

Saimaa University of Applied Sciences
Faculty of Technology, Lappeenranta
Double Degree
Information Technology

Petr Bartusek, Lukáš Zajac

Gate-Check System for Outotec Company

Thesis 2014

Abstract

Petr Bartusek, Lukáš Zajac

Gate-Check System, 60 pages

Saimaa University of Applied Sciences

Faculty of Technology, Lappeenranta

Double Degree

Information Technology

Thesis 2014

Instructors: Lecturer Yrjö Utti, Saimaa University of Applied Sciences

Eero Enovaara, Senior Specialist - CRM applications, Outotec

The main purpose of the thesis was to create the interactive, electronic web application Gate-Check for Outotec Company, which replaced the old system based on MS Excel document. Another purpose was gaining new skills and experiences in communication with the company, in which the team acted as a Supply Company.

The created web application (Gate-Check System) should be compatible with any device with integrated web browser with support HTML5, JavaScript and CSS3 and active internet connection. State-of-the-art technology, such as HTML5, CSS3, JavaScript and PHP were used during the development process.

The final result of this thesis is a web application, which has met all customer requirements. The complete application enables the end user to take a survey, save it and show a graphic result. Also there is an administrator section for managing surveys and exporting results.

The work was consulted and supervised by Mr. Juha Kohvakka. Source data for development and the MS Excel version of Gate-Check was provided by Outotec Company as well.

Keywords: PHP, MYSQL, CSS3, HTML5, JavaScript

Table of contents

Terminology	4
1 Introduction	7
1.1 Outotec Company	7
2 Analysis	10
2.1 Old Gate-Check system	10
2.2 New Gate-Check system	12
3 Development.....	17
3.1 Development tools	17
3.1.1 NetBeans IDE 8.0.....	17
3.1.2 Git.....	18
3.1.3 Bugzilla.....	20
3.1.4 MySQL Workbench 6.0	21
3.1.5 XAMPP server.....	22
3.1.6 HelpNDoc.....	23
3.2 Technologies	24
3.2.1 PHP.....	24
3.2.2 HTML5	26
3.2.3 CSS3.....	27
3.2.4 JavaScript	28
3.3 Development process	29
3.3.1 Iterative and incremental development.....	30
4 Gate-Check application	32
4.1 Database Layer	32
4.2 User Interface	33
4.2.1 Siebel connection	33
4.2.2 Main application	37
4.2.3 History	41
4.2.4 Gate Check	43
4.2.5 Administration	44
4.3 Business Logic.....	48
4.3.1 Login	48
4.3.2 Intelligent GID completion	49
4.3.3 Inputs validation	49
4.3.4 Risk calculation	51
4.3.5 Other features	52
5 Conclusion	53
List of images.....	54
List of source codes	56
List of references	57
Annex:.....	58
Database data dictionary:.....	58

Terminology

- ASP.NET** Server-side web application framework designed for web development to produce dynamic Web pages.
- BSD** Berkeley Software Distribution – UNIX operating system.
- Bugzilla** Web-based general-purpose bugtracker and testing tool originally developed and used by the Mozilla project.
- CSS** Cascading Style Sheets
- CSV** Comma-Separated Values
- C, C++** Programming Language
- Dibi** PHP library for database connection.
- EPUB** Electronic Publication
- ER** Entity–relationship model is a data model for describing the data or information aspects of a business domain or its process requirements.
- Gate-Check** Name of old a new Outotec risk system.
- GID** Gate-Check identifier – A main uniquely identifier in database
- Git** Distributed revision control and source code management system.
- HTML** HyperText Markup language
- IDE** Integrated development environment
- IS** Information system
- JAVA** Programming language
- JavaScript** Dynamic computer programming language.
- JQUERY** JavaScript library
- JSON** JavaScript Object Notation is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs.
- J2EE** Java platform to Enterprise Edition

Localhost Server running on your computer

MS Excel Table editor produced by Microsoft

MySQL Open source SQL database

ODBC Standard programming language middleware API for accessing database management systems.

PDF Portable document format

PHP PHP: Hypertext Preprocessor, Server side scripting language

PHPMailer Library for sending emails via PHP codes from a webserver.

RTF Rich Text Format is a proprietary document file format with published specification developed by Microsoft Corporation.

SA Service area

SAID Service area identification number

SCM The management of changes to documents, computer programs, large web sites, and other collections of information.

SHA1 Strong cryptographic hash algorithm

Siebel IS customer relationship management application developed by Oracle Corporation.

SQL Standardised language for database queries

TAG HTML basic element

TCP Transmission Control Protocol

UI User interface

Use case diagram

W3C Main international standards organization for the World Wide Web.

Wireshark Open source network packet analyser

WYSIWYG „What you see is what you get“

XAMPP Free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP.

XML Extensible Markup Language

1 Introduction

These days computers have infiltrated into all branches of human lives. Using a computer can save a lot of time, decrease chance of human mistake and increase the productivity of work.

Computer applications widely replace a lot of office work. Storing and collecting information are the two most important purposes of surveys themselves. Replacing an MS Excel version of survey by an interactive web application provides several advantages in front by high user comfort during processing and synoptically representing results in real time after filling in.

The most important thing of developing process is gathering information about the actual process to manage all customer requirements. This information is necessary to correctly design the final product to reach a satisfied customer after analysis.

There are many ways and technologies how to develop the web application. It is possible to use very popular Microsoft technologies and their ASP.NET or Java and their platform J2EE (Java Platform, Enterprise Edition), but after discussion with the customer PHP (PHP: Hypertext Preprocessor) was chosen. So the web applications contain files in HTML5, CSS3 (Cascading style sheets), JavaScript and PHP format. For storing data MySQL database is used.

The final deployed web application is connected to the main information system in Outotec, Siebel, through which a normal user is accessing services provided by the software. Only the administrator user has access directly to the application.

1.1 Outotec Company

It is very important to get familiar with the company and what the company does. Outotec is a worldwide company providing solid-liquid separation solutions. This company has been keeping a strong position across the whole manufacturing

chain from mine processing to metal manufacture and currently employs almost 5000¹ employees in the whole world.

Currently Outotec Company provides solutions, development and technology mainly for the following industry areas:

- Minerals and metals
- Chemical
- Energy
- Water treatment

Sophisticated global network of sales and service centres combined with using environment friendly technologies and manufacturing processes helps to keep high profitability and strong market position. Continuous innovations such as automation and new information management (information technology solutions, information systems, et cetera), combined with state-of-the-art research facilities predetermine this corporation to potential growth in future and getting even larger market share.

The basic division of Outotec business areas:

1. Non-ferrous solutions

This area includes all solutions for processing copper, nickel, zinc, lead, gold, silver and PGM's from mine to pure metal. It also contains research and development facilities dealing with this area.

2. Ferrous solutions

This area covers all solutions for processing iron ore, ferroalloys and titanium feedstock.

3. Energy, Light Metals and Environmental Solutions

¹ Wikipedia contributors, "Outotec," *Wikipedia, The Free Encyclopedia*, <http://en.wikipedia.org/w/index.php?title=Outotec&oldid=600951029> (accessed March 25, 2014).

These are solutions for new applications of proven technologies. Outotec Company also provides after sale service based on unique approach of combination providing sustainable solutions, products and related customer service.

2 Analysis

An analysis of the problem is the basic thing which should be done at the beginning of every project. An analysis is the core of software development process. When you make a mistake in the analysis and find it somewhere during developing, it could be a big problem, because you will probably have to change many implemented things.

It is very important to have good communication between the customer and the developer from the very beginning. It is necessary to discuss with the customer about his requests, ideas and technical requirements. We have to know how exactly processes are running to implement the system in the right way. Any limitations which will be found later, can delay and increase cost of development.

2.1 Old Gate-Check system

Currently they are using Microsoft Excel document (Figure 1), in which all questions of the survey are written.

Gate	Ch.L.	Sere	Quest	Alternatives	Risk Points	Description / Instruction
Gate 2	Ch.L. 1	SAI-SA 1	1.1	All/Some/None checked	10/10	Customer's legal status (the country of customer's legal company entity)
Gate 2	Ch.L. 2	SAI-SA 1	1.2	All/Some/None checked	10/10	Define customer type
Gate 2	Ch.L. 3	SAI-SA 1	1.3	All/Some/None checked	10/10	Define project type
Gate 2	Ch.L. 4	SAI-SA 1	1.4	All/Some/None checked	10/10	Customer's financial situation (e.g. systems of creditworthiness etc.)
Gate 2	Ch.L. 5	SAI-SA 1	1.5	All/Some/None checked	10/10	Are the customer's primary business models identified?
Gate 2	Ch.L. 6	SAI-SA 1	1.6	All/Some/None checked	10/10	Required technology / service
Gate 2	Ch.L. 7	SAI-SA 1	1.7	All/Some/None checked	10/10	Are there other relevant cases existing?
Gate 2	Ch.L. 8	SAI-SA 1	1.8	All/Some/None checked	10/10	Control system history
Gate 2	Ch.L. 9	SAI-SA 1	1.9	All/Some/None checked	10/10	Complexity?
Gate 2	Ch.L. 10	SAI-SA 1	1.10	All/Some/None checked	10/10	Does the scope include new or existing equipment/systems?
Gate 2	Ch.L. 11	SAI-SA 1	1.11	All/Some/None checked	10/10	Does the scope include design/engineering?
Gate 2	Ch.L. 12	SAI-SA 1	1.12	All/Some/None checked	10/10	Does the scope include installation/commissioning?
Gate 2	Ch.L. 13	SAI-SA 1	1.13	All/Some/None checked	10/10	Does the scope include training?
Gate 2	Ch.L. 14	SAI-SA 1	1.14	All/Some/None checked	10/10	Does the scope include support systems included in the scope?
Gate 2	Ch.L. 15	SAI-SA 1	1.15	All/Some/None checked	10/10	System integration/interoperable?
Gate 2	Ch.L. 16	SAI-SA 1	1.16	All/Some/None checked	10/10	Are needed customer resources available for preparation of sales case?
Gate 2	Ch.L. 17	SAI-SA 1	1.17	All/Some/None checked	10/10	Is full scope of contract price based on actual production performance?
Gate 2	Ch.L. 18	SAI-SA 1	1.18	All/Some/None checked	10/10	Does the sales case include fixed fee pricing?
Gate 2	Ch.L. 19	SAI-SA 1	1.19	All/Some/None checked	10/10	Is full scope of the contract based on grossed and actual consumption?
Gate 2	Ch.L. 20	SAI-SA 1	1.20	All/Some/None checked	10/10	Are all necessary support systems included?
Gate 2	Ch.L. 21	SAI-SA 1	1.21	All/Some/None checked	10/10	Are all necessary support systems included?

Figure 1 – Old Gate-Check system

There are seven columns in the document:

- Checklists
- Service product category
- Question number
- Question(s)
- Alternatives
- Risk points
- Description / instructions

The employee who wants to answer the survey has to choose questions, which are required for his task. This is made by choosing questions with right Gate ID (Gate 2, Gate 3 or Gate 4) from Checklist column and service product category (SA1 – SA7) from Service product category column. Questions on rows where checklist and service product category have right values should be answered.

In column Question(s) there is a question on which you have to choose one answer from column Alternatives. Every opinion is marked by letter A – Z, in column Risk points you find by letter the value of risk for chosen answer. There are special answers, where you should specify why you chose this option and also there is special risk value on some answers, where you have to specify your own value of risk. These special options and risk values are noticed by word *specify*.

Currently the results are written to the new MS excel document, in which the number of question is written on every row. The chosen answer, if the answer should be specified is written the specify answer, the next column contains the risk value (copied from the document with all data or your own, if the risk value should be specified). After answering all necessary questions, the risk manager has to make a sum of all risks to get the final result - the risk value for selected gate (Checklist) and service area (Service product category).

This procedure is not very non-effective, because there is high possibility of human mistake, there is no system for fast searching in history and there is also no system for global statistics.

The system is for today's needs and technologies quite unusable. The solution is to make a new and user friendly information system from scratch.

2.2 New Gate-Check system

The system is designed according to the Outotec company requirements. This part describes only the final version of Gate-Check system, the prototype functionality will be described later in the developing section.

The basic idea of the new system is to be like electronic surveillance systems. The main functions of the system are described in use case diagram (Figure 2).

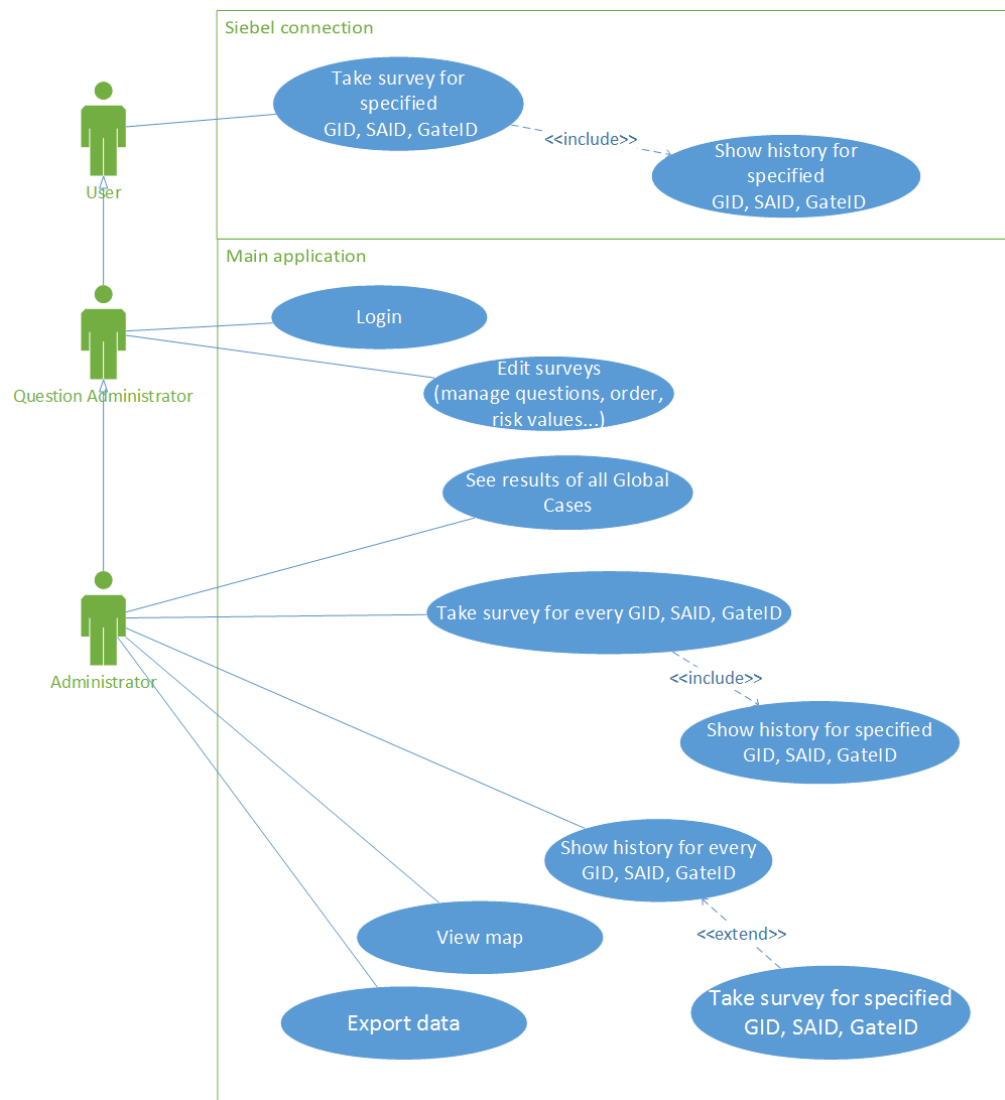


Figure 2 – Use case diagram

There are three levels of users in the new system:

- *User* – Connection from Siebel.
- *Question Administrator* – Directly connected to main Application.
- *Administrator* – Directly connected to main Application.

The *User* can:

- Take survey for specified GID, SAID, GateID – Fill in the survey generated by given parameters from Siebel and save it to the local database.
- Show history for specified GID, SAID, GateID – The history is shown after every successfully saved survey. The results of the survey are shown in the user friendly look. The result risk is sent to the Siebel database.

Question Administrator can do:

- The same possibilities as *User*.
- Login – Log into the system with his user name and password.
- Edit surveys – Access to administrator section of surveys, where user can change/add/delete question(s) in gates and change order of questions in gate. User needs to be logged in.

Administrator can do:

- The same possibilities as *Question Administrator*.
- Take survey for every GID, SAID, GateID – Fill in the survey generated by parameters and save it. User needs to be logged in.
- Show history for specified GID, SAID, GateID – The history is shown after every successfully saved survey. Results of the survey are shown in the user friendly look. User needs to be logged in.
- Show history for every GID, SAID, GateID – The history is shown after every successfully saved survey.. Results of the survey are shown in the user friendly look. User needs to be logged in.
- Take survey for specified GID, SAID, GateID – Starts if history of that survey does not exist. Fill in the survey generated by given parameters and save it. User needs to be logged in.

- View map – Map shows to the user average risk values for every gate in states. User needs to be logged in.
- Export data – Export data to csv file.
- See the results of all Global Cases – Show table of all global cases in the local database. User needs to be logged in.

The application is divided into two parts as shown in use case diagram (Figure 2). The first part is accessible only through the Siebel IS (Figure 3)

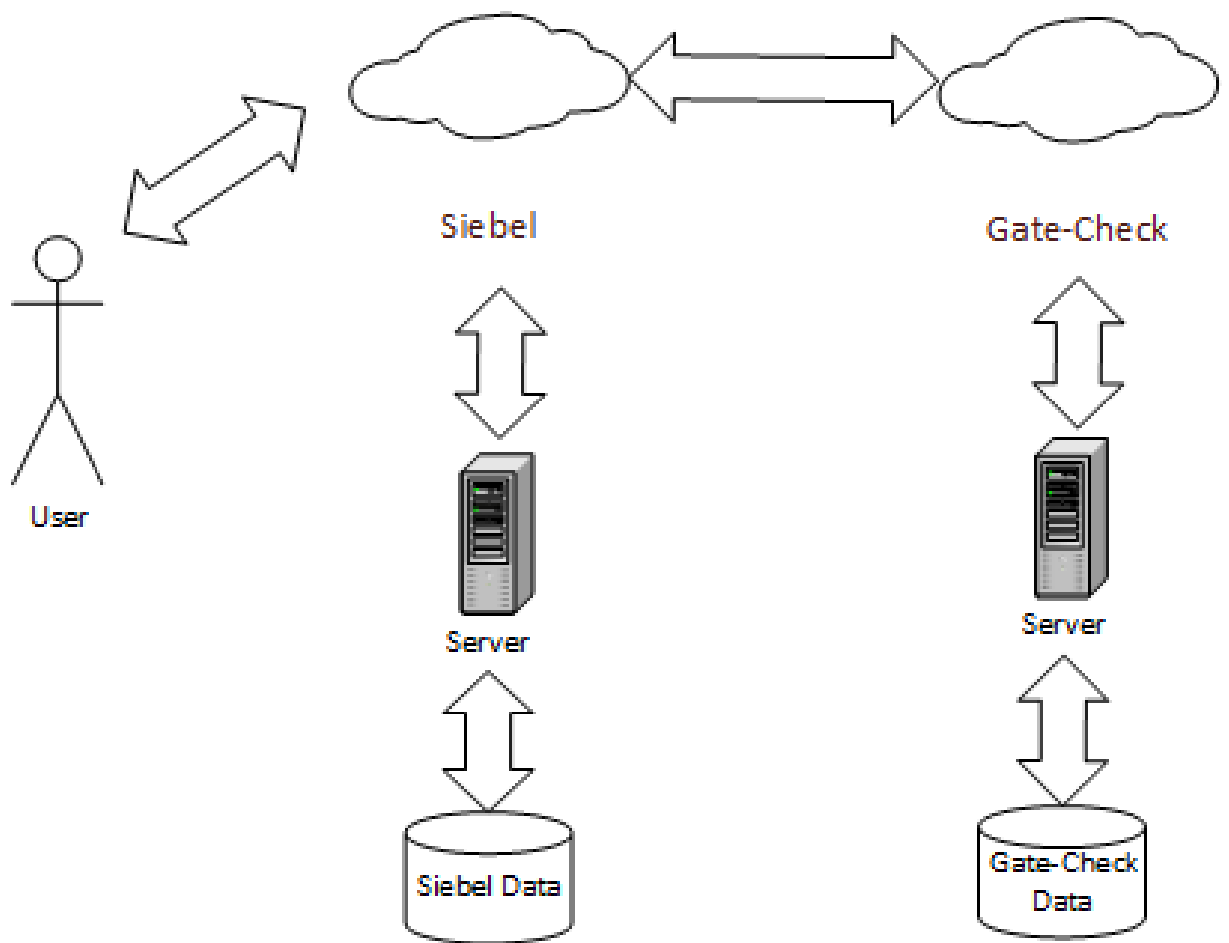


Figure 3 – Connection through Siebel

The second part of the application is accessed directly, which means that the *User* is connected to the Gate-Check application (Figure 4).

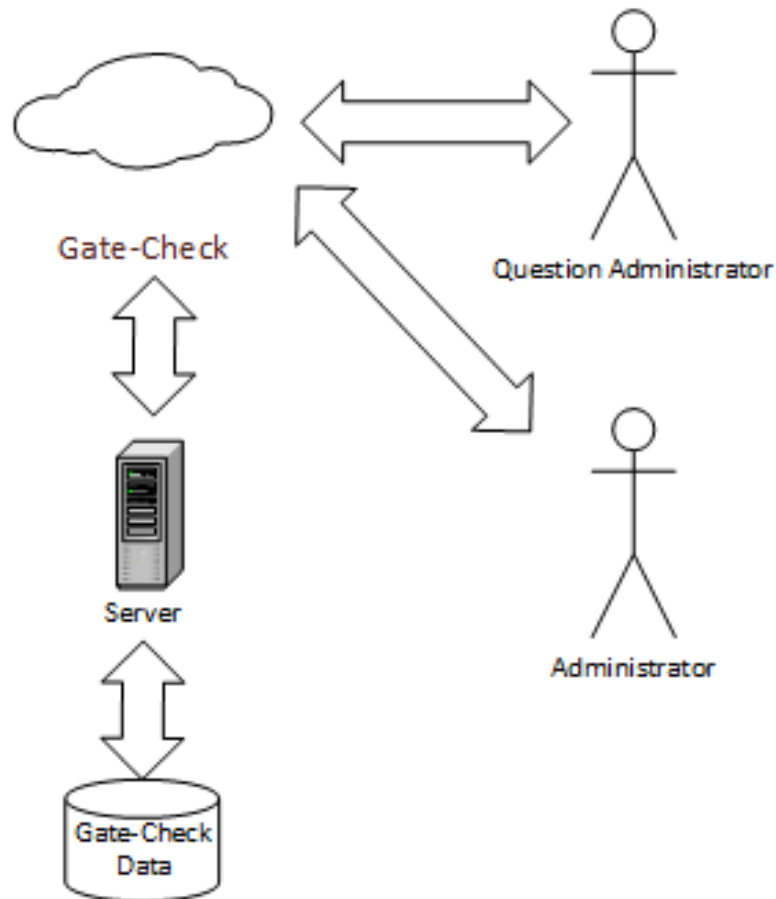


Figure 4 – Connection directly to Gate-Check application

The architecture of the new version Gate-Check system is divided into three layers (Figure 5). Splitting architecture into layers is highly recommended for future maintenance and scalability. For example the User Interface can be easily changed without any changes in Business Logic or Database Layer.

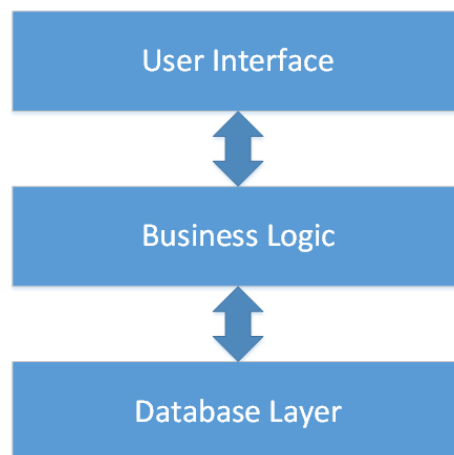


Figure 5 – Architecture

Database structure (Figure 6) is designed on customer needs and with all new and recommended practises for creating a robust database.

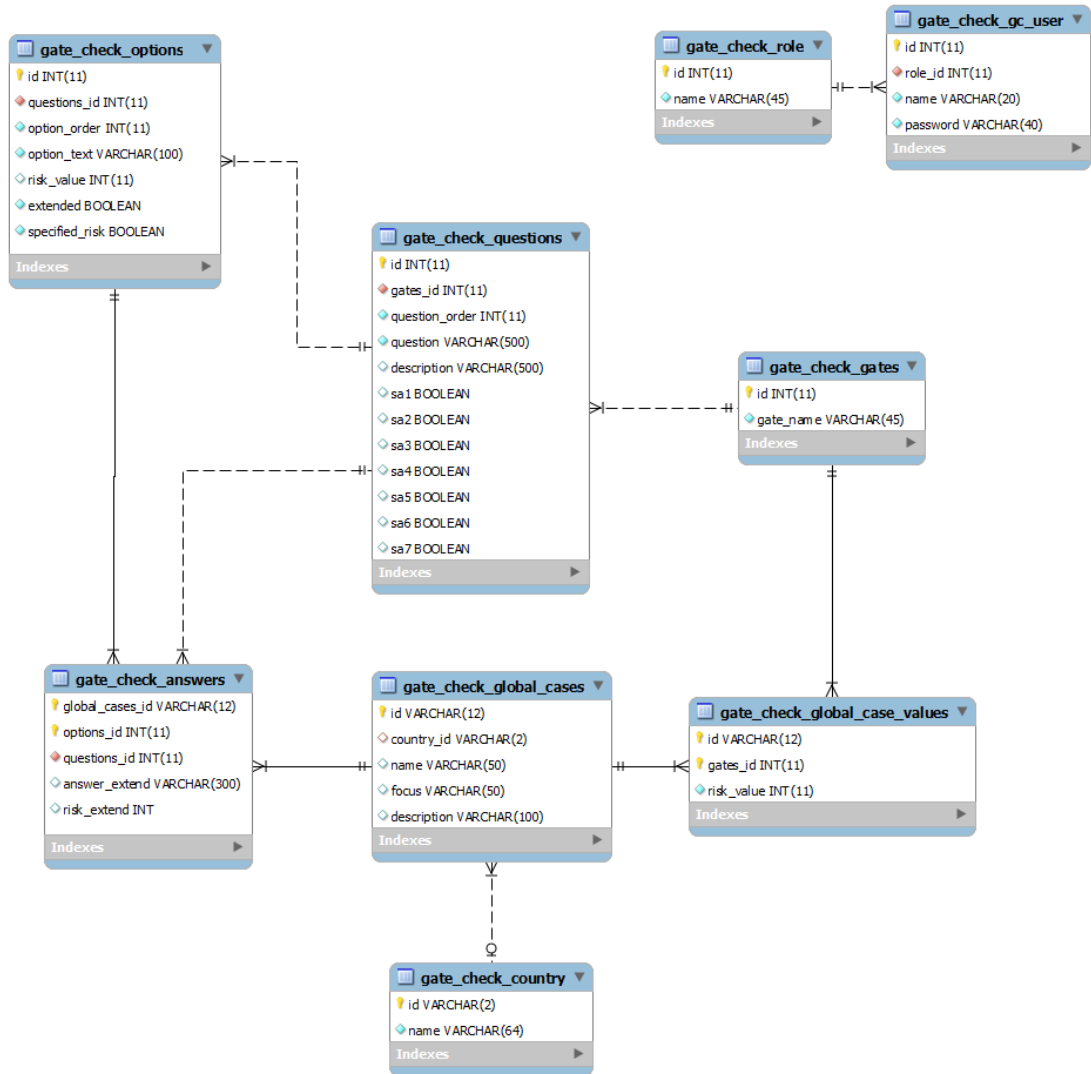


Figure 6 – Database structure

Example of tables, complete data dictionary is in thesis annex.

gate_check_global_cases – table contain all global cases whis was inserted into Gate-Check system		
id	VARCHAR(12)	clear identification of global case
country_id	VARCHAR(2)	clear identification of country in ISO 3166-1 alpha 2 format
name	VARCHAR(50)	name of the company which has this global case number
focus	VARCHAR(50)	focus of company which has this global case number
description	VARCHAR(100)	description – not used now

3 Development

Development is the main part of software development process. This part includes customer requirements, which were specified during the analysis of software development process and transformed into software application by programmers. Programmers are using many development techniques, tools, frameworks, technologies and platforms to reach the goal (the fully functional application according to customer needs).

3.1 Development tools

Development tools are the most important things, which can help the programmer in work. If right tools are chosen, you can save a lot of time in development, because programmers make less mistakes and work better.

3.1.1 NetBeans IDE 8.0

NetBeans IDE (Integrated Development Environment) in Figure 7 is an open-source development environment basically purposed for developing in JAVA, but there are many plugins. By installing plugin you can very easily improve support for other platforms like C, C++, Android and PHP and so on.

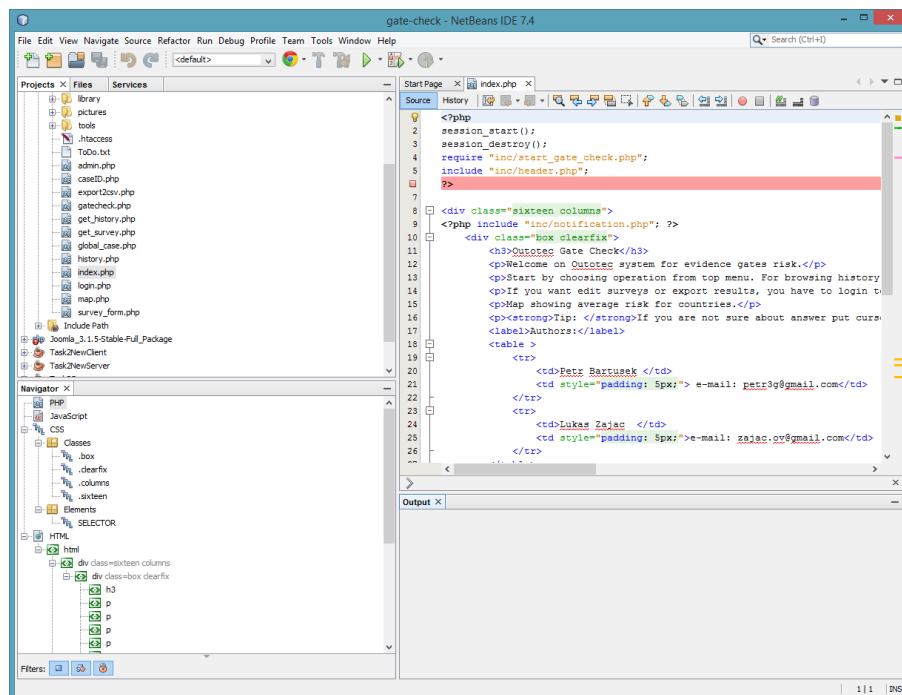


Figure 7 – NetBeans IDE 8.0

NetBeans IDE was chosen as the main developing tool for many reasons:

- It is free, it can be used for a commercial project without any investments.
- Support of PHP added by plugin.
- Support of HTML5, CSS3 and XML files.
- Integrated connection on GIT server.
- Integrated connection to Bugzilla server.
- User friendly.
- Support for team projects.

Functionality of PHP plugin:

- Debugging of PHP code.
- Highlighting of syntactic and semantic code.
- Intelligent sense in pop-up window.

NetBeans IDE was used as the main tool for code editing for the whole developing part of software developing process.

More interesting information about and how to use NetBeans IDE can be found in [1].

3.1.2 Git

Git is a distributed revision control and source code management (SCM) system. It was designed and developed by Linus Torvalds for Linux kernel development in 2005. It is an open-source project and you can easily integrate it to the NetBeans IDE (Figure 8).

Code management is very important in every team software project, because more than one person can modify code at the same time.

Git needs access to a server for more users. For the needs of this project a service provided by <https://bitbucket.org> was used. This service provides Git services for developing teams of maximally five members for free.

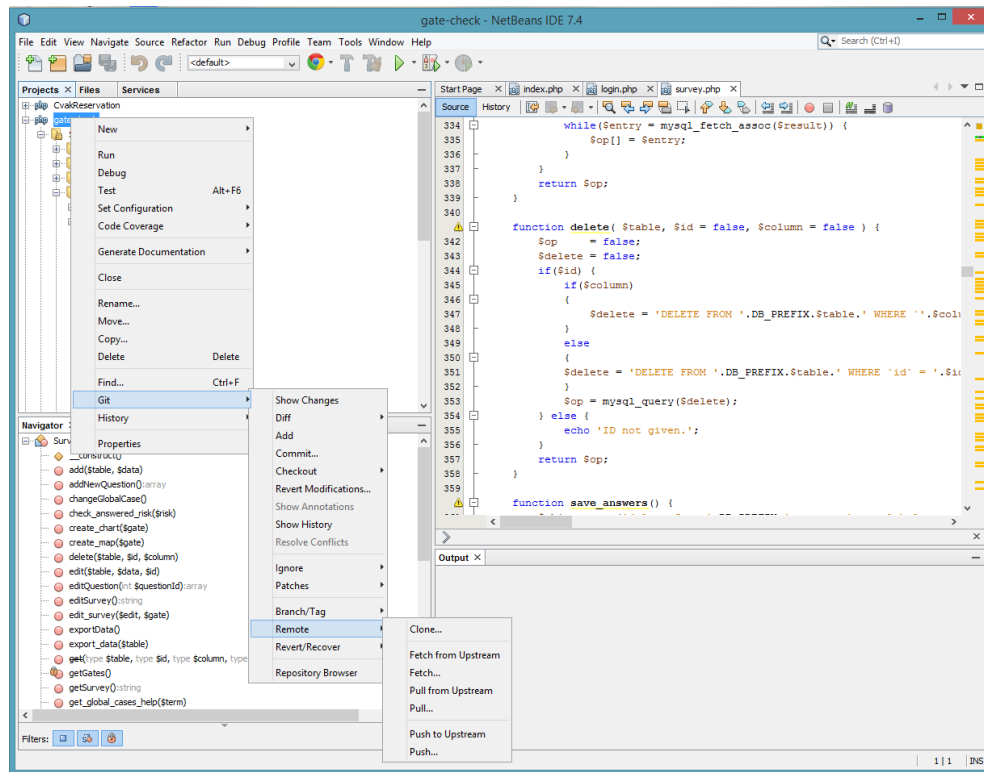


Figure 8 – Git integration in NetBeans IDE 8.0

Git has a local database of changes, so internet connectivity is not needed and you can work offline, if you do not need to download a new version from the server. Another very useful feature are branches. Branches brought a possibility to have a few version of your source code, like a stable version and unstable version where programmers are implementing some new functionality.

Git was used during the whole developing process to maintain and version source the code. Git was often used for solving differences in the same source files which were caused by multiuser editing source. Functionality of branches was used when the first demo version of software for presentation was completed, but at the same time there were developing processes of new functionality to the system.

More interesting information about and how to use Git can be found in [2].

3.1.3 Bugzilla

One part of every software developing process is testing. In every new software there are bugs hidden which are found during testing. It is very important to manage the found bugs for future fixing. There is Bugzilla for this purpose.

Bugzilla is a Web-based general-purpose bugtracker and testing tool. It is an open-source application which can be used in a commercial project for free. For running Bugzilla a web server is needed. Bugzilla can be connected to the NetBeans IDE (Figure 9 – Bugzilla in NetBeans IDE).

During development a service offered by <http://www.bugheaven.com/> was used. Bugzilla account can be created for a team of maximum five members for free.

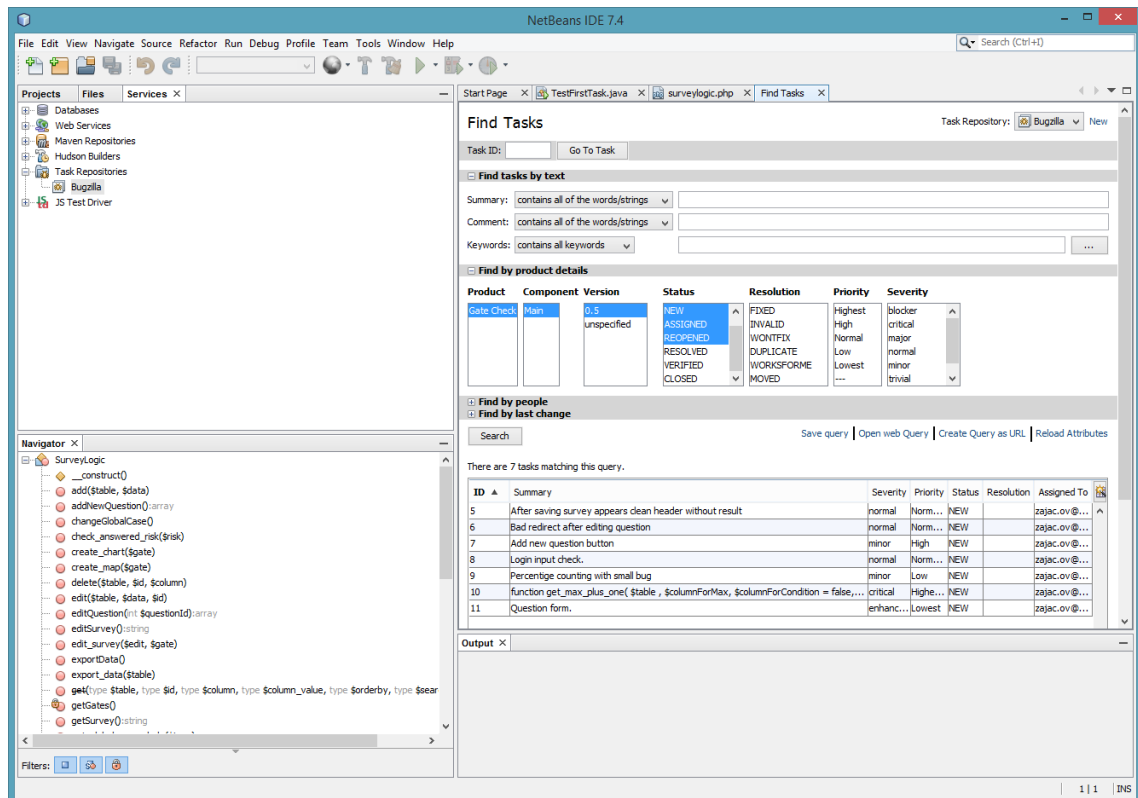


Figure 9 – Bugzilla in NetBeans IDE

Bugzilla was used during the development process in testing phases. When some bug in the application was found, the bug was immediately inserted to the Bugzilla database and later in the phase which was focused on fixing the bugs were fixed.

More interesting information about and how to use Bugzilla can be found in [3].

3.1.4 MySQL Workbench 6.0

Most of the web applications use SQL (Structured Query Language) database. Currently there are many user friendly applications for designing, deploying and maintaining the database. In Gate-Check application MySQL database is used which provides to users an application called MySQL Workbench (Figure 10).

In MySQL Workbench the user can design a database in user-friendly graphic mode like a classic ER diagram, then connect to the database server and create there a real database based on ER diagram. In the new database the user can easily add data or make changes in the database structure.

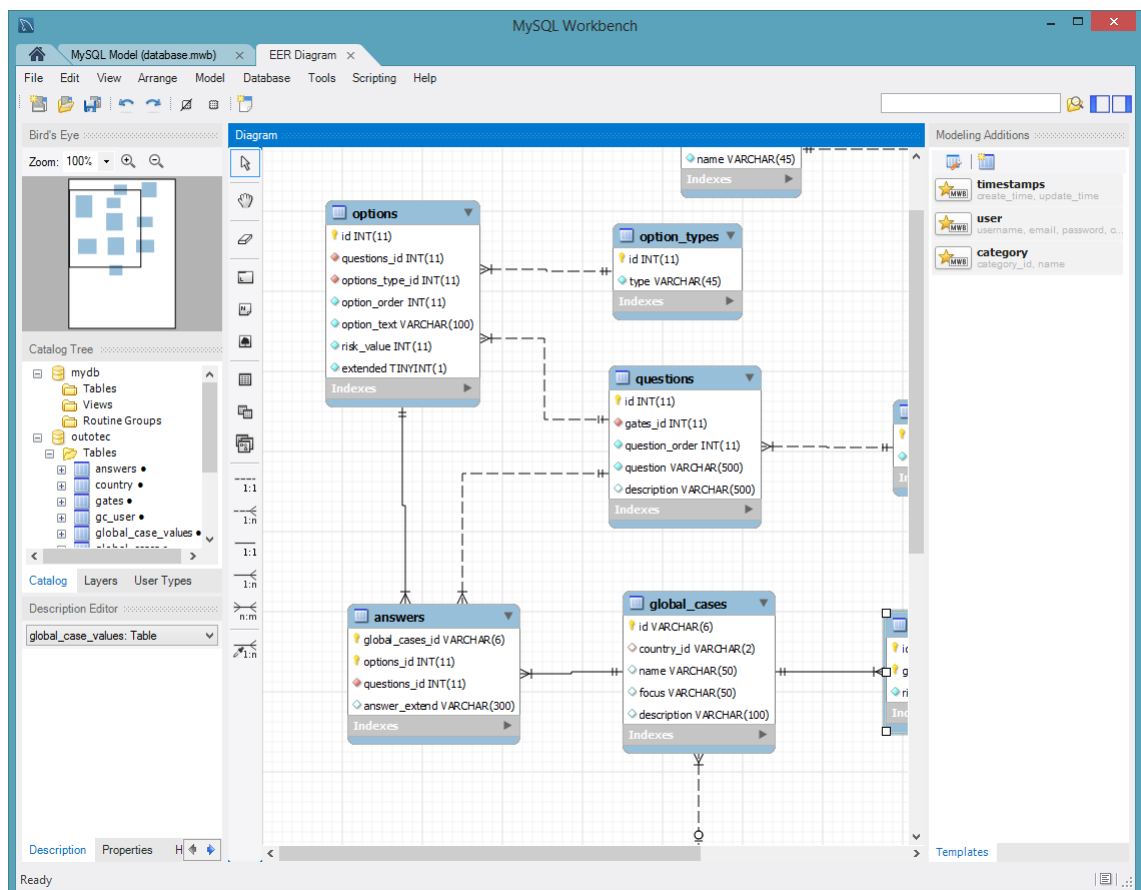


Figure 10 – MySQL Workbench 6.0

One interesting feature is reverse engineering, you can get the existing database on server to the ER diagram, then make changes in the structure and finally apply the changes by forward engineering on the existing database on the server.

In this software development process MySQL Workbench was used for designing, deploying and maintaining during the whole development process.

More interesting information about and how to use MySQL Workbench can be found in [4].

3.1.5 XAMPP server

To develop a web application it is necessary to have an environment for testing. The solution of testing environment is to install Apache server and PHP in the required version and run it on localhost. Or you can install all of this integrated in one application with minimal configuration. This is the advantage of XAMPP server (Figure 11).

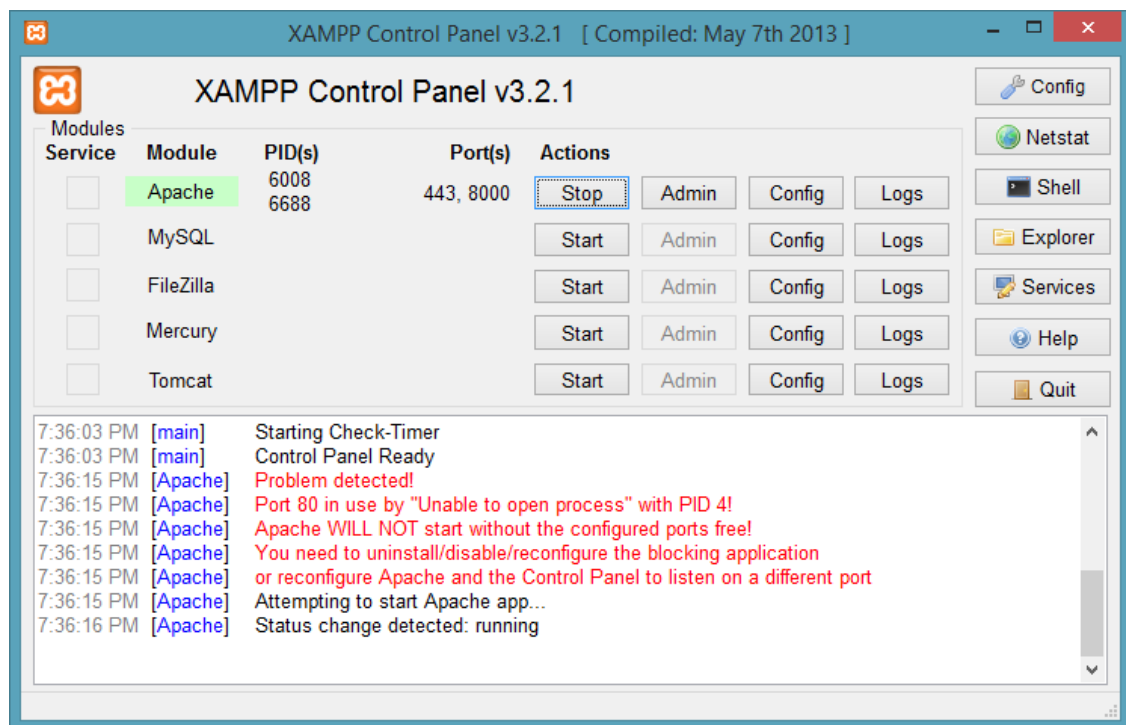


Figure 11 – XAMPP Control Panel

For Gate-Check MySQL database is required. XAMPP can provide installation of MySQL database on localhost, but the Gate-Check application was developed in the team, so it was needed to have a database accessible to each member of the team. The solution was to use an online database service on www.db4free.com, this service gives you free database for testing purposes.

More interesting information about XAMPP server can be found in [5].

3.1.6 HelpNDoc

The main purpose of HelpNDoc is creating manuals for applications. It allows the writer to create a single source text which can then be converted to a number of target formats such as HTML Help, PDF, RTF, HTML documentations as well as EPUB and Amazon Kindle compatible E-books.

HelpNDoc provides a full-featured WYSIWYG editor which aims at looking like a popular word processing software such as Microsoft Word or OpenOffice.org Writer (Figure 12).

The major focus of HelpNDoc is the ease of use for the end user. This is why most of the applications are available through the main window.

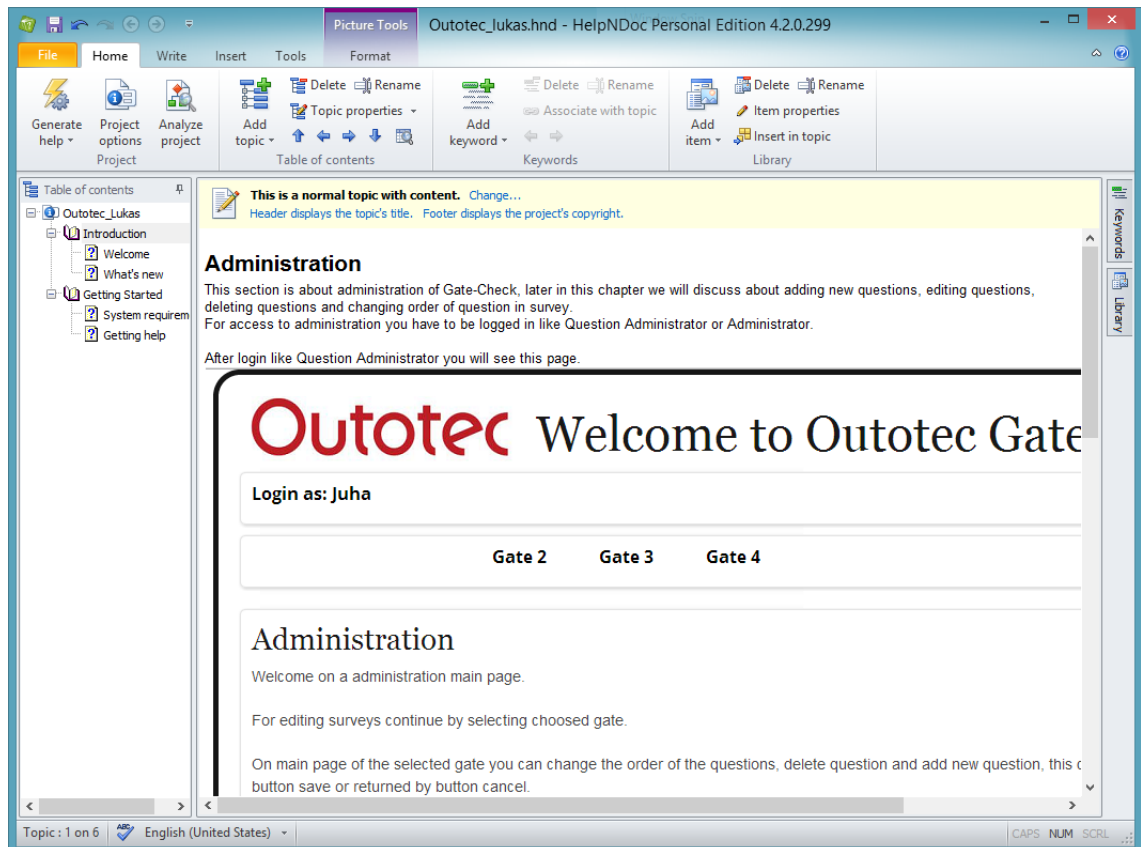


Figure 12 – HelpNDoc

One of the Outotec requirements was to write a short manual for Question Administrator and this manual was created with support of this application.

More interesting information about and how to use HelpNDoc can be found in [14].

3.2 Technologies

This section is about the technologies, which were used to create the application. The right choice of technologies is another important part of software development. If wrong technologies are used, you can later find limitations, which can make it impossible to realize the project according to customer's requirements or it can be more difficult.

3.2.1 PHP

PHP is a server-side scripting language designed basically for creating web applications, but it can also be used for normal applications.

PHP code is interpreted on a web server which has a PHP processor module. The module executes PHP commands and generates a webpage. PHP commands can be written directly into an HTML document or to the external file.

PHP is free and can be used on every computer without any charge.

Every PHP code which should be executed by a server has to be placed in `<?php source code ?>` as is shown in Source code 1.

```
<!DOCTYPE html>
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <?php echo '<p>Hello World</p>'; ?>
  </body>
</html>
```

Source code 1 – PHP Hello World!

The previous example shows a very simple PHP code, which is in the content of HTML document.

More interesting information about and how to use PHP can be found in [6].

Dibi

For PHP many frameworks are available, which can save programmer's time. This project use the Czech framework Dibi created as part of a bigger open source framework called Nette. Dibi mediates database connection between by adding a layer between SQL database and the application. Dibi supports many manufacturers of SQL databases such as MySQL, ODBC, PostgreSQL, SQLite, MS SQL, Oracle and in the application database can be changed without any changes in the source code.

```
// connect to database
dibi::connect(array(
    'driver' => 'mysql',
    'host'   => 'localhost',
    'username' => 'root',
    'password' => '****',
));
// Insert, update and select
$text = "I'm fine";
dibi::query('UPDATE [table] SET [text] = %s', $text);
// MySQL: UPDATE `table` SET `text`='I\'m fine'
// ODBC:  UPDATE [table] SET [text]='I\'m fine'

$arr = array(
    'name' => 'John',
    'is_admin' => TRUE,
);
dibi::query('UPDATE `table` SET ', $arr, 'WHERE `id`=%i', '123');
// UPDATE `table` SET `name`='John', `is_admin`=1 WHERE `id` = 123

// getting results
$result = dibi::query('SELECT * FROM `table`');

$value = $result->fetchSingle(); // single value
$all = $result->fetchAll(); // all rows
$assoc = $result->fetchAssoc('id'); // all rows as associative array
$pairs = $result->fetchPairs('customerID', 'name'); // all rows as key
=> value pairs
```

Source code 2 – Dibi example

Source code 2 shows how easily you can make a database connection and queries as select, insert or update with Dibi in your PHP project. Database connection is made by calling static function *connect*. In next part of the Source code 2 is in

comment also shown difference of query output for different databases and possibility of calling Dibi's function *query* with array or list of objects as parameter. The next part is about processing the results. Dibi provides a few types of fetch functions, every function is useable in different cases. If a right fetch function is chosen it can save a lot of time.

More information about Dibi usage can be found in chapter 4.1 and more about how to use Dibi framework can be found in [7].

3.2.2 HTML5

HTML5 is a markup language used for creating and presenting content for Web presentations and is the main part of the core technology of the Internet. The first revision of the standard was published in 1990. HTML5 is still developed and in these days it is actually the fifth revision. The W3C organization is responsible for the HTML standardization.

The fifth revision brings new functionalities, like video, audio, canvas and many new tags, which make multimedia more available for a mobile device, instead of using flash, which is very performance consuming and on mobile devices unusable.

HTML5 is basically a classic XML with special markup tags. Every tag has to have a start tag "`<html>`" and an end tag "`</html>`". A basic example of an HTML5 document is shown in Source code 3.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Title of the document</title>
  </head>
  <body>
    Content of the document.....
  </body>
</html>
```

Source code 3 – Basic HTML5 code

`<!DOCTYPE html>` – Specified HTML5 version.

`<html></html>` – Determines the start and the end of the document.

`<head></head>` – Elements inside `<head>` can include scripts, instruct the browser where to find style sheets, provide meta information, and more.

`<body></body>` – The main content of the webpage.

When webpages are completed it is good to make HTML validation on a validator, which is provided by W3C organization for free. Validator can be found on <http://validator.w3.org/>.

More interesting information about and how to use HTML5 can be found in [8].

3.2.3 CSS3

The CSS is a language used for designing a look and formatting a document created in markup language. It is often used for styling webpages created in HTML, but it can be used also for styling a classic XML document.

The primary purpose of the CSS is to separate the content of document from document presentation. This separation can provide more flexibility and control in maintaining the presentation.

The CSS also provides methods of how to present one document in different ways (on screen, for print, for voice presentation). It also allows to display differences depending on the given resolution of screen.

The CSS has a priority rule, which is applied when more than one rule is applied on the same element. With this rule the results are always predictable.

The CSS is maintained by the same authority as HTML, W3C. W3C also provides validator for CSS files, which you can find on <http://jigsaw.w3.org/css-validator/>.

```
h3 {  
    color: red;  
    text-align: left;  
    font-size: 8pt;  
}
```

Source code 4 – Basic CSS file

A basic structure of CSS file is shown in source code 4. Tag *H3* specified area in which are used rules written in braces. Every rule has to be ended by semicolon. Colour specified in this case specified the colour of text, text-align specifies the position of text and font size sets the font size.

The CSS can be written into the head of an HTML document and directly into tags. If CSS is used as an external file, it has to be linked in the head of a HTML document.

More interesting information about and how to use CSS can be found in [9].

The project used CSS template Skeleton. It is a small collection of CSS files that can help in developing webpages that can be used at any screen resolution. The page looks user-friendly on a 17-inch laptop screen and on small mobile phone. Both are using the same CSS.

More interesting information about and how to use Skeleton can be found in [15].

3.2.4 JavaScript

The JavaScript is a multiplatform object oriented programming language. These days it is mainly used like an interpreted programming language for creating websites. Usually it is used for interactive GUI elements and effects.

Syntax is close to C/C++ and Java. The first version was standardized in 1997 and was developed in Netscape Company.

The JavaScript is executed after downloading a webpage to the client, instead of other interpreted languages like PHP or ASP.NET, which are executed on server before downloading to the client.

There is a very user-friendly library of functions for programmers called JQuery. It is an open-source project, so it can be used free in projects. It provides many functions like validation of inputs, adds special functionality on inputs and so on.

```
<script type="text/javascript">
document.write("Hello World!")
</script>
```

Source code 5 – JavaScript Hello world!

Source code 5 shows a simple "Hello World!" application written in JavaScript. Script tag is used inside the HTML document and the content of script tag is interpreted by JavaScript interpreter.

More interesting information about and how to use JavaScript can be found in [10] and [11]

Other JavaScript modules used in project:

- JVectorMap – World map which is based mainly on JavaScript, CSS and HTML, so it does not need any special plugins.

More interesting information about and how to use JVectorMap can be found in [12]

- CanvasXpress – Standalone module for showing graphs written in JavaScript that includes a simple and unobtrusive user interface to explore complex data sets. It is supported in all major browsers in computers and mobile devices.

More interesting information about and how to use CanvasXpress can be found in [13]

3.3 Development process

On the start of December 2013 an employee from Outotec Company with a new task comes to school. The new task was about creating a new version of Gate-Check system. After getting first information, a team which started working on the task was created.

The team contained two members:

- Petr Bartusek
- Lukáš Zajac

The team was internally called "Saimia". The team leader was Petr Bartusek, but the responsibilities of work were the same for everyone and everyone did everything.

On the first meeting it was discussed which technologies will be used for the new system. The main programming language was selected PHP with support of classic HTML5, CSS3 and JavaScript. For data storage MySQL database was chosen. The technologies were chosen because the Outotec Company is currently running web applications based on these technologies.

The next meeting was one week after the first one. On this meeting the first functionality of the application was introduced.

The customer could not specify all their needs at the second meeting. The customer wanted to show some working parts of application on presentation to target employees to consult what they needed next.

For customer needs the software development was started iteratively. So, the final application was divided into small parts. In one iteration one part was developed, completely tested and shown to the customer. If everything was alright with that part, the next iteration was started in which a new functionality was added to the previous fully working part of the application.

The whole development process of the application had five iteration cycles. There was a meeting and discussion about the implementation of new functions and changes in the existing application after every iteration cycle.

3.3.1 Iterative and incremental development

The basic idea of this method is to develop a system in repeated cycles (iterative) (Figure 13) and in smaller portions at a time (incremental). This allows developers to take advantage of what was learnt during the development of earlier parts or versions of the system. There were many iterations until the full system was implemented. At each iteration, design modifications were made and new functional capabilities were added.

The procedure consists of the initialization step, the iteration step, and the Project Control List. The initialization step creates a base version of the system. The goal is to implement an application, which can be shown to the customer.

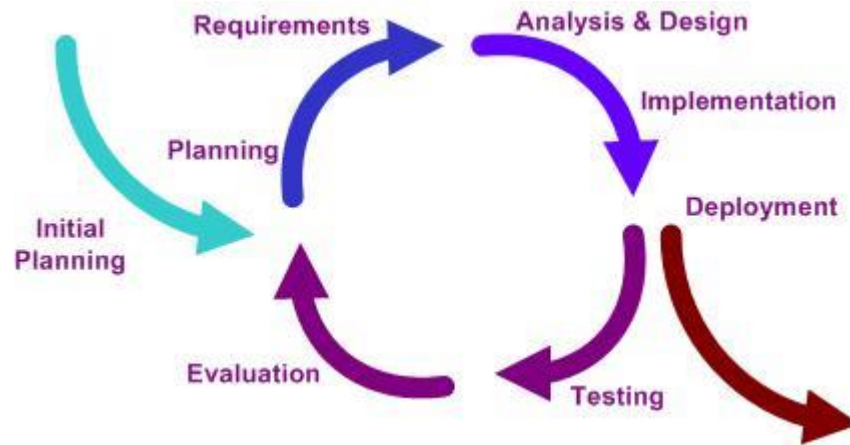


Figure 13 – Iteration cycle

The iteration involving the redesign and implementation of iteration is to be simple, straightforward, and modular, supporting redesign at that stage or as a task. The analysis of an iteration is based upon user feedback, and the program analysis facilities available. It involves analysis of the structure, modularity, usability, reliability, efficiency and achievement of goals.

More interesting information about and how to use Iterative development can be found in [16].

4 Gate-Check application

In modern web pages there is a big gap between information technology (Database layer and Business logic or together also back-end) and visual communications (User interface or also front-end). It requires data for end-user interactivity, dynamic page display, database publishing and etc. Most of the data must be specified and later coded during the page layout process. This divides the work on web pages between artists and engineers, or the artist/engineer to do both.

This chapter describes the final application and its distribution into three subchapters based on the architecture of the program. The following subchapters describe database layer, business logic and user interface basic ideas and advance features.

4.1 Database Layer

The first and the lowest layer of application is database layer. This layer mediates connection between MySQL database and application business logic. It is implemented using Dibi framework concisely described in chapter 3.2.1.

```
function dibi_get($table, $columns = false, $orderby = false,
$search_col = false) {
    $query[] = '';
    if ($search_col) {
        array_push($query, 'SELECT %n', $search_col);
    } else {
        array_push($query, 'SELECT *');
    }
    array_push($query, ' FROM %n', "::$table");
    if ($columns) {
        array_push($query, ' WHERE %and', $columns);
    }
    if ($orderby) {
        array_push($query, ' ORDER BY', $orderby);
    }
    return $result = dibi::query($query)->fetchAll();
}
...
```

Source code 6 – dibi_get() function

One of the important constructed functions on database layer is function *dibi_get* which constructs every simple select query needed by business logic. Query is constructed gradually according to the specified parameters. The function accepts four parameters and three of them are optional.

```
$qts = $this->dibi_get('questions', array('gates_id' => $GateID),  
'question_order');
```

Source code 7 – Using *dibi_get()* function

The source code 7 captures a query that selects all questions from the database from specified Gate sorted by column *question_order* of the whole database structure is captured in Figure 6 and data dictionary is in the first annex on page 58.

4.2 User Interface

UI (User interface) is a gate for interaction of end-user of application with computer. It is very important to design a user friendly interface, because a lot of end users are not very computer skilled. The end user does not see what is inside the system and he probably does not care about it. The user only needs to do his work as effectively as possible.

For the previous reasons it is very important to spend a lot of time to design the UI and consult your design many times with end users. The chapters below show the UI design of Gate-Check system.

4.2.1 Siebel connection

Users which are connected to the Gate-Check application through Siebel get the minimal version of UI (Figure 14), which is supported on mobile devices.



Figure 14 – Siebel UI

In the top right corner an information banner with global case number (GID), gate number and service area number is placed.

If the user puts the cursor on question or on option in tooltip window (**Chyba! enalezen zdroj odkazů.**), detailed information about question or risk value of option is shown.



Figure 15 - Tooltip window

On the bottom of the page there is button *Submit*, which submits the survey, puts results in the Gate-Check database and sends the final risk to the Siebel database and redirects the user to the history page. Button *Save as draft* only puts the results to the Gate-Check and the user can continue filling in later. The user stays on the same page and the success of saving is informed in notification (Figure 16). Button *Cancel* clears all changes made in the actual session.

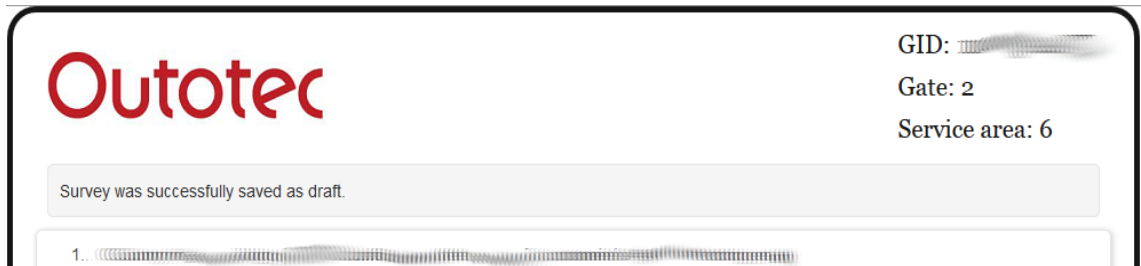


Figure 16 – Saved as draft

If the user chooses an option which has extended answer or own risk value, the user has to answer this question, because there inputs are mandatory. If the user leaves it empty and presses button *Submit* or *Save as draft*, the user will be noticed (Figure 17) about this problem.



Figure 17 – Verification

If all mandatory inputs are filled in, the user with pressing button *Submit* is redirected to the history page (Figure 18)

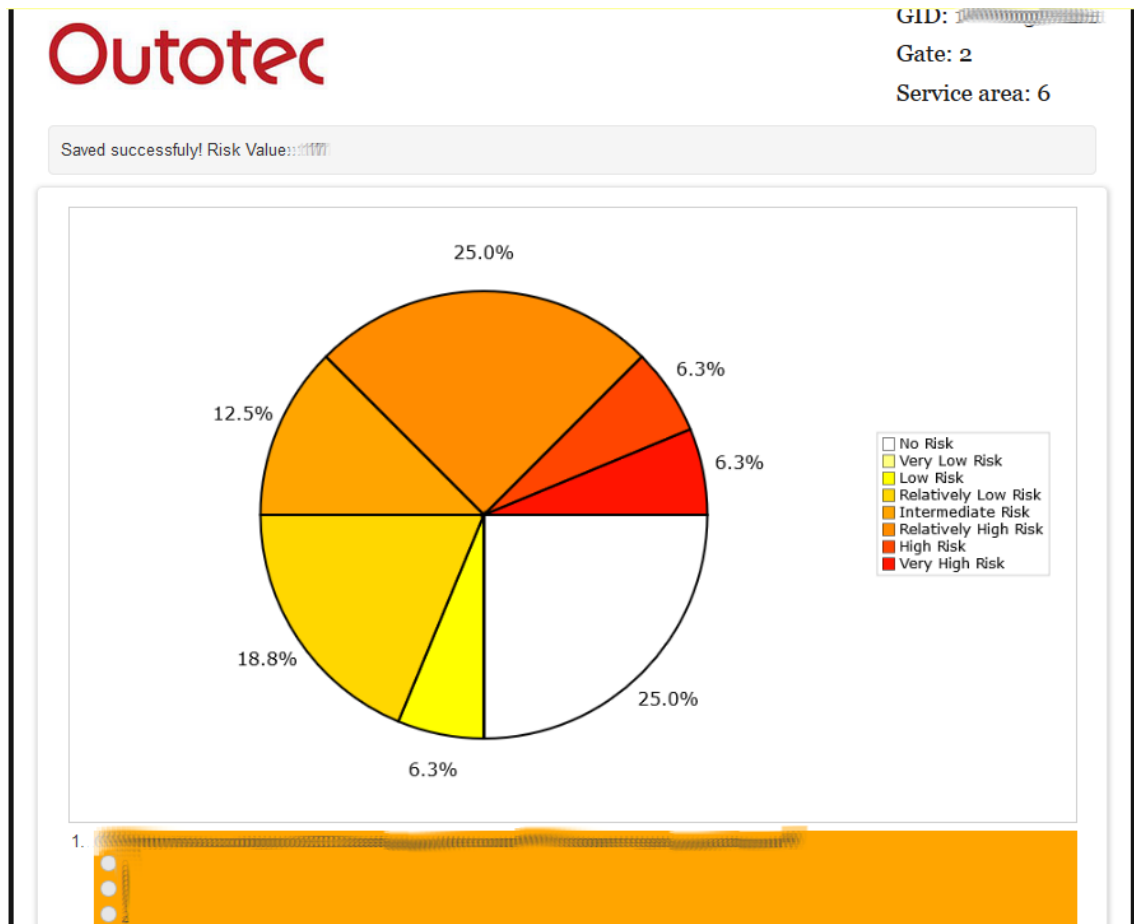


Figure 18 – History UI

There is a notification on the top of the history page about the result of risk. If more than 70% of questions are answered with the highest risk there is also information about that.

Next on the history page there is a graph, which shows how many answers are in the specified risk layer.

The whole survey with answers is under the graph. Every question has background colour based on the risk layer of the answer.

If the user wants to make some changes in answers, the user can use button *Edit answers* on the bottom of the page (Figure 19).



Figure 19 – Edit answers

These are all functions, which users accessing to the Gate-Check system through the IS Siebel have. This is only a small part of functions which are offered by Gate-Check, other functions for like administration of surveys, viewing history, are accessible only from the main Gate-Check application.

4.2.2 Main application

If the user wants to use the main application, at first the welcome page is shown (Figure 20).



Figure 20 – Welcome page

The user should click on login to get login form (Figure 21).

Figure 21 – Login form

After successful login the user is redirected to the main page. The content of menu depends on user privileges. If the user acts as *Question administrator* the content of menu is limited (Figure 22), if the user acts as *Administrator* he has access to all functions (Figure 23).

Figure 22 – Question administrator menu

Figure 23 – Administrator menu

The top menu on the left shows the username of user which is logged in, on the right side there is a button for logout. *Question administrator* does not have any other buttons in the top menu, but *Administrator* has four choices:

Map – Display page with map (Figure 26), in which the average risk value for each country is shown.

Before use of the rest of the top menu buttons, the user should set global case number. This can be done by clicking on *Change GS ID* button on the right side of the bottom menu. The user will be redirected to a page with a form for inserting global case number. If the user does not make this step and clicks on some of

the rest of the menu buttons he will be redirected to the page with a form for inserting global case number too.

In a simple form for inserting global case number (Figure 24), the user can insert only global case number which is stored in Gate-Check database. The form uses intelligent sense showing the possible global case numbers.



Figure 24 – Set global case id form

If the user has set global case number or not is indicated in the bottom menu on the left side (Figure 24). This function is described in more detail in chapter 4.3.2.

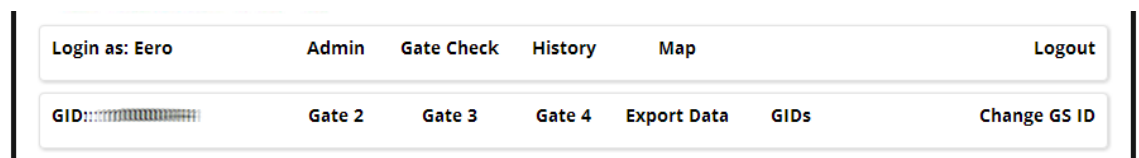


Figure 25 – Menu with global case number

History – Changes functionality of buttons (*Gate 2*, *Gate 3* and *Gate 4*) in the bottom menu and shows default history page.

Gate Check – Changes functionality of buttons (*Gate 2*, *Gate 3* and *Gate 4*) in the bottom menu and shows default Gate-Check page.

Admin – Changes functionality of buttons (*Gate 2*, *Gate 3* and *Gate 4*) in the bottom menu and shows default *Admin* page.

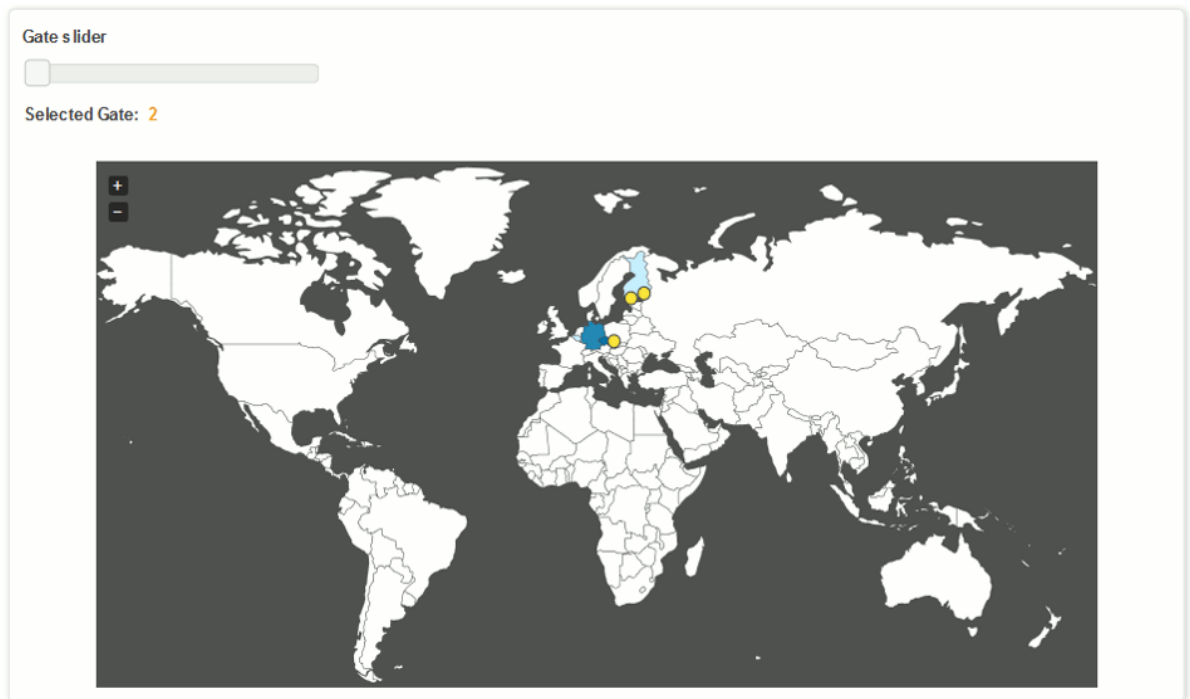


Figure 26 – Risk map

The bottom menu contains *Gate 2*, *Gate 3* and *Gate 4* button for *Question Administrator* and *Administrator* and only for *Administrator* contains *Export Data* and *GIDs* buttons.

Export Data – Downloads a CSV file which contains all records of results in Gate-Check database.

GIDs – Shows to the user a table which contains all global cases with the results of risk (Figure 27). If the user clicks on global case number, the user will be redirected to the default history page, if he clicks on risk value the user will be redirected to the history page about the selected gate, if there is answer instead of risk value, it means that the survey for this global case and gate was not filled in yet, clicking on the answer the user is redirected to a page where he can fill in the survey. These pages are discussed later in the following text.

Login as: Eero	Admin	Gate Check	History	Map	Logout	
GID: [redacted]	Gate 2	Gate 3	Gate 4	Export Data	GIDs	Change GS ID
GID	Gate 2	Gate 3	Gate 4			
[redacted]	142	8	20			
[redacted]	13	answer	answer			
[redacted]	58	answer	answer			
[redacted]	13	answer	67			
[redacted]	answer	answer	30			

Outotec Gate Check

Figure 27 – GIDs table

4.2.3 History

As was written earlier, only *Administrator* has access to this part. If *History* button was selected in the top menu and previously global case number was set, the user was redirected to default *History page* (Figure 28).

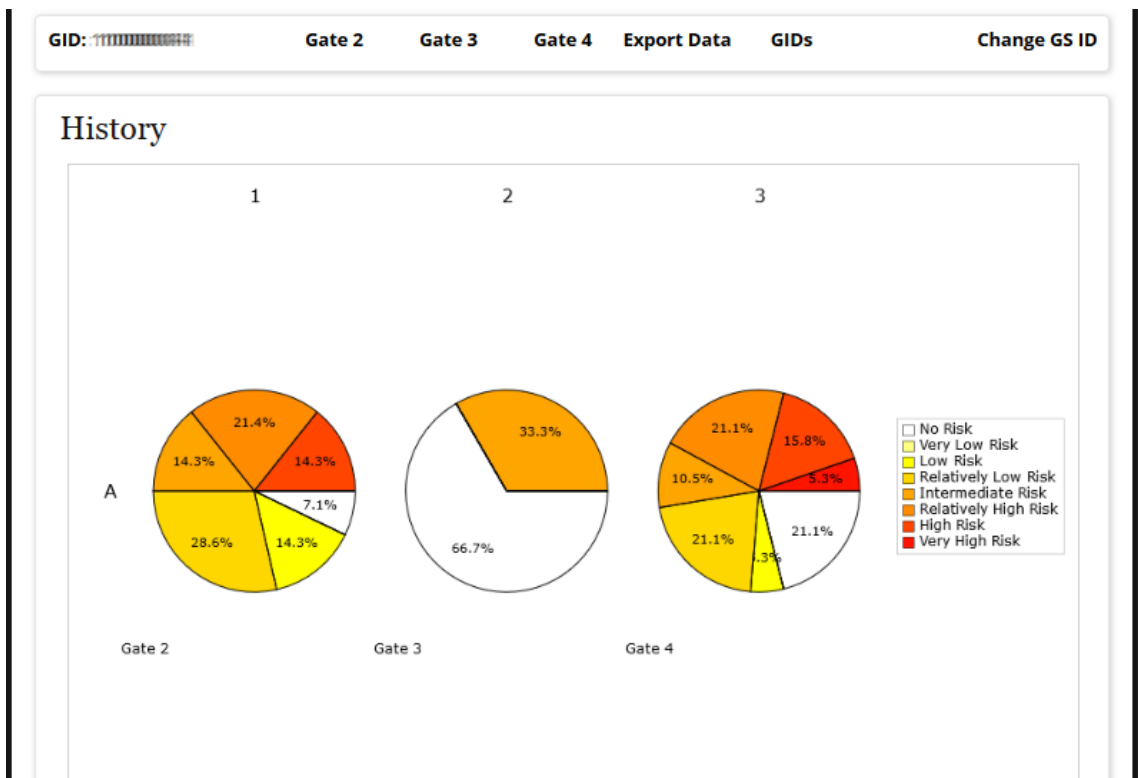


Figure 28 – Default history page

Graphs for every gate on default *History page* are shown. If the user selects *Gate 2*, *Gate 3* or *Gate 4* from the bottom menu, the application is redirected on history page for the selected gate (Figure 29).

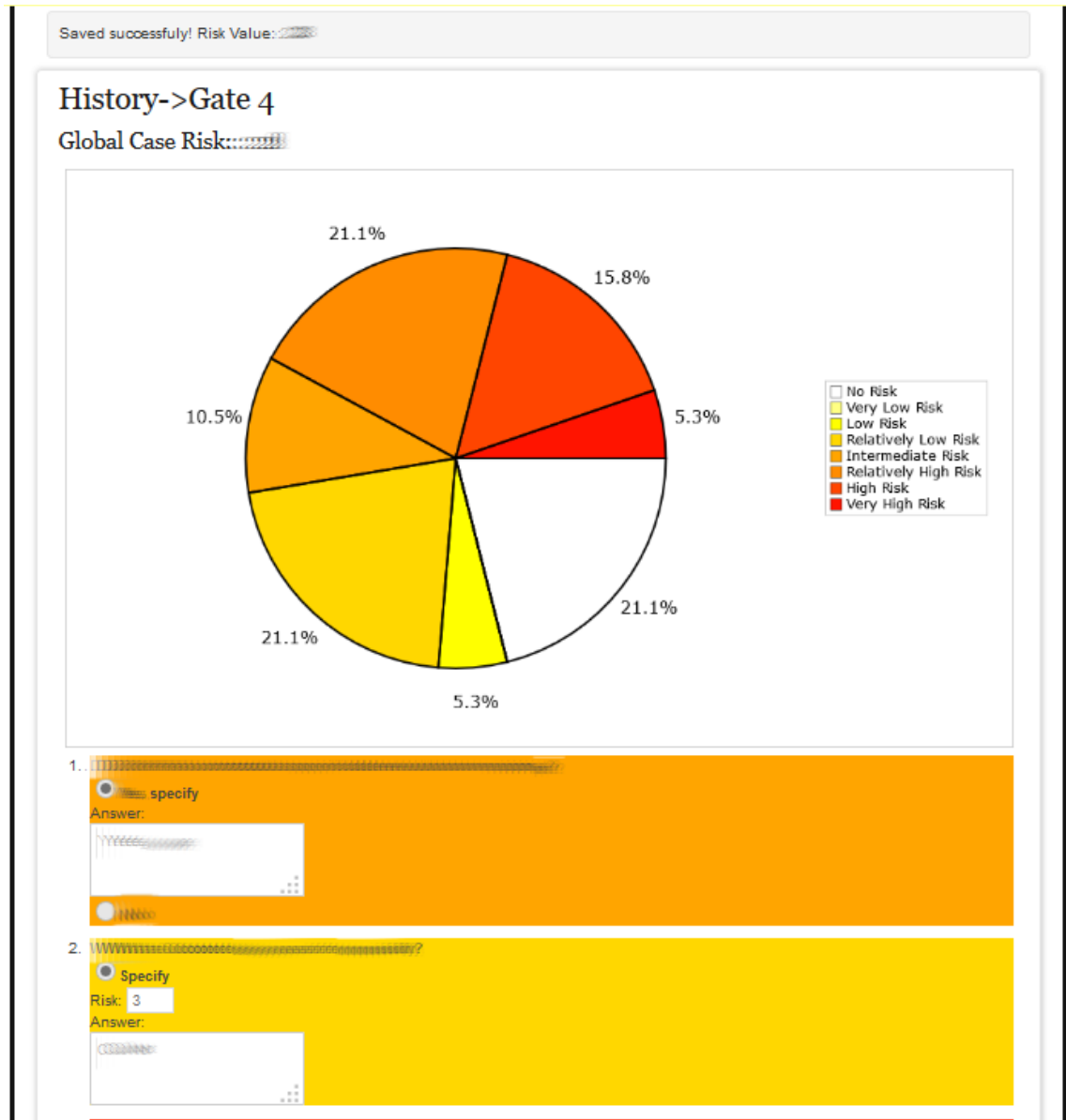


Figure 29 - History page

As it is shown in Figure 29, *History page* in main application is very similar to *History page* described in chapter 4.2.1. If the user selects *History page* for gate with no answers yet, the user is automatically redirected to a page for filling in surveys.

4.2.4 Gate Check

As it was already written earlier, only *Administrator* has access to this part if *Gate Check* button in the top menu was selected and global case number was previously set, the user was redirected on default *Gate Check page* (Figure 30). *Gate Check* page is for filling surveys. If the survey was answered previously it is shown prefilled with the previous answers and the user can make changes.



Figure 30 – Gate Check default page

If the user selects *Gate 2*, *Gate 3* or *Gate 4* from the bottom menu, the application is redirected to *Gate Check page* for the selected gate (Figure 31).

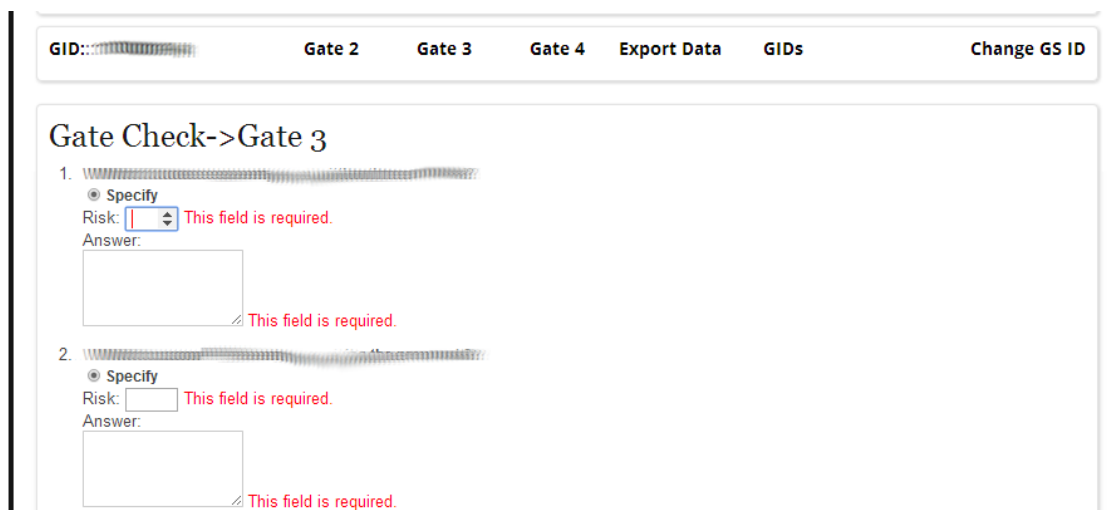


Figure 31 - Gate Check

As it is shown on Figure 31, *Gate Check page* in main application is very similar to *Gate Check page* described in chapter 4.2.1. After saving the survey the user is redirected to *History page* (Figure 29). The functionality is the same as it was described in chapter 4.2.1.

4.2.5 Administration

As it was already written earlier, *Administrator* and *Question administrator* have access to this part. *Administrator* has to select *Admin* button in the top menu which redirects the user on default *Admin page* (Figure 32). *Question administrator* does not do this step, because in login everything is done automatically.



Figure 32 - Administration default page

There is a difference in menus for *Administrator* and *Question administrator*. All figures in administration are taken with menu for *Administrators*. The rest of the functionality of administration is the same for both levels of users.

If the user selects *Gate 2*, *Gate 3* or *Gate 4* from the bottom menu, the application is redirected to *Administration page* for the selected gate (Figure 33).

Outotec Welcome to Outotec Gate Check

Login as: Juha

Logout

Gate 2 Gate 3 Gate 4

Add new question

Administration->Gate 2

1. 






































On the main page of administration (Figure 33) for the gate the user can delete a question by clicking on *Delete* button and change the order of questions in survey by dragging the question on the grey rectangle and moving it to the new position (Figure 34).

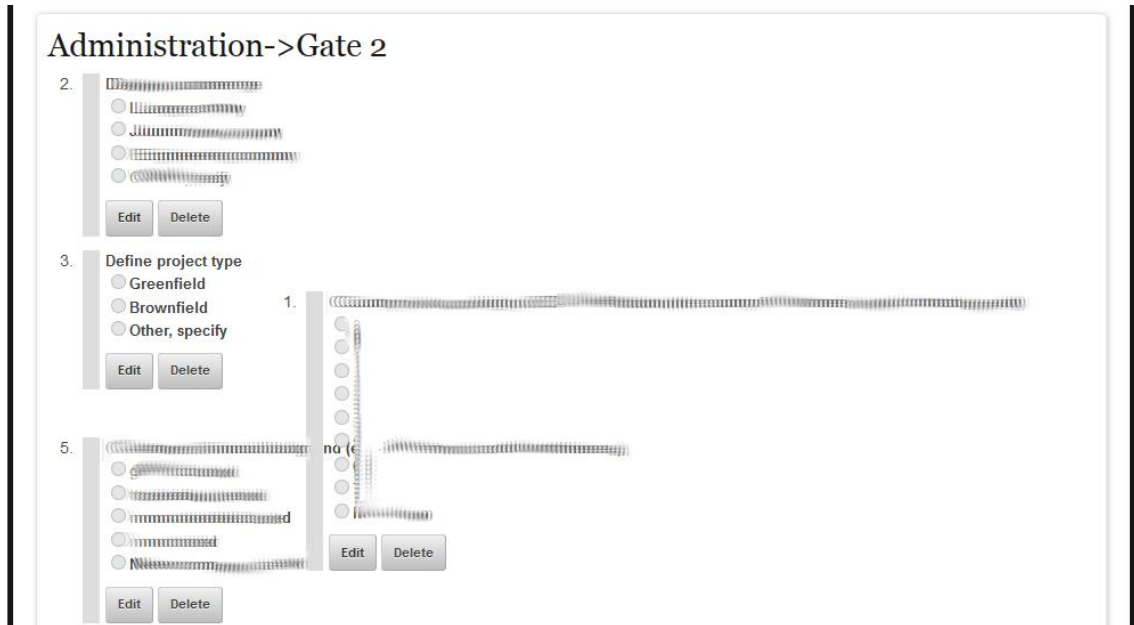


Figure 34 - Drag and Drop

After all these changes the user has to use button *Save changes* for saving changes or use *Cancel* button to reload the old version of survey.

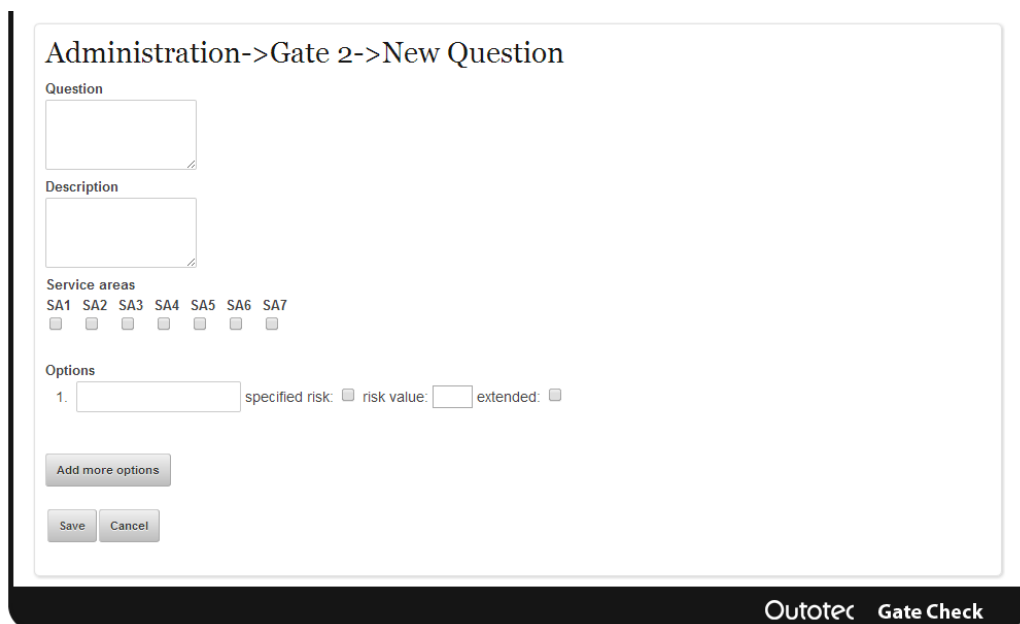


Figure 35 - Add new question

For editing the existing question the user has to use button *Edit*, if the user wants to add a new question, he has to use *Add new question* button.

In both cases the user gets the same form, but it is empty for a new question (Figure 35) and for edit question (Figure 36) is filled in with question data.

The screenshot shows a web form titled "Administration->Gate 4->Edit Question". It contains several sections:

- Question:** A text area with a scroll bar.
- Description:** A text area with a scroll bar.
- Service areas:** A row of checkboxes labeled SA1 through SA7. SA4, SA5, SA6, and SA7 are checked.
- Options:** A list of three options. Each option has a text input field, a "specified risk" checkbox, a "risk value" input field, an "extended" checkbox, and a "-----" button.
 - Option 1: "specified risk" is unchecked, "risk value" is 20, "extended" is checked.
 - Option 2: "specified risk" is checked, "extended" is checked.
 - Option 3: "This field is required." is shown in red. "specified risk" is unchecked, "risk value" is RTM (highlighted in red), "Please enter a valid number." is shown in red, "extended" is unchecked.

At the bottom of the form are buttons for "Add more options", "Save", and "Cancel". The footer of the page reads "Outotec Gate Check".

Figure 36 - Edit question

The user can add a new option by using *Add more options* button and if the user wants to remove some option, the user can use ----- button next to the option which the user wants to remove.

The new question is added to the survey or changes in the old question applied by button *Save* and the user will be redirected to the administration of gate page.

If the user does not want to add or save question, simply use *Cancel* button and the user will be redirected to the administration of gate page.

4.3 Business Logic

Business logic is such a part of the program that encodes the real-world regularities that determine how data can be displayed, stored or changed. Gate-Check application contains several logics described below.

4.3.1 Login

One of the main requirements of the application was also make to it as secure as possible. This definitely includes login on web pages. So the application was made with great emphasis on this element.

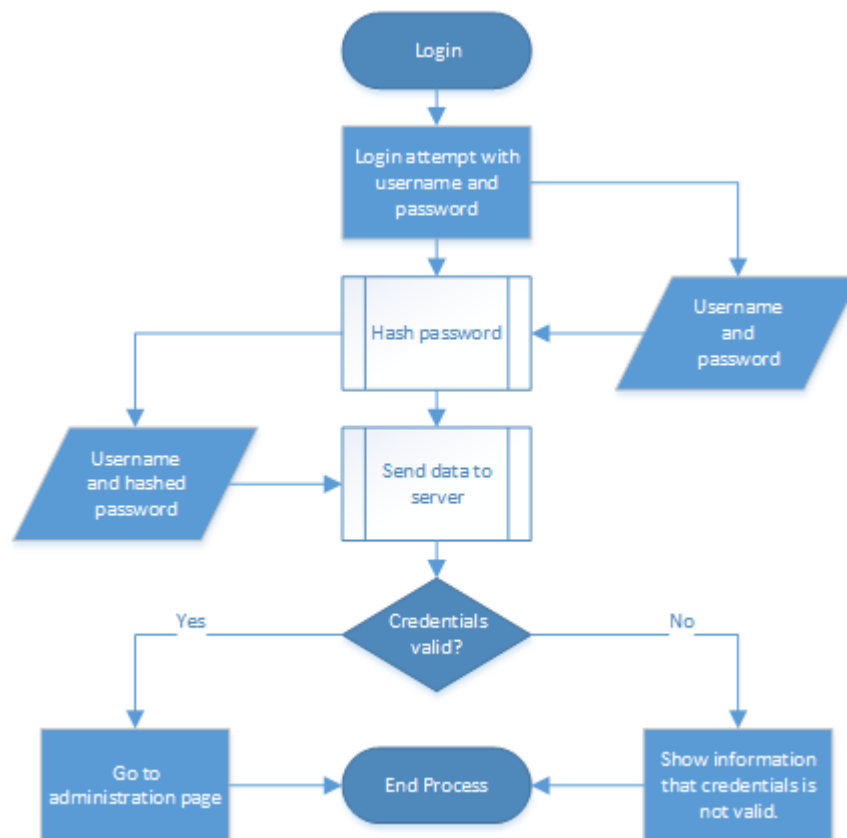


Figure 37 – Login workflow

As it is captured in Figure 37 the password is after filling in and before sending to the server hashed on the client side in the user browser by JavaScript function. This procedure ensures that the password is not sent in plain text to server. If the password was not hashed the attacker would be able to obtain it by a simple

analysis of network flow for example with open source tool Wireshark². The password is hashed by strong SHA1 algorithm which currently guarantees indecipherability against brute-force attack in normal time even on supercomputer³. Hash is also made with the help of cryptographic salt. The function of cryptographic salt is to defend the password against dictionary attacks or pre-computed rainbow table. This approach makes the password even more secure. JavaScript library SHA1.js by Greg Holt distributed under BSD license⁴ is used for hashing purposes.

4.3.2 Intelligent GID completion

Gate-Check system contains some fields concerning details such as GID and Case ID. Some of these details are already stored in the database. This enables to use this data for suggesting possible text values based on database content (Figure 24). The selected data is filled in to the corresponding fields. This feature is called intelligent code completion and the most cases makes web pages more user-friendly, speeds up the filling process and helps to prevent mistakes.

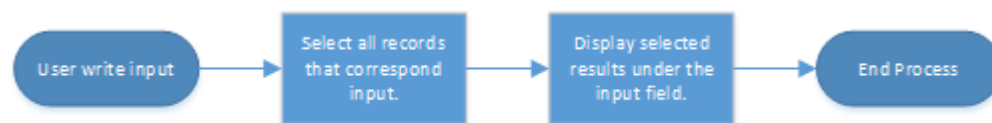


Figure 38 – Intelligent GID completion workflow

As it is captured on Figure 38, JavaScript calls function immediately after the user writes input to the web page field. This call performs a selection for all records that correspond with input. After the selection JavaScript displays a result under the input field.

4.3.3 Inputs validation

Good programming techniques also include not relying on the automatic support of web technologies on the client side. For this reason, dual data validation was

² More about Wireshark at <http://en.wikipedia.org/wiki/Wireshark> or <http://www.wireshark.org/>

³ More at <http://en.wikipedia.org/wiki/SHA-1>

⁴ Available from <http://pajhome.org.uk/crypt/md5/sha1.html>

implemented to ensure the correct format of the entered data. The first data validation is made on the client side by JavaScript inside user web browser and this validation is more user-friendly. The second data validation is made on the server side by PHP server and needs a web page to reload which makes it less user-friendly. JavaScript validation is captured on Figure 31 and Figure 36 as red text which indicates a wrong entry and describes where the mistake has occurred.

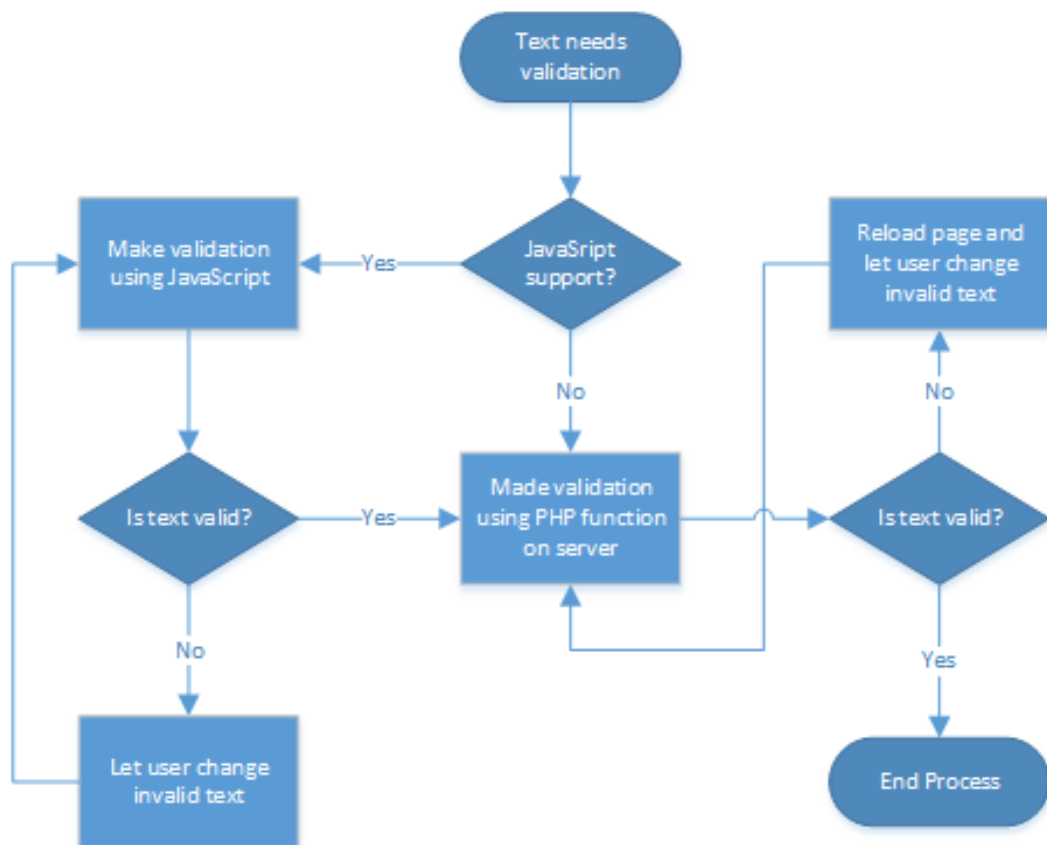


Figure 39 - Validation workflow

So as it is captured on Figure 39 validation is made both on client side (JavaScript) and server side (PHP server), but since both validators are testing the same requirements if the client side validator decides that text is valid, the server side validator should not reload the page with more errors. The main reason of this dual data validation is the possibility that the user's browser does not support JavaScript.

4.3.4 Risk calculation

One of the key features of Gate-Check system is calculation of risk for specified GID and possibility for graphical display. Risk calculation is defined as amount of risk in all questions. Every question can have a hard-coded risk value selected during the creation of question or the user has to specify the risk during replying on survey. One way or another, the final risk has to count both values.

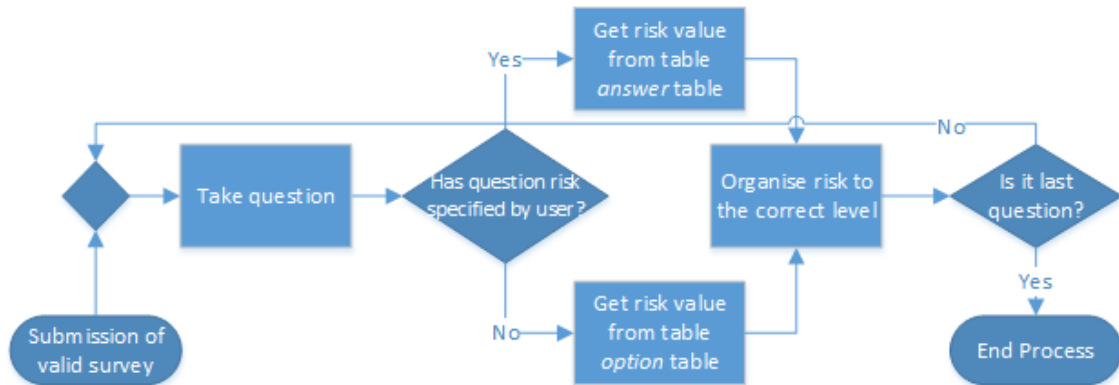


Figure 40 – Risk calculation

The final risk is stored in the database. Besides the final risk, Gate-Check also organises risk to levels that are configurable by the administrator and displays these levels in history in three ways.

- As colourful chart. (Figure 28).
- As tingle of colour question (Figure 29).
- As tingle country on world map (Figure 26).

Chart is formed by the CanvasXpress library described in chapter 3.2.4 and tingle of colour is made using CSS style. Risk data is also used for the display of average country risk values on word map. Slider on the top of Figure 26 will change risk values for selected *Gate* and markers on map show the location of company with its risk value. Map is formed by the JVectorMap library described in chapter 3.2.4. Because the process of making these things is too complicated in excess of the scope of this thesis, it will not be described.

4.3.5 Other features

The Gate-Check also includes a function to export data to CSV (coma separated values) format. Data from the database is pre-processed and saved in standard format to the file through web browser.

The last but not least function of Gate-Check system mentioned in this document is the function for sending emails to Outotec risk manager if 70% of questions has the highest risk level. This feature is coded using PHPMailer library⁵ and it is configurable from configuration file.

⁵ PHPMailer is a code library to send emails safely and easily via PHP codes from a webserver. Framework available from <http://wiki.dreamhost.com/PHPMailer>

5 Conclusion

During our work period in Outotec Company we developed a Gate-Check web application. The application meets all customer requirements and can be deployed to the end user. The new application is a full replacement of the old version and brings many new features.

In the development process we learnt and got deeply into the web development. We used many new technologies and features for us like JavaScript, PHP, CSS3 and HTML5. Also we learned how to develop a responsible layout design.

Also we got new experience in team work and corporate environment. We were met with business processes in very large companies and this is very good experience to the real life and future job. Regular meetings taught us many social and presentations skills.

Our work period in Outotec was a very exciting experience and gave us a lot of information about processes and workflows in the company.

List of images

Figure 1 – Old Gate-Check system.....	10
Figure 2 – Use case diagram.....	12
Figure 3 – Connection through Siebel.....	14
Figure 4 – Connection directly to Gate-Check application	15
Figure 5 – Architecture.....	15
Figure 6 – Database structure.....	16
Figure 7 – NetBeans IDE 7.4	17
Figure 8 – Git integration in NetBeans IDE 7.4	19
Figure 9 – Bugzilla in NetBeans IDE.....	20
Figure 10 – MySQL Workbench 6.0.....	21
Figure 11 – XAMPP Control Panel.....	22
Figure 12 – HelpNDoc	23
Figure 13 – Iteration cycle.....	31
Figure 14 – Siebel UI	34
Figure 15 - Tooltip window	34
Figure 16 – Saved as draft.....	35
Figure 17 – Verification	35
Figure 18 – History UI	36
Figure 19 – Edit answers	37
Figure 20 – Welcome page.....	37
Figure 21 – Login form.....	38
Figure 22 – Question administrator menu	38
Figure 23 – Administrator menu.....	38
Figure 24 – Set global case id form	39
Figure 25 – Menu with global case number	39
Figure 26 – Risk map.....	40
Figure 27 – GIDs table.....	41
Figure 28 – Default history page	41
Figure 29 - History page	42
Figure 30 – Gate Check default page	43
Figure 31 - Gate Check.....	43
Figure 32 - Administration default page	44

Figure 33 - Administration of gate	45
Figure 34 - Drag and Drop	46
Figure 35 - Add new question	46
Figure 36 - Edit question	47
Figure 37 – Login workflow	48
Figure 38 – Intelligent GID completion workflow	49
Figure 39 - Validation workflow	50
Figure 40 – Risk calculation	51

List of source codes

Source code 1 – PHP Hello World.....	24
Source code 2 – Dibi example.....	25
Source code 3 – Basic HTML5 code.....	26
Source code 4 – Basic CSS file.....	27
Source code 5 – JavaScript Hello world!	28
Source code 6 – dibi_get() function.....	32
Source code 7 – Using dibi_get() function.....	33

List of references

1. NetBeans: Official website [online]. [cit. 2014-02-13]. Accessed by: <http://netbeans.org/>
2. Git: Official website [online]. [cit. 2014-02-13]. Accessed by: <http://git-scm.com/>
3. Bugzilla: Official website [online]. [cit. 2014-02-13]. Accessed by: <http://www.bugzilla.org/>
4. MySQL Workbench: Official website [online]. [cit. 2014-02-13]. Accessed by: <http://www.mysql.com/products/workbench/>
5. XAMP Server: Official website [online]. [cit. 2014-03-06]. Accessed by: <http://sourceforge.net/projects/xampp/>
6. PHP: Official website [online]. [cit. 2014-03-06]. Accessed by: <http://php.net>
7. Dibi framework: Official website [online]. [cit. 2014-03-06]. Accessed by: <http://dibiphp.com>
8. HTML5: Official website of W3C [online]. [cit. 2014-03-06]. Accessed by: http://www.w3schools.com/html/html5_intro.asp
9. CSS3: Official website of W3C [online]. [cit. 2014-03-06]. Accessed by: http://www.w3schools.com/css/css3_intro.asp
10. JavaScript: tutorial [online]. [cit. 2014-03-07]. Accessed by: <http://www.w3schools.com/js/DEFAULT.asp>
11. JQuery: Official website [online]. [cit. 2014-03-07]. Accessed by: <http://jquery.com/>
12. JVectorMap: Official website [online]. [cit. 2014-03-07]. Accessed by: <http://jvectormap.com/>
13. CanvasXpress: Official website [online]. [cit. 2014-03-07]. Accessed by: <http://canvasxpress.org/>
14. HelpNDoc: Official website [online]. [cit. 2014-03-09]. Accessed by: <http://www.helpndoc.com>
15. Skeleton: Official website [online]. [cit. 2014-03-10]. Accessed by: <http://www.getskelton.com/>
16. Skeleton: Official website [online]. [cit. 2014-03-10]. Accessed by: http://en.wikipedia.org/wiki/Iterative_and_incremental_development

Annex:

Database data dictionary:

gate_check_questions – table contain all questions in Gate-Check system		
id	INT(11)	clear identification of question
gates_id	INT(11)	clear identification of gate in which is question placed
question_order	INT(11)	position of question in gate
question	VARCHAR(500)	text of question
description	VARCHAR(500)	text of question description
sa1	BOOLEAN	Is question part of service area 1?
sa2	BOOLEAN	Is question part of service area 2?
sa3	BOOLEAN	Is question part of service area 3?
sa4	BOOLEAN	Is question part of service area 4?
sa5	BOOLEAN	Is question part of service area 5?
sa6	BOOLEAN	Is question part of service area 6?
sa7	BOOLEAN	Is question part of service area 7?

gate_check_options – table contain all options in Gate-Check system		
id	INT(11)	clear identification of option
questions_id	INT(11)	clear identification of question for which is this option
option_order	INT(11)	position of option in question
option_text	VARCHAR(100)	text of option
risk_value	INT(11)	risk value of option
extended	BOOLEAN	Is option with extended answer?
specified_risk	BOOLEAN	Is option with extended risk value?

gate_check_answers – table contain all answers of every survey which was taken in the Gate-Check system		
global_cases_id	VARCHAR(12)	clear identification of global case
options_id	INT(11)	clear identification of option, which was chosen as answer
question_id	INT(11)	clear identification of question for which is this answer

answer_extend	VARCHAR(300)	if the chosen option has extended answer, the text is stored here
risk_extend	INT(11)	if the chosen option has extended risk value, the value is stored here

gate_check_global_cases – table contain all global cases whis was inserted into Gate-Check system		
id	VARCHAR(12)	clear identification of global case
country_id	VARCHAR(2)	clear identification of country in ISO 3166-1 alpha 2 format
name	VARCHAR(50)	name of the company which has this global case number
focus	VARCHAR(50)	focus of company which has this global case number
description	VARCHAR(100)	description – not used now

gate_check_gates – table contain all gates in the Gate-Check system		
id	INT(11)	clear identification of gate
gate_name	VARCHAR(45)	name of the gate

gate_check_country – table contain all countries in the world		
id	VARCHAR(2)	clear identification of country in ISO 3166-1 alpha 2 format
name	VARCHAR(64)	full name of country

gate_check_global_case_values – values of counted risk for every global case and gate for which was taken survey		
id	VARCHAR(12)	clear identification of global case
gates_id	INT(11)	clear identification of gate
value	INT(11)	value of risk for specified global case and gate

gate_check_role – table contain all roles of user, which can user has in Gate-Check system		
id	INT(11)	clear identification of role
name	VARCHAR(45)	name of the role

gate_check_gc_user – table contain all user which are registered to Gate-Check system		
id	INT(11)	clear identification of user
role_id	INT(11)	clear identification of role
name	VARCHAR(20)	user name
password	VARCHAR(40)	user password