UNIVERZITA KOMENSKÉHO V BRATISLAVE

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Webová aplikácia pre správu súťaže FIRST LEGO League

Bakalárska práca

2017 **Matej Vilk**

UNIVERZITA KOMENSKÉHO V BRATISLAVE

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Webová aplikácia pre správu súťaže FIRST LEGO League

Bakalárska práca

Študijný program: Aplikovaná informatika
Študijný odbor: 2511 Aplikovaná informatika
Školiace pracovisko: Katedra aplikovanej Informatiky
Školiteľ: Mgr. Pavel Petrovič, PhD.

Bratislava 2017 **Matej Vilk**

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Matej Vilk
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: aplikovaná informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Webová aplikácia pre správu súťaže FIRST LEGO League
Web application for the FIRST LEGO League contest managment

Cieľ: Aktuálne používané stránky fll.sk majú 6 rokov a nevyhovujú dnešným očakávaniam. Je potrebné vybudovať od začiatku nový systém, či ako rozšírenie nejakého CMS alebo ako samostatnú webovú aplikáciu. V súčasnom systéme je niekoľko modifikácii štandardnej správy používateľov, ktorá je premenená na registráciu tímov do súťaže - čo má svoje nevýhody a očakávame čistejšie riešenie. Študent analyzuje súčasný stav a nové potreby správy súťaže (napríklad, možnosť párovania trénerov a tímov, vyhľadávanie tímu pre nezaradeného súťažiaceho, zobrazovanie prihlásených tímov na mape).

Literatúra: Petrovic P., Onacilová D., Svetlík J. (2010) Skúsenosti s prípravou sútaze v stavbe a programovaní robotov FIRST LEGO League z pohladu organizátora, trénera a rozhodcu, Didinfo 2010, Banská Bystrica, April 8-9 2010.
National Aeronautics and Space Administration, www.nasa.gov.

Vedúci: Mgr. Pavel Petrovič, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.

Dátum zadania: 03.10.2016

Dátum schválenia: 17.10.2016 doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Čestné vyhlásenie

Čestne prehlasujem, že som túto bakalársku prácu vypracoval samostatne s použitím uvedených zdrojov.

V Bratislave .....

Poďakovanie

Touto cestou by som chcel poďakovať môjmu školiteľovi a konzultantovi Mrg. Pavlovi Petrovičovi, PhD., za pomoc, ochotu, cenné rady a pripomienky pri tvorbe mojej záverečnej práce.

Abstrakt

Cieľom tejto bakalárskej práce je vytvoriť novú modernú webovú aplikáciu slúžiacu na prezentáciu a správu najväčšej robotickej súťaže FIRST LEGO League. Riešení na tento problém existuje mnoho, veľa z nich vyvíjaných mnohopočetnými profesionálnymi tímami. Napriek tomu, ani jedno z riešení plnohodnotne nespĺňa naraz všetky požiadavky.

Moja aplikácia sa zamerala hlavne na slabú stránku momentálnej aplikácie a to prácu s turnajmi a tímami, ktoré sa prihlasujú na turnaje. Administrátori majú jednoduchý prístup k najdôležitejším nastaveniam ako nastavenia dátumov a aktívnych regiónov pre aktuálny ročník. Jednoducho môžu modifikovať existujúce alebo pridať ďalšie položky. Pri archivácii ročníka je potrebné zadať len rok nového ročníku a zvyšok procesu je plne automatizovaný. Všetky dáta sa nám prenesú na určené miesta pre archívne turnaja a nový ročník začne s regiónmi a dátumami z toho predchádzajúceho. V prípade registrácie tímu do turnaja aplikácia funguje podobne sama, vytvorí požiadavky na zaslanie faktúry a stavebnice pre daný tím. Tieto požiadavky následne spravuje poverená osoba a vypĺňa tak ako vybavuje jednotlivé požiadavky.

Správa noviniek a užívateľov je samozrejmosťou a takisto sa nachádza pre organizátorov v aplikácii. Podstatným faktorom systému je aj úprava textov dôležitých pre zúčastňujúce sa tímy. Tieto texty obsahujú zadania úloh, informácie a kontaktné osoby pre jednotlivé regióny a zoznam tímov v danom ročníku, ktorý berieme z našej databázy. Pre každý ročník, na základe týchto textov generujeme dynamické menu na prezenčnej stránke.

Kľúčové slová: *CMS, framework, Laravel, webová aplikácia, robotická súťaž*

Abstract

The goal of this thesis is to create a new and modern web application used for presentation and administration purposes of the biggest robotic competition FIRST LEGO League. Many solutions exist, a lot of them being developed by big companies with numerous profesionals. However, none of those fully satisfies the application needs.

Main goal of my application is aimed towards the weak spot of the current application which is maintaining tournaments and teams registring to tournaments. Administrators have an easy access to the most important settings, such as dates and active regions for current year. They can easily modify existing entries or add new ones. In order to archivate the current year the only neccessary thing to do is to fill out the year that will be used next, the rest is fully automatized. All the data is moved to intended places for archive tournaments and new year starts with prepared regions and dates, used from the previous year. In case of team registration application is too self opearable and creates requests for invoices and kits. Those requests are maintaned by authorised personnel.

News and user maintenance is a must and therefore can also be found in application for organizers. Another necessity is editing of texts important for participants. Those texts consist of challenge task, region information, contact informations and list of participating teams generated from our database. Dynamic menu is generated based on those texts.

Keywords: *CMS, framework, Laravel, web application, robotic competition*

Obsah

ÚVOD **12**

1 SLOVNÍK POJMOV **13**

2 ANALÝZA TECHNOLÓGIÍ **14**

2.1 Technológie FLL vo svete 14

2.2 Laravel 15

2.2.1 O Laraveli 15

2.2.2 MVC 16

2.2.2.1 Model 16

2.2.2.2 View 17

2.2.2.3 Controller 17

2.2.3 Bezpečnosť 18

2.2.3.1 SQL Injection 18

2.2.3.2 Cross-Site Request Forgery 18

2.2.3.3 Cross-Site Scripting 19

2.2.4 Eloquent 19

2.2.4.1 Snake Case 21

2.2.5 Facades 21

2.2.6 Blade 22

2.2.7 Routes 23

2.3 Bootstrap 23

2.4 PHP 7.0 24

2.5 jQuery 24

3 POŽIADAVKY NA SYSTÉM **26**

3.1 Prezenčná stránka 26

3.1.1 Návštevník 26

3.1.1.1 Registrácia/Prihlásenie 26

3.1.2 Registrovaný užívateľ 26

3.1.2.1 Stať sa trénerom 26

3.1.2.2 Hľadať trénera 26

3.1.2.3 Založenie inzerátu 27

3.1.3 Tréner 27

3.1.3.1 Hľadanie tímu 27

3.1.3.2 Založenie inzerátu 27

3.1.3.3 Založenie tímu 27

3.1.3.4 Administrácia tímu 27

3.1.3.5 De-registrácia tímu 27

3.1.3.6 Absencia/Potvrdenie účasti členov tímu 28

3.2 Administračná stránka 28

3.2.1 Administrátor 28

3.2.1.1 Nastavovanie dátumov 28

3.2.1.2 Administrácia tímov 28

3.2.1.3 Mazanie tímov 28

3.2.1.4 Tvorba nových/Úprava/Mazanie starých užívateľov . . . 28

3.2.1.5 Priraďovanie/Odoberanie administratívnych rolí užívateľov 29

3.2.2 Manažér 29

3.2.2.1 Tvorba/Úprava článkov 29

3.2.2.2 Zasielanie stavebníc 29

3.2.2.3 Kontrola faktúr 29

3.2.3 Ekonóm 29

3.2.3.1 Práca s faktúrami 29

3.2.4 Distribútor stavebníc 29

3.2.4.1 Zasielanie stavebníc 29

3.2.5 Zahraničný partner 30

3.2.5.1 Export tímov 30

4 NÁVRH **31**

4.1 Používatelia 31

4.1.1 Používateľské role 31

4.2 Používateľské rozhrania 33

4.2.1 Rozhranie administrátorského prostredia 33

4.2.2 Rozhranie prezenčného prostredia 34

4.3 Databáza 34

4.3.1 Návrh databázy 34

4.3.2 Archivovanie 37

4.4 Štruktúra aplikácie 37

5 ZÁVER **39**

ÚVOD

First Lego League je najväčšia robotická súťaž pre mládež vo veku 9 až 16 rokov. Jej účastníkmi sú tímy po celom svete. Každý rok je potrebné vyriešiť novú úlohu, rozdelenú na štyri časti: Robot Design, Robot Game, Prezentácia výskumného projektu, Tímová práca. Súťaž tak nielen preverí programátorské schopnosti tímu, ale aj jeho schopnosť spolupracovať, či vymyslieť riešenie problému výskumnej témy. Súťažiaci na Slovensku si môžu vybrať zo 7 regionálnych turnajov - v Petržalke, v Poprade, v Žiline, v Žiari nad Hronom, v Bratislave, v Košiciach a v Banskej Bystrici. Nakoľko však záujem o súťaž vzrastá pribúdajú ďalšie regióny.

Stránka fll.sk slúži hlavne súťažiacim trénerom na prihlásenie svojho tímu a členov do prebiehajúceho turnaja. Tréneri tu nájdu potrebné materiály k turnaju, zadania úloh, adresy miest konania a mnoho ďalších potrebných informácií. Pre návštevníkov, sa na stránke nachádza galéria z takmer všetkých doteraz uskutočnených ročníkov a kompletný archív turnajov.

Existujúci systém prezenčnej stránky a jej administrátorského rozhrania je v nezmenenom stave už vyše 6 rokov. Chýbajú mu základné princípy myšlienky moderného webu. Jedným z hlavných problémov je neprístupnosť na mobilných zariadeniach, zastaralý dizajn a komplikovanosť administračných úkonov.

Cieľom mojej práce je tieto stránky zmodernizovať a zjednodušiť tak prístup pre návštevníkov a trénerov, ako aj pre ľudí, ktorí stoja za jej organizovaním. Množstvo administratívnych úkonov vyžadovalo vyššiu technickú zručnosť, čomu som chcel v novom systéme predísť. Nakoľko sú obe časti systému plne responzívne, nie je problém takéto a podobné činnosti v prípade potreby vykonávať na mobilných zariadeniach.

Dostupných je mnoho redakčných systémov, ktoré by spĺňali jednotlivé požiadavky na aplikáciu, no ani jeden z nich nekombinoval všetky naraz. Z tohto dôvodu som sa rozhodol vyvinúť vlastný systém, ktorý by vyhovoval všetkým požiadavkám

a v prípade potreby sa dal ľahko modifikovať. Podrobnejšiemu porovnaniu momentálneho riešenia, môjho a podobného dostupného sa budem venovať v ďalších kapitolách.

Práca je rozdelená na niekoľko častí. V prvej kapitole sa budem venovať požiadavkám na aplikáciu, ktoré opisujú vyžadované zmeny potrebné implementovať alebo z momentálneho systému vylepšiť. Nasledujúca kapitola obsahuje technológie využité na tvorbu aplikácie, hlavne framework Laravel, na ktorom je aplikácia postavená. V poslednej kapitole som podrobne rozpracoval systém prezenčnej stránky, doplnil základné princípy moderného webu a implementáciu môjho navrhovaného systému.

1. SLOVNÍK POJMOV

Framework - Poskytuje nám základnú programovaciu kostru a uľahčuje nám mnoho vykonávaných operácií.

CMS - Content Managment System, systém na správu digitálneho obsahu.

SQL - Structured Query Language, jazyk vytvorený na správu dát v relačných databázových systémoch.

SEO - Search Engine Optimization alebo Optimalizácia pre vyhľadávače je zabezpečenie vyhľadania vo webových vyhľadávačoch na základe slov relevantných našej stránke.

Rollback - Operácia, ktorá vráti databázu do predchádzajúceho stavu v prípade neočakávanej chyby.

Closure - Uchováva pre nás záznam premenných zadefinovaných vo funkcii. Tým pádom môžeme, k týmto vpremenným pristupovať aj mimo dajne funkcie.

2. ANALÝZA TECHNOLÓGIÍ

V nasledujúcich častiach sa budem venovať porovnaniu redakčných systémov iných stránok FLL, využitým technológiám, ktoré sa v projekte nachádzajú, tomu čo je to MVC, fungovanie a súčasti Laravel Frameworku, na ktorom je aplikácia stavaná.

2.1 Technológie FLL vo svete

V tejto časti sa pozrieme na redakčné systémy, ktoré používajú niektoré FLL súťažné stránky po svete a zameriamae sa na ich najväčšie nevýhody.

Južná Afrika/Austrália - Obe tieto stránky a mnohé iné využívajú populárny CMS Wordpress. Aj napriek jeho rozšírenosti však má Wordpress mnoho nedostatkov. Kvôli svojej popularite je útočiskom mnohých hackerov, predstavuje tak väčšie riziko pre prípadný útok. Wordpress takisto nepodporuje iný typ databázy ako MySQL, sme teda viazaný na tento typ a nemáme možnosť výberu[1].

Spojené štáty/Kanada - Severná Amerika využíva ako jedna z mála Drupal. Tento systém je známy svojou učebnou náročnosťou. Vytvorenie stránky v tomto systéme vyžaduje množstvo štúdie a prehľadávania v dokumentácii. Jeho sila sa skrýva v jeho jadre a nie je teda vhodný v prípade, že nemáme veľmi záujem toto jadro využívať a chceme vytvoriť vlastnú špecifickú aplikáciu[2].

Nový Zéland - Tento región má nasadený systém Joomla. Tento CMS obsahuje veľké množstvo dostupných rozšírení no žiaľ niektoré z nich sú spoplatnené. Navyše je veľa týchto rozšírení prístupných zadarmo v ostatných redakčných systémoch. Slabinou systému

[1] SAWYER, Maggie. 2016. *WordPress Users Beware: 19 Disadvantages of Using WordPress*. [online]. : 2016. [cit. 8.4.2013]. Dostupné na internete: `https://www.adngin.com/blog/blogging-best-practices/wordpress-users-beware-19-disadvantages-using-wordpress/` (Dátum prístupu 01/06/2017)

[2] ALBERT, J. 2015. *Drupal Pros and Cons: An In Depth Look*. [online]. : 2015. [cit. 8.4.2013]. Dostupné na internete: `https://www.mcdpartners.com/news/drupal-pros-and-cons/` (Dátum prístupu 01/06/2017)

je aj neschopnosť zvládnuť väčšie množstvo návštevníkov stránky[3].

Slovensko - Slovenské stránky momentálne používajú MODX CMS. Kľúčovým problém tohto systému ako aj ostatných je jeho zložitosť. Moja aplikácia má veľkú väčšinu nastavení predom definovaných v kóde a pre administrátorov a manažérov sú teda viditeľné len tie najdôležitejšie a najpotrebnejšie.

Pre technické porovnanie niektorých spomenutých CMS a mojej aplikácie máme technickú tabuľku porovnávajúcu niekoľko z ich vlastností.

	Laravel + Moja Aplikácia	**MODX**	**Wordpress**
Minimálna pamäť	9MB	64MB	32MB
Odporúčaná pamäť	50MB	256MB	128MB
E-mailové Protokoly	IMAP, SMTP, POP3	SMTP	X
Podporované VCS	Akýkoľvek	Git	X
Databázový model	Objektovo-orientovaný, Relačný	Objektovo-relačný	Objektovo-relačný

Tabuľka 2.1: Tabuľka porovnávajúca Laravel a moju aplikáciu, MODX a Wordpress

2.2 Laravel

2.2.1 O Laraveli

Laravel vznikol 9.júna 2011, kedy vyšla jeho beta verzia a o mesiac neskôr vyšla prvá verzia. Vytvoril ho Taylor Otwell s mienkou pokročilejšej alternatívy k framework-u CodeIgniter, ktorému v tom čase chýbala napríklad podstatná funkcia a to autentifikácia používateľov. Prvá verzia neobsahovala komponent Controller, čo Laravel-u bránilo v tom, aby bol plnohodnotný MVC framework. To však autor a komunita zmenili, keď v druhej verzii Laravel 2 Controller už doplnili[4]. Odvtedy framework prešiel početnými zmenami

[3]WILDIN, Robin. 2011. *Joomla: Advantages and Disadvantages of Choosing Joomla as Your CMS Solution*. [online]. : 2011. [cit. 8.4.2013]. Dostupné na internete: `http://www.socialtechnologyreview.com/articles/joomla-advantages-and-disadvantages-choosing-joomla-your-cms-solution` (Dátum prístupu 01/06/2017)

[4]O'BRIEN, Jesse. 2016. *A Brief History of Laravel*. [online]. : medium.com, 2016. [cit. 8.4.2013]. Dostupné na internete: `https://medium.com/vehikl-news/a-brief-history-of-`

a jeho momentálna stabilná verzia je 5.4.

Podľa hodnotení frameworkov je Laravel momentálne na siedmom mieste vo svetovom rebríčku[5], takže od svojho vzniku sa vyšplhal na popredné miesta a je jedným z najuznávanejších a najpoužívanejších frameworkov na tomto trhu.

Laravel už pri inštalácií prichádza s množstvom vbudovaných funkcií, najdôležitejším z nich sa budem venovať v ďalších kapitolách. Ak by sme si chceli náš projekt rozšíriť ďalšími balíčkami, ktoré vybudovala komunita použijeme na to Composer.

Veľkou súčasťou Laravel-u je aj Artisan, príkazový riadok Laravelu, ktorý sa nachádza v Lareveli od tretej verzie. Jeho prítomnosť budeme môcť vidieť aj v mnohých ukážkach nižšie.

2.2.2 MVC

MVC alebo Model-View-Controller je softvérový návrhový vzor, ktorý delí aplikáciu na 3 logické komponenty Model, View a Controller. Každá z týchto častí je stavaná na svoju špecifickú rolu a všetky sú na sebe navzájom závislé.

Táto architektúra je jedna z najpoúživanejších v obore vývoja webových aplikácií. Dôvodom je hlavne urýchlenie programovania veľkých projektov, ktoré majú tendenciu sa rozvíjať v budúcnosti[6]. Jednotlivé prvky sú rozdelené do samostatných tried vďaka čomu je kód aplikácie čistý a prehľadný.

Samozrejme zvoliť si tento typ vývoja nie je vždy vhodné. Pri malých projektoch alebo statických stránkach nepotrebujeme takúto veľkolepú škálovateľnosť. Pre naše účely je priam stvorený.

2.2.2.1 Model

Reprezentuje dátovú štruktúru aplikácie, či už v podobe databázy, súborov alebo aj obyčajných polí. Napríklad model Zákazníka môže byť namapovaný na tabuľku Zákazník

laravel-5d55970885bc (Dátum prístupu 05/11/2017)

[5]FRAMEWORKS, Hot. 2017. *Hodnotenie Framework-ov*. [online]. : 2017. [cit. 8.4.2013]. Dostupné na internete: http://hotframeworks.com/ (Dátum prístupu 03/03/2017)

[6]TUTORIALSPOINT. 2014. *MVC Framework Introduction*. [online]. : 2014. [cit. 8.4.2013]. Dostupné na internete: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm (Dátum prístupu 01/19/2017)

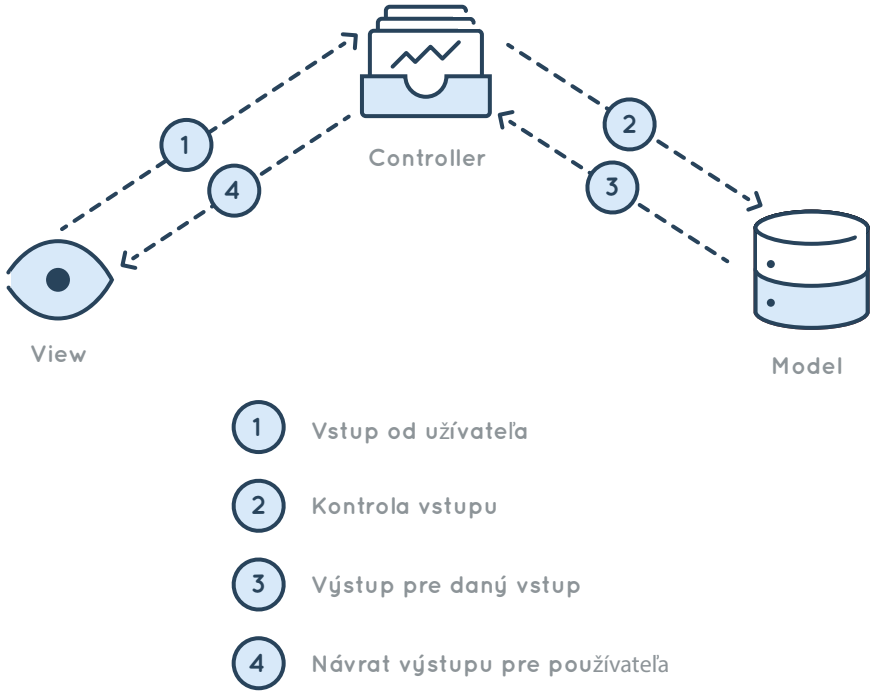
v databáze. Stará sa teda o riadky tohto modelu ich výber, zápis či mazanie.

2.2.2.2 View

Je to prezenčná stránka aplikácie, zjednodušene povedané, je to to čo vidíme ako návštevník stránky. View pre zákazníka, by teda obsahovalo všetky potrebné vstupy, s ktorými by užívateľ mohol následne pracovať a vypĺňať.

2.2.2.3 Controller

Controller je spojka medzi časťami Model a View, spracováva všetky prichádzajúce žiadosti od používateľa z View komponentu. Získava dáta z databázy
za pomoci Model komponentu. Tieto dve zložky následne spracuje a vráti späť do View komponentu. Controller by v našom príklade teda spracoval všetky vyplnené dáta z určitého Zákazníkovho formulára, vložil by ich do databázy za pomoci Model-u a znova ich vrátil na View zákazníka, napríklad v podobe faktúry o objednávke.



Obr. 2.1: Obrázok popisujúci Architektúru MVC

2.2.3 Bezpečnosť

Laravel obsahuje niekoľko bezpečnostných prvkov, ktoré sa starajú o nepriepusnosť nežiadaných vstupov a plynulý beh našej aplikácie. Samozrejme je na nás programátoroch, aby sme neurobili zbytočné chyby a aj na samotnom užívateľovi, aby si napríklad zvolil dostatočne silné heslo.

2.2.3.1 SQL Injection

Eloquent ORM (Object-relation mapping), jednoducho povedané zaručuje, aby SQL dopyty boli ošetrené, resp. aby sa celý vstup bral ako jeden ucelený reťazec. Týmto spôsobom znemožníme útočníkovi zaslať do našej databáźy nežiadúci vstup.

Na príklade nižšie môžeme vidieť, ako by útočník mohol získať všetky dáta z tabuľky *users* za pomoci jednoduchej podmienky "or 1 = 1", ktorá by sa vyhodnotila pravdivo. V prípade, ak by sme nemali vytvorenú zálohu databázy, mohlo by dôjsť k oveľa väčšiemu problému (obr. č.2.3).

```
SELECT * FROM users WHERE email = 'mail@mail.com' or 1 = 1
```

Obr. 2.2: Možný nežiadúci vstup, ktorý by obišiel autentifikáciu e-mailom a heslom

```
SELECT * FROM users WHERE email = 'mail@mail.com'; drop table users;
```

Obr. 2.3: Neošetrený query vstup, ktorý by nám zhodil tabuľku s používateľmi

```
SELECT * FROM users WHERE email = 'mail@mail.com or 1 = 1'
```

Obr. 2.4: Ošetrený vstup, v podobe jedného uceleného reťazca

2.2.3.2 Cross-Site Request Forgery

Za použitia CSRF token-ov znemožníme tretím stranám vyvolanie nežiadúcich požiadaviek v našich formulároch.

```
<form method="POST" action="">
    {{ csrf_field() }}
    ...
</form>
```

Obr. 2.5: Použitie csrs_field token-u vo formulári na znemožnie CSRF

Tento token je potrebný na správne fungovanie vo všetkých formulároch v aplikácii.

2.2.3.3 Cross-Site Scripting

Syntax Laravelu nám zabezpečí, že všetky HTML tagy, ktoré by sme zaslal za pomoci View komponentu budú nahradené *escape* tagmi.

Na príklade nižšie môžeme vidieť funkčnosť tohto prvku. Kde sa znaky < a > nahradia textom &lt a &gt z anglického "less than" a "greater than".

```
<script>alert("SPAMSPAM")</script>
```

Obr. 2.6: Skript, ktorý by nám zavolal JavaScript funkciu alert()

```
"&lt;script&gt;alert("SPAM SPAM")&lt;/script&gt";
```

Obr. 2.7: Skript z obr. č. 2.6 s nahradenými < a > značkami

Toto je samozrejme len názorná ukážka, útočnik by mohol situáciu využiť aj oveľa horším spôsobom, napríklad na krádež osobných údajov používateľa[7].

2.2.4 Eloquent

Poskytuje pohodlný prístup k databáze. Každej z tabuliek prislúcha Model, slúžiaci na interakciu s danou tabuľkou. Tieto modely nám za pomoci dopytov umožňujú záznamy

[7]GILMORE, W. Jason. 2015. *Laravel 5's Key Security Features*. [online]. : easylaravelbook.com, 2015. [cit. 8.4.2013]. Dostupné na internete: `http://www.easylaravelbook.com/blog/2015/07/22/laravel-key-security-features/` (Dátum prístupu 02/02/2017)

čítať, upravovať a zapisovať.

Najjednoduchší spôsob tvorby modelu je za pomoci Artisan príkazu make:model. Ak by sme chceli vytvoriť migráciu spolu s modelom, stačí do príkazu pridať prepínač –migration alebo -m.

```
php artisan make:model User --migration
php artisan make:model User -m
```

Obr. 2.8: Vytvorenie modelu *User* spolu s migráciou

Migrácie nám slúžia na modifikovanie schémy databázy cez PHP bez nutnosti prístupu do databázového rozhrania. Nahradia nám teda manuálne a únavné pridávania riadkov do tabuľky.

Eloquent nám taktiež uľahčuje prácu s reláciami medzi jednotlivými tabuľkami. Zákazník môže mať napríklad viacero nákupov, poprípade komentárov alebo len jeden telefón či adresu. Slúžia nám na to nasledujúce relácie a ich príslušné zápisy a inverzné zápisy v Laravel-i:

- One To One - hasOne / belongsTo
- One To Many - hasMany / belongsTo
- Many To Many - belongsToMany
- Has Many Through - hasManyThrough
- Polymorphic Relations - morphMany
- Many To Many Polymorphic Relations - morphedByMany

```
class Zakaznik extends Model
{
    public function objednavka()
    {
        return $this->hasMany('App\Objednavka');
    }
}
```

Obr. 2.9: Názorná ukážka Modelu s použitím relácie *hasOne*

```
$objednavka = User::find(1)->objednavka;
```

Obr. 2.10: Využitie modelu z obr. č. 2.9 pre jednoduchý prístup na získanie objednávky používateľa

2.2.4.1 Snake Case

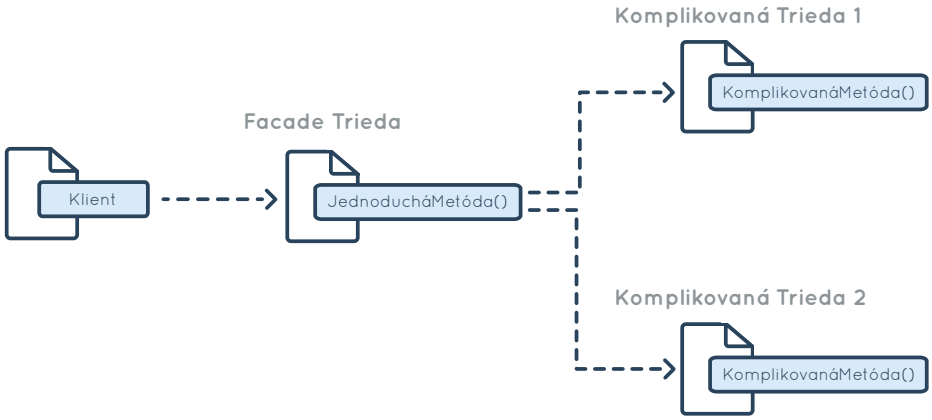
Pri vytvorení tabuľky "User" Laravel Eloquent bude automaticky počítať s tabuľkou "users", teda množné číslo nášho modelu spolu s malým prvým písmenom. Samozrejme nie je problém laravelu nastaviť tabuľku s vlastným názvom, postačí ak ho priradíme do premennej $table oute

```
protected $table = 'moje_objednavky';
```

Obr. 2.11: Priradenie tabuľky s vlastným názvom do premennej $table

2.2.5 Facades

Je to softvérový návrhový vzor zaužívaný v objektovom programovaní vo veľkom množstve zjednodušuje prehľadnosť kódu zaobalením komplexnejších tried do jednoduchších[8].



Obr. 2.12: Ilustračný obrázok návrhového vzoru

[8]LAVARYAN, Reza. 2015. *How Laravel Facades Work and How to Use Them Elsewhere*. [online]. : 2015. [cit. 8.4.2013]. Dostupné na internete: `https://www.sitepoint.com/how-laravel-facades-work-and-how-to-use-them-elsewhere` (Dátum prístupu 06/02/2017)

Na obrázku môžeme vidieť na prvom riadku komplexnejšie volanie metódy a na druhom jej zjednodušený zápis vďaka spomínanému návrhovému vzoru.

```
App::make('nazov_sluzby')->nazovMetody();
nazovSluzby::nazovMetody();
```

Obr. 2.13: Zápis tých istých funkcií bez a s návrhovým vzorom

2.2.6 Blade

Veľkou súčasťou Laravelu je šablónovací systém *Blade*. Dovoľuje nám písať PHP kód spolu s HTML kódom. V zjednodušenej podstate sú pri behu všetky *Blade* šablóny skompilované do obyčajného PHP. Na príklade môžeme vidieť základný ”master layout”, ktorý budeme môcť používať na všetky View, ktoré ďalej len rozšírime o konkrétne prvky na danom View.

```
<html>
    <head>
        <title>Moja aplikácia - @yield('title')</title>
    </head>
    <body>
        @section('menu')
            Časť vyhradená pre bočné menu.
        @show

        <div class="container">
            @yield('obsah')
        </div>
    </body>
</html>
```

Obr. 2.14: Master layout v Blade šablónovacom systéme

Ak by sme chceli uprostred HTML kódu zavolať určitú PHP funkciu poslúžia nám na to dve zložené zátvorky, teda “{{ php_funkcia() }}“.

2.2.7 Routes

Routes nám zaistia presmerovanie do príslušného Controller-a, takisto nám pomáhajú k SEO optimalizácii a vytvárajú čitateľnejšie URL linky. Na obrázkoch môžeme vidieť kód názorných "routes".

```
Route::get('fun', function () {
    return "Hello World";
    });
```

Obr. 2.15: Jednoduchý príklad tvorby takejto adresy obsahujúci URI a Closure

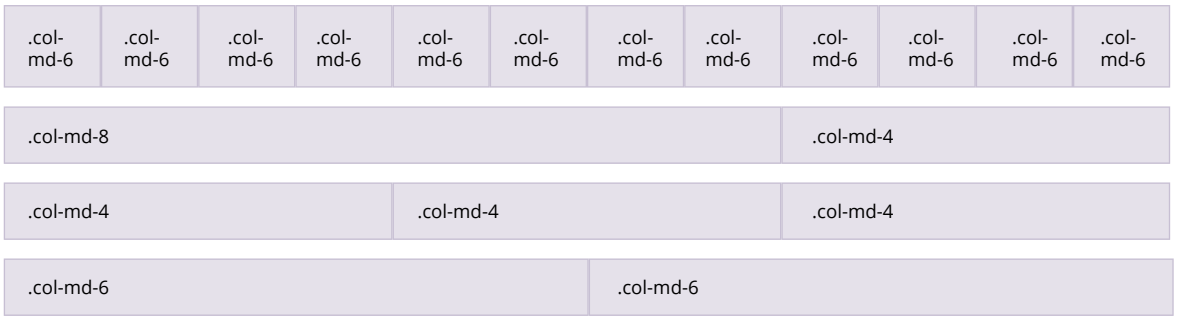
```
Route::get('user/{id}', function ($id) {
    return 'Užívateľ'.$id;
    });
```

Obr. 2.16: Routes môžu obsahovať aj parametre ako napríklad id používateľa

2.3 Bootstrap

Laravel je náš framework pre serverovú stránku aplikácie, Bootstrap je pre zmenu framework, ktorý nám bude pomáhať na strane klienta. Bootstrap je zadarmo a open-source, čo znamená že zdrojový kód je všetkým prístupný, môžeme ho slobodne používať, kopírovať a meniť ako chceme. Obsahuje množstvo pred programovaných funkcií a kaskádových štýlov, ktoré nám uľahčujú umiestňovanie jednotlivých objektov na stránke a ich plynulý prechod medzi rozlíšeniami, nakoľko sa v ňom veľmi dbá na mobilnú verziu stránok.

Podstata Bootstrapu funguje na systéme 12 stĺpcovej mriežky, zabezpečujúcu plnohodnotnú responzivitu, v ktorej sa nachádza náš obsah. Celý ho máme možnosť zaobaliť do *container* s pevnými css šírkami pre osobitné zariadenia alebo do *container-fluid*, kde nám šírku určuje veľkosť monitora, resp. jeho rozlíšenie. Stránku si môžeme rozdeliť rôznymi spôsobmi, no súčet stĺpcov vždy musí byť 12, zopár príkladov môžeme vidieť nižšie.

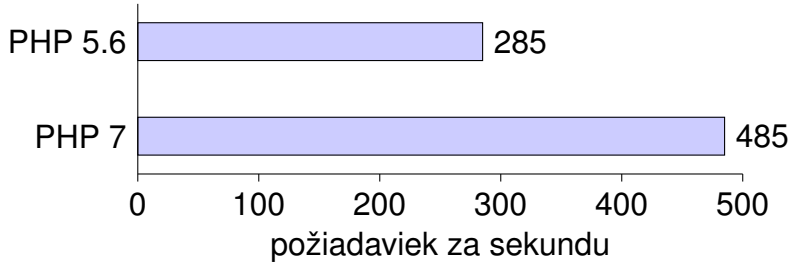


Obr. 2.17: Rôzne rozdelenia stránky v mriežkovom systéme bootstrap-u

Bootstrap v sebe samozrejme zahŕňa omnoho viac od tlačidiel, navigácie, formulárov až po ikonky[9].

2.4 PHP 7.0

Hlavným vylepšením oproti starším verziám je podstatné zrýchlenie. V mnohých prípadoch je to takmer dvojnásobné až dvojnásobné zrýchlenie.



PHP 7 obsahuje niekoľko ďalších zaujímavých zmien ako napríklad Spaceship operátor <=>, Null Coalescing, Anonymné funkcie, návratové typy funkcií. PHP 7 takisto odstraňuje množstvo zastaraných funkcií, ktoré však môžu spôsobiť ťažší prechod z PHP 5.x na PHP 7.

2.5 jQuery

Je multiplatformová JavaScript knižnica, vytvorená pre zjednodušenie, zrýchlenie a uľahčenie Javascript-u a HTML. Slúži nám na animáciu, spracovávanie udalostí, prácu s Ajax-om, ale aj manipuláciu HTML súborov.

[9]TEAM, Bootstrap Core. 2011. *Bootstrap, Oficiálna stránka*. [online]. : 2011. [cit. 8.4.2013]. Dostupné na internete: `http://getbootstrap.com/` (Dátum prístupu 10/09/2017)

```
$( 'span' ).each(function() {
    $( this ).addClass( 'nazov_triedy' );
});
```

Obr. 2.18: Názorná ukážka, ktorá pridá každému span tag-u triedu *nazov_triedy*

3. POŽIADAVKY NA SYSTÉM

V tejto kapitole sa budem venovať požiadavkám na aplikáciu dodaných od zadávateľa. Je rozdelená na dve časti tak ako aj samotný systém, prezenčná stránka a administrátorské rozhranie.

3.1 Prezenčná stránka

Užívateľ prezenčnej stránky je návštevník stránky, ktorý sa chce dozvedieť informácie o súťaži, zúčastniť sa jej alebo v prípade trénera administrovať svoj tím/y.

3.1.1 Návštevník

3.1.1.1 Registrácia/Prihlásenie

Každý návštevník bude mať možnosť prihlásiť alebo registrovať sa.

3.1.2 Registrovaný užívateľ

3.1.2.1 Stať sa trénerom

V prípade, že má užívateľ záujem o pozíciu trénera, vyplní dodatočný formulár s potrebnými informáciami.

3.1.2.2 Hľadať trénera

Registrovaný užívateľ bude mať možnosť vyhľadať trénera spomedzi ponúknutých inzerátov od trénerov.

3.1.2.3 Založenie inzerátu

Registrovaný užívateľ si môže založiť vlastný inzerát v ktorom bude hľadať trénera, inzerát môže taktiež upravovať alebo zmazať.

3.1.3 Tréner

3.1.3.1 Hľadanie tímu

Ak je používatel trénerom, má možnost kontaktovať tím spomedzi ponúknutých inzerátov, ktoré založili registrovaní užívatelia.

3.1.3.2 Založenie inzerátu

Tréner takisto môže založiť vlastný inzerát, ktorý môže následne upravovať alebo zmazať.

3.1.3.3 Založenie tímu

Používateľ s rolou trénera môže založiť tím (poprípade aj viac tímov) vyplnením potrebného formulára obsahujúceho dodatočné tímove informácie a informácie členov tímu.

3.1.3.4 Administrácia tímu

Pokiaľ je administrácia tímov ešte otvorená, tréner môže upravovať informácie svojho tímu ako aj jeho členov. Tieto dátumy má na starosti administrátor vo funkcionalite 2.1.1.

3.1.3.5 De-registrácia tímu

V prípade potreby môže tréner (v súlade s de-registračnými pravidlami) svoj tím z turnaja odhlásiť aj po uzávierke dátumov. Tím zostane označený ako de-registrovaný spolu s de-registračným dátumom.

3.1.3.6 Absencia/Potvrdenie účasti členov tímu

Ak sa člen tímu nebude môcť turnaja zúčastniť, tréner môže tohto člena označiť za neprítomného. V prípade, že sa jedná o trénera tímu, napísať náhradníka, ktorý bude pre tím dozorom.

3.2 Administračná stránka

Užívatelia administračnej stránky sa odlišujú od užívateľov prezenčných stránok, prihlasujú sa do špeciálneho administrátorského rozhrania a starajú sa o chod stránky, súťaže alebo v prípade zahraničných partnerov vyžadujú zoznam tímov pre daný ročník. Jednotlivé roli back-end užívateľov sa dajú pre jedného back-end užívateľa kombinovať.

3.2.1 Administrátor

3.2.1.1 Nastavovanie dátumov

Môže upravovať jednotlivé dátumy ako napríklad dátum registrácie tímov, registrácie do regiónu, úpravy tímov, úpravy členov tímu.

3.2.1.2 Administrácia tímov

Aj po uzavretí administrácie tímov pre používateľskú rolu “Tréner”, má administrátor možnosť tímy upravovať.

3.2.1.3 Mazanie tímov

Administrátor má možnosť zmazať tím.

3.2.1.4 Tvorba nových/Úprava/Mazanie starých užívateľov

Administrátor má ako jediný schopnosť vytvárať, mazať alebo upravovať back-end používateľov.

3.2.1.5 Priraďovanie/Odoberanie administratívnych rolí užívateľov

Administrátor môže jednotlivým užívateľom priradiť novú alebo odobrať starú administrátorskú rolu.

3.2.2 Manažér

3.2.2.1 Tvorba/Úprava článkov

Vytváranie článkov a ich úprava, nachádzajúcich sa na stránke FLL.sk v príslušnej sekcii / panely.

3.2.2.2 Zasielanie stavebníc

Tabuľkový prehľad o zaslaných stavebniciach, bude obsahovať jednoznačne viditeľný počet momentálne dostupných stavebníc.

3.2.2.3 Kontrola faktúr

Tabuľkový prehľad o zaplatených potrebných faktúrach.

3.2.3 Ekonóm

3.2.3.1 Práca s faktúrami

Rola používateľa ekonóm umožňuje prezerať potrebné faktúry.

3.2.4 Distribútor stavebníc

3.2.4.1 Zasielanie stavebníc

Tabuľkový prehľad o zaslaných stavebniciach, bude obsahovať jednoznačne viditeľný počet momentálne dostupných stavebníc.

3.2.5 Zahraničný partner

3.2.5.1 Export tímov

Exportovať a stiahnuť zoznam tímov vo formáte .csv.

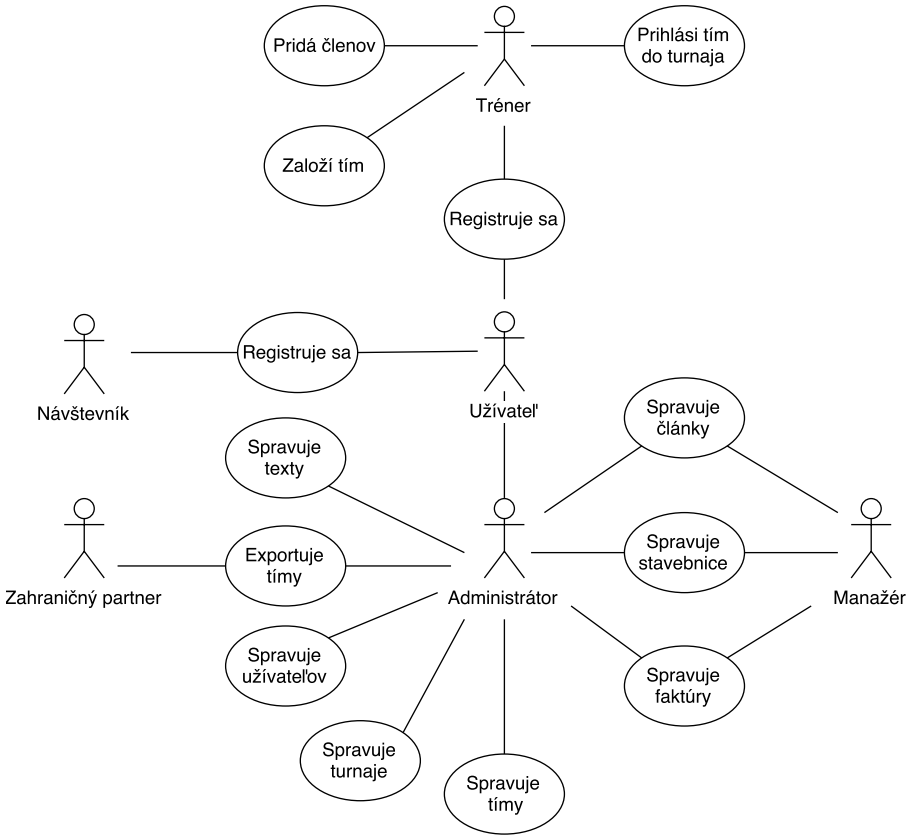
4. NÁVRH

Kapitola sa skladá z troch častí, používateľské role, návrh používateľského rozhrania a návrh štruktúry databázy. V prvej časti sa budem venovať jednotlivým používateľským roliam a ich činnostiam. Nasledujúca časť bude o navrhnutom používateľskom rozhraní oboch stránok, prezenčnej a administrátorskej. Posledná časť je určená databáze, jednotlivým tabuľkám a opisu ich funkcie pri tvorbe dynamického obsahu na prezenčnej stránke.

4.1 Používatelia

4.1.1 Používateľské role

V aplikácií sa nachádza šesť preddefinovaných používateľských rolí s rôznymi úrovňami autorizácie. Týmito úrovňami sme oddelili užívateľov do dvoch hlavných skupín, tí čo majú prístup len do prezenčnej stránky a tí čo majú prístup aj do administrátorskej časti aplikácie. Na obrázku nižšie môžeme vidieť jednotlivé role zobrazené za pomoci *use case* diagramu.



Obr. 4.1: Use case diagram používateľov

Návštevník - ne-registrovaný užívateľ

Užívateľ - registrovaný užívateľ stránky, ktorý sa môže stať trénerom vyplnením dodatočných informácií.

Tréner - registrovaný užívateľ, ktorý vyplnil potrebné dodatočné údaje na tom aby mohol byť trénerom. Môže založiť svoj vlastný tím, pridať členov tímu a prihlásiť sa s ním na momentálny ročník.

Manažér - môže pridávať novinky, upravovať informácie o zaslaných stavebniciach a faktúrach.

Administrátor - má najvyššie právomoci na stránke, spravuje jej obsah cez administrátorské rozhranie. Môže otvoriť nový a ukončiť starý ročník, upravovať tímové informácie a texty jednotlivých turnajov. Ako jediný môže pridať administračných užívateľov a priradiť im používateľské role. Navyše má všetky právomoci role *Manažér*.

Zahraničný partner - užívateľ, ktorý má právo jedine na export databázy systému do formátu .csv.

4.2 Používateľské rozhrania

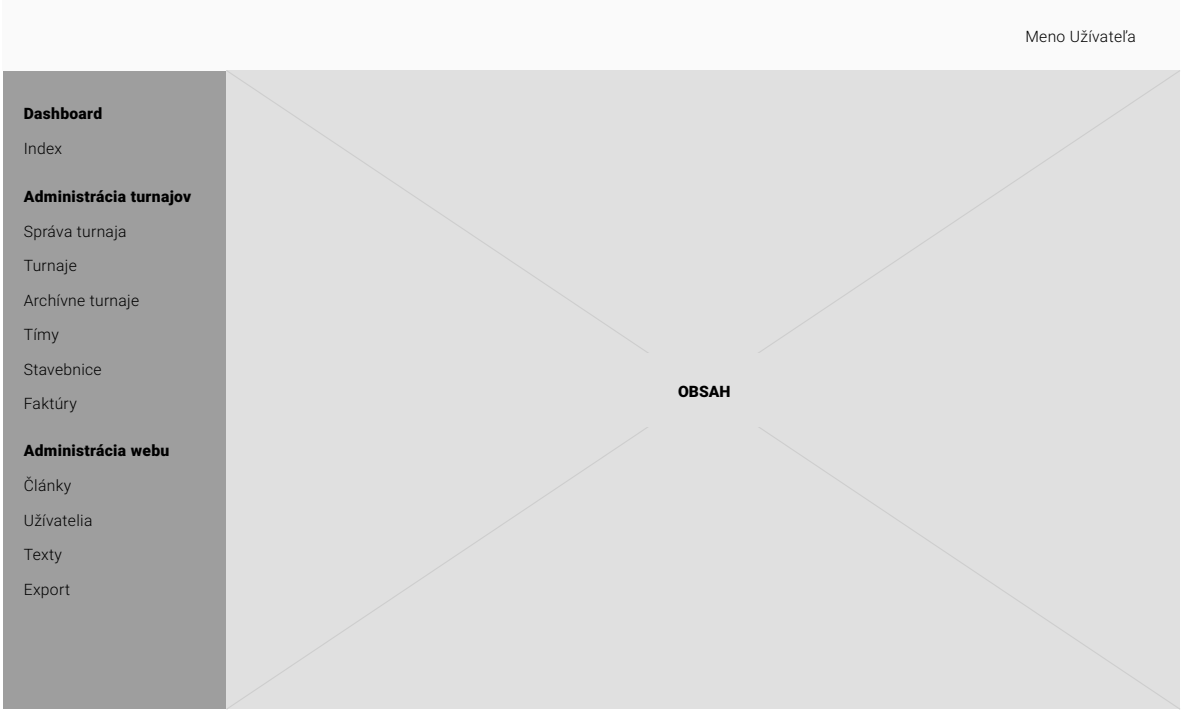
4.2.1 Rozhranie administrátorského prostredia

Toto rozhranie je navrhnuté pre jednoduchosť používania ako klasické administrátorské rozhranie s hlavným menu vľavo, sekundárnym menu navrchu a dynamicky generovaným obsahom v strede.

Hlavné menu - obsahuje všetky potrebné odkazy na prácu so stránkou. Je plne responzivné a v prípade mobilného rozlíšenia sa minimalizuje a je dostupné za pomoci tzv. *hamburger icon* v sekundárnom menu.

Sekundárne menu - slúži ako informačný panel. Nachádza sa v ňom stav prihláseného používateľa a možnosť jeho odhlásenia.

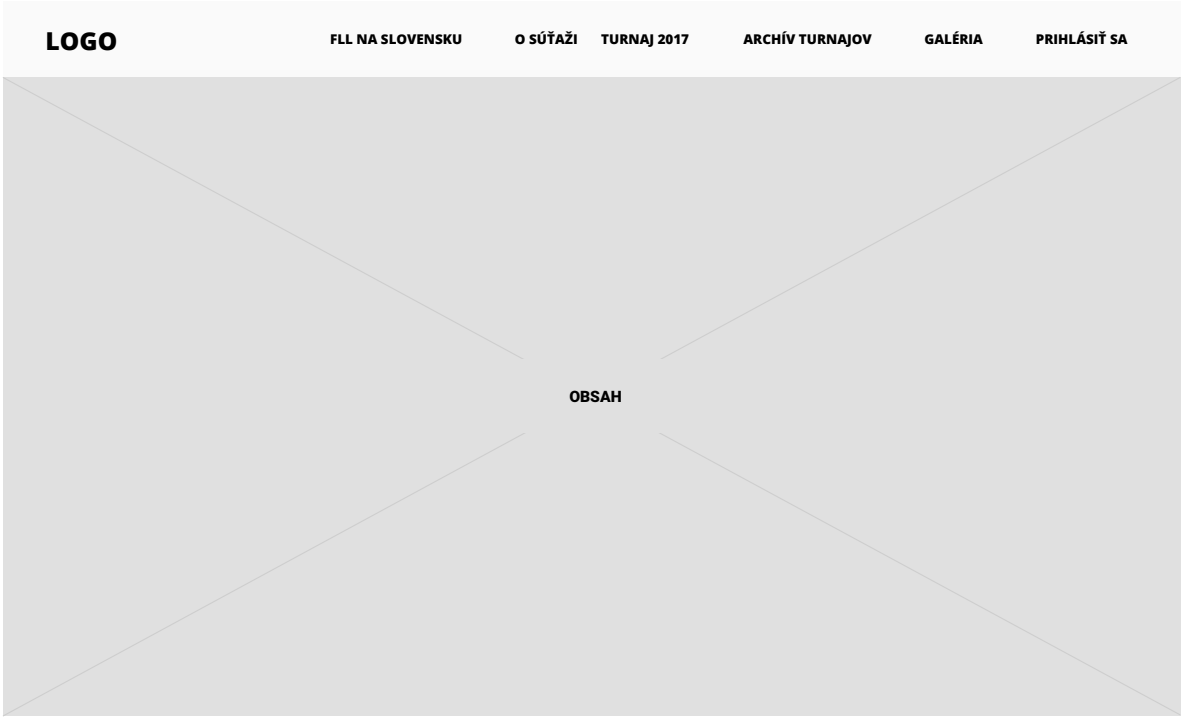
Obsah - nachádzajúci sa v strede stránky je generovaný na základe voľby používateľa z hlavného ľavého menu.



Obr. 4.2: Návrh administrátorského prostredia

4.2.2 Rozhranie prezenčného prostredia

Jedným z hlavných vylepšení stránky oproti predchádzajúcej verzii je plná responzivita na každom zariadení. Tréneri tak nemajú problém spravovať svoj tím alebo prihlásiť sa na turnaj z mobilného zariadenia. Menu bolo zjednodušené do najdôležitejších položiek a malo by tak uľahčiť hľadanie potrebných údajov pre turnaj.



Obr. 4.3: Návrh administrátorského prostredia

4.3 Databáza

Databáza slúži aplikácii nie len na ukladania dát, no takisto sa za jej pomoci vypisujeme dynamické menu na prezenčnej stránke.

4.3.1 Návrh databázy

Kits - slúži na evidenciu zaslaných stavebníc spolu s evidenčným číslom z pošty.
Pri registrácií do turnaja sa automaticky vytvorí pred pripravený záznam s daným tímom v tejto tabuľke.

Members - členovia jednotlivých tímov. Obsahuje názov člena, dátum narodenia a *id*

príslušného tímu.

Tournaments, Tournament Archives - tieto dve tabuľky slúžia na momentálne prebiehajúci turnaj a všetky doposiaľ archivované turnaje.

Invoices - slúži na evidenciu dátumu registrácie tímu do turnaja spolu s dátumami zaslania a zaplatenia faktúry. Pri registrácií do turnaja sa automaticky vytvorí pred pripravený záznam s daným tímom v tejto tabuľke.

Teams - tabuľka obsahuje potrebné údaje o tíme, názov tímu, názov organizácie, mesto z ktorého tím pochádza a *id* odkazujúce na trénera tímu.

Coaches - dodatočné údaje adresa, telefónne číslo o užívateľovi v prípade, že je trénerom.

Users - tabuľka všetkých užívateľov registrovaných na stránke. Obsahuje unikátny mail, ktorý slúži na prihlásenie, meno a priezvisko a zašifrované heslo.

Role User, Roles - v tabuľke *roles* sa nachádzajú jednotlivé užívateľské role s rôznymi úrovňami autorizácie. Obsahuje skrátený názov role, dlhší názov slúžiaci na zobrazenie napríklad v registračnom formulári a krátky popis role. Tabuľka *Role User* je väzobná tabuľka medzi touto a *user* tabuľkou.

Permission Role, Permissions - dodatočná tabuľka k tabuľkám *Role* a *Role User*. Slúži na evidenciu povolení pre jednotlivé role a ich napojenie na tieto role.

News - uchováva novinky, ktoré sa zobrazujú na hlavnej stránke.

Databáza obsahuje šesť pomocných tabuliek, ktoré nie sú priamo prepojené so žiadnou inou v našej databáze, no nie sú o nič menej podstatné.

Additionals - multi-funkčná tabuľka, slúžiaca na uchovanie roku momentálneho ročníka a počet ostávajúcich stavebníc.

Dates - nachádzajú sa v nej všetky dôležité dátumy týkajúce sa turnaja.

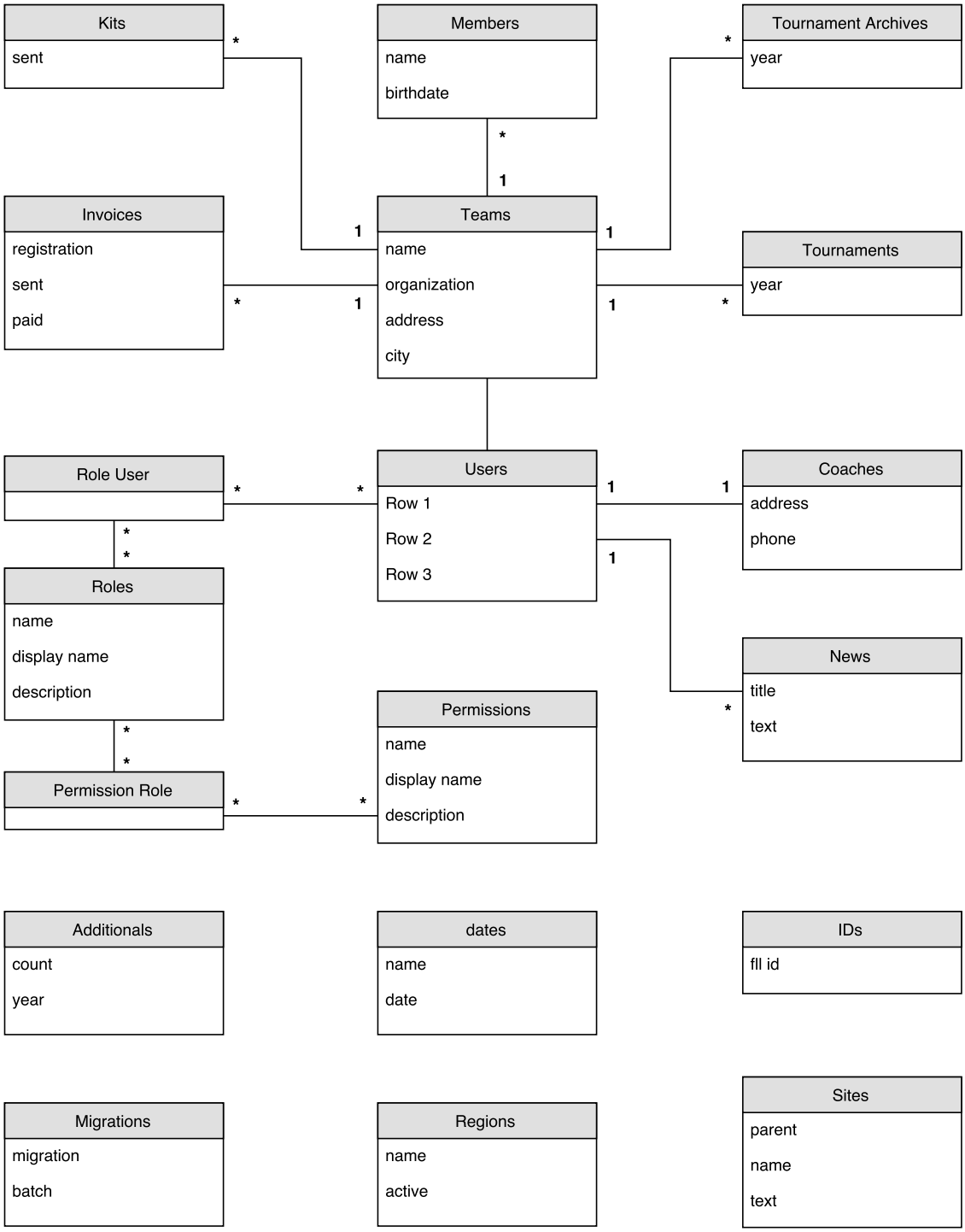
IDs - tabuľka obsahuje riadky *ID* dodávaných od zahraničných partnerov, pod ktorými budú naše tímy evidované v ich databáze.

Migrations - pomocná tabuľka slúžiaca na *migrácie* a *rollback-y* databázy.

Regions - zahŕňa regióny, do ktorých sa tímy môžu prihlásiť. Stĺpec *is_active* uvádza stav daného regiónu, jeho aktívnosť v danom ročníku v prípade, že sa región v ročníku nemusí

nachádzať.

Sites - tabuľka slúži na automatické generovanie časti menu prezenčnej stránky. Uchováva html texty používané pre zadania úloh, informácie k región v ročníku a pod.



Obr. 4.4: Entitno relačný model databázy

4.3.2 Archivovanie

Pre zjednodušenie mnohých vecí bol celý proces archivácie momentálneho ročníka zautomatizovaný. Všetky súťažné tímy sa z aktuálneho turnaja presunú do tabuľky archivovaných tímov a hneď na to sa nám otvorí nový ročník s požadovaným rokom ako vstup od užívateľa. Súťažné regióny z minulého roka a ich stav (aktívny/neaktívny) sa nám ponechajú spolu s dátumami.

4.4 Štruktúra aplikácie

Jednotlivé prvky a komponenty sú rozdelené do príslušných adresárov, zmena tejto štruktúry nepredstavuje pre Laravel problém. Štruktúru teda môžeme ľubovoľne meniť pokiaľ je Composer schopný nájsť naše triedy.

app - najdôležitejší adresár vzhľadom na biznis logiku aplikácie. Priamo v ňom sa nachádzajú naše Modely, v zložke *Controllers* nájdeme všetky nami zadefinované Controller-y.

bootstrap - obsahuje súbory štartujúce framework a nastavenia pre tento štart.

config - všetky hlavné nastavenia aplikácie. Nastavenia názvu, url adresy našej aplikácie, jej časovú zónu a potrebné triedy.

database - ako naznačuje názov, adresár slúži na prácu s databázou. Obsahuje pod-adresár migrations, v ktorom máme create a drop skripty pre naše tabuľky a v prípade podreby adresár seeds, slúžiaci na naplnenie databázy.

public - zahŕňa potrebné súbory pre front-end aplikácie.

resources - v tomto adresáry sa nachádza podstatná časť MVC a to View komponent. Obsahuje všetky *blade* súbory, ktoré aplikácia využíva. Zložka *lang* slúži na preloženie textov, v našom prípade do slovenčiny. V *assets* nájdeme neskompilované *sass* a *less* súbory.

routes - smerovacia logika aplikácie. Súbor *web.php* obsahuje všetky *GET, POST, PATCH* a *DELETE* metódy.

storage - obsahuje skompilované *blade* súbory aplikácie, sessions a cache súbory. Tento adresát môže byť využitý aj ako úložisko súborov od našich užívateľov.

test - v prípade potreby nám slúži na uloženie *PHPUnit* testov.

vendor - v zložke sa nachádzajú závislosti pre Composer, takisto aj všetky dodatočne stiahnuté rozšírenia vytvorené komunitou[1].

Obr. 4.5: Stromová štruktúra adresárov aplikácie

[1]LARAVEL. 2017. *Directory Structure*. [online]. : 2017. [cit. 8.4.2013]. Dostupné na internete: `https://laravel.com/docs/5.4/structure` (Dátum prístupu 07/03/2017)

5. ZÁVER

Cieľom bakalárskej práce bolo navrhnúť a zostrojiť webovú aplikáciu, či už za použitia existujúceho CMS alebo vytvoriť vlastné. V počiatočnom štádiu prebehla analýza dostupných riešení, po ktorej som sa rozhodol systém vyvinúť vo vlastnej réžií. Nakoľko som však pred začatím tejto práce framework Laravel nepoznal a predstava potrebných veci bola zúžená, znamenalo to pre mňa značnú výzvu. V konečnom dôsledku si ale myslím, že vytvoriť vlastný CMS bola náročná cesta no napriek tomu správna voľba. Aplikácia v mnohých veciach zjednodušila prácu a zamerala sa na správu tímov a turnajov a vynechala tak množstvo zložitých prvkov nachádzajúcich sa vo väčšine dostupných redakčných systémoch.

V prípade potreby a plnohodnotného uspokojenia zadávateľa je ďalší rozvoj mojej aplikácie možný. Rozšírenia o nové prvky by nemali narušiť jej beh a nemali by teda predstavovať problém pre webmasterov a užívateľov ani pri nasadení do zabehnutej prevádzky. Možným rozšírením aplikácie by bola evidencia sponzorských príspevkov a požiadaviek na sponzorov na základe chýbajúcich položiek v rozpočte. Sponzori by teda videli, na ktoré časti súťaže chýbajú peniaze a mohli by tak podporiť napríklad túto konkrétnu činnosť. Umožnené by mohli byť takisto finančné príspevky od individuálnych ľudí. Následne by bolo viditeľne zobrazené kto, kam a akou čiastkou prispel.

Aplikácia bola nasadená na testovací server, kde sme mohli vidieť a testovať jej správanie v reálnom použití. Tento proces nám odhalil chyby, ktoré bolo potrebné doladiť, no pripravil nás na možné presunutie na iný server. V celku sa aplikácia správala stabilne a myslím, že by obstála aj pod prípadným väčším nátlakom zo strany užívateľov.

Zdroje

[1] SAWYER, Maggie. 2016. *WordPress Users Beware: 19 Disadvantages of Using WordPress*. [online]. : 2016. [cit. 8.4.2013]. Dostupné na internete: https://www.adngin.com/blog/blogging-best-practices/wordpress-users-beware-19-disadvantages-using-wordpress/ (Dátum prístupu 01/06/2017).

[2] ALBERT, J. 2015. *Drupal Pros and Cons: An In Depth Look*. [online]. : 2015. [cit. 8.4.2013]. Dostupné na internete: https://www.mcdpartners.com/news/drupal-pros-and-cons/ (Dátum prístupu 01/06/2017).

[3] WILDIN, Robin. 2011. *Joomla: Advantages and Disadvantages of Choosing Joomla as Your CMS Solution*. [online]. : 2011. [cit. 8.4.2013]. Dostupné na internete: http://www.socialtechnologyreview.com/articles/joomla-advantages-and-disadvantages-choosing-joomla-your-cms-solution (Dátum prístupu 01/06/2017).

[4] O'BRIEN, Jesse. 2016. *A Brief History of Laravel*. [online]. : medium.com, 2016. [cit. 8.4.2013]. Dostupné na internete: https://medium.com/vehikl-news/a-brief-history-of-laravel-5d55970885bc (Dátum prístupu 05/11/2017).

[5] FRAMEWORKS, Hot. 2017. *Hodnotenie Framework-ov*. [online]. : 2017. [cit. 8.4.2013]. Dostupné na internete: http://hotframeworks.com/ (Dátum prístupu 03/03/2017).

[6] TUTORIALSPOINT. 2014. *MVC Framework Introduction*. [online]. : 2014. [cit. 8.4.2013]. Dostupné na internete: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm (Dátum prístupu 01/19/2017).

[7] GILMORE, W. Jason. 2015. *Laravel 5's Key Security Features*. [online]. : easylaravelbook.com, 2015. [cit. 8.4.2013]. Dostupné na internete: http://

www.easylaravelbook.com/blog/2015/07/22/laravel-key-security-features/ (Dátum prístupu 02/02/2017).

[8] LAVARYAN, Reza. 2015. *How Laravel Facades Work and How to Use Them Elsewhere*. [online]. : 2015. [cit. 8.4.2013]. Dostupné na internete: https://www.sitepoint.com/how-laravel-facades-work-and-how-to-use-them-elsewhere (Dátum prístupu 06/02/2017).

[9] TEAM, Bootstrap Core. 2011. *Bootstrap, Oficiálna stránka*. [online]. : 2011. [cit. 8.4.2013]. Dostupné na internete: http://getbootstrap.com/ (Dátum prístupu 10/09/2017).

[10] LARAVEL. 2017. *Directory Structure*. [online]. : 2017. [cit. 8.4.2013]. Dostupné na internete: https://laravel.com/docs/5.4/structure (Dátum prístupu 07/03/2017).

[11] —, 2017. *Laravel oficiálna dokumentácia*. [online]. : laravel.com, 2017. [cit. 8.4.2013]. Dostupné na internete: https://laravel.com/docs/5.4 (Dátum prístupu 01/01/2017).

[12] THE PHP DEVELOPMENT TEAM, Zend Technologies. 1995. *PHP*. [online]. : 1995. [cit. 8.4.2013]. Dostupné na internete: http://php.net/ (Dátum prístupu 09/12/2017).

PRÍLOHY

CD obsahujúce:

- Elektronickú verziu bakalárskej práce
- Zdrojové súbory aplikácie