UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Spletna trgovina

Poročilo seminarske naloge pri predmetu
Elektronsko poslovanje

Študenti
Matevž Fabjančič (63150086)
Andraž Povše (63150224)

Mentor
David Jelenc

Ljubljana, 14. januar 2018

Kazalo

1 Uvod **2**

2 Navedba realiziranih storitev **3**

3 Podatkovni model **4**
3.1 Opis tabel 5
3.2 Opis sprožilcev 5
3.3 Denormalizacije 6

4 Varnost sistema **7**
4.1 Preprečevanje injekcije kode SQL 7
4.2 Shranjevanje gesel 7
4.3 Preverjanje uporabnikovih vnosov 7
4.4 Prijava in odjava 8
4.5 Omejevanje dostopa 8
4.6 Preklapljanje na zavarovan kanal 9
4.7 Registracija strank z uporabo filtriranja reCAPTCHA 9
4.8 Prijava in odjava za mobilno aplikacijo 9

5 Izjava o avtorstvu seminarske naloge **11**

Poglavje 1

Uvod

Seminarsko nalogo sva naredila po arhitekturnem vzorcu MVC. Uporabila sva podatkovno bazo MySQL, nekatera vnosna polja sva realizirala s pomočjo Pear QuickForms2, pri izdelavi čelnega dela pa sva uporabila knjižnico Bootstrap. Slike izdelkov sva shranjevala na datotečni sistem.

Poglavje 2

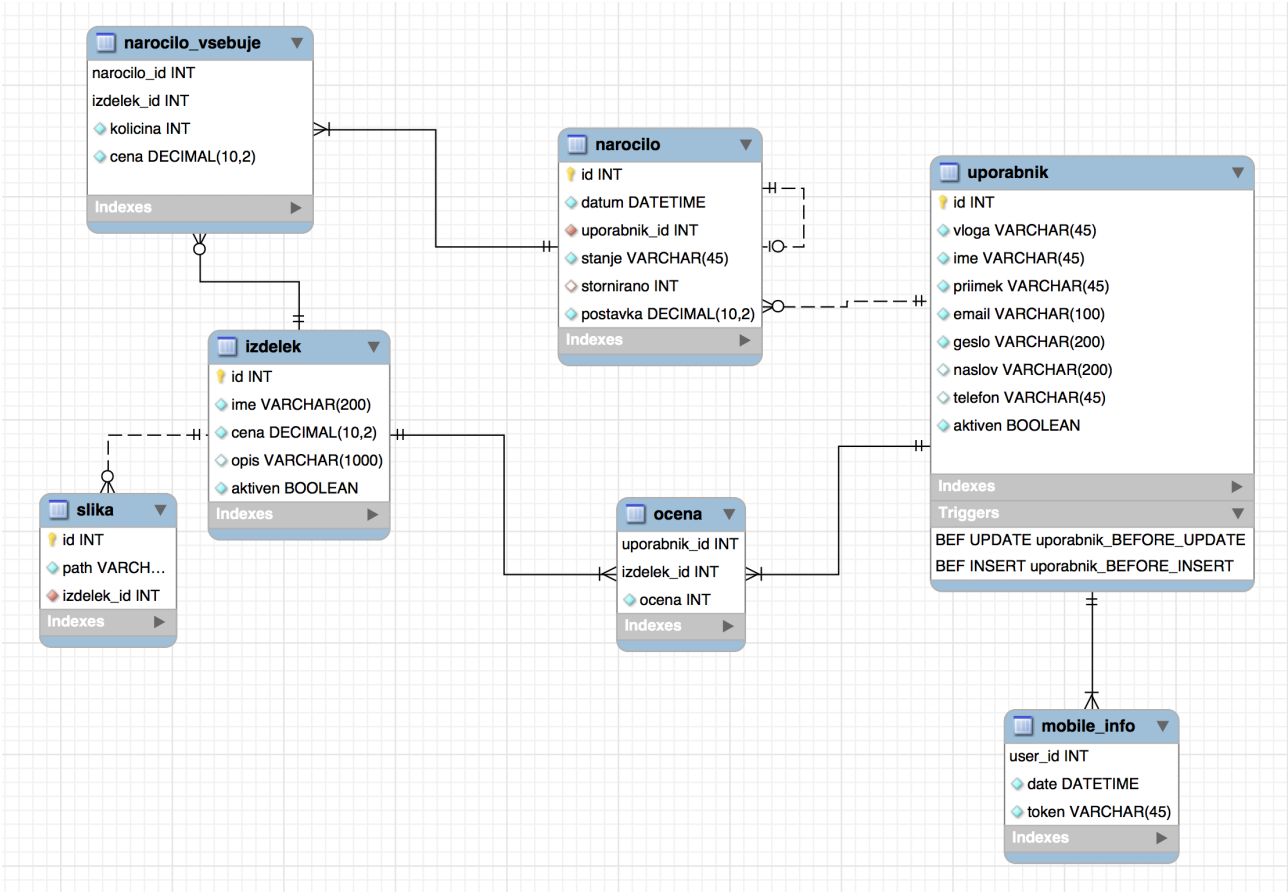
Navedba realiziranih storitev

Poleg obveznih storitev (ki so implementirane v celoti) so v seminarski nalogi implementirane tudi naslednje razširitve.

- Varnost
  - Vodenje dnevnika uporabnikov Administrator in Prodajalec
  - Registracija strank z uporabo filtriranja CAPTCHA
- Uporabniški vmesnik
  - Smiselna organizacija in izvedba uporabniškega vmesnika s pomočjo tehnologij kot so sta CSS in JavaScript (brez tehnologije AJAX)
  - Predstavitev artiklov s slikami
  - Ocenjevanje izdelkov in prikaz povprečne ocene
  - Iskanje po artiklih
- Napredne funkcije mobilne aplikacije
  - Prijava in odjava (opisano v 4.8)

Poglavje 3

Podatkovni model



Slika 3.1: Logični model podatkovne baze

3.1 Opis tabel

Tabele, ki se nahajajo v logičnem modelu 3.1, so opisane spodaj.

- Uporabnik. Tabela vsebuje stranke, prodajalce in administratorje. Stranke in zaposleni se razlikujejo po vlogi in atributih. O zaposlenih ne hranimo podatkov o naslovu in telefonu.
- Naročilo. Tabela vsebuje vsa naročila. Atribut postavka vsebuje celotno ceno naročila. Če naročilo storniramo se atribut štornirano"nastavi na id novega naročila, ki vsebuje negativno celotno ceno naročila. Stanje tega novega naročila z negativno ceno je "negativna-stornirano".
- Izdelek. Tabela vsebuje izdelke z imenom, ceno in opisom.
- Slika. Tabela vsebuje tuji ključ izdelka in ime slike v datotečnem sistemu.
- Ocena. Tabela vsebuje tuji ključ od uporabnika in izdelka, ter oceno, ki jo je uporabnik podal izdelku.
- Naročilo_vsebuje. Tabela vsebuje tuji ključ do izdelka, naročila, trenutno ceno izdelka in količino naročenega izdelka.
- Mobile_info. Tabela vsebuje tuji primarni ključ user_id, datum in token. Token se uporablja za avtentikacijo pri mobilni aplikaciji.

3.2 Opis sprožilcev

Sprožilce sva nastavila pri tabeli 'uporabnik', kjer se izvedejo pred vnosom in posodobitvijo vrstice. Z njimi sva omejila vrednosti atributa 'vloga', omejila število administratorjev na enega in preverjala izpolnjenost atributov naslov in geslo pri novih uporabnikih z vlogo 'stranka'.
Sprožilec pred vnosom v tabelo:

```
CREATE DEFINER = CURRENT_USER TRIGGER
 ↪  `ep_trgovina`.`uporabnik_BEFORE_INSERT` BEFORE INSERT ON `uporabnik`
 ↪  FOR EACH ROW
BEGIN
        IF NEW.vloga = 'administrator' AND EXISTS (SELECT * FROM
         ↪  uporabnik WHERE vloga = 'administrator') THEN
                SIGNAL SQLSTATE '45000' SET message_text = 'Obstaja lahko
                 ↪  le en administrator';
        END IF;
        IF NEW.vloga NOT IN ('stranka', 'administrator', 'prodajalec')
         ↪  THEN
        SIGNAL SQLSTATE '45000' SET message_text = 'Nedovoljena vloga';
```

```
    END IF;
        IF NEW.vloga IN ('stranka') AND (NEW.naslov IS NULL OR
         ↪  NEW.telefon IS NULL) THEN
                SIGNAL SQLSTATE '45000' SET message_text = 'Za uporabnika
                 ↪  `naslov` ali `telefon` ni nastavljeno';
        END IF;
END;
```

3.3 Denormalizacije

Denormalizacijo sva uvedla pri atributu naslov. Shranila sva ga kot en atribut (namesto ločitve na ulico, kraj-pošta), ker se bo uporabljal samo pri dejanski dostavi izdelkov.

Poglavje 4

Varnost sistema

4.1 Preprečevanje injekcije kode SQL

Z uporabo izključno pripravljenih poizvedb (angl. Prepared statements) in preverjanja uporabniških vnosov, kot je opisano v odseku 4.3, se nevarnosti injekcije kode SQL odpravijo.

4.2 Shranjevanje gesel

Geslo, ki po zavarovanem kanalu ob registraciji pride v strežniško okolje, se z uporabo vgrajene funkcije password_hash() pretvori v zgoščeno vrednost. Funkcija se izvede s privzetimi parametri; uporabi se zgoščevalni algoritem bcrypt, kot sol pa se uporabi naključno proizvedena vrednost. Namen uporabe naključne soli je ta, da so v primeru enakih gesel uporabnikov zgoščene vrednosti različne. Cena (oz. računska zahtevnost zgoščevanja) je prav tako ostala na privzeti vrednosti 10. Ta parameter napadalcu, ki bi poznal zgoščene vrednosti gesel, dodatno oteži iskanje dejanskega gesla.

Zgoščena vrednost gesla se nato zapiše v podatkovno bazo. Ob prijavi se geslo, ki ga poda uporabnik, primerja z zgoščeno vrednostjo v podatkovni bazi. Pri tem se uporabi vgrajena funkcija password_verify().

4.3 Preverjanje uporabnikovih vnosov

Z vnosi uporabnikov vedno ravnajo razredi tipa Controller. V vsaki funkciji, ki ima opravka s potencialno zlonamernim vnosom uporabnika, se ta preveri. Za preverjanje se uporabi vgrajena funkcija filter_input_array(). Pravila so podana v ločeni tabeli (v večini primerov imenovani $rules).

4.4 Prijava in odjava

Kontrola prijave in odjave je centralizirana v razredu UporabnikiController. Ko je prijava uspešna (po postopku opisanem v odseku 4.2, se ponovno nastavi identifikator seje, v sejo pa se zapiše tudi ID uporabnika in njegova vloga.

Prijava je mogoča na dveh naslovih:

- /prijava

  Če se uporabnik prijavi na tem naslovu, se v sejo vedno zapiše vloga *stranka*. Takrat se tudi prodajalci in administrator postavijo v vlogo stranke.

- /x509login

  Na tem naslovu se od uporabnika zahteva overitev z digitalnim potrdilom. Ob prijavi se v sejo shrani dejanska vloga uporabnika. Ko se nekdo overi z digitalnim potrdilom, ki vsebuje lastnikov elektronski naslov, se uporabnik prijavi le v tisti račun, ki uporablja enak elektronski naslov.

4.5 Omejevanje dostopa

V razredih, izpeljanih iz razreda AbstractController, se pred izvedbami funkcij za urejanje stanja trgovine po potrebi preveri vloga uporabnika. To preverjanje je implementirano v funkciji preveriVlogo().

```php
<?php
static $VLOGE = [
    'stranka' => 3,
    'prodajalec' => 2,
    'administrator' => 0
];

/**
 * V primeru da je podana vloga nadrejena prijavljeni vlogi
 * preusmeri na /
 * @param string $vloga vloga
 * @return NULL
 */
protected static function preveriVlogo($vloga) {
    if (isset($_SESSION['user_vloga']) &&
     ↪  self::$VLOGE[$_SESSION['user_vloga']] <=
     ↪  self::$VLOGE[$vloga]) {
        return;
    } else {
        echo ViewHelper::alert('Prepovedan dostop', '/');
        exit();
```

```
        }
    }
```

4.6 Preklapljanje na zavarovan kanal

Spletna trgovina je dostopna po nezavarovanem in zavarovanem kanalu. Na nezavarovanem kanalu je mogoče pregledovati izdelke, jih dodajati v košarico (stanje košarice ni persistentno). Ko uporabnik obišče mesto /prijava ,/registracija ali /x509login, strežnik Apache povezavo preusmeri na zavarovan kanal.

```
<LocationMatch
    "/netbeans/ep-trgovina/php/index.php/(x509login|registracija|prijava)">
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule (.*) "https://%{HTTP_HOST}%{REQUEST_URI}"
</LocationMatch>
```

4.7 Registracija strank z uporabo filtriranja reCAPTCHA

Ker pri registraciji nismo ravno strogo preverjali ustreznosti e-poštnega naslova in ostalih osebnih podatkov, je bilo potrebno za preprečevanje robotskih (angl. botted) vnosov uvedeti drug način filtriranja.

To sva naredila s pomočjo Googlovega sistema reCAPTCHA, ki sumljivemu uporabniku poda logično uganko, ki je po navadi izbor slik ki prikazujejo določen predmet na matriki devetih slik. Pri pridobivanju reCAPTCHE sva pridobila javni in privatni ključ. Javni ključ se uporabi pri generiranju reCAPTCHE, medtem ko privatni ključ uporabimo za pošiljanje POST zahtevka. POST zahtevek se pošlje na Googlov URL, v njem pa podamo svoj privatni ključ in povratno kodo reCAPTCHE, ki se je generirala, ko je oseba rešila oziroma ni rešila reCAPTCHO (v našem primeru stranka, ki se poizkuša registrirati). Odgovor prejmemo v obliki JSON, v katerem obstaja atribut 'success', ki nam pove ali je bila reCAPTCHA rešena pravilno ali ne. Na podlagi tega lahko nadaljujemo z registracijo stranke, ali pa jo preusmerimo nazaj na spletno stran za registracijo in obvestimo, da je napačno rešila reCAPTCHO.

4.8 Prijava in odjava za mobilno aplikacijo

Ker naj storitve REST ne bi uporabljale stanja povezave, je bilo treba poiskati in implementirati ustrezno zamenjavo za sejo HTTP.

Ko uporabnik v mobilni aplikaciji vpiše podatke za prijavo (t.j. elektronski naslov in geslo), se ti preko varnega kanala HTTPS pošljejo na strežnik, na končno točko /api/prijava. Strežnik podatke preveri in v primeru pravilnih podatkov proizvede

naključen žeton z uporabo vgrajene funkcije random_bytes(). Ta žeton ohrani v pomnilniku do odgovora odjemalcu. Pred tem zgoščeno vrednost niza skupaj s podatki o lastniku žetona zapiše v podatkovno bazo. Žeton (v prvotni obliki) pošlje odjemalcu, ki si ga zapomni. Odjemalca se od takrat naprej več ne overja z uporabniškim imenom in geslom, temveč z žetonom, ki ga mora odjemalec posredovati ob vsakem zahtevku, ki za uspešno obdelavo potrebuje avtentikacijo. Žeton se pošilja v zaglavju zahtevka HTTP Authorization.

Poglavje 5

Izjava o avtorstvu seminarske naloge

Spodaj podpisani *Matevž Fabjančič*, vpisna številka *63150086*, sem soavtor seminarske naloge z naslovom *Spletna trgovina*. S svojim podpisom zagotavljam, da sem izdelal ali bil soudeležen pri izdelavi naslednjih sklopov seminarske naloge:

- Vloge uporabnikov
- Osnovne storitve
- Spletni vmesnik anonimnega odjemalca
- Spletni vmesnik vloge Stranka
- Ostale zahteve
- Mobilna aplikacija
- Razširjene storitve - Napredne funkcije mobilne aplikacije
- Razširjene storitve - Uporabniški vmesnik

Podpis: Matevž Fabjančič, l.r.

Spodaj podpisani *Andraž Povše*, vpisna številka *63150224*, sem soavtor seminarske naloge z naslovom *Spletna trgovina*. S svojim podpisom zagotavljam, da sem izdelal ali bil soudeležen pri izdelavi naslednjih sklopov seminarske naloge:

- Vloge uporabnikov
- Osnovne storitve
- Spletni vmesnik anonimnega odjemalca
- Spletni vmesnik vloge Prodajalec
- Spletni vmesnik vloge Stranka
- Ostale zahteve
- Razširjene storitve - Varnost
- Razširjene storitve - Uporabniški vmesnik

Podpis: Andraž Povše, l.r.