1. **Uvod**
   1. **Cilj razvoja**

Cilj ovog projekta je razvoj veb aplikacije za elektronsku trgovinu naktiom

* 1. **Obim sistema**

Sistem se realizuje korišćenjem node.js platforme, korišćenjem nest.js razvojnog okvira. Za skladištenje informacija se koristi relaciona MySQL baza podataka. Grafički korisnički interfejs je realizovan korišćenjem HTLM i CSS programskih jezika.

* 1. **Prikaz proizvoda**

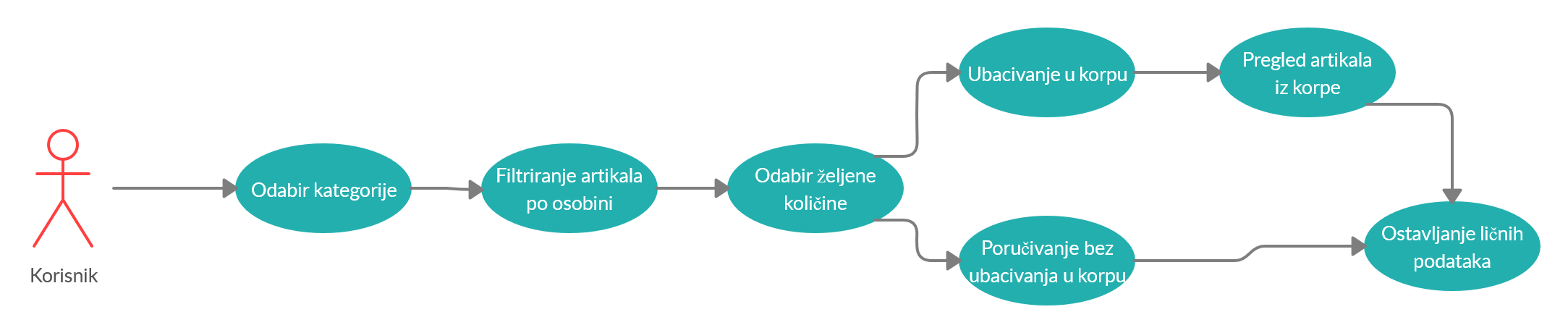
Korisnik traži željeni artikal u odgovarajućoj kategoriji. Pomoću filtera pretrage, sužava svoj izbor. Kada pronađe ono što traži, proizvod ubacuje u korpu, ili ga odmah poručuje. U koliko ima potrebe, vrši izmenu količine proizvoda. Popunjava formular sa svojim ličnim podacima za dostavu. Na kraju dobija informaciju da je porudžbina prihvaćena ili odbijena, a na svoju email adresu dobija listu poručenih proizvoda, kao još jednu potvrdu da je poružbina realizovana.

Prednost ovog sistema je u tome što korinik nije u obavezi da registruje svoj lični nalog, već je dovoljno da ostavi svoje podatke i poružbina će biti prihvaćena.

* + 1. **Perspektiva proizvoda**

Ova aplikacija omogućava korisnicima da preko veb browser-a pronađu željeni artikal, izvrše poručivanje robe i dobiju povratnu informaciju da je porudžbina primljena.

* + 1. **Funkcije proizvoda**



* + 1. **Karakteristike korisnika**

Za uspešno korišćenje aplikacije nije potrebno predznanje korisnika. Prethodno iskustvo korišćenja sličnih aplikacija nije potrebno jer je dizajn aplikacije intuitivan.

* + 1. **Ograničenja**

Kupci nemaju mogućnost registracije naloga, već sve funkcije obavljaju kao posetioci sajta.

Naloge jedino mogu da imaju administratori sajta.

* 1. **Definicije**

[Web aplikacija](https://en.wikipedia.org/wiki/Web_application)

[Elektronska trgovina](https://en.wikipedia.org/wiki/E-commerce)

[Operativni sistem](https://en.wikipedia.org/wiki/Operating_system)

[Web browser](https://en.wikipedia.org/wiki/Web_browser)

[Front end i back end](https://en.wikipedia.org/wiki/Front_end_and_back_end)

[Baza podataka](https://en.wikipedia.org/wiki/MySQL)

1. **Reference**

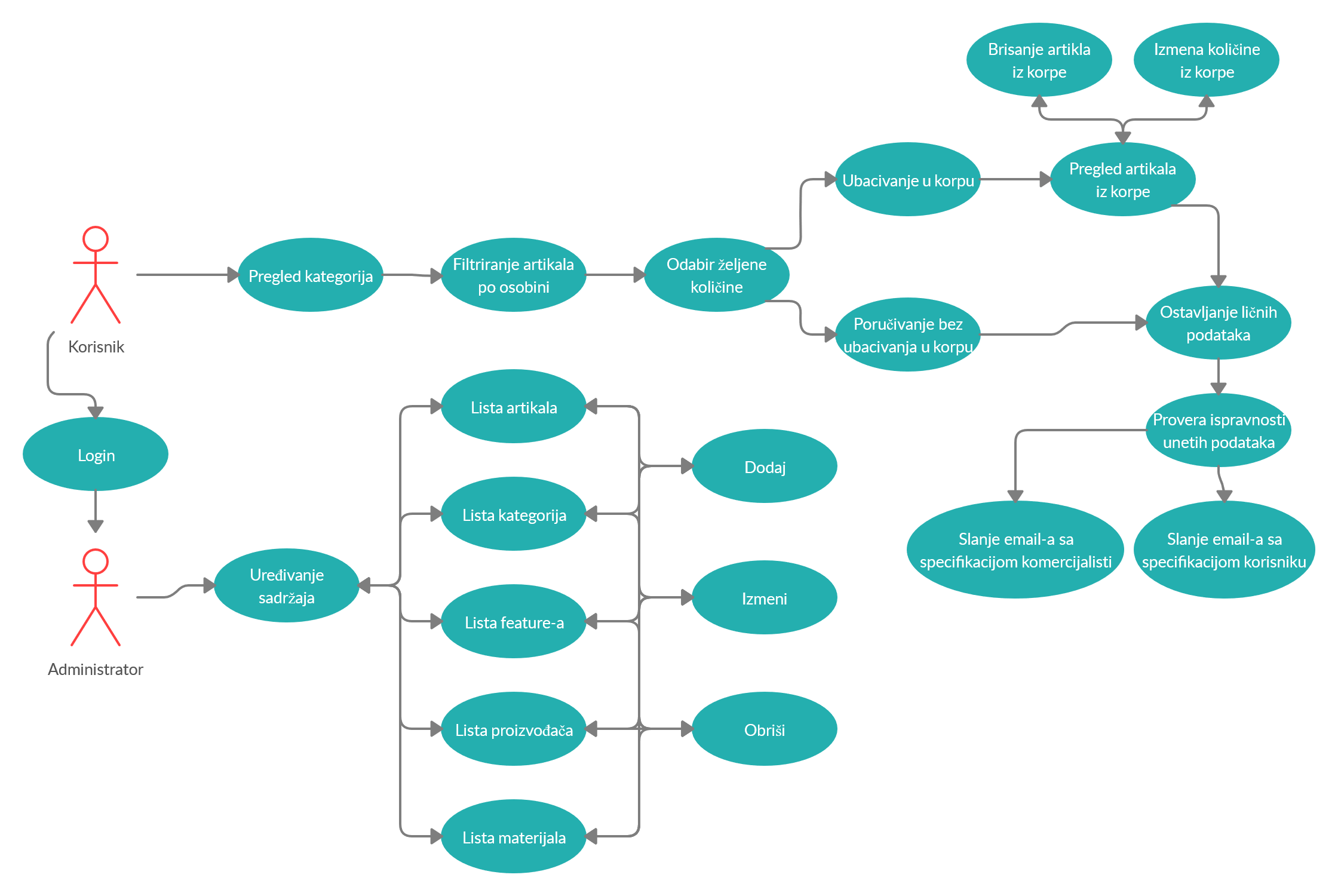
[Materijali sa stranice predmeta](http://predmet.singidunum.ac.rs/course/view.php?id=454)

[Video materijali](https://www.youtube.com/watch?v=xiUIQRkYYuI&list=PL-UTrxF0y8kKkQ7YPWPjYAAentBEbGA9g)

1. **Specifikacija zahteva**
   1. **Spoljašnji interfejsi**

Stabilna internet konekcija je obavezna za korišćenje ove aplikacije, pristupa se preko http protokola. Aplikacija je povezana sa spoljašnjom bazom podataka.

* 1. **Funkcije**



Funkcije koje se realizuju od strane korisnika su pretraga artikala po kategoriji, dodavanje željenih artikala u korpu, izmena količine ili brisanje iz korpe, filtriranje artikala po određenoj karakteristici i poručivanje artikala.

Funkcije koje realizuje administrator su dodavanje, brisanje ili izmena artikala, kategorija, osobina artikala, proizvođača i materijala kao i dodavanje i uklanjanje fotografija.

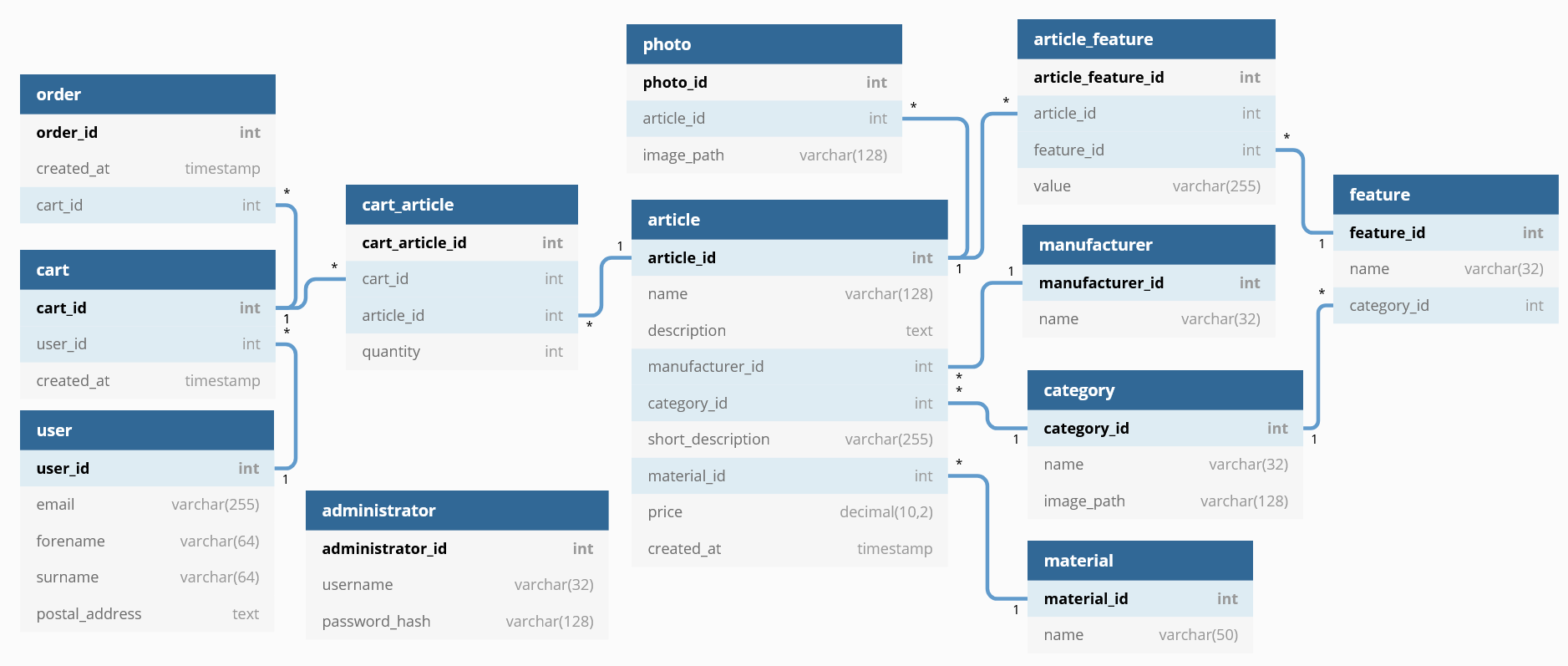
* 1. **Pogodnost za upotrebu**

Sistem je responsiv-e sa intuitivnim dizajnom, kako bi se korisniku omogućilo što prijatnije iskustvo korišćenja aplikacije. Navigacija kroz sistem je jasna i prilagođena prosečnom korisniku. Omogućava upotrebu sa minimalnim brojem grešaka.

* 1. **Zahtevane performanse**

Potrebno je da vreme odziva veb sajta bude ispod 2 sekunde. Očekuje se da u proseku, istovremeno aplikaciju koristi manje od 50 korisnika.

* 1. **Zehtevi baze podataka**



* 1. **Projektna ograničenja**

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. 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) 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.

Izrada projekta mora da bude sprovođena korišćenjem alata za verziranje koda Git, 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. **Sistemske karakteristike projektnog sistema**

Aplikacija može da se pokreće na bilo kojoj platformi koja poseduje veb browser. Bilo da se radi o mobilnom telefonu, laptopu ili računaru. Takođe može da se pokreće na većini operativnih sistema (Windows, MacOS, Android).