****

Tehnički fakultet

Smer: Elektrotehnika i računarstvo

Praktikum iz internet i veb tehnologija

Aplikacija za pretragu recepata

Dokumentacija projekta

Profesor: Vladislav Miškovic Student: Katarina Jakšić, 2018/203803

Asistent: Milan Tair Student: Saša Bogojević, 2018/201681

Beograd, 2019

1. Uvod

Nekada davno, nase bake su skupljale recepte u stare sveske I na taj nacin belezile svoje kulinarske poduhvate. Taj nacin se suocava sa raznim problemima kao sto su brza pretraga recepata, cuvanje velike kolicine istih I slicno. Sagledavsi sve te probleme, pravo resenje je da se ceo proces optimizuje kroz web aplikaciju.

* 1. Cilj razvoja

Glavni cilj razvoja je olaksanje pretrage recepata. Uz pomoc ovog projekta, evidencija recepata kao I pretraga bice efikasnije.

* 1. Obim sistema
* Prijava i odjava administratora na sistem
* Dodavanje novog recepta
* Izmena postojaćeg recepta
* Dodavanje nove kategorije
* Dodavanje novog sastojaka
* Pregled liste svih recepata
* Pregled recepata po kategorijama
* Pregled detaljnih informacija o receptu
* Pretraga recepata po sastojcima(tagovima)

Aplikacija je osmišljena tako da bude zatvorenog tipa i da samo administratori koji su registrovani (koji postoje već u bazi) imaju pravo pristupa glavnom panelu za unos novih recepata, njenih sastojaka i kategorija. Samo prijavljeni korisnici, koji se na portal prijave sa ispravnih parametrima naloga administratora sadržaja mogu da pristupe administrativnom panelu portala. U ovom panelu se mogu dodavati novi recepti koji se sastoje od naslova, slike, detaljnog opisa postupka pripreme, kao i liste sastojaka koji su potrebni za novi recept. Sastojci se sa količinom, formiraju u vidu tagoiva tj. obeležja. Svaki sastojak ako ne postoji u bazi tj. ako ne postoji taj tag, prvo se kreira zapis u adekvatnoj tabeli, pa se onda kreira veza. Sa korisničke strane aplikacije, prikazuje se lista recepta sa slikom i nazivom za recept. Svaki recept pripada jednoj ili više kategorija, što se vrši dodavanjem recepta u bazu podataka od strane administratora. Kada korisnik otvori recept, recept treba da bude prikazan a pregledan način, a lista sastojaka se generiše dinamički na osnovu tagova i količina pripisanih za svaki tag (sastojak). Kada korisnik klikne na neki od tagova u listi, aplikacija treba da ga odvede na listu svih recepata jela u čiji sastav ulazi taj sastojak, tj. za koji je to tag. Treba implementirati i vrstu pretrage gde korisnik može da navede više tagova, tj. sastojaka za koje želi da vrši pretragu baze podataka recepata i aplikacija treba da prikaže samo one recepte za koje su potrebni svi od navedenih sastojaka, tj. sa kojima je vezan svaki od navedenih tagova i to da budu sortirani na taj način da se prikazuju prvo recepti u koje ulazi najveća količina navedenih sastojaka (suma). Grafički interfejs veb sajta treba da bude realizovan sa responsive dizajnom.

* 1. Prikaz proizvoda

Lista recepata će predstavljati sistematizovani prikaz dostupnih recepta, pri čemu će posetilac sajta moći da vrši pregled po različitim kriterijumima, kao što su predjela salate, supe i čorbe, glavna jela, dezerti i ostalo . Recepti će biti organizovani u kategorije, što potencijalnom posetiocu sajta olakšava pretraživanje. Administrator sajta može dodavati nove recepte i vrši izmenu postojećih, čime se obezbjeđuje da lista recepata uviek bude ažurana i da samim tim odgovori na potrebe posetioca sajta - jednostavan i u svakom trenutku dostupan uvid u različite vrste recepata.

* Prijava i odjava administratora na sistem

Na ovaj način se već registrovani adminstrator loguje na sistem unosom svog korisničnog imena i lozinke, na kraju svoje aktivnosti na sajtu ima opciju da se odjavi (izloguje) sa sistema. U bazi se na ovaj način beleži aktivnost svakog administratora u *is\_active* polju, koji je samo *flag* za to da li administrator trenutno jeste ulogovan (1), ili nije (0).

* Dodavanje novog recepta

Administrator ima mogučnost dodavanja novog recepta u bazu. Prilikom popunjavanja forme za ubacivanje novog recepta dodaje naslov, sliku, detaljan opis postupka pripreme, potrebne sastojke, njihovu količinu i odabira jedinicu mere za iste.

* Izmena postojaćeg recepta

Administrator takođe može izmeniti postojeći recept.

* Dodavanje novog sastojaka

Prilikom popunjavanja forme za dodaju novog recepta, administrator može uneti i sastojak koji ne postoji trenutno u bazi, i time dodati naziv novog sastojka u bazu, koji od tog trenitka mogu i ostali koristiti.

* Pregled liste svih recepata

Svaki posetilac sajta može pregledati sve recepte, načine pripreme i njene sastojke.

* Pregled recepata po kategorijama

Posetiocu sajta se nudi mogućnost filtriranja recepata kroz kategorije.

* Pregled detaljnih informacija o receptu

Prilikom razgledanja liste recepata, posetilac sajta ima mogućnost da klikne na željeni recept i uđe u detaljniji opis načina pripreme i ostalim dodatnim informacijama vezanih za isti.

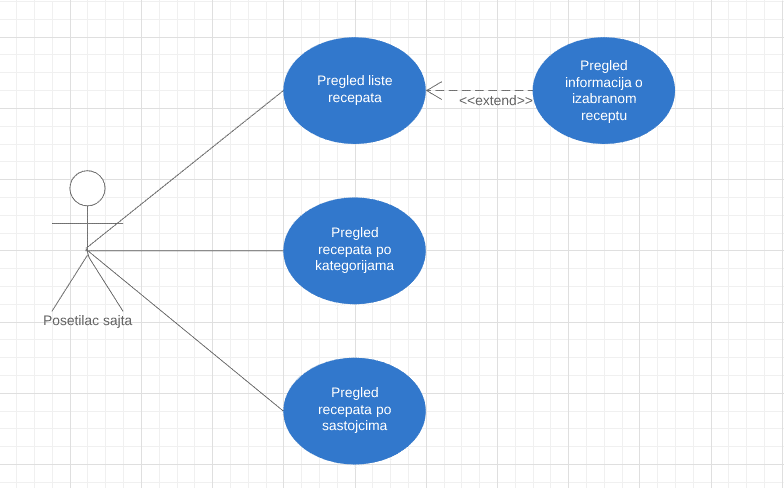
* Pretraga recepata po sastojcima(tagovima)

Postoji i mogućnost pretrage po sastojcima, kada prilikom unosa korisnik označi neke sastojke i za te sastojke se na ekranu ispišu svi recepti u čiju pripremu ulaze ti isti sastojci, sortirani u listu od onog u koji ulazi najveća količina tog sastojka.

1.3.1 Perspektiva proizvoda

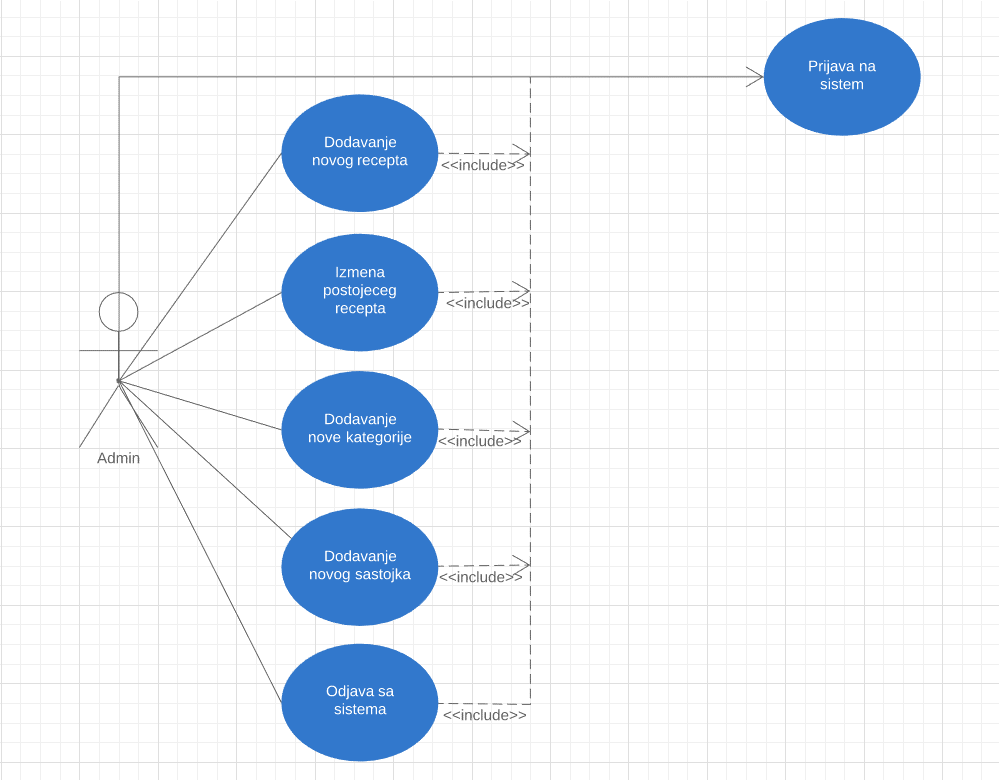
Svrha on-line liste recepata je da jednim klikom bude dostupan i da zameni stare tradicionalne sveske, isečke iz novina i razne druge načine čuvanja recepata koji su naše bake upotrebljavale, i tako ih unapredi da ih učinimo dostupnim široj populaciji. I s tim u poređenju se brže i jednostavnije on-line pretražuju recepti, jer je taj nekada ručni pregled papirnih sadržaja prevaziđena metoda pretraživanja istih.

1.3.2 Funkcije proizvoda



*1.3.2.1 UML dijagram slučajeva korišćenja (Posetilac sajta)*

Posjetilac sajta je u mgućnosti da vrši: pregled liste svih recepata, pregled recepata po kategorijama, pregled recepata po nekom zadatom kriterijumu (kategoriji). Iz pregleda liste svih recepata može da pristupi pregledu informacija o željenom receptu.



*1.3.2.1 UML dijagram slučajeva korišćenja (Administrator sajta)*

Administrator se prvo primarno prijavljuje na sajt sa adekvatim pristupnim parametrima (validnim korisničkom imenom i lozinkom), a potom može da vrši dodavanje ili izmenu recepata i dodavanja sastojaka, kao i da se u bilo kom trenutku odjavi sa sajta.

1.3.3 Karakteristike korisnika

S obzirom da se ova aplikacija izvršava samo na administratorskoj strani sa potpunim prilegijama korišćenja, neophodno je poznavanje rada sa bazom podataka kako bi mogli da manipulišu podacima, kao i da putem HTML formi vrše promenu u sistemu.

Uloge korisnika

U našoj aplikaciji razlikujemo dva tipa korisnika:

* Posetilac sajta
* Administrator sajta

Podrazumijevana uloga korisnika prilikom pristupanja sistemu je Posetilac sajta.

Ukoliko se korisnik uloguje pri unosu korisničkog imena i lozinke, dobija se status Administratora, tako Posetilac sajta zaprvo postaje Prijavljeni korisnik.

1.3.4 Ograničenja

* PHP, CSS, JS, SQL, HTML
* UML (use – case dijagram)
* MVC
* W3C
* GitHub
  1. Definicije
* **XML (Xtensible Markup Language)** predstavlja podatke za opis podataka tj. sintaksu, tekstualnom formatu. On je kreiran sa namjerom da bude jednostavan za učenje, jeftin, brz i optimizovan za Internet. XML je tu da opiše strukturu, integriše protokole između aplikacija, da razmenjuje podatke. XML je skup pravila koja omogućavaju kreiranje tekstualnog formata koji opisuje strukturu podataka. [1]
* **HTML (HyperText Markup Language)** je jezik za označavanje hiperteksta. Hipertekst dokument stvara se pomoću HTML jezika. HTML je standardizovani jezik koji se koristi pri strukturiranju tekstova, medija i ugrađenih objekata u Web stranice i elektronsku poštu, HTML je opisni jezik specijalno namijenjen opisu veb stranica, pomoću koga se jednostavno mogu odvojiti elementi kao što su naslovi, paragrafi, citati i sl. HTML nije programski jezik, njime ne možemo izvršavati nikakve zadatke, čak ni najjednostavniju operaciju sabiranja ili oduzimanja. On služi samo za opis naših hipertekstualnih dokumenata. HTML datoteke su zapravo obične tekstualne datoteke sa ekstenzijom .html. [1]
* **CSS (Cascading Style Sheets)** je jezik formatiranja pomoću kog se definiše izgled elemenata veb stranice. Tri osnovne karakteristike CSS jezika su mogućnost za definisanje klasa za izgled, boje i fontove. Ovi elementi omogućavaju pristupačniji i fleksibilniji sadržaj kao i kontrolu web dizajnera nad određenom grupom HTML elemenata u sadržaju. CSS se, dakle, koristi za definisanje opštih pravila o tome kako se elementi ponašaju unutar Web stranica i kako izgledaju, gdje se nalaze, njihove veličine, transparentnost itd. [1]
* **JavaScript** je kompaktan i objektivno-orijentisan skriptni jezik za razvoj Internet aplikacija, po sistemu klijent-server. Programski kod se upisuje direktno u HTML stranicu i omogućava izradu dinamičkih Web stranica. Jezgro jezika JavaScript podržava brojeve, znakovne nizove i logičke vrijednosti kao osnovne tipove podataka. Najčešće se koristi u pretraživačima Weba (Web browsers), pa se jezgro opšte namjene proširuje objektima koji omogućavaju skriptovima interakciju sa korisnikom, upravljanje pretraživačem Weba i izmjene sadržaja dokumenta koji se pojavljuje unutar prozora čitača. Pošto se JavaScript interpretira umjesto da se prevodi, često se smatra da je to jezik za skriptovanje, a ne pravi programski jezik. [1]
* **DMBS (Data Base Management System)** je softverski sistem koji se koristi za kreiranje, održavanje i manipulisanje podacima, kao i za kontrolu prava pristupa bazi podataka. Osnovni cilj primjene DMBS-a je da se obezbijedi način za smještanje podataka i dobijanje informacija iz baze podataka na najpogodniji i najefikasniji način. Primjenjuje se za upravljanje velikim količinama podataka. [2]
* **SQL (Structured Query Language)** se koristi za pristup i manipulaciju sistemima za upravljanje bazom podataka (DBMS - DataBase Management System), što najčešće podrazumeva čitanje i izmjenu podataka u različitim bazama podataka. SQL je najviše korićeni programski jezik za baze podataka. Za njega kažemo da je deklarativni jezik jer se njegovim rečenicama prvenstveno iskazuje šta se želi učiniti, prepuštajuči DBMS-u da odredi način na koji će se to izvesti. SQL obuhvata unos podataka, upite, ažuriranje i brisanje, šeme kreiranja i mijenjanja, kao i podatke za kontrolu pristupa. [2]
* **PHP (Hypertext Preprocessor)** je skriptni jezik prvobitno dizajniran za pravljenje dinamičkih Web stranica. Primarni način izvršavanja programa napisanih u PHP jeziku je u okviru obrade zahtjeva ka Web serveru, što podrazumijeva postojanje PHP interpretera/izvršioca unutar Web servera. PHP obrađuje zahtjeve uz pomoć Apache servera. Kada korisnik pošalje zahtjev za određenim resursom, zahtjev će preuzeti Apache server. PHP će započeti komunikaciju sa bazom podataka i iščitaće potrebne podatke. Nakon toga, uspostaviće konekciju sa folderima kako bi preuzeo potrebnu strukturu stranice i CSS stilove. Kada preuzme sve što je potrebno, počeće sa obradom i vratiće korisniku željeni prikaz, bez PHP sintakse u izvornom kodu, pošto se sve izvršava na strani servera. [4]
* **UML (Unified Modeling Language)** predstavlja skup grafičkih notacija zasnovanih na jedinstvenom metamodelu. Pod grafičkom notacijom podrazumijeva se skup grafičkih elemenata koji definišu sintaksu jezika. Metamodel je dijagram koji opisuje koncepte jezika za modelovanje. UML je široko prihvaćen u praksi, između ostalog, i zbog toga što se ni grafička notacija, ni metamodel ne moraju strogo primjenjivati. [3]
* **Klijent** - proces ili jedan njegov dio koji zahtijeva uslugu od drugog procesa (bilo na lokalnom ili na udaljenom računaru). [4]
* **Server** - proces koji čeka zahtjeve drugih procesa i isporučuje odgovor na njih. [4]
* **Web/Veb** - najpopularniji servis Internet mreže. [4]

1. Reference

[1] Cvetković D., Marković D., Savanović N., Multimedija, Univerzitet Singidunum, 2017

[2] Veinović M., Šimić G., Jevremović A., Franc I., Baze podataka, Univerzitet Singidunum, 2015

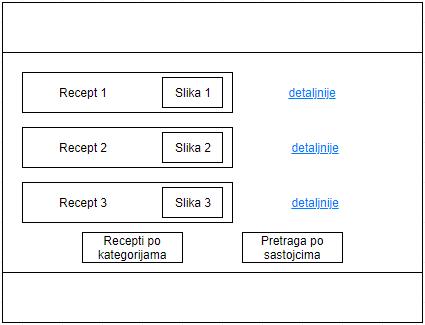
[3] Tomašević V., Razvoj aplikativnog softvera, Univerzitet Singidunum, 2017

[4] Veinović M., Jevremović A., Internet tehnologije, Univerzitet Singidunum, 2017

1. Specifikacija zahteva
   1. Spoljašni interfejsi

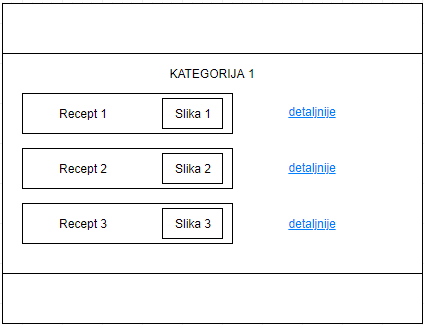
Aplikacija treba da sadrži sledeće komponente, prikazane u obliku skice funkcija koje korisnici obavljaju u sistemu, bilo da su posetioci sajta ili administratori.

Pregled svih recepata (slika, detaljan opis pripreme) – Posetilac sajta ima jasan i detaljan uvid u raspoložive recepte.



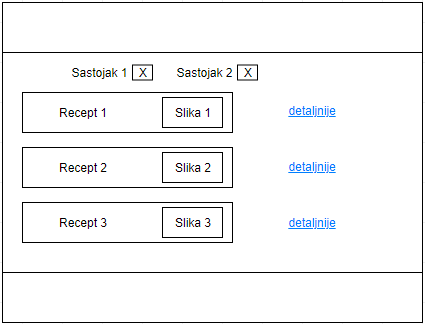
*3.1.1 Pregled svih recepata (slika, opis)*

Pregled antikviteta po kategorijama – Posetilac sajta može kroz kategorije pregledati recepte klikom na određenu kategoriju.



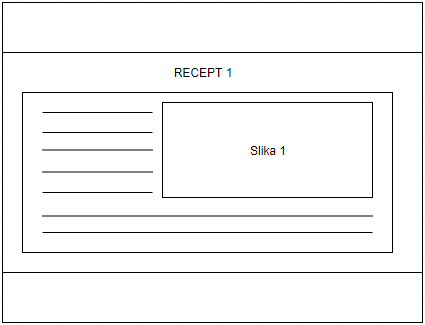
*3.1.2 Pregled recepata po kategorijama*

Pretraga recepata po zadatom sastojcima – Posetilac ima mogućnost da klikom postojeće sastojke pretraži recepte u čiji sastav ulaze označeni sastojci, sortirani u poretku koji recept ima najveću količinu tog sastojka.



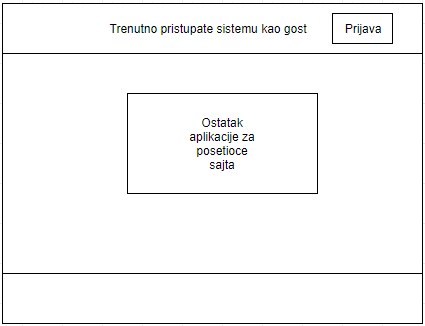
*3.1.3 Pretraga recepata po izabranim sastojcima*

Pregled informacija o pojedinačnom receptu – Posetiocu sajta se na ovoj stranici prikazuju detaljne informacije o receptu na koji je prethodno kliknuo.

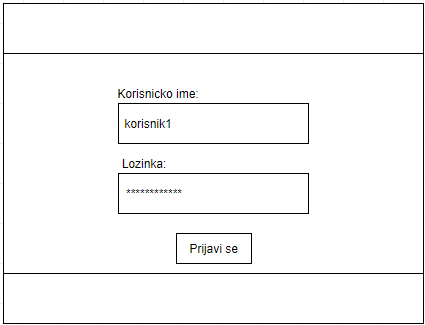


*3.1.4 Pregled informacija o pojedinačnom receptu*

Prijava na sajt sa validnim parametrima – Korisnik se prijavljuje na sistem unošenjem validnih parametara za pristup administrativnom panelu.

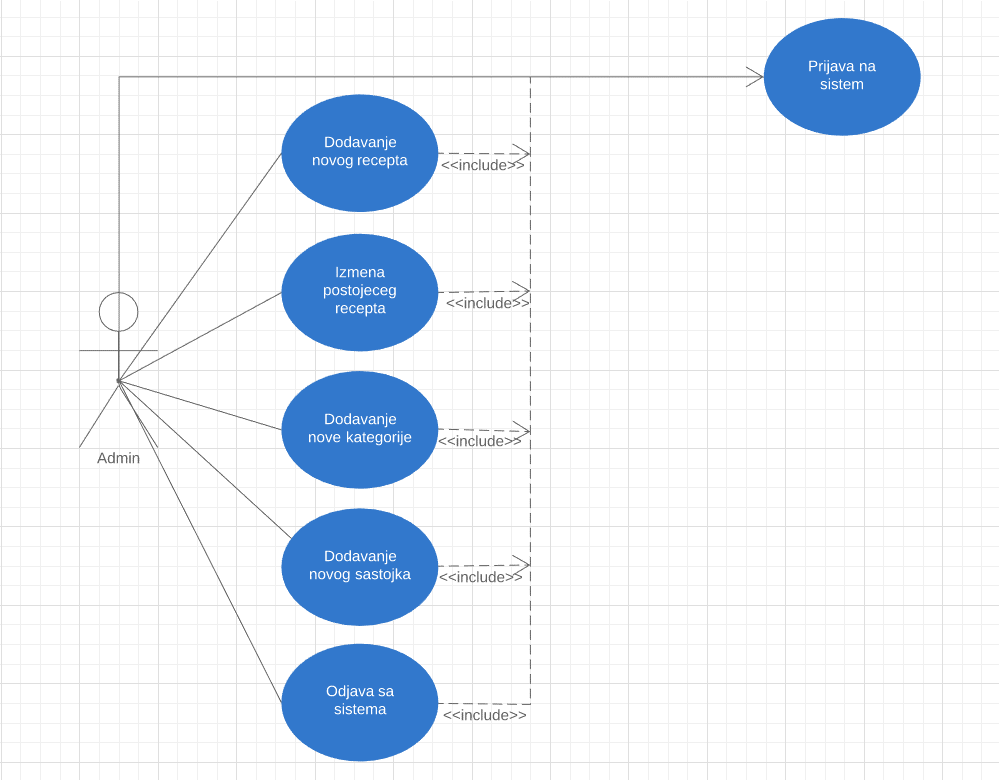
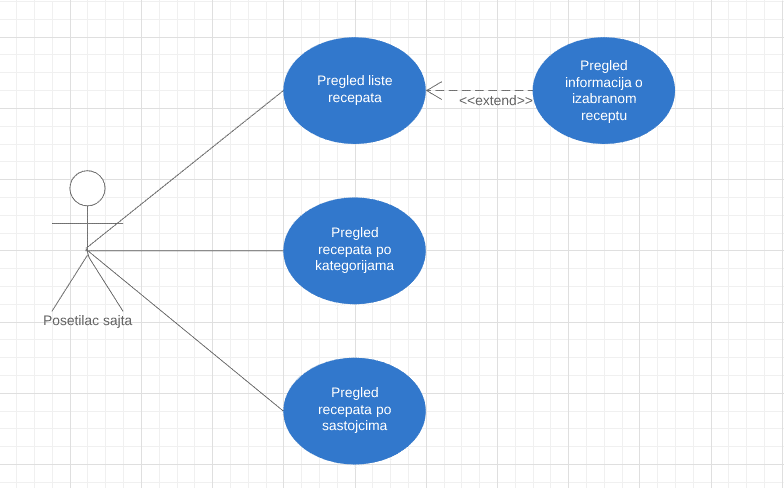


*3.1.5 Izgled sajta pre pristupanja administrativnom panelu*



*3.1.5 Izgled sajta prilikom prijave na sistem*

* 1. Funkcije sistema



*3.2.2 UML dijagram slučajeva korišćenja (Posetilac sajta i Administrator)*

Posjetilac sajta - može da pregleda listu svih dostupnih antikviteta. Treba da mu bude omogućeno da pretražuje antikvitete po kategorijama ili po nekom drugom zadatom kriterijumu (npr. po cijeni ili državi porijekla), kao i da eventualno mijenja mehanizam za sortiranje antikviteta. Kada korisnik vrši pregled liste antikviteta, može da pristupi pregledu informacija o konkretnom, tj. izabranom antikvitetu (extend zavisnost). Nadalje, kada korisnik ima pristup informacija o odabranom antikvitetu, može i da pogleda adresu na kojoj je antikvitet dostupan na prodaju ili na uvid (extend zavisnost).

Administrator - mogao da pristupi administrativnom panelu portala, neophodno je da se prvo prijavi na sajt sa ispravnim parametrima svog naloga. Dakle, da bi administrator mogao da izvršava bilo koju drugu aktivnost na sajtu, prvobitno mora da se prijavi, što znači da sve druge aktivnosti koriste prijavu kao svoje polazište (include zavisnost). Nakon uspješne prijave, administrator može da mijenja postojeće ili dodaje nove antikvitete/kategorije. Nakon izvršenih aktivnosti, odjavljuje se sa sajta.

* 1. Pogodnost za upotrebu

Aplikacija koja se realizuje treba svojim korisnicima da omogući brz i jednostavan pregled i pretragu antikviteta po različitim kriterijumima, kao i realizaciju sortiranja u odgovarajućem poretku. Ovakav katalog antikviteta treba da pruži pregledan popis svih informacija o dostupnim antikvitetima, bilo da su u pitanju podaci o mogućnosti kupovine ili o dostupnosti antikviteta na uvid.

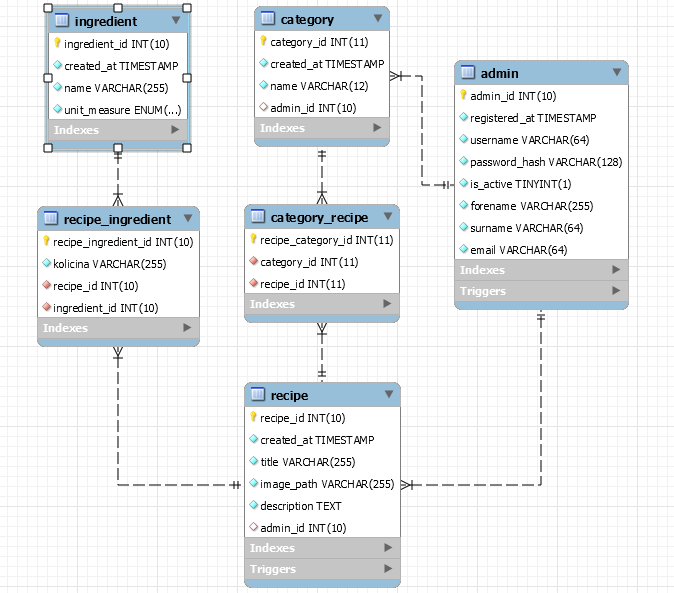
* 1. Zahtevne performance

Aplikacija treba da garantuje brz odziv na komande posjetilaca sajta i da podrži nesmetano pregledanje kataloga antikviteta od strane većeg broja korisnika koji pristupaju sajtu.

* 1. Zahtevi baze podataka

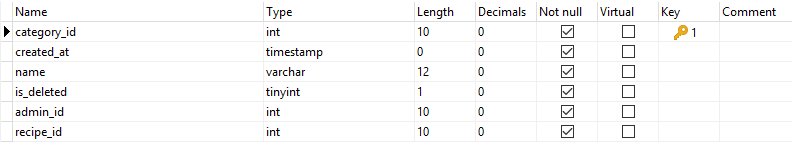
Sistem za pretragu recepata sačinjava baza podataka u kojoj je uvezano 6 tabela:

* Category
* Admin
* Recipe
* Ingridient
* Recipe-ingredient
* Category-recipe

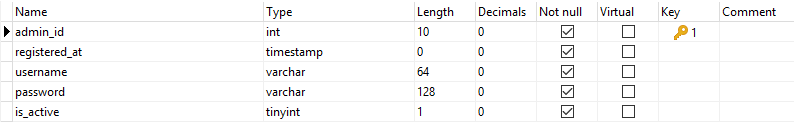


*3.5 Prikaz baze podataka u MySQL Workbench-u*

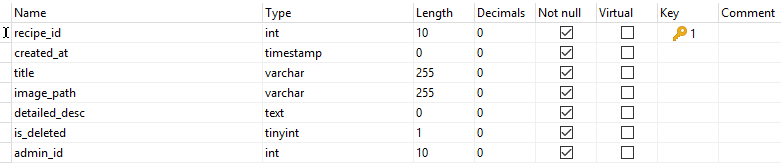
**category** - sadrži podatke o kategorijama u koje će se smeštati recepti. Sastoji se iz polja category\_id koji predstavlja indentifikaciju svake kategorije posebno, created\_at datum i vreme kada je ta kategorija uneta u bazu, name kao ime kategorije u kojoj se recepti nalaze, is\_deleted je flag koji se postavlja na 1 ukoliko je kategorija obrisana. Admin\_id i recipe\_id su strani ključevi. Tabela category je povezana sa tabelom Admin i sa tabelom Recipe. Jednom adminu je dozvoljeno da napravi više kategorija, a u jednoj kategoriji može postojati više recepata kao i obrnuto – jedan recept može pripadati više kategorija.



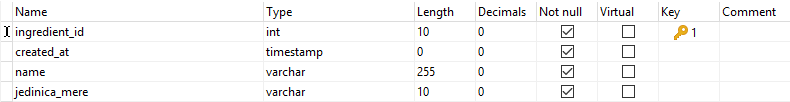
**admin** – sadrži prikaz podataka o administratorima sajta. Tabela admin ima polje ya indentifikaciju admina, jedinstveni admin\_id, takođe sadrži vreme kreiranja datog administratora u polju registered\_at. Username I password hash za jedinstvene parameter prilikom logovanja administratora, kao i is\_active flag koji označava aktivnost datog administratora u datom trenutku.



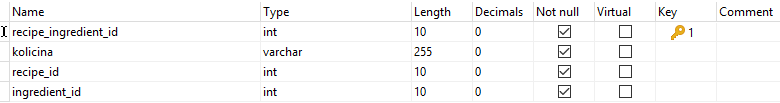
**recipe** - sadrži sve neophodne podatke o svakom receptu. Recipe\_id je polje za jedinstveni broj recepta, created\_at vreme i datum kreiranog recepta, title predstavlja naslov određenog recepta, image\_path putanju do slike za svaki recept ponaosob, detailed\_desc je detaljan opis postupka pripreme recepta. Flag is\_deleted takođe i prikazuje obrisane recepte (koji više nisu vidljivi na sajtu), a ako su postavljeni na 1. I admin\_id kao polje za strani ključ, nasleđen iz tabele Admin.



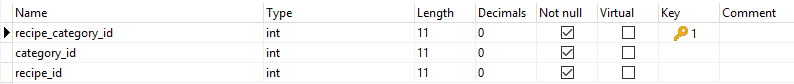
**ingredient** - sadrži sve neophodne podatke o svakom sastojku. Ova tabele sa sastoji od četiri polja, a to su ingredient\_id kao jedinstvena oznaka svakog sastojka posebno, created\_at kao vremenski pokazatelj kada je određeni sastojak unet u bazu. Name označava naziv sastojka, a jedinica\_mere označava da li se radi o komadima nekog sastojka, gramima ili decilitrima.



**recipe-ingredient** - prikazuje vezu izmedju sastojaka i recepata. Ova tabela je vezna tabela koja ima dva nasleđena ključa od tabela Recipe i Ingredient, kao i količinu određenog sastojka koji se nalazi u određenom receptu. Takođe kao posebno polje ima svoj id – recipe\_ingredient\_id.



**category-recipe -** prikazuje vezu izmedju kategorije i recepata. Ova tabela je vezna tabela koja ima dva nasleđena ključa od tabela Category i Recipe.Takođe kao posebno polje ima svoj id – category\_recipe\_id.



* 1. Projektna ograničenja

Aplikacija mora da bude realizovana na PHP jeziku i sav kod aplikacije treba da bude organizovan prema pravilima MVC arhitekture. Sav generisani HTML kod koji proizvodi aplikacija mora da bude 100% validan, tj. da generisani kôd prođe provjeru W3C Validatorom (dopuštena su upozorenja, ali ne i greške).

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; (3) PHP provera korišćenjem adekvatnih testova ili korišćenjem regularnih izraza na strani servera 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 deteljnim opisom svih tabela, njihovih polja i relacija; (2) dijagram organizacije elemenata aplikacije sa akcentom na isticanje MVC arhitekture na konkretnom primjeru jednog odabranog zahtjeva, tj. rute 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.

Izrada projekta mora da bude sprovođena korišćenjem alata za verziranje koda, tj. Git alata, a da kompletan kôd aplikacije bude dostupan na javnom Git repozitorijumu, npr. na besplatnim GitHub ili Bitbucket servisima.

Za razvoj aplikacije odabrana su sledeća okruženja: Visual Studio Code za pisanje programskog koda i Navicat za izradu baze podataka.