

Working As A Full Stack Developer In A Large Project

Lea Katalina Kivinen



Author(s) Lea Katalina Kivinen	
Degree programme Business Information Technology	
Report/thesis title Working as a Full Stack developer in a large project	Number of pages and appendix pages 71+1
<p>The report covers 10 weeks of work type of a full-stack developer trainee in a Lean System application project. Its scope concentrates on tasks and problems the author of the report was facing on a daily basis, its goal was to show solutions for these tasks and problems and their thorough analysis that included coverage of technical background and weekly conclusions.</p> <p>The thesis was written in 12 weeks and it covers the period from 29.08.2016 till 27.11.2016. Its goal is to show the working environment and daily routine of a full-stack developer in Roima Intelligence company.</p> <p>This thesis is done in a diary format and is in fact a qualitative research, that included gathering data and its further analysis. Data was collected in a textual form and based on observation and practice.</p> <p>This thesis would be useful for anyone who might be interested in what kind of tasks a full-stack developer is facing in general and in Roima company in particular. It also gives analysis of methods and tools used during in Web Development process, starting from database development and ending with applying styles on the Web page. There are description of some problems any software developer might run into, their solutions and analysis of their reasons.</p>	
Keywords Software development, full-stack development, Lean Systems, programming, web application development	

Table of contents

1	Introduction	1
2	The start situation.....	3
2.1	Analysis of current job situation.....	3
2.1.1	Description	3
2.1.2	Evaluation	4
2.1.3	Development.....	5
2.2	Interest groups in the workplace	6
2.3	Interaction skills in the workplace	6
3	Diary Entries	8
3.1	Week 1 (05.09.2016 – 09.09.2016)	8
3.2	Week 2 (12.09.2016 - 16.09.2016)	12
3.3	Week 3 (19.09.2016 - 23.09.2016)	18
3.4	Week 4 (26.09.2016 - 30.09.2016)	24
3.5	Week 5 (03.10.2016 - 07.10.2016	32
3.6	Week 6 (10.10.2016 - 14.10.2016)	39
3.7	Week 7 (17.10.2016 - 21.10.2016)	44
3.8	Week 8 (24.10.2016 - 28.10.2016)	48
3.9	Week 9 (31.10.2016 - 04.11.2016)	55
3.10	Week 10 (07.11.2016 - 11.11.2016).....	59
4	Discussion and Conclusions.....	66
5	References.....	69
	Appendix 1. Oracle and MySQL differences (Rapid Programming)	72

Figures, Diagrams and Tables

Diagram 1	stakeholders in Roima Intelligence.
Figure 1	Example of flexbox use when creating centered layout
Figure 2	How acceptance TDD and developer TDD work together
Figure 3	Example of batch code written in Notepad and command interpreter
Figure 4	Example of a simple directive
Figure 5	Directive with a template
Figure 6	Directive example with function binding
Figure 7	Syntax of LISTAGG
Figure 8	Example of a LISTAGG function
Figure 9	Example of Select queries used in php code
Figure 10	Examples of HTTP GET and POST methods
Figure 11	Usage of [Authorize] in a C# controller
Figure 12	Example of an applied AuthorizeAttribute filter globally
Table 1	Oracle and MySQL differences

Abbreviations

cmd	Command or Command prompt
CSS	Cascading Style Sheets
DOM	Document Object Model
GUI	Graphical User Interface
hta	HTML Application
HTML	HyperText Markup Language
HTTP	The Hypertext Transfer Protocol
ID	Identity document
IIS	Internet Information Services
Inc	Incorporated
JS	JavaScript
JSON	JavaScript Object Notation
Ltd	limited
MES	Manufacturing Execution System
MVC	Model View Controller
Ro- bocopy	Robust File copy
SQL	Structured Query Language
TDD	Test Driven Development
URL	Uniform Resource Locator

1 Introduction

The writing of the thesis begins on 29.08.2016 and the exact time period for writing it is 12 weeks. The planned completion date is 27.11.2016.

The first week of the working process is devoted to writing introduction and the start situation. After that 10 weeks (till 11.11.2016) are devoted to describing the daily assignments in Roima Intelligence Inc company and solutions of those assignments weekly. And two last weeks are booked for working on conclusions and analysis of the work done during the time of the active data collection.

The background needed in the company corresponds well with the education in Haaga Helia: the application I'm working on has C# back-end with database done in Oracle (so knowledge of SQL is also required). The front-end is an AngularJS application. Also as soon as it's a Web application, HTML5, CSS and JavaScript skills are a must.

The working environment used is Visual Studio, Oracle SQL developer is used to work on the database and the work is shared between team members using Microsoft Visual Source safe where files are checked in and out. The application itself needs to be installed on the server using IIS, and knowledge to configure it is needed to be able to use the application. Though another team is responsible for the installation of the application and for the computers' performance to allow the programming teams to concentrate on their own work.

There are several teams working on different parts of the project. Each team works on its own solutions and is responsible for developing new version of the product, maintaining and perfecting the current version and taking care of the customers who still may be using an older version of the product.

The company itself is called Roima Intelligence Inc. Their motto is "Roima boosts its manufacturing and intralogistics customers' operations towards higher profitability". The solution I'm assigned to is called Lean Systems and it is a "solution for manufacturing and service operations management." Like it says on their website, Roima Intelligence Inc. is a company specialized in improving its customers' competitiveness. The company's name is rather new on the market, but the company itself was created a while ago and has "extensive experience within Finnish manufacturing and logistic industries". An idea of creating the company came to Korona Invest private equity company. They've researched the mar-

ket and realized that there are too many small IT companies that serve logistics and manufacturing business, and decided to invest into creating a large company that would be able to answer the demands of the largest manufacturers in Finland. The process started by getting majority ownership of Oy Delta-Enterprise Ltd and rebranding it into Roima Intelligence Inc. The company is fresh and fast-growing, which shows that Korona has succeeded in its plans. The company has gathered four branches that are now parts of Roima Intelligence Inc.: Operations management business (Lean System), Service business (Lean Forward), Intralogistics (Done Software Solutions) and MES Business (Roima 2016).

2 The start situation

2.1 Analysis of current job situation

2.1.1 Description

My current general list of tasks includes in itself:

- correcting errors in the application
- perfecting and refactoring the existing code
- researching and implementing new features.
- writing work tasks and descriptions

I could describe the tasks as the tasks of a full stack programmer, because they include both working on lower application levels and front end Web development. And it can be noticed in further description of tasks I have done.

Correcting errors in the application. If one or more features of the application do not work like they are supposed to, a technical task to correct these errors is made. Errors can be of different types: from simple CSS bugs to server-side problems. But for each case a technical task is made, where the description of the error is listed together with a description of under what circumstances the error occurred. The task is then assigned to one of the team members.

These kinds of tasks are the best way to learn the application and system algorithms it uses. While making error corrections, I've gotten to test the existing code thoroughly, following the debugger step by step. While correcting CSS and html errors I have learnt to use the browser debugger. In the case of server-side errors, I got to work with both the browser debugger (in case the error was caused by AngularJS) and the Visual-Studio debugger (in case the error was on a deeper application layer) and followed the code in order to find where the error occurred and then looked for reasons why it occurred.

Perfecting and refactoring the existing code. This kind of work is also important for the application. To make it run faster and smoother, code refactoring is needed along with other methods of code optimization. Sometimes there is hard-coded data which needs to be replaced with other better and more general methods (because hard-coded data always leads to copy-paste of code parts which should always be avoided). Sometimes there are similar methods that could be placed into one, etc. This kind of work needs a lot research and knowledge of how the application runs, when and where certain methods are used and which techniques of code optimization can be implemented in each particular case.

Researching and implementing new features. The project is growing and new versions are implemented regularly. Before implementation, planning, research and development are done. As soon as a new feature is planned, a research phase begins where the feature should be analysed (this includes analysing different kinds of materials such as related literature, Internet forums and examples, also the feature's own documentation). After thorough research a test code is written (if needed) or if it is a larger feature that somehow affects the whole project, a test implementation is done on a local computer. There will usually be a lot of errors caused by such implementations and they have to be corrected before the version leaves the local machine. When everything is working properly, the development version is also updated and the feature stays in that version before the next release.

Writing work tasks and descriptions. All the work done is documented, starting from planning of each task ending with all test results. The documentation helps all the team members to follow the steps of each change made to the project, and it also helps to find when the change was made and who was the person responsible for that change.

Skills and knowledge needed in the tasks. Apart from programming skills in C#, knowledge of SQL, CSS, HTML5, AngularJS and other skills such as the ability to write clear descriptions, the ability to look for necessary information and use it, good logical thinking and good communication skills are required. It is also important to foresee possible outcomes of code changes and know how to test the code thoroughly to avoid all possible errors.

So far, the most unknown area for me has been AngularJS. I have never worked with this framework before, so when I started my work at the company, it took me some time to get familiar with it. Also, there were some new techniques used in CSS. I have also learnt about browser debugging.

2.1.2 Evaluation

I have received positive feedback from my colleagues about the tasks I have managed to complete. I could say that the position of trainee is most fitting me for now, because I still require some instruction about the assignments I'm getting, though it is enough for me to get a general description. After that I manage to do necessary research on my own, asking advice and help only in cases when I'm not able to solve the problem myself for a while. It sometimes happens that the solution of the problem is also unknown to other members of the team, so I continue my research and look for solutions online.

I have managed to complete all assignments I received and based on that I could say that my performance meets the requirement level of the assignments. Even though I haven't gotten any tasks that require deep knowledge yet, I still grow as a professional and constantly develop my skills and increase my performance. Based on these facts I could evaluate myself as being very close to a skillful performer. I can't tell I'm already exactly at that point, because I sometimes still need examples and instructions of the task I am about to start. As soon as I'm 100% familiar with the product I'm working on, those instructions won't be needed anymore and I will be able to pick and complete assignments in a more independent manner.

2.1.3 Development

Stage of vocational development. As I wrote above, I feel that I'm very close to be a skillful performer. Before I become one, I am well fit to be a trainee who still needs some kind of instruction and teaching, but capable of completing the work with a certain level of skill. I am still at the beginning of my programming career and I clearly know there are still a lot of things I am about to learn.

How is it visible in actions and work. I'm good in following instructions and in research. I believe I can complete all given tasks on a good level. I realise I'm not yet capable to solve some issues with the efficiency a skilled professional would, but I'm challenging myself to move towards it. Completing all assignments to the best of my capacity is my top priority. There are some moments in my work when I don't pay necessary attention to some aspects because a lack of knowledge in those fields. This is yet to be perfected. But I trust myself to be able to learn from my own mistakes well and in most of the cases it is enough for me to make a mistake once to learn from it and to avoid similar ones in the future. So all in all I see myself as a growing professional and I believe that this growing is efficient and beneficial for both me and my employer.

Where should investment in the future be in? What is still to learn. There is a lot I can still learn. I need to work more on AngularJS technologies, to deepen my knowledge in C#. I do well know that my skills in CSS still need a lot of work. And I would like to concentrate more on knowing what to do, instead of trial and error methods, which I'm still using from time to time. So all in all I'm planning to invest in those areas, mainly Web technologies. Also it could be beneficial for me to learn about Cloud services and testing. I like working as a full-stack programmer, because it gives a very good picture of the whole application and also tasks related to this kind of programming are very diverse and require good level of expertise in wide areas which would give me very strong professional skills.

2.2 Interest groups in the workplace

Stakeholders are represented by all of the company's customers, its ownership, board members, management team and personnel. In other words, by everyone that affects or can be affected by organisation's actions.

The company's customers are in the food production and in technology (machine and instrument manufacturers, engineering companies) industries.

The company's ownership is divided between Korona Invest services fund and management and key personnel.

The government also could be named as a stakeholder, because the company acts accordingly with Finnish laws regulated by the Finnish government.

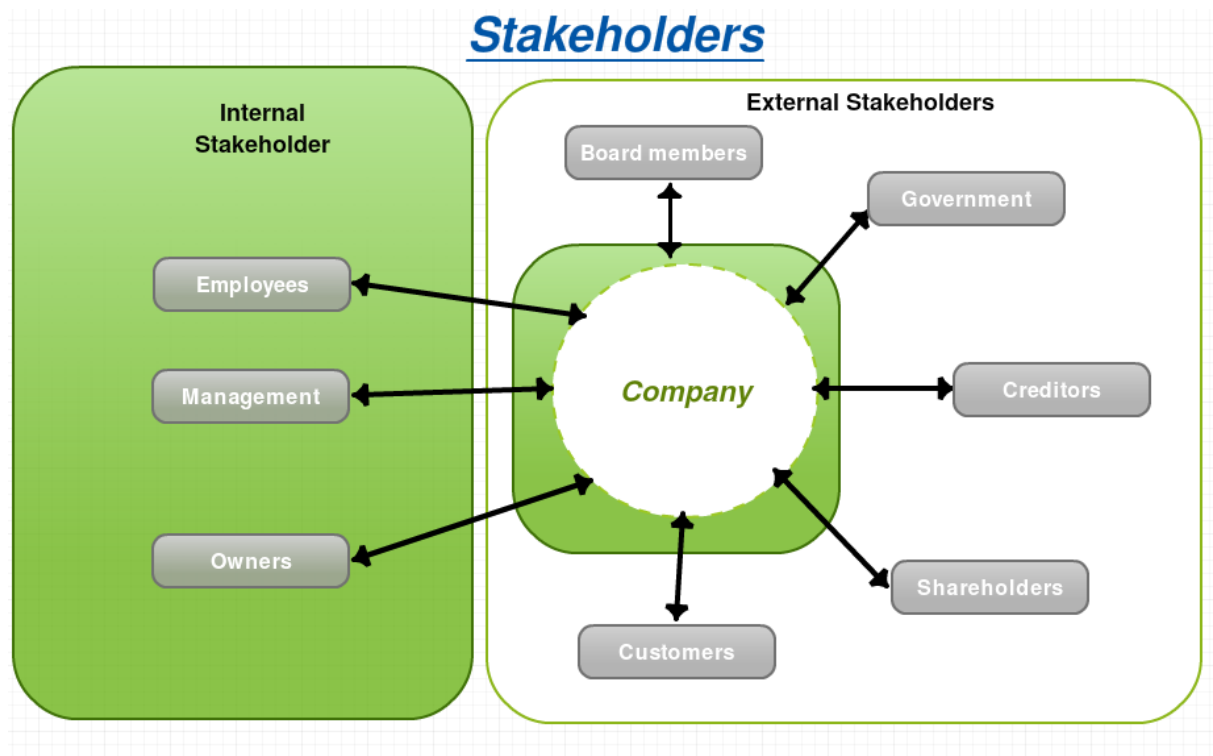


Diagram 1. Stakeholders.

2.3 Interaction skills in the workplace

The largest part of interaction at my workplace occurs with my teammates. We discuss current assignments, the problems and complications connected to the assignments and overall just general topics of possible future development. All these discussions happen mainly during breaks, because there nobody is concentrating on just their own doings. If

there is a bigger problem one of the trainees can't solve, there is always a possibility to ask for assistance from a more experienced workmate.

The general information flow about work process can be found by any employee on the company's intranet website. There it is possible to find planned and assigned assignments, wishes for development, watch your own ToDo-list and pick up assignments that are planned but haven't been assigned to anyone. The person assigned to the task is responsible for proper documentation and needs to describe all the steps done during the work process.

When there is a bigger topic to discuss and plan, a meeting for the team is set up. There are also monthly online meetings about the current situation in the company and its plans for the near future.

Apart from these interactions that are tightly connected to the work I'm doing, there are also others concerning paperwork, social events, technical issues and other general topics. Most of the communication is done online through the company's own internal channels and also by email and Skype.

As to communication with customers, my job position doesn't yet include this important part in itself. The only interaction I have with the customers is with their systems, when there is an update to be added and it is then installed to the customer's server.

3 Diary Entries

3.1 Week 1 (05.09.2016 – 09.09.2016).

05.09.2016 and 06.09.2016.

In the end of last week, I was working on creating a C# method that would handle data retrieval generically (for any class type that needed data access). The method was written in two parts: one handles the general response and creates a list of elements and another method, that fills the element's properties with data from the database. This is not only a generally good practice: to separate method type from each other, but also is used in the project: there are types of controllers that don't use the response system, but handles each element separately, so that is why another method could be used there directly.

The main task of the day is implementing the mentioned above methods in all controllers that can use this kind of data handling instead of existing hard-coded data access. Also if needed, models need to be changed accordingly. Also each controller needs to be tested to check the data gotten with generic method is correct and database column mapping is done in the right way. The amount of code needed to be refactored is rather large, so the task is planned for several days in advance and objectives are going to be the same.

In the end of the first day less than a half of controllers and their models were refactored and tested through. There were no big issues during the task performance apart from small errors that were emitted, like some null and empty checks were missing, also the check for true/false values had to be rewritten so, that only 0 would give a negative value.

By the end of the second day there was almost everything done apart from a couple tricky controllers that needed a few more checks some more precise testing. The reason for that is those controllers are still partially under construction, which makes the testing more difficult. The main issues were an error of false "true" value, gotten by a Boolean check. It appeared to be my mistake in value declaration (the declaration was done outside the inner properties loop function, so the value never changed if it didn't get into the check. And it never did, if the property type wasn't a bool value).

This task didn't require any new knowledge, so no additional research was needed. Also my skills were sufficient for the task, the only thing I would like to develop is to predict the behavior of a function better to avoid suck mistakes as I got with a Boolean check.

07.09.2016.

The morning is set to finish the previous task. Also there is another correction to be done: a command, that copies files by Robocopy, copied extra files and folders. That is set to be removed. No additional tasks are planned for the day.

The rest of controllers were refactored and tested with only one minor problem. One of the model's database column mapping was done by a previous programmer by a name that was never used in the table (this error could have come from some older version and never noticed, because the mapping was never used before). I haven't noticed this naming problem right away, so it took me a while to understand why one of the values, that came from the database was null. First I had to open the SQL Oracle Developer and look for the requested table and make sure there was a column of the required name). And only then I've checked the model and noticed, that the required parameter had another column mapped to it (the column mapped wasn't even in the table). After correcting this error everything started to work as it should. I have returned all the required files to the server and wrote a report about the work done in the system.

The next task was to remove unnecessary files from the .cmd file. The shell scripting was already familiar for me from previous work experience, but I haven't had any experience with robocopy. So the first thing I had to do, was to look for some short manuals of the tool. Then I formed the picture of what is to be achieved: the command like it was, took the certain file types and copied them to a new location. All the extra files belonged to those types, so that was the reason for that behavior. I have decided to rewrite the code so that all the files are copied apart from the files and file types that aren't needed during the copy process. The list of the files to exclude was shorter in the end. To complete this task I needed to get myself familiar with robocopy's syntax and for that I have used materials from the Internet. The commands I used were /XF and /XD. Also I refactored the code to be on the same line, which made the whole coping process also faster.

08.09.2016.

The main goal of the day was to establish tasks to do. As soon as during previous days I have completed everything I had on my work list. There are still a lot of opened tasks in the system and I was looking for something I could start without extra guidance. I went through a list of error-correcting tasks and picked the one which in my opinion was manageable for me to complete. Pretty soon I got a couple of more error correction tasks on

my lists and a few ready tasks which were needed to be tested that were assigned to me by my colleagues.

Errors were all minor, so it just took very little effort to correct them. One was just a simple html mistake, another (the application didn't check for null field values after the field was once saved) took a bit more time to fix, because the reason of the error to occur wasn't very clear for me at first. But the testing tasks were useful that way, because in one of those tasks there was a description of a controller with a method that were likely to cause the error. The theory was correct and after figuring out the reason of the error, there was no problem to correct it by adding the null check to the correct place.

Testing tasks also didn't bring any problems. If I couldn't find anything wrong with the application concerning the task's edits, I would write a comment "tested, works" and marked the task as "ready". If there were still problems, that I would describe them and return the tasks to the list of a person, who was working on the tasks.

09.09.2016.

During the previous day the application was prepared for the new version and apart from other changes, a namespace has been changed. It brought an error with local storage where the namespace was used as a part of String to save user data. The error causes problems with data load and also prevent new data from saving into database.

So the goal for today was to determine the reason for the error and work on its correction. The plan is to change the namespace of an entity on the local storage programmatically and also to place the change as early in the page load as possible.

I started the task with debugging to determine a place where data is saved to a local storage from entity manager. When the place was found I used a local storage key to get the item. When the Item was found the idea was to determine if it has the wrong namespace in it. For that an angular function `indexOf("string");` was used. The function return the index of the string element and -1 if none is found. Then a function `replace("what is replaced", "what to replace with");` was used. After the item is modified, it needed to be parsed back to JSON object and set back to the local storage with a function `localStorage.setItem()`.

During the debugging I got a problem that the Item value was never changed. The reason for the error was the fact that replace function doesn't modify the given string, like I though it should, but it returns the whole new string. After assigning it to the variable I got the local storage value to change for a correct namespace.

Today was probably the most useful day for the whole week. I got to learn about new AngularJS functions (indexOf and replace) and also for the first time got to work with local storage items and learnt the process precisely, so after today I will have a good picture of how and when the values are saved to the local storage, what functions are used and how the data is gotten from there.

Week 1 evaluation. Discussion and technical background.

During this week I became more familiar with the application I'm working on. The refactoring of all controllers that are used by the application helped to get very clear picture about data flow in the application. Even though this task hasn't developed my programmer skills, it still developed my general knowledge that will be very helpful in my future work. Refactoring task also pointed at me some of my week points, when writing code. I need to be more careful with null values and always make sure that, when needed, values are reset in loops.

The description, given in the book "Refactoring: Improving the Design of Existing Code" is following: "Refactoring is the process of changing a software system in such a way that it does not alter the external behaviour of the code yet improves its internal structure". I believe that with my implementation, I have achieved the main goals of refactoring: the code remained its functionality, the near-duplicated code was replaced with general method that is able to take care of any class and the method I have wrote has passed the tests. The code needed refactoring, because about 12 controllers were using the near-duplicate code patterns and wasn't following the general practices anymore. So all in all I believe I managed well with this task and there is nothing I could do in a better way.

Another thing to learn for this week was robocopy and its methods. Robocopy is an own Windows tool for heavy-duty file management. The name of the tool means Robust File Copy. It is a generally faster tool than xcopy and simple copy, uses unconventional syntax that allows the tool to work with two directories simultaneously (it is also robocopy's main feature) (TechNet).

There are many other tools for file management, but I am not eligible to change anything in the already ready script that uses robocopy, so this assignment couldn't have been done differently. Also I couldn't find a better option for a hta (visualScript based) GUI application, which should run an external file than adding a batch file. Power Shell might have been a better practice for this case, it also gives an option to build own graphical interface, but would have taken rewriting of the whole program. I will suggest this option to

my company later in my work, when there would be enough time to do this kind of rewriting.

As I mentioned above, the most useful experience this week was working with local storage and establishing the error reason in the application. I got to learn the code used in the application to access the local storage, explored new functions of AngularJS that are used to work with strings like looking for a certain character in a string and replacing a part of the string with another one. The same functions are used with arrays, but it is very logical to use them with strings because strings are themselves arrays of symbols, like it used to be in older languages (Fowler & Beck 2002, 73-79).

Local Storage data is saved in a JSON file in a string format. When it's accessed it's parsed back into a local storage item. That is why my team and I couldn't come up with a better option of handling the name space problem in the local storage than replacing a part of the string. This way no user data would be corrupted, because when parsed back, it returns local storage items in a right format with a correct namespace in it.

3.2 Week 2 (12.09.2016-16.09.2016).

12.09.2016.

The plan for the day is to continue working on the local storage and make it working. The last week's efforts brought me to the point that the function I've written seemed to work, but still the same error popped up when tried to save information. That is to be fixed as soon as possible. Also I got another task on my list: to replace an icon button with a text one and to change its functionality. But priorities remained: first problems with local storage, after that everything else.

I have started by trying to debug the application one more time. The result was the same: the entity I could see in the local storage has the correct value, several times the function gets entities with a correct namespace, but when breeze own function was used for the second time to get entities from the local storage, wrong namespace was there. At this point I've asked for help from my colleague. He went through the debugging process again with me, we got the exact same result. His idea was the breeze had somehow its own storage where it saved the values, and entities there were never changed somehow. The idea seemed to be right, so I went to the breeze own net page and started the search for the answer there. At the same time, I tried to look for information on corrupt data in the

storage and how to fix that. Apart from clearing the storage, there was no useful information. Though breeze own pages had information on how to import and export entities from the storage, so the plan how to achieve the entities' replacement in there also was born.

This day was generally devoted to searching and looking for useful information. I've learnt more about local storage, about how to work with breeze's Entity Manager and how to access cached data with its help. And even though I couldn't solve the problem during this day, the plan for the next one is formed.

13.09.2016.

Today's goal is to continue on working on the same task till the problem is fixed and start the interface task next. Yesterday's research showed, that the error lies in the application code, and it could be corrected by accessing breeze's own storage, but wasn't necessary because it can be fixed on upper levels. The plan was to catch the error by very precise step by step debugging.

It showed that the items from local storage get the old namespace after they are accessed from the local storage the second time. This was very confusing, because the first value returned from the storage was with correct namespace, the function of string search was accessed and the value changed, the string returned (the part that have the namespace in it shown in the debugger had the correct namespace after the change), still the items got the wrong namespace value all over again. After running the same code for several times, I realized that the check for old namespace was always triggered, which shouldn't happen if the value is changed.

It made me wonder if there is still a part of a string somewhere that gives the wrong namespace value in returned items which for some reason doesn't get changed. In this stage a help from my co-worker was very useful, because he run the code with me a couple of times, wondered the same with me, but then he opened the debugger's window which showed that the string stored into storage is much longer, and there are still a couple of entities with the wrong namespace. That was the answer to the check that was always true in the end. Google helped to find out that the main reason for the whole situation was the fact that *replace("a", "b")* function replaces just the first appearance of the string and doesn't search any others. The whole problem was resolved as soon as a global pattern for the string replacement *replace(/a/g, "b")* was used.

14.09.2016.

The tasks for the day were to do the button replacement that was set earlier and another was some CSS fixing (lines went a bit off and it needed fixing).

The button replacement was in a way an easy task. I just had to remove the old cancel icon and replace it with the general styled text button with an icon in another place while keeping the functionality. The only challenge was to assign the functionality to another script file (the icon are calling one js.manager while text buttons the other). To get the right styled button I just had to copy it from another page in our application. The functionality assignment also didn't cause any problems, I just copied and changed the code by the example of other buttons.

The fixing that needed to be done with CSS involved correction of <div> box sizes. The problem was that when resizing, the part of the buttons that are supposed to fit on the line didn't, because another <div> was taking all the space. The reason for that was using col (column) attribute. First <div> was taking 6 columns and another one took the rest 6. Apparently it wasn't enough for several buttons to fit on the line. The top row of the buttons was similar to lower one, when the list on the page got too long. So first I checked what attributes were used for the lower row as it scaled like it supposed to. That row used flexboxes, so the whole concept was different from the columns.

My tries to implement flexboxes together with columns didn't really give any good result, so in the end I just made different distribution of the column space between two <div> elements by giving one of them 2 columns and leaving 10 for another. The behavior of the element when scaling wasn't the same as flexboxes but at least it made scaling of the element more even.

The rest of the working day I was reading documentation about testing that I am about to start tomorrow.

15.09.2016 – 16.09.2016.

The new version of the application is going to be released soon. So, as mentioned above, the task for the rest of the week was to do thorough testing of the application. The application is recommended to work with certain version of browsers and that is also mentioned in the application's documentation. So my assignment was to test the application's com-

patibility with two different browsers (others were taken care by other people). I was assigned to test Microsoft Edge and Google Chrome. I could just use the version I already had on my computer which were Edge 25.11082 and Chrome 53.0.2785.116.

The testing needed to be done on a local machine (we usually use virtual one for developing), because it would be the closest to the real user's environment. As a guide I used the document where all steps of testing was described and also the results needed to be put in the same document. For every wrong browser behavior a description needs to be written and the task should be made.

I'm not eligible to write details about some errors I've found during the testing because the directly concern the features of application. One of the was actually pretty big and didn't depend on any browser. So it is good it was found that early and went straight to the competent person for fixing. But I could say that chrome is responding to the application very well and there were no browser-related problems found during the testing. Edge, on the contrary, showed that it is still a row browser, and there was a lot of weird browser-related behavior. Here are some of the problems that are more general and can be described. During the testing, some CSS styles just disappeared without any reason for it: the debugging window showed all the styles in place and there were no indication for the style not to be in use. Still it randomly disappeared from the page (there was no way I could repeat the error with some precise steps). Another error appeared when I was testing input fields: Edge offered a drop down of all previous user inputs like logins, numbers and others for any field, even fields that are supposed to accept only integer values. And one more small error was year being in shown in a wrong money format (like 2 016 instead 2016).

In the Friday evening I documented all of the founding and made works for them. There is still a large part of testing left for the next week (offline functionality).

Week 2 Evaluation. Discussion and technical background.

Working with local storage was completely new to me. Before this week, I was completely unaware of this kind of functionality overall and that it was implemented in our application. After this week, a lot of processes happening in the application became much more clear to me and I started to realize how the information is passed between sessions without being saved to the database.

This kind of introduction was very confusing, because even my colleagues had no idea that changing of namespace would lead to such circumstances, and they were very serious because wrong namespace in metadata just blocked all the data saved into local storage during the time previous namespace was in use and prevented it from showing.

This kind of issue is not a big deal for the tester or a developer, because we get to clean local storage entities all the time, but for a common user, who just trusts everything to be working and doesn't save information because of local storage, it can be very dangerous.

Before I have started working on local storage, we have discussed an option of emptying it for all the users, but that could lead to a loss of sensitive data. So in the end this option was removed from consideration right away. Of course, it would be much easier for us, developers, just empty the storage together with a new version, but clients come first, so there had to be a way to manipulate metadata.

And there was one: `EntityManager` was the tool to manipulate the data, saved in local storage. "The *EntityManager* is the gateway to the persistence service and holds a cache of entities that the application is working with, including entities that have been queried, added, updated, and marked for deletion. The `EntityManager` is the core class in Breeze, and this page discusses its primary capabilities" (Breeze). It has a lot of methods to access and manipulate the data in the local storage, and I tried to find the method to manipulate the breeze's local storage data. In the end it became clear, that the error was in the metadata, not the breeze code itself. But even though the error's reason wasn't the breeze's code, it was still very useful to do the research about the local storage and understand the principles of its work. The local storage is very powerful tool to keep users' data between different browser sessions, for example user preferences, unsaved data (keeps the data in the storage even if the page is refreshed, until the data is removed manually or programmatically from local storage or saved).

Another task of the week was CSS manipulation. The `col` attribute also belongs to the bootstrap CSS.

Bootstrap allows to use HTML elements and CSS properties. That is why it requires Doctype of HTML5. Bootstrap was developed to support mobile versions and its layout is quite flexible and responsive to different screen resolutions. According to the bootstrap own website, the project was developed with keeping the rule of mobile devices come first in mind (Bootstrap).

The <col> attribute belongs to the grid system, which is quite flexible and handy. Column layout allows to create the webpage using a series of rows and columns. That is how the creators of bootstrap present the working of the grid system:

“Rows must be placed within a .container (fixed-width) or .container-fluid (full-width) for proper alignment and padding.

- Use rows to create horizontal groups of columns.
- Content should be placed within columns, and only columns may be immediate children of rows.
- Predefined grid classes like .row and .col-xs-4 are available for quickly making grid layouts. Less mixins can also be used for more semantic layouts.
- Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on .rows.
- The negative margin is why the examples below are outdented. It's so that content within grid columns is lined up with non-grid content.
- Grid columns are created by specifying the number of twelve available columns you wish to span. For example, three equal columns would use three .col-xs-4.
- If more than 12 columns are placed within a single row, each group of extra columns will, as one unit, wrap onto a new line.
- Grid classes apply to devices with screen widths greater than or equal to the breakpoint sizes, and override grid classes targeted at smaller devices. Therefore, e.g. applying any .col-md-* class to an element will not only affect its styling on medium devices but also on large devices if a .col-lg-* class is not present.” (Bootstrap).

The flexbox is also a bootstrap directive, but it is newer and has more features. Flexbox is simple, because it is order-controlled grid. According to the article Using Flexbox: Mixing Old and New for the Best Browser Support, the main problem of flexbox is Internet Explorer support. It can support only version 10 and up. Otherwise it is a very powerful tool to get the nice aligned user interface. And as soon as a lot of application are going mobile, flexbox is soon to become one of the most popular CSS tools to use in Web developing, because it's works with mobile extremely well (Coyier, 2013).

Below there is an example of code to perfect centering. Very simple, if using flexboxes:

```
CSS
.parent {
  display: flex;
  height: 300px; /* Or whatever */
}

.child {
  width: 100px; /* Or whatever */
  height: 100px; /* Or whatever */
  margin: auto; /* Magic! */
}
```

Figure 1. Example of flexbox use when creating centered layout (Coyier, 2016).

3.3 Week 3 (19.09.2016-23.09.2016).

19.09.2016 - 22.09.2016.

The whole week was dedicated to testing with a deadline on Friday. I decided to unite these days, because in all of them I was working on the same task.

During Monday I continued to test the offline functionality of the browsers, because last week it still needed some fixing and now it supposed to be ready. Same as during previous week I followed the given instructions. During this testing there were several a general problem: the page couldn't be refreshed while offline. The application got just stuck on the welcome screen and couldn't load further.

There was also several Edge specific problem that some of the application features weren't working like they were supposed to. I wrote reports and made works for all the problems. The browser-testing, that was assigned to me was done. So I could start with general testing.

For that there is also a separate document that gives very general instructions like "press on every button, open every link, add an entry, edit entry, delete the entry". There is also a list with parts of the application to be tested. I've consulted a colleague about the one I should start with, and, when assigned one page, started testing.

The testing took me three days. It is pretty slow process to thoroughly test all the windows and functions of the application. And it was also complicated by the fact I needed to write separate works for every error I have found. There were also cases when I wasn't sure if it was an error or some kind of application feature. For that I needed to contact a person who had details about the issue. Unfortunately, there is not much to write about during these days. Testing is very technical and monotonous process which means clicking, clicking and more clicking. After finishing one partition of the application, I consulted my team which I could next (to make sure someone else isn't already doing it). Overall I have tested three partitions and have found some issues during the carrying-out the task. Some of errors were high priority for fixing and other were marked with low priority which meant they went in the end of the task list and will be fixed, probably, never. And as soon as testing is very close related to the application own features, I'm not eligible to describe them. I am going to mention those errors in the following days in case I am going to be the person to fix them.

23.09.2016.

Today's main goal was to test and solve the problem with the batch file that (as testing by one of my colleagues showed yesterday evening) didn't unzip an archive that it should've (apart from other things the file does). Also I've got some more testing to do. This time it was about testing the tasks marked as ready for testing (means some development or error-correction is done, but isn't taken yet into production or new version under development), because thorough check if everything is working needed to be done.

The batch file had a higher priority, so I started with it. The file was a script written for a hta application which is a GUI Installation file with simple checkboxes. The hta application launches the batch file with passing the arguments from the checkboxes. The description of the error was the following: the application never do the unzipping and never even start the related part of the script. So my first Idea was that there was something wrong with the part of the code that answers for the unzipping, or there is a problem with the parameter. I checked the parameters and arguments for the unzipping and everything looked fine. Then I started to check if the code had any errors. I tried to run the script on my machine and indeed I couldn't see that if-else block that is supposed to check for the zip file argument is reached. There was an error in the syntax: a bracket didn't have any space before variable and it. It looked to me after adding the space, the problem was solved. By adding pauses and echos to the script, I have checked that the necessary part of the code is reached when the right checkbox is checked, and wrote to my colleague he could test the application again.

While waiting for the answer, I managed to test one of the tasks. There were no problem with the functionality, so I marked it as done. Meanwhile got an answer from my colleague, that the error still remained.

I decided to properly test the code and go through the whole process of extracting (which I haven't before, because as soon as I know that part of a code was tested before and it worked) I had to create every folder like the program demanded to. The paths for the archive and the program are hard-coded, so they needed to be placed in specific places. I carefully followed the paths when creating folders, placed the archive and the GUI application with the batch file to the right places. Run the hta application, chose to create the archive and pressed Run. And everything was fine: the argument was passed like it was supposed to, the files were extracted into a temp folder and then copied into the final des-

mination. I wrote the email about the results to my colleague. He tested it again and answered, that the issue still remained. We decided that we should look at it together and try to understand where is the reason for the error.

There were still a couple of testing tasks I needed to do, but didn't have time for them anymore, so they are left for the next week.

Week 3 Evaluation. Discussion and technical background.

Testing is an important stage of every pre-production process. It's extremely important to get through all the code and check that application is responding, that respond is correct and the data is what it is expected to be. That all error messages (if they appear) are clear and give users clear picture of what has happened. That means these error messages need to be controlled (Amman & Offuff 2008, 15).

What I had done during this week was so called "monkey testing": pressing all the buttons, opening every tab and trying to get all the possible data. These tasks though weren't completely useless, as I got a better picture of the application, checked the functionality of the pages I have never opened before. Also I have learnt how to write error reports.

Of course, this kind of testing is very time consuming and not very exciting. Also takes a lot of patience and attention. We don't use any automated tests right now and this just left much more work for humans in the application testing, when automated tests could have helped a lot in this process, as these kinds of tests could catch access error and page render errors.

It is very difficult to establish which is really the best practice for testing. Some say that Test driven development is the only "healthy lifestyle" for coders, others criticize it and suggest to use other ways. Below there is a short review of both points of view.

Test driven development (TDD) is a commonly-known practice when each line of code is written with testing in mind. Kent Beck (2003, 17-24) in his book mentions following rules of test-driven development:

- "Don't write a line of new code unless you first have a failing automated test.
- Eliminate duplication.
- You must design organically, with running code providing feedback between decisions.
- You must write your own tests, since you can't wait twenty times a day for someone else to write a test.

- Your development environment must provide rapid response to small changes.
- Your designs must consist of many highly cohesive, loosely coupled components, just to make testing easy.

According to Beck, Test driven development gives an opportunity to predict the behavior of application and allows to involve customers into development process. The code becomes clear and easy to understand, and, when there are no sudden crashes of the application, new code can be put into production even daily (Beck 2003, 27).

Another author, Scott W. Ambler (2006) describes TDD as a Refactoring plus Test First Development (TFD), because the only thing that separates Test First Development technique, that includes adding a failing test, running it, updating functional code to pass the test and running the tests again, is refactoring, which is good to add in the end of the TFD to remove all the duplications out of test design.

Ambler also describes two levels of TDD:

- Acceptance TDD. At this level a single acceptance test (behavior specification) is written and after that a minimum production code to fulfill that test. The goal of this acceptance test is to provide specification for executable requirements in just in time basis. Another name for these kinds of test would be Behavior Driven Development.
- Second level is Developer TDD. The only difference from the acceptance test is that at this stage a developer test (unit test) is written. And goal for it would be to specify detailed executable design for the solution (Ambler 2006).

Figure 2 is a diagram by Ambler that shows how these two stages work together. His comment about it is following: "Ideally, you'll write a single acceptance test, then to implement the production code required to fulfill that test you'll take a developer TDD approach. This in turn requires you to iterate several times through the write a test, write production code, get it working cycle at the developer TDD level" (Ambler 2006).

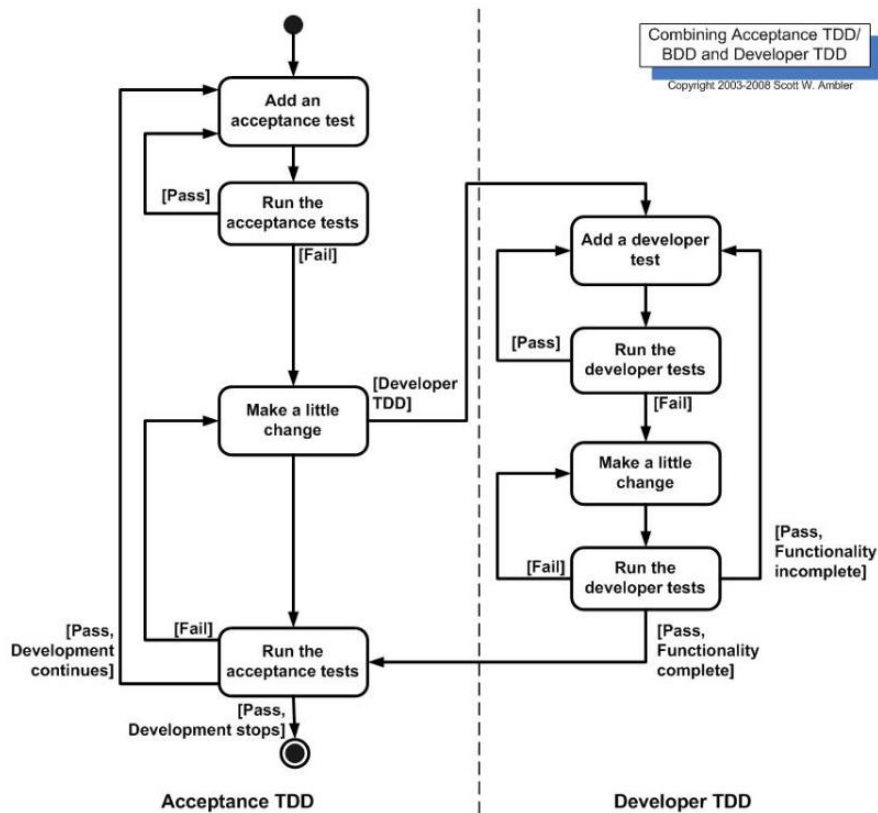


Figure 2. How acceptance TDD and developer TDD work together (Ambler 2006).

A lot of authors talk about benefits of Test Driven Development. Kane Mar (2012) in his 3 part article gives very wide picture about them and raises the discussion why TDD is important in production. He stresses out that: "TDD allows us to break the negative feedback loop and maintain a constant cost of change." If to give a short summary of what facts support TDD it would be following:

- Code becomes less complex and easier to understand
- Units (methods and classes) are more likely to be smaller than with test-last approach
- These above facts lead to the situation that more codes are going to be written and test-coverage of the application is going to be quite high.
- The result is overall more functional and cleaner code.

The opposite opinions that Test Driven Development is not an efficient practice also exist. According to the article by Christos Matskas (2015), there are several issues with TDD:

- False Confidence. TDD adepts sometimes make it look like their method is the only right one to be used, and, according to Matskas (2015), there are a lot developers that try to follow the TDD path but can't do it 100% of the time due to deadlines (and indeed TDD is very time-consuming). Tests that are written after the code aren't less efficient than those, written first.

- Testing too much. It's very easy to get obsessed with tests and developers set the goal to test everything they can. But any line of code takes time to write and understand. And any test is code as well. It takes time to write, so too much code can become very expensive. There also should be a balance between what code does and what it takes to make it.
- Test-induced architecture damage. Some of the rules of TDD are to write as less code as possible, test should be fast and work in isolation. And following this conception blindly may lead to situations that application architecture suffers because application future needs aren't taken into consideration and to make new functions older ones might need to be completely rewritten.
- Heavy mocking. Tests for TDD need to work in isolation. The application code rarely does, because there are models, controllers, services, data access layers etc. To be able to make isolated test developers need to mock those things. And unless there is own mocking framework, the code needs to be written in certain way to be able to provide those mocking.
- TDD in the real world. It's quite an often situation that a developer joins a mature project with existing thousands line of code. And rarely it is written with full TDD. So the real life situation is it's very difficult to apply TDD practices to the existing code, and very often companies don't have thorough testing included in budgets and developer's time. So the most probable solution in this case would be to leave legacy code as it is and start writing new code with tests in mind.
- And last point, mentioned by Matskas (2015) is that not any developer can think in TDD terms. There are people who feel more comfortable from going from more general to specific and it's easier for them to take a bigger problem and solve it in smaller iterations. And these developers would be struggling with TDD and forcing this approach on them would lead to negative circumstances for the whole project.

But even people who find disadvantages in TDD testing still agree that testing is a necessary part of software development process.

Best practices for testing, by Belatrix Web site, are (Belatrix Software 2015):

- "Be flexible, so the team's members can switch tasks dynamically to adapt to changing requirements.
- Create a ranking to prioritize the test cases.
- Maintain traceability between the requirements, test cases and bugs.
- Cover the main business flows".

They also name advantages of automation and manual testing. By their opinion, advantages of automation tests improve the speed, accuracy, and flexibility of the software testing process, enabling your company to find and fix more defects earlier.

Among advantages of manual testing they name the fact, that tester as a person can visualize the bugs in the GUI. Manual testing is quite effective for applications that are under heavy development, manual tests provide fast access to testing information and results,

and are quite easy to understand by all users. But these kinds of test take too long and don't provide complete coverage of the code (Belatrix Software 2015).

Roima doesn't have any automated testing and it's quite a problem in the company. Testing is done manually and indeed takes a long time and consumes a lot of personal resources.

Though automatization of functional tests (end-to-end) is planned in the company, it is going to be very difficult to implement due to the amount of legacy code involved.

3.4 Week 4 (26.09.2016 - 30.09.2016).

26.09.2016.

The plan for today was to continue working on the error in cmd till it is fixed.

I needed more information on the error and program behavior from the colleague that was testing the script.

While waiting I went through the list of available tasks, which was quite large because of all the testing that was on during the previous weeks and picked the task that seemed interesting and related to the part of application I am working on. The task describes the error that if a new user has an empty preset password he gets the dialog that prompts him to change the password, but the change password button is deactivated. The task looks interesting, because I didn't have a chance before to work with the user logging mechanisms in the systems and would have a chance to get to know it better by doing the error correction. I didn't have too much time to research the problem, because I had to continue working on cmd. Plus before taking the task to my list I have to consult the person who created the task anyway.

With my colleague, we went through the script. As I wrote before, apart from a lot of other tasks the script should extract an archive into a specific folder. But apparently last week I understood wrongly the reason of the error. I thought that program doesn't see the archive and worked on that part (see 23.09.2016 entry), but what really happened was the program skipped the whole block with archive completely for some reason. After some more checking like trying to look through the script, checking all necessary files and paths (everything was like it should be), I decided to try to run just a part of the script (by checking only several checkboxes in the GUI). And the script ran just fine. So it became clear for me that there is in the end a problem with arguments passed to the batch file.

The next step was to find the problem. I am not very much familiar with batch files, so had to use different information about them. There is enough good information about batch scripting in the Internet like (https://www.tutorialspoint.com/batch_script/index.htm), so I used that information on the go. First of all I had to figure out how to debug the program while it had no debugger. The answer was found pretty fast: it was enough to put *echo on*. This means that every single line of the code is printed in the console. This is not at all comfortable for production, but useful for debug purposes. What I was interested in was the argument parsing part, so I put some extra *pause* lines in the code to stop the program after the argument parsing. After running the script from hta file I saw that after one point all the arguments were messed up. It was the first clear error in the code. It appeared so that if there is an error in the parameter or argument the value was reset to default and the *shift /* which was inside an *else* statement never triggered and the same argument was passed to the next parameter and messed them up (because they all were activated so there was no place for parsing the parameter twice). My first thought was to do the shifting outside else, I've changed the code, tested it and everything went smoothly. I was already in a hurry to leave, so I haven't tested thoroughly (which I of course should have) and assumed that the code is fixed and replaced old version with my solution.

27.09.2016.

Plan for the day was to continue to work on the login error.

I managed to went through a debugger a couple of times and check all the functions which are involved into the login process. But then my colleague told me that he tried to use the batch file I fixed yesterday and it didn't work: my fix doesn't work in the end. I had to return to the batch file problem. After a quick look at it I realized that thorough testing was indeed necessary, because when I tested the program so that one of the parameters was missing, the value of the seventh argument became an empty string and no parameters after that one got a value. It happened because my fix always shifted the argument, even though the next value still had to be compared to the argument. It became clear I had to start all over again. I analyzed the original code: basically, if all of the parameters and arguments are right, the program should run smoothly, but if any of them is corrupt for any reason, there would be problems with the code again.

I decided to write a piece of code that would go through all the arguments and assign each of them to the necessary value. I went to look for the answer how to implement it in the Internet. I have found a couple of examples of a loop code for the batch file, one of

them I found quite interesting and decided to try it out. This is the example of the code from stack overflow rewritten for the 6 static arguments (Stackoverflow 2010):

```
@echo off
setlocal enableDelayedExpansion
set "options=-username:"opt1" -option2:"opt2" -option3:"opt3" opt4:"opt4" -option5:"opt5" -option6:"opt6"-flag7: -flag8:"
for %%O in (%options%) do for /f "tokens=1,* delims=" %%A in ("%%O") do set "%%A=%%~B"
:loop
if not "%~7"==" " (
    set "test=!options:*%~7=! "
    if "!test!"=="!options! " (
        echo Error: Invalid option %~7
    ) else if "!test:~0,1,2,3,4,5!"==" " (
        set "%~7=6"
    ) else (
        set "%~7=%~8"
        shift /7
    ) shift /7
    goto :loop
)
set -
```

But this code didn't work. I tried a couple of more simple versions but the still wasn't anything I was looking for because I didn't want to rewrite the whole file. So in the end I have created my own question there, where I wrote an example of original code. That how it looks like:

```
REM arg7 parsing
set ARG7=%7
if not "%ARG7%" == "ARG7" (
    set ARG7=
) else (
    shift /7
)
REM arg8 parsing
set ARG8=%7
if not "%ARG8%" == "ARG8" (
    set ARG8=
) else (
    shift /7
```

```

)
REM arg9 parsing set ARG9=%7
if not "%ARG9%" == "ARG9" (
set ARG9=
) else (
shift /7
)

```

It is obvious this kind of code needs refactoring, because there are same patterns used all over.

I got an advice how to refactor this kind of code in a batch file quite fast in the end, but couldn't use it directly. Had to combine it with the loop structure to make it working. In the end the code looked like the following:

```

:loop
for %%I in (
ARG7 ARG8 ARG9
) do if "%7" equ "%%I" (set "%%I=%7" & shift /7 & goto: loop)
If "%7" == "" (goto: endloop
:endloop

```

Of course, I had to write down all of the 12 arguments left, and there was still a possibility for an error among them, but in any case all arguments will be processed and a parameter automatically will be created for each value with the exact same name. And another issue getting solved is generalizing the code itself. It takes much less space and there are no similar fragments anymore.

Though my supervisor decided we should use the original code in the end, because it was more obvious and easier to read. He might be right, because values and parameters shouldn't be changed anymore and as soon as everything is correct, the original code should run smoothly.

28.09.2016.

Plan for the day was so to go back to working on login error. I had to research the way a new user is created and added to the database. And how the user is given rights.

To do that I contacted people who are responsible for this process and got additional information from them. They explained me the process details and after that I had to try to create a new user myself. After I got a new test user I needed to establish how to set the password to be changed on the first log in. For that I again had to ask my colleagues and research how this kind of functionality is set. I won't describe the exact algorithm, but there's a special field that is set to "set password" value. And in that case the user get "change password" dialog as soon as username is put into its field.

After that I wanted to check one hypothesis, that error appears because of a wrong empty string recognition. That the empty password should be interpreted as an empty string and for some reason the value could be returned as null. For that I set breakpoints in the controllers and ran the code. The value was returned as an empty string. So I decided to test what happens if it is a "null" value. So I just replaced the variable that should be returned by the app with null and ran the code again. What was important is that I got was an error of invalid username/password. That meant that even if the value returned as null for some reason it is still processed as a password option and not accepted as soon as it doesn't match an empty string. This way the hypothesis was proven wrong.

After that we had a discussion with my colleagues about the case. We came into conclusion that as soon as this error couldn't be reproduced with exact same result (just nothing happens: no error, no other reaction to the password string), this could be some isolated case and we should put the task on hold till the scenario repeats itself in some other case. For the rest of the day I have researched browser related errors. I haven't started any particular task, just checked the list of the errors and picked up several I could start the next day.

29.09.2016.

The task I pick up for this day was to fix a problem with angular script. The script related to a counter that displayed a number of elements in a list. For the general list it was correct, but for the list of favorites the number was displayed wrongly.

The part of the project, the issue was related to, was completely new to me. So, I have started with exploring controllers and views that were new to me. This part of the project is built in another way than the one I used to work with during all the past time. The view was rendered using mostly angular directives, not regular controllers and this part was difficult for me to understand, because I haven't had a lot of experience with angular directives. It took me half a day to figure out which directive's function is responsible for what,

and only with a help of my colleague, who was working mostly on this part, I have found the function that was responsible for the counter.

After that I passed to debugging. I have run the application a several time and noticed, that there was a function that was run twice after page load. The function apart from other things also set the counter for favorites. I checked the function for favorites controller and followed the number of items in the list during the debugging. The debugging showed that the favorites were counted twice and that was the reason for the wrong number. The only thing I had to do was to reset the counter before favorites are counted all over. This very small correction fixed the issue. I tested the application several times to make sure everything worked and number of favorites is set correctly. Also, sent the task to be tested to my colleagues working list.

30.09.2016.

In the morning I set myself another task from the work list. The problem I picked up for fixing was the following: there was a window for assigning a worker to the task. If the worker field was empty and an "Accept" button was pressed, there was an SQL-typed error showing. And as soon as there shouldn't be any database related information showing anywhere, this issue needed to be fixed.

I started, as usual, with a discussion with my colleague about the issue. We discussed that there are two correction options for this issue. The first was to replace the error message with something more user friendly like "No worker is selected". Another one was to deactivate "Accept" button, if the selection field was empty. We agreed that the second one is more logical for this.

I started to analyze the code to figure out how to disable the button when the field is empty. The task, that appeared to be easy at first, started to look much more complicated: first of all, all the functionality was based in different directives. Second, disabling or enabling the button was already tightened into another functionality where several Use Cases possible, and depending on that there was a separate check for a pop/up error message. It took me a while to figure out which controller is responsible for which functionality, and I carefully went through with algorithm of options for error pop-up messages. There were three option and each of them I have to take into consideration. One of them were if Saving is allowed without user being logged in, another is logged in is required and last one no error message.

The research took a significant amount of time today, so all the corrections are to be left for the next week.

Week 4 Evaluation. Discussion and technical background.

Writing hta application with a batch file support is completely different experience, than working with some more complicated code. Batch scripting uses very different syntaxes and it takes a while to get used to it and start understanding some aspects properly. For example, it is very important to follow line order and get spacing right, especially with if-else statements. Writing loops in batch scripts also differs from writing loops in other more advanced programmer languages.

Hta applications in its turn have own syntaxes as well. They are built using visual basic and can support graphical interface. Batch scripts can be easily called from inside a hta application and the launch of the script can be bound to a button, for example.

The reason Roima is using this kind of applications lies in the simplicity and small size of batch files and at the same time its power to make changes at the system level (install programs and do register changes). Also both tools can be launched from any Windows-system computer, which is important for the company, because these applications are going to be used on client computers, where there is no possibility to install additional software.

Let's have a look on theoretical aspect of batch files and hta applications.

"Batch files or scripts are small easy-to-write text files that carry out a series of commands. They can be simple enough that even the average home computer user can take advantage of them" (Laurie 2012). According to Laurie (2012), batch file is a powerful tool for carrying routine tasks.

The batch file itself is a file with code, these commands get executed in a sequence one by one. These files can have extensions as BAT or CMD. Command interpreter is a system file that provides interface for executing these files. In later version of windows it is a tool, known to all system administrators and just advanced users named command prompt (cmd.exe) (Laurie 2012).

The coding environment can be any text editor, even as basic as Notepad.

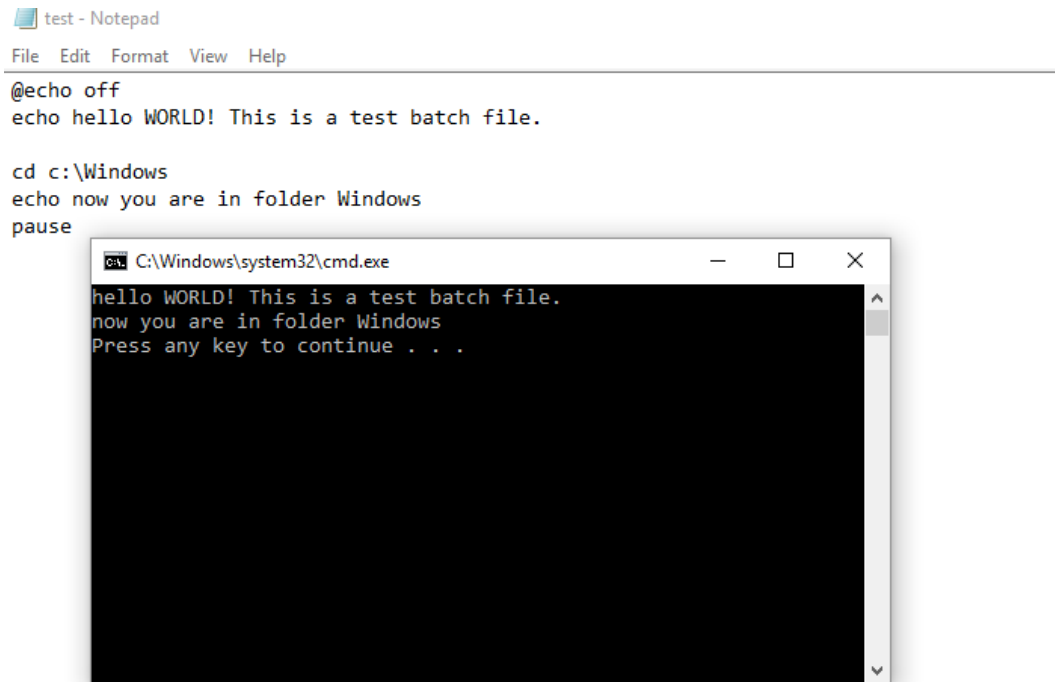


Figure 3. Example of batch code written in Notepad and command interpreter.

Another more modern tool with much more possibilities would be Windows PowerShell. According to Microsoft's own Web Site, "Windows PowerShell is a task-based command-line shell and scripting language designed especially for system administration". The tool is made using .NET framework, it has rich expression parse and uses own fully developed scripting language (Microsoft 2104).

This tool has all the features batch files can do, but it goes further and gives an opportunity to customize tools and utilities. It also gives an opportunity to build own graphical user interfaces which allow data parsing without any additional applications and methods.

I'm not going to write detailed description of PowerShell, but it would be very worthy replacement of batch file and hta application, because everything can be done just in PowerShell. Why Roima doesn't want to use PowerShell though?

Steve Jansen describes batch files as a high benefit tool with low cost. And he points out, that batch files are portable and have low friction unlike PowerShell script files. Also Windows own command prompt is very stable and there is no need to worry about PowerShell version compatibilities with the server and it's interpreter path (Jansen, 2013).

And, as mentioned above, the application is planned to be used on customer computers without any possibility of installing anything else. It's very crucial that the application runs every time on every computer. That is the main reason for the company to use, some might say, primitive batch files to take care of installation processes.

This week's tasks gave me a good opportunity to get to know batch coding deeper and try some complicated things with the code. I have learnt how to make more complicated loops and read about argument parsing.

3.5 Week 5 (3.10.2016-7.10.2016).

3.10.2016.

Today's task is to continue working on the same error.

Luckily all the research has been done during the previous week. So I just went through the debugger once to refresh in my memory all the details and began to correct errors.

I started with make the button inactive in case when user isn't logged in. The user's log in state check was added to the directive's activate function. The check returned true-false value and was assigned directly to the button state.

The next stage was to take into consideration the case, when saving was allowed even without user being logged in. That meant I had to change the check further and if-else statement there. I used shorthand syntax for the statement to make the code more compact:

```
var y = (x == true ? true : false);
```

I've learn this kind of syntax a while before and try to use it now whenever possible, because it turns four or even five line of code into a single row. So I just added another check as an expression to be evaluated as true or false as well.

The same expression is very handy, because at the same time it handled the third case automatically: if the user isn't logged in and if save is allowed without user being logged in, then activate the button.

Though I decided to change the order of checks: it was more logical first to check if log in was necessary and only then check if the user is logged in or not.

Next I needed to send the button's value to the directive. I haven't dealt with directives a lot before, so had to read more about them and try different tricks. In the end I have discovered that values could be passed to controllers if the "bind to controller" parameter has "=" pipe in it. Using this, I have successfully passed the value inside the directive and the button began to work like it should.

I tested that my changes work and had to ask my supervisor about what I should do next. He said that there was a request from the customer to make some changes and I can handle them. He sent me a list by email. I read the requests and wrote tasks based on them, put them on my worklist.

Tasks were: to save working state between session, to make some changes into the way data was handled into several places (add more restrictions), to add new view to the page.

4.10.2016.

Plan for today is start working on adding the element Id to localStorage.

I decided to start with the task I can handle myself, without any help and advices. This task was saving state through sessions. Local storage is the best way to handle it.

So as always, I started the task with research of what functions are involved with element Id. And run the debugger. There was pretty much just one function involved, and because of it the state should have been saved: it was an item select function that further got elements from the database with this id as a restriction.

So the goal of the task became clear right away: on element select save the element id into local storage, and on page load get it from the local storage.

This part was easy to do, and I implemented it very fast. We discussed the situation with my supervisor, and he explained there was another situation that I haven't taken into consideration: when the user wants to open the page with a link. That means that the value should not be saved just in local storage, it should also go to URL and be read from there if it's present.

To implement this feature, I had to research about location features and how to work with them.

When I had a picture of how location services work, I got to think about the situations where URL set up needs to be used. There was also a question of which services to use, as there are two options: to set the itemId in the route or to set it as a search parameter. This question I had to resolve with my colleague, who recommended to use it a search parameter for now.

There again was only one situation when the URL needed to be set - on item select. And checks, same as localStorage checks, were needed on page's load, but logically this check should be before anything else. In other words, if user enters the page using the link, the link should be primary, and only if there is nothing in the URL, then localStorage should come into play. If both are empty, page loads with default settings.

And at this point the fact that URL and local storage should set each other also came into the picture: if URL has value, set it local storage and vice versa.

5.10.2016.

Today's goal is to research SQL tasks. Also, I got a small testing task to my list. I set myself a plan to get more details about SQL tasks and learn more about the structure of company's own application called AppBar.

As soon as testing wasn't supposed to be very long, I started the day with it. Opened the task's description, read what was done, opened the related part of the application and tested that the mentioned changes really work like they supposed to. It was important to test that each browser responds to the changes well and there are no browser-related errors. Everything worked fine, so I left the comment that changes are tested and everything works fine and marked it as "done" in the source control.

After I continued customer specific tasks. And first task, that involved SQL, was to add a restriction to the data set. The restriction was fairly easy: the query had several where statements and I just needed there "AND VALUE_X IS NOT NULL". The trick was to get it into the right place.

All the changes to SQL are done in the AppBar application. This application is used for defining views by building SQL queries from different set ups. The problem was that I had no idea how to work with the tool as it has hundreds of different set up possibilities, as it requires to give definitions for any database table we have (and there are a lot).

I had to ask my supervisor for help. He showed me the place where to put the query restriction and after we discussed the remaining tasks. All of them required changes through the AppBar. He told me to place these tasks on hold and that we are going to set a training session for this tool in a few days. For start I had to contact the person who should lead the training and get other team members' timetables so everyone who might need the same training session would be able to participate.

I did that and set the training to be on next Tuesday.

6.10.2016.

As soon as I had nothing to do, I asked my supervisor for some tasks. He sent me a link to two tasks. One task was set already in spring and no one had time to do that. The point was that hour input field in the calendar didn't have any maximum input restriction. My job was to first clear out if this was an error at all. Another error was the fact that page didn't refresh on company change in settings.

I started by writing an email, trying to get more details about hour input issue. While waiting for the answer, went to research the related code, as always. In a while got an answer that this situation wasn't an error at all and there is no restriction because some users want to input all weeks' hours into one day instead of writing them down every day. In my opinion it was a bit weird, because companies use the hour input to maintain employee's work hours, so I would guess they would want to know which day people work and how many hours are spent a day (in case there are any overtime which needs to be paid in another rate).

But as soon the official response was that it wasn't error but a feature, I just wrote a comment to the task and closed it.

Next task was make the page refresh on company changes. As always, I started with code research and debugging. And indeed, when the company was changed in the settings, nothing happened on the page and all the data shown was still related to the company that was set originally. Even when the data refreshed in the view, it still didn't take into consideration the new company. The rest of the day was busy with careful research.

7.10.2016

Plan for the day is to continue working on the same errors.

Yesterday's research showed there was no event handler in the controller. After more research, it became clear, that there was a directive that should handle the changes in settings. And I returned to the situation when I had to go deeper into directive-controller communication. This wasn't just a value I needed to pass to the controller, but a whole event. I started to go through the code trying to find examples and found one: it was java script's *element.onChange* function I needed to use.

The next issue was how to pass the event into the controller? The answer came fast: I didn't need to work with even, I needed a value to pass. A Boolean value *company-Changed* was quite enough to make the controller listen to the value (I set the directive to pass the value into controller). After making these changes, the controller started to get data after company's change.

After everything was set, I tested that everything worked, wrote the completion steps' description in the source control and put the task for testing to my colleague.

Week 5 Evaluation. Discussion and technical background.

During this week I have learnt more compact ways to write if-else statements which allow their usage as one line-expressions. I read and understood the conception of location services that allow passing values to URL. But the main aspect of this week was usage of Angular directives and their communication with controllers which was also the main problem to be solved.

Below is some information on directives that was useful for me.

Ng-book's description of directive is following: "The simplest way to think about a directive is that it is simply a function that we run on a particular DOM element. The directive function can provide extra functionality on the element." Jesús Rodríguez Rodríguez (2015, 31) in his book calls directive "the star feature of Angular".

```

1  angular.module('app').directive('myDirective', function() {
2    return {
3      template: '<div>{{message}}</div>',
4      link: function(scope, element, attrs) {
5        scope.message = 'Hello, World';
6      }
7    };
8  });

```

Figure 4. Example of a simple directive (Rodríguez 2015, 34).

Directives have different options: template, scope, link, controller, transclusion. These options are a part of a directive definition object.

Directives are very flexible: it is possible to write any kind of directive to do pretty much anything the developer would want to. Because of that it is the main feature of Angular and exactly directives make Angular very powerful development tool.

The directive shown in the figure 4 can be used on any element with a name “my-directive”. The naming convention is important because it uses camel case in the name of the directive and snake-case on a html element.

Directives can also be used as templates and then become a separate html element. To do that, a declaration of a template is needed, like it is shown in figure 5.

```

1  angular.module('app').directive('jumbotron', function() {
2    return {
3      template: '<div class="jumbotron">' +
4                '  <h1>{{header}}</h1>' +
5                '  <p>{{message}}</p>' +
6                '</div>'
7    };
8  });

```

Figure 5. Directive with a template (Rodríguez 2015, 36).

It is also possible to move the template into separate file and use a link to it like this: *templateUrl: 'jumbotron.html'*.

In other words, with template (usage in a directive or as a URL) it is possible to abstract a piece of DOM and insert it anywhere (Rodríguez 2015, 45).

But how to make directive communicate with parent controller? If there is any change happened in the directive, we would like the parent controller to know about it and probably react on that change.

For that we can use function bindings.

```
app.directive('someDirective', function () {
  return {
    scope: {
      oneWay: '@',
      twoWay: '=',
      expr: '&'
    }
  };
});
```

Figure 6. Directive example with function binding (Precht 2015).

“=” mark two ways binding: the controller can pass the value to directive and get it from the directive It is called inherited property. @ binding means, that some value can be only passed to the directive and & means the function passing (Precht 2015).

Jeremy Zerr (2014) names some of the best practices when using AngularJS directives:

- To remember about naming convention
- Use own html elements and directives as attributes if concerned about HTML validation. Using directive as an element might cause trouble with validations.
- Use prefix “data-” which is allowed within HTML and any attributes with this prefix are ignored and allowed to be custom attributes.
- AngularJS suggests to use an element when making a component that is in control of the template.
- When using isolated scope in the directive, variables inside directives should be accessed using attributes passed to the template.

And in the end here is some of the AngularJS Directive Best Practice Guidelines:

1. “Use your directive as an element name instead of attribute when you are in control of the template”.
2. “Use your directive as an attribute instead of element name when you are adding functionality to an existing element”.
3. “If you do use a directive as an element, add a prefix to all elements to avoid naming conflicts with future HTML5 and possible integrations with other libraries”.
4. “If HTML5 validation is a requirement, you’ll be forced to use all directives as attributes with a prefix of “data-“ ”.

5. "If XHTML5 validation is a requirement, same rules as HTML5 validation except need to add "=" and a value onto the end of attributes".
6. "Use isolate scope where possible, but do not feel defeated if you can't isolate the scope because of the need to two-way data-bind to an outside scope" (Zerr 2014, 86).

I could still say that using controllers feel easier for me, because I got used to working with controller structure, and there is no need for variable passing back and forth. But indeed, directives is a powerful tool, because they can set the DOM directly and element usage allows to create any behavior in the view needed. So I just need to get more familiar with directives and start using them in code I write myself for the front end.

3.6 Week 6 (10.10.2016-14.10.2016).

10.10.2016.

Didn't have any specific task for the day, also as soon as I had SQL tasks pending while waiting for the training, I couldn't start anything big and time-consuming.

I took a styles correction task where some small changes needed to be applied (like changing some colors and element transparency).

For the rest of the day read articles about Oracle SQL.

11.10.2016.

Today we had a training about AppBar, where my teammates and me got shown how to work with the tool, what were the main features, where all the definitions for one view could be done (where we define what table we need, what columns we need to get, what joins are there going to be and where restrictions are set). Also we went through some extra definitions that needed to be applied when making changes to the application structure.

All this information was extremely useful for me, because connected directly to the tasks I needed to perform.

After the training was over, I spent the rest of the day researching the tool on my own and planning how I am going to apply the changes.

12.10.2016.

SQL-related tasks are on the ToDo list today. As was mentioned during the training, all SQL changes first needed to be tested in OracleSQL developer and only after the query is tested and fine, it is allowed to put the changes into AppBar. Because a query with an error will prevent other developers working on the application part that uses that data to do their work.

So following the rule, first I needed to get a ready query and make changes to it in the sql developer.

First, I thought that it is possible to see the whole query in the AppBar. There was a final query available and I try to run it in the developer, but got an error. The application was working, so there was something weird going on.

To save some time I asked my colleague what is happening. Got an answer, that the query from AppBar still get processed during the runtime, and to get the right query, I needed to catch it in the debugger.

I haven't checked any deep database connection and data process classes in our application, so apparently, now was the time to do that.

To save some time I've asked my colleague where should I look at. He sent me the exact function which return the SQL query it gets from the AppBar after the process. After running the debugger and checking the returned string every time the breakpoint got hit, I finally got the string with a right query.

I copied the string into the SQL developer. The changes I needed to make, was to add a time restriction (to show data where date in Date column is no more than two weeks after current date).

To fulfill this task, I had to refresh my memory about SQL queries and look how to get current data and then read what kind of operation could be done with it. The decision wasn't very difficult: get the current date, extract date from it and add 14 and compare that the data's day in the date is less than the expression.

I wrote the task, ran it, and got the right kind of data. We have a rule not to change anything in the end of the working day, so left the actual changes to AppBar till morning.

13.10.2016.

Plan for today is to add the new restriction to AppBar, test that it works and to make first changes in the customer's project (save data to URL and local storage).

I have added the new restriction to the AppBar and made sure the data was still correct and that application overall works. To do that I have started the project, checked the application without the debugger first (everything seemed working) and to make sure the query contained my changes, ran the debugger, and made sure the returned string had the restriction.

After that I have downloaded customer project's files, and tried to launch the project. Got a connection string error.

I made sure that the string was correct, that I have all of the files, that IIS had new project in settings: nothing helped. I even tried to solve the problem with my colleagues' help, but they also confirmed that everything seemed in order.

I tried to research the situation myself for a while, but nothing helped. In the end, I used our local databases' connection string. And the project launched. That meant I had wrong data, but at least project files were the same. I didn't need to work with data, at least for now, just to change project files and test it, so decided this would be a good enough temporary solution.

I pretty much copied the changes I have made to our own project files to the customer's project. There were no conflicts, so the saving to localStorage began to work there.

I had noticed that the customer's project had older file versions than ours. That meant I needed to be careful when making changes to the customer's project, because some functions were missing and others worked in a bit different way.

14.10.2016.

Today's task was to do some more manipulations with AppBar and to add a new column to the view. As the current version (version under development) didn't need such a change, the new column was to be delivered only to the customer's project.

So, I opened the AppBar with a customer database connection. Then From the customer's project I have found the string with SQL query that returned the view and copied it

to the SQL developer. What I needed was to make a data return to one column and the data needed to be a list of items related to the table. That meant I needed a Join statement and I needed to make it a list somehow.

The search in the Internet gave a solution to list issue: LISTAGG was the function I needed to use in the SQL query. I tried several options before I got the right one by using an example from Database SQL Reference Web Site (2016).

So now I had a working query and needed to add a new column. First, a new column needed to be added in AppBar in the column section of the View. The SQL restriction also is defined there. I added the column and query and only had time to test that nothing is broken and the application launching.

Week 6 evaluation. Discussion and technical background.

This week was full of new skills and information: I learnt about our own tool we use in the development process, I got to really work with Oracle SQL Developer for the first time, and Oracle SQL differs from MySQL, which I'm familiar with. So writing queries in Oracle SQL takes more time for me than it would with MySQL. I also got familiar with new function LISTAGG.

When I was looking for the answer, why Oracle instead of MySQL, I have found a nice comparison table at Rapid Programming web site. The full table can be seen in the Appendix 1. It is just clear from the comparison, that Oracle fits much better for the enterprise use due to better security (many backup mechanisms compared to only two in MySQL, uses enhanced database security with many step authentication), it generally has wider number of features and functions, so all in all generally more flexible. And Roima is a large company with many customers, and of course Oracle is a much better fit for as for an enterprise.

The SQL language used in Oracle is PL/SQL which is a procedural language extension to SQL and Steven Feuerstein (2001) offers several guidelines when working with PL/SQL:

- “Never repeat a SQL statement
- Encapsulate all SQL statements behind a procedural interface (usually a package)
- Write your code assuming that the underlying data structure will change
- Take advantage of PL/SQL-specific enhancements for SQL”.

I got to write just simple data accessing queries. And first of them was getting Date type data.

In PL/SQL language there are several types of datetime data: DATE, TIMESTAMP, TIMESTAMP WITH TIMEZONE, and TIMESTAMP WITH LOCAL TIMEZONE. Also, there are interval datatypes: INTERVAL YEAR TO MONTH and INTERVAL DAY TO SECOND (Feuerstein & Pribyl & Dawes 2004, 27-34). I used just a simple DATE datatype in my query.

Now about LISTAGG function. Its syntax is shown in Figure 7.

Syntax

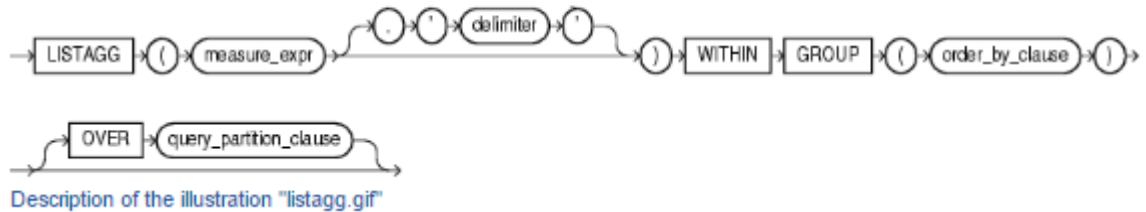


Figure 7. Syntax of LISTAGG (Oracle Help Center 2016).

Its purpose is to order data “within each group specified in the ORDER BY clause” and then combine the received data in a measure column.

Features of this function are:

- It operates in all rows and returns just a single one
- Returns an output row for each group specified by GROUP BY clause.
- It divides the query result set into groups based in one or more expression in the query_partition_clause.

And there are certain rules applied to the function:

- The *measure_expr* can be any expression. Nulls are ignored.
- The *delimiter_expr* designates the string that is to partition the measure values. The default for this clause is NULL, which means it is optional.
- The *order_by_clause* defines the order in which the data is returned and then concatenated. Column list should achieve unique ordering for the function to become deterministic.

The data type returned is RAW if the destination column is RAW, otherwise the value of type VARCHAR2 is returned (Oracle Help Center 2016). Figure 8 shows an example a query with LISTAGG function and data returned by it.


```

SELECT LISTAGG(last_name, '; ')
       WITHIN GROUP (ORDER BY hire_date, last_name) "Emp_list",
       MIN(hire_date) "Earliest"
FROM employees
WHERE department_id = 30;

```

Emp_list	Earliest
Raphaely; Khoo; Tobias; Baida; Himuro; Colmenares	07-DEC-02

Figure 8. Example of a LISTAGG function (Oracle Help Center 2016).

3.7 Week 7 (17.10.2016 - 21.10.2016).

17.10.2016.

Today's task to test, that SQL working and data for the new column is coming as it should. And to do that I need to add the column to the table in the view (on html level).

In the morning I have noticed, that there was an error in the query. So, had to do some changes to it and test in in the developer thoroughly. When the data was correct, added the changes to the AppBar as well.

After that I made changes at the html level: added new column, applied same styles for it as for other columns in the view. Then I needed to add the name to the column. I checked if there is a string constant for the required name already, there was none. That meant I needed to add a new one using the AppBar. Then I tested, that changes looked good on the page and everything is fine with string constraint and translations.

Now I needed to connect data to the view. First, the model needed to be changed. When I started editing, I wanted to add a list, but then realized I didn't need anything complicated and a string would be just enough.

After adding a new variable to the model, I changed the controller, so it connected data for new column with correct variable. No more changes on the server side were needed and neither in JavaScript controllers, because all data for the view was gotten from the list of elements and put into the table with help of *ng-repeat*.

I made all the necessary changes and run the project. The view showed all the data needed. To be sure, I checked the string with SQL query, but it wasn't necessary as soon as data was showing just fine.

This task was the last one for the current view. Next involved building a new view.

18.10.2016.

Today I needed to start a new tab for one of the portals. In plans is to have a small meeting with supervisor and get clear instructions of what to do. But generally, strategy for his task would be to first get a working SQL query, then set the view configuration in the AppBar using as basis the working query, after that make changes to the project's code: new model, controller, new html page with a table, a JavaScript controller for it, and all the changes to common controllers that connect server side with client side.

The table I needed to show had some basic columns from one table and more complicated column that should show some data about related tables. After talking to my supervisor about the view and the task in general, I got the good picture of what how the view in the tab should look like, it's functionality: basically, I also needed to add the functionality to select the row and open a related table on-select.

For now, it was decided to start with the basic view: to show all data related to one table, and after I have that working, then to add the last column. The functionality for row selection was left till stage two.

I started to work on SQL. I copied the existing SQL for another tab on the same page, because they should have been very similar, and started editing the query so that it showed the required data for my view.

As soon as I copied the new view from existing one, apparently, there were some configurations left, that messed the SQL, because I got a message, that the query has errors. Luckily the new view wasn't in use and couldn't break the application. So, I left this error for tomorrow.

19.10.2016.

Plan for today is to continue working on the view, but before that needed to make sure SQL is working again.

I copied the SQL string to the SQL-developer and saw the error right away: it tried to access a table that I didn't need, and in the join, there was a value from the view which I copied. That meant there was some configuration I haven't changed.

In the end I have found the configuration in question and removed the join from there. Got the SQL working.

When I had the string with some basic data working in SQL developer, I made necessary changes in the AppBar: added table columns there. After that I needed to add permissions and assign roles. Which I did in the AppBar.

The rest of the day I was planning the changes I am about to make in the code: checked how another view was handled, what data-context controllers I needed to change to set up the data flow from server to client side.

20.10.2016. In the morning I have noticed that changes I made to the columns didn't update in the SQL query for the view.

Took me all day to find the reason for this and fix the table. Now everything was pretty much ready with the data.

21.10.2016. Today I started writing the code in the project. As soon as this changes would go both to the client and daily version first the changes to be made in daily version and after that taken to the client.

I started with building a model for my view, then wrote a controller and made planned changes to the data-context controllers (wrote some dummy code and commented it out). Then I copied another tab's java script controller, renamed there all the values so it fitted my view and made dummy empty cshtml page with "Hello World" on it.

To make the new tab showing on the page I needed to made changes to the routing and web.config files. But that was left for next week.

Week 7 Evaluation. Discussion and technical background.

During this week, I got to work with PL/SQL a lot. I understood, I learned more about configurations in AppBar. Even though it doesn't sound like a lot, I really got overwhelmed with new information during this week. I wrote a lot of things down and had to go back to them several times. Apart from these subjects, I also got to make a new tab and make changes to routing which I never did before. This helped me to understand the structure of our application and some principles of its working better.

This week's main subject was SQL. I think if I knew SQL principles better, I would have managed with these week's tasks much faster and without any big issues. So, I would like to concentrate on best practices for SQL queries.

Database operations quite often become a bottleneck for a large amount of web applications. Developers can and need to do their part by optimizing queries and do generally better code.

One of the most important method for optimization is enabling query cache which means that when same query is executed several times, the result is accessed from cache and not from the database, which significantly improves the performance and data access speed. Though non-deterministic functions like CURDATE, NOW and RAND prevent usage of cache. The cache is handled in the application code (Guzel 2009).

Guzel (2009) also offers to use LIMIT 1, when searching for unique rows, especially, when the query is supposed to get the value which shows if something exists in the dataset. In such cases, it would be enough to stop query execution when a match is found.

Many other authors, for example Guzel (2009), Feuerstein (2001), McCown (2010), also recommend to use indexes. Indexes can greatly increase the performance of an SQL query by allowing quicker access in frequently used queries and situations. According to Guzel (2009), some searches won't benefit from indexes, for example, precise searches with restriction "LIKE '%STREET%'",. Also, it's important to remember that when using JOINS it is important, that all the tables under JOIN has index and are of the same type, otherwise the program would be unable to use at least one of the indexes and the performance would suffer.

According to Feuerstein (2001, 67) it is a good practice to avoid ORDER BY statements whenever possible, because it takes much more type for the SQL runner to process the data and then sort it than just to return unsorted data. Also when ORDER BY statement is used in a subquery which supposed to return just partial information, Oracle would still need to get all the data to be able to sort the one required.

Another advice given by McCown(2010) would be very useful for me: "don't blindly reuse code". Of course, nothing can be easier than copying an existing piece of code that doesn't exactly what you need. The usual problem is that it quite often does even more things which most of developers don't bother to trim and the result is supersets of unwanted data and bad performance.

Another best practice advice from all the authors already mentioned is to avoid SELECT * queries. Like it is shown in Figure 9. It has the same problem as the issue mentioned above: the amount of data gotten exceeds the data needed.

```
01 // not preferred
02 $r = mysql_query("SELECT * FROM user WHERE user_id = 1");
03 $d = mysql_fetch_assoc($r);
04 echo "Welcome {$d['username']}";
05
06 // better:
07 $r = mysql_query("SELECT username FROM user WHERE user_id = 1");
08 $d = mysql_fetch_assoc($r);
09 echo "Welcome {$d['username']}";
10
11 // the differences are more significant with bigger result sets
```

Figure 9. Example of Select queries used in PHP code (Guzel 2009).

Based on the techniques mentioned above, I took the most useful advice for myself: do not get more data than you need. It is much better to start with minimum query and increase it as it becomes necessary. After this research, I understood, that I do the copy-paste mistake quite often, and it is very likely that I end up with extra data that I won't use. And this happens not only with SQL code, but also in the application development.

If talking about Roima's practices, I could say that we use indexing for database tables and senior developers usually follow the techniques mentioned above. There is one that could be useful in the future for me: to try LIMIT 1 statement.

3.8 Week 8 (24.10.2016 - 28.10.2016).

24.10.2016.

The task for today is to continue working on a new tab.

Tried to test routing, got "Unable to start debugging. The operation has timed out" error. Restarted IIS. Debugging started, new error "This type of page is not served." Had to reload the project, but after that Visual Studio just got frozen. After restart started getting Parser Error.

While trying to solve the reason of the error, noticed a lot of new files, that weren't included in the project. When included them, got a lot of errors that came from these extra files. Apparently the application started to behave weirdly and instead of pure ASP.NET project tried to behave like MVC.Net which it wasn't. Took me a while to solve the issue,

but deleting extra files and restarting the IIS once again, solved the issue. So now I could go back to testing the routing configuration of the new tab.

I've added the route to the route config file and new rule to web.config. And got "HTTP Error 500.52 - URL Rewrite Module Error." The reason was the rule name I gave for a new tab was already present in the rules' list. After the name correction, application began to launch, but there were problems with the content of a new page. So my goal number one is to get "Hello World" showing up in the <div> on this page.

When I have tried to open my page with a link, I just got the 404 error (file not found). So apparently, the reason was in the routing code. My rule had redirect action type, as I wrote it by an example of the rules that were in the end of the code. But apparently, I didn't need redirect, because the global address was still the same, I just needed another tab. And yes, I found the rules for tabs earlier in the list: the action type was "rewrite". I have changed my code to that and the page began to load. Now when the routing setup is in place I can start working on the view itself.

The part of the portal is made based on directives. It allows to control parts of the code by using encapsulated directives. So I started to build the code based on the examples of other tabs.

For that I used *[HttpGet]* *[Authorize]* tags for C# controllers and put a receiving code to the general controller. That piece of code is supposed to redirect data to tab's own controller. Also I wrote some html code to defined the layout for the data.

25.10.2016 - 26.10.2016.

Plan for these days was to continue working on the view and get the data working.

Today I worked on the code for tab controller and assigned it to the html page. For me front-end is the most difficult part of writing an application, especially when there is something I'm not familiar with involved. Directives still look very tricky for me, so I'm glad I got to work on one, but at the same time it took me a lot of time to understand the principles of the very similar directive for another tab. All in all, I needed to find four different pieces of code: directive, its controller, the tab page itself, controller for it and a page with a section for a table which is going to be populated with data from the context controller and C# code itself.

During the first day I was copying the code of another tab and changed it according to the needs of the new tab. The working principle of a new tab is going to be very similar to the existing one, yet even more simple. For example, there would be no selection of table rows and no manipulating with data, at least not just yet. So I had to take a lot of unnecessary functions away and be very careful while changing other ones. While doing that I managed to mess up the existing tabs code, because, I guess, I haven't removed some existing reference and the existing tab tried to access the data of a new tab. It was easier to get the old code from the server and start copying my changes piece by piece than trying to catch the error in the debugger. I haven't realized where the exact error was, but at least the old tab started working again.

The next day I continued with working on modifying my code for the tab. I got the right functions finally, but while testing noticed that directive was called from the section code properly. I was trying to solve the issue myself, but with no success. So after few hours of trying different manipulations I asked my colleague for help. He found the error quite fast: the reason was a controller declaration in the section code (ng-controller). And as soon as the section is added to the main tab page by ng-include, it still supposed to use the existing controller. Calling it the second time just reset the values, so everything the directive got was undefined. After removing the extra ng-controller, I got the right values in the directive, but got another error: the query getting data failed. Solving of the problem is left for another day.

27.10.2016.

Today's plan is to continue working on the new view and get the SQL query working.

I started with looking for the reason why the query failing. As soon as the debugger didn't give me any specific error, I needed to get the query from the code and then run it in the SQL developer's environment. To do that I have found the controller that handled getting data from database and found the place, where the string from AppBar was formed and sent to the database server. Breakpoint was hit successfully and I got the query string in the debugger. I didn't have to run anything in the SQL developer, because have noticed the error myself: there was a part of the query with wrong table in it (the error happened because I copied the whole view from the existing one, so all the additional settings were copied as well and looks like I haven't noticed something).

So, started to analyze the query to find out what definition I need to find and change. The problem was with a Join and I didn't know where the Join definitions are. So I had to contact my colleague to get help. It appeared there was a separate form for Join definitions which had Joins copied from the previous view. After removing two extra rows I ran the code again and it began to work. I got all the basic data showing. CSS classes were still missing, so the page was a bit messed up, but decided to leave the CSS for last.

Apart from the basic data that comes for one single table I needed to add indicators that showed status of child elements for that table. The status should be defined based on following: "parameter for the item is not required", "at least one item has parameter missing" and "all items have the parameter missing". I need three statuses for different parameters and all from different tables. So first of all I had to understand the structure of the tables and the data I needed. Again had to get help from my colleague. He explained the relations between main tables I needed, and the main principle I could use to get the related data. So the idea was pretty much the following: there a parent element A. It has two child elements B and C. C is related to B as many to many. C has a child table D, which has child table E. And there is a link from E to a table F that had the status indicators I needed. So I needed to get those indicators from F to B somehow. The task began to look quite difficult.

During the rest of the day I just had the time to get a look on the tables (get different sets of data in SQL developer) and try to understand the relations between tables and think of some ways to get something relevant.

28.10.2016.

Today's plan is to work on the SQL for statuses. After that to work on CSS.

During the day, I tried out different approaches to get the required data from the database. I read different articles about working with NULL values. Tried several approaches like using IF Exists method, NVL and NVL2, tried to write the query with Cases. And even if I got the data it either was the wrong one, or more than one row. I needed to get the status indicator in exactly one column so the query I needed supposed to return one row with one number. In other words, during this day I haven't gotten anything really useful.

There was another work event planned for today: a recruitment evening with students from Aalto University. Before leaving for the event, we had a meeting with other representatives from our company about the presentation we were going to give about Roima and

discussed each of our roles in the presentation. I was asked to go as a most recent worker at the company and at the same time as a representative from our work branch, Lean System.

Almost right after the presentation we left for the event. There I told about myself and shortly about Lean System programmer path in the company. That included telling about some of my work tasks, the languages of programming and the tools we are using for development. There were questions about some specific cases, customer related work and more general questions. After the presentation we had free relaxed discussion with students as about work related topics as general things like studying and life.

The event's goal was to present the company to students close to graduation so there would be possible applicants. The goal of the company is to get 200 employees in the nearest months (the current employee amount is about 150).

Week 8 Evaluation. Discussion and technical background.

This week I was building the view on the client side. I like working on the server side, because C# is very familiar and I feel safe there. Front end side gets a bit more complicated, because there are still nuances that seems to be mysterious for me. Research easily solves this problem.

First, shortly about C# Models and Controllers. In ASP.NET MVC applications controllers are responsible for handling the way the user interacts with an application. It contains the flow control logic and determines what response to send back to a user when the user makes a browser request. A controller is a C# class and usually is situated in Controllers folder in application structure. The class have methods that handle the application logic.

Methods can be public and private. Public methods are exposed as controller actions which could in some cases create a threat to application security as soon as they can be accessed by anyone who has access to application.

A model in ASP.NET MVC application contains all the logic that wasn't handled in the view and in the controller. It contains validation logic, application's business logic and database access logic (Walther 2008).

The LeanSystem application is ASP.NET based but we don't use common MVC logic, so the structure of controllers and models is different from MVC typical structure.

There are HTTP Get and Authorize tags used in controllers that handle the code logic that have access to the client's side.

Shortly, the HTTP Get is a method that handles get requests from a specific resource and receive an HTTP response. Figure 10 shows an example of usage of HttpGet and HttpPost methods in a C# controller.

```
[HttpGet]
public ActionResult Login() {
    return View();
}

[HttpPost]
public ActionResult Login(string userName, string password) {
    // do login stuff
    return View();
}
```

Figure 10. Examples of HTTP GET and POST methods (StackOverflow 2012).

According to Chaihan (2013) the key points about HttpGet method usage for submitting data are:

- Like any Get method, HttpGet requests data from a specific source.
- A hyperlink or action tag are using HttpGet
- Data is submitted as a part of URL.
- Data is posted as a query string, and is visible to the user.
- httpGet method isn't secure, but is used because it is fast.
- Stack method is used for passing form variables.
- Query string has max length and data is limited to it.

Another method used was [Authorize] attribute. Figure 11 shows its usage in a C# controller where it enforces security in the user account section of the web application. [Authorize] attribute adds authentication at the controller level and which is very handy for any web developer, this attribute can be extended with custom code to create own authorization system (Diary of a ninja 2011).

```

<#
[Authorize]
public ActionResult ChangePassword(ChangePasswordModel model)
{
    ...
}

```

Figure 11. Usage of [Authorize] in a C# controller (Diary of a ninja 2011).

The attribute itself provides “granular security” that is implemented directly into a class or a method in the application and allows granular approach to the ASP.Net membership provider.

The annotation used before the method (Figure 11) lets the asp.Net MVC framework the signal, that user needs to be authenticated before the controller code is executed. When the attribute is placed on a method, several calls are made to the AuthorizeAttribute class before each request to the controller in order to authenticate the user (Diary of a ninja 2011).

In which cases these both tags are used together? Authorize attribute above HTTP Get attribute is used to provide better security for the get request. As no of the controllers or methods, marked with authorize, provide access to the controller without checking if the user has authorization.

According to the article by Rick Anderson (2012), the better practice to make a secure ASP.Net application would be to apply the [Authorize] attribute globally. Global filters allow to add AuthorizeAttribute filter to the global.asax file. This action would protect every controller with all its methods. Figure 12 shows how such a filter may be applied.

```

public static void RegisterGlobalFilters(GlobalFilterCollection filters)
{
    filters.Add(new HandleErrorAttribute());
    filters.Add(new System.Web.Mvc.AuthorizeAttribute());
}

```

Figure 12. Example of an applied AuthorizeAttribute filter globally (Anderson 2012).

But Anderson mentions one problem that comes together with global authorization check: how the user would be able to log in? This problem can be solved with applying another

attribute called "AllowAnonymous" which allows user access to login and register functions by overriding authorize. This kind of handling the security application is considered to be best practice in securing action methods (Anderson 2012).

After this week's research, I learnt the reason for those attributes to be used in the application and also learn about best practices in making the ASP.net application secure. I believe this information would be very useful more me in my future work.

I can't make any conclusions about Lean System's security mechanisms because of confidentiality issues.

But I know for sure that I would like to learn more about our own application security and in general.

3.9 Week 9 (31.10.2016-04.11.2016).

31.10.2016 - 01.11.2016.

The task for nearest day remains the same: to find the solution for SQL query.

On Monday I continued to try different solutions to get the data. Nothing worked. I read more about Null expressions, I tried different where expressions, but no success.

I continued my searches on Tuesday and after a while finally got a working expression. At least there were no errors anymore, it took into consideration 3 other tables and returned some data in a format of a status for that table.

02.11.2016.

In the morning, I worked on perfecting SQL: removed everything extra from the string. Then I made changes to the model (as soon as this column wasn't added there at all. After that added changed to the html page adding a new column to the table, and tree balls there. Used Font awesome CSS for that.

Then came another problem: There are three status balls that indicate different situation for the related data: data missing completely, only some missing, check not required and all entered. The SQL query I made returned a number based on the status (say, 1, 2, 3 and there was a separate check for required).

I needed to use an expression to make the color for the ball change depending on the data returned. For that I used ng-class that allows to add CSS styles dynamically. Took me a while to get it working and display the right colors.

03.11.2016.

Plan for today to finish the view: apply some CSS fixes, test the data and to return

In the morning, I asked my colleague to check my queries to make sure the data was fine, because I still wasn't sure about the status column. He checked the queries and yes, the data I got wasn't correct. It appeared that the whole part of the query I wrote to get data was wrong and I needed to do it all over. Well, at least I had working view, and as soon as I can get this query right, the new tab will be almost ready, at least stage one of it.

Rest of the day I was trying different solutions for the query again, and with luckily my colleague wrote me an example for one of the status balls. I just needed to write similar ones for other two. Writing based on example is much easier, so I managed to make a similar query for two other status indicators fast.

I checked that data is coming and got exact examples to test the data. Everything seemed to be working again and my colleague checked it and told that all is now good to go.

I just needed to finish the page now, but suddenly IIS stopped working again and I had to figure out how to start it again. Apparently, there was something wrong with application pools, so I just reinstalled them using own written GUI and project started running. But finishing the page is left for tomorrow.

04.11.2016.

Finishing the view is left for today.

In the morning, we talked about values, the query for status indicator returns. We decided to make it a little bit different, as soon as simple 1, 2, 3 values didn't follow the convention. Also, there was a unnecessary join left from the previous view configuration. I corrected these things in the SQL and after polished the view by making last adjustments into CSS: I needed to table to have some borders and to make item list disappear when item was selected (table was shown based in the chosen element).

When everything was ready I tested everything thoroughly, checked that data is showing for different items, made sure that status indicators are working (had to look for an item with a table that would have all indicators available).

After everything was tested, I returned the files to SourceSafe which also took a while: some of the files I made changes to were already changed by other people and merge doesn't work in Source Safe very well. So, I had to copy my own changes into a separate file, get a fresh version and put my changes back, and to repeat these for all the files I made changes to. Luckily controller, model, JS controller and html file I could just add there without any changes.

The work done this week was quite devastating, because I really had troubles with SQL query, but overall I was very satisfied I have completed the first stage for the task.

Week 9 Evaluation. Discussion and technical background.

I would say that this task has been the most difficult for me since the beginning of working practice at Roima. We have a huge database with many tables. The database is quite old, so there is no model for it anywhere. That means it is impossible to see table relations on a map.

I tried to talk to my colleagues about possibility of building such a map using automated tools, but database administrator from my team told me, that tools the database was built with, are more than 20 years old (which is also the database's age). Which means that there are no possibilities for such automatization in the current version. Which in its own turn, in my opinion, makes working with database very challenging, especially for new people.

Hopefully in the future the database will migrate to a new version and would be somehow built all over. That would probably increase performance of the application significantly. There are many servers for different customers, and I know that their databases are copies of our test development version (in structure). That fact would probably make migration almost impossible and that is why it is very unlikely to happen.

My biggest problem in the current task was the fact i didn't understand the structure of tables I used completely. I completely left out one of the tables (which I didn't even know related), and that affected the query. I think if I knew the structure better I wouldn't have struggled with one SQL query for almost 2 weeks.

Nonetheless, I still learned a lot about SQL queries and subqueries and I believe I began to understand them much better. And of course, I learnt a lot of new things about our own application.

I believe that the most time from recent weeks were devoted to NULL functions in SQL.

Missing unknown data is represented with NULL values in SQL. Table columns can hold NULL values by default. New record or update for an existing record without any value can be inserted to the column if it is optional. And such update means that the value becomes NULL.

NULL values have a different approach than values of any other kind, because basically it is just a placeholder for an unknown or inapplicable value. It is important to notice, that NULL isn't equivalent to 0 and these two values can't be compared. That is the reason the syntax for comparing the value to NULL is different: IS NULL and IS NOT NULL operators are used instead. IS NULL and IS NOT NULL are used in a WHERE clause to get a dataset where this value is presented or not presented (W3Schools.com 1999-2016).

There are different NULL-Related Functions in Oracle that can be used for datasets access: NVL, DECODE, NVL2, COALESCE, NULLIF, LNNVL, NANVL, SYS_OP_MAP_NONNULL.

NVL function allows to replace null values with default values. This function checks if the first value in the parameter is null and if it is it replaces it with the value in parameter 2. It doesn't do any action in case the first value is of any other type and it returns without any changes. Example use: NVL(value1, 'not specified').

NVL2 works very similar to NVL, but accepts already three parameters and replaces the first value in both cases: if it is null and if it isn't null. Second value defined the replacement in case the value1 is not null, the second one is replacement for null values (Hall 2000-2016). I tried to use this function in my queries and the idea in the end was almost correct. I believe it is perfect for getting a Boolean value to check if the column cell is empty or not. The idea for it would be: NVL2(value1, 1, 0) which means in case value1 IS NOT NULL, replace it with 1, otherwise with 0. This query shows if the data for that table row exists for value1.

The DECODE function is used in our database a lot. And even though it isn't directly related to NULL values, it still can be used for replacing them. DECODE can be used also in

cases when value is something else than NULL. The example would be: DECODE(value1, '100', '100 percent', value1). This query would mean: "replace value1 with '100 percent' if it equals 100, otherwise return same value". The second parameter could be NULL as well. But I believe NVL2 is handier in such case.

Another useful function would be COALESCE. It can be used in case there are several columns which values we would like to use, but, for example, only one of the two columns has non-null value for the row and we would like to get the not-null value from these. The syntax would be: COALESCE (value1, value2, value3). This function shouldn't be used in case we want to get all not null values (in case there are more than one in a row), because it returns only the first one.

NULLIF, LNNVL, NANVL and SYS_OP_MAP_NONNULL functions were introduced in later Oracle versions and are also handy. I haven't used them in my work, so won't concentrate on them in details. Main pros of NULLIF function is that it allows to compare two values and returns null if they are both null. LNNVL works the same way as NVL2, but looks for NaN value (which is not a number). And I find SYS_OP_MAP very interesting, because it allows to bypass the fact that two NULL values can't be compared directly to each other (NULL != NULL) . With SYS_OP_MAP_NONNULL it becomes possible (Hall 2000-2016).

What I knew before these weeks, was just basic idea about NULL functions. After this week's research, I now can manipulate the data better using these functions. I had a chance to try several of them, but of course I would like to test them all. Will do that as soon as I have more time.

As I noticed, a lot of queries in our data access code has DECODE and NVL functions in use. But I haven't noticed newer functions. I'm sure there are a lot of cases where it would be possible to get all the benefits of newer Oracle functions and get better queries.

3.10 Week 10 (07.11.2016-11.11.2016).

07.11.2016.

Next stage was to move the content to client's project which was an older version.

As soon as there were a lot of functions that worked in a different way I couldn't just migrate the code from daily version to the client's side and I had to do all the changes by hand.

Model could just be copied, because there was nothing different, but the controller has different way of getting data from the database in the older and newer version. In the newer version, there is a generic method that can process any kind of class with any number of columns. In the older version, the data binding to the model needs to be done for each variable in the model.

After the server side was done, I could copy most of the code parts into related parts in this project. It was clear that some changes needed to be made, as soon as the biggest issue was the fact, that this version didn't have any tabs, so I needed to figure some other way to display the data.

I decided to add another table to the same view and make a switch button between two tables. My supervisor accepted my idea, so I put this kind of switch on my working list.

But before I would make any more changes I decided to check if the data was accessed. For now, my view looked like two table one on top of another.

I run the project, and there was no data in the table. I decided to look the reason the next day, and just wrote some descriptions of the work done at the moment to the source control in the task description.

08.11.2016.

The main task of the day was to solve the data access issue.

It took me a while to find the reason, because when I ran the debugger it made clear, the data was gotten correctly on the server side but never made it to the client's side. I realized that the reason could have been one of the data context controllers that was responsible for the data handling from C# to JavaScript.

In the end, I was right. Found out, that the reason was `String cultureId = ApplicationSecurity.GetSetDefaultCultureId();`. This isn't used in the new version anymore and that why my piece of code didn't have it. After adding that line into the code, the client's side started to get the data just fine.

After the data was showing on the screen, I began to work on the toggle switch button. It is supposed to be a radio button that looks like normal button and it activates certain table

with ng-if and tableVisible variable. I added the code, tested the button and everything seemed fine with functionality. Not I needed to make the styles fit.

First couldn't get float to work. Then took a while to make the active button to stay active. In the end managed to get it working. After that for a while tried to solve the problem with rounded corners: the style supposed to give round outer corners, but in the end the toggle button had them all rounded and it looked bad in the middle of the button. Finally, I decided to remove it completely. Then I fixed indicator colors because old project file didn't have same color theme as the new project did.

09.11.2016.

Today I needed to continue the same task.

The whole day went on cleaning the code: removed extra functions that were not needed in older version, made changes so there were no double function, removed the unnecessary controller (that handled the new view in the new version and wasn't needed anymore as in this version the tables were in the same view), copied some necessary function from it to the current controller, etc.

After cleaning I added string all missing string constants.

And as soon as we want to make experience more comfortable for the user, it was necessary to save the table the user was working at into local storage, so the view was always opened on the current table. I added the function together with toggle-button switch function and load from the local storage on page load (check if the table is in the local storage, if it is, selecting the table and if not, default load).

10.11.2016.

For now, stage one of the task was ready also for the client's project. Not I needed to start stage two.

As soon as our daily version was very close to be changing to the newer one and there were no code changes allowed during this process, the decision was made to make changes first to the client and then add them to the daily version.

Started working on one row selection from the table. Began thinking of how to make the selection better. My first idea was to hide all the other rows but selected. I started to re-search about putting ng-hide inside ng-repeat. But it seemed to be almost impossible solution. So in the end was decided to do another way: with assigning different value to controller array and change this depending on the situation.

Next problem was setting up the communication between two directives: first wanted to find out how to call one's function from another then remembered that there's controller binding and I need nothing else than just assign value to controller.

When all the code was written, I have launched the project and it didn't work. It appeared that I haven't passed the parameter in a right way to the functions.

Corrected the passing and assigned get Methods according to new parameter. Got an error, that offset is less than 1000. It appeared I haven't noticed that there was one more row to pass the parameter. After adding that and doing hard-reset on the page, got the data coming.

After that tried to set up single row view. Got problems with column visibility. Going to correct tomorrow.

11.11.2016.

Continued with the directory issue. Realized that the reason for failure was wrong binding between directory and controller. Didn't pass the *singleMode* value at all back to the controller.

After that got a problem with data loading again. After some research realized, that the controller that handled passing data to server side, did stringify on the array with null value and at that time null turned into empty string and that messed up the query. Added a null check before the stringify.

After that had to change controller so the data handling is responsive to front end changes. Made more changes to controllers (cleaned the code from extra functions and comments). After that wrote about all the changes to source control.

After that started to correct CSS styles. Was very difficult to make the styles working and align tables, because I wasn't allowed to make changes to existing styles, because the

version was directly connected to the production. So, I had to write completely my own and assign them to both tables. Was very difficult to get it working.

Week 10 evaluation. Discussion and technical background.

This was the last week of the diary writing, but of course, there is a lot of work left that is going to be done before the end of the year. The view still needs a lot of work, especially in the daily's version now. So the plan for the next week would be to build row selection and related functions there. I would already foresee a lot of problems, because, like it was already said, the versions are very different and the code needs to be written almost anew there. Also, there are already changes done to that version, so first of all I would have to get to know the changes, analyze what impact they have on view etc.

This week I haven't done anything new in terms of technology and was using everything I have learnt during my time working here. So, I would like to raise a topic of Source Control software used in the company.

As mentioned earlier, we are using Microsoft Visual SourceSafe, which is a source control software that was released in 1994 and which last version was released in 2005, which is 11 years ago, and support has been completely shut down in 2012 (Microsoft).

Like any source control system, it created virtual library of project files and permits working on several project versions at the same time. The minimal functions of SourceSafe are following:

- Protection from accidental file loss
- Tracks each file's history
- Has support for project branches, file sharing and merging
- Has support for file release management
- Allows to track project versions
- Has a track of modular code (if same file is reused or shared by multiple projects (Microsoft | Developer Network)).

While this description might sound good, a lot of developers have the opinion that SourceSafe isn't that safe, and that it lacks some features other Source Control Systems (SCS) provide for their users.

Alan De Smet (2002), for example, doesn't recommend to use this software at all. And the main reason for that is the fact Source Safe doesn't provide any branching support that

would allow developers to make their minor revisions of old versions while there is ongoing work on the new one. Experimental code should also be checked into a branch that would keep it separated from the main project but still available for other developers. Source Safe provides nothing of that. It also freezes the work on the project when the version is changes.

Source safe also has very poor merging system which is integrated with check in process and doesn't allow developers to first merge the code, test it and only the check in to the production code. That means there is a high possibility of non functional code to be checked in into control system (De Smet 2002). The author lists a lot of other cons of the software which makes a very clear point why this source control system shouldn't be used in any kind of project, and especially a large one.

Michael Bolton (2003) is his essay present analysis of SourceSafe disadvantages from a bit another point of view. He names following issues:

- Data Integrity Issues. If there is any kind of change in the file name, its format, the file is moved then it isn't seen by Source Safe as the same file anymore. It loses the connection to the previous versions completely.
- Irritants and Inconsistencies. Share and Rename functions don't permit any comments, project is identified as files, and files and project movement is handled inconsistently
- Automation API issues. In a lot of cases API can't tell the identity of some items (added, deleted, destroyed), and others.
- Documentation issues. VSS has a lot of actions that are not documented, like Pinned, Purged, Recovered, Rollback or Unpinned.

Developers in our company are already well-aware of all the difficulties that SourceSafe brings into our lives. We have experienced problems with merging when committing changes, very uncomfortable UI and just general feeling that the tool is outdated (like it is). And there has been a plan for a long time to migrate all the projects into another Source Control System. Which in my opinion should happen as soon as possible, because it won't only make the life of developers easier, but also increase the safety level. It is weird for me that a big company with a lot of customers and clients is using eleven years old software which has no support. And eleven years is more than a lifetime for any software.

But what are other options? Nowadays there are quite a lot of SCS introduced to the market also for big and multiple projects. The best SCS products are determined by customer satisfaction level and scale. G2 Crowd has made a research and came with the best prod-

ucts based on that. They have separated them into four categories: Ones, that have everything mentioned (scale, market share, resources, customers' satisfaction), high performance, high market presence and resources, and those that has good feedback from users but are low scale.

Leaders include Microsoft Team Foundation Server and Subversion, high performance are Git, Helix and Mercurial, and niche products are RationalClearCase (G2 Crowd 2016).

There are also many other systems like Subversion, AccuRev, SVC, SVN and so on.

As it is shown above, there are a lot of different options that would fit Roima much better than Visual Source Safe. There have been talks already that Roima will migrate the project to the GitHub in the nearest future. But other options are still under consideration. But at least it is very good that it is decided to change the way the project files are handled.

4 Discussion and Conclusions

How have you progressed and developed yourself? I have started my work at the company in May. Back then I was very “green” developer with no previous work experience, with very little knowledge about web developing, especially front end, and little background of back end development that was based on building two project for Software development courses. I had very narrow picture of HTML and even narrower of CSS. I was completely incompetent in using AngularJS and JavaScript in general. The only real help I got in my starting situation was the environment: we use VisualStudio and I was already quite familiar with it and some helpful tricks. Also C# was used for backend, and even though I had to start with learning how to work there with front end first, it was still calming to know that there is something in the project I would be able to do with higher performance. But that was in the very beginning.

By the time I started on working diary I have already learnt a lot about AngularJS, about structure of the project and principles of its working. And in the beginning of September, I believe, I have started to do the same tasks other developer in the team were doing. The efficiency wasn't that high still, because I had to research a lot, to read about some aspects of technology and every time I started something new I spent a lot of time on figuring out how that part of our application works.

Today I am much more competent that I was in the beginning, and know more about some technologies and especially our project, than in the beginning of this thesis. During these weeks, I learnt a lot of details about the application, I got familiar with additional configurational possibilities and learn the ways to add new features to the application. There are still a lot of challenges for me in terms of technology and issues with my coding. For example, I still find SCC and its logics to be a mystery, I still haven't gotten familiar with some parts of our application (I would like to spend some free time on that, but unfortunately the work load was so high I just can't find the time to brows freely through the application), so I will have to my research when I get a task from that unknown area. I still write a lot of extra functions in angular controllers. I really need to become more careful and professional in terms of writing the code and teach myself how to get a clearer picture of requirements.

But despite everything, during this period I grew as a professional a lot, learnt a lot of new things and developed myself as a full stack programmer.

What kind of new approaches or methods you have found for your work? What did you learn? As it was already said, I have learnt a lot during this time period. I believe that my main achievements were:

- getting familiar with AngularJS directives. I have learnt everything about them from scratch. I didn't have an opportunity to learn how they work before. At first the concept of directives seemed tricky for me, but very soon I got ahold of this technology and realized it is a powerful tool for application customization and in a way, they are much easier to build than controllers.
- I have learnt how to add a new view for the portal and how to assign configurations for it. As it seems the approach for adding new views isn't quite general at Roima, so I really had to learn how to use our own tool, how to make setups there and how to make setups in the application so the application and the tool can communicate with each other.
- I had a rapid and harsh introduction into PL/SQL and usage of our database. Metaphorically I have been thrown into the deadly sea of our database and survived (with help though), at the same time learning how to manage there on my own. I learnt about new functions for data fetching queries and also about a lot more functions it is possible to with SQL, like assigning triggers for some events. I believe any other SQL task I would get wouldn't look that scary anymore.

As for the tools and methods, I haven't done any revolutionary changes into the way the project works. I did the thorough research on the subjects that were on my work list, I raised discussions with my colleagues about some ways new features could be implemented. But there are big obstacles in the way a mere trainee can change anything: first of all, the project is very large, and none of the changes, especially revolutionary changes, are welcomed here. Many customers' work depends on the project directly. Any big changes done to the project would probably demand changes on the client servers which may lead to the domino effect of some of the features crushing. So, every change should have extremely good reason to happen. That is the reason some tools used in the company are so old and even outdated – no one wants to touch anything while it already works.

Another reason I really couldn't implement anything new is the fact we have very good specialists working in the company that are aware of best practices for the tools. So, they were already implemented in the project or are planned to be implemented, like migrating Source Control System and automatization tests.

Did you get any new ideas for future development? I like working in this company and I like working as a full stack developer. It would be very difficult to tell which side of the development I like more. Backend feels safe and familiar (because of C# which I know already quite well), it is exciting what it is possible to do with SQL, even though I am not

very fluent with it. I would really like to know more now about SQL and database management.

I also like frontend, both writing functionality and making the design. But here there already a clear need for development: I need to learn more about CSS so I won't spend so much time on making the styles with try-and-error, but instead just understand well what I am doing and increasing the efficiency of my performance. And of course, there are still a lot of possibilities to learn more about AngularJS, especially with Angular2 released.

I am really happy I got an opportunity to work as full stack developer, because I tried myself in pretty much all the web development areas and learned a lot about both back end and front end, that would give me very good opportunity in the future as a specialist when finally, I would be ready to decide which track I would take.

If talking about development area, I still don't know well where I would like to head. Roima is very good company with very competent and nice employees so I really enjoy working there. But I still would like to head somewhere where my work would have a bigger impact on the world, and help people somehow.

How have you been able to take advantage of the work analysis? I would have done the research and analysis anyway but just at a smaller scale. I would probably just consider the particular problem, without trying to get the perspective on other methods available.

The work analysis gave me the perspective on the problem, so I learnt about other methods and tools that can be used in different situations, tried them out in practice and of course this research had a positive impact on me as a professional. Apart from that, I got new topics to discuss with my colleagues, I can now raise new questions and discuss possible features to develop in the future (even though it is extremely hard in the company, but I feel like everyone's opinion matters here).

I know that I will do my best to add this kind of research to my work routine, because I have seen the benefits it comes with.

5 References

Ammann, P., Offutt, J. 2008. Introduction to Software Testing. 1ed. Cambridge University Press. New York.

Anderson, R. 2012. Securing your ASP.NET MVC 4 App and the new AllowAnonymous Attribute. Microsoft | Developer. URL: <https://blogs.msdn.microsoft.com/rickandy/2012/03/23/securing-your-asp-net-mvc-4-app-and-the-new-allowanonymous-attribute/>. Accessed: 01.11.2016.

AngularJS. 2010-2016. Developer Guide/ Using \$location. URL: [https://docs.angularjs.org/guide/\\$location](https://docs.angularjs.org/guide/$location). Accessed: 09 October 2016.

Beck, K. 2002. Test Driven Development: By Example. 1 ed. Addison-Wesley Professional. Canada.

Belatrix. 2015. Whitepaper: Functional testing Best practices. URL: <http://www.belatrixsf.com/whitepapers/functional-testing-best-practices/>. Accessed: 01 October 2016.

Bolton, M. 2003. Visual SourceSafe Version Control: Unsafe at any Speed? Essay. URL: <http://www.developsense.com/testing/VSSDefects.html>. Accessed: 13 November 2016.

Breeze. EntityManager. URL: <http://breeze.github.io/doc-js/entitymanager-and-caching.html>. Accessed: 27 October 2016.

Chauhan, S. 2013. Difference between HttpGet and HttpPost Method. DotNetTricks. URL: <http://www.dotnettricks.com/learn/aspnet/difference-between-httpget-and-httpost-method>. Accessed: 30 October 2016.

Coyier, C. 2016. A complete Guide to Flexbox. URL: <https://css-tricks.com/snippets/css/a-guide-to-flexbox>. Accessed: 11 September 2016.

De Smet, A. 2002. Visual SourceSafe: Microsoft's Source Destruction System. High Programmer. URL: <http://www.highprogrammer.com/alan/windev/sourcesafe.html>. Accessed: 12 November 2016.

Diary of a Ninja. 2011. Writing your own custom ASP.Net MVC [Authorize] attributes. URL: <http://www.diaryofaninja.com/blog/2011/07/24/writing-your-own-custom-aspnet-mvc-authorize-attributes>. Accessed: 29 October 2016.

Feuerstein, S. 2001. Oracle PL/SQL Best Practices. 1 ed. O'Reilly Media. Sebastopol.

Feuerstein, S., Pribyl, B., Dawes, C. 2004. Oracle PL/SQL Language Pocket reference. 3 ed. O'Reilly Media. Sebastopol.

Fowler, M., Beck, K. 2002. Refactoring: Improving the Design of Existing Code. Addison Wesley.

G2 Crowd. 2016. Best Version Control Systems. URL: <https://www.g2crowd.com/categories/version-control-systems>. Accessed: 13 November 2016.

Guzel, B. 2009. Top 20+ MySQL Best Practices. Enavtotuts+. URL: <https://code.tutsplus.com/tutorials/top-20-mysql-best-practices--net-7855>. Accessed: 22 October 2016.

- Hall, T. NULL-Related Functions. Oracle-Base.com. URL: <https://oracle-base.com/articles/misc/null-related-functions>. Accessed: 05 November 2016.
- Kearn, M. 2015. Securing and securely calling Web API and [Authorize]. Microsoft. URL: <https://blogs.msdn.microsoft.com/martinkearn/2015/03/25/securing-and-securely-calling-web-api-and-authorize/>. Accessed: 30 October 2016.
- Laurie, V. 2012. Batch Files (Scripts) in Windows. URL: <http://commandwindows.com/batch.htm>. Accessed: 01 October 2016.
- LISTAGG. 2016. Database SQL Language Reference. URL: https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions089.htm#SQLRF30030. Accessed: 14 October 2016.
- Mar, K. 2012. The benefits of Test Driven Development. URL: <https://scrumology.com/the-benefits-of-tdd-why-tdd-part-3/>. Accessed: 02 October 2016.
- McCown, S. 2010. 7 performance tips for faster SQL queries. InfoWorld. URL: <http://www.infoworld.com/article/2628420/database/database-7-performance-tips-for-faster-sql-queries.html>. Accessed: 22 October 2016.
- Microsoft | Developer Network. Introducing Visual SourceSafe. URL: [https://msdn.microsoft.com/en-us/library/3h0544kx\(v=vs.80\).aspx](https://msdn.microsoft.com/en-us/library/3h0544kx(v=vs.80).aspx). Accessed: 17 November 2016.
- Microsoft. ref (C# Reference). URL: <https://msdn.microsoft.com/en-us/library/14akc2c7.aspx>. Accessed: 02 October 2016.
- Precht, P. 2015. Binding to directive Controller in Angular 1.3. URL: <http://blog.thoughttram.io/angularjs/2015/01/02/exploring-angular-1.3-bindToController.html>. Accessed: 08 October 2016.
- Rodríguez, J.R. 2015. Deep dive into Angular.js development. Electronic version. Leanpub.
- Snook, J. Scalable and ModularArchitecture for CSS. A flexible guide to developing sites small and large. URL: <https://smacss.com/book/about>. Accessed: 05 November 2016.
- SS64.com. ROBOCOPY.exe. URL: <http://ss64.com/nt/robocopy.html>. Accessed: 27 September 2016.
- Stackoverflow. 2010. Windows Bat file optional argument parsing. URL: <http://stackoverflow.com/questions/3973824/windows-bat-file-optional-argument-parsing?noredirect=1&lq=1>. Accessed: 27 September 2016.
- TechNet. 2010. Robocopy and a Few Examples. URL: <https://social.technet.microsoft.com/wiki/contents/articles/1073.robocopy-and-a-few-examples.aspx>. Accessed: 03 October 2016.
- TechNet. 2016. Tip: Get to Know Robocopy for More Powerful File Management. URL: <https://technet.microsoft.com/en-us/magazine/ee851678.aspx>. Accessed: 03 September 2016.
- TechNet. 2016. Using batch files. URL: <https://technet.microsoft.com/en-us/library/bb490869.aspx>. Accessed: 01 October 2016.

W3schools. HTML <input> autocomplete Attribute. URL: http://www.w3schools.com/tags/att_input_autocomplete.asp. Accessed: 05 October 2016.

W3schools. HTML5 Local Storage. URL: http://www.w3schools.com/html/html5_webstorage.asp. Accessed: 05 October 2016.

W3schools. SQL Null Values. URL: http://www.w3schools.com/sql/sql_null_values.asp. Accessed: 06 November 2016.

Walther, S. 2008. Understanding Models, Views, and Controllers (C#). ASP.NET. Microsoft. URL: <https://www.asp.net/mvc/overview/older-versions-1/overview/understanding-models-views-and-controllers-cs>. Accessed: 30 October 2016.

Wembler, S. 2013. Introduction to Test Driven Development (TDD). URL: <http://agiledata.org/essays/tdd.html>. Accessed: 02 October 2016.

Zerr, J. 2014. AngularJS Directive Best Practices. Jeremy Zerr's Tech Blog. URL: <http://www.jeremyzerr.com/angularjs-directive-best-practices>. Accessed: 09 October 2016.

Appendices

Appendix 1. Oracle and MySQL differences (Rapid Programming).

This table show comparison of MySQL and Oracle features.

MySQL	Oracle
MySQL is an open source and MySQL is available for free download and installation.	Only Oracle Express Edition is free of cost. But Oracle Express Edition has very limited features when compared to MySQL. For extensive features, either Oracle Standard Edition or Oracle Enterprise Edition has to be purchased.
User authentication is performed in MySQL by using only location, username and password.	Oracle provides enhanced database security. User authentication is performed in Oracle by specifying global roles in addition to location, username and password. In Oracle, User authentication is performed by different authentication methods including database authentication, external authentication and proxy authentication.
Flexibility of creating stored procedures and functions using PL/SQL is very less in MySQL.	Oracle provides more flexible features for creating stored procedures and functions using PL/SQL.
MySQL offers very few commands related to generating output as report and defining variables. MySQL includes only very simple SQL commands.	Oracle includes extensive SQL commands in SQL*Plus including commands for generating output as report and defining variables.
MySQL does not have the audit vault feature in the server.	Oracle provides audit vault facility.
MySQL does not offer tools at enterprise level.	Oracle offers tools at enterprise level.
MySQL has only table locking facility.	Oracle provides the row locking facility as well.
MySQL does not have extensive storage features like tablespace, synonym, packages and many others.	Oracle has a very extensive storage features. Oracle supports tablespace, synonym, packages and all other features.
MySQL database does not support XML.	Oracle supports and uses XML.
MySQL supports only two character types namely CHAR and VARCHAR.	Oracle supports four different character data types namely: CHAR, VARCHAR2, NCHAR, NVARCHAR2.
In MySQL, temporary tables are visible only within the current active session. When the session expires, the temporary tables are removed automatically.	In Oracle, temporary tables are persistent across sessions. The temporary table has to be explicitly removed by the User.
MySQL has only two backup mechanisms namely mysqlhotcopy and mysqldump.	Oracle offers many backup mechanisms including hot backup, backup, import, export and many others.