**Veb aplikacija za e-katalog mobilnih telefona**

Praktikum - Internet i veb tehnologije

**Student:**

Petar Milovanović 2017202002

**Sadržaj**

1. **Uvod**
   1. Cilj razvoja
   2. Obim Sistema
   3. Prikaz proizvoda
      1. Perspektiva proizvoda
      2. Funkcije proizvoda
2. **Specifikacija zahteva**
   1. Spoljašnji interfejs
   2. Funkcije
   3. Pogodnost za upotrebu
   4. Zahtevane performanse
   5. Zahtevi baze podataka
   6. Projektna ograničenja

1. Uvod

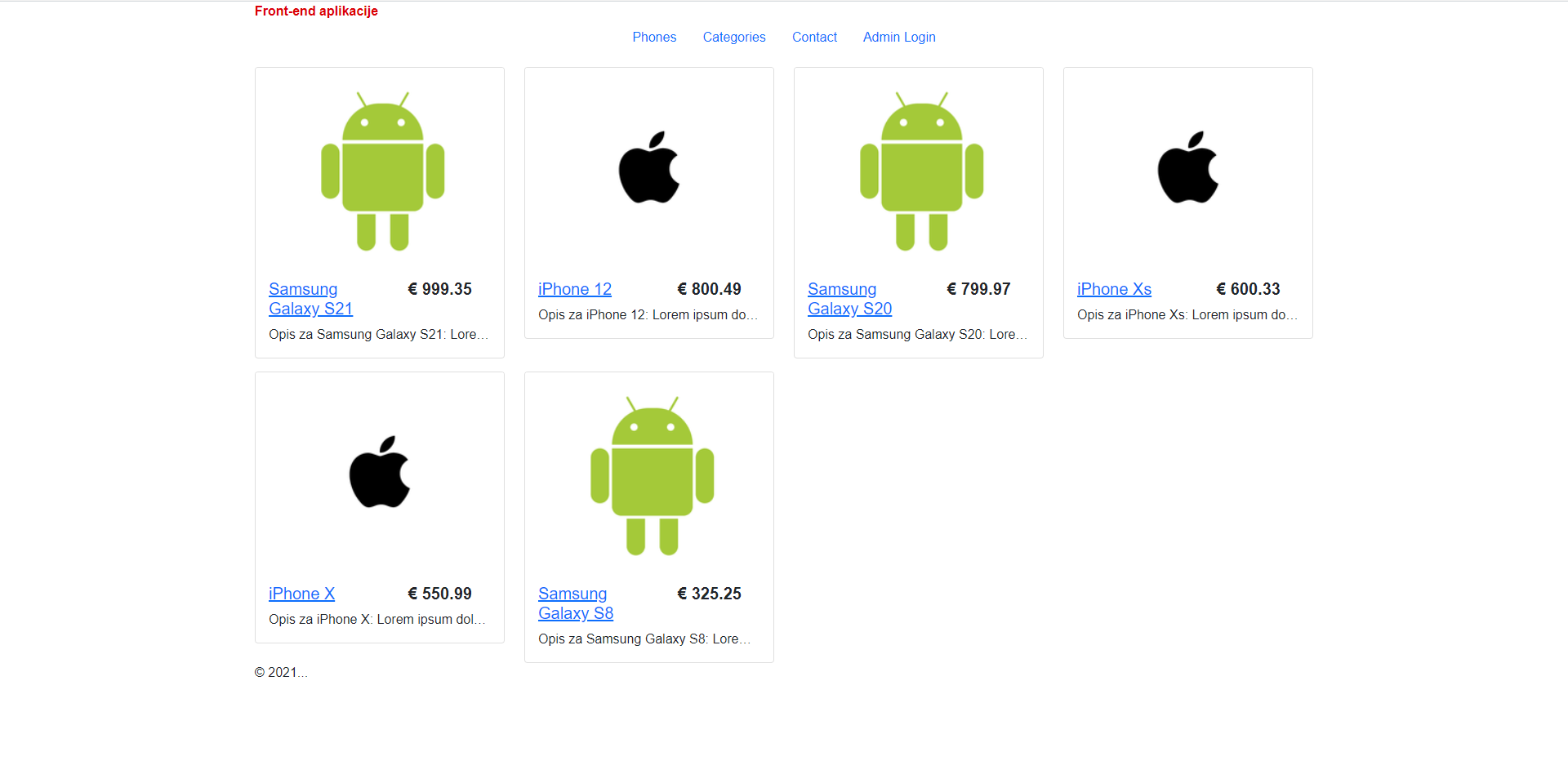
**1.1 Cilj razvoja**

Cilj razvoja ovog projekta jeste napraviti funkcionalnu veb aplikaciju za pregled e-kataloga za pregled mobilnih telefona.

**1.2 Obim sistema**

Aplikacija je rađena po MVC arhitekturi i podeljena je na dva dela, na back-end i front-end i radi na node.js platformi. Back-end, koji koristi express biblioteku, komunicira sa MySQL bazom podataka koja je relacionog tipa. Sa druge strane, front-end je urađen korišćenjem React biblioteke.

**1.3 Prikaz proizvoda**

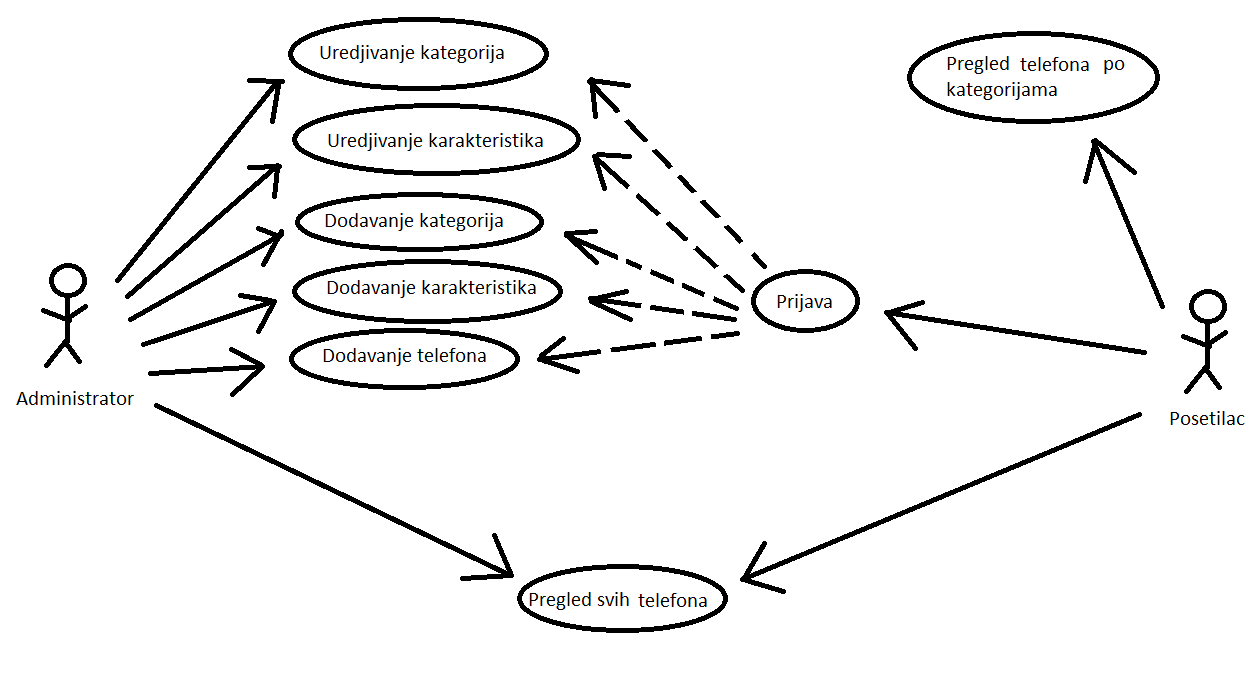


**1.3.1 Perspektiva proizvoda**

Aplikacija koja je namenjena za studijske potrebe i time nema komercijalnu perspektivu.

**1.3.2 Funkcije proizvoda**

Funkcija proizvoda se sastoji od jednostavnog pregleda i informisanja korisnika o različitim mobilnim uređajima i njihovim karakteristikama.



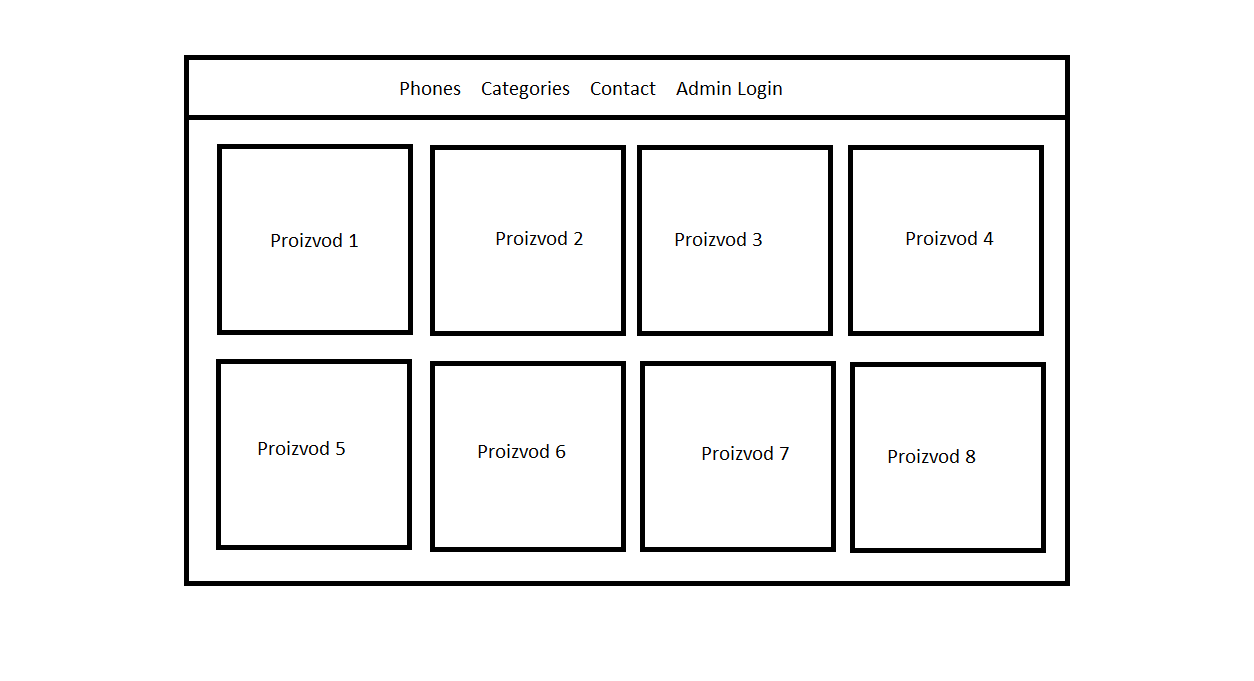
2. Specifikacija zahteva

Aplikacija treba da omogući administratorima da dodaju nove artikle u bazu podataka aplikacije za elektronski katalog za prodaju mobilnih telefona. Samo prijavljeni korisnici, koji se na portal prijave sa ispravnim parametrima naloga administratora sadržaja mogu da pristupe administrativnom panelu portala. U ovom panelu mogu da dodaju nove telefone koji se sastoji od naslova, slike, detaljnog opisa (sa predefinisanim setovima karakteristika), cene i treba da pripadaju određenoj kategoriji. Svaki telefon treba da ima posebno polje za vrstu operativnog sistema (Android, Windows, iOS itd), polje za količinu radne memorije, količinu skladišnog prostora, dijagonalu ekrana, kao i podatke o tome koje telekomunikacione standarde podržava (GSM, 3G, LTE, 4G itd) i druge važne karakteristike za kataloški prikaz. Sa korisničke strane treba omogućiti prikaz svih telefona poređanih po ceni. Kategorije prikazati u vidu menija na veb sajtu. Korisniku treba omogućiti pretragu po filterima u kojima korisnik može da uključi operativni sistem za koji pretražuje telefone, opsege radne i skladišne memorije, kao i vrste telekomunikacionog standarda koji telefon podržava. Prikazati samo one telefone koji odgovaraju podešenoj grupi filtera. Kada korisnik otvori stranicu nekog telefona treba da vidi sve detalje, kao što su naslov, slika, opis, cena, kao i prikaz ostalih podataka koji su sastavni deo zapisa telefona u bazi podataka (operativni sistem, memorija itd). Grafički interfejs veb sajta treba da bude realizovan sa responsive dizajnom

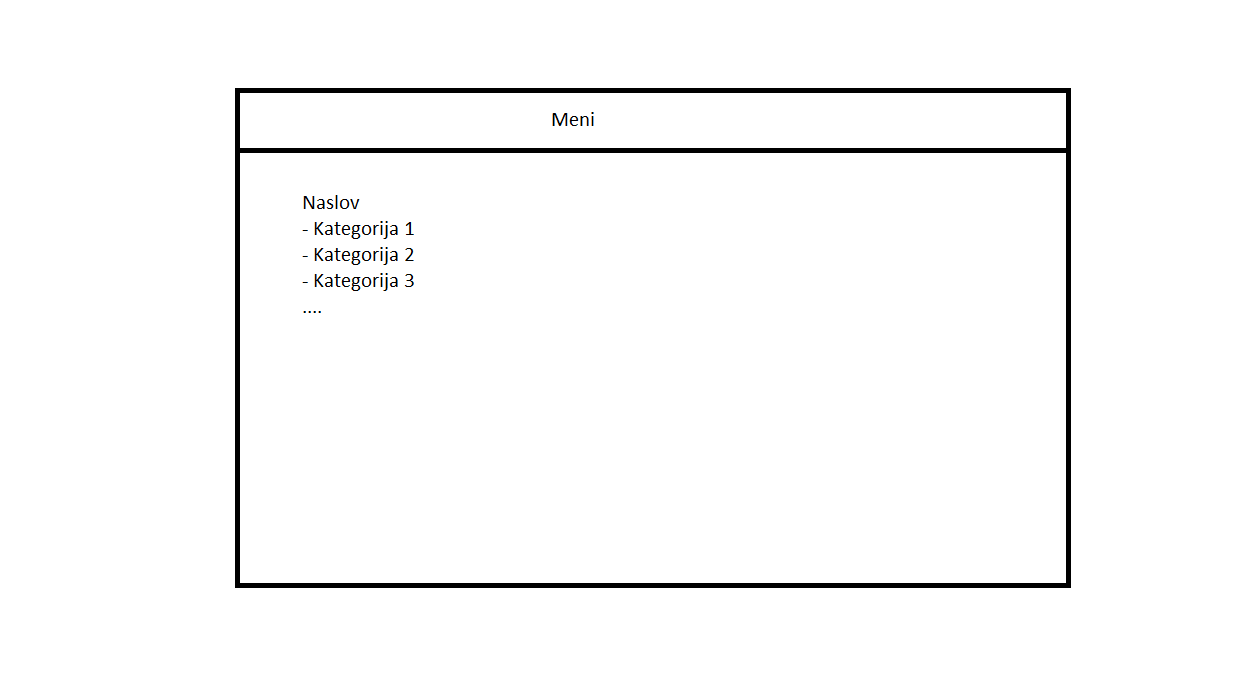
**2.1 Spoljašnji interfejs**

U nastavku su slike skice interfejsa

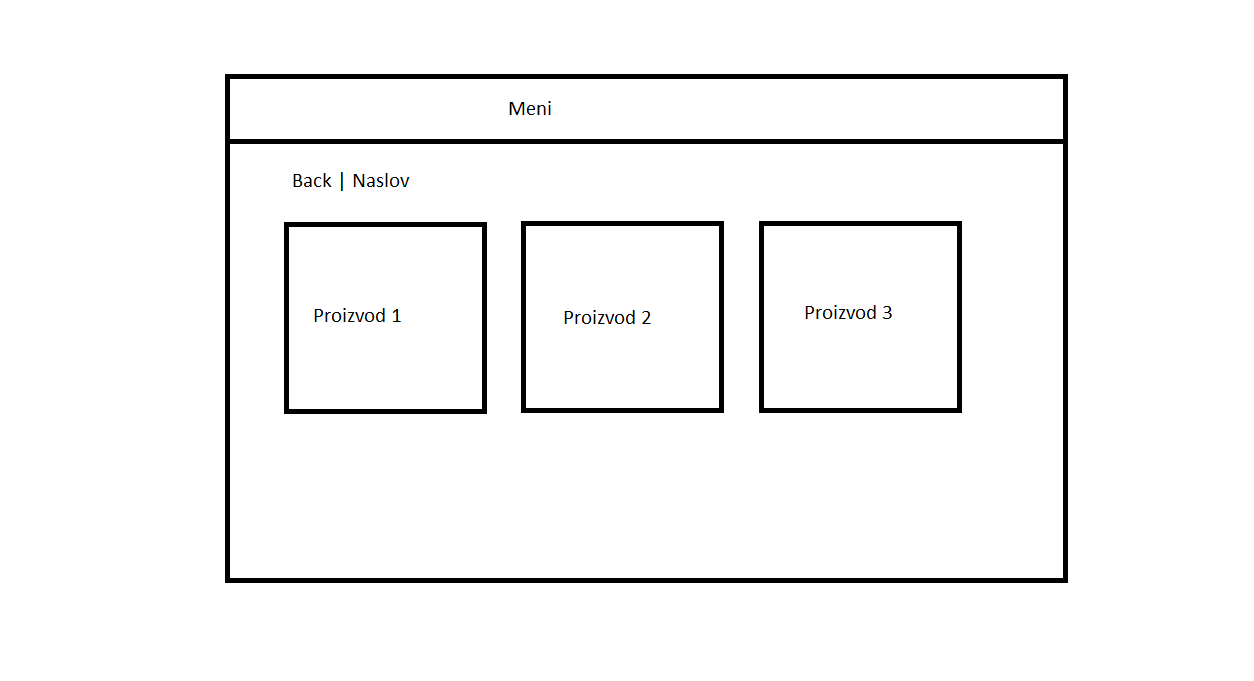
Homepage



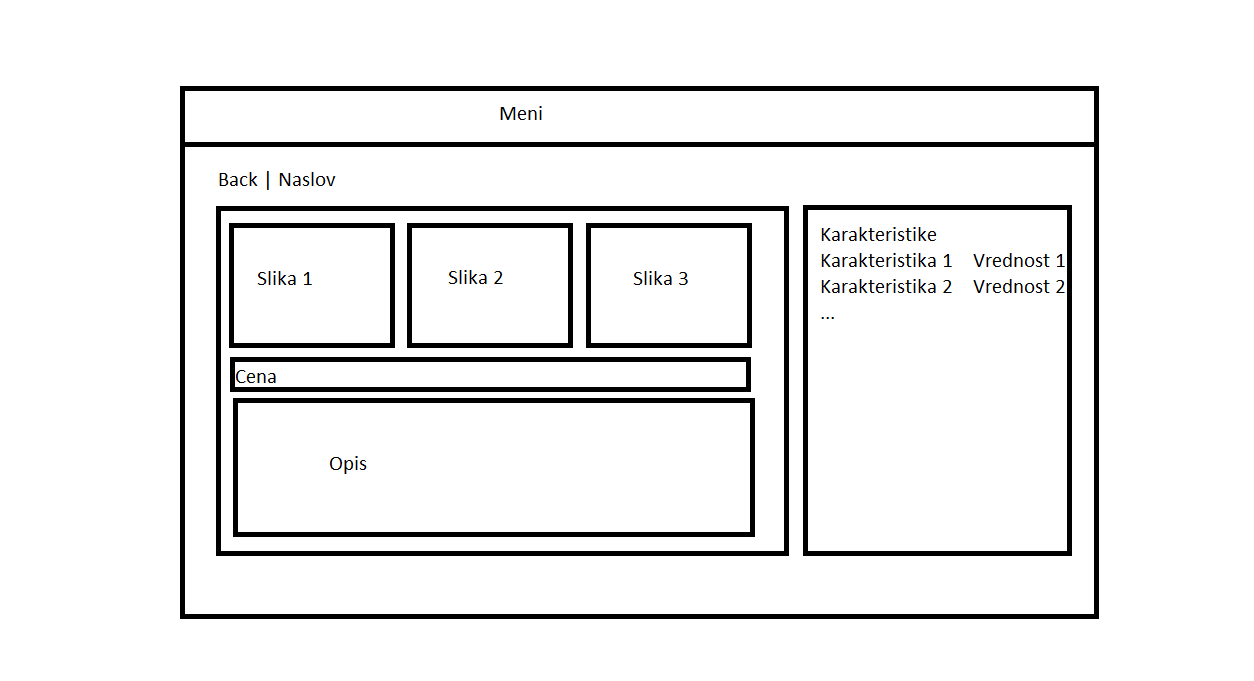
Stranica za pretragu telefona po kategorijama

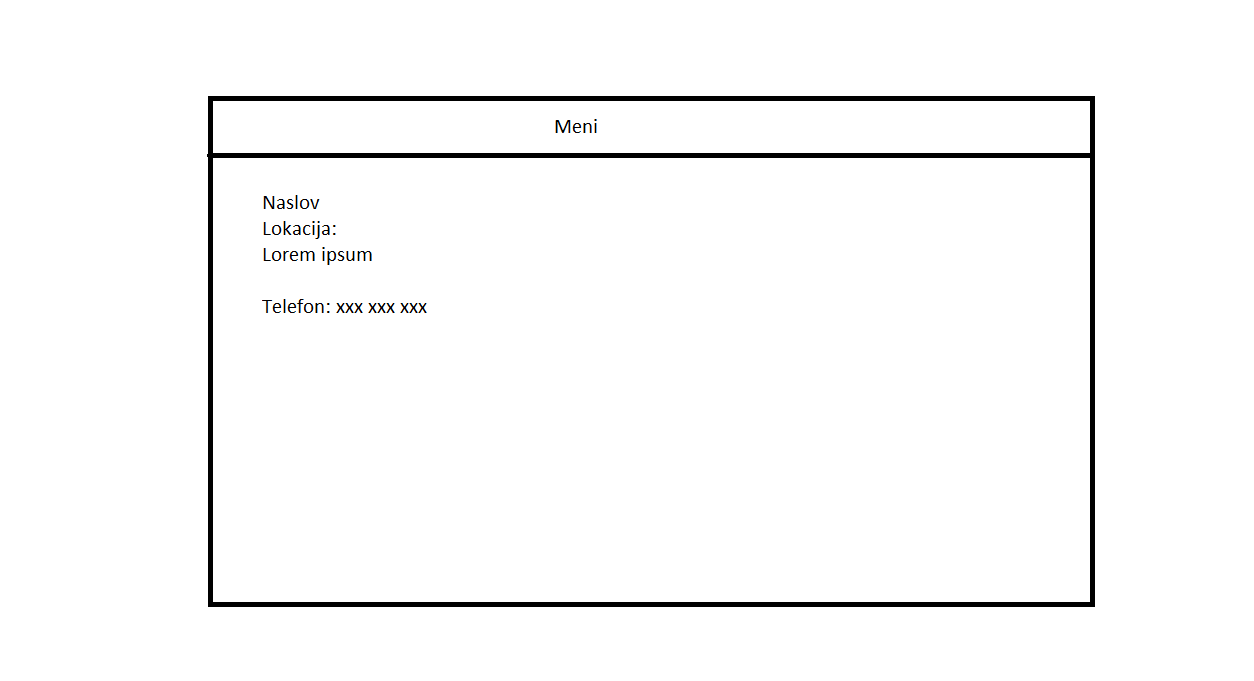


Prikaz telefona na osnovu izabrane kategorije

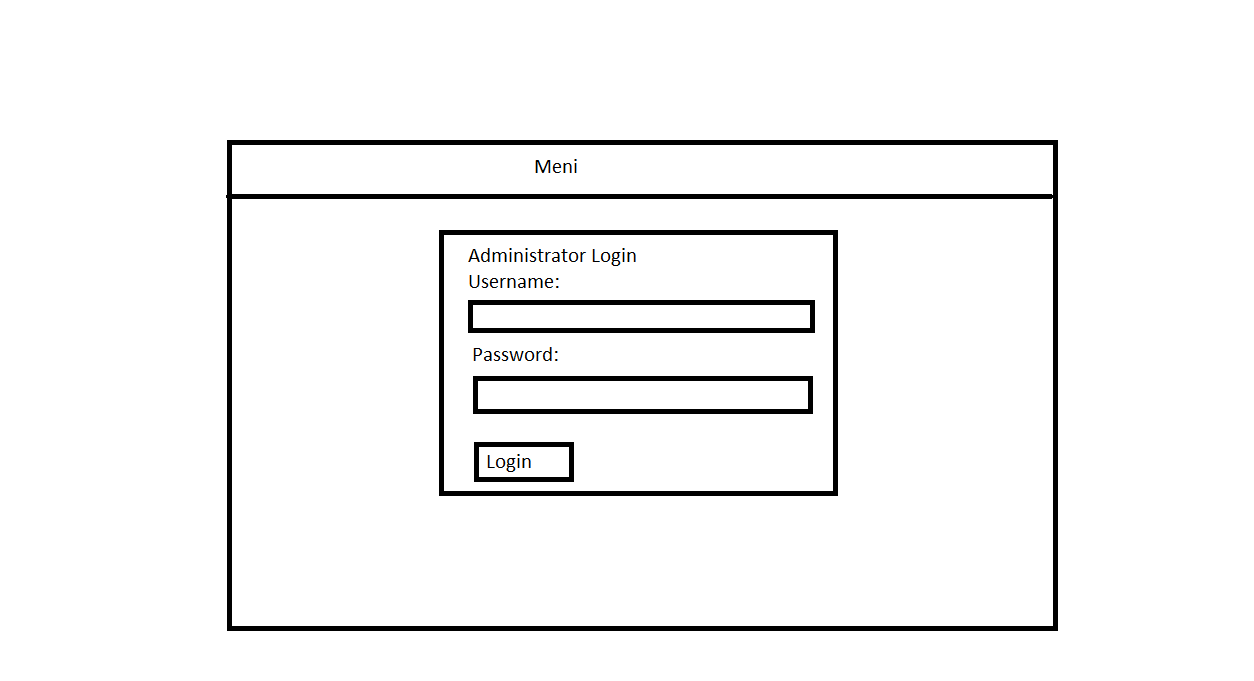


Prikaz pojedinačnog telefona

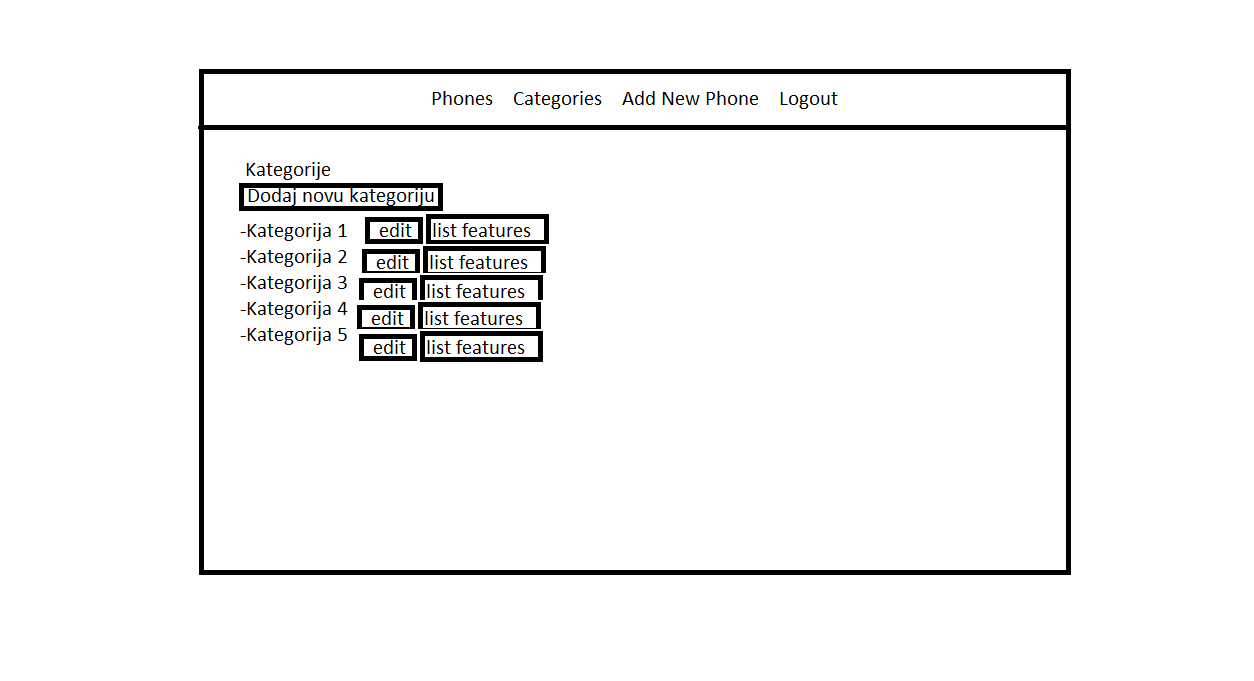


Kontakt stranica

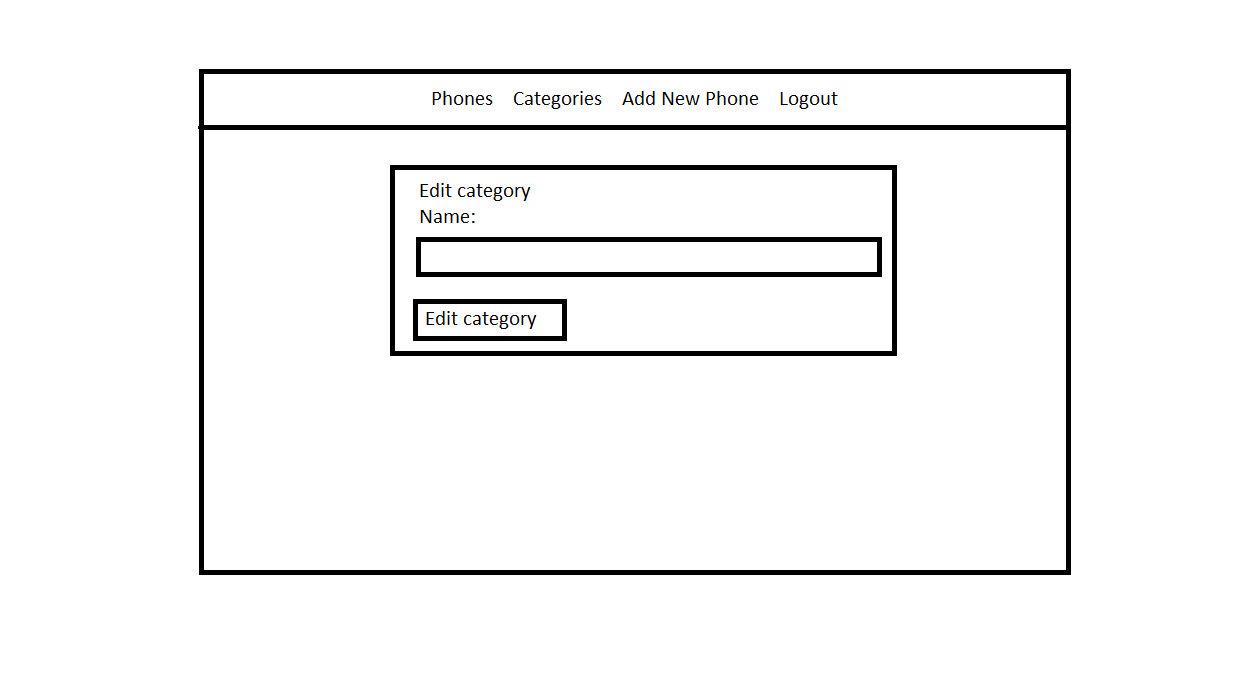
Login stranica za administratore



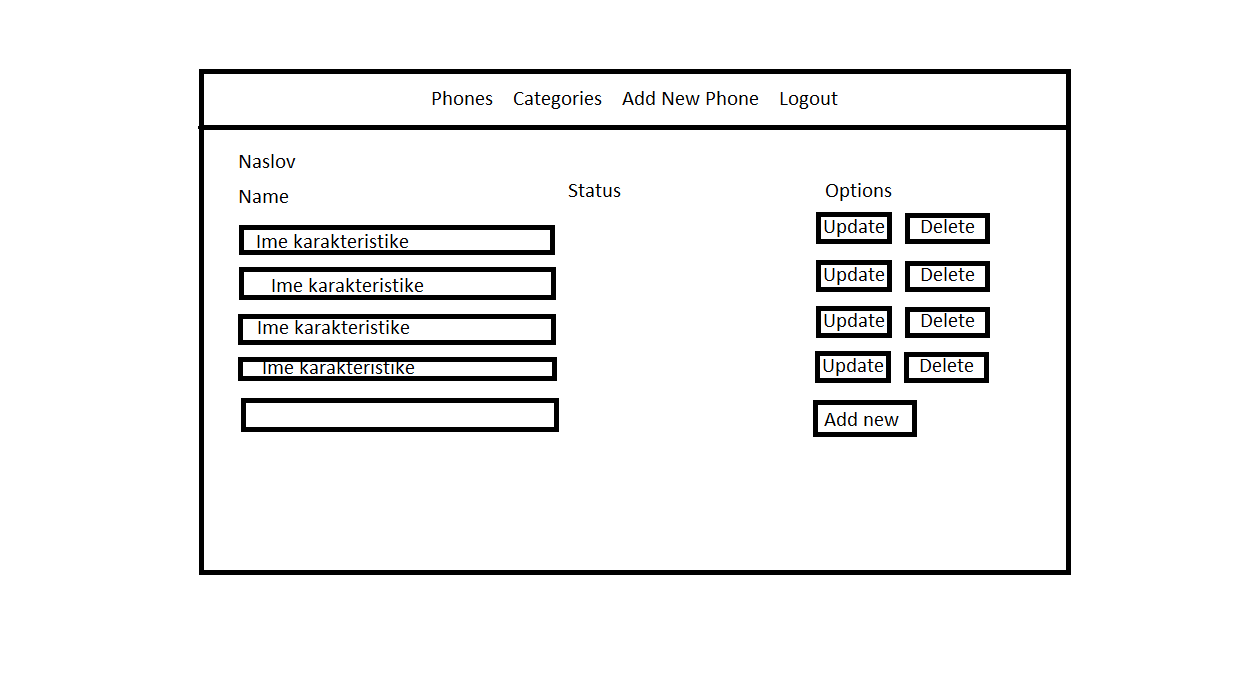
Prikaz kategorija u dashboard-u za administrator

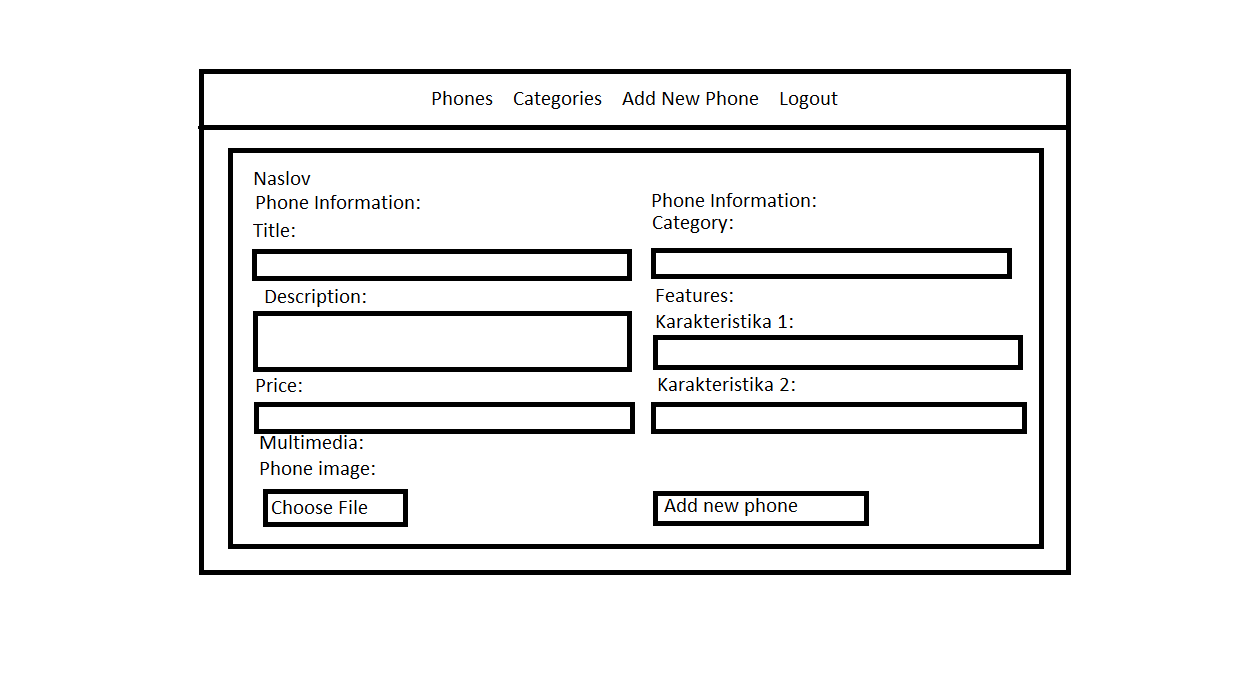


Stranica za editovanje kategorije



Stranica za prikaz feature-a kategorije

Stranica za dodavanje novog telefona



**2.2 Funkcije**

Posetilac:

* Pregled i pretraga telefona i njegovih karakteristika

Administrator:

* Autorizacija korišćenjem JWT-a
* Dodavanje i ažuriranje kategorija
* Dodavanje i ažuriranje karakteristika
* Dodavanje novih telefona

**2.3 Pogodnost za upotrebu**

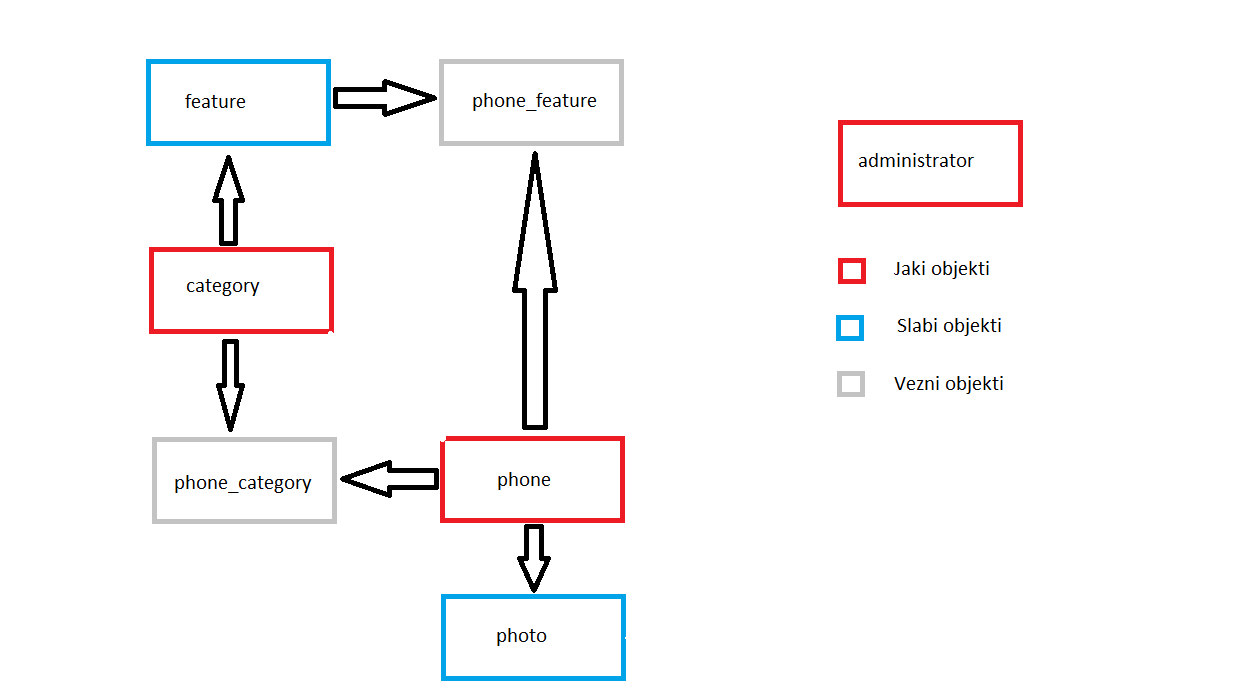
Aplikacija je informacionog tipa i time je napravljena da bude pogodna za svakodnevnu upotrebu.

**2.4 Zahtevane performanse**

Aplikacija nije hardverski zahtevna tako da ne bi trebalo da predstavlja problem pokretanja čak i na slabijim platformama.

**2.5 Zahtevi baze podataka**

Slika modela baze



⦁ administrator

⦁ administrator\_id PK UNSIGNED INT NOT NULL AUTO\_INCREMENT

⦁ username UQ VARCHAR(64) NOT NULL

⦁ password\_hash VARCHAR(255) NOT NULL

⦁ is\_active UNSIGNED TINYINT(1) NOT NULL DEFAULT 1

⦁ category

⦁ category\_id PK UNSIGNED INT NOT NULL AUTO\_INCREMENT

⦁ name UQ VARCHAR(128) NOT NULL

⦁ phone

⦁ phone\_id PK UNSIGNED INT NOT NULL AUTO\_INCREMENT

⦁ created\_at TIMESTAMP CURRENT TIMESTAMP

⦁ title UQ VARCHAR(128) NOT NULL

⦁ description TEXT NOT NULL

⦁ price UNSIGNED DECIMAL(10,2)

⦁ feature

⦁ feature\_id PK UNSIGNED INT NOT NULL AUTO\_INCREMENT

⦁ name UQ' VARCHAR(64) NOT NULL

⦁ category\_id FK+UQ' UNSIGNED INT NOT NULL AUTO\_INCREMENT

⦁ phone\_feature

⦁ phone\_feature\_id PK UNSIGNED INT NOT NULL AUTO\_INCREMENT

⦁ phone\_id FK+UQ' UNSIGNED INT NOT NULL

⦁ feature\_id FK+UQ' UNSIGNED INT NOT NULL

⦁ value VARCHAR(64)

⦁ phone\_category

⦁ phone\_category\_id PK UNSIGNED INT NOT NULL AUTO\_INCREMENT

⦁ phone\_id FK UNSIGNED INT NOT NULL

⦁ category\_id FK UNSIGNED INT NOT NULL

⦁ value VARCHAR(255)

⦁ photo

⦁ photo\_id PK UNSIGNED INT NOT NULL AUTO\_INCREMENT

⦁ image\_path UQ VARCHAR(255) NOT NULL

⦁ phone\_id FK UNSIGNED INT NOT NULL

**2.6 Projektna ograničenja**

* Aplikacija mora da bude realizovana na Node.js platformi korišćenjem Express biblioteke. Aplikacija mora da bude podeljena u dve nezavisne celine: back-end veb servis (API) i front-end (GUI aplikacija). Sav kôd aplikacije treba da bude organizovan u jednom Git spremištu u okviru korisničkog naloga za ovaj projekat, sa podelom kao u primeru zadatka sa vežbi.
* Baza podataka mora da bude relaciona i treba koristiti MySQL ili MariaDB sistem za upravljanje bazama podataka (RDBMS) i u spremištu back-end dela aplikacije mora da bude dostupan SQL dump strukture baze podataka, eventualno sa inicijalnim podacima, potrebnim za demonstraciju rada projekta.
* Back-end i front-end delovi projekta moraju da budi pisani na TypeScript jeziku, prevedeni TypeScript prevodiocem na adekvatan JavaScript. Back-end deo aplikacije, preveden na JavaScript iz izvornog TypeScript koda se pokreće kao Node.js aplikacija, a front-end deo se statički servira sa rute statičkih resursa back-end dela aplikacije i izvršava se na strani klijenta. Za postupak provere identiteta korisnika koji upućuje zahteve back-end delu aplikacije može da se koristi mehanizam sesija ili JWT (JSON Web Tokena), po slobodnom izboru.
* Sav generisani HTML kôd koji proizvodi front-end deo aplikacije mora da bude 100% validan, tj. da prođe proveru W3C Validatorom (dopuštena su upozorenja - Warning, ali ne i greške - Error). Grafički korisnički interfejs se generiše na strani klijenta (client side rendering), korišćenjem React biblioteke, dok podatke doprema asinhrono iz back-end dela aplikacije (iz API-ja). Nije neophodno baviti se izradom posebnog dizajna grafičkog interfejsa aplikacije, već je moguće koristiti CSS biblioteke kao što je Bootstrap CSS biblioteka. Front-end deo aplikacije treba da bude realizovan tako da se prilagođava različitim veličinama ekrana (responsive design).
* Potrebno je obezbediti proveru podataka koji se od korisnika iz front-end dela upućuju back-end delu aplikacije. Moguća su tri sloja zaštite i to: (1) JavaScript validacija vrednosti na front-end-u; (2) Provera korišćenjem adekvatnih testova ili regularnih izraza na strani servera u back-end-u (moguće je i korišćenjem izričitih šema - Schema za validaciju ili drugim pristupima) i (3) 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 baze podataka; (3) dijagram organizacije delova sistema, gde se vidi veza između baze, back-end, front-end i korisnika sa opisom smera kretanja informacija; (4) popis svih aktivnosti koje su podržane kroz aplikaciju za sve uloge korisnika aplikacije prikazane u obliku Use-Case dijagrama; kao i (5) sve ostale elemente dokumentacije predviđene uputstvom za izradu dokumentacije po ISO standardu.
* Izrada oba dela aplikacije (projekata) i promene kodova datoteka tih projekata moraju da bude praćene korišćenjem alata za verziranje koda Git, a kompletan kôd aplikacije bude dostupan na javnom Git spremištu, npr. na besplatnim GitHub ili Bitbucket servisima, jedno spremište za back-end projekat i jedno za front-end projekat. 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).