

Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Technologies



Bachelor Thesis

Developing web application using modern practices

Petr Klika

© 2017 CULS Prague

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

BACHELOR THESIS ASSIGNMENT

Petr Klika

Informatics

Thesis title

Developing web application using modern practices

Objectives of thesis

The goal of this Bachelor thesis is to develop the web application using modern programming languages and best practices. The emphasis is on clean code and compliance of MVC (Model, View, Controller) design. Subgoals are development of server side using one of the most popular PHP open source frameworks Laravel based on Symfony released in 2012. The implementation of user interface using the Blade templating system and JavaScript. The data will be stored in SQL database MySQL.

Methodology

In the literature search part, there will be defined and described major definitions of the project, requirements, basic theoretical knowledge of the chosen PHP framework, programming languages and other used technologies, advantages and disadvantages of programming languages, frameworks and database systems which will be used in the project. There will be a comparison of competitive technologies, exploration of actual trends, number of Laravel users and their geographical classification and options of testing the server side and user interface part. The practical part will be composed of detailed description of individual parts of the project and implementation steps: the process of designing of the database model, basic components of the system following the MVC (Model, View, Controller) design. There will be indicated potential problems and complications during development of the application. The Bachelor thesis will document the development cycle of the application including project definitions, analysis, designing, implementation and testing.

The proposed extent of the thesis

60-80 pages

Keywords

PHP Framework Laravel web application programming MVC Blade SQL MySQL development

Recommended information sources

Martin Pokorný, PHP nejen pro začátečníky, Computer Media, 2005, ISBN: 8086686388

Miloslav Ponkrác, PHP a MySQL bez předchozích znalostí, COMPUTER PRESS, 07.01.2013, EAN: 9788025117583

POKORNÝ, J. – GILMORE, W J. *Velká kniha PHP 5 a MySQL : kompendium znalostí pro začátečníky i profesionály*. Brno: Zoner Press, 2011. ISBN 978-80-7413-163-9.

Timothy Boronczyk, MySQL – Ovládněte MySQL za víkend, Computer Press, 2016, ISBN: 9788025147375

W. Jason Gilmore, RNDr. Jan Pokorný, Velká kniha PHP 5 a MySQL, Zoner Press, 2011, ISBN: 978-80-7413-163-9

Expected date of thesis defence

2016/17 SS – FEM

The Bachelor Thesis Supervisor

Ing. Martin Havránek, Ph.D.

Supervising department

Department of Information Technologies

Electronic approval: 21. 10. 2016

Ing. Jiří Vaněk, Ph.D.

Head of department

Electronic approval: 24. 10. 2016

Ing. Martin Pelikán, Ph.D.

Dean

Prague on 09. 03. 2017

Declaration

I declare that I have worked on my bachelor thesis titled "Developing web application using modern practices" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the bachelor thesis, I declare that the thesis does not break copyrights of any their person.

In Prague on 2017

Acknowledgement

I would like to thank Ing. Martin Havránek, Ph.D. for leading me during my Bachelor Thesis development process, for all his advices and willingness. I would also like to thank my family for information provided to me. Also, thank to others who have been supporting me during the creation.

Vývoj webové aplikace pomocí moderních technologií

Souhrn

Tato Bakalářská práce řeší vývoj webové aplikace za použití moderních programovacích technik, jazyků a dalších technologií. Cílem této práce je analýza moderních programovacích jazyků a technologií, porovnání vybraných technologií a za pomoci vybraných programovacích jazyků, frameworků, databázové technologie a frameworků pro vytvoření stylu aplikace následné naprogramování webové aplikace. V praktické části této práce jsem po analýze vybraných technologií zvolil za programovací jazyk serverové části PHP ve spojení s PHP frameworkem Laravel, který vychází z populárního PHP frameworku Symfony. Jako databázový systém pro ukládání dat jsem zvolil MySQL databázi. Při implementaci klientské části webové aplikace jsem využil šablonovací systém Blade, který se běžně používá ve spojení s Laravel frameworkem. Pro responzivní design aplikace jsem využil Bootstrap css framework a funkce, které aplikace plní na straně klienta jsou psány v Jazyce JavaScript, dále je také využíván AJAX. Postup při vytváření webové aplikace jsem zdokumentoval. Práce zahrnuje definici projektu, jeho potřeby, plánované využití projektu v budoucnu, návrh a jednotlivé implementační kroky.

Klíčová slova: webová aplikace, PHP, Laravel, JavaScript, SQL, MySQL, INCASE, Blade, Bootstrap

Developing web application using modern practices

Summary

This Bachelor thesis handles the development process of a web application using modern programming practices, programming languages and other technologies. The goal of this Bachelor thesis is to analyze modern programming languages and technologies, to compare chosen technologies and using them developing web application with specific requirements and future use. After analyzing backend, frontend and database technologies I decided to choose as my primary backend programming language PHP in connection with popular PHP framework Laravel. For implementation of user interface, I decided to choose Blade templating system frequently used in connection with Laravel Framework. As database technology for data storage I chose MySQL database. For responsive design of the web application I decided to use Bootstrap css framework. Frontend functions provided by the web application are developed using JavaScript language and there are also present AJAX requests. I documented the development process. The thesis is composed of project definition, its requirements, future use of the project, design and individual development steps.

Keywords: web application, PHP, Laravel, JavaScript, SQL, MySQL, INCASE, Blade, Bootstrap

Table of content

1	Introduction.....	10
2	Objectives and Methodology	11
2.1	Objectives.....	11
2.2	Methodology	11
3	Literature Review.....	12
3.1	Introduction to the internet and WWW.....	12
3.1.1	The principle of client/server	13
3.1.2	URL	14
3.1.3	Protocol HTTP	14
3.2	Dynamic websites and web applications.....	15
3.2.1	Comparison to static websites.....	15
3.2.2	Object-oriented programming.....	16
3.3	General development tools.....	16
3.3.1	IDE.....	16
3.3.2	Version control system	16
3.4	Server side scripting.....	17
3.4.1	Samples of server side programming languages.....	17
3.4.2	TIOBE index of top 10 popular programming languages	18
3.4.3	PHP	18
3.4.4	Comparison of PHP and Python	19
3.4.5	Motivation to choose PHP	20
3.4.6	Framework	20
3.4.7	Samples of PHP frameworks	21
3.4.8	Laravel	21
3.4.9	Comparison of Laravel and Yii and motivation to choose Laravel.....	22
3.5	Database systems	24
3.5.1	SQL.....	24
3.5.2	Samples of most popular relational database systems	26
3.5.3	MySQL	26
3.5.4	Comparison with PostgreSQL and motivation to choose MySQL.....	27
3.6	Client side scripting.....	28
3.6.1	HTML	28
3.6.2	CSS	29
3.6.3	Blade templating system.....	29
3.6.4	JavaScript.....	30
3.6.5	Popular JavaScript extensions and technologies	30

4	Practical Part	32
4.1	Motivation for chosen technologies	32
4.2	Motivation for developing of the web application and its description	33
4.2.1	Protentional company using the web application	33
4.2.2	Incase	34
4.3	Use cases	34
4.4	The development process	39
4.4.1	Starting local Apache and MySQL server	39
4.4.2	Installation and configuration of Laravel framework	39
4.4.3	Version controlling	40
4.4.4	Designing of the database structure	41
4.4.5	Scaffolding technique	44
4.4.6	Routing	45
4.4.7	Blade views	46
4.4.8	Application conclusion	49
5	Conclusion	50
6	References	51
7	List of used pictures	53

1 Introduction

Internet became a global concept and phenomenon. From the view of technology, internet is just a computer network above which there are running a lot of services which are exploited by people all around the world. The most popular service became WWW so called web pages. Although the number of internet users have been increasing in the first 10 years very fast (1995 - 16 million, 2005 - 1,018 million users), the number of users have increased in the last 11 years even faster (2005 - 1,018 million users, 2016 - 3,675 million users) [1]. It proves that technologies are developing and much more people are connected to the internet. Technologies used for web are developing each year and we are mostly not satisfied with just two or three programming languages or technologies during our development process. We usually need a lot of them to finish the project according to nowadays needs. Developers use version control systems to control their development process. Project are being automatized and frameworks usage and development is growing very fast because they make programming and coding much easier and much safer. It is hard nowadays to have a total overview over all web technologies.

After the huge expansion of web technologies, developers were looking for ways how to add more dynamic elements to the website. During a few years, the process went as far as we are nowadays able to create web applications which are completely the same as desktop applications which was originally the main purpose. For example MS Word, which is the text editor available as desktop version is now also available in a web application called Office 365 or using Google Docs. These applications are originally called web applications.

2 Objectives and Methodology

In this section I will describe goals and methodology of the Bachelor thesis and I will introduce you my intentions.

2.1 Objectives

The goal of this Bachelor thesis is to develop a web application using modern programming practices and modern technologies and techniques which is going to be at the end of the development process placed in a company. Analysis of modern backend, frontend and database technologies and after decision development of server side using PHP in connection with popular PHP framework Laravel released in 2012 based on Symfony. Development of user interface using Blade templating system frequently used with Laravel framework. Database integration using MySQL database. The thesis will document the project definition, its requirements, future use of the project, design and individual development steps.

2.2 Methodology

In the literature review, there will be stated definitions of the project, the applications actual purpose in the real world its requirements and basic theoretical statements of selected technologies and programming languages which are used in the practical part of the Bachelor thesis. Comparison with other competitive technologies, comparison of nowadays trends, known advantages and disadvantages of these programming languages and technologies.

The practical part will be composed of description of the development process and all individual implementation steps. Creating the visual design. Preparing of the Laravel framework and all its necessary parts for using. Designing of the database structure, scaffolding and implementing of user interface.

3 Literature Review

In this section I will describe basic definitions of the project, the applications actual purpose in the real world its requirements and basic theoretical statements of selected technologies and programming languages. These all terms are necessary to understand the architecture of the practical part.

3.1 Introduction to the internet and WWW

Internet is a worldwide network of computers connected together which are able to store data in an electronic form and share them among other computers. It is a set of technical resources allowing distribute of data electronically through the world without any restriction of the type and content [2].

The WWW was launched in 1990 on the field of CERN science centre. It is the most expanded service of the present Internet. It is universal method of accessing information based on hypertext documents. WWW is distributed system which means that information is divided among a huge number of computers connected to the internet. The system of WWW holds just pointers and links to all data regarding to searched subject although they are held worldwide.

Most of the people use web pages without any knowledge how does WWW (World Wide Web) work. The web page is just a text document containing information which are provided to web users. Text inside the document is then edited by special elements using which we can add there for example links to other pages of the web page, pictures and other options.

The World Wide Web is built on four basic technologies.

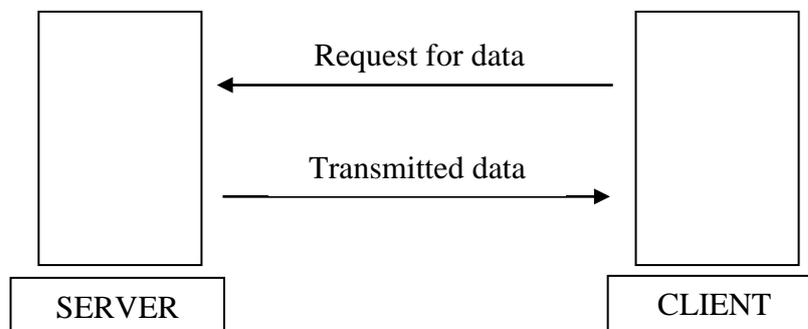
- Web pages and technologies connected to them (HTML, XHTML, CSS, ...)
- The principle of client/server
- URL
- Protocol HTTP

For the web page itself on the frontend side we use markup languages to be able to add special features to them. The most known and used markup language nowadays is HTML – The original markup language for creating websites was released between 1991 and 1993 by Tim Bernes-Lee and Daniel Connolly [2]. The first versions of this markup

language (0.9-1.2) has not supported graphical interface. Support of the graphical interface was added in 2.0 version of HTML released in October of 1995 [22].

3.1.1 The principle of client/server

Most of the services not just on the internet are based on the principle of client/server. WWW service is one of them. If you want to use WWW service and get access to a web page, two sides communicate together. The first side so called server side is the source of data, it stores data in an electronic form and provides them to clients. The second side so called client side mostly internet browser is used by user who has an interest to get a content of any web page or web application. The communication of client and server can be simply illustrated like this:



Picture 1: Architektura klient/server. Source: [2]

Model client/server became the most important and main idea of the network technology. This model is nowadays used by most of the business applications and it is also used by following network protocols HTTP, SMTP, Telnet, DNS etc. Each client instance can create requests for data on one or more connected servers. Most frequently used clients are basically browsers and servers are mostly web servers, database servers and email servers

The advantage is that all data are stored on servers which are mostly much safer than clients. Server can easily control access to data which provides that modify and access data can only authorized client.

The huge disadvantage is overloading the network. In view of the fact that number of parallel requests of clients on the server increases, the server can be overloaded [2].

3.1.2 URL

URL (Uniform Resource Locator) creates in the world of network technologies something like postal address is in the real world. Every information source has accurately described location where it truly is held. Using this we can find on the internet everything what we have access to. Each of us had probably a possibility to meet with URL. In case of programming, the URL is very important part of it. The programmer is using it able to pass parameters across the web application. Using parameters, we tell the application what to do or for example what to show on the web page [2]. URL can be composed of following parts:

<http://www.example.com/getPicture.php?id=3>

- Protocol HTTP
- The domain name
- Path to the PHP script (this script displays a picture)
- Parameter “id” (we want to display the picture which has ID: 3)

3.1.3 Protocol HTTP

HTTP (Hypertext Transfer Protocol). It is a protocol which transfers hypertexts and allows user mostly of an internet browser to get the content of a web page. User fills the domain name to the internet browser, the browser analyses domain, using DNS (Domain Name System) and recognizes to which IP address must be requested. Through TCP protocol creates a connection with server on the recognized IP address, the internet browser then sends HTTP request to the server which looks like that:

```
GET /www/test.html HTTP/1.1  
HOST: www.test.com
```

GET method tells the server that user wants to get something mostly a content of some web page. **HOST** determines a domain which will be requested, it is necessary because practically we can have many domains on one physical server [22]. Last row is left blank. It tells that it is the end of the call. So, the complete request looks like server goes to the root which is / looks for **www** folder and gets file **test.html**. The server can see from the request that it must send the file using **HTTP** (version 1.1), it adds HTTP headers which are very important right before the file and everything is sent to the client [3] [2] [6].

A header can contain for example information about redirect, caching, cookies, referrer, etc.

Server responses:

- 200 – OK
- 301 – Moved permanently
- 302 – Mover temporarily
- 304 – Not modified
- 404 – Not found
- 403 – Forbidden
- 500 – Internal server error

3.2 Dynamic websites and web applications

A dynamic website is a website where its content was generated individually for each display. Its content changes in time (blogs) and the content may be different for each user of the website (logged user vs. anonymous user) also content of the website may change after user's interaction with the website (web games). Dynamic websites often have a possibility to change its content by user's settings (stored in Cookies). The same goes with web applications.

Dynamic applications are often connected with some database system mostly SQL system which stores data (titles, products, contents, prices, users etc.). Server on client request to display a specific web page calls data from the database using mostly SQL query and sends data to the client's browser. Web page is compiled based on "template" which determines appearance and data from the database. Dynamic websites and applications against static websites are not physically stored and its content changes [2].

3.2.1 Comparison to static websites

A static website is a set of web pages which are connected to each other by permanent links. Each page is completed in state which author left it. The biggest disadvantage of this website is that it is very hard to maintain or extend it. To maintain the website author always must go through the code and change it. So, the website maintenance is very hard for beginners sometimes impossible. Static web pages can be developed only by person who has already some knowledge about at least HTML.

Static websites are not able to store the data in database systems. User is not able to interact with the website. Static website should be nowadays used only for very small projects. Static websites are nowadays very outdated by dynamic websites.

3.2.2 Object-oriented programming

Object-oriented programming (OOP) is programming pattern. Difference between OOP and procedural programming is sensible. The goal of OOP is to construct small and comparatively independent pieces of code which can be used repeatedly at many places in the application. That pieces of code are in OOP called objects. Development of an OOP application is faster and easier but the biggest advantage of OOP application is its easy maintenance and debugging. Objects takes some amount of memory so as disadvantage we can say that perfectly coded OOP application may be slower than perfectly coded procedural application [2] [6].

3.3 General development tools

3.3.1 IDE

IDE is a software which makes developing easier, it is mostly focused on one or more programming languages. It contains editor of a source code, compiler eventually interpret and mostly debugger. Development environment are designed to raise productivity of a programmer. Some of them are open sources and completely free [17].

List of most popular IDEs:

- **Notepad++**
- **SublimeText**
- **Eclipse**
- **NetBeans**
- **Visual Studio**
- **PhpStorm**

3.3.2 Version control system

Version control systems are a category of software tools that help a software team manage changes to source code over time. Version control software keeps track of every

modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake [18].

Git

Distributed system for controlling code versions by Linus Torvalds originally developed for Linux kernel development process. Git is currently used by number of famous projects. Git is held under the GPL license and is free to use.

Web pages providing repository management based on Git

- Assembla
- Bitbucket
- GitHub

Bitbucket

A web service which supports development of the software with using version control tool Git. Bitbucket provides free hosting for open-source projects. It is a service like GitHub [18] [19].

3.4 Server side scripting

User requests are executed on the side of the web server. Websites using server side scripting usually access the database or another kind of database storage. That's the main difference between the client side scripting and the server side scripting. The client side scripting does not need a web server to execute scripts. The biggest advantages of server scripting are the possibility to affect access rights, control logins and log failures. The security advantage is that the content of the script is never sent to the client's browser. The user is not able to read the script [2].

3.4.1 Samples of server side programming languages

- PHP
- Python
- Ruby
- C#
- C++
- Java
- Erlang

3.4.2 TIOBE index of top 10 popular programming languages

Feb 2017	Feb 2016	Change	Programming Language	Ratings	Change
1	1		Java	16.676%	-4.47%
2	2		C	8.445%	-7.15%
3	3		C++	5.429%	-1.48%
4	4		C#	4.902%	+0.50%
5	5		Python	4.043%	-0.14%
6	6		PHP	3.072%	+0.30%
7	9	▲	JavaScript	2.872%	+0.67%
8	7	▼	Visual Basic .NET	2.824%	+0.37%
9	10	▲	Delphi/Object Pascal	2.479%	+0.32%
10	8	▼	Perl	2.171%	-0.08%

Picture 2: The TIOBE Programming Community index. Source: [5]

The TIOBE programming community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system [5].

3.4.3 PHP

PHP (Hypertext preprocessor) is a server side scripting language created for web development designed by Rasmus Lerdorf in 1994. PHP code can be directly included into the HTML code or it can be also used with the various web template systems or web content management systems and web frameworks. PHP supports various variables. Each variable has a name sometimes called identifier and value. The value of variable is in each moment clearly identified but the value can change during the run. Type of the variable is defined when the value is assigned to the variable. That's one of the biggest advantages of PHP programming language.

Scripts are executed on side of server while using PHP. The user always get only result of some operations which happened during execution. It is possible to call an interpreter of a PHP script using command line, HTTP requests or by using web services. The syntax of PHP is inspired by languages like Perl, C, Pascal or Java. PHP supports

many libraries with different purpose (graphics, file management, text processing, access to most of database systems – MySQL, Oracle, PostgreSQL, MMSQL...), It also supports various internet protocols (SMTP, FTP, POP3, IMAP, ...). Other programming languages sometimes require programmer to define type of the variable before its use. Each variable must have a clear name and the name starts with “\$” sign (\$car = ‘bmw’;). Every PHP command ends with “;” sign. PHP allows functions, classes and other Object-oriented programming signs [2] [6].

Release history:

- 1.0 (8 June 1995). Called Personal Home Page Tools (First use of PHP).
- 2.0 (1 November 1997)
- 3.0 (6 June 1998)
- 4.0 (22 May 2000)
- 5.0 (13 July 2004)
- 5.6 (28 August 2014) - Stable
- 7.0 (3 December 2015) - Stable
- 7.1 (1 December 2016) – Stable

3.4.4 Comparison of PHP and Python

Both languages are interpreted, high level with dynamic typing, they are also Open source which means that they can be used and distributed by anyone. They have very large community of programmers and developers who are active on the internet, supporting each other, developing new libraries, tools, extensions, writing books focused on many problematics regarding to programming by using them. These server side programming languages are friendly to beginners, they are easy to learn compared to C++ and others. They are portable so, there are almost no problems to run them on all platforms.

Another feature is that both languages supports dates which are not limited to UNIX timestamps (<1970, >2038). Both languages have debuggers and IDEs which are completely made just for them or which know their syntax. These IDEs supports code refilling and can highlight syntax errors.

PHP may have some security disadvantages which showed up during past years. PHP is an open source programming language which means that anyone can look into the code and explore its weakness. Implicit conversions can lead to bugs.

3.4.5 Motivation to choose PHP

PHP has some strengths that Python does not have. Regarding to decisions and cycles PHP have switch and do while statements which are sometimes very useful and powerful.

Increment, decrement and assignment operators are present in PHP programming language which is a huge advantage against Python. PHP has private, protected and public modifiers for both properties and methods, it also has abstract modifiers for classes and methods. Regarding to PHP, most of the tools, software, IDEs, libraries, extensions, etc. are open source (WordPress, Prestashop, OpenCart, ...), which makes developing using PHP very cheap and effective.

PHP is completely flexible to database connectivity, it can be connected to various database systems such as MySQL, PostgreSQL, MS SQL, SQLite. PHP supports many PHP frameworks which are immediately ready to use, which are easy to learn and they are well documented. They make job easier a lot of things is solved right from the beginning, there are emphasis on security and clear code.

The PHP community is wide so there are mostly no problems to find information on the internet or read some book which talks about PHP problematics. Another reason for learning PHP is job opportunity because PHP belongs to most demanded programming languages worldwide [4] [2].

3.4.6 Framework

Framework is a software structure serving like a support while programming a developing software projects. It can contain libraries, API, support for design patterns or recommended development procedures. The purpose of framework is to take over some typical problems in the specific area and provide a solution. So, the developer can then focus just on the real solution.

3.4.7 Samples of PHP frameworks

- Laravel
- Symfony
- CodeIgniter
- Yii
- Phalcon
- CakePHP
- Zend Framework
- Slim
- FuelPHP
- Nette

3.4.8 Laravel

Laravel is open source PHP framework which was developed for web applications by programmer Taylor Otwell. First release of Laravel was in 2012 so it is quiet new PHP framework with huge potential. This framework is completely free because it follows the open source policy under the MIT licence. Laravel uses the MVC architecture (model-view-controller). Model – application data and function, View – the representation of data on the client side mostly by using HML language and some templating system corresponding with framework, controller – interaction among user, model and view.

This framework is popular because it includes from the beginning all necessary functionalities like authentication, routing, database, mail, sessions, caching. All these functionalities are important nowadays while developing web applications.

Laravel is based on popular PHP framework and technologies like Symfony, Composer, Eloquent ORM and Blade templating system, further it is well inspired by Ruby on Rails technologies which are nowadays also very popular and powerful.

The base of Laravel is Symfony PHP framework which provides modules like Browserkit, Console, Debug and FileSystem, another important component is Composer (PHP dependencies manager). Another point is that Laravel uses very important Eloquent ORM (object relation mapper), which was specially developed for Laravel framework. Eloquent uses ActiveRecord design pattern, which provides INSERT, UPDATE and DELETE on database. It can easily manage database tables with relations 1:1, 1:n or n:m.

The last important component of Laravel framework is Blade (templating system), which manages all views with **.blade.php** suffix [7] [8].

Directory structure

The directory structure is created to provide a possibility to build both large and small applications [8]

The root directory

- The app directory
- The bootstrap directory
- The config directory
- The database directory
- The public directory
- The resources directory
- The routes directory
- The storage directory
- The tests directory
- The vendor directory

The app directory contains the core code of the application.

The bootstrap directory contains files that Bootstrap autoloads.

The config directory contains configuration files of the application.

The database directory contains database migrations and seeds.

The public directory contains all public resources which are visible for clients.

The routes directory contains route definitions for the application.

The storage directory contains file based sessions, file caches, etc. [8]

3.4.9 Comparison of Laravel and Yii and motivation to choose Laravel

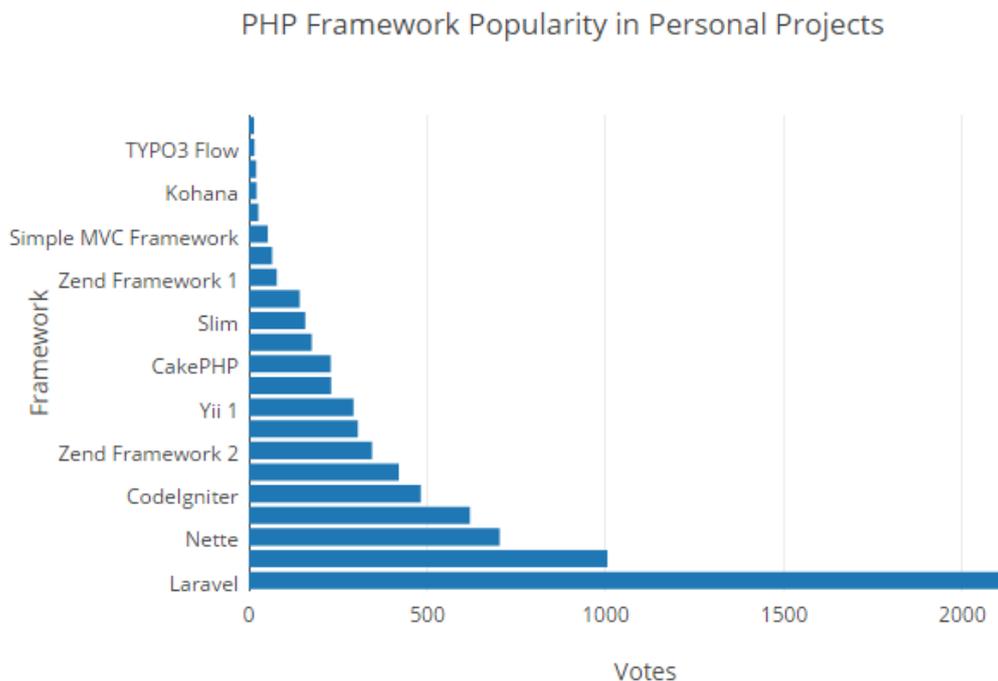
Unlike Yii, Laravel has its own templating system called Blade which lets programmer use PHP code in the views, Blade has no impact to performance, because the Blade views files are stored in **.blade.php** extensions. All the code in the files is converted into raw PHP during processing [9]. Yii has no own templating system by default, it means that coder/programmer must use some third-party templating system externally. There are various templating system such as Smarty or Twig which is used in Symfony. The

installation is very easy for all mentioned frameworks which can be installed by using composer (PHP dependency manager).

Laravel framework and its community is growing very fast. If a programmer has knowledge of PHP and want to start project as fast as possible, consider Laravel framework, because Laravel has an easy curve and it is possible to find many tutorials online to get started. Both frameworks provide code scaffolding for faster code generation and development [9]. Laravel performance is quite slow by default sometimes but mostly it does not matter because you will find easily online resources for speeding up performance. Yii has one of the best performances across all frameworks. Database supports of Laravel and Yii are totally similar.

Both frameworks support basic and most used database systems like MySQL, SQLite, PostgreSQL, MongoDB, etc. One of the most important criteria while making decision what kind of PHP framework should programmer use is how wide is the community of the framework or how fast is it growing. In this case, Laravel stands out although Symfony and Yii are not far behind. Laravel has perfectly documented all components and technologies online. It has wide community and frequently used forums.

Laravel is the most popular framework in 2015-2016, it supports Composer package management, offers tons of package for extending framework functionality, has a great community with lots of learning resources [9].



Picture 3: PHP Framework Popularity in Personal Projects. Source: [10]

3.5 Database systems

The database is a place where data are stored. We usually use a software which can access the database and interact with user, other applications and database itself and work with data inside. These programs are called DBMS (Database Management systems). Among DBMS belong many programs, some of them are free (open source) and some of them are charged. Most used database systems are still MySQL, Firebird, PostgreSQL, ... which are free to use and can be used commercially. Most of database systems are made on relational model. In this case, data are structured in database tables where each table stores data of specific type [2] [6] [11].

3.5.1 SQL

SQL (Structured Query Language) is a domain-specific language used for manipulating data in the relational database management systems. The process of managing data in the database is based on communication with the database server and is simply composed of these steps:

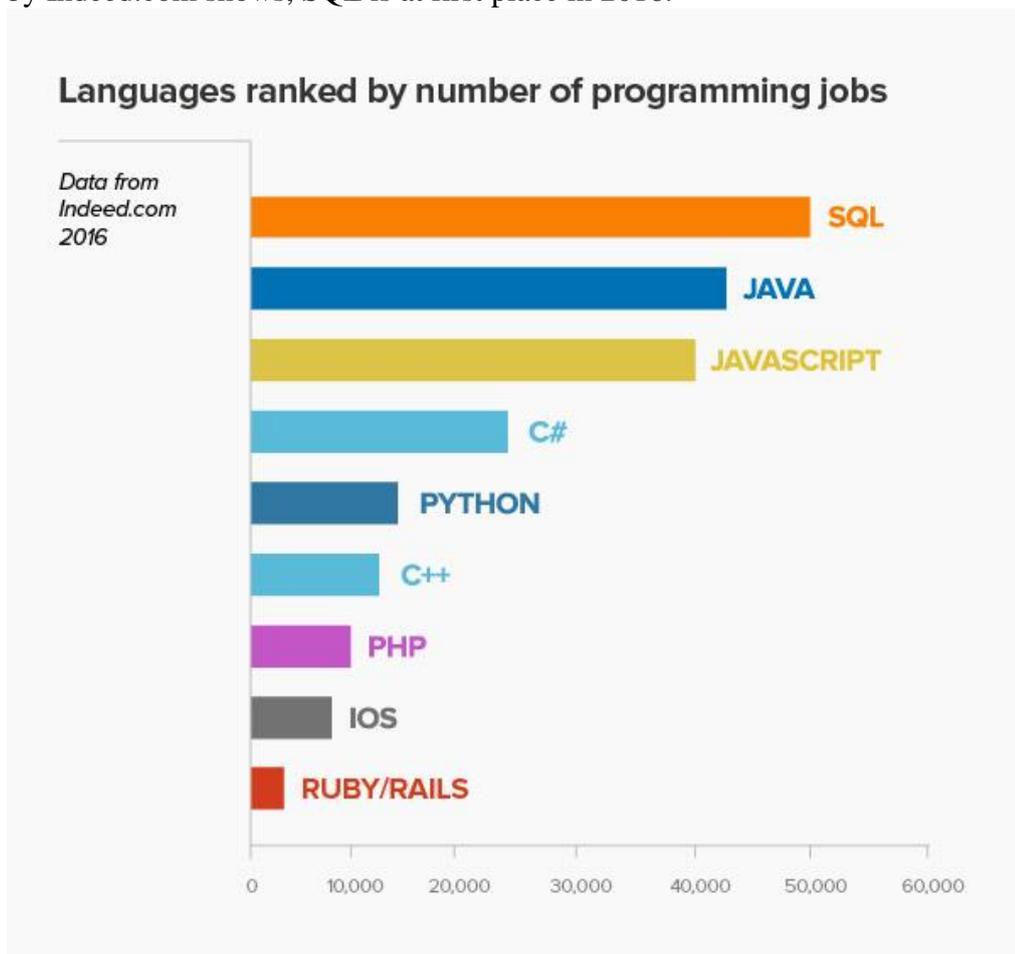
- Create connection with database server
- Send request for manipulation with data
- Eventually pick up data which was requested
- End connection with database server

For the communication with database server, SQL does exist. Every programmer should have at least basic knowledge of the SQL language because nowadays is very popular to create dynamic websites and web applications which are connected to the database. SQL language has a great history which began in 1974 in the IBM company. It is a successor of SEQUEL (Structured English Query Language). SQL language can execute queries against a database, retrieve data from a database, insert, update or delete records in a database. This language is also able to create new database, create new tables in existing database, create views for a user or set permissions on tables. SQL become a standard of the American National Standards Institute (ANSI) in 1986, and of the international Organization for Standardization (ISO) in 1987. [2] [6] [12].

SQL statements:

- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index [16]

SQL is very popular language considering number of programming jobs as this graph by Indeed.com shows, SQL is at first place in 2016.



Picture 4: Languages ranked by number of programming jobs. Source: [13]

3.5.2 Samples of most popular relational database systems

- Microsoft Access
- MySQL
- Oracle
- Microsoft SQL Server
- PostgreSQL
- FileMaker

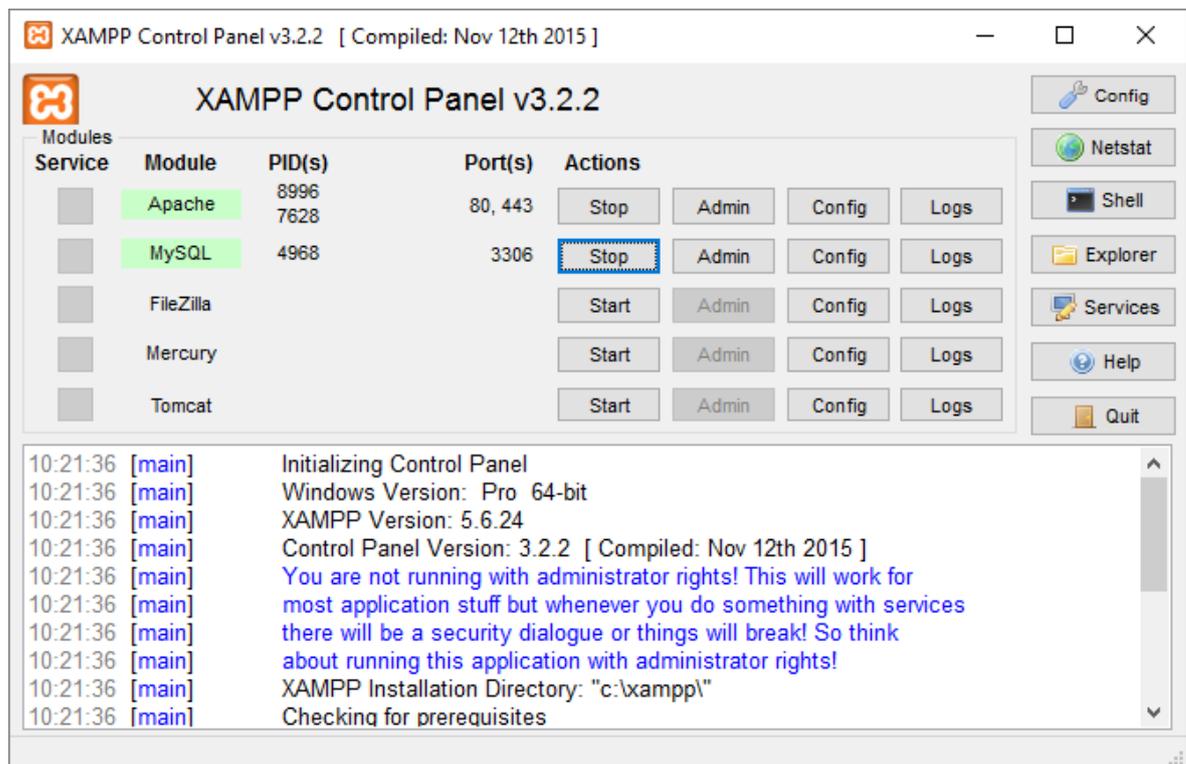
3.5.3 MySQL

MySQL is one of the most popular database management systems worldwide created by MySQL AB Sweden company and nowadays owned by Sun Microsystems company. It is based on relational database model and communication with database is made on SQL language. MySQL is very easy to implement on many platforms (Linux, MS Windows, etc.), it has also suitable performance for small projects even for very complex project like online stored and its popularity is mainly based on fact that it is an open source software. Very popular and often used combination of technologies is Linux, Apache, MySQL and PHP [11].

Tools for working with MySQL

There must be running own MySQL server while using MySQL database management system in case of the local development process. Linux provides possibility to install complete database server using just few commands and configuration by text files which is very popular. The web application described in this bachelor thesis in the practical part was developed locally on the Windows operating system so I was forced to use some external third party application which could provide a database server for me. I used **XAMPP**, which is one of the most popular PHP development environment and it is completely free. By using this tool, I could install and start Apache distribution which has a support of PHP on my computer easily and the database server.

For management of the database I used **phpMyAdmin**, which is traditionally used tool for working with data in the database. XAMPP allowed me to access phpMyAdmin via internet browser by filling the following URL: `http://localhost/phpmyadmin`. Before using phpMyAdmin you must fill information about the database connection which can be found in the XAMPP database server configuration file (username, password) [2] [14].



Picture 5: Example of XAMPP tool. Source: own processing

3.5.4 Comparison with PostgreSQL and motivation to choose MySQL

Both database management systems have similar functions and they are not so different from each other. The main reason for choosing MySQL is widely expanded community. Another thing is that MySQL has perfect support of coding, for each database, table and column is possible to determine independent coding. It supports ON DUPLICATE KEY UPDATE query command which is powerful when you want to update or ignore duplicate records. MySQL also supports data type enum which represents enumeration of some values.

PostgreSQL offers an option of oid (object identifiers) column which makes working with records in any table easy. By default, if there is an error in the one of the processing queries, next queries will be not executed which is desirable behaviour.

As written above, both database management systems are very similar but MySQL is in the February 2017 by db-engines.com the second most popular DBMS in the world which makes it still up to date and growing [2] [6] [15].

3.6 Client side scripting

Client side scripting is used primarily for changing content of the dynamic website. Scripts executed on the client side are mostly included in the HTML or XHTML file. They may be also placed in the external file on which there is a link in the source file. These scripts are mostly JavaScripts or some JavaScript extensions. Web browser executes scripts and displays its results to the client. Scripts can also include instructions for the web browser of how to react on some user events (button click, focus on some HTML element etc.).

The main thing about client side scripting is that scripts do not use server side. Each script can be executed without server independently. Source code of the script can be displayed by user. It is very popular to use client side scripting nowadays for the web development to improve user interfaces.

3.6.1 HTML

HTML (HyperText Markup Language) is basic markup language used in informatics for creating websites which are connected by hypertext links. It is the primary language in the world wide web developed by Tim Berners-Lee and Robert Cailliau in 1990, the last version is 5.1. HTML is based on previously used universal markup language SGML (Standard Generalized Markup Language).

HTML is characterized by number of tags which have attributes. These tags are mostly paired tags. The basic structure of a HTML document is declared by standards. Tag `<!DOCTYPE html>` tells the browser that it is an HTML document. The root element of the HTML document represents `<html>` tag which is paired. The head of a document is represented by `<head>` tag which is also paired. In the head, there are mostly defined features like coding standard, name of the document, author, description, key words, title of the document, css files and other extensions. The body of the document is held in the `<body>` tag [22].

3.6.2 CSS

CSS (Cascading Style Sheets) is a style sheet language which describes the look and function of elements contained mostly in the HTML source code [2] [22].

Samples of CSS frameworks

- **Bootstrap**
- **Pure**
- **Gumby**
- **Foundation**

Bootstrap

Bootstrap is a world popular open source CSS framework developed for web designing which includes predesigned templates based on HTML and CSS serving for typography adjustment, adjustment of forms, buttons, navigations and other components of interfaces. There is needed very basic knowledge of HTML and CSS in case of using Bootstrap. Bootstrap is compatible with latest versions of all internet browsers and provides support to older internet browsers like Internet Explorer 8 which may be during the development process in some cases problematic.

Bootstrap was developed by Mark Otto and Jacob Thornton on Twitter. In August 2011 was released Twitter Bootstrap as open source and in February of 2012 it became the most favourite developer project on GitHub.

Since 2.0 version, Bootstrap supports the responsive design which means that web page is dynamically adaptive considering used device (desktop PC, laptop, tablet, smart phone). The basic form of Bootstrap includes compiled CSS and JS file also compiled minimalized CSS and JS file [20]. To use Bootstrap its necessary to link CSS and JS file in the HTML file as following:

```
<!-- Bootstrap -->  
<link href="css/bootstrap.min.css" rel="stylesheet" media="screen">  
<script src="js/bootstrap.min.js"></script>
```

3.6.3 Blade templating system

Blade is a powerful templating engine provided with Laravel. It does not force the developer to use plain PHP code in the HTML template but it uses special elements to display variables or shortened syntax to create cycles or decisions right inside the view.

Blade view files use the .blade.php file extension and are typically stored in the resources/views directory. The greatest advantages of Blade are inheritance and sections. The developer can easily design a master template which includes all needed parts of the web application like <head>, <body> and <footer>. These parts are then present everywhere. Then it is just about creating sections which are typically displayed in the <body> of the web application [8].

3.6.4 JavaScript

JavaScript is a high-level, dynamic and object-oriented scripting language. It supports objects and functional programming styles. Nowadays JavaScript is used as an interpreted programming language for websites often directly inserted into the HTML document. JavaScript mostly controls GUI interactive elements (buttons, textboxes etc.) or creates interactive elements, animations and effects. The syntax of JavaScript belongs to the family of C/C++/Java programming languages.

3.6.5 Popular JavaScript extensions and technologies

jQuery

jQuery is one of the most popular JavaScript libraries which is allowed by most of the web browsers. jQuery emphasizes on interaction between JavaScript and HTML. jQuery was released on January 2006. It is an open source software under MIT license. jQuery most of the time exists as one JavaScript file which contains all DOM functions, AJAX, events and effects.

React JS

React JS is very popular and new JavaScript project. React is mostly the V in MVC so it is possible to compare it for example with Latte in Nette framework. It is a perfect solution for building user interface. React solves the problem that users do not want to reload the whole web page in each action. It is very comfortable for the user.

Angular JS

Angular is the popular framework used to develop complex websites and web applications. Organizations using AngularJS: iTunes, PayPal Checkout, Google. It is an

open source framework supported by Google. Angular describe itself as an extension to HTML for building complex web applications.

AJAX

It is comfortable when user don't have to completely reload the page to display the data which changed during his browsing. This can be provided by AJAX (Asynchronous JavaScript and XML). AJAX is nowadays very popular and often used extension for JavaScript which allows programmers to communicate with the web server in the background (user does not have to completely reload the web page) and display results. This leads to user interface improvement.

4 Practical Part

I will in this section describe the development process of the web application which was developed using modern programming languages and other current technologies and techniques after analysis in the theoretical part.

I have decided to use PHP and the primary backend programming languages in connection with popular PHP framework Laravel. To have a control over the all process I used the Bitbucket web service which supports development of the software using version control tool Git. I chose MySQL relational database system to store all data used by the application. The application was developed on Windows 10 operation system so I had to use XAMPP to start my own Apache and MySQL server to be able to begin programming and coding. As frontend technology, I used Blade templating system which is originally provided by Laravel itself, CSS framework Bootstrap and JavaScript extensions.

The integral part of the practical part will describe the motivation for developing the web application and its opportune use in the future. All necessary parts of the application and whole development process.

4.1 Motivation for chosen technologies

PHP

PHP is very flexible to database connectivity, fast and friendly programming language which has rapidly growing community of developers. The language is still up to date and all bugs are being corrected over time. PHP has many available frameworks so there is a wide option of choosing between several technologies and procedures of the development process. PHP is in comparison with other programming languages a bit easier to learn so it may be a perfect starting point for beginner developers.

Laravel

Laravel is mainly in Europe still new PHP framework which gives the developer an advantage to bring something new to the community or company. It was the most popular framework used in personal projects in 2016 which creates a perfect support from Laravel community in case of any troubles during the development. Laravel has perfectly

documented all components online at one place. It brings a possibility to scaffold some parts of the application with small effort which makes development effective. It supports by default very clear templating system called Blade which makes HTML files organized.

MySQL

All distributions of Linux, UNIX and Windows are supported which creates a perfect flexibility. The open source nature of MySQL allows complete customization for those wanting to add unique requirements to the database server. The architecture of MySQL allows to configure the database server specifically for particular applications. MySQL can meet with perfect configuration and optimization fast performance. MySQL offers a quick start capability with the average time from software download to installation completion being less than fifteen minutes. For smaller projects, it is very inexpensive. Community is large and because of it, it is easy to debug any error [21].

Git

The biggest motivation of using Git during the development was to have a total control over the code and all changes. If there any error occurred, I could easily revert all changes to the previous version of the application with few commands. All changes are stored remotely so I could access them from anywhere and continue my development process. Of course, Git provides much better results and effectivity if there are more people in the developer team. So, for larger projects it is very powerful solution and technology.

4.2 Motivation for developing of the web application and its description

4.2.1 Protentional company using the web application

The application may be developed for company securing complete enforcement and management of debts, further „company”. Their service is provided in the entire territory of Czech Republic. Debt collection is provided using all legal means within the framework of legislation in the field of commercial, civil and criminal law.

The company offers to their future clients a flexible and efficient service through which they will be able to get their financial funds back. The company takes over all administrative duties and provides a payment morale.

4.2.2 Incase

The Incase is a web application developed for Czech company which holds a business in the field of enforcement and management of debts. The goal of the project was to develop an application which can serve to employees and managers of the company. The requirement was to create an application which can store all information about registered clients of the company, debtors who owe financial funds to company's clients, cases opened in the company and other information connected with cases (invoices, commissions, notes, etc.).

The emphasis was to create a responsive design which would be suitable for all devices used nowadays and pleasure environment for employees.

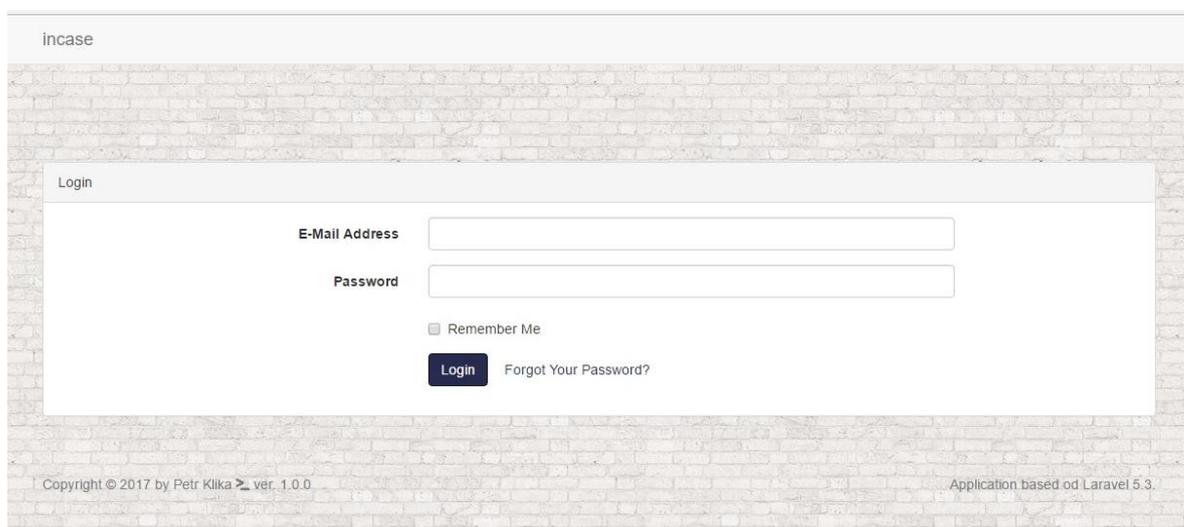
Another requirement was to create and secure complete authentication process and to add two permission roles (user and administrator) to be able to change content of the application based on the role.

The employee must be able to completely manage all information connected with the service provided by the company and must be able to dynamically search in all records.

4.3 Use cases

Use case 1

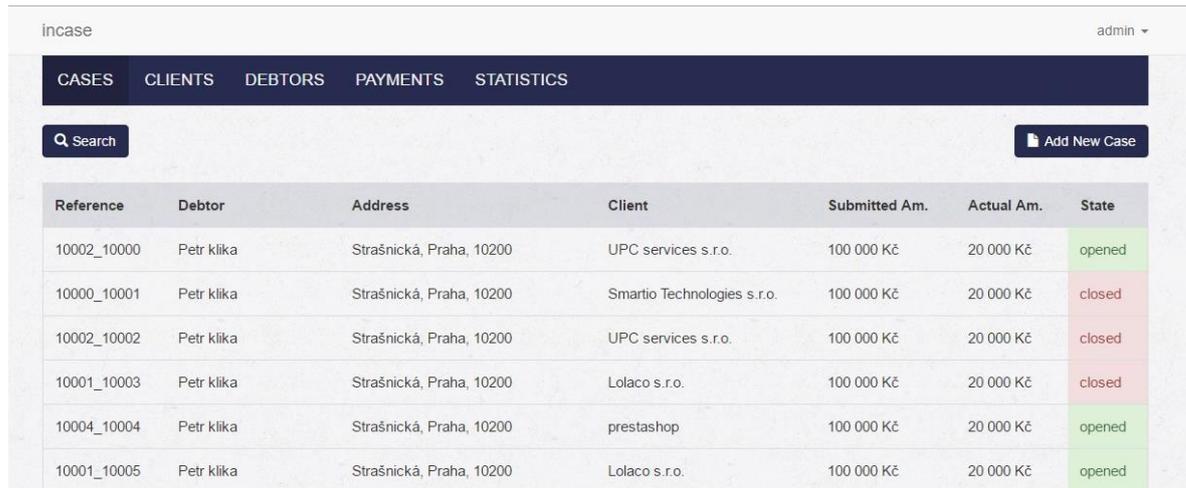
An employee of the company logs in the web application using login details which was generated and given to him by the administrator. The administrator has the possibility to create a new employee account inside of the Incase.



Picture 6: Login. Source: own processing

Use case 2

After login to the Incase, the employee see the extraction of cases created by the company. By clicking the search button, he opens search fields in which he can search through the table of cases dynamically (using jQuery extension) without clicking any confirmation button. The employee can show all details related to the case by clicking on it or create a new case by clicking the button in the right-hand side corner.



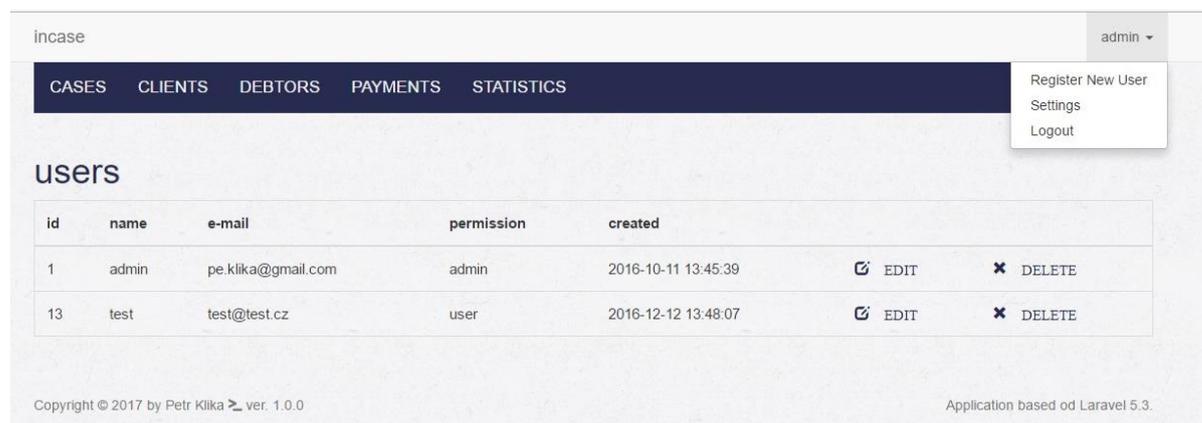
The screenshot shows the Incase landing page. At the top, there is a navigation bar with tabs for CASES, CLIENTS, DEBTORS, PAYMENTS, and STATISTICS. Below the navigation bar, there is a search bar and an 'Add New Case' button. The main content area displays a table of cases with the following data:

Reference	Debtor	Address	Client	Submitted Am.	Actual Am.	State
10002_10000	Petr klika	Strašnická, Praha, 10200	UPC services s.r.o.	100 000 Kč	20 000 Kč	opened
10000_10001	Petr klika	Strašnická, Praha, 10200	Smartio Technologies s.r.o.	100 000 Kč	20 000 Kč	closed
10002_10002	Petr klika	Strašnická, Praha, 10200	UPC services s.r.o.	100 000 Kč	20 000 Kč	closed
10001_10003	Petr klika	Strašnická, Praha, 10200	Lolaco s.r.o.	100 000 Kč	20 000 Kč	closed
10004_10004	Petr klika	Strašnická, Praha, 10200	prestashop	100 000 Kč	20 000 Kč	opened
10001_10005	Petr klika	Strašnická, Praha, 10200	Lolaco s.r.o.	100 000 Kč	20 000 Kč	opened

Picture 7: Landing page. Source: own processing

Use case 3

The administrator crates a new employee or administrator of the company or edits information or permission to existing one.



The screenshot shows the Incase administrator page. At the top, there is a navigation bar with tabs for CASES, CLIENTS, DEBTORS, PAYMENTS, and STATISTICS. Below the navigation bar, there is a dropdown menu with options: Register New User, Settings, and Logout. The main content area displays a table of users with the following data:

id	name	e-mail	permission	created		
1	admin	pe.klika@gmail.com	admin	2016-10-11 13:45:39	EDIT	DELETE
13	test	test@test.cz	user	2016-12-12 13:48:07	EDIT	DELETE

Copyright © 2017 by Petr Klika, ver. 1.0.0. Application based od Laravel 5.3.

Picture 8: Administrator. Source: own processing

Use case 4

An employee views all registered clients of the company and search them by names or adds a new one by clicking the button in the right-hand side corner. An employee displays complete details about the client or edits the record.

incase admin ▾

CASES CLIENTS DEBTORS PAYMENTS STATISTICS

Search For Names **Add New Client**

Name	IČO	Bank Account	Commission	Configuration
Smartio Technologies s.r.o.	12345678	115-1234567874	8 %	info edit
Lolaco s.r.o.	87654321	115-1234567874	10 %	info edit
UPC services s.r.o.	45612398	115-12345678	5 %	info edit
Test s.r.o.	45698712	115-12345678	4 %	info edit

Picture 9: Clients. Source: own processing

incase admin ▾

CASES CLIENTS DEBTORS

Search For Names **Add New Client**

Complete information

Name	Svaly.cz s.r.o.
IČO	78956425
Bank Account	252633548/0100
Commission	4 %
Contact Person	Petr Kvaska
Email	test@test.com
Phone	604558779
Created	2016-12-22 17:51:19

Name	IČO	Bank Account	Commission	Configuration
Smartio Technologies s.r.o.	1234			info edit
Lolaco s.r.o.	8765			info edit
UPC services s.r.o.	4561			info edit
Test s.r.o.	4569			info edit
prestashop	1234			info edit
prestashop	12345678	115-12345678	10 %	info edit

Picture 10: Client info. Source: own processing

incase admin ▾

CASES CLIENTS DEBTORS PAYMENTS STATISTICS

Add New Client

Name	<input type="text"/>
IČO	<input type="text"/>
Bank Account	<input type="text"/>
Commission %	<input type="text"/>
Contact Name	<input type="text"/>
E-Mail Address	<input type="text"/>
Phone number	<input type="text"/>

Add Client

- Required Fields *
- IČO 8 Numbers

Picture 11: Add new client. Source: own processing

incase admin ▾

CASES CLIENTS DEBTORS PAYMENTS STATISTICS

Edit Client

Name *

IČO

Bank Account *

Commission % *

Contact Name

E-Mail Address

Phone number

Update

• Required Fields *
• IČO 8 Numbers

Picture 12: Client update. Source: own processing

Use case 5

An employee creates a new case by clicking the „Add New Case” button in the landing page’s right-hand side corner. The shown form gives a list of previously registered clients from which the employee can choose. After filling all necessary information, Incase creates a debtor automatically.

DEBTORS PAYMENTS STATISTICS

Add New Case

Client *

Company/Name *

Debtor Surname

Debtor Firstname

Street

Town

Post Code

RČ/IČO

Phone Number

E-Mail Address

Commission %

Add Case

Client

- Smartio Technologies s.r.o.
- Smartio Technologies s.r.o.**
- Lolaco s.r.o.
- UPC services s.r.o.
- Test s.r.o.
- prestashop
- prestashop
- Lukáš Marek s.r.o.
- Ferdinand Mravenec s.r.o.
- Company s.r.o.
- Svaly.cz s.r.o.
- Testák s.r.o.0000
- Lukáš Marek

• Required Fields *
• IČO 8 Numbers

**Picture 13,14: Client update
Source: Practical part**

Use case 6

After creating a new case the employee is redirected to the page which displays complete details related to it. The employee adds new invoices to the case by clicking the “Add Invoice” button in the right-hand side corner. These invoices are then immediately displayed. In the same way, the employee adds notes and payments to the case.

incase admin ▾

CASES CLIENTS DEBTORS PAYMENTS STATISTICS

test s.r.o. 10002_10000 OPENED Add Invoice Add Note

opened ▾

test s.r.o. - 12346578
Strašnická, Praha, 10200
Petr Klíka
+420 604667552
pidr12@centrum.cz

Invoices

Invoice Number	12554874	
Due Date	2016-12-16	
Exposed Amount	200000 Kč	pay edit delete
Paid Amount	0 Kč	
Pending Amount	200000 Kč	
Commission	5 %	

Notes

admin	testovací poznámka	edit delete
2016-12-29 13:37:36		

Copyright © 2017 by Petr Klíka ver. 1.0.0 Application based od Laravel 5.3.

Picture 15: Case details. Source: own processing

Use case 7

An employee adds notes to the case which are important to track the case process.

Use case 8

An employee adds invoices to the case which describes how much money the debtor owes to the client.

Use case 9

An employee closes the case if all invoices are paid properly.

4.4 The development process

4.4.1 Starting local Apache and MySQL server

I ran my own Apache (port 80) and MySQL (port 3306) server using XAMPP Control Panel v3.2.2. Edited MySQL configuration file and set up the username and password for the database. I could by using phpMyAdmin tool access the database and create all necessary tables for the Incase web application.

4.4.2 Installation and configuration of Laravel framework

Installation of Laravel framework

Installation was made using Composer dependencies manager.

```
composer global require "laravel/installer"
```

Creating of new Laravel project

After installation, the `laravel new` command created a fresh Laravel installation in the directory which I specified.

`laravel new incase` created a new directory *incase* containing a fresh Laravel installation with all Laravel's dependencies already installed.

Starting the application

I have been developing the Incase locally so I used PHP's build-in development server to serve my application. By executing the command `php artisan serve` I started the application on URL: <http://localhost:8000> (similar principle like Ruby on Rails has) [8].

Connecting of the project to the MySQL database

The database configuration is located at *config/database.php* file. In this file, I defined the connection to the database. I set username, password of my MySQL database and host (127.0.0.1) which is localhost. It is also possible to define more databases at ones and one of the Laravel feature is that you may define one database just for selects and another database just for inserts which may provide a perfect security for the application.

I also set the default charset and collation standard for the MySQL database to utf8 (utf8_bin) which is perfectly suitable for English and Czech characters [8].

```
'mysql' => [
  'read' => [
    'host' => 127.0.0.1,
  ],
  'write' => [
    'host' => 127.0.0.1
  ],
  'driver'    => 'mysql',
  'database'  => 'incase',
  'username'  => 'root',
  'password'  => "",
  'charset'   => 'utf8',
  'collation' => 'utf8_bin',
  'prefix'    => "",
],
```

4.4.3 Version controlling

After creating the account on www.bitbucket.org, I could create my own private repository to store everything regarding to Incase application. Through this system, I could control all my code modifications and eventually track errors which occurred during the development process.

To be able to communicate with Bitbucket servers, there must be Git installed in the machine. I have downloaded windows version by visiting the official site of Git:

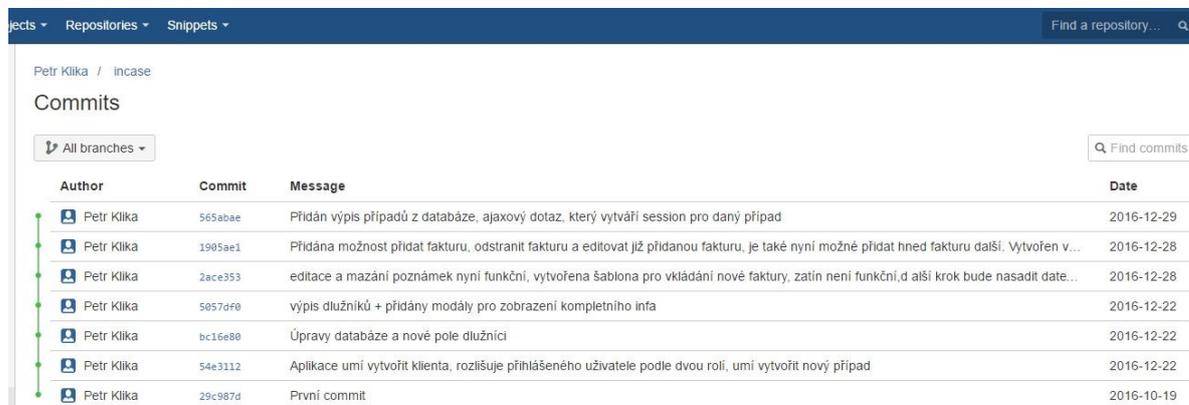
www.git-scm.com

Git can be controlled through command lines which are present in each operating system. I have been developing on Windows so I used the windows command line (cmd.exe) to work with Git. There are few commands which are necessary to be familiar with.

Firstly, right after creating the project folder in `D:\projects\incase`, I used in this folder `git init` command to create an empty Git repository. After that I executed `git clone path_to_remote_repository` command to create a working copy of a local repository. The remote path is available right from the back office of Bitbucket service after logging in. Then I performed a commit so called the initial commit which should be the first step in the basic git workflow (at the beginning I want to add whole my project to Bitbucket) `git add *` (adds everything). I added a message to the initial commit which is necessary, each commit must have a message which describes which changes have been made. For the

initial commit, I leaved the message as “první commit” `git commit -m “initial commit”`. To push the changes to the remote repository I executed the command `git push`.

After every important change in the code I performed steps from `git add` command.



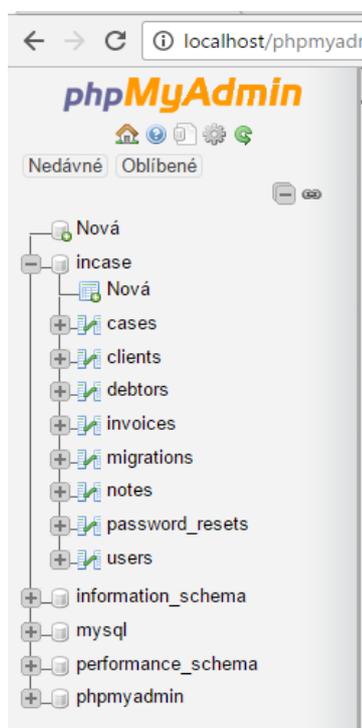
Author	Commit	Message	Date
Petr Klíka	565abae	Přidán výpis případů z databáze, ajaxový dotaz, který vytváří session pro daný případ	2016-12-29
Petr Klíka	1905ae1	Přidána možnost přidat fakturu, odstranit fakturu a editovat již přidanou fakturu, je také nyní možné přidat hned fakturu další. Vytvořen v...	2016-12-28
Petr Klíka	2ace353	editace a mazání poznámek nyní funkční, vytvořena šablona pro vkládání nové faktury, zatím není funkční, další krok bude nasadit date...	2016-12-28
Petr Klíka	5057df0	výpis dlužníků + přidány modály pro zobrazení kompletního info	2016-12-22
Petr Klíka	bc16e80	Úpravy databáze a nové pole dlužníci	2016-12-22
Petr Klíka	54e3112	Aplikace umí vytvořit klienta, rozlišuje přihlášeného uživatele podle dvou rolí, umí vytvořit nový případ	2016-12-22
Petr Klíka	29c987d	První commit	2016-10-19

Picture 16: Bitbucket commits. Source: own processing

4.4.4 Designing of the database structure

I designed the database structure which follows basic normal forms. There were needed several tables for the Incase web application.

Ass names of the tables suggest, they represent what they really are.



Picture 17: Database structure. Source: own processing

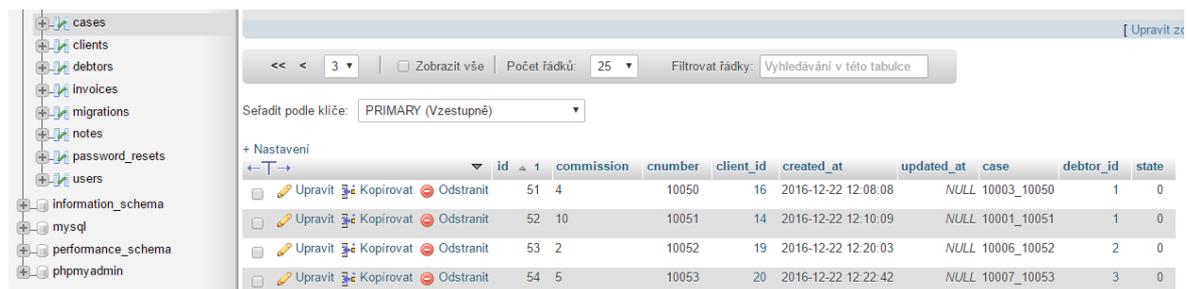
All tables were created with utf8_bit collation suitable for English and concurrently Czech characters.

The cases table structure holds foreign keys for clients and debtors and data inside the table are atomic which makes it formatted according to basic normal forms. The `id` column is the primary key for the table. All columns except `created_at` and `updated_at` must not be NULL which prevents non-insertion and provides reliability of the web application. Other tables are designed completely in the same way.



#	Název	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Další
1	id	int(10)		UNSIGNED	Ne	Žádná	AUTO_INCREMENT
2	commission	varchar(255)			Ne	Žádná	
3	cnumber	int(11)			Ne	Žádná	
4	client_id	int(10)		UNSIGNED	Ne	Žádná	
5	created_at	timestamp			Ano	CURRENT_TIMESTAMP	
6	updated_at	timestamp			Ano	NULL	
7	case	varchar(255)			Ne	Žádná	
8	debtor_id	int(10)		UNSIGNED	Ne	Žádná	
9	state	int(255)			Ne	Žádná	

Picture 18: Cases table structure. Source: own processing



	id	commission	cnumber	client_id	created_at	updated_at	case	debtor_id	state
Upravit	51	4	10050	16	2016-12-22 12:08:08	NULL	10003_10050	1	0
Upravit	52	10	10051	14	2016-12-22 12:10:09	NULL	10001_10051	1	0
Upravit	53	2	10052	19	2016-12-22 12:20:03	NULL	10006_10052	2	0
Upravit	54	5	10053	20	2016-12-22 12:22:42	NULL	10007_10053	3	0

Picture 19: Cases table data samples. Source: own processing

Tables in the database are generated in Laravel framework using Migrations which are like version control system for a database allowing easily to modify and share application's database schema. Migrations are typically paired with Laravel's schema builder to easily build the application's database schema.

Firstly, I generate a migration using this command in the command line project directory: `php artisan make:migration create_cases_table` which immediately placed a new generated migration file in the database/migrations directory. Each migration file name contains a timestamp which allows Laravel to determine the order of the migrations. The `--create` property can be added to migration command to basically specify the table name, it will pre-fill generated migration with the specified table: `php artisan make:migration create_cases_table --create=cases` [8].

```

<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateCasesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('cases', function (Blueprint $table) {
            $table->increments('id');
            $table->string('company');
            $table->string('dsurname');
            $table->string('dfirstname');
            $table->string('street');
            $table->string('town');
            $table->integer('postcode');
            $table->string('ico');
            $table->string('commission');
            $table->integer('cnumber');
            $table->integer('client_id')->unsigned();
            $table->foreign('client_id')->references('id')->on('clients');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('cases');
    }
}

```

This is the sample of generated migration file for cases table. I added basic table columns specifications to the schema. After schema was done, I executed command: [php artisan migrate](#) which transferred the schema into the real database, created table and all its dependencies. It may happen that you want to add additional column to the database which

happened in my case at least one time so I had to create another migration which adds or modify column and then execute the command again.

4.4.5 Scaffolding technique

Laravel provides a possibility to scaffold and prepare a simple skeleton for the application. In this case, I used the scaffolding technique to create a skeleton for the whole authentication process. The command should be executed at the beginning of the development to stand by correct programming practices.

```
php artisan make:auth
```

This command provides a quick way to scaffold all routes and views which are needed for authentication. It generates HomeController which handles post-login requests to the application's dashboard. The base view which uses Bootstrap CSS framework for the application is generated by the command and stored in the resources/views/layouts directory. Other necessary views like login and registration views with authentication forms are also generated and stored in the resources/views/auth directory [8].

HomeController

```
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class HomeController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('auth');
    }

    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        return redirect('/cases');
    }
}
```

The HomeController in its construct function uses an authentication middleware which control the login process, the index function of the controller redirects to the route /cases which is specified in the route file web.php in the routes directory.

4.4.6 Routing

Each Laravel route is defined in the route file web.php stored in the routes directory which is automatically loaded by the Laravel framework. A route basically accepts a URI and provides an action. The web.php file defines all routes which are used by the web application. These routes are assigned the web middleware group, which provides features like session state and CSRF protection [8].

Each HTML form which provides a post, put or delete action must include the CSRF token field which provides a CSRF protection. Otherwise the request will be rejected. This simplified form for example points to the POST route /add-note-information in the web.php file which accepts data from the inputs and then provides insert to the notes table in the database.

```
<form method="POST" action="{{ url('/add-note-information') }}">
    {{ csrf_field() }}
</form>
```

HomeController has specified in its index function a route /cases which serves as a landing page of the Incase application. /cases route is specified as a GET route which should in case that everything is done correctly return a view and display selected data from the database to the client of the web application [8].

route /cases

```
Route::get('/cases', function () {
    if (Auth::check()) {
        $cases = DB::table('cases')
            ->leftjoin('debtors', 'cases.debtor_id', '=', 'debtors.id')
            ->leftjoin('clients', 'cases.client_id', '=', 'clients.id')
            ->select('cases.id AS case_id', 'cases.case', 'cases.state', 'clients.*', 'debtors.*')
            ->orderBy('cases.id', 'asc')
            ->get();
        return view('cases', ['cases' => $cases]);
    } else {return 'Permissions denied';}
});
```

The route action firstly checks if the user is authenticated, if not it simply displays an error “Permissions denied”. If yes, it performs a select of data from the cases table from the database left joined on two additional tables debtors and clients, ordered by id of the case ascendingly. Data are then stored in php variable \$cases. At the end, it returns a view which is basically an HTML file supported by blade templating system and all selected data by the SQL query and stored in the variable are sent to the view.

4.4.7 Blade views

The primary benefit of using Blade templating system is the possibility to create a master view which holds all main elements of the web application which mostly do not change during the run (styles, scripts, title, description, footer, navigation bar, etc.). I created a master view called app.blade.php in the resources/views/layouts directory which uses Bootstrap CSS framework and imported JS extensions and JS scripts programmed on my own. Then, it is necessary to specify an area where to put the content of the web application by “@yield”. In my case, I created the @yield called content between the navigation bar and footer. The @yield('content') represents each view which extends the master view.

Code placed between navigation bar and footer:

```
<div class="container">
  <div class="row">
    @yield('content')
  </div>
</div>
```

In the navigation bar of the master view there is also specified an access of different roles to various links. In this case, logged user as the administrator of the application has access to register a new user of the application and to modify user information and roles.

Code placed in the navigation bar of the master view:

```
@if (Auth::user()->permission == 1)
  <li><a href="{{ url('/register') }}">Register New User</a></li>
  <li><a href="{{ url('/settings') }}">Settings</a></li>
@endif
```

After requesting the /cases GET route, the function returns view cases which inherits from the master view. So, the master view is loaded and instead of the `@yield('content')`, there is displayed the real content on the cases view.

Blade templating system gives the possibility of using basic php statements like loops and conditional statements right inside the blade file.

If statements

The if statements may be constructed using `@if`, `@elseif`, `@else` and `@endif` directivities. These directivities execute same actions like PHP conditional statements [8].

```
1. @if (count($records) === 1)
2.     I have one record!
3. @elseif (count($records) > 1)
4.     I have multiple records!
5. @else
6.     I don't have any records!
7. @endif
```

Cycles

Blade templating system also provides directivities for working with PHP loop structures. Again, they work identically to their PHP counterparts [8].

```
1. @for ($i = 0; $i < 10; $i++)
2.     The current value is {{ $i }}
3. @endfor
4.
5. @foreach ($users as $user)
6.     <p>This is user {{ $user->id }}</p>
7. @endforeach
8.
9. @forelse ($users as $user)
10.    <li>{{ $user->name }}</li>
11. @emptyempty
12.    <p>No users</p>
13. @endforelse
14.
15. @while (true)
16.    <p>I'm looping forever.</p>
17. @endwhile
```

Comments

Blade templating system provides a possibility to create well readable comments. The advantage of Blade comment is that they are not rendered and the comment will not be included in the HTML returned by the application [8].

```
1. {{!-- This comment will not be present in the rendered HTML --}}
```

cases.blade.php

```
1. @extends('layouts.app')
2. @section('content')
3. <div class="col-md-12">
4.     @if (session('status'))
5.     <div class="alert alert-success">
6.         {{ session('status') }}
7.     </div>
8.     @endif
9.     <a role="button" id="showSearchInputs" class="btn btn-
    primary" onclick="showSearch()"><span class="glyphicon glyphicon-
    search"></span> Search</a>
10.     <span id="searchInputs">
11.         <input type="text" id="myInput" class="myInput" onkeyup="searchReferences()"
    placeholder="Search For References">
12.         <input type="text" id="myInput2" class="myInput" onkeyup="searchDebtors()" pl
    aceholder="Search For Debtors">
13.         <input type="text" id="myInput3" class="myInput" onkeyup="searchAddresses()"
    placeholder="Search For Addresses">
14.     </span>
15.     <a role="button" href="{{ url('/add-case') }}" class="btn btn-primary pull-
    right"><span class="glyphicon glyphicon-file"></span> Add New Case</a>
16.     <table id="myTable" class="cases" data-search="true">
17.         <tr class="header">
18.             <th>Reference</th>
19.             <th>Debtor</th>
20.             <th>Address</th>
21.             <th>Client</th>
22.             <th>State</th>
23.         </tr>
24.         @foreach ($cases as $case)
25.             <tr case="{{ $case->case_id }}" debtor="{{ $case-
    >id }}" onclick="setSession(this)">
26.                 <td>{{ $case->case }}</td>
27.                 <td>{{ $case->dfirstname }} {{ $case->dsurname }}</td>
28.                 <td>{{ $case->street }}, {{ $case->town }}, {{ $case->postcode }}</td>
29.                 <td>{{ $case->name }}</td>
30.                 <td id="caseState" state="{{ $case->state }}">@if($case-
    >state == 0) opened @else closed @endif</td>
31.             </tr>
32.         @endforeach
33.
34.         {{ csrf_field() }}
35.         <script type="text/javascript">
36.             function setSession(tr) {
37.                 var caseid = $(tr).attr('case');
38.                 var debtorid = $(tr).attr('debtor');
39.                 $.ajax({
40.                     headers: {'X-CSRF-TOKEN': $('meta[name="csrf-
    token"]').attr('content')},
41.                     data: {case: caseid, debtor: debtorid},
42.                     url: '/setSession',
43.                     type: 'POST',
44.                     dataType: 'JSON',
45.                 });
46.                 window.location.href = "/case-info";
47.             }
48.         </script>
49.     </table>
50. </div>
51. @endsection
```

cases.blade.php annotation

Line: 1

This line indicates that file inherits from the master view names app.

Line: 2-51

These lines marks code inside and close it into the section called “content” which means that whenever the master view contains `@yield('content')`, it will immediately put the code instead of it.

Line: 4-8

I there is a flash message present, it will display it. This message is mostly used to inform a client of the application about some action which occurred (the client has been added successfully).

Line: 10-14

Search inputs which calls on key up event a custom JS functions which provides searching in the table of displayed cases.

Line: 24-32

A foreach loop statement which goes through all selected cases from the database and display an individual information about the case.

Line: 35-48

An AJAX call developed to set a session which contains information about chosen case. The information is then used to display the case in another view.

4.4.8 Application conclusion

Incase web application was developed to serve in a Czech company which holds a business in the field of enforcement and management of debts based on requirements of the founder. The web application can be now used by the company without any restrictions. Incase was developed according to theory described by the author of this Bachelor thesis.

5 Conclusion

The goal of this bachelor thesis was to analyse and introduce current trends of programming and coding a web application using modern practices and technologies. Comparing of selected ones to each other and based on the analysis choose a suitable solution for the development process.

The idea of development a web application focused on field of debt enforcement arose based on requirements of the founder of Czech company which holds a business in that area. It is necessary to keep records about clients, debtors, invoices and payments regarding to this type of business. Records can be stored in many ways but nowadays it is very comfortable and modern to store most of the data on remote servers to be able to access them anytime and from anywhere just by using basic technologies like desktop PC, tablet or smartphone. Data are shared across the employees of the company and can be manipulated independently. This is the main reason for development web/business applications. The productivity and effectiveness of the company raises.

The process of development of the web application was not complicated but of course, sometimes I have had to deal with small issues which occurred during the development. Programming, Algorithm development and web design university courses really helped to understand some problematics which I met during the practical part. I solved most of the issues by using the internet and visiting web pages related to the problematics mainly StackOverflow and Laravel.io. I evaluated that necessary part of programming and coding is the community.

The real application is now being used by the company. For now, it holds just few records. There could be more features, it could send automatically emails or it could be connected to the SMS gateway which could be sending automatically payment reminders to debtors or information about the debts to clients. This would certainly provide greater efficiency.

6 References

- [1] Internet World Stats. *Internet World Stats* [online]. [cit. 2017-02-13]. Available from: <http://www.internetworldstats.com/emarketing.htm>
- [2] **PANKRÁČ, Miroslav**. PHP a MySQL bez předchozích znalostí. Cpress, 2013. ISBN 978-80-251-1758-3.
- [3] *Jak psát web* [online]. [cit. 2017-02-16]. Available from: <https://www.jakpsatweb.cz/server/http-protokol.html>
- [4] *Python* [online]. [cit. 2017-02-20]. Available from: <https://wiki.python.org/moin/PythonVsPhp>
- [5] *The TIOBE Programming Community index* [online]. In: . [cit. 2017-02-20]. Available from: <http://www.tiobe.com/tiobe-index/>
- [6] **KOSEK, Jiří**. *PHP - tvorba interaktivních webových aplikací*. Grada Publishing, spol., 1999. ISBN 80-7169-373-1.
- [7] Laravel. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2017-02-20]. Available from: https://cs.wikipedia.org/wiki/Laravel#Z.C3.A1kladn.C3.AD_funkcionalita
- [8] *Laravel* [online]. [cit. 2017-02-20]. Available from: <https://laravel.com>
- [9] *Opensource* [online]. [cit. 2017-02-20]. Available from: <https://opensource.com/business/16/6/which-php-framework-right-you>
- [10] *Spyre studios* [online]. [cit. 2017-02-20]. Available from: <http://spyrestudios.com/top-php-frameworks-2016/>
- [11] Tvorba databází v MySQL - I [online]. In: . [cit. 2017-02-20]. Dostupné z: <http://www.abclinuxu.cz>
- [12] **STEPHENS, Ryan K., Ronald R. PLEW a Arie JONES**. Naučte se SQL za 28 dní: [stačí hodina denně]. Brno: Computer Press, 2010. ISBN 978-80-251-2700-1.
- [13] *Indeed* [online]. [cit. 2017-02-20]. Available from: <https://cz.indeed.com/?r=us>
- [14] *Apache Friends* [online]. [cit. 2017-02-21]. Available from: <https://www.apachefriends.org>
- [15] *Db- engines* [online]. [cit. 2017-02-21]. Available from: <http://db-engines.com/en/ranking>
- [16] *W3 School* [online]. [cit. 2017-02-21]. Available from: <http://www.w3schools.com/>

- [17] *Code anywhere* [online]. [cit. 2017-02-21]. Available from: <https://blog.codeanywhere.com/most-popular-ides-code-editors/>
- [18] *Atlassian* [online]. [cit. 2017-02-21]. Available from: <https://www.atlassian.com/git/tutorials/what-is-version-control>
- [19] Git. In: Git [online]. [cit. 2017-02-21]. Dostupné z: <https://git-scm.com/about>
- [20] Bootstrap. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2017-02-27]. Available from: <https://cs.wikipedia.org/wiki/Bootstrap>
- [21] MySQL [online]. [cit. 2017-03-07]. Available from: <https://www.mysql.com/why-mysql/topreasons.html>
- [22] **KOSEK, Jiří**. HTML: tvorba dokonalých WWW stránek : podrobný průvodce. Praha: Grada, 1998. ISBN 8071696080.

7 List of used pictures

Picture 1 - Architektura klient/server [2]

Picture 2 - The TIOBE Programming Community index. Source: [5]

Picture 3 - PHP Framework Popularity in Personal Projects. Source: [10]

Picture 4 - Languages ranked by number of programming jobs. Source: [13]

Picture 5 - Example of XAMPP tool. Source: own processing

Picture 6 - Login. Source: own processing

Picture 7 - Landing page. Source: own processing

Picture 8 - Administrator. Source: own processing

Picture 9 - Clients. Source: own processing

Picture 10 - Client info. Source: own processing

Picture 11 - Add new client. Source: own processing

Picture 12 - Client update. Source: own processing

Picture 13 - Client update. Source: own processing

Picture 14 - Client update. Source: own processing

Picture 15 - Case details. Source: own processing

Picture 16 - Bitbucket commits. Source: own processing

Picture 17 - Database structure. Source: own processing

Picture 18 - Cases table structure. Source: own processing

Picture 19 - Cases table data samples. Source: own processing