

# Projektni zahtev

Aplikacija e-trgovine za prodaju tepiha

Masa Gajevic 2016200213

## Sadržaj

1. Uvod.....	3
1.1. Cilj razvoja.....	3
1.2. Obim sistema.....	3
1.3. Prikaz proizvoda.....	3
1.3.1. Perspektiva proizvoda .....	4
1.3.2. Funkcije proizvoda.....	4
1.3.3. Karakteristike korisnika .....	5
1.3.4. Ograničenja .....	5
1.4. Definicije .....	6
2. Reference.....	7
3. Specifikacija zahteva .....	7
3.1. Spoljašnji interfejs.....	8
3.2. Funkcije .....	11
3.3. Pogodnosti za upotrebu .....	12
3.4. Zahtevane performanse.....	13
3.5. Zahtevi baze podataka .....	13
3.6. Projektna ograničenja .....	18
3.7. Sistemske karakteristike softvera.....	18
3.8. Dopunske informacije .....	18

## 1. Uvod

Aplikacija za internet prodaju tepiha stvorena je sa ciljem da administratorima tj zaposlenima omoguci jednostavniji i brzi nacin za unosenje novih artikala (tepiha) u bazu podataka, kao i da omoguci potencijalnim kupcima laksu internet kupovinu. Sistem ce biti na raspolaganju za koriscenje svim zaposlenima putem panel portala, gde je njihov zadatak da u bazu unesu sve artikle koji su na stanju. Da bi sto verodostojnije potencijalnim kupcima priblizili artikle uz naziv i fotografiju artikla neophodno je da unesu detaljan opisa, spisak velicina u kojima je tepih dostupan i cene za svaku od velicina. Na taj nacin olaksana je pretraga i kupovina. Celokupna komunikacija sa serverom baze podataka kao i sa klijentima vrsice se putem interneta.

### 1.1. Cilj razvoja

Postojanje ove aplikacije ima za cilj da kroz efikasniji i fleksibilniji rad zaposlenih utice na njihovu produktivnost i doprinosi boljem poslovanju. S obzirom da se bazi podataka i aplikaciji pristupa putem web pregledaca, zaposlenima je ovo dostupno u svakom trenutku kada imaju internet konekciju, sa izuzetkom casova u kojima se vrsi backup servera ili se radi na resavanju problema i poboljsanju portala. Jos jedna pogodnost postojanja panel portala jeste ta sto zaposlenima omogucava i rad od kuce.

### 1.2. Obim sistema

Aplikacija ce biti hostovana na veb serveru i pristup je moguc putem web pregledaca. Zauzimaće samo deo resursa servera. Neophodan je i server baze podataka na kome ce biti smeštena baza podataka, koja se koristi u radu portala. Baza ce imati na raspolaganju celokupne resurse servera. Sistem ne može biti kompletan bez konekcije sa Internetom. Korisnik pristupa portalu putem Internet pregledača, a i komunikacija izmedju web servera i servera baze podataka takodje se vrši uz pomoć Interneta.

### 1.3. Prikaz proizvoda

Funkcionalni zahtevi aplikacije su:

- prijava korisnika, koja omogucava autentifikaciju korisnika
- pregled spiska tepiha, dodavanje novog tepiha i izmena podataka postojećeg, što korisniku omogucava rad sa podacima o artiklima
- kreiranje nove kategorije artikala i pregled postojećih , što omogucava korisniku brži i automatizovani rad pri kreiranju i pretrazi
- prikaz materijala odredjenog artikla, kao i dodavanje novih u bazu podataka
- prikaz artikala po najnižoj ceni ili velicini sto omogucava korisnicima jednostavniju pretragu po datom kriterijumu
- prikaz i pregled korpe sto omogucava korisnicima da izmene svoje porudzbine (dodaju ili obrisu artikle), kao i da potvrde kupovinu
- nakon potvrde kupovine vrsi se prikupljanje podataka o korisniku (ime, prezime, poštanska adresa i adresa elektronske pošte)
- prikaz uplatnice sa pozivom na broj za datog kupca i njegovu porudzbinu

Nefunkcionalni zahtevi:

- Aplikacija treba da bude postavljena na veb server koji će omogućiti brzu reakciju na korisničke akcije
- U okviru baze podataka biće implementirana ograničenja i procedure koje će vršiti proveru prilikom unosa i izmena podataka
- Iznenadni prestanak rada sistema može uzrokovati samo nestanak Interneta ili problem sa funkcionisanjem servera.
- Korisnik ne može uticati na rad sistema. Svi podaci koje unosi u bazu podataka će biti proveravani na više načina: (1) HTML [9] pattern u poljima za unos podataka u formularima; (2) JavaScript [9] validacija vrednosti unetih u polja za unos podataka u formularima na korisničkoj strani; (3) Provera korišćenjem adekvatnih testova ili korišćenjem regularnih izraza na strani servera u Node.js [9] aplikaciji (moguće je i korišćenjem izričitih šema - Schema) i (4) provera na nivou baze podataka korišćenjem okidača nad samim tabelama baze podataka.

Ovom aplikacijom se postize efikasniji i fleksibilniji rad zaposlenih i njihova produktivnost , doprinosi boljem poslovanju.Podaci su zaposlenima dostupni kad god imaju konekciju na Internet.

#### 1.3.1. Perspektiva proizvođača

Sistem će biti pristupačan uvek i svuda gde postoji konekcija sa Internetom. Ono što odlikuje ovaj sistem jeste da omogućava potencijalnim kupcima da brzo i lako pretraže i pronadju artikla koji ih interesuju, da ih poruče i dobiju isporuku na kućnu adresu bez odlaska u prodavnicu. Tokom pretrage portala na raspolaganju su im iscrpni opis i karakteristike artikala (materijal, dimenzije, fotografije tepiha) i u svakom trenutku mogu pregledati, izmeniti ili ponistiti svoju narudžbinu.

#### 1.3.2. Funkcije proizvođača

Na dijagramu slučajeva korišćenja prikazane su funkcije sistema namenjene krajnjem korisniku. Postoje dva tipa krajnjih korisnika. Prvi tip je administrator, odnosno korisnik koji je prijavljen na sistem sa ispravnim parametrima (korisničko ime i lozinka) i ima pristup administrativnom panelu. Drugi tip je neprijavljeni korisnik tj kupac koji može da pristupi panelu u smislu da može da vrsi pretragu artikala i uz prijavu da ih doda u korpu i izvrši internet kupovinu.

Administrator u slučaju korišćenja pregleda artikala ima uvid u sve artikle određenog tipa odakle može pregledati pojedinačni artikal, kategorije i materijale artikala.

Može da vrsi izmenu postojećih i dodavanje novih artikala u bazu podataka koji će biti vidljivi na panelu koji je dostupan potencijalnim kupcima.

Prijava predstavlja neophodan slučaj korišćenja panela, jer bez prijave administratoru neće biti omogućen uvid u stanje baze niti mogućnost izmena postojeće. Samim tim onemogućava se uvid u stanje rada internet prodavnice.

### 1.3.3. Karakteristike korisnika

Korisnik aplikacije treba da poseduje osnovnu informatičku pismenost. Mora da ima pristup računaru i Internetu. Potrebno je da ima iskustva u radu sa Internet pregledačem i osnovnim funkcionalnostima računara. Očekuje se da ima obrazovanje i iskustvo neophodno za razumevanje principa rada osiguravajuće kuće.

### 1.3.4. Ograničenja

Aplikacija mora da bude realizovana na Node.js platformi korišćenjem Nest.js [9] razvojnog okvira [9] i sav kôd aplikacije treba da bude organizovan prema pravilima MVC arhitekture [9]. Baza podataka mora da bude relacionala [9] i treba koristiti MySQL/MariaDB [9] RDBMS. Sav generisani HTML kôd koji proizvodi aplikacija mora da bude 100% validan, tj. da generisani kôd prođe proveru W3C Validatorom (dopuštena su upozorenja, ali ne i greške). Aplikacija može grafički korisnički interfejs da generiše na strani servera, korišćenjem šablona za generisanje HTML koda (proizvoljan templating engine, integrisan sa Nest.js aplikacijom) ili da bude serviran statički deo stranice koji pomoću JavaScript-a dinamički formira komponente na front-end-u, a podatke doprema asinhrono kroz veb servis (API) [9] metode obezbeđene u okviru same aplikacije.

Potrebno je obezbediti određeni stepen provere podataka koji se od korisnika upućuju aplikaciji. Moguća su četiri sloja zaštite i to: (1) HTML pattern u poljima za unos podataka u formularima; (2) JavaScript validacija vrednosti unetih u polja za unos podataka u formularima na front-end-u; (3) Provera korišćenjem adekvatnih testova ili korišćenjem regularnih izraza na strani servera u Node.js aplikaciji (moguće je i korišćenjem izričitih šema - Schema) i (4) provera na nivou baze podataka korišćenjem okidača nad samim tabelama baze podataka.

Neophodno je napisati prateću projektnu dokumentaciju o izradi aplikacije koja sadrži (1) model baze podataka sa detaljnim opisom svih tabela, njihovih polja i relacija; (2) dijagram organizacije elemenata aplikacije sa akcentom na isticanje MVC arhitekture na konkretnom primeru jednog odabranog zahteva, tj. rute/putanje koju Vaša aplikacija obrađuje; (3) popis svih aktivnosti koje su podržane kroz aplikaciju za sve uloge korisnika aplikacije prikazane u obliku Use-Case dijagrama; (4) popis svih kontrolera i njihovih metoda koji obavljaju potrebnu programsku i poslovnu logiku sa ciljem izvršavanja svih predviđenih aktivnosti aplikacije, kao i (5) sve ostale elemente dokumentacije predviđene uputstvom za izradu dokumentacije objavljenom na stranici predmeta u sekciji sa predavanja.

Izrada projekta mora da bude sprovedena korišćenjem alata za verziranje koda Git [13], a kompletan kôd aplikacije bude dostupan na javnom Git repozitorijumu, npr. na besplatnim GitHub ili Bitbucket servisima. Ne može ceo projekat da bude otpremljen u samo nekoliko masovnih Git commit-a, već mora da bude pokazano da je projekat realizovan u kontinuitetu, da su korišćene grane (branching), da je bilo paralelnog rada u više grana koje su spojene (merging) sa ili bez konflikata (conflict resolution).

#### 1.4. Definicije

1. Veb pregledač - (engl. web browser) poseban koji korisniku omogućava pregledanje veb-stranica i multimedijalnih sadržaja. Služi za pronalaženje, pregledanje, preuzimanje i predstavljanje podataka na svetskoj mreži.
2. Grafički korisnički interfejs - (GUI – Graphical User Interface) koristi vizuelne elemente tipa prozora, menija, dugmića, ikona i sl. Kao primarni ulazni uređaji za zadavanje komandi koriste se miš i tastatura. Sa napretkom hardvera, napreduju i korisnički interfejsi koji postaju sve lepše dizajnirani.
3. Hostovanje – određene kompanije koje pružaju usluge koje omogućavaju da vasa prezentacija bude vidljiva na Internetu.
4. Veb server - je računar na kojem se nalaze web stranice. Kako je pristup svakom računaru određen portovima (ulazima), tako je i pristup web serveru određen portom 80. Primarna uloga servera je da se na njemu hostuju stranice i fajlovi, i tako postanu dostupni na vebu.
5. Server baze podataka - server koji koristi aplikaciju baze podataka koja pruža usluge baze podataka drugim računarskim programima ili računarima, kao što je definisano modelom klijent-server.
6. Internet - globalni informacioni sistem, logički povezan jedinstvenim sistemom adresiranja putem internet protokola. Obezbeđuje, koristi ili omogućava servise visokog nivoa za ličnu i poslovnu primenu.
7. Backup – označava postupak pravljenja kopija podataka kako bi se u slučaju njihovog nepredviđenog gubitka, takve kopije iskoristile za povratak na početno stanje.
8. Java Script – programskom jeziku koji omogućava pravljenje interfejsa koji omogućavaju korisnicima visoku funkcionalnost bez potrebe za novim učitavanjem stranice. To je orijentisan skriptovan jezik visokog nivoa, dominantno se koristi pri izradi web stranica uz HTML i CSS.
9. Skriptni jezik- programski jezik čiji se kod najčešće izvršava interpretiranjem. Pored toga, skriptni jezici se koriste najčešće za pisanje manjih programa (skripti) koji se brzo pišu i služe za obavljanje manjih poslova.
10. MVC - (eng. Model - View - Controller) to je arhitekturni dizajn patern koji opisuje jedan specifičan način građenja aplikacija/programa/sistema. Podrazumeva postojanje tri vrste komponenti (Model, View, Controller) koje se međusobno nalaze u posebno definisanom odnosu.
11. Nest.js – JavaScript razvojni okvir.
12. Node.js - višeplatformsko JavaSkript radno okruženje otvorenog koda za izvršavanje JavaSkript-a na serverskoj strani.
13. API - (eng. Application Programing Interfaces) interfejs za programiranje aplikacija.
14. Relaciona baza podataka – baze podataka koje organizuju podatke u tabele koje mogu biti povezane zajedničkim podacima. [9]
15. RDBS - (eng. Relational Database Management System) sistem za upravljanje relacionim bazama podataka.
16. SQL – (eng. Structured Query Language) programski jezik namenjen za upravljanje podacima u relacionim sistemima za upravljanje bazama podataka. Obuhvata unos podataka, upite, ažuriranje i brisanje, šeme kreiranja i menjanja, kao i podatke za kontrolu pristupa.
17. HTML - (eng. Hyper Text Markup Language) opisni jezik specijalno namenjen opisu veb stranica. Pomoću njega se jednostavno mogu odvojiti elementi kao što su naslovi, paragrafi, citati i slično.

18. Dijagram slučajeve korišćenja – predstavlja grafički interakcije korisnika sa sistemom koji pokazuje odnos između korisnika i različitih slučajeva korišćenja u kojima je korisnik uključen.
19. Informatička pismenost - obuhvata poznavanje računara ( hardvera i softvera ) kao i korišćenje računara i računarskih programa u svakodnevnom životu.
20. Git – sistem kontrole verzija za praćenje promena u izvornom kodu tokom razvijanja softvera. Primarno se koristi za verzionisanje koda pri razvoju softvera,[ ali se može koristiti za praćenje promena u bilo kojim fajlovima.
21. Izvorni kod - (engl. Source code) je kolekcija kompjuterskih instrukcija napisana na nekom ljudski-čitljivom programskom jeziku. Izvorni kod programa je specijalno dizajniran za rad programera koji precizira akcije koje će obavljati sa kompjutera uglavnom pisanjem kodova.
22. HTTP protokol- predstavlja protokol namenjen prenosu informacija na webu.

## 2. Reference

- [1] <https://www.oxfordwebstudio.com/da-li-znate/sta-je-web-browser.html>
- [2] <http://www.vebnstudy.com/tema.php?id=korisnicki-interfejs>
- [3] <https://infogate.rs/sta-je-hosting/>
- [4] <https://skolakoda.org/server>
- [5] [https://en.wikipedia.org/wiki/Database\\_server](https://en.wikipedia.org/wiki/Database_server)
- [6] <https://sh.wikipedia.org/wiki/Internet>
- [7] <https://www.netapp.com/us/info/what-is-backup-and-recovery.aspx>
- [8] <https://www.youtube.com/channel/UCoabUK8jyHYwuxY3DFIbWcQ>
- [9] Tomašević, Violeta, *Razvoj aplikativnog softvera*, Univerzitet Singidunum, Beograd, 2017.
- [10] <https://epale.ec.europa.eu/hr/blog/informaciona-i-informaticka-pismenost>
- [11] [https://sr.wikipedia.org/sr-el/Гит\\_\(софтвер\)](https://sr.wikipedia.org/sr-el/Гит_(софтвер))
- [12] [https://sr.wikipedia.org/sr-el/ИЗВОРНИ\\_КОД#cite\\_note-ctutorial-1](https://sr.wikipedia.org/sr-el/ИЗВОРНИ_КОД#cite_note-ctutorial-1)
- [13] <https://skolakoda.org/http>

## 3. Specifikacija zahteva

Bazi podataka ce se pristupati kroz realizaciju razlicitih prostih i slozenih upita.

Komunikacija se uspostavlja uz pomoc HTTP protokola. Neophodno je da administrator i korisnici poseduju web pregledac koji podrzava HTTP protokol verzije 1.0 ili 1.1

Funkcionalnosti koje su na raspolaganju zaposlenima treba da budu implementirane tako da im omoguće što jednostavniji unos podataka, njihovu obradu i pretragu po razlicitim kriterijumima.

Dijagramima slučajeva korišćenja ( slika 7, slika 8 i slika 9 ) biće prikazani slučajevi korišćenja za sve učesnike u sistemu.

Dijagramom klasa ( slika 11) biće prikazan model podataka.



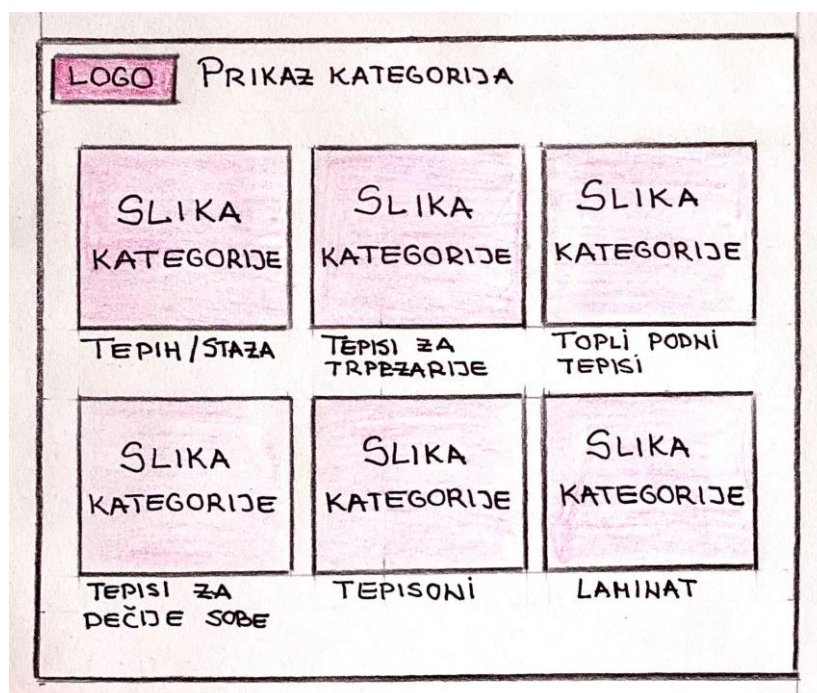
### 3.1. Spoljašnji interfejs

Nakon pokretanja aplikacije korisniku se prikazuje stranica za prijavu ( slika 1). Ova opcija daje mogućnost administratoru/korisniku da uz unosenje ispravnih parametara pristupi panelu portala.



slika 1

Ukoliko korisnik unese pogrešne parametre obaveštava se da je došlo do unosa nevalidnih parametara. Ukoliko su uneti ispravni parametri administratoru/korisniku se otvara prozor koju nudi pregled svih kategorija artikala. Kategorije su prikazane u vidu fotografije i naziva artikla. (slika 2)



slika 2



Na ovoj stranici samo administrator ima mogućnost da doda ili izmeni postojeće kategorije tepiha, tako što će uz fotografiju artikla dodati i odgovarajuće parametre kao što su: naziv, opis, velicina, cena ... Klikom na dugme kreiraj, vrši se promena na portalu. A klikom na dugme odustani ponistava se akcija. (slika 3).

slika 3

Kategorije artikala su prikazane u vidu menija na portalu. Korisnik može da bira po kom parametru će pretraživati artikle, npr može izabrati po ceni, dimenziji, materijalu ... Izbor vrši tako što čeka određena polja ispred kategorije za pretragu. Web stranica automatski filtrira i vrši prikaz određenih artikala. (slika 4).

slika 4

Klikom na ikonicu Korpa u gornjem desnom uglu, korisnik ima uvid u listu izabranih artikala. Na ovoj stranici ostavlja mu se mogućnost da nakon pregleda liste izabranih artikala koja sadrži naziv artikla, sliku, cenu, količinu nastavi kupovinu, ponisti kupovinu ili naruci izabrane proizvode. To čini klikom na jedan od tri ponudjena dugmeta. (slika 5)

PROIZVOD	SLIKA	CENA	VELIČINA	KOLIČINA
1. ALHIRA 276 LILA		1882,00	80X150	1
2. ASSOS 16917		3235,00	80X150	1
UKUPNO ZA PLAĆANJE			5.117,00 RSD	

NASTAVI KUPOVINU      NARUČI

slika 5

Ukoliko korisnik potvrdi kupovinu klikom na dugme, uzimaju se lični podaci o korisniku (ime, prezime, poštanska adresa i adresa elektronske pošte) a nakon toga sistem generise jedinstvenu uplatnicu za datog korisnika sa pozivom na broj koji je identifikacioni broj porudžbine, sa prikazom tekućeg računa i svrhom uplate. Uplatnica će biti poslata korisniku na e-mail adresu koju je koristio pri prijavi na web stranicu i time će porudžbina biti potvrđena. (slika 6).

LOGO

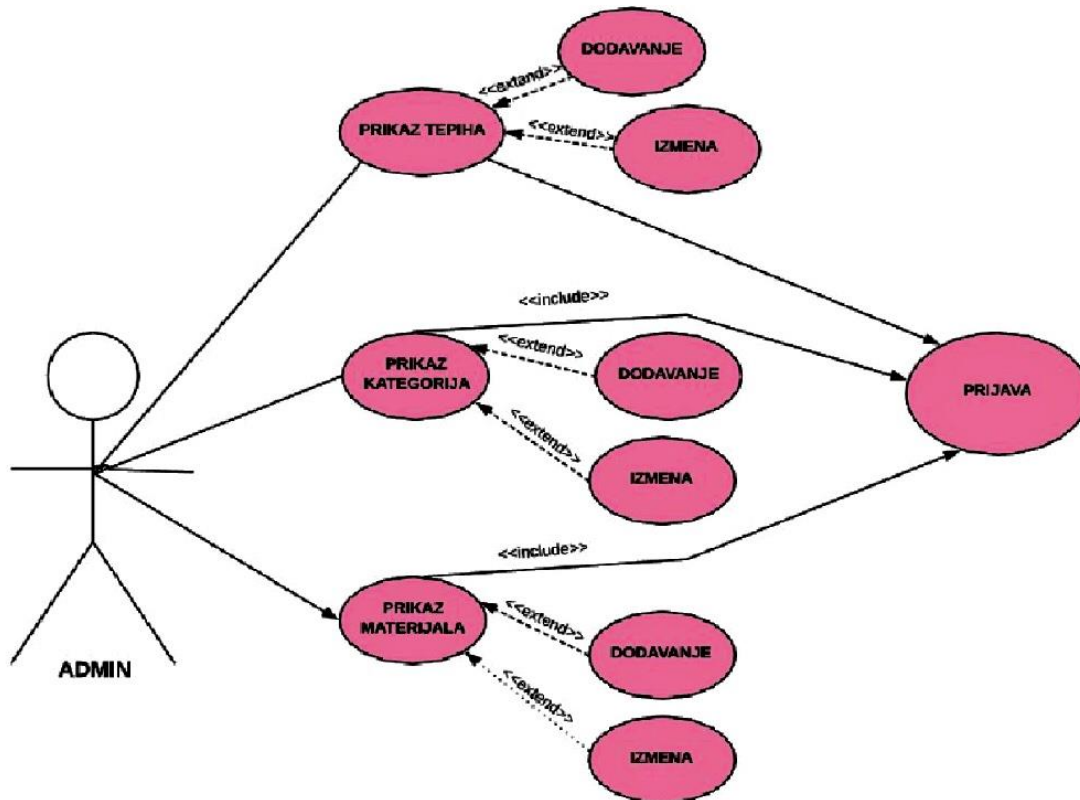
UPLATILAC NAŠA GAJEVIĆ	ŠIFRA PLAC	VALUTA	IZNOS = 5.117,00
SVRHA UPLATE E-KUPOVINA TERPIHA	RAČUN PRIMALCA 175 - 342-11		
PRIMALAC GALERIJA PODOVA	MODEL I PORIV NA BR		

slika 6

### 3.2. Funkcije

U okviru sistema postoje dva tipa korisnika: administrator (zaposleni) i potencijalni kupac (korisnik)

Dijagram slučajeva korišćenja za administratora bice prikazan u nastavku:



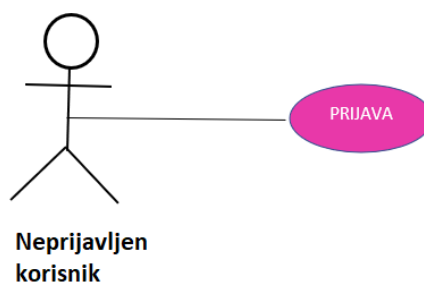
slika 7

Ono sto administrator moze da uradi jeste da upravlja sledecim opcijama: prikaz tepiha, prikaz kategorija, prikaz materijala.

Ovim opcijama moze da manipulise kada je prijavljen i to mu daje mogucnost da iste dodaje ili izmeni.

Ukoliko nije ispravno prijavljen administrator nece moci da pristupi panelu i samim tim nece moci da vrsi ni jednu od navedenih funkcionalnosti.

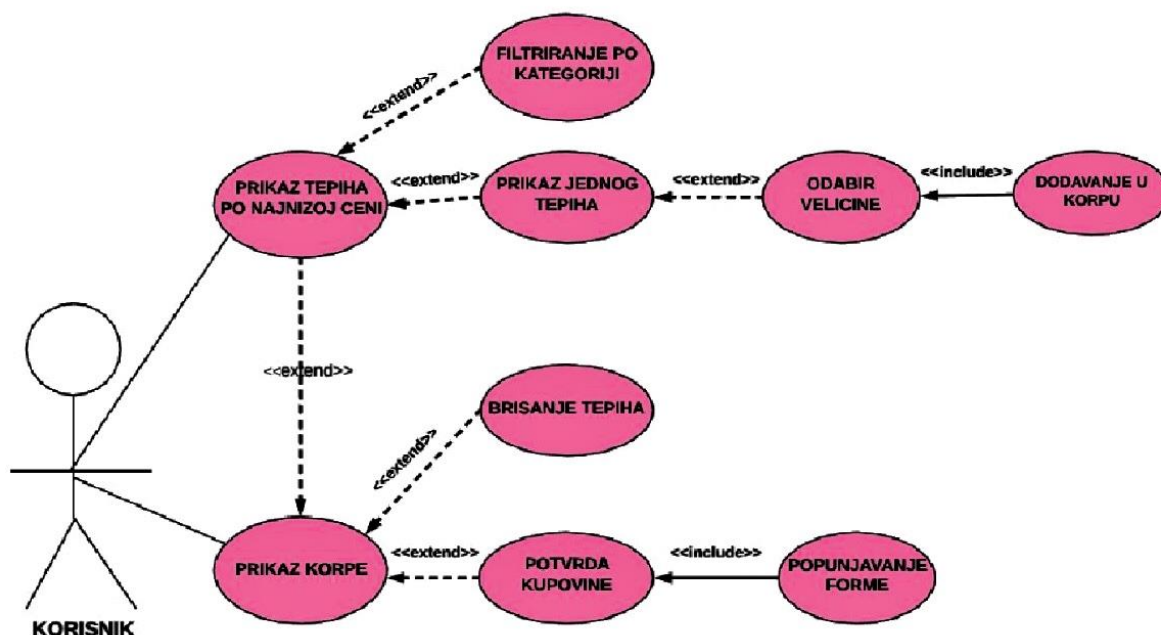
Dijagram slučajeva korišćenja za potencijalnog kupca koji nije prijavljen:



slika 8

Neprijavljeni korisnik može samo da se prijavi na sistem i ukoliko se prijavi sa ispravnim parametrima ( korisničko ime i lozinka) može pristupiti funkcionalnostima sistema. Prostije rečeno, prijavom dobija ulogu zaposlenog korisnika.

Dijagram slučajeva korišćenja za prijavljenog potencijalnog kupca prikazan je u nastavku:



slika 9

Korisnik ukoliko je prijavljen ima mogućnost da vrši pretragu po kategorijama koje su prikazane na dijagramu. Sve to obavlja klikom na određeno dugme koje je jasno prikazano na web strani.

Nakon obavljanja pretrage i odabira artikala korisnik može izvršiti pregled korpe gde mu se daje na uvid poručeni artikali koji može da potvrdi ili ponisti. Ukoliko potvrdi porudžbinu sledi korak u kom popunjava formu sa ličnim podacima za slanje i plaćanje porudžbine.

### 3.3. Pogodnosti za upotrebu

Da bi potencijalni kupac pristupio portalu neophodno je da poseduje osnovno znanje o korišćenju Interneta. Ovaj portal ne zahteva od korisnika da instalira dodatni softver. Takođe Internet portal omogućava zaposlenima rad od kuće čime se postiže ušteda na troškovima prevoza zaposlenih i daje im mogućnost da samostalno organizuju svoje radno vreme u toku radne nedelje. Kupcima internet prodaja omogućava da iz udobnosti sopstvenog doma da na brz i lak način izvrše pretragu i kupovinu željenih artikala.



### 3.4. Zahtevane performanse

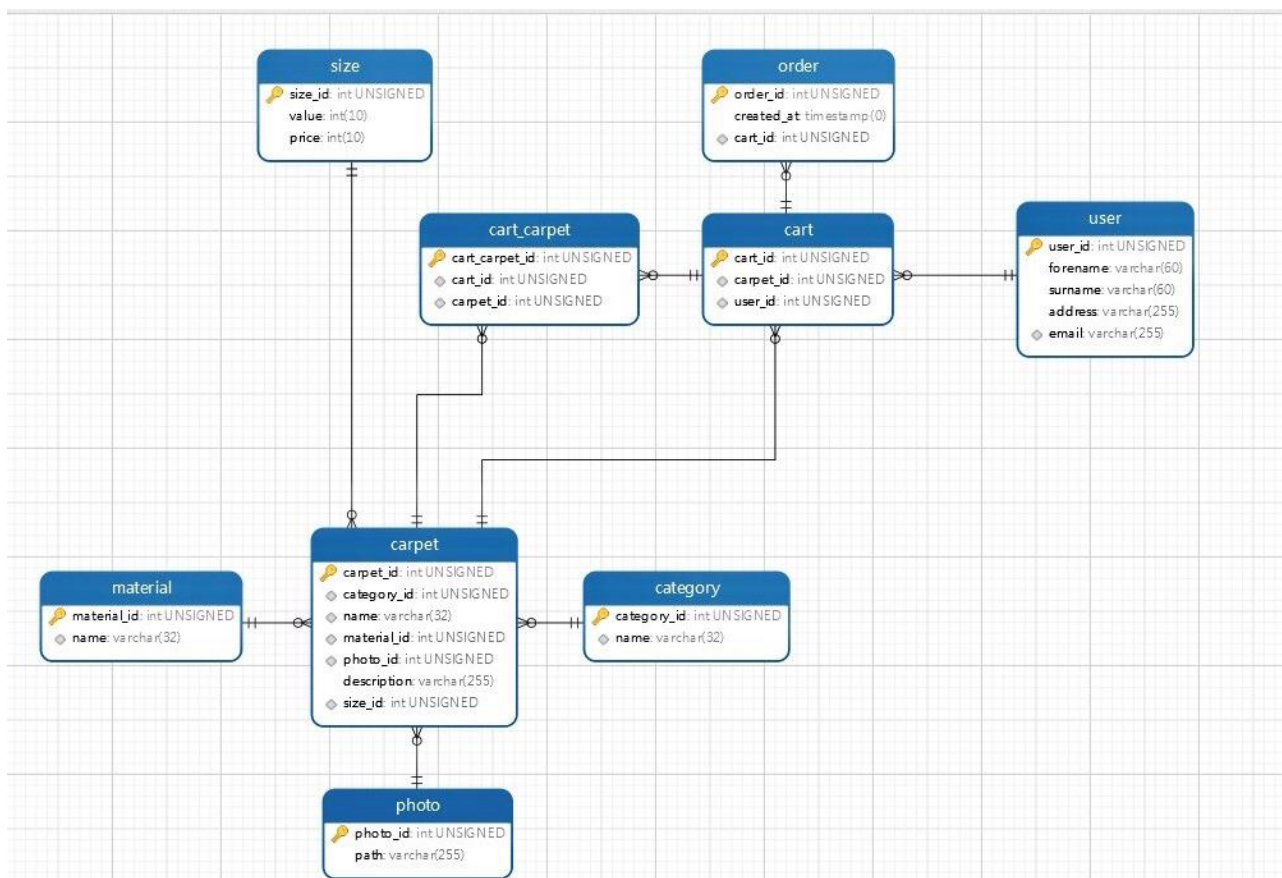
Aplikacija treba da pruži klijentima sto jednostavniji i brzi način kupovine, bez obzira na broj korisnika koji istovremeno pristupaju platformi. Sa druge strane portal omogućava administratorima da u nekoliko koraka dodaju i izmene artikla koji su na stanju u internet prodavnici.

Slabosti aplikacije su sto ona zahteva internet konekciju i neophodno je da korisnici poseduju određeno predznanje za upotrebu Internet pretraživaca i Internet kupovine. Problem može nastati usled izostanka živog kontakta i komunikacije, te potencijalni kupci mogu pogrešno poručiti artikal ili ponistiti željenu porudzbini. Uvek postoji rizik za preuzimanje ideja od strane konkurentskih kompanija.

Sanse su ušteda vremena zaposlenih i potencijalnih kupaca kao i povećanje broja novih klijenata sto doprinosi povećanom obimu posla i visim prihodima kompanije.

### 3.5. Zahtevi baze podataka

U nastavku je prikazan model baza podataka (slika 10)



slika 10

Za rad aplikacije neophodna je upotreba strukturiranih podataka. Radi se o veoma organizovanim podacima koje racunar lako prepoznaje i razume. Koriste se relaciona baza podataka u koju se podaci mogu na jednostavan način uneti, pretraživati i izmeniti, kao i obrisati iz baze podataka.

Prisutno je ograničenje resursa baze podataka što implicira da je ograničena količina podataka u bazi.

Sve promene koje se izvrše u bazi su usko povezane sa sistemom i svaka izmena se automatski vidi na portalu. Iz tog razloga baza podataka mora uvek biti na raspolaganju za korišćenje, dodavanje i vršenje izmena.

Opis tabela sa njihovim karakteristikama i poljima biće prikazan sledećim slikama:

Oznake:

1. PK – primarni ključ tabele
2. NN – ne dozvoljava NULL vrednost
3. UQ – jedinstveni zapis
4. UN – da li je vrednost u koloni UNSIGNED (prirodan broj)
5. AI – auto inkrementalna vrednost

## BAZAA TEBALE FALE

Tabela administrator (slika 11):

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	administrator...	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...				
2	username	VARCHAR	32	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'		utf8_unicode_ci		
3	password_hash	VARCHAR	128	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'		utf8_unicode_ci		

slika 11

Tabela administrator\_token (slika 12):

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	administrator...	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...				
2	administrator_...	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default				
3	created_at	TIMESTAMP		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	current_timestam...				
4	token	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8_unicode_ci		
5	expires_at	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default				
6	is_valid	TINYINT	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'				

slika 12

Tabela user (slika 13):

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation
1	user_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...		
2	email	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'		utf8_unicode_ci
3	password_hash	VARCHAR	128	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'		utf8_unicode_ci
4	forename	VARCHAR	64	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'		utf8_unicode_ci
5	surname	VARCHAR	64	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'		utf8_unicode_ci
6	phone_number	VARCHAR	24	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'		utf8_unicode_ci
7	postal_address	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8_unicode_ci

slika 13

Tabela user\_token (slika 14):

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation
1	user_token_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...		
2	user_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8_unicode_ci
3	created_at	TIMESTAMP		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	current_timestam...		
4	token	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8_unicode_ci
5	expires_at	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8_unicode_ci
6	is_valid	TINYINT	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'		

slika 14

Tabela carpet (slika 15):

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation
1	carpet_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...		
2	name	VARCHAR	128	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'		utf8_unicode_ci
3	category_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'		utf8_unicode_ci
4	excerpt	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'		utf8_unicode_ci
5	description	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8_unicode_ci
6	status	ENUM	'XXL', 'XL', 'X...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	XXL		utf8_unicode_ci
7	is_promoted	TINYINT	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'		
8	created_at	TIMESTAMP		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	current_timestam...		

slika 15



Tabela carpet\_price (slika 16):

Host: localhost Database: aplikacija Table: carpet\_price Data Query\*

Basic Options Indexes Foreign keys Partitions CREATE code ALTER code

Name: carpet\_price

Comment:

Columns: Add Remove Up Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	carpet_price_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...
2	carpet_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
3	price	DECIMAL	10,2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0.00'
4	created_at	TIMESTAMP		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	current_timestam...

slika 16

Tabela cart (slika 17):

Host: localhost Database: aplikacija Table: cart Data Query\*

Basic Options Indexes Foreign keys Partitions CREATE code ALTER code

Name: cart

Comment:

Columns: Add Remove Up Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	cart_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...
2	user_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
3	created_at	TIMESTAMP		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	current_timestam...

slika 17

Tabela cart\_carpet (slika 18):

Host: localhost Database: aplikacija Table: cart\_carpet Data Query\*

Basic Options Indexes Foreign keys Partitions CREATE code ALTER code

Name: cart\_carpet

Comment:

Columns: Add Remove Up Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	cart_carpet_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...
2	cart_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
3	carpet_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
4	quantity	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'

slika 18

Tabela category (slika 19):

Host: localhost Database: aplikacija Table: category Data Query\*

Basic Options Indexes Foreign keys Partitions CREATE code ALTER code

Name: category

Comment:

Columns: + Add - Remove ▲ Up ▼ Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	category_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...
2	name	VARCHAR	32	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
3	image_path	VARCHAR	128	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
4	parent_cat...	INT	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	No default

slika 19

Tabela order (slika 20)

Host: localhost Database: aplikacija Table: order Data Query\*

Basic Options Indexes Foreign keys Partitions CREATE code ALTER code

Name: order

Comment:

Columns: + Add - Remove ▲ Up ▼ Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	order_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...
2	created_at	TIMESTAMP		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	current_timestam...
3	cart_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
4	status	ENUM	'rejected', 'ac...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	pending

slika 20

Tabela phot (slika 21)

Host: localhost Database: aplikacija Table: photo Data Query\*

Basic Options Indexes Foreign keys Partitions CREATE code ALTER code

Name: photo

Comment:

Columns: + Add - Remove ▲ Up ▼ Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	photo_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...
2	carpet_id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
3	image_path	VARCHAR	128	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'

slika 21

### 3.6. Projektna ograničenja

Projekat zahteva modernu mrežnu opremu, servere nove generacije i veliki propusni opseg podataka. Neophodni su dodatni uređaji i softveri koji obezbeđuju bezbedan prenos podataka. Aplikacija mora da bude realizovana na Node.js platformi korišćenjem Nest.js razvojnog okvira i sav kôd aplikacije treba da bude organizovan prema pravilima MVC arhitekture. Baza podataka mora da bude relaciona i treba koristiti MySQL/MariaDB RDBMS.

### 3.7. Sistemske karakteristike softvera

Baza podataka usko je povezana sa sistemom i svaka izmena u bazi vidljiva je i u sistemu i na portalu. Iz tog razloga mora se obratiti posebna paznja na bezbednost i tacnost podataka, s obzirom da se podaci konstantno razmenjuju izmedju klijenta i web servera, kao i izmedju web servera i baze podataka. Da bi sve funkcionisalo moramo imati kvalitetnu Internet konekciju. Svi podaci koje klijent unosi pri prijavi na portal ili pri izvršenju kupovine su zasticeni kako se ne bi nariusio integritet klijenata. Postoji unapred utvrdjen raspored kada se održava sistem i obavlja backup podataka.

### 3.8. Dopunske informacije

Pored standardne kupovine u prodavnicama omogućavanje internet prodaje donosi sa sobom benefite za kompaniju, zaposlene i potencijalne kupce. Pre svega smanjuju se troškovi najma prostora za prodavnicu, troškovi prevoza zaposlenih kao i dodatni troškovi firme. Sto se tice kupaca dobijaju na uštedi svog vremena jer iz udobnosti svog doma mogu da pretraže i poruče artikule koji ih interesuju. Smanjena je upotreba papira (racuni, fakture, uplatnice) jer se sve odvija elektronski i time se podstiče ekologija i zaštita životne sredine.