```

Img. 2.3.4_9 – TeX source code

| | 1 kB | 10 kB | 50 kB | 200 kB | 500 kB |
	10 kB	50 kB	200 kB	500 kB	2 MB
bzip	**2.010**	**2.309**	**2.543**	2.564	2.581
7z	1.829	2.113	2.407	**2.572**	**2.700**
bzip	2.010	2.309	2.623	2.687	2.727
bzip2	2.010	2.309	2.623	2.813	3.000
bzip	2.010	2.309	2.623	2.813	3.000
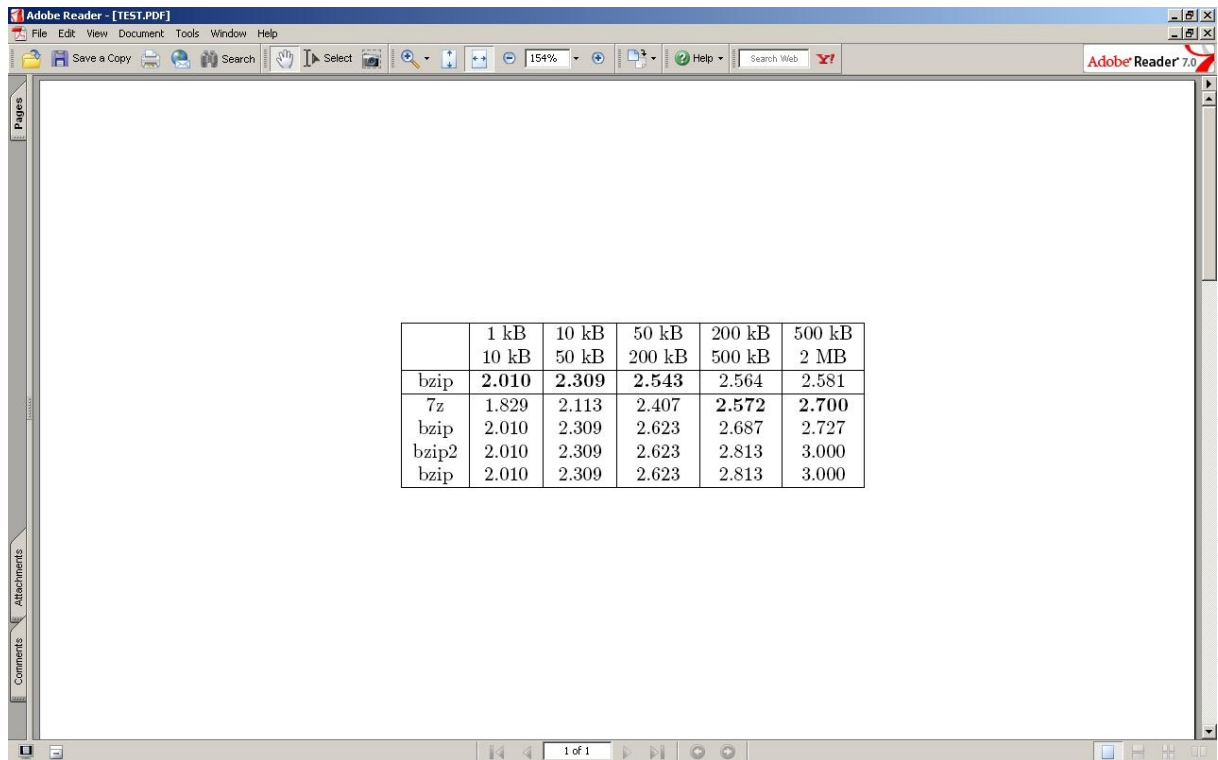
Img. 2.3.4_10 – TeX result in PDF file

R-Graph report – by pressing button *Graph*, system generates *R-Graph* source code of formatted table, and user can download and save this source code into his local disc. Then he can use this downloaded source code to generate graph from result values of tests and insert graph into his publications. (img. 2.3.4_11 – 2.3.4_13)
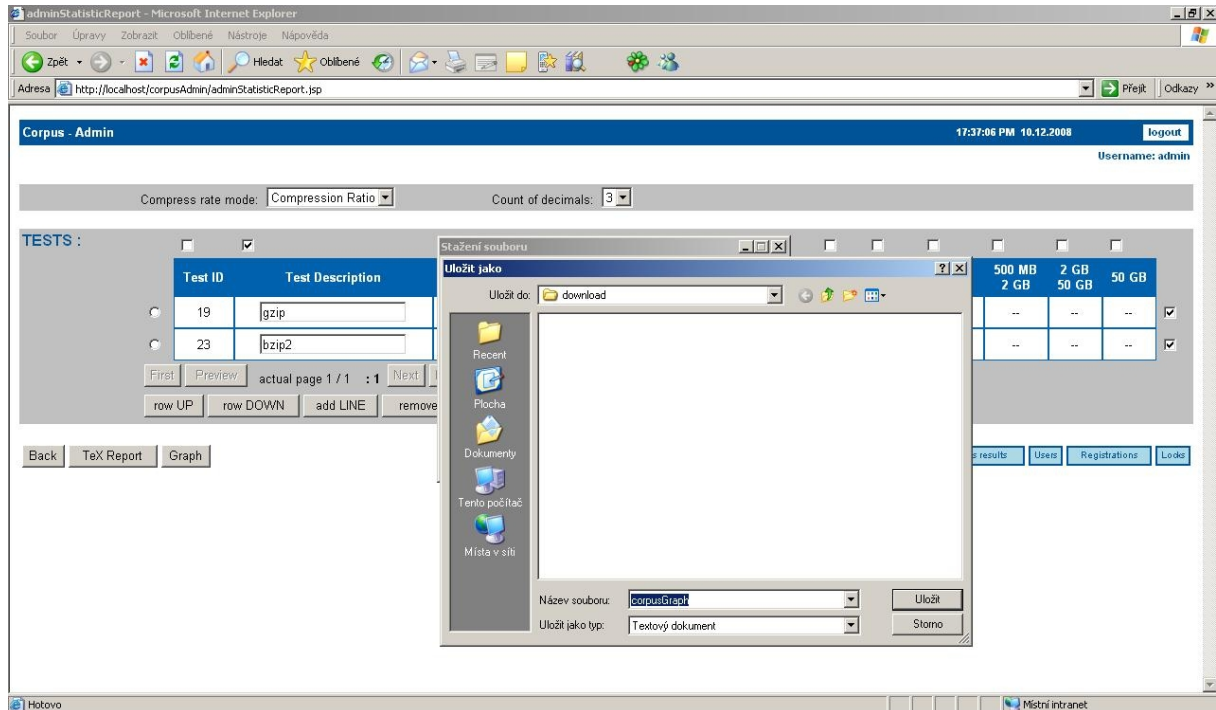
Img. 2.3.4_11 – R-Graph report
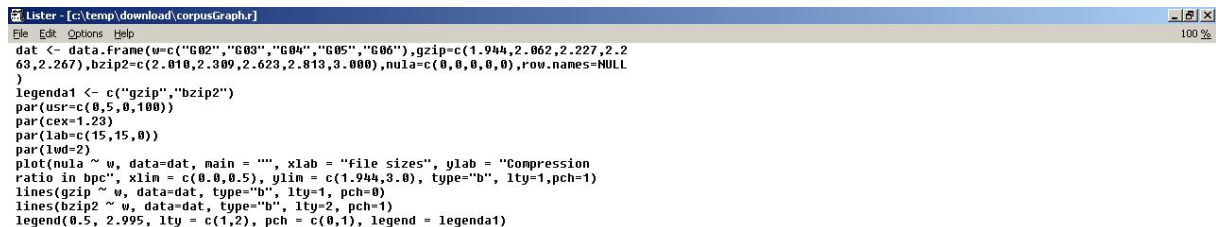
Img. 2.3.4_12 – R-Graph source code

79

Img. 2.3.4_13 – R-Gaph result

2.4. User tests results

Functionality for browsing in list, which contains all useful information about each test of compression programs, which were processed yet.

Page with table containing basic info of all tests is displayed at first.
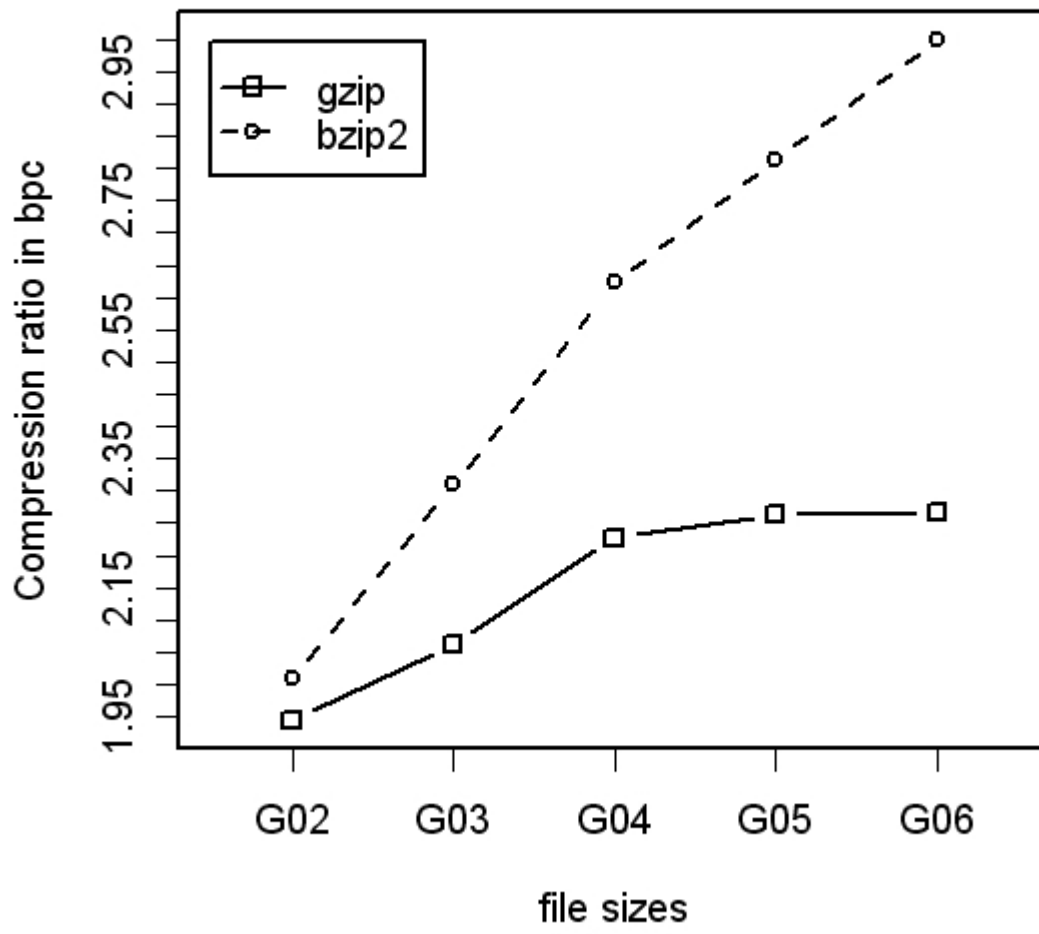
2.4.1. Basic info

In *Basic info* page, systems displays table with all tests, which were done yet. Each row represents one test. Test's information contains:

Test ID – unique number of test, set up by system
Test name – name of test, set up by user
Login name – name of user, who made test
Server ID – unique number of server, where the test was done, set up by system
Status – status of test result
Time – time when test start
Corpus ID – unique number of *Corpus hierarchy*, number represents modification of *Corpus hierarchy*, on which was test done
Category – path in *Corpus hierarchy*, path represents subpart of hierarchy, on which was test done

(img 2.4.1_1)

Admin operations with each test (row):

Download – by clicking this button, admin can download compression program, on which was this test processed. All files, which were uploaded and required for test, are packed into zip file, and admin downloads this zip file. To download compression program admin must define destination on his local driver (by using popup window with file manager).

Delete – by clicking this button, admin removes test from database, he removes it completely, that means with whole preview data, result data and files required for test.

Error info – by clicking on specific row, system displays error messages, which were generated during test process

Table operations:

Sort test list according to selected column – by clicking on headline of column, admin can sort list of tests by each column by ascending or descending order. Not using arrow up or down, system sorts list by ascending order by default.

Choose page – admin can choose page, which he wants to display by:
- pressing button *First* – system displays first page of tests
- pressing button *Preview* – system displays preview page to actual page
- pressing button *Next* – system displays next page to actual page
- pressing button *Last* – system displays last page of table
- clicking on number of page – system displays exact page

2.4.2. Error info

In *Error info* page system displays a list of source data, on which was selected test processed. Each row represents one source data. Source data information contains:

Source ID – number identification of concrete source data
Hit Ratio – value which represents if source file and decompressed file are the same, so if decompression after compression was successful
Status – final status of compression and decompression process with this source data

(img. 2.4.2_1)

Admin operations with each test (row):

Detail error info - by clicking on specific row, system displays error messages, which were generated during test process only for one selected source data

 Table operations:

Choose page – admin can choose page, which he wants to display by:
- pressing button *First* – system displays first page of tests
- pressing button *Preview* – system displays preview page to actual page
- pressing button *Next* – system displays next page to actual page
- pressing button *Last* – system displays last page of table
- clicking on number of page – system displays exact page
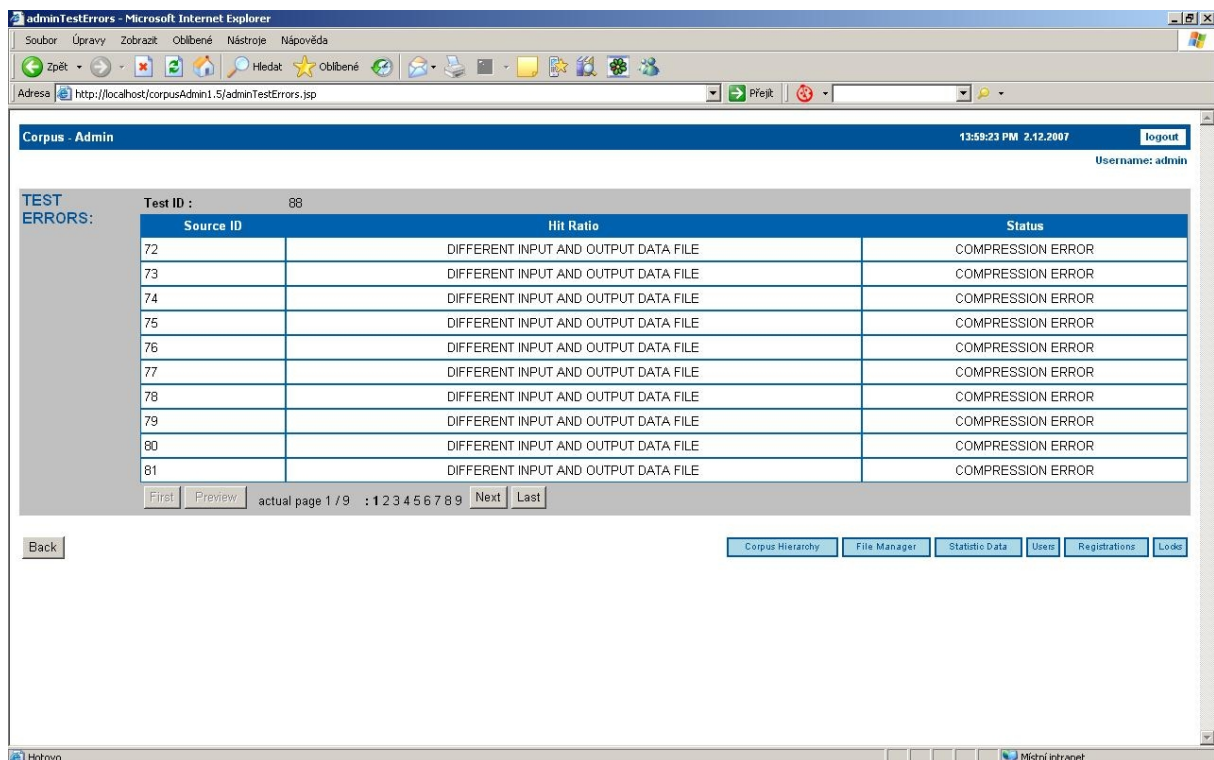
Img. 2.4.2_1

2.4.3. Detail error info

System displays error messages, which were generated during test process only for one selected source data. (img 2.4.3_1)
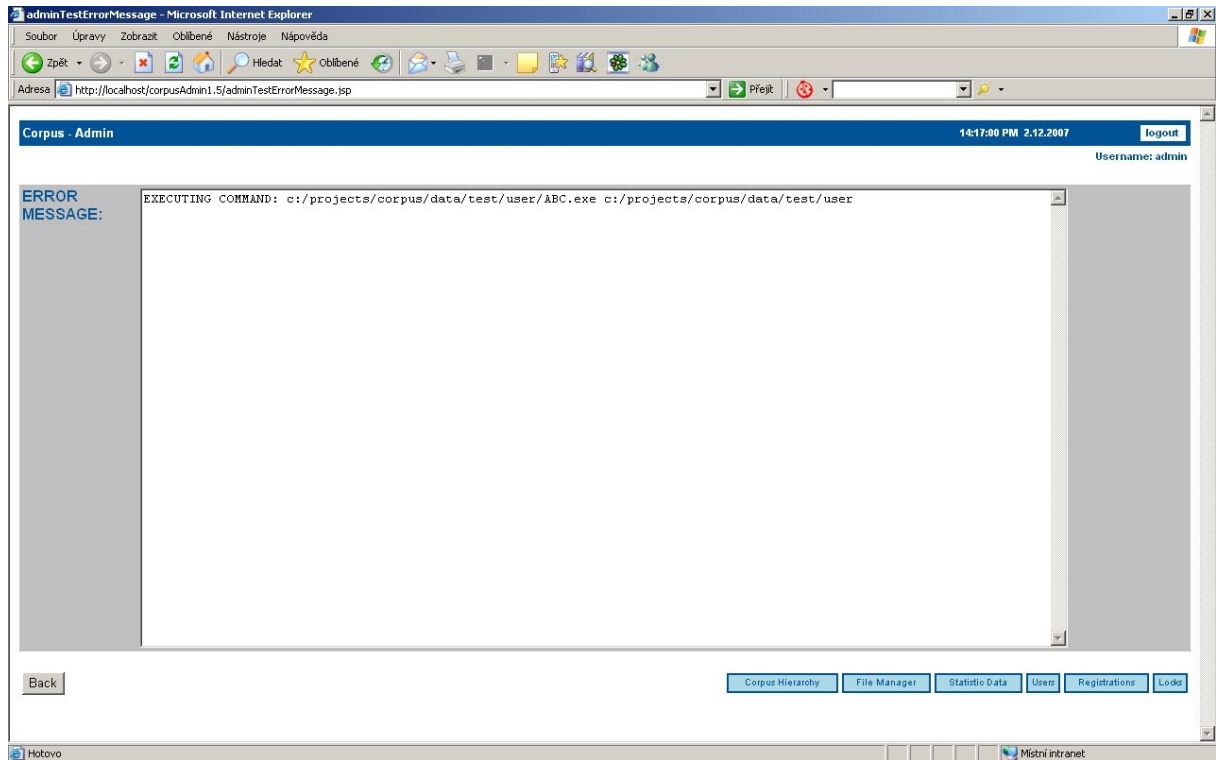
Img. 2.4.3_1

2.5. Users

System functionality for browsing and working with user accounts of the whole *Corpus system.*

Page with table containing list of all user accounts is displayed at first. (img. 2.5_1)
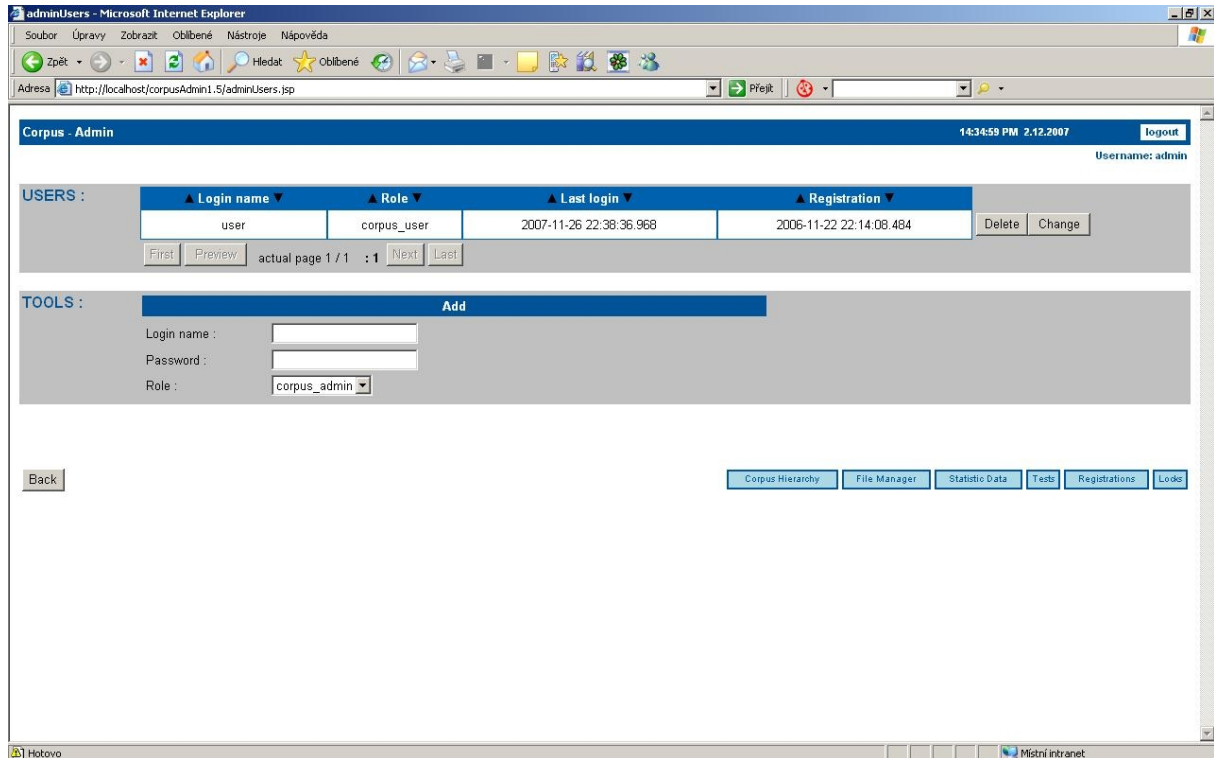
Img. 2.5_1

2.5.1. User accounts

System displays list with all user accounts, which are permitted for *Corpus system.* Only account for actual admin is not displayed, because admin account can be modified only by other admin. This feature is built because of security rules. For each account, system shows info:

Login name – account name

Role – permission for accessing specific *Corpus web* part, there are only two possible values – *corpus_admin* and *corpus_user*, *corpus_admin* role has access to *Corpus Admin* web part, *corpus_user* role has access to *Corpus User* web part

Last login – time of last login of user

*Registration* – time of registration

Admin operations with each account (row):

Change – by clicking on button *Change*, admin can modify account preferences, and confirm them by pressing button *Save* (img. 2.5.1_1)

Delete – by clicking on button *Delete*, admin can remove this account from *Corpus system*

Add account – admin can add new account, he must define mandatory parameters of new account – *Login name, Password* and *Role*, allowed *Login name* length is 20 chars, and the same length is allowed also for *Password* (img. 2.5.1_2)

Detail info – by clicking on specific account, system displays detail (optional) information about account
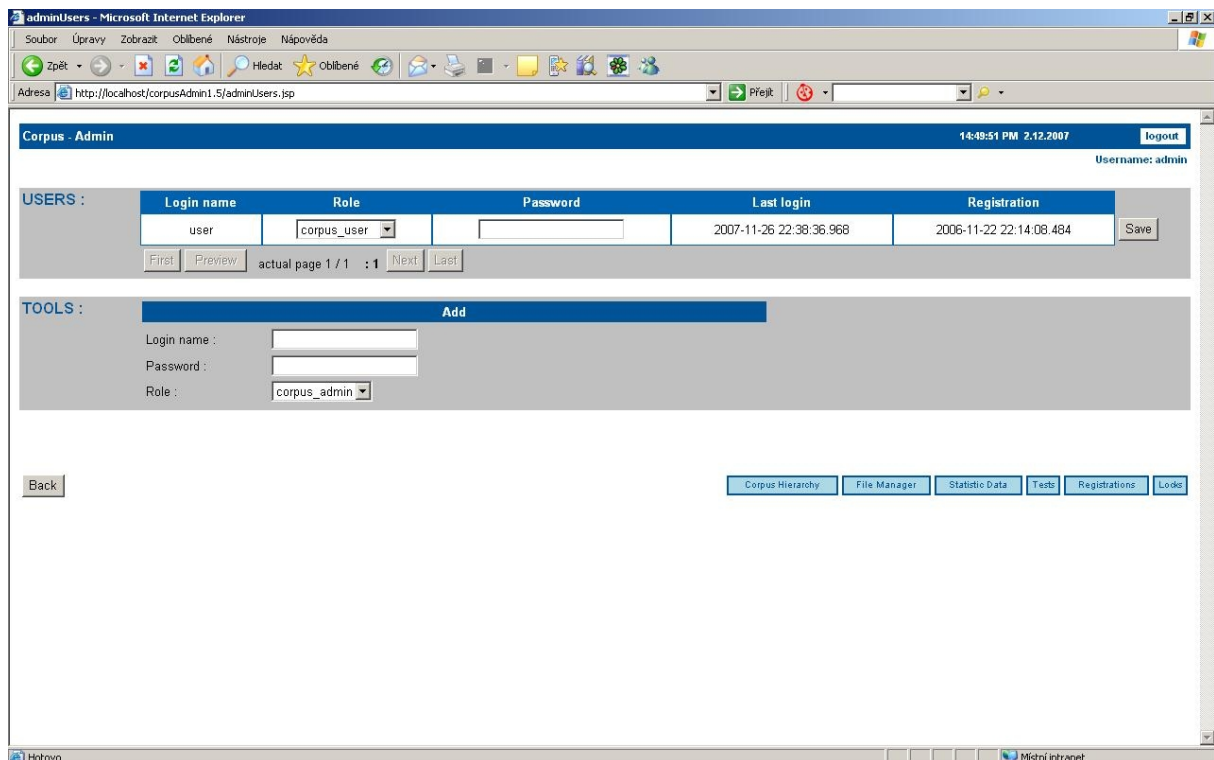
Img. 2.5.1_1

Table operations:

Choose page – admin can choose page, which he wants to display by:
  - pressing button *First* – system displays first page of tests
  - pressing button *Preview* – system displays preview page to actual page
  - pressing button *Next* – system displays next page to actual page
  - pressing button *Last* – system displays last page of table
  - clicking on number of page – system displays exact page

86
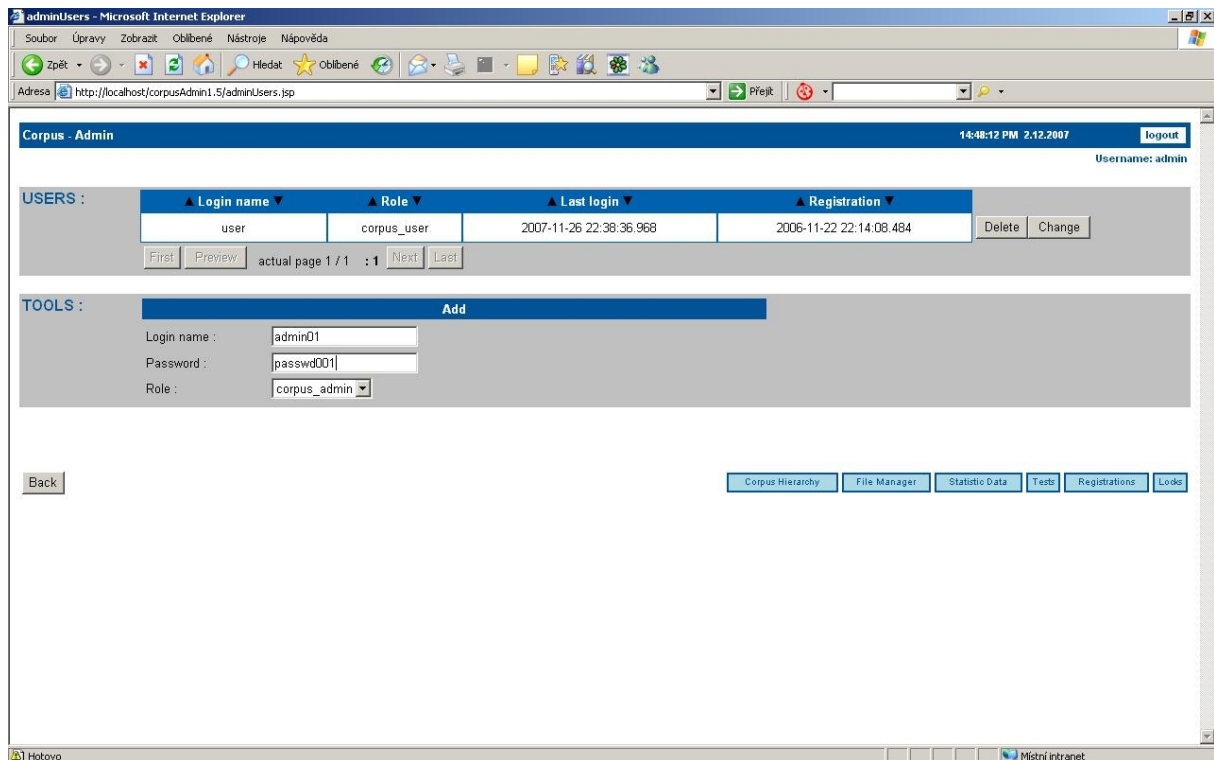
Img. 2.5.1_2

2.5.2. Detail info

System displays user detail info. These account parameters are optional. System shows parameters:

Name – user's first name, this field is mandatory, allowed maximum 20 char-length string
Surname – user's surname, this field is mandatory, allowed maximum 20 char-length string
*Login name* – user's nick name, this field is mandatory, allowed max 20 char-length string

E-Mail – user's contact email, this field is mandatory, allowed max 20 char-length string
Telephone – user's telephone, this field is optional, allowed length maximum 20

Organization – name of user's organization, optional field, name of organization can have maximum 20 chars
Sex – user's sex
State – user's place of living, optional field, allowed max 20 char-length string

Admin can modified all of these parameters and confirm them by clicking button *Confirm user info.* (img. 2.5.2_1)
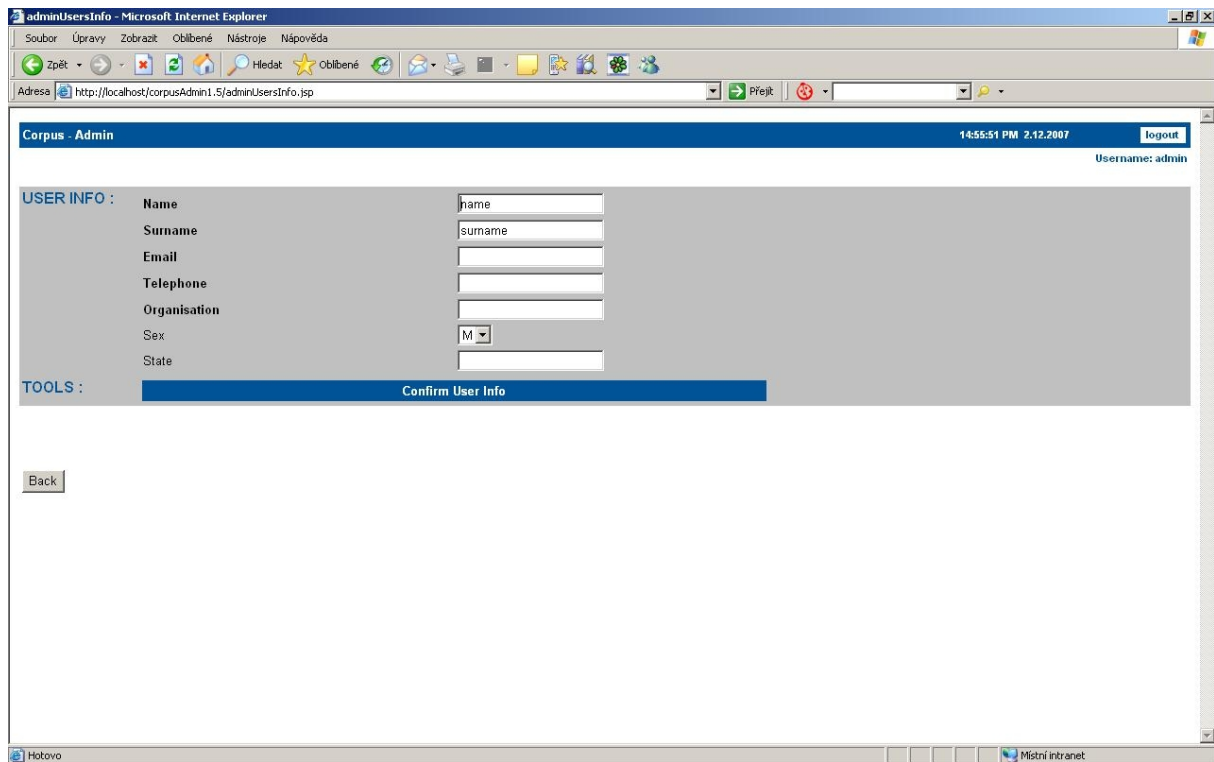
Img. 2.5.2_1

2.6. Registration

System functionality for applying or discarding new registrations.

System displays page with list of new registrations at first.

2.6.1. New registrations

This page shows new requests for getting access to *Corpus User* web part. The table with new registrations displays for each registration following parameters:

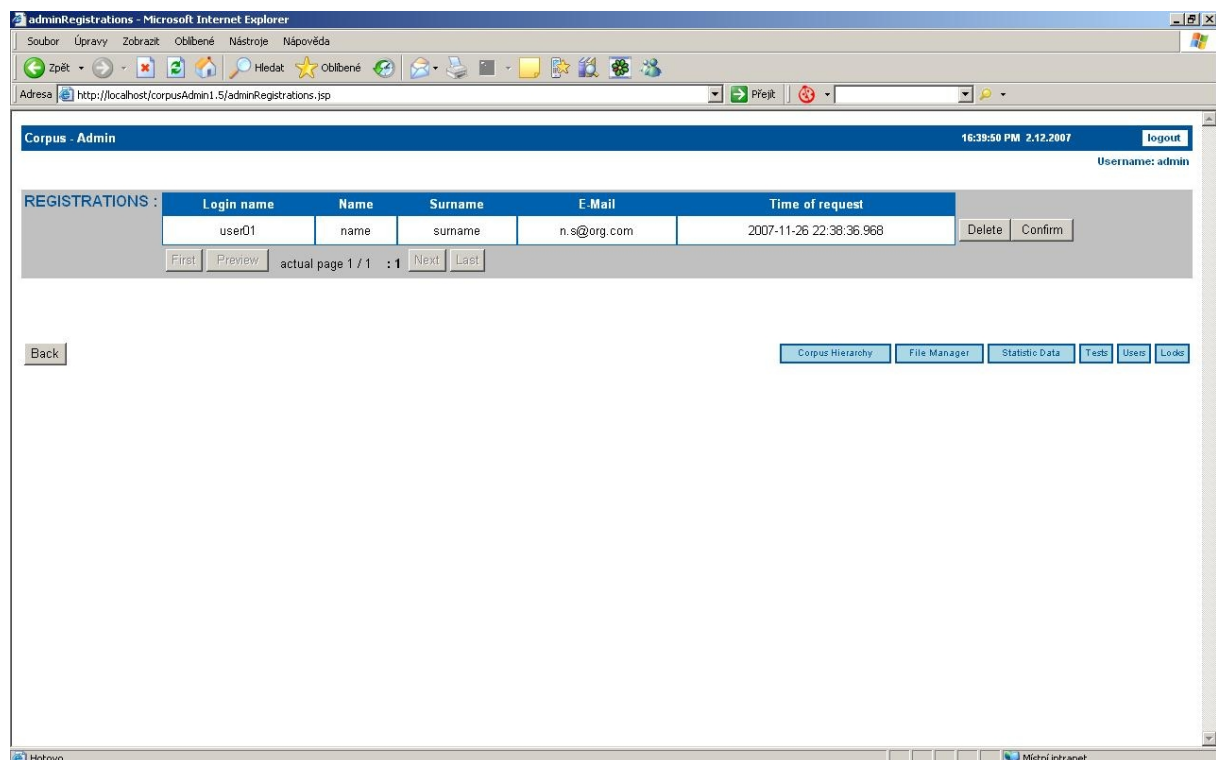*Login name* – new login name for new user
Name – name of new user
Surname – surname of new user
Email – email address of new user, where can admin send important instructions how to login into the *Corpus system*
Time of request – the time, when was request created and sent to *Corpus system*
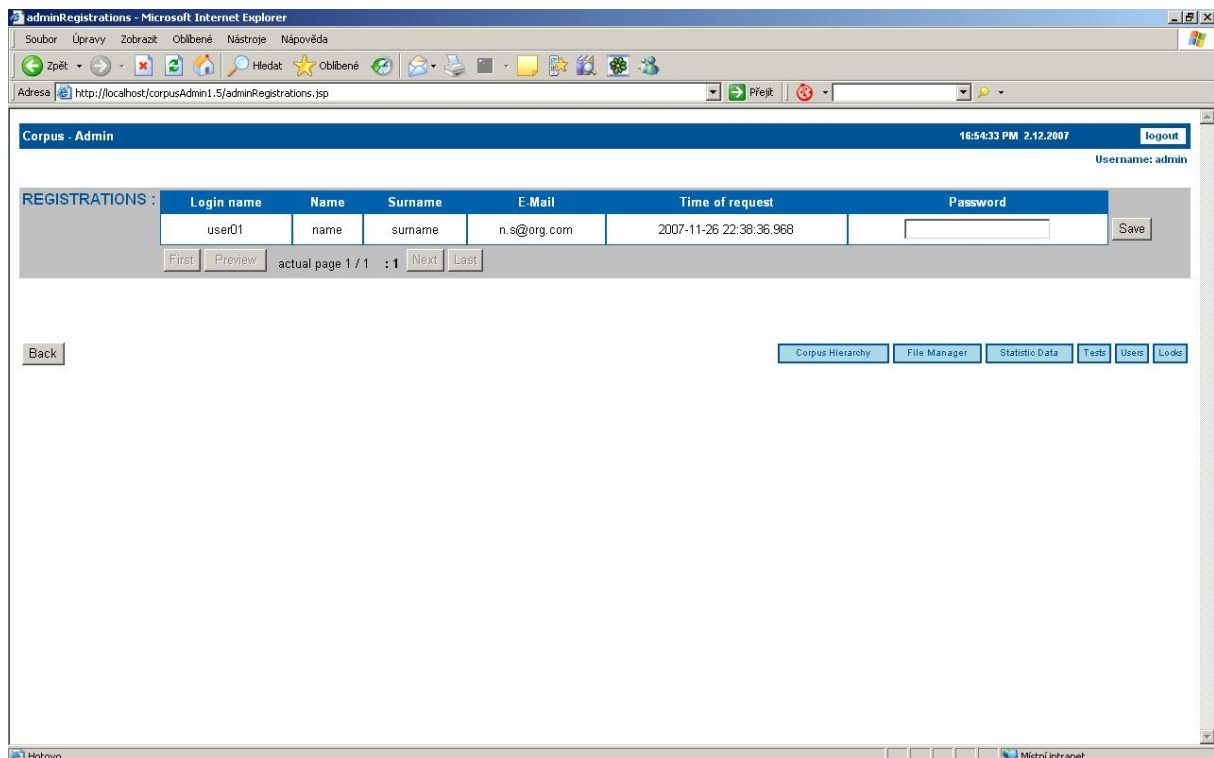
(img. 2.6.1_1)

Img. 2.6.1_1

Admin operations with each registration (row):

Confirm – admin can confirm request for new access into *Corpus User* web part by pressing button *Confirm*, after this confirmation, system asks admin to fill in the password for new account, and after pressing button *Save*, new account is created and also become active, selected registration is then removed from the list of registrations (img. 2.6.1_2)

Delete – admin can discard new registration by clicking on button *Delete*, that means that selected registrations is removed from the list of registrations


Img. 2.6.1_2

Table operations:

Choose page – admin can choose page, which he wants to display by:
- pressing button *First* – system displays first page of tests
- pressing button *Preview* – system displays preview page to actual page
- pressing button *Next* – system displays next page to actual page
- pressing button *Last* – system displays last page of table
- clicking on number of page – system displays exact page

2.7. Locks

System functionality for activating or deactivating *Corpus system locks.*

System displays page with list of system locks at first. (img. 2.7_1)
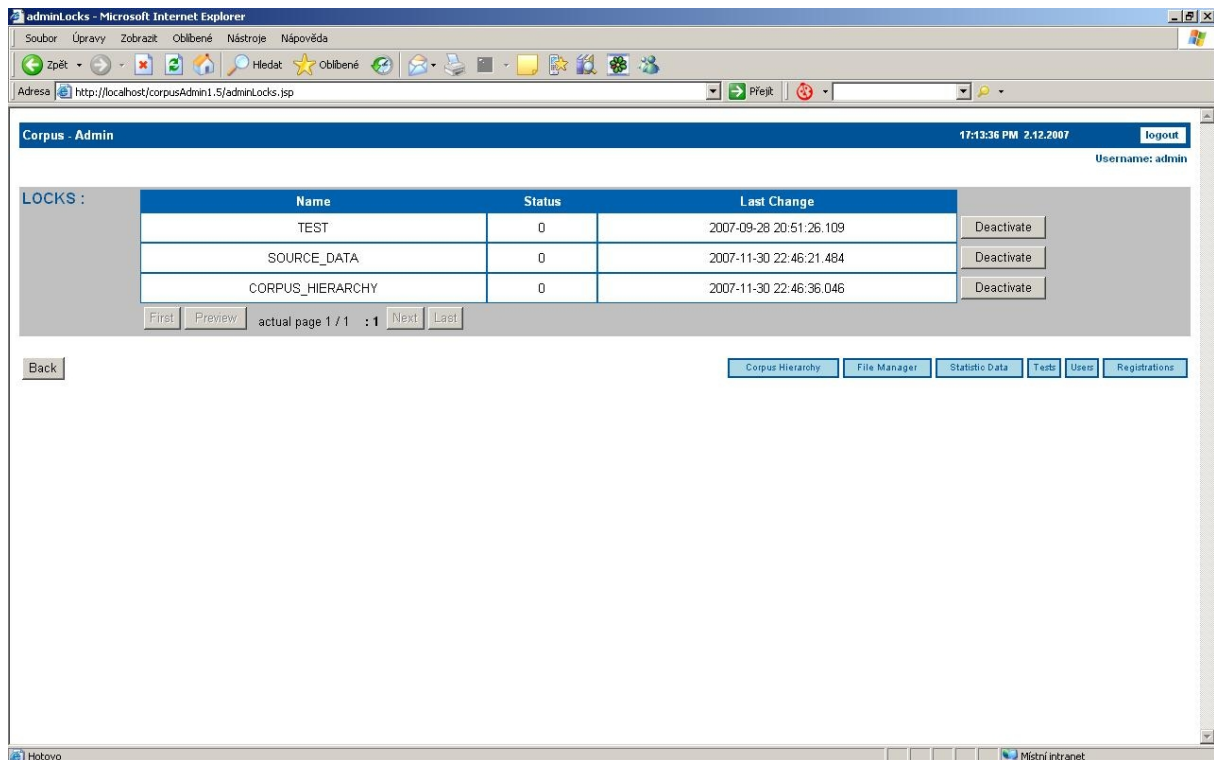
Img. 2.7_1

2.7.1. System locks

System locks page displays all accessible system locks. System shows three main parameters of each system lock:

Name – system name of lock
Status – number for specific status of lock, 0 – unlocked, 1 – write lock, 2 – read and write lock
Last change – last time, when was specific lock used, or its status modified

Unpredictable error in *Corpus web system* can let any of system lock active, with no chance to deactivate it. So that admin has possibility to deactivate any of them by pressing button *Deactivate*.