UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Intranetová aplikácia v oblasti výživového mentoringu

Bakalárska práca

2022 Viktor Freyer

UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Intranetová aplikácia v oblasti výživového mentoringu

Bakalárska práca

Študijný program: Aplikovaná informatika

Študijný odbor: 2511 Aplikovaná informatika

Školiace pracovisko: Katedra didaktiky matematiky, fyziky a informatiky

Školiteľ: PaedDr.RomanHrušecký, PhD.

2022 Viktor Freyer

**Čestné prehlásenie**

Čestne prehlasujem, že som bakalársku prácu vypracoval samostatne pod vedením školiteľa a s použitím uvedenej literatúry.

V Bratislave, 25.5. 2022 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Meno autora

**Poďakovanie**

Touto cestou by som sa chcel poďakovať svojmu školiteľovi PaedDr. Roman Hrušecký, PhD. za cenné rady a usmernenia, konzultácie a čas, ktoré mi venoval počas tvorby a písania bakalárskej práce.

Abstrakt

Cieľom bakalárskej práce bolo vytvorenie intranetovej aplikácie v oblasti výživového mentoringu. Aplikácia rozdeľuje 2 typy užívateľov a to admin a rádový zamestnanec. Rádový zamestnanci môžu mať rozličné práva. Aplikácia umožňuje rádovým zamestnancom vytvoriť klienta, priradiť mu jeho kontakty, základné informácie, rôzne odznaky, ktoré slúžia ako kľúčové informácie(napríklad diagnóza, šport a podobne) a samozrejme zaznamenávať rôzne typy meraní klienta. Admin má funkcionalitu rádového zamestnanca a navyše vie vytvárať užívateľov systému, distribuovať im práva, konfigurovať profily, vytvárať nové produkty a priraďovať užívateľov klientom. Aplikácia taktiež umožňuje používateľom tvoriť rôzne udalosti, či už firemné alebo s klientmi. Aplikáciu sme vyvíjali použitím open source PHP frameworku Laravel.

Kľúčové slová: Laravel, Mentoring, Intranet

Abstract

TODO abstract

Keywords:

Obsah

[Úvod 9](#_Toc104230954)

[1 Východiská 11](#_Toc104230955)

[1.1 Prehľad teórie 11](#_Toc104230956)

[1.1.1 Intranet 11](#_Toc104230957)

[1.1.2 MVC Architektúra 12](#_Toc104230958)

[1.2 Existujúce softvérové riešenia 13](#_Toc104230959)

[1.2.1 Intranet s webovým rozhraním pre sektor softwarového testovania 14](#_Toc104230960)

[1.2.2 Informačný systém pre materské školy 14](#_Toc104230961)

[1.2.3 Nutriadapt 15](#_Toc104230962)

[1.2.4 Metabolic Balance 16](#_Toc104230963)

[1.3 Technológie 17](#_Toc104230964)

[1.3.1 Laravel 17](#_Toc104230965)

[1.4 Testovanie použiteľnosti 18](#_Toc104230966)

[1.4.1 Cieľ 19](#_Toc104230967)

[1.4.2 Použiteľnosť 20](#_Toc104230968)

[1.4.3 Testovací scenár 21](#_Toc104230969)

[2 Návrh 22](#_Toc104230970)

[2.1 Use cases 22](#_Toc104230971)

[2.2 Dátový model 23](#_Toc104230972)

[2.3 Stavový diagram pre produkt 27](#_Toc104230973)

[3 Implementácia 28](#_Toc104230974)

[3.1 Implementačné jazyky a knižnice 28](#_Toc104230975)

[3.2 Organizácia kódu 28](#_Toc104230976)

[3.3 Práva 29](#_Toc104230977)

[3.3.1 Práva na klienta 30](#_Toc104230978)

[3.4 Vytváranie udalostí 30](#_Toc104230979)

[3.5 Stavový automat pre produkty 30](#_Toc104230980)

[4 Testovanie 32](#_Toc104230981)

[4.1 Cieľ 32](#_Toc104230982)

[4.2 Priebeh 32](#_Toc104230983)

[4.3 Testovacie scenáre 32](#_Toc104230984)

[4.3.1 Scenáre pre admina 33](#_Toc104230985)

[4.3.2 Scenáre pre používateľa 34](#_Toc104230986)

[4.4 Vyhodnotenie 36](#_Toc104230987)

[4.4.1 Zistené chyby pri testovaní 37](#_Toc104230988)

[4.4.2 Úpravy aplikácie 38](#_Toc104230989)

[Záver 40](#_Toc104230990)

[Použitá literatúra 41](#_Toc104230991)

[Zoznam obrázkov 42](#_Toc104230992)

# Úvod

V dnešnej dobe je veľký dopyt po službách zaoberajúcich sa zdravým a chudnutím. Konkurencia je veľmi silná, existuje mnoho firiem zaoberajúcich sa presne touto tematikou. Z tohto dôvodu je veľmi dôležité keď chceme v tejto sfére uspieť, aby bolo všetko správne a sto percentné. Preto sme sa rozhodli, navrhnúť aplikáciu pre jednu z firiem, ktorá potrebuje zjednodušiť ich pracovný proces. Firma sa rozrastá značne rýchlo a nápor klientov pre ich navrhnutý proces začína byť nezvládnuteľný. Firma pracuje na rôznych platformách a s rôznymi aplikáciami. Na komunikáciu s klientmi používa Facebook, na udalosti používajú Google Kalendár a zložky klientov skladujú v papierových formách a na diskovom poli v rôznych Excelovských tabuľkách, čo je už v tejto dobe zastaralé.

Cieľom tejto práce je teda navrhnúť, implementovať a otestovať intranetovú aplikáciu pre danú firmu a uľahčiť im ich pracovný proces. Aplikácia bude rozdeľovať používateľov na admina a rádových zamestnancov. Rádovému zamestnancovi bude umožnené vytvárať klientov, ktorým vie priradiť základné informácie, jeho kontakty a odznaky, ktoré priraďujú klientovi nejakú dôležitú informáciu(napríklad diagnóza, koľko schudol, či športuje). Zamestnanci taktiež budú schopní ponúkať produkty a zaznamenávať jednotlivé meranie klientov. Admin bude mať funkcionalitu rádového zamestnanca, ale navyše mu aplikácia umožní vytvárať užívateľov, distribuovať im práva a konfigurovať profily, vytvárať nové produkty, odznaky a zasadačky a bude môcť spájať klientov a užívateľov. Užívateľom bude umožnené tiež vytvárať firemné a klientske udalosti.

Danou aplikáciou zaistíme, aby firma nemusela používať Google Kalendár, nakoľko udalosti si bude môcť tvoriť v rámci aplikácie. Taktiež firme návrhom zaistíme, aby nemusela používať excel a papierové zložky na skladovanie dát.

Práca je rozdelená na 4 kapitoly. V prvej kapitole sa zaoberáme technológiami využitými v aplikácii a analýzou podobných bakalárskych prác a existujúcich riešení konkurencie. V druhej kapitole popisujeme návrh celej aplikácie, štruktúru databázy a funkcionalitu aplikácie. Tretia kapitola uvádza implementáciu, kde popisujeme niektoré časti kódu, ich funkcionalitu a celkovú organizáciu kódu. V štvrtej kapitole je opísané testovanie aplikácie, jeho štruktúra, priebeh a vyhodnotenie. Práca taktiež obsahuje záver, kde sme zhodnotili čo všetko sa podarilo vytvoriť a aké sú možné plány do budúcna s touto aplikáciou.

# Východiská

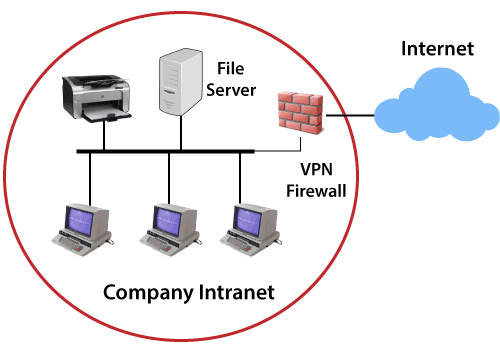
V tejto kapitole uvedieme základne teoretické pojmy týkajúce sa našej práce, ukážeme si existujúce riešenia, popíšeme si ich funkcionalitu a popíšeme technológie využité pri tvorbe a vývoji našej aplikácie.

## Prehľad teórie

V tejto kapitole si prejdeme základné teoretické pojmy použité v našej práci. Popíšeme si základné nástroje, s ktorými budeme pracovať a nad niektorými týmito nástrojmi si ukážeme príklady použitia.

### Intranet

Intranet je súkromná sieť zavedená vo firme, ktorá sa používa na odosielanie a zdieľanie informácii a zdrojov medzi zamestnancami bezpečne a mimo dosah niekoho mimo firmu. Intranet podporuje komunikáciu v rámci danej organizácie a umožňuje zamestnancom prístup napríklad k informáciám, linkom, databáze firemných registrov, formulárom a podobne. Používa zvyčajne používa databázu, ktorá v sebe uchováva používateľské mená zamestnancov, ktorí majú rôzne úrovne prístupových práv do siete a k dátam poskytovaných intranetom, aby sa zachovalo súkromie a bezpečnosť. (1)



Obrázok 1 - Základná schéma intranetu

Intranet sa dá použiť rôznymi spôsobmi, záleží od potrieb danej organizácie. Poznáme napríklad centrálny repozitár, ktorý v sebe uchováva všetky dôležité informácie, verzie, zmeny a dáta. Ako ďalší typ použitia máme napríklad kolaboráciu, ktoré ponúka zdieľanie a správu informácií, ktorá je pre zamestnancov intuitívna. Taktiež môže byť použitý ako komunikačný prostriedok, ktorý zjednocuje firemnú komunikáciu. V neposlednom rade môže poskytovať automatizáciu firemných procesov, ktoré je potrebné vykonávať na dennej báze. (1)

Kvalitný, spoľahlivý a bezpečný intranet vyžaduje web server, ktorý je schopný riadiť požiadavky umiestnené na serveri. Tento webový server nájde a pošle vyžiadané dáta používateľovi, ktorý si ich vyžiada. Intranetovský webový server používa klasické protokoly ako http, tcp/ip a mnohé iné. Zvyčajne sa všetky spojenia posielajú cez jednu, alebo aj niekoľko gateway počítačov do internetu a iných vonkajších zdrojov. (1)

Na pripojenie do intranetu musí mať užívateľ prístupové údaje a byť pripojený priamo na lokálnu sieť(ďalej len LAN) danej firmy. Užívatelia, ktorí sa potrebujú pripojiť, ale nie sú priamo pripojený na LAN, môžu využiť vzdialené pripojenie, ktorým získajú prístup k intranetu. (1)

Najvýznamnejšie výhody intranetu sú zlepšená komunikácia, jednoduchá správa záznamov, jednoduché sledovanie užívateľských požiadaviek a transparentnosť. (1)

### MVC Architektúra

MVC je architektonický vzor, ktorý sa delí na 3 hlavné logické celky. Model, ktorý reprezentuje objekty v systéme a poskytuje operácie súvisiace s doménou aplikácie. View poskytuje užívateľské rozhranie, ktoré slúži na zobrazovanie jednotlivých modelov aplikácie. Posledným logickým celkom je Controller, ktorý spája interakciu užívateľa s modelom a viewom. Jedná sa o jednu z najpoužívanejších architektúr na tvorbu webových aplikácií. (2)

#### Model

Je na najnižšom leveli, definuje štruktúru dát v aplikácii, zodpovedá za ich konzistentnosť a je mapovaním databázových objektov. Poskytuje rôzne nástroje, na základe ktorých je možné filtrovať dáta, ale neobsahuje žiadnu biznisovú logiku nad dátami. Môže ale obsahovať základné validačné kontroly špecifické pre model. (3)

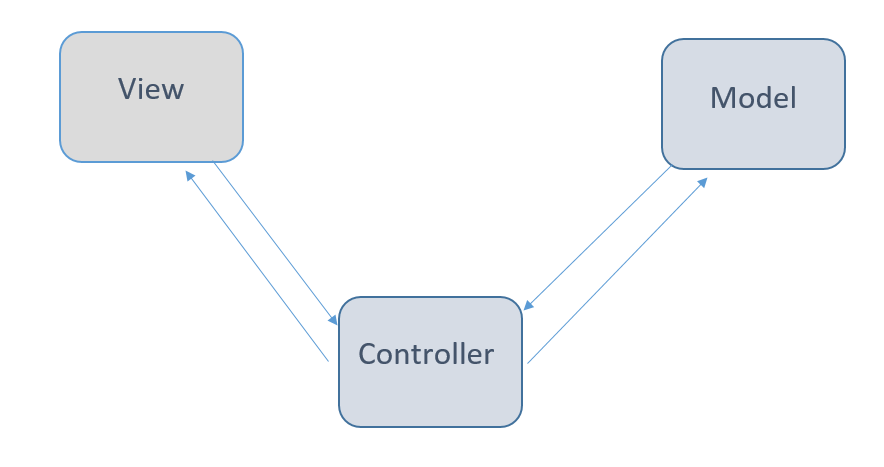
#### View

View zodpovedá za používateľské rozhranie. Každý view je definovaný ako šablóna, kde štandardne každá šablóna je definovaná pre konkrétny model. Môže poskytovať rozhranie na zobrazenie všetkých modelov, konkrétneho modelu a formulára na vytvorenie alebo editáciu modelu. (3)

#### Controller

Controller spája model a view. Klient zadá špecifickú požiadavku na server, ktorý sa rozhodne, na ktorý controller bude delegovať danú požiadavku. Controller na základe požiadavky získa všetky potrebné dáta z modelu. Tieto dáta môže podľa typu požiadavky upraviť, vytvoriť, vymazať alebo vrátiť klientovi pomocou view. View na základe vstupu z controllera vygeneruje čitateľný výstup pre klienta. (3)

MVC architektúra uľahčuje vývoj aplikácie, nakoľko rozdeľuje logiku do troch celkov. Každý celok je možné vyvíjať paralelne aj s väčšou skupinou programátorov.



Obrázok 2 - Základná schéma mvc architektúry

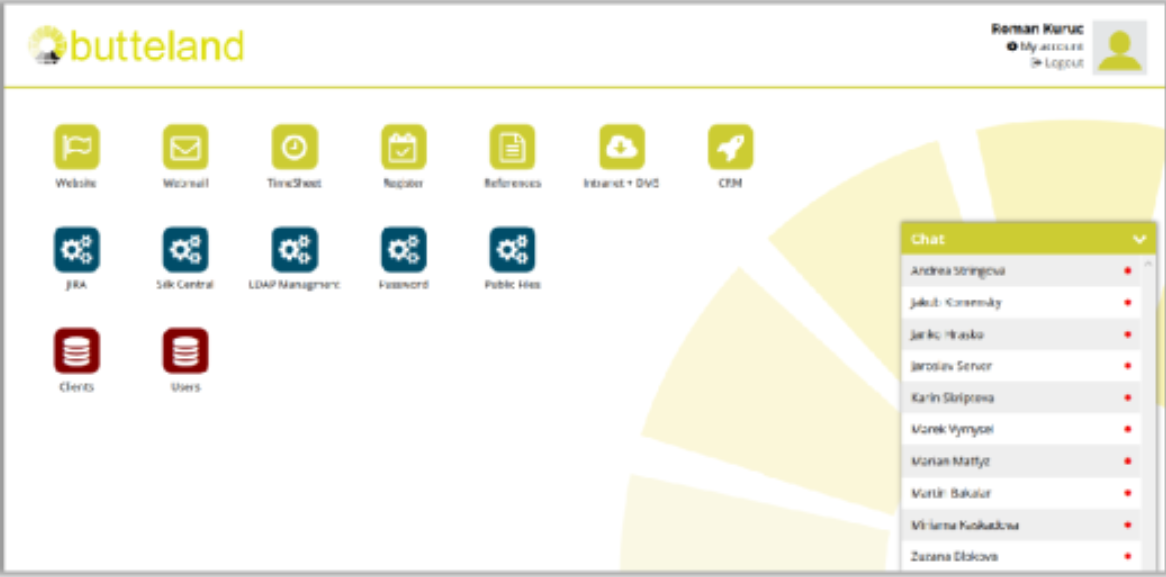
## Existujúce softvérové riešenia

V tejto časti si ukážeme práce ktoré tvorili intranetové systémy a existujúce riešenia v našej doméne výživového mentoringu.

### Intranet s webovým rozhraním pre sektor softwarového testovania

S týmto intranetom prišiel Roman Kuruc v roku 2018. Aplikácia slúži na softvérové testovanie pre firmu. Poskytuje aj správu firemných záznamov, automatizuje firemné procesy, eviduje odpracovaný čas zamestnancov, organizuje rozdelenie zamestnancov na projekty a zaznamenáva činnosť zamestnancov. (4)

Aplikácia bola robená na špecifickú firmu, čiže v sebe obsahuje časti, ktoré nesúvisia s výživovým mentoringom. Poskytuje ale evidenciu používateľov a automatizáciu firemných procesov. Neposkytuje ale organizáciu udalostí, odmeňovanie klientov, alebo komunikáciu s klientom. (4)



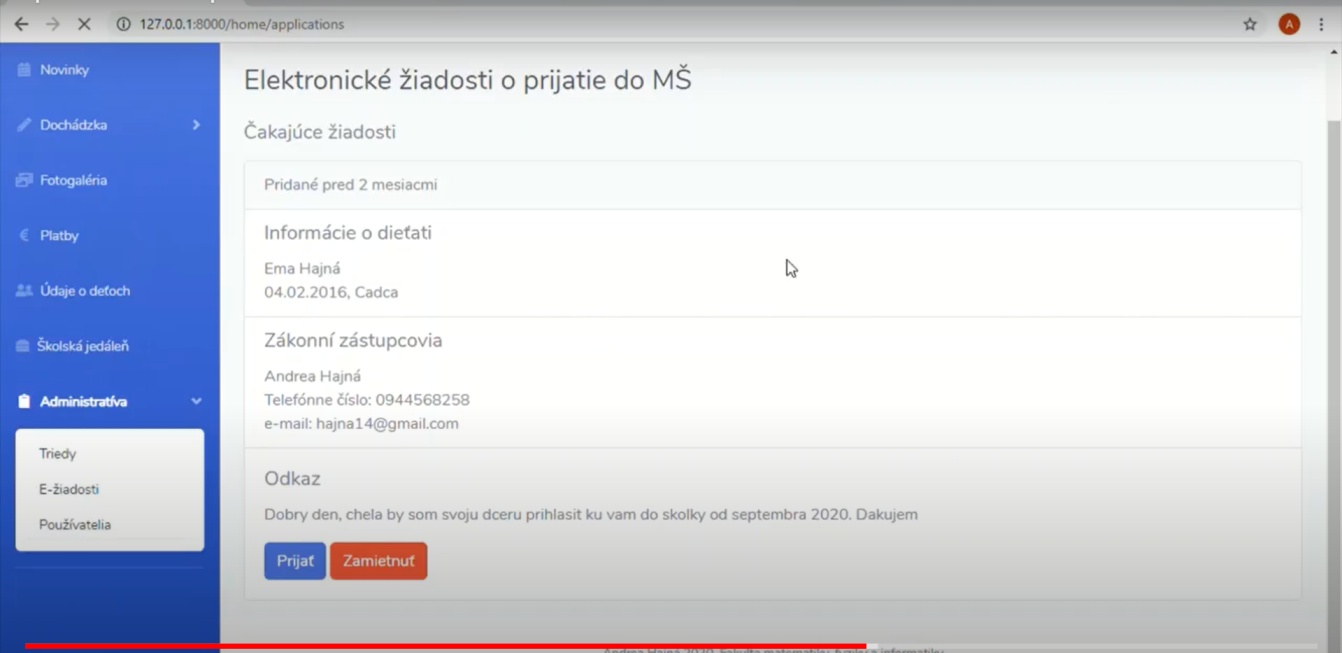
Obrázok 3 - Grafické rozhranie intranetu Romana Kuruca

### Informačný systém pre materské školy

Cieľom systému je vytvorenie informačného systému pre materské školy, ktorý zabezpečuje evidovanie dochádzky, rôzne typy platieb, vytváranie platobných predpisov, prihlášok a ich schvaľovanie, evidenciu detí, tried a učiteľov. Rodičia cez tento systém môžu nahlasovať neprítomnosť detí, dopĺňať a meniť osobné údaje o deťoch a prezerať si informácie a oznamy materskej školy. (5)

Aplikácia je robená pomocou open source frameworku Laravel. Na komunikáciu so serverom sa využíva asynchrónna komunikácia Ajax, ktorá slúži na získavanie dát z databázy. (5)

Aplikácia má v sebe 4 typy používateľov- učiteľ, rodič, riaditeľ, vedúci školskej jedálne. Každý typ používateľa má prístup ku konkrétnym funkcionalitám aplikácie. Systém si eviduje zoznam klientov, detí, ktoré spadajú pod konkrétneho rodiča a učiteľa. Každý učiteľ má svoju triedu, v ktorej vie zaznamenávať dochádzku, vytvárať predpisy platieb pre jeho triedu a evidovať k nim uskutočnené platby. Rodič má právo iba na prezeranie dochádzky svojho dieťaťa, platobné príkazy. Taktiež si vie pozrieť denné menu materskej školy. (5)

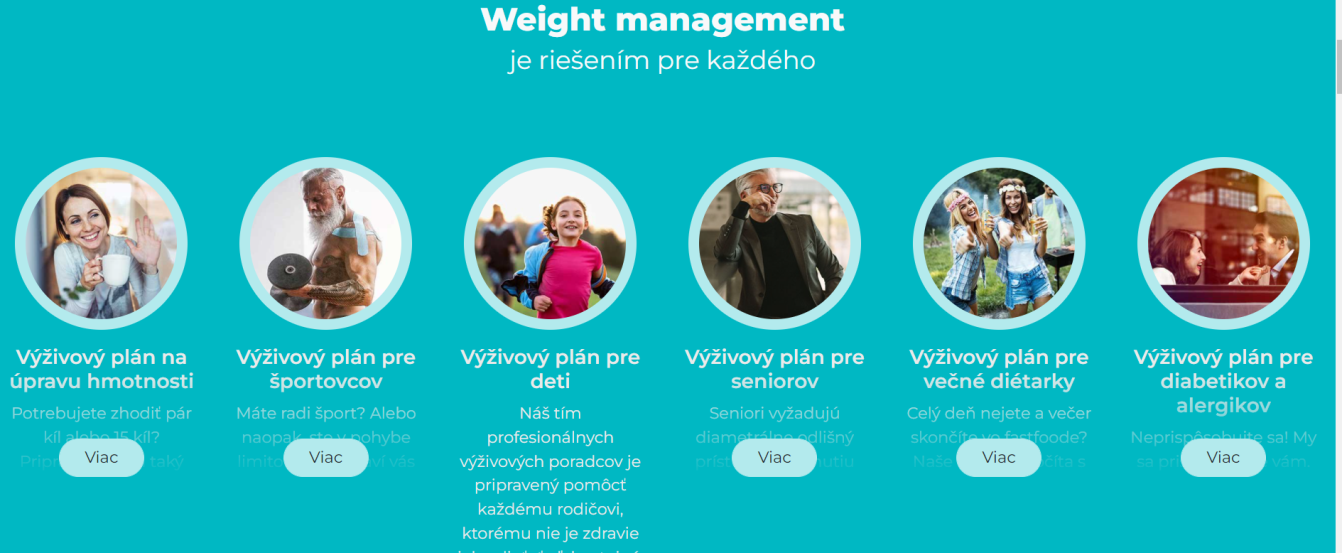


Obrázok 4 - Grafické rozhranie aplikácie Andrei Hajnéj

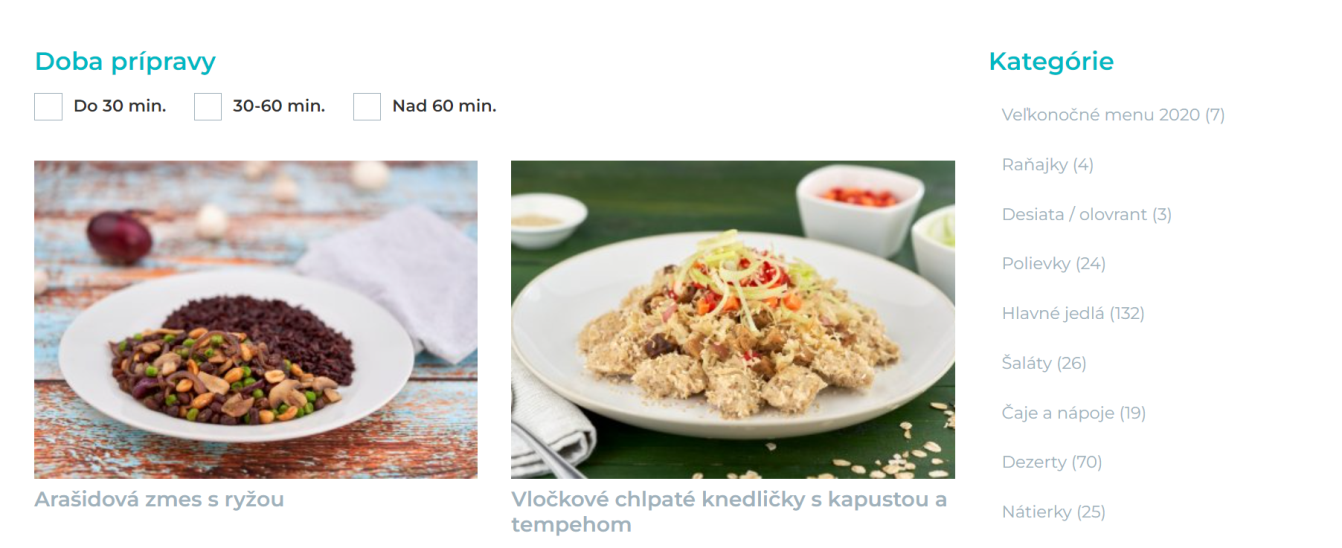
### Nutriadapt

Nutriadapt je firma, ktorá ponúka rozsiahlu škálu programov pre klientov, ktorí majú záujem sa zdravo stravovať, zredukovať hmotnosť, riešiť svoju diagnózu, stravu pre športovcov a podobne. Funguje zvyčajne na báze objednávania si konzultácie, ktorú klient absolvuje naživo v jednej z ich kliník. Stránka navyše ponúka takzvaný môj nutriadapt, kde klienti môžu sledovať svoje úspechy, viesť si rôzne štatistiky a môžu sa jednoducho spojiť so svojím osobným výživovým špecialistom. Stránka navyše ponúka receptár rozdelený do rôznych kategórií.

Ich pracovný proces sa ale výrazne líši s procesom firmy, pre ktorú aplikáciu vyvíjame. Preto ako inšpirácia slúžila minimálne. (6)



Obrázok 5 - webová stránka nutriadaptu 1

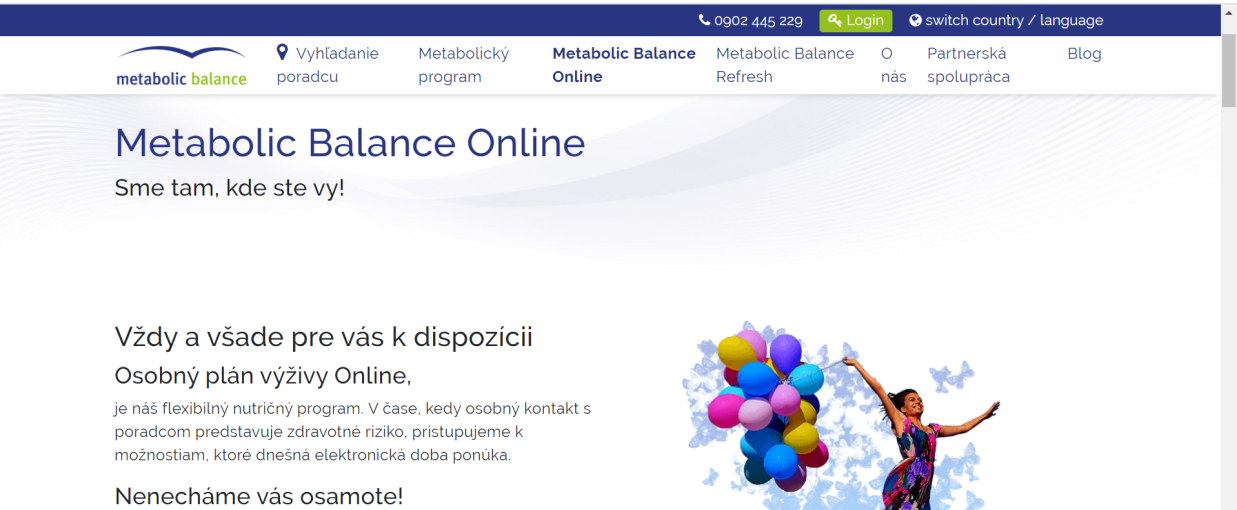


Obrázok 6 - webová stránka nutriadaptu2

### Metabolic Balance

Metabolic Balance je ďalšia firma, ktorá ponúka výživové poradenstvo. Fungujú na základe vyplnenie formulára, následne musíte absolvovať online konzultáciu s ich špecialistom a ako posledné Vám urobia odber z krvi, na základe ktorej Vám nastavia na mieru Vaše jedálne lístky. Stránka navyše ponúka rôzne programy.

Ich pracovný proces sa ale výrazne líši s procesom firmy, pre ktorú aplikáciu vyvíjame. Preto ako inšpirácia nám slúžila minimálne. (7)

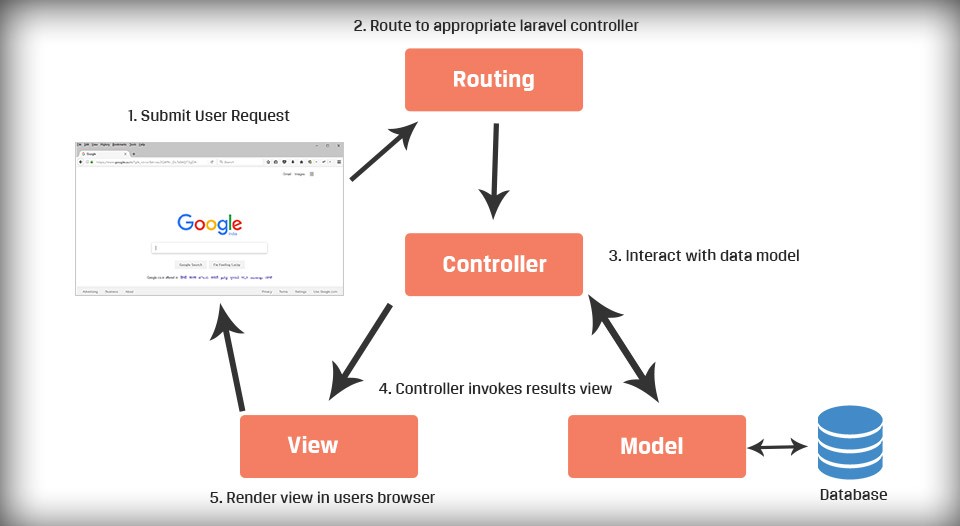


Obrázok 7 - webová stránka MetabolicBalance

## Technológie

### Laravel

Laravel je open source Php framework, ktorý sa používa na tvorbu webových aplikácií. Používa architektonický štýl nazývaný model-view-controller, v skratke MVC. Vznikol v roku 2009 a jeho zdrojový kód je dostupný na githube. Jeho obľúbenosť pramení v jeho pokročilých funkciách a vývojových nástrojoch. Laravel navyše obsahuje prehľadnú dokumentáciu, je intuitívny a flexibilný. (8)

x

Obrázok 8 - ako funguje laravel

#### Laravel Artisan

Artisan je rozhranie príkazového riadku, ktorý ponúka Laravel. Poskytuje mnoho užitočných príkazov, ktoré vieme získať použitím príkazu *php artisan list*. Každý príkaz zahrňuje help, ktorý dopodrobna opisuje daný príkaz. Napríklad príkaz *php artisan serve* spustí aplikáciu na lokálnom zariadení. Taktiež dokáže generovať model a controller pomocou príkazu *php artisan make:model NazovModelu -mcr.* (9)

#### Laravel Migration

Laravel Migration je základná funkcia v Laraveli, ktorá poskytuje možnosť vytvoriť tabuľku v databáze. Navyše umožňuje upravovať a zdieľať schému databázy. Pre vygenerovanie migrácie môžeme použiť príkaz *php artisan make:migration*. Pre zbehnutie všetkých migrácií môžeme použiť *php artisan migrate*. Každá migrácia obsahuje 2 metódy a to up a down. Up metóda slúži na vytváranie nových tabuliek, riadkov, alebo indexovanie. Down slúži na zvrátenie metód, ktoré robí up. (10)

#### Laravel Seeding

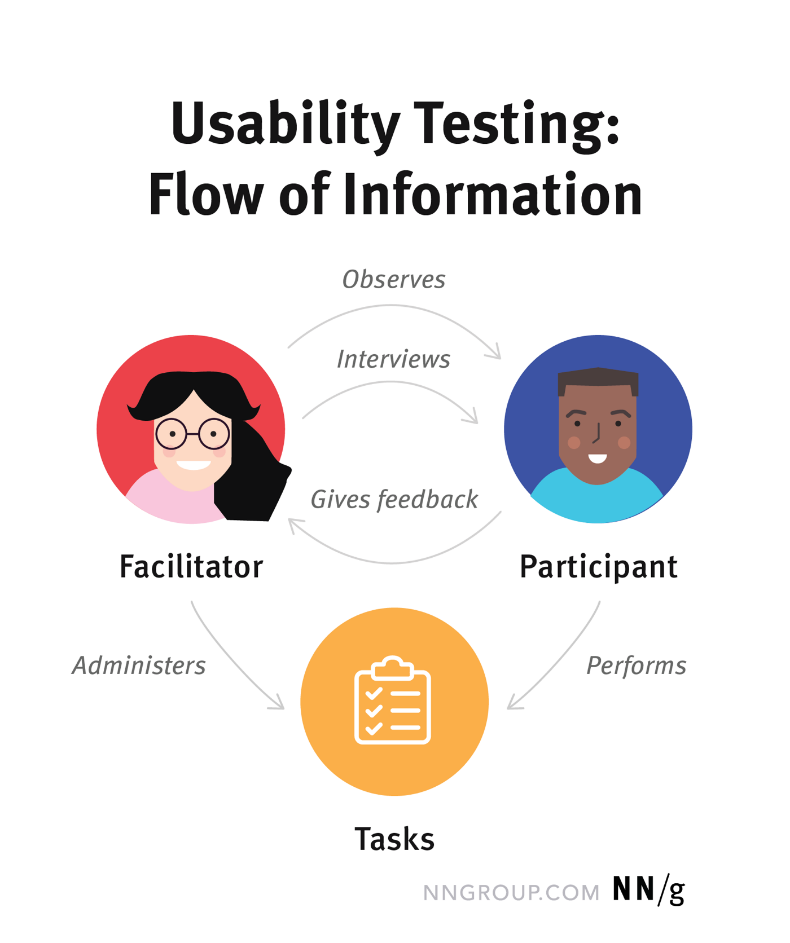
Laravel Seeding slúži na naplnenie tabuliek statickými dátami. Na vygenerovanie seedera môžeme použiť príkaz *php artisan make:seeder*. Seeder obsahuje iba jednu metódu a to run. Pre spustenie seedera používame metódu *php artisan db:seed*. Pri prvotnom spustení aplikácie na serveri vieme použiť príkaz *php artisan migrate:refresh –seed,* ktorý vytvorí a naplní databázu. (11)

#### Laravel Query Builder

Laravel Query Builder je nástroj Laravelu, ktorý ponúka vytváranie a spustenie databázových dopytov. Je možné ho použiť pri takmer všetkých databázových operáciách v aplikácii a funguje fantasticky so všetkými podporovanými databázovými systémami. Navyše je veľmi silno chránený proti SQL injection útokom. (12)

## Testovanie použiteľnosti

Testovanie použiteľnosti je typ testovania webovej stránky, aplikácie, alebo nejakého produktu, pri ktorom pozorujeme vybraných ľudí ako sa pokúšajú s produktom interagovať. (13)



Obrázok 9 – Testovanie použiteľnosti

Na testovanie potrebujeme prototyp aplikácie, prípadne už finálnu hotovú aplikáciu, testovacie subjekty, ktoré vykonávajú sadu zadaných testovacích scenárov a interagujú s daným prototypom. Ďalej takzvaného moderátora, ktorý si robí poznámky a pozoruje testovacie subjekty ako plnia zadané testovacie scenáre. Interakcia medzi testovacím subjektom a aplikáciou sa počas testovania použiteľnosti zvykne nahrávať. Testovanie sa zvykne vykonávať aj viac krát počas vývoja aplikácie, aby sa zistili nedostatky aplikácie včas. Počet testovacích subjektov na jedno testovanie je ideálne 5, nakoľko je ich dosť na to, aby sa potrebné nedostatky aplikácie objavili. 5 zároveň nie je priveľa, kde by sa mohlo stať že väčšina testovacích subjektov už nájde nedostatky len rovnaké ako tie predošlé. Od subjektov sa vždy moderátor snaží získať čo najväčšiu spätnú väzbu, najčastejšie formou diskusie alebo dotazníka. (13)

### Cieľ

Cieľom testovania použiteľnosti je zistiť, či je daná aplikácia, systém, web stránka alebo nejaký produkt jednoduchý, zrozumiteľný, bez nedostatkov, ktoré by cieľovej skupine mohli chýbať, prípadne či neobsahuje veci zbytočne naviac. (13)

### Použiteľnosť

Použiteľnosť je definovaná viacerými komponentmi, ktoré musíme sledovať. (13)

Používateľ

Zameranie aplikácie na používateľa je veľmi dôležitý parameter použiteľnosti, nakoľko snaha o uspokojenie cieľovej skupiny musí byť vždy prvoradá pre úspech aplikácie, systému, webovej stránky a podobne. (13)

Cieľ

Každé testovanie použiteľnosti a každý testovací scenár musí mať jasne stanovený cieľ, ktorý chceme aby používateľ pomocou aplikácie dosiahol. (13)

Efektívnosť

Používateľ by mal byť schopný sa ľahko a efektívne dostať k dosiahnutiu cieľa testovacie scenára, ideálne bez pomoci inej osoby. (13)

Satisfakcia

Aplikácia by mala v používateľovi zanechať subjektívne spokojný dojem. (13)

Zapamätateľnosť

Akpoužívateľ opätovne využíva aplikáciu, mal by si zapamätať postupy, ktoré nadobudol v predchádzajúcich testovacích scenároch, ktoré absolvoval. (13)

#### Kontext

Aplikácia by mala poskytovať iba funkcionalitu, na ktorú je určená. (13)

#### Výkon

Každý testovací scenár by mal obsahovať informáciu o optimálnom čase/počtu krokov na splnenie testovacieho scenára. (13)

### Testovací scenár

Testovacie scenáre sú časťou testovania použiteľnosti, kde každý scenár musí mať zadefinované poradie, v akom ho má testovací subjekt vykonať, musí mať jasne definovaný cieľ a optimálny čas na riešenie. Môže taktiež obsahovať rôzne následnosti testovacích scenárov podľa výsledku používateľovho testovania. Testovacie scenáre vieme rozdeliť podľa viacerých faktorov na: (14)

Efektívnosť systému

Zisťujeme, či je systém ľahko naučiteľný, či je užitočný pre cieľovú skupinu a či má systém priaznivý vzhľad.

Účinnosť systému

Skúmame, či je navigácia intuitívna a dostane nás kde potrebujeme, či formát stránok je jednotný a stránka poskytuje účinné vyhľadávanie.

Presnosť systému

Zisťujeme, či informácie a dáta nie sú zastarané, či sú presné a v aplikácii nechýbajú dôležité informácie.

Užívateľská prívetivosť

Ovládanie by malo byť intuitívne a ak sú nutné inštrukcie k ovládaniu, mali by byť prítomné.

# Návrh

V tejto kapitole si popíšeme návrh našej intranetovej aplikácie, uvedieme čo naša aplikácia poskytuje.

## Use cases

V tejto podkapitole si prejdeme jednotlivé prípady použitia pre rôzne typy používateľov systému.

V našej aplikácii rozlišujeme 4 typy userov.

#### Neprihlásený

Tento user sa môže jedine prihlásiť, nakoľko iba admin dokáže vytvárať userov.

#### Systémový user

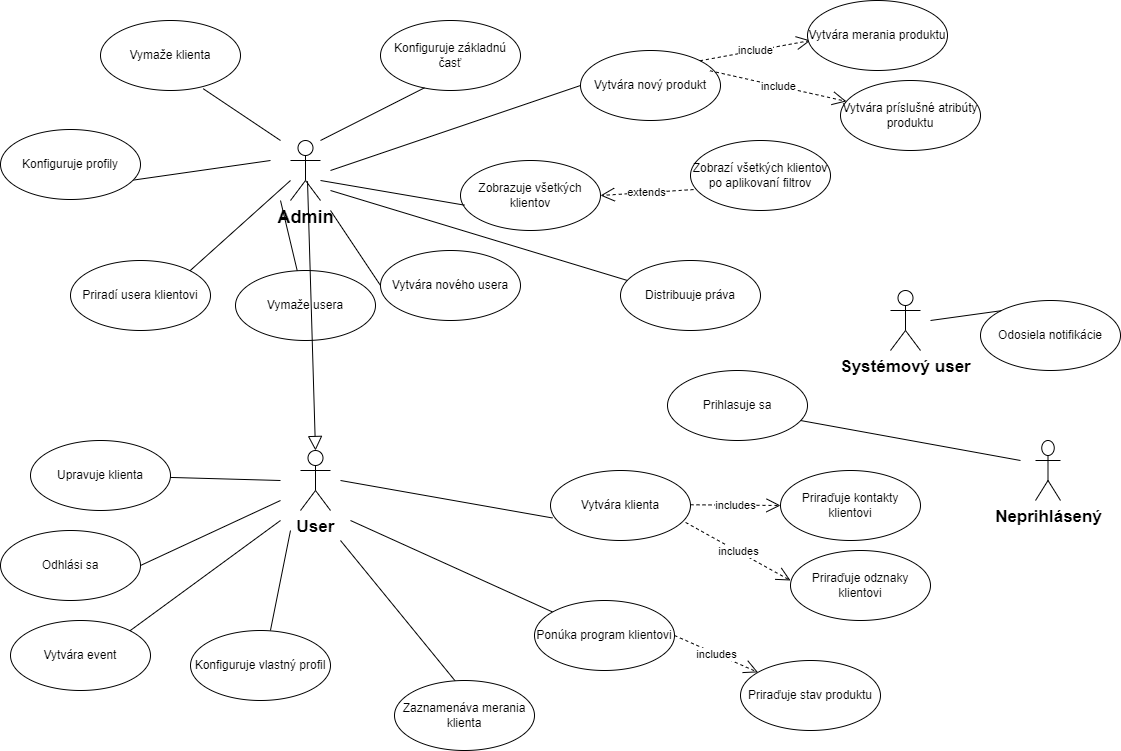
Úlohou tohto usera je posielať notifikácie užívateľom aplikácie a klientom mimo systém. Môže ich odoslať formou emailu, SMS, prípadne oboch.

#### User

User vie napríklad upraviť klientovi informácie, vie si konfigurovať svoj vlastný profil, vytvoriť novú udalosť, zaznamenať klientovi meranie, „ponúknuť“ (ďalej rozpísane v časti 2.3) klientovi produkt čo zahŕňa priradenie stavu produktu a možné vytvorenie klienta k čomu sa viaže priraďovanie odznakov a jeho kontaktov.

#### Admin

Admin môže napríklad vytvoriť alebo vymazať klienta alebo usera, konfigurovať profily, či už vlastný alebo aj iných userov a klientov, vie priradiť užívateľa klientovi, vytvoriť nový produkt ku ktorému sa viaže vytvorenie meraní pre daný produkt a ich príslušné atribúty. Admin vie taktiež zobrať všetkých klientov, prípadne aplikovať filtre. Je zodpovedný aj za distribúciu práv pre userov. Admin má automaticky všetky tieto práva. Jeho posledný špecifický prípad je konfigurácia základnej časti, čo znamená napríklad definovanie nových meeting roomov, nové odznaky, nové produkty a podobne. Admin má taktiež prístup k funkcionalite bežného usera.



Obrázok 10 - Use case diagram

## Dátový model

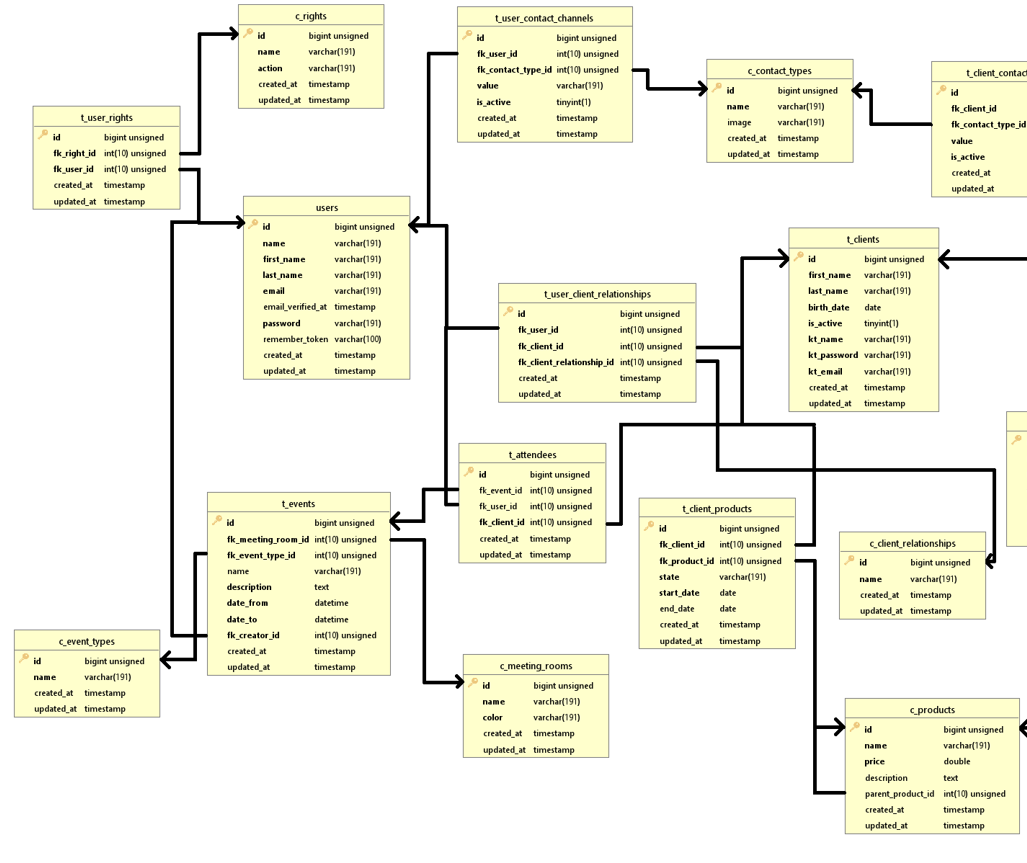
V tejto podkapitole si ukážeme nami navrhnutý dátový model použitý v našej práci, jeho základné informácie a opis.

Dátový model obsahuje 23 tabuliek, ktoré sú na obrázkoch nižšie rozmiestnené do pomyslených skupín ako používatelia (users), klienti, udalosti, produkty, vzťahy a merania. Všetky tabuľky obsahujú prefix c alebo t. Tieto prefixy znamenajú nasledovné:

Prefix c predstavuje statické dáta, teda dáta ktoré sa zvyčajne nemenia, prípadne naozaj zriedkavo. Upravovať ich môže iba admin a niektoré z nich iba databázový admin. Patria sem tabuľky rights, event\_types, meeting\_rooms, contact\_types, client\_relationships, products, measurings, measuring\_attributes, badges.

Prefix t predstavuje dáta, ktoré majú tendenciu sa meniť často. Tvorí ho aj užívateľ stránky, nie len admin. Patria sem tabuľky use\_rights, events, attendees, user\_contact\_channels, user\_client\_relationships, client\_products, clients, client\_contact\_channels, client\_measuring\_attributes, client\_badges, allowed\_measuring\_attributes, client\_measurings aclient\_measuring\_images.

Pre príklad v praxi si zoberme tabuľku c\_contact\_types a tabuľkut\_client\_contact\_channels. Admin vytvorí na začiatku kontaktné typy ako napríklad email, či telefónne číslo. Tieto už nemajú dôvod sa meniť, preto ich radíme medzi statické dáta. Naopak keď klientovi priradíme nejaký kontaktný typ, môže sa hocikedy zmeniť, nakoľko klient začne používať inú adresu, zmení si telefónne číslo a podobne. To budú presne tie dynamické dáta.



Obrázok 11 - Dátovy model - používatelia

Na obrázku môžeme vidieť kategóriu používateľov(userov), ktorí v našej aplikácii môžu mať definované práva, kontaktné údaje, udalosti a vzťahy s jednotlivými klientami.

User má pridelené kontaktné údaje, tie môžu byť aktívne a pokiaľ daný kontaktný údaj nie je aktuálny, je možné ho deaktivovať. Slúžia na posielanie notifikácií, kde tieto notifikácie sú realizované emailom, alebo SMS.

Ďalej máme kategóriu udalostí, ku ktorej sa viaže zasadačka v ktorej sa má daná udalosť konať, typ udalosti a samozrejme kto sa udalosti zúčastní. Pokiaľ sú účastníkmi iba useri, jedná sa o interný meeting, ako je školenie a pokiaľ je učastníkom aj klient, tak sa jedná napríklad o osobnú konzultáciu. Tvorcom udalosti je samotný user (creator\_id), ktorý si môže vybrať kto sa má na danej udalosti zúčastniť.

Práva sa budú pridelovať userovi podľa toho, akú časťprocesu zastáva. Môžeme mať napríklad administratívnu pracovníčku, ktorá klientov vytvára a zaznamenáva prijatie platby za produkt. Tak isto vie prideliť konzultanta jednotlivým klientom.Konzultant vie klientovi upraviť jednotlivé údaje, ako napríklad kontaktné typy, pridanie odznakov, alebo zaevidovať meranie. Práva budú tvorené na základe názvu a akcie, kde názov bude meno controllera a akcia jeho funkcia.



Obrázok 12 - Dátovy model - klienti

Tu na obrázku môžeme vidieť kategóriu klientov. K nim sa viažu kontaktné typy, jednotlivé produkty, merania a odznaky. Odznakom sa myslí špecifická informácia, ktorá kategorizuje klientov. Príkladom môže byť nejaká diagnóza, vernostný klient, informácia o tom že splnil stanovený cieľ atď. Taktiež podobne ako user, klient má kolekciu kontaktných údajov rozšírenú o korešpondenčnú a fakturačnú adresu.

Produkt obsahuje rôzne merania, ktorými si klient prejde. Každý atribút má nakonfigurovaný typ hodnoty ako je napríklad číslo, alebo text a taktiež má definovanú jednotku merania, ktorá môže byť percento, kilo, centimeter. Klient pri rôznych meraniach posiela aj fotky, aby videl svoje vizuálne rozdiely. Produkt môže mať ďalšie pod-produkty, ktoré môže absolvovať po splnení rodičovského produktu.

## Stavový diagram pre produkt

V tejto podkapitole si prejdeme stavy pre produkt klienta.

Keďže naša aplikácia je intranetová, klient k nemu nemá žiaden prístup. Klientom sa produkt ponúka mimo aplikácie, s tým že v aplikácii sa zaznačí ponúknutie programu klientovi.

Klientovi teda na začiatku produkt ponúkneme [offered]. Ponúknutý produkt samozrejme môže odmietnuť [rejected]. Klient ale o produkt záujem mať môže [accepted], vtedy mu vystavíme faktúru. Od tohto momentu sa môže v akomkoľvek bode stať, že klient nechce s produktom pokračovať [rejected], alebo si ho môže z objektívnych dôvodov (napríklad choroba, čakanie na peniaze) pozastaviť [paused]. Po zaevidovaní platby administratívnym pracovníkom klient začne aktívne využívať produkt [started]. Klient následne pošle vyplnený dotazník a začne jeho prvé meranie [in progress]. Po úspešnom absolvovaní produktu a záverečného merania klient produkt dokončí [finished].



Obrázok 13 - Stavový diagram pre produkt

# Implementácia

## Implementačné jazyky a knižnice

Pre našu aplikáciu sme zvolili využitie frameworku Laravel, ktorý je implementovaný v jazyku Php. Poskytuje nám rozhranie na komunikáciu back-endu s databázou a generovanie stránky na strane servera.

Nakoľko pri každej požiadavke usera na server nastáva generovanie stránky nanovo, sme zvolili Java-scriptovú knižnicu sweetalert2. Táto knižnica nám poskytuje modálne okná zobrazujúce časti aplikácie, na ktoré nie je potrebná samostatná stránka.

Pri vytváraní a upravovaní záznamov v aplikácii často využívame Html objekt select, ktorý ale prehliadače štandardne poskytujú bez vyhľadávania nad možnosťami selectu. Pre prívetivejšie používanie sme zvolili Java-scriptovú knižnicu SlimSelect, ktorá poskytuje krajšie a jednoduchšie zaobchádzanie s Html objektom select.

V aplikácii umožňujeme vytváranie udalostí pre klientov a userov. Tieto udalosti sme sa rozhodli zobrazovať v kalendárovom rozhraní FullCalendar. Jedná sa o Java-scriptovú knižnicu, ktorá poskytuje zobrazenie kalendára na dennej, týždňovej a mesačnej báze. Tento kalendár sme spojili s Java-scriptovou knižnicou sweetalert2, kde po zakliknutí udalosti sa nám zobrazí modálne okno o detaile tejto udalosti.

## Organizácia kódu

Keďže sme využili Laravel framework, ktorý využíva MVC architektúru, organizácia nášho kódu sa delí na 3 hlavné časti.

#### /app

Na tejto ceste sa nachádzajú všetky modely našej aplikácie. Pomocou týchto modelov riadime spracovanie a získavanie údajov voči databáze. V rámci modelov poskytujeme aj naše vlastné mapovania medzi jednotlivými modelmi databázy pomocou privátnych premenných a verejných funkcií. Taktiež si v nich aj definujeme názvy stĺpcov, s ktorými vie aplikácia narábať.

#### /resources/views

V tejto časti sa nachádzajú všetky views, ktoré slúžia ako šablóny na generovanie stránok pre klienta. V pod priečinku /layouts sa nachádza základná kostra každej webovej stránky ktorú poskytuje naša aplikácia. Nachádzajú sa v nej základné Java-scriptové funkcie, generovanie hlavičky stránky a navigácie. /windows obsahuje views pre modálne okná, ktoré sa získavajú volaním Java-scriptových funkcií na požiadavku klienta.

#### /app/Http/Controllers

V tomto priečinku vieme nájsť controllery, kde každý jeden controller prislúcha modelu. V controlleroch sú okrem základných volaní viewov definované navigačné linky $links definujúce možné akcie, ktoré vidí klient v navigácii stránky. Na získanie všetkých navigačných linkov má každý controller definovanú funkciu getLinks(), ktorý si získa kolekciu navigačných linkov pre danú akciu. Taktiež v konštruktore controllera definujeme middleware-y obsahujúce kontrolu autentifikácie usera a skontrolovanie práv na akciu ktorú chce vykonať.

#### /app/Enums

Nakoľko v aplikácii využívame Php verziu 7.x nevieme ešte definovať typ objektu enum. Tieto sme si definovali pomocou objektu, konštánt v objekte a statických funkcií.

## Práva

Práva sú staticky definované v celej aplikácii. Aké práva poznáme si uchovávame v tabuľke c\_rights a práva, ktoré majú useri priradené v t\_user\_rights. Tieto práva vie priraďovať iba admin jednotlivým userom, ale nevie vytvoriť nové. V aplikácii máme definované samostatné akcie na uloženie alebo upravenie záznamu do databázy, ktoré sa vyhodnotia rovnako ako by sa jednalo o vytvorenie alebo upravenie záznamu userom.

Pomocou práv je možné prideliť vytváranie a úpravu nových odznakov, nových typov udalostí, konfiguráciu produktov, zasadačiek a vytváranie nových klientov.

Kontrolu týchto práv riešime v middleware-y v CheckRights. V tejto funkcionalite si zistíme o akú akciu sa prihlásený používateľ pokúša a vyhodnotíme, či má dostatočné práva na danú akciu. Pokiaľ je prihlásený používateľ adminom, má na ľubovoľnú akciu vždy právo. Pokiaľ neexistuje právo na danú akciu, tak na ňu má štandardne prihlásený používateľ právo. Ak existuje právo na danú akciu, musíme skontrolovať či prihlásený používateľ toto právo má. Ak ho ale nemá, vráti klientovi chybovú stránku.

### Práva na klienta

Vytvoriť klienta dokáže iba používateľ, ktorý je adminom, alebo má priradené špeciálne právo na vytvorenie klienta. Akonáhle je klient vytvorený, je možné mu prideliť userov, ktorí ho spravujú. Pre organizáciu userov priradených klientovi sme zvolili definovanie vzťahu medzi klientom a userom. Prístup ku klientovi je možný iba userom, ktorí majú s klientom nejaký vzťah. Títo useri môžu klientovi ponúknuť produkt, zaznamenávať merania, definovať kontaktné typy a odznaky a plánovať jednotlivé udalosti s klientom.

## Vytváranie udalostí

Vytvoriť udalosť je umožnená každému užívateľovi aplikácie. Ako zúčastnených si vie zvoliť zo všetkých klientov a používateľov. V rámci jednej udalosti si user definuje zasadačku, zoznam klientov a userov ktorí sa udalosti zúčastnia. Danú udalosť vie upravovať alebo zrušiť iba user, ktorý ju vytvoril. Admin nedokáže udalosť, ktorú vytvoril iný používateľ upraviť alebo zrušiť.

Danú udalosť je možné vytvoriť zo stránky klienta, zasadačky alebo hlavnej navigácie.

## Stavový automat pre produkty

V objekte State sme si zadefinovali konštanty pre všetky možné stavy produktu.

const*OFFERED* = 'OFFERED';  
const*REJECTED* = 'REJECTED';  
const*ACCEPTED* = 'ACCEPTED';  
const*STARTED* = 'STARTED';  
const*PAUSED* = 'PAUSED';  
const*IN\_PROGRESS* = 'IN\_PROGRESS';  
const*FINISHED* = 'FINISHED';

Určili sme si koncové stavy, z ktorých už neexistuje možnosť prejsť do iného stavu. Poznáme stav rejected, ktorý predstavuje klientove odmietnutie produktu, prípadne predčasné skončenie programu. Taktiež máme stav finished predstavujúci úspešné ukončenie produktu.

publicconst*FINAL\_STATES* = [  
self::*REJECTED*,  
self::*FINISHED*];

Zadefinovanie možných a povolených prechodov z jedného stavu do druhého stavu, presne ako je to znázornené v diagrame vyššie, sme si zadefinovali pomocou slovníka, kde každý jeden stav má povolenú množinu stavov, na ktoré môže prejsť.

publicconst*ALLOWED\_CHANGES* = [  
self::*OFFERED* => [  
self::*REJECTED*,  
self::*ACCEPTED*],  
self::*REJECTED* => [],  
self::*FINISHED* => [],  
self::*ACCEPTED* => [  
self::*STARTED*,  
self::*REJECTED*,  
self::*PAUSED*],  
self::*STARTED* => [  
self::*PAUSED*,  
self::*IN\_PROGRESS*,  
self::*REJECTED*],  
self::*PAUSED* => [  
self::*IN\_PROGRESS*,  
self::*REJECTED*],  
self::*IN\_PROGRESS* => [  
self::*PAUSED*,  
self::*FINISHED*,  
self::*REJECTED*]  
];

Pre definovanie stavov, ktorým môžeme zadávať merania, sme si definovali konštantu RUNNING\_STATES. Stav started popisuje stav produktu, kedy je produkt zakúpený, ale ešte klient neodoslal úvodný dotazník. Pokiaľ je produkt in\_progress, tak je možné zadávať priebežné a koncové merania.

publicconst*RUNNING\_STATES* = [  
self::*STARTED*,  
self::*IN\_PROGRESS*];

Pokiaľ je prihláseným userom admin, stavový diagram sa nevyhodnocuje a adminovi je umožnené zadávať ľubovoľné zmeny stavu produktu.

# Testovanie

V tejto časti i popíšeme nami navrhnuté testovacie scenáre, priebeh testovania a vyhodnotíme výstup z testovania.

## Cieľ

Cieľom nášho testovania a nami navrhnutých testovacích scenárov je otestovať správne fungovanie aplikácie ako celku, jej používateľskú prívetivosť a odhaliť možné chyby, nedostatky, neintuitívne akcie, ktoré môže užívateľ vykonať v rámci systému.

Testovanie prebehlo v rámci firmy, ktorej daný systém bude používať. Testovacie scenáre boli navrhnuté na základe firemného procesu. Počas testovania sme sledovali ako sa tester orientuje v našej aplikácii, s ktorými testovacími scenármi mal väčšie problémy a ktoré boli naopak pre testera triviálne.

## Priebeh

Priebeh testovania prebehol priamo vo firme za účasti testera a moderátora. Testerov sme rozdelili na dve skupiny, kde prvou skupinou bol majiteľ, ktorý bude mať právo admina v aplikácii a druhou skupinou boli radoví zamestnanci. Pre každú skupinu bola vytvorená séria po sebe nasledujúcich testov, kde každá séria obsahovala 4 testovacie scenáre. Každá skupina absolvovala svojou sériou testov a navyše admin absolvoval testy určené aj pre rádových zamestnancov.

Moderátorom počas testovania bol autor práce, ktorý zaznamenával testovanie jednotlivými testermi pomocou zvukovej a videovej nahrávky a vlastných poznámok.

Po ukončení testovania prebehla krátka diskusia s testerom o jeho dojme o aplikácii.

## Testovacie scenáre

Testovacie scenáre sme tvorili podľa firemného procesu. Každý scenár obsahuje identifikátor, ktorý využívame na definovanie predpokladov v ďalších scenároch. Cieľ stručne popisuje, čo sa v danom scenáre dosiahne. Vstupy definujú dáta, ktoré tester potrebuje na úspešné splnenie testovacieho scenára. Predpoklady definujú, kedy je možné daný scenár splniť. Postup ktorý má tester zvoliť popisujeme v časti Kroky. Vyhodnotenie úspešnosti scenáru sú definované v časti Kritériá úspešnosti.

### Scenáre pre admina

V nasledujúcich tabuľkách sú definované testovacie scenáre určené pre admina aplikácie, v ktorých testujeme tvorbu nového používateľa, vytvorenie konfigurácií aplikácie, naplánovanie udalosti s klientom a používateľom a vytvorenie nového produktu s jedným meraním.

|  |  |
| --- | --- |
| Identifikátor | TEST-ADMIN-1 |
| Cieľ | Vytvorenie nového usera |
| Vstupy | * User name: * Meno: Jožko * Priezvisko: Mrkvička * Email: jozko.mrkvicka@bodyclinic.sk * Heslo: heslo123 * Kontakty:   + Adresa: Mlynská dolina, 811 04 Bratislava   + Tel.č.: 0954 129 876   + Email: mrkva@mail.sk |
| Predpoklady | Používateľ admin a kontaktné typy sú vytvorené |
| Kroky | 1. Prihlásenie sa pod svojim používateľom(admin) 2. Vytvorenie používateľa cez webový formulár podľa zadaných vstupov 3. Priradenie jeho kontaktov |
| Kritéria úspešnosti | Testovací subjekt úspešne vytvoril používateľa podľa zadaných vstupov |

|  |  |
| --- | --- |
| Identifikátor | TEST-ADMIN-2 |
| Cieľ | Vytvorenie nového typu udalosti a zasadačky |
| Vstupy | * Meno udalosti: Status meeting * Meno zasadačky: Laravelroom   + Farba zasadačky: Odtieň modrej |
| Predpoklady | Bez predpokladov |
| Kroky | 1. Vytvorenie typu udalosti a zasadačky cez webový formulár podľa zadaných vstupov |
| Kritéria úspešnosti | Testovací subjekt úspešne vytvoril typ udalosti podľa zadaných vstupov |
| Identifikátor | **TEST-ADMIN-3** |
| Cieľ | Vytvorenie novej udalosti s klientom používateľom |
| Vstupy | * Klient: Ľubovoľný * Používateľ: Jožko Mrkvička * Typ udalosti: Status meeting   + Zasadačka: Laravelroom |
| Predpoklady | Test TEST-ADMIN-1 a TEST-ADMIN-2 bol úspešne zvládnutý |
| Kroky | 1. Vytvorenie udalosti |
| Kritéria úspešnosti | Testovací subjekt úspešne vytvoril novú udalosť s používateľom |

|  |  |
| --- | --- |
| Identifikátor | TEST-ADMIN-4 |
| Cieľ | Vytvorenie nového produktu a jeho jedného merania |
| Vstupy | * Meno: Výživový mentoring * Cena: 499.99€ * Popis: To chceš! * Meranie   + Meno: Priebežné meranie   + Popis: na týždennej báze * Zoznam atribútov:   + Výška, obvod pása, obvod hrude, boky |
| Predpoklady | Atribúty v aplikácii už existujú |
| Kroky | 1. Vytvorenie produktu 2. Vytvorenie merania 3. Pridanie atribútov k meraniu |
| Kritéria úspešnosti | Testovací subjekt úspešne produkt a meranie podľa zadaných vstupov |

Tabuľka 1 - Scenáre pre admina

### Scenáre pre používateľa

V testovacích scenároch pre používateľa testujeme vytvorenie nového klienta, priradenie produktu, zaznamenanie merania, priradenie odznakov a naplánovanie stretnutia s klientom.

|  |  |
| --- | --- |
| Identifikátor | TEST-USER-1 |
| Cieľ | Vytvorenie nového klienta a ponúknutie produktu |
| Vstupy | * Meno: Jozef * Priezvisko: Mak * Dátum narodenia: 24.12.1999 * Kalorické tabuľky:   + Užívateľké meno: mak24   + Heslo: secret   + Email: jozef.mak@mail.sk * Kontakty:   + Adresa: Maková úžina 5, 841 00 Poprad   + Tel.č.: 0989 546 123   + Email: jozef.mak @mail.sk |
| Predpoklady | Testovací subjekt má vytvoreného používateľa. |
| Kroky | 1. Prihlásenie sa pod svojim používateľom 2. Vytvorenie klienta cez webový formulár podľa zadaných vstupov 3. Pridanie ďalších vlastností klientovi na základe vstupov 4. Ponúknutie produktu (Intenzívny program zmeny stravovania) na ľubovoľný termín |
| Kritéria úspešnosti | Testovací subjekt úspešne vytvoril klienta a klientovi bol ponúknutý produkt. |

|  |  |
| --- | --- |
| Identifikátor | TEST-USER-2 |
| Cieľ | Aktivovanie produktu klientovi a zapísanie údajov z merania |
| Vstupy | * Klient: Jozef Mak * Meranie: Priebežné meranie * Poradie: 1 * Dátumy: dnešné * Namerané hodnoty:   + Váha: 80   + Obvod pása: 90   + Obvod hrude: 120   + Boky: 120   + Biceps: 37   + Horné stehno: 60   + Dolné stehno: 50 |
| Predpoklady | Test TEST-USER-1 bol úspešne zvládnutý. |
| Kroky | 1. Klientovi aktivujeme produkt 2. Klientovi zaznamenáme namerané hodnoty |
| Kritéria úspešnosti | Testovací subjekt úspešne aktivoval produkt a vyplnil informácie z merania. |

|  |  |
| --- | --- |
| Identifikátor | TEST-USER-3 |
| Cieľ | Priradenie odznakov klientovi |
| Vstupy | * Zoznam odznakov:   + Celiatik, Štítna žľaza, Športovec |
| Predpoklady | Test TEST-USER-1 bol úspešne zvládnutý. |
| Kroky | 1. Klientovi postupne pridáme odznaky |
| Kritéria úspešnosti | Testovací subjekt úspešne priradil odznaky klientovi. |

|  |  |
| --- | --- |
| Identifikátor | TEST-USER-4 |
| Cieľ | Naplánovanie klientovi konzultáciu s jeho konzultantom |
| Vstupy | * Klient: Jozef Mak * Zasadačka: Zasadačka 2 * Typ udalosti: Osobná konzultácia * Meno: Konzultácia na požiadavku klienta * Dátum od: dnes o 10:00 * Dátum do: dnes o 10:45 |
| Predpoklady | Test TEST-USER-2 bol úspešne zvládnutý. Je vytvorená zasadačka kde sa konzultácia bude konať. |
| Kroky | 1. Vytvoríme novú udalosť na stanovený čas 2. Priradíme danej udalosti zasadačku, v ktorej sa má konať |
| Kritéria úspešnosti | Testovací subjekt úspešne vytvoril udalosť na stanovený čas, na danú zasadačku pre daného klienta a jeho konzultanta. |

Tabuľka 2 - scenáre pre používateľa

## Vyhodnotenie

Na základe získaných poznatkov z testovania a opätovného prehratia záznamov z testovania sme dospeli ku nasledovným záverom.

Pre testerov, ktorí testovali scenáre pre admina bolo najľahším testom test TEST-ADMIN-2, kde testeri mali vytvoriť zasadačku a typ udalosti. Najťažším testom bolo vytvorenie nového produktu (TEST-ADMIN-4).

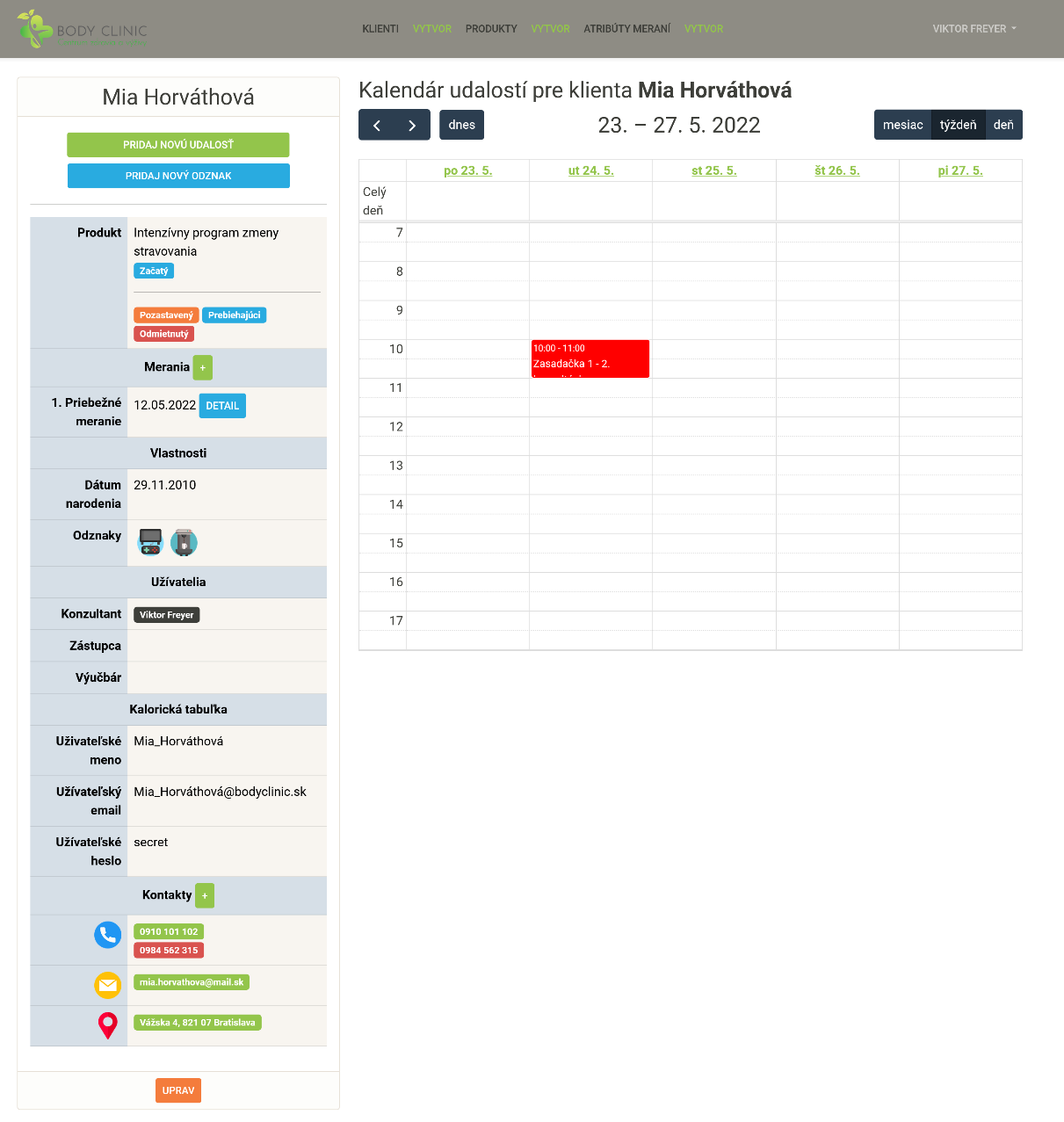
Testeri, ktorí testovali používateľské scenáre prevažne nemali problém so žiadnym testom. Každý tester splnil všetky zadania, ale nebolo pre nich prívetivé zaznamenávanie merania, ktoré bolo robené pomocou vyskakovacieho okna, ktoré sa po každej zmene hodnoty zavrelo a bolo potrebné ho opätovne otvoriť.

### Zistené chyby pri testovaní

Pri teste TEST-USER-3 bolo možné priradiť ten istý odznak klientovi, ktorý už bol priradený. Upravili sme filter nad možnými odznakmi, ktorý vylúči odznaky ktoré klient už má priradené.

Pri testovaní sa zistilo, že niekde nebol preklad do slovenského jazyka.

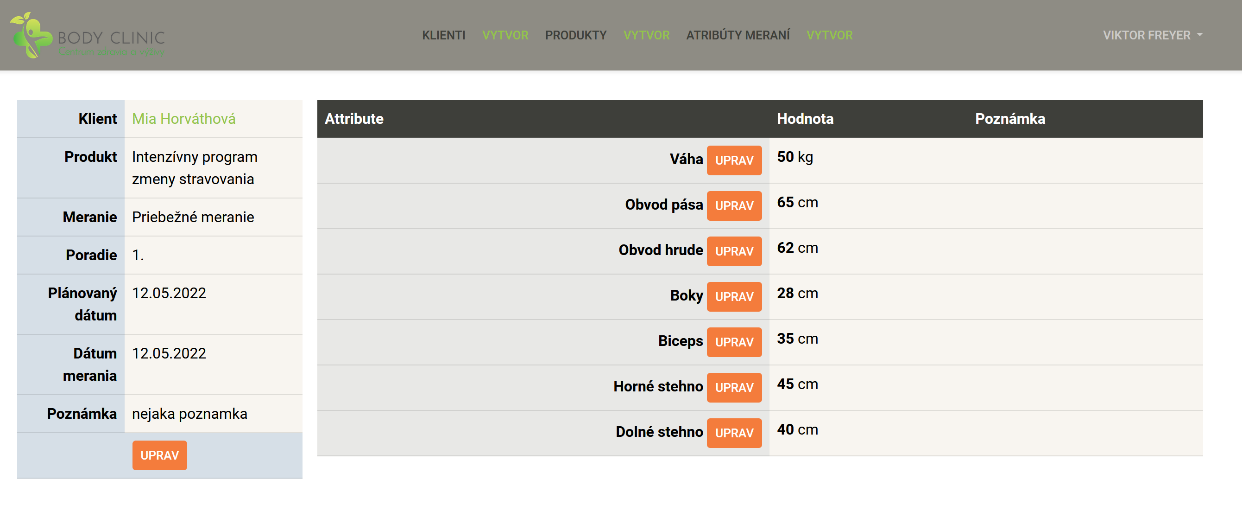
Pri zmene stavu produktu z prijatý do začatý chýbala možnosť odmietnuť produkt.



Obrázok 14 - Detail klienta

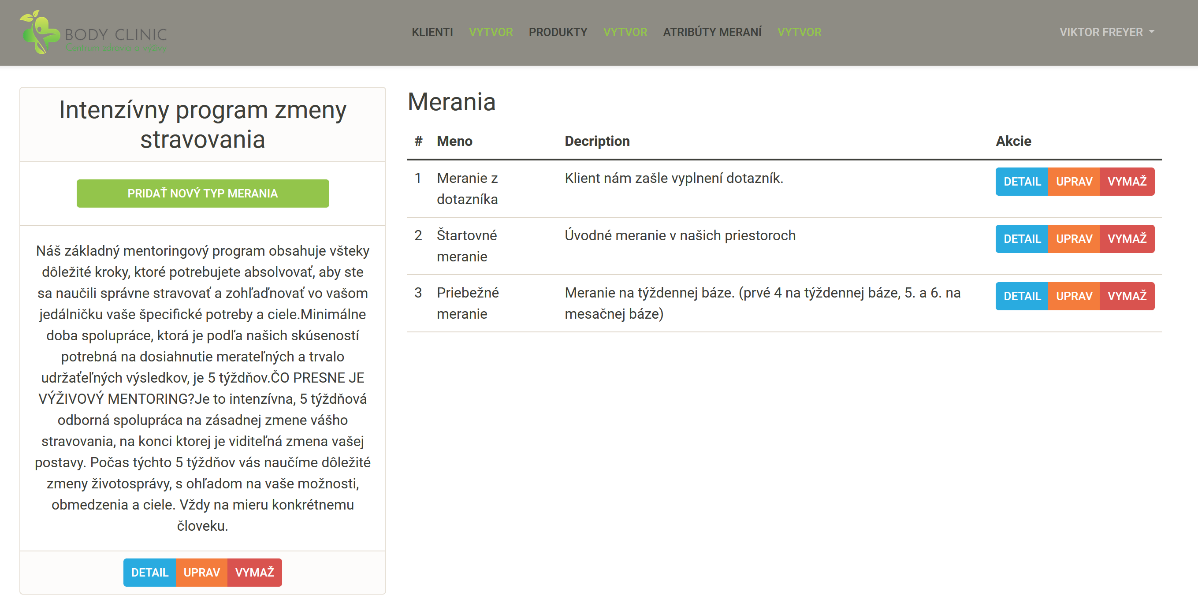
### Úpravy aplikácie

Na základe pripomienok testerov, sme popracovali na používateľskej prívetivosti aplikácie, kde sme zmenili vyskakovacie okno pri zaznamenávaní merania na samostatnú webovú stránku. Taktiež sme pri výpise všetkých klientov pridali informáciu o aktuálnom produkte klienta.



Obrázok 15 - Zaznamenávanie merania

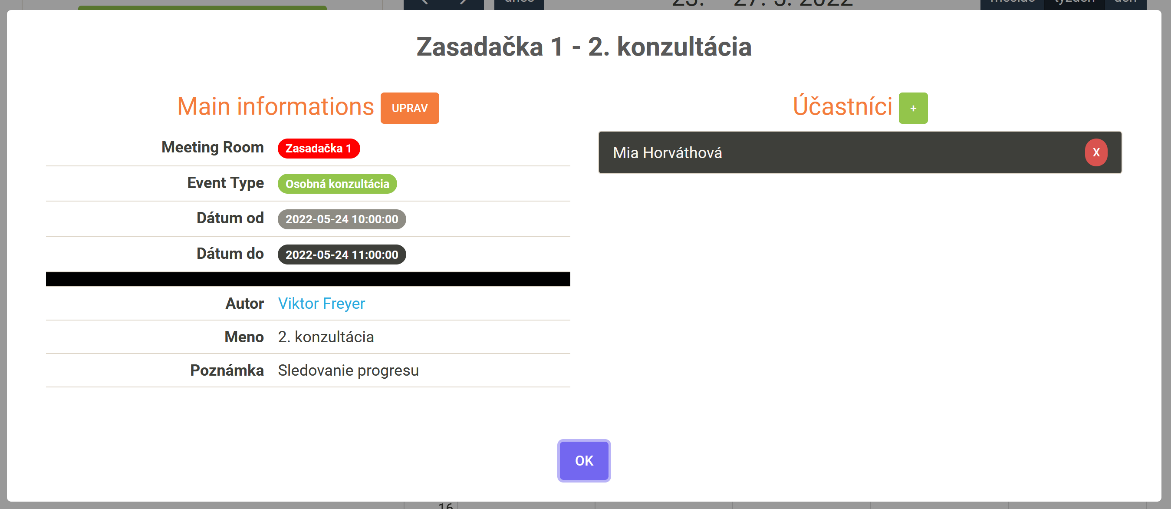
Pri teste TEST-USER-2 sme vypozorovali, že testerovi sa niekedy podarilo zakliknúť iný stav produktu ako bolo plánované a na základe tohto pozorovanie sme doplnili dodatočné potvrdzovacie okno o zmene stavu produktu a pridali sme adminovi možnosť meniť stav produktu mimo pravidiel stavového automatu pre produkt.



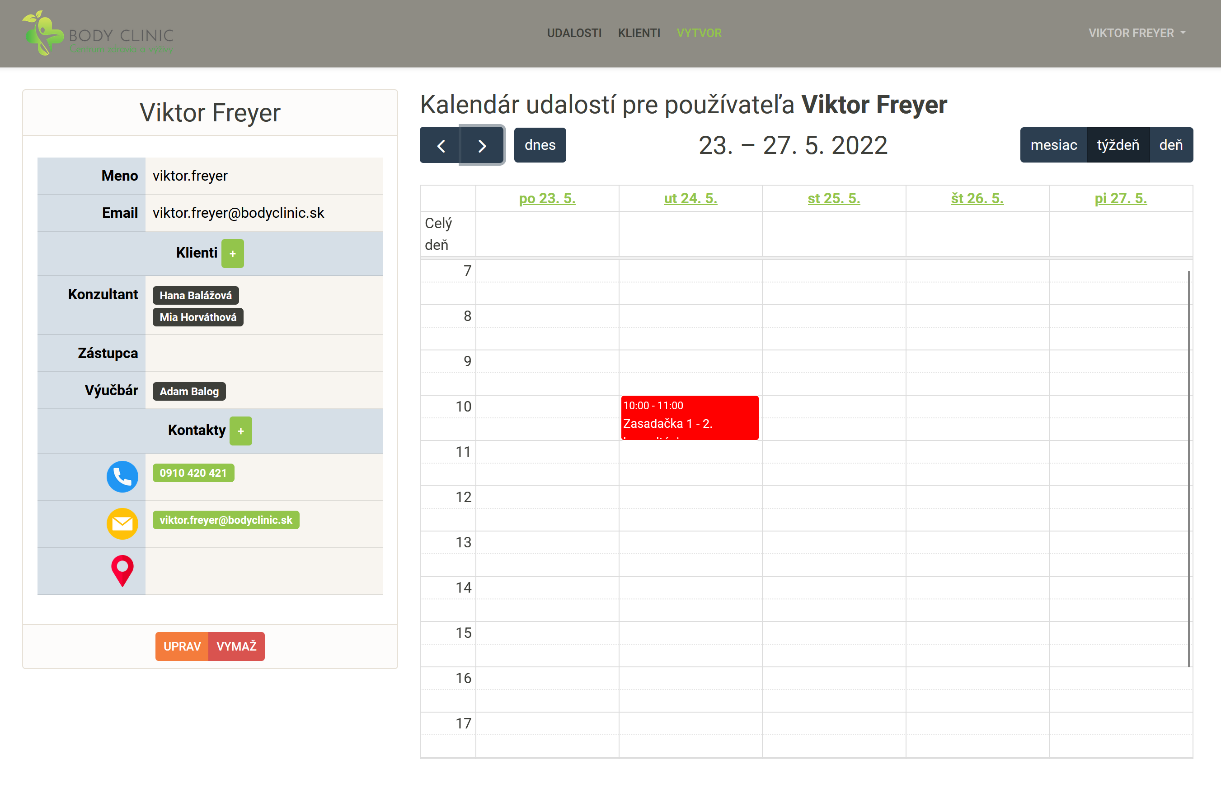
Obrázok 16 - Detail produktu

Pokiaľ sa chcel tester v TEST-USER-2 vrátiť na detail klienta, nebol na stránke žiaden prelink. Tento prelink bol pridaný.

Na základe postrehov admin testera sme pridali pri zobrazení všetkých používateľov informáciu o aktuálnom počte klientov rádového zamestnanca. V prehľade klientov sme pôvodne zoraďovali klientov podľa priezviska a mena, ale testerom prišlo efektívnejšie najprv zoradiť klientov, podľa toho či sú alebo nie sú aktívny a až následne ich zoradiť podľa priezviska a mena.



Obrázok 17 - Detail udalosti



Obrázok 18 - Detail používateľa

# Záver

Cieľom bakalárskej práce bolo vytvoriť intranetovú aplikáciu v oblasti výživového mentoringu. Vytvorili sme danú aplikáciu a umožnili rádovým zamestnancom vytvárať klientov, priraďovať im kontakty, rôzne základné informácie, priraďovať im odznaky a priraďovať im rôzne merania zahrnuté v produktoch. Adminovi sme navyše umožnili vytvárať užívateľov, konfigurovať profily, priraďovať rádových zamestnancov klientom, vytvárať zasadačky, odznaky a typy udalostí. Aplikácia taktiež umožňuje všetkým užívateľom plánovať udalosti, či už firemné alebo s klientom, ktorá sa napíše do kalendára klientovi, užívateľom a aj zasadačke kde sa udalosť má konať.

Vytvorenú aplikáciu sme testovali na zamestnancoch firmy pre ktorú je daná aplikácia vyvíjaná a myslíme si, že je po menších zmenách veľmi nápomocná pre proces zaužívaný v danej firme. Firma už nemusí používať aplikácie ako Google Kalendár, zapisovanie údajov klientov do excelov a nepotrebuje diskové pole na ukladanie dát. Okrem vytvorenia aplikácie sme sa taktiež naučili ako pracovať s open source PHP frameworkom Laravel.

V budúcnosti by bolo dobré rozšíriť aplikáciu o zobrazovanie štatistických diagramov nad klientom ako jedincom a nad vyfiltrovanou skupinou klientov na základe údajov z meraní. Aplikácia by taktiež mohla zahŕňať vlastnú implementáciu kalorických tabuliek, ktorá by komunikovala s vytvorenou mobilnou aplikáciou kde by si klient vedel nastaviť svoje makro nutrienty (bielkoviny, sacharidy, tuky) a kalórie a tvoriť si vlastné jedálne lístky. Mobilná aplikácia by umožňovala vytvorenie jedálneho lístka na základe balíčka potravín, ktoré v aplikácii budú predvolené a navyše by si mohol povytvárať aj vlastné. Mobilná aplikácia by taktiež notifikovala klienta o blížiacich sa udalostiach, novej ponuke produktov a ďalších možných ponukách firmy.

# Použitá literatúra

1. **TechTarget.** [Online] [Cited: December 30, 2021.] https://whatis.techtarget.com/definition/intranet.

2. **Educative.** [Online] [Cited: 12 31, 2021.] https://www.educative.io/blog/mvc-tutorial.

3. **ITnetwork.** [Online] [Cited: 12 31, 2021.] https://www.itnetwork.cz/navrh/mvc-architektura-navrhovy-vzor.

4. **Kuruc, Roman.** Intranet s webovým rozhraním pre sektor softwarového testovania. 2018.

5. **Hajná, Andrea.** Informačný Systém Pre Materské Školy. 2019.

6. **Nutriadapt.** [Online] [Cited: 1 5, 2022.] https://www.nutriadapt.sk/.

7. **Balance, Metabolic.** [Online] [Cited: 1 5, 2022.] https://www.metabolic-balance.sk/.

8. **Laravel.** [Online] [Cited: 1 5, 2022.] https://laravel.com/.

9. —. https://laravel.com/docs/8.x/artisan. [Online] [Cited: 1 5, 2022.]

10. —. [Online] [Cited: 1 5, 2022.] https://laravel.com/docs/8.x/migrations.

11. —. [Online] [Cited: 1 5, 2022.] https://laravel.com/docs/8.x/seeding.

12. —. [Online] [Cited: 1 5, 2022.] https://laravel.com/docs/8.x/queries.

13. **Digitalmag.** [Online] [Cited: 1 6, 2022.] https://www.digitalmag.sk/testovanie-pouzitelnosti-2/.

14. **Guru99.** [Online] [Cited: 1 7, 2022.] https://www.guru99.com/usability-testing-tutorial.html#3.

# Zoznam obrázkov

[Obrázok 1 - Základná schéma intranetu 11](#_Toc104230922)

[Obrázok 2 - Základná schéma mvc architektúry 13](#_Toc104230923)

[Obrázok 3 - Grafické rozhranie intranetu Romana Kuruca 14](#_Toc104230924)

[Obrázok 4 - Grafické rozhranie aplikácie Andrei Hajnéj 15](#_Toc104230925)

[Obrázok 5 - webová stránka nutriadaptu 1 16](#_Toc104230926)

[Obrázok 6 - webová stránka nutriadaptu2 16](#_Toc104230927)

[Obrázok 7 - webová stránka MetabolicBalance 17](#_Toc104230928)

[Obrázok 8 - ako funguje laravel 17](#_Toc104230929)

[Obrázok 9 – Testovanie použiteľnosti 19](#_Toc104230930)

[Obrázok 10 - Use case diagram 23](file:///C:\Users\freye\Downloads\bakalarka-RH-2-1%20(1).docx#_Toc104230931)

[Obrázok 11 - Dátovy model - používatelia 25](#_Toc104230932)

[Obrázok 12 - Dátovy model - klienti 26](#_Toc104230933)

[Obrázok 13 - Stavový diagram pre produkt 27](#_Toc104230934)

[Obrázok 14 - Detail klienta 37](#_Toc104230935)

[Obrázok 15 - Zaznamenávanie merania 38](#_Toc104230936)

[Obrázok 16 - Detail produktu 38](#_Toc104230937)

[Obrázok 17 - Detail udalosti 39](#_Toc104230938)

[Obrázok 18 - Detail používateľa 39](#_Toc104230939)

Zoznam tabuliek

[Tabuľka 1 - Scenáre pre admina 34](#_Toc104230940)

[Tabuľka 2 - scenáre pre používateľa 36](#_Toc104230941)