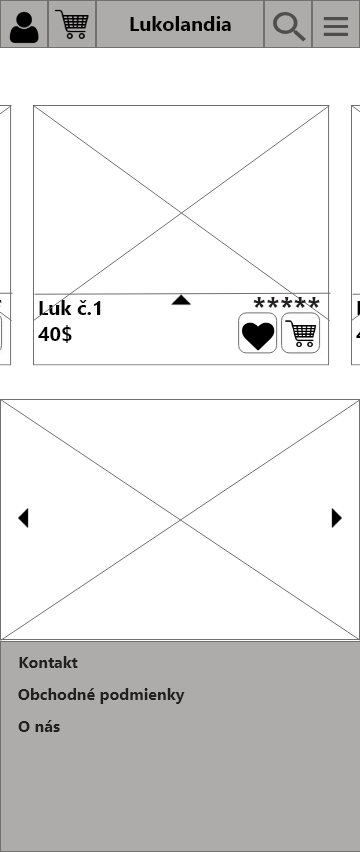
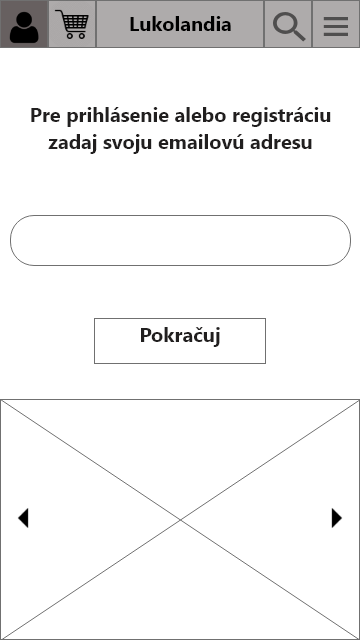
**Skice rozhrania – mobilná veľkosť**

**Landing page**

****

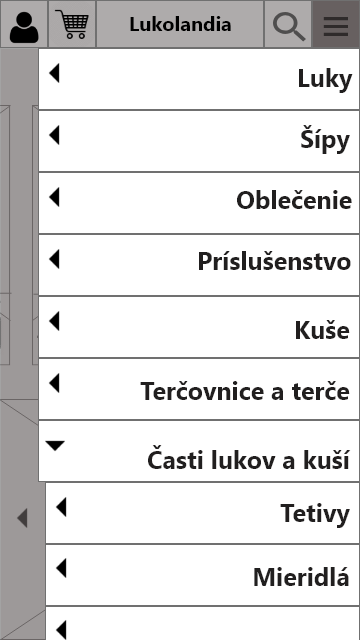
**Prihlásenie**

****

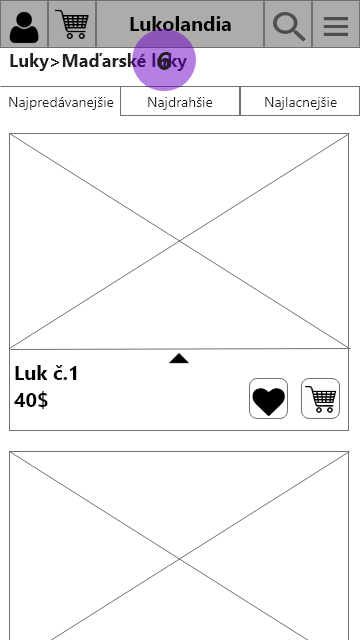
**Registrácia**

****

**Menu**

****

**Zobrazenie kategórie produktov**

****

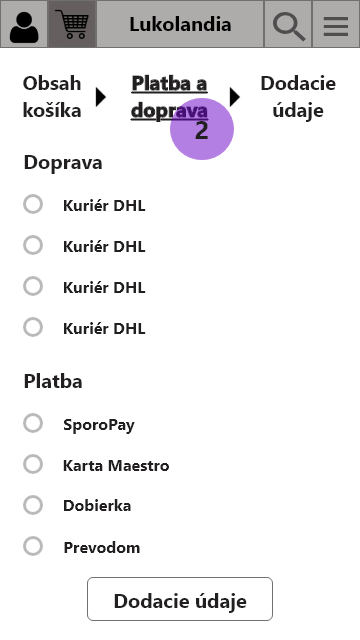
**Detail produktu**

****

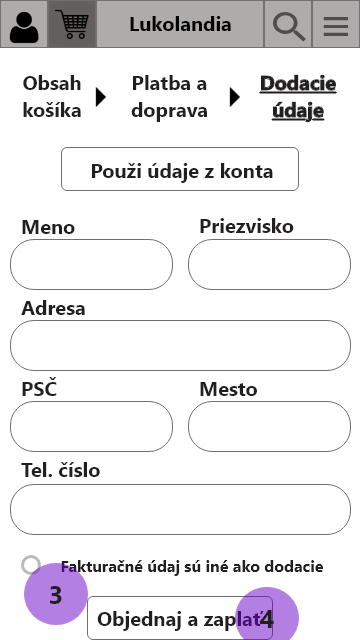
**Nákupný košík prvá časť**

****

**Nákupný košík-druhá časť**

****

**Nákupný košík-tretia časť**

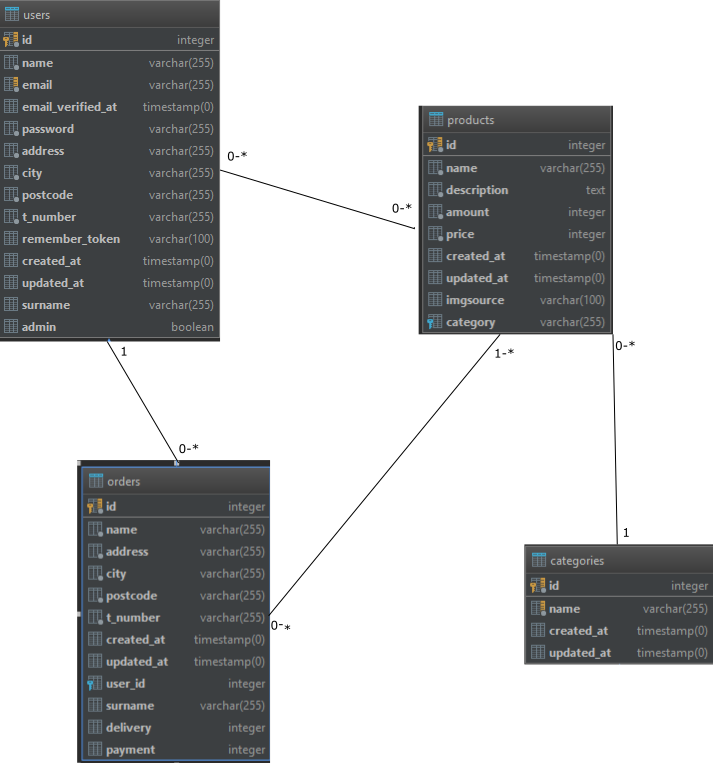
****

**Dátový model**

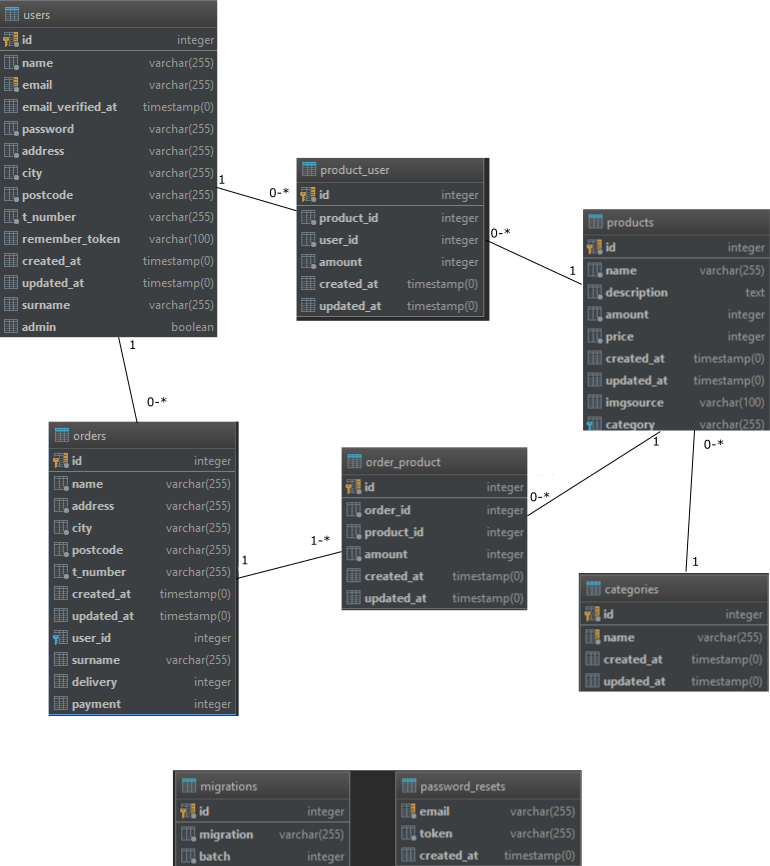
**Stručný opis**

V dátovom modeli sú len tri tabuľky users, products a orders. Vzťah medzi users a products reprezentuje to, že daný produkt je v košíku používateľa. Vzťah medzi users a orders reprezentuje, objednávku. A vzťah medzi orders a products, reprezentuje všetky produkty v danej objednávke. Pri vytvorení objednávky sa všetky vzťahy medzi daným userom a produktami presunú ako vzťahy medzi objednávkou a produktami.

**Logický model**

****

**Fyzický model**

****

*Poznámka: Tabuľka migrations je generovaná ORM Eloquent rámca Laravel, na prehľad migrácií tabuliek. Spojovacie tabuľky order\_product, product\_user sú tiež generované ORM, preto sa v nich nenachádzajú FK. Tabuľka password\_resets je tabuľka generovaná autentifikáciou generovanou Laravelom.*

**Významnejšie návrhové rozhodnutia**

**Košík**

Košík je realizovaný dvoma spôsobmi a to spôsobom, pre prihlásených užívateľov, ktorý je viac menej opísaný pri dátovom modeli. Pre neprihlásených používateľov sa košík ukladá to terajšej session-y, ktorá sa po prihlásení premaže.

**Obrázky ku produktom**

Obrázky ku produktom sú ukladané tak, že v tabuľke produktov sa nachádza stĺpec obsahujúci názov adresára, v ktorom sa nachádzajú obrázky k tomu to produktu (imgsource). Následne sa v tomto adresári nachádzajú obrázky vo formáte .jpg, ktoré sú pomenované rovnako vo všetkých takýchto adresároch. Týmto spôsobom je obmedzený počet obrázkov ku produktu. Retrospektívne by som toto návrhové riešenie možno riešil inač, či už cez media library, ktorá pridelí modelu média, alebo osobitnou tabuľkou pre fotky.

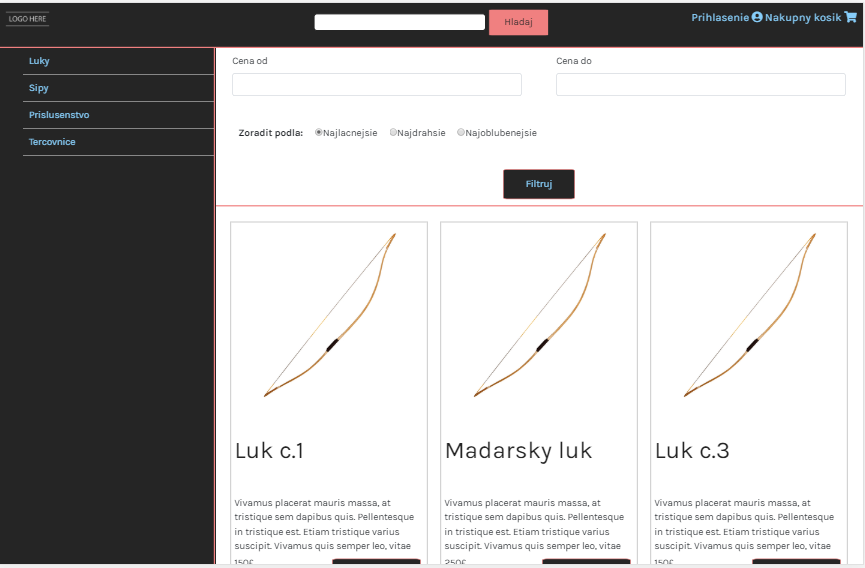
**Programovacie prostredie**

PHP/Laravel – PHPStorm, JavaScript,Vue.js/Quasar – Visual Studio 2017, PostreSQL – DataGrip, Apache server vamp

**Implementácia jednotlivých prípadov použitia**

**Zobrazenie produktov kategórie + filtrovanie**

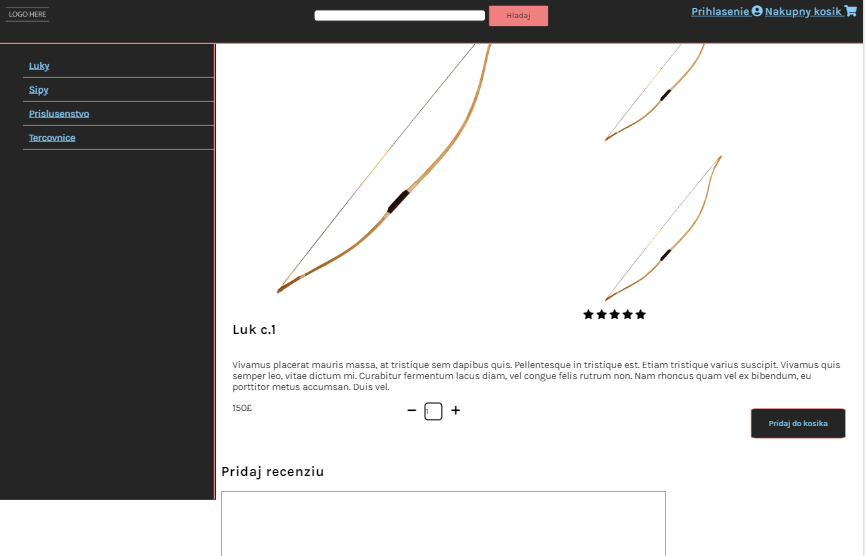
IndexController – show, category.blade.

Toto je docielené poslaním parametrov filtra a kategórie v url parametroch a následné postupné pridávanie do query („$products = $products->newQuery();“) podľa toho, či sa parameter v URL query nachádza.

**Detail produktu, pridanie do košíka**

ProductDetail.blade, ProductController – show, CheckoutController - addToCart.

Produkty sa do košíka po jednom dajú pridávať aj z thumbnailu, v detail sa to dá aj po viacerých kusoch. Množstvo v inpute sa mení pomocou js funkcie v časti scripts.blade.



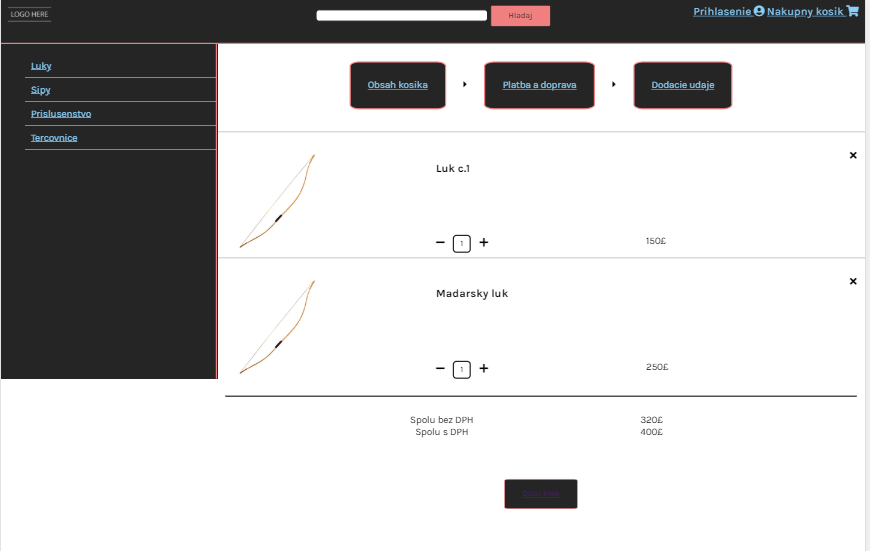
**Košík**

Košík sa skladá z troch okien. V prvom okne si môže používateľ prezrieť svoje produkty, zmeniť ich množstvo alebo ich úplne zrušiť. To je dosiahnuté volaním ajax funkcií.

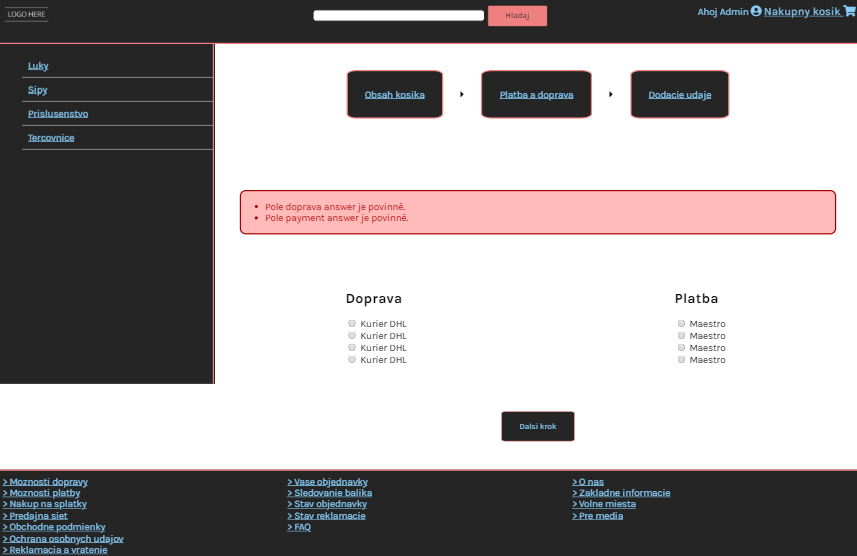
V druhom okne sa nachádza výber dopravy a platby, len jednoduchý formulár, ktorý túto voľbu dočasne uloží do session-y.

V treťom okne sa zadávajú dodacie údaje, ak je používateľ prihlásený vyplnia sa tieto údaje, údajmi z konta.

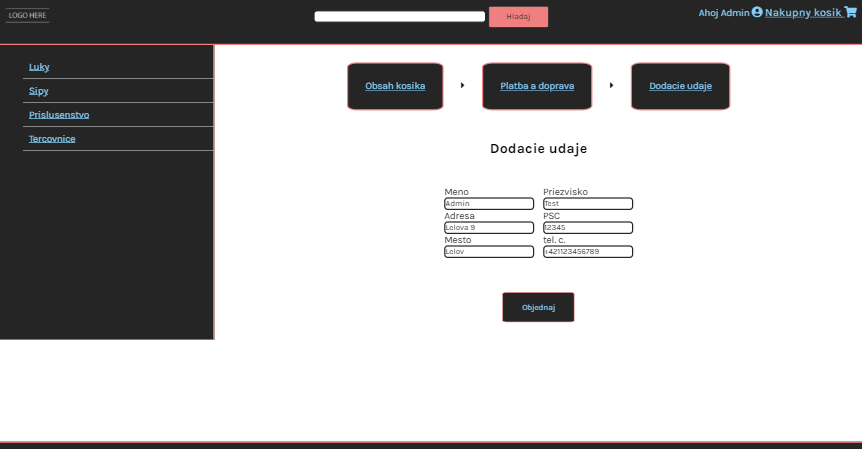
CheckoutController – index, updateCart, deleteCart; scripts.blade, checkout1.blade



CheckoutController – index2, index3 – na uloženie dopravy a platby



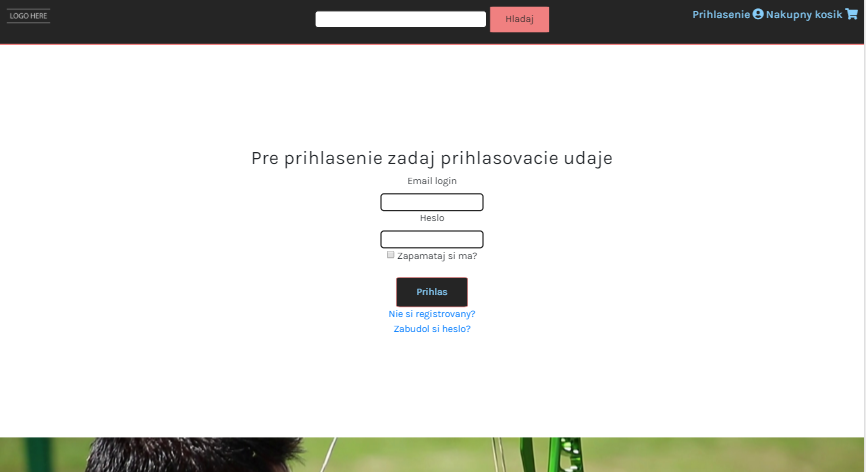
OrderController – store – vytvorenie objednávky prenesenie produktov z košíka na objednávku v DB, CheckoutController – index 3 – return view checkout3.blade



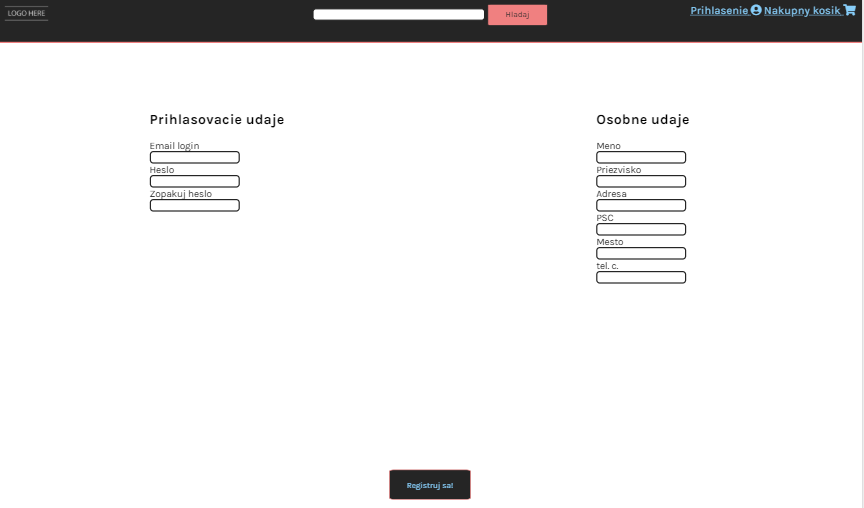
**Registrácia a prihlásenie/odhlásenie**

Registrácia a prihlásenie sú vytvorené pomcou Laravelu php artisan make:auth. Upravené sú len šablóny a pridané sú len niektoré polia do validácie a vytvorenia nového používateľa. Odhlásenie je možné v rolovacom menu.

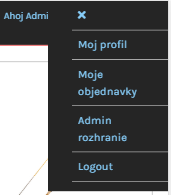
LoginController, login.blade



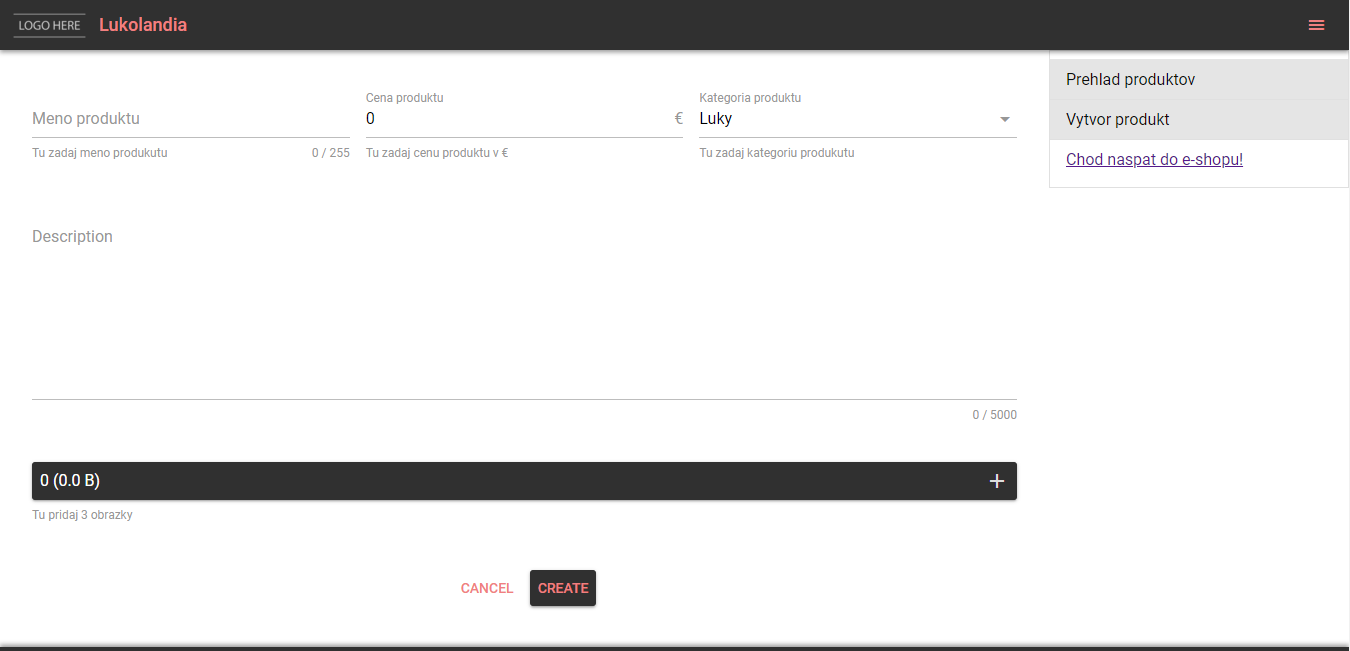
RegisterController, register.blade



Odhlásenie

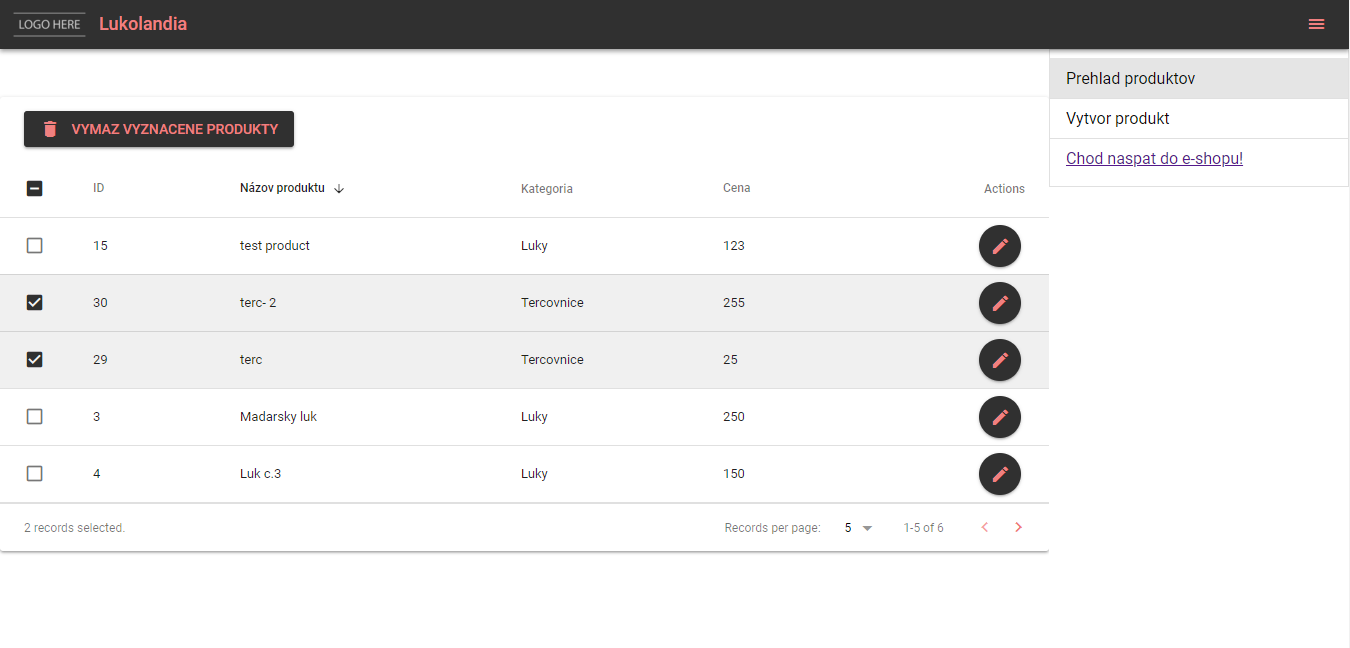


**Vytvorenie nového produktu**

Produkt sa tvorí pomocou axios requestu. Najprv sa pridávajú obrázky z qloadera, keď sa úspešne pridajú, tak sa tvorí produkt. Obrázky sa postupne pridávajú s predvybranými menami, ak už existujú, tak sa obrázok nepridá. 

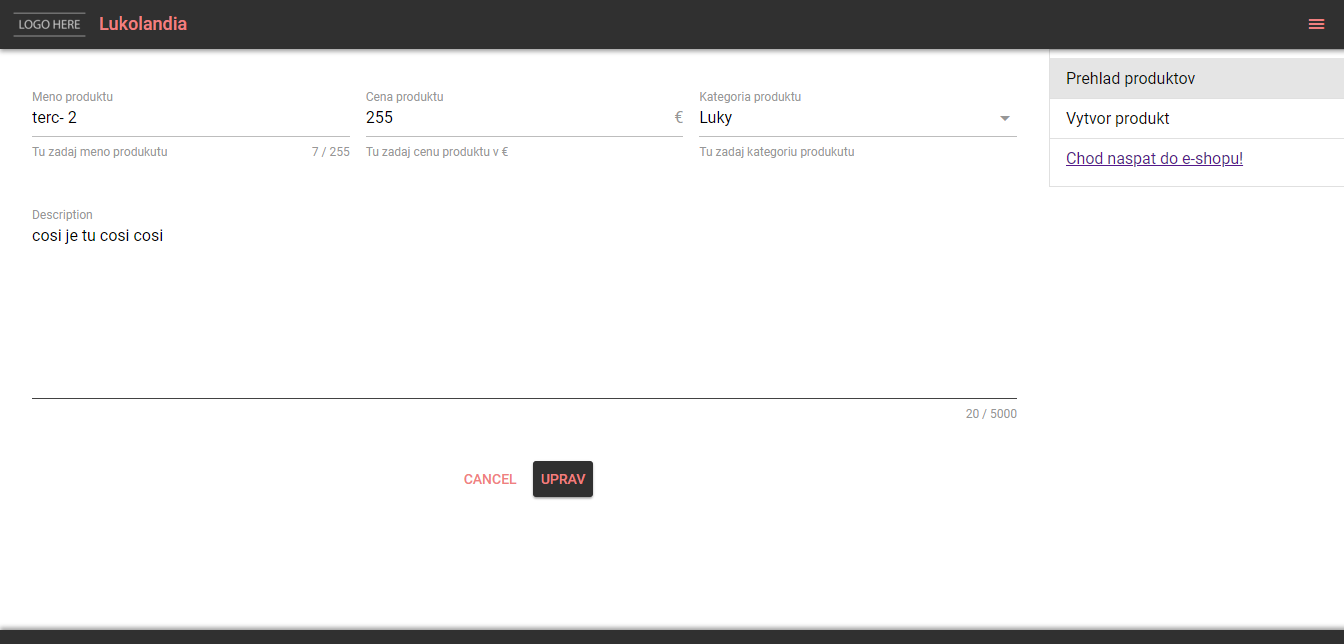
ProductController imageUpload, Store; Create.vue

**Zmazanie produktu a prehľad produktov**

Po označení produktov v prehľade sa zobrazí tlačidlo zmazať produkty. Po jeho stlačení sa odošle požiadavka na server s id produktov na zmazanie. Tam sa v produkty postupne mažú, najprv adresáre s obrázkami, potom produkty z databázy. Po zmazaní produktu, sa produkty znova načítajú z DB. 

ProductController – destroy; Overview.vue

**Upravenie produktu**

Pri zobrazení formulára sa najprv pomocou axios requestu načítajú jeho dáta, ktoré vyplnia formuláre. Potom po ich zmene sa zasa odošle axios request na server, kde sa daný produkt s daným id upraví. 

ProductController – update, Edit.vue