Objektno oblikovanje

Seminarski rad

|  |  |
| --- | --- |
| ČLANOVI TIMA | „Lovro i prijatelji“ |
| ČLANOVI GRUPE | KLARA BANIĆ, LOVRO RAČKI, LEON SULJIĆ |
| SAžETAK SEMINARA | APLIKACIJA ZA TRAŽENJE RECEPATA |
| GIT REPOZITORIJ | <HTTPS://GITHUB.COM/RACKILOVRO/SEMINAROBJEKTNO> |

sadržaj

[UVOD 3](#_Toc62906210)

[ZAHTJEVI 4](#_Toc62906211)

[OPIS MODELA 8](#_Toc62906212)

[OPIS IMPLEMENTACIJE PERZISTENCIJE 9](#_Toc62906213)

[DESKTOP APLIKACIJA 10](#_Toc62906214)

[Uvod 10](#_Toc62906215)

[Screenshots 11](#_Toc62906216)

[WEB APLIKACIJA 15](#_Toc62906217)

# UVOD

Odrasli ljudi, ali pogotovo studenti dnevno muče muku što skuhati za doručak, ručak ili večeru, uz određeni budžet i limitirano vrijeme. U današnje vrijeme postoji puno dostupnih aplikacija koje će na temelju upisa naziva recepta vratiti cijeli recept i postupak pripreme određenog jela. Ipak, to nije dovoljno s obzirom da većina nas nije čulo za 90% jela pa nam tražilica po nazivu recepta pretjerano i ne pomaže.

Ideja za aplikaciju je došla upravo od te činjenice. U hladnjaku imamo desetak namirnica, gledamo kako bismo ih mogli povezati i pomoću njih napraviti neko jelo. Nažalost, od prekomjernog učenja i izrade razno raznih projekata, mozak nam je prazan i nije spreman za kreativno povezivanje i izmišljanje recepata. Dovoljni je napor sama priprema namirnica i kuhanje, na koje se moramo odvažiti radi studentskog budžeta.

*RecepTour* je sve što jedan student, ali i zaposleni čovjek treba. *RecepTour* je jednostavna aplikacija, dostupna u web i desktop verziji, u koju korisnik unosi raspoložive namirnice te na temelju toga dobije popis recepata koje može ostvariti koristeći te sastojke. Korisnik zatim može odabrati recept koji mu se najviše sviđa te pogledati detalje recepta i sam postupak pripreme. Svidi mu li se ovaj recept i želi li ga možda ponoviti koji put, korisnik može dodati recept pod omiljene recepte. Kako bi to ostvario, korisnik se najprije treba registrirati kako bi imao svoj račun uz koji bi se svi podaci vezali. Tako prijavljeni korisnik može upravljati svojim omiljenim receptima. Nadalje, prijavljeni korisnik može unijeti svoje recepte u aplikaciju, a nakon toga i upravljati istima. Aplikacija ima dostupne još dvije tražilice – pomoću jedne korisnik može pretraživati recepte po nazivu, a pomoću druge recepte određenog korisnika.

Web aplikacija se može smatrati proširenjem desktop aplikacije koja predstavlja osnovne funkcionalnosti, a to je pretraga recepata. Web aplikacija dodatno pruža mogućnost registracije, prijave, dodavanje vlastitih recepata, omiljenih recepata i upravljanje svojim korisničkim računom.

Aplikacija je oličenje minimalizma i jednostavnosti kako ne bi zbunila korisnike i maksimalno olakšala njenu uporabu. Idealna je za korisnike kojima su prioriteti jednostavnost, organiziranost i pedantnost.

# ZAHTJEVI

Skup mogućnosti koje naš sustav pruža je najlakše prikazati koristeći user story-e.

Kako bi korisnici mogli koristiti sve funkcionalnosti aplikacije najprije će se morati registrirati te potom prijaviti u aplikaciju, iako to nije nužno za osnovno korištenje aplikacije.

Desktop verzija aplikacije predstavlja pojednostavljenu verziju aplikacije bez funkcionalnosti prijavljivanja korisnika u sustav te je namijenjena korisnicima koji imaju računala u blizini kuhinje i žele gledati recept za vrijeme kuhanja. Ako žele dodati svoj recept, to će napraviti kada završe sa kuhanjem i na miru sjesti i prijaviti se u web aplikaciju.

Pomoću jednostavnog izbornika korisnik određuje namirnice koje trenutno ima na raspolaganju, na temelju kojih mu aplikacija nudi prijedloge za recepte koje korisnik može skuhati. Ako se korisnicima nakon kuhanja svidio recept, mogu ga spremiti pod svoje omiljene recepte. Ako korisnici nisu zadovoljni niti jednim pronađenim receptom, mogu objavljivati svoje recepte, ali samo ako su registrirani i prijavljeni u sustav. Svi zahtjevi su dani i opisani u nastavku:

1. Registracija korisnika:

Korisnik se želi registrirati pa odabire opciju za registraciju, unosi potrebne podatke i potvrđuje svoju odluku.

1. Login u sustav:

Korisnik želi pristupiti aplikaciji preko vlastitog korisničkog računa pa unosi potrebne podatke za prijavu i ulazi u aplikaciju preko korisničkog računa.

1. Promjena lozinke:

Korisnik ima potrebu da promijeni svoju postojeću lozinku. Odabire opciju za promjenu lozinke, unosi potrebne podatke i, ako su podatci ispravni, lozinka se mijenja.

1. Promjena podataka o korisniku:

Korisnik želi promijeniti neke svoje podatke. Odabire opciju za promjenu podataka i unosi nove, ažurne podatke. Ukoliko je zadovoljan, potvrđuje svoju odluku i sprema promjene klikom na gumb za potvrdu.

1. Dodavanje recepta pod „Favorites“:

Korisniku se sviđa određeni recept i želi ga spremiti kako bi ga mogao pregledavati i kasnije. To čini lagano, klikom na gumb za spremanje pod „Favorites“ koji se odnosi na pojedini recept.

1. Micanje recepta iz „Favorites“:

Korisniku se više ne sviđa recept pa ga želi maknuti iz „Favorites“ kategorije. To radi klikom na gumb koji to radi, recept se miče iz kategorije „Favorites“, a korisnik može odmah nastaviti s radom.

1. Dodavanje svojih recepata:

Korisnik želi dodati vlastiti recept u aplikaciju. Odabire opciju za dodavanje recepta i unosi sve podatke koji su potrebni za unos recepta. Nakon spremanja, recept je dodan u aplikaciju i ostali korisnici ga mogu vidjeti i pristupiti mu.

1. Brisanje vlastitog recepta:

Korisniku se više ne sviđa recept koji je dodao pa ga želi maknuti iz aplikacije. To čini lagano klikom na gumb za brisanje kod pregleda recepta i recept se odmah briše iz aplikacije.

1. Uređivanje recepta:

Korisnik treba promijeniti neke podatke vezane za svoj recept pa odabire opciju za mijenjanje podataka i unosi nove. Gumbom za spremanje potvrđuje odluku.

1. Pregledavanje recepata:

Korisnik pregledava sve recepte koji se nalaze u aplikaciji.

1. Pretraživanje recepata po sastojcima:

Korisnik označava sastojke koje želi pronaći u receptu što rezultira time da aplikacija prikaže recepte koji sadrže takve sastojke. Nakon toga korisnik može pojedinačno pregledati recept koji mu se sviđa.

1. Pretraživanje recepta po imenu:

Korisnik se sjetio imena recepta kojeg želi naći i unosi njegovo ime u tražilicu te mu aplikacija prikazuje recept koji sadrži to ime, ukoliko takav recept postoji u aplikaciji.

1. Pretraživanje recepta po autoru:

Korisnik se sjetio osobe koja je napravila recept koji on traži ili želi vidjeti recepte svoga prijatelja. Unosi njegovo ime u tražilicu prema autoru i aplikacija prikazuje recepte koje je ta osoba unijela u aplikaciju.

1. Detaljnije prikazivanje recepta:

Korisnik želi više toga saznati o receptu i klikne na njega kako bi vidio detaljnije informacije o njemu, kao npr. potpuni tekst recepta.

1. Prikaz korisničkog profila:

Korisnik želi vidjeti svoj korisnički profil. To ostvari tako da klikne na svoje ime u kutu i pregledava svoj korisnički profil.

Zahtjevi s obzirom na sadržaj zahtjeve bi mogli klasificirati na funkcionalne, nefunkcionalne zahtjeve i zahtjeve domene primjene.

Funkcionalni zahtjevi su izjave o uslugama koje sustav mora pružati, kako će sustav reagirati na određeni ulazni poticaj te kako bi se sustav trebao ponašati u određenim situacijama. Ove zahtjeve možemo lako identificirati tako da si pomognemo sa „*system shall do <requirement>“*.

Često uz ovo vežemo dionike, aktore i njihove funkcionalne zahtjeve. Dionici u ovom projektu su korisnici aplikacije i razvojni tim. Problem kod zahtjeva je u tome što dionici ne znaju što stvarno žele i izražavaju zahtjeve na različite načine. Različiti dionici mogu imati konflikte zahtjeve (moram priznati da ih je bilo 😊), a zahtjevi su se mijenjali za vrijeme procesa analize zahtjeva, ali i kasnije. U nastavku su kratko i jasno, oku ugodno, pobrojani aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:

a. pregledavati recepte

b. pretraživati recepte po sastojcima, imenu ili autoru

c. pristupiti detaljnom prikazu recepta

d. se registrirati u sustav, stvoriti novi korisnički račun za koji su mu potrebni korisničko ime, e-mail i lozinka

2. Registrirani korisnik (inicijator) može:

a. se ulogirati u sustav sa svojim korisničkim podacima

b. pregledavati i mijenjati osobne podatke

d. dodati, izbrisati i promijeniti vlastite recepte

e. pregledavati recepte

f. pretraživati recepte po sastojcima, imenu ili autoru

g. pristupiti detaljnom prikazu recepta

h. dodavati recepte pod „Favorites“ i ukloniti iste

i. pregledavati profile drugih korisnika

3. Administrator (inicijator) može:

a. vidjeti popis svih registriranih korisnika i njihovih osobnih podataka

b. brisati korisnike

c. brisati recepte

4. Baza podataka (sudionik):

a. pohranjuje sve podatke o korisnicima i njihovim ovlastima

b. pohranjuje sve podatke o receptima, sastojcima i drugim detaljima

Nefunkcionalni zahtjevi su ograničenja u uslugama i funkcijama, kao što su vremenska ograničenja, standardi, ograničenja u procesu razvoja i oblikovanja i slično. Naglasak je na karakteristikama sa *„system shall be <requirement>“*.

Naš sustav zadovoljava većinu nefunkcionalnih zahtjeva, a kada bismo htjeli zadovoljiti sve nefunkcionalne zahtjeve, ipak bismo trebali uložiti više novaca i više napora. 😊

Zahtjevi domene primjene su funkcionalni i nefunkcionalni zahtjevi koji proizlaze iz domene primjene sustava, kao i oni koji karakteriziraju tu domenu.

U nastavku je opisano kako se sustav treba ponašati i koja su neka njegova ograničenja – formalni nazivi za ove zahtjeve su nefunkcionalni zahtjevi i zahtjevi domene primjene.

1. Sustav treba omogućiti rad više korisnika u stvarnom vremenu.
2. Jezik korisničkog sučelja i jezik u programskom kodu je engleski
3. Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
4. Vrijeme potrebno da se dohvati neki podatak iz baze podataka treba biti relativno kratko (ne više od nekoliko sekundi)
5. Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
6. Sustav treba biti jednostavan za korištenje, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
7. Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške
8. Pristup sustavu preko javne mreže je zaštićen pomoću HTTPS protokola
9. Koristi se objektno orijentirana paradigma, a kao posljedica toga je korištenje C# u sklopu *backend*-a

# OPIS MODELA

# OPIS IMPLEMENTACIJE PERZISTENCIJE

# DESKTOP APLIKACIJA

## Uvod

Desktop aplikacija implementirana je pomoću *windows presentation frameworka (WPF)*. WPF koristi *model-view-viewmodel (MVVM)* oblikovni obrazac. Ključan koncept *MVVM* oblikovnog obrasca je razdvajanje rukovanja prezentacijskim sadržajem i podacima te njihove prezentacije.  
Model domene definira se u *model* dijelu obrasca, koji je u *RecepTour desktop* aplikaciji *(DesktopTour)* ostvaren i *O/R* mapiranjem preslikan pomoću *entityFrameworka* iz baze podataka u projekt.   
*Viewmodel* dio *MVVM* obrasca zadužen je za dohvat i pripremu podataka koji su potrebni *View* dijelu obrasca. U *DesktopTouru*, *Viewmodel* je implementiran kroz nekoliko jednostavnih razreda, koji nemaju previše odgovornosti jer je *DesktopTour* namijenjen samo za pregledavanje sadržaja.  
*View* dio *MVVM* oblikovnog obrasca namijenjen je za prezentaciju sadržaja. Prilikom inicijalizacije *View* dijela oblikovnog obrasca postavlja se *context* razreda u kojemu se nalaze svi potrebni podaci za prikaz. Ključno je istaknuti da *code-behind* dio *WPF view* razreda ne dodaje niti mijenja podatke koji su mu u *context* postavljeni prilikom inicijalizacije na bilo koji način, u skladu sa razdvajanjem odgovornosti *MVVM* oblikovnog obrasca. U razredu *view* definiran je izgled i prikaz podataka sintaksom sličnom *HTMLU*.  
Kako bi se *View* razred povezao sa konkretnim podacima iz modela, *View* u svojim konstruktorima ili metodama stvara novi primjerak razreda *Viewmodel*, koji u svom konstruktoru iz repozitorija povlači potrebne podatke, zatim se kao *context* *View* razreda postavlja novostvoreni *Viewmodel*.

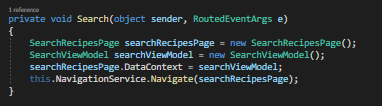


Figure 1 Primjer povezivanja VIewmodela i viewa preko DataContext svojstva

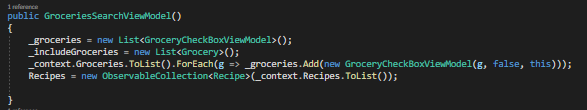


Figure 2 Dohvaćanje svih podataka modela u konstruktoru razreda Viewmodel

## Screenshots

Nakon pokretanja aplikacije korisnici dolaze na početnu stranicu gdje imaju mogućnost odabira između pretraživanja recepata po jednom od tri kriterija, ili pregleda svih recepata koji se nalaze u aplikaciji.

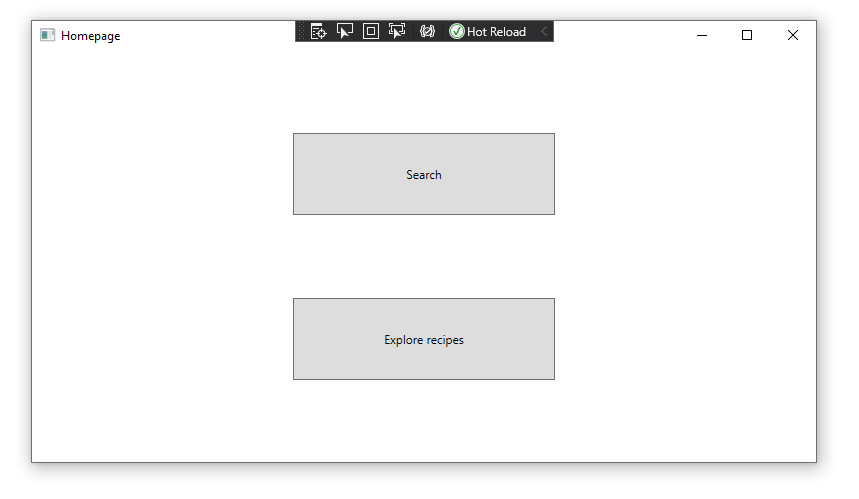


Figure 3 Početna stranica

Odabirom mogućnosti pretraživanja recepata po kriterijima korisnici se preusmjeravaju na stranicu pretraživanja recepata po sastojcima. U početku nijedna namirnica nije označena te se prikazuju svi recepti u bazi podataka. Kako bi korisnik pretražio bazu podataka po receptima koji sadrže neke sastojke mora označiti koje sastojke za koje želi doznati recepte, lista recepata se osvježava pri svakom označavanju ili odznačavanju sastojka.

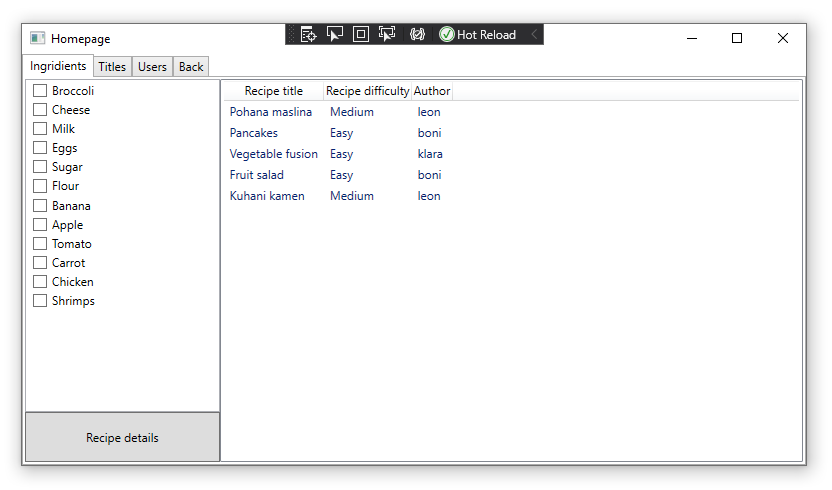


Figure 4 Pretraživanje recepata po sastojcima

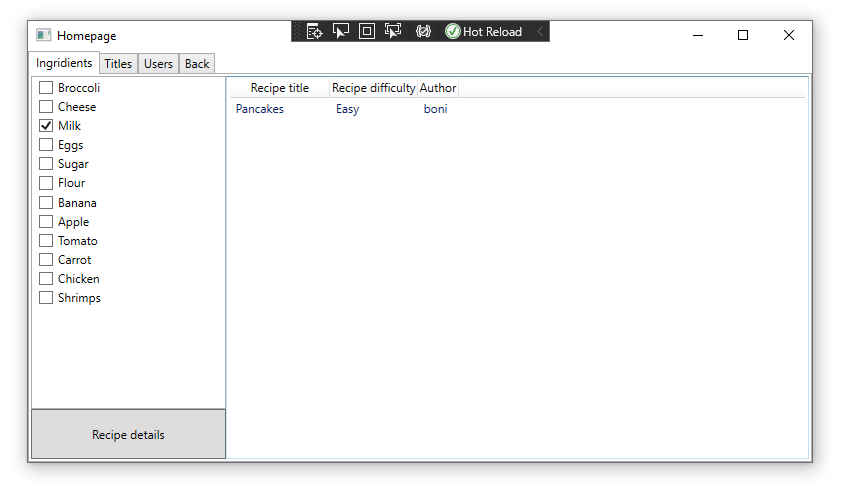


Figure 5 Recepti koji pri izradi zahtijevaju jaja

Ako korisnik želi doznati više informacija o nekom receptu najprije mora odabrati recept koji ga zanima iz liste recepata koja se nalazi na desnoj strani prozora te potom pritisnuti gumb *Recipe details*. Nakon pritiska na gumb korisnici su preusmjereni na prozor gdje se nalaze detalji o receptu poput težine izvedbe, slika, popisa sastojka i njihove količine.

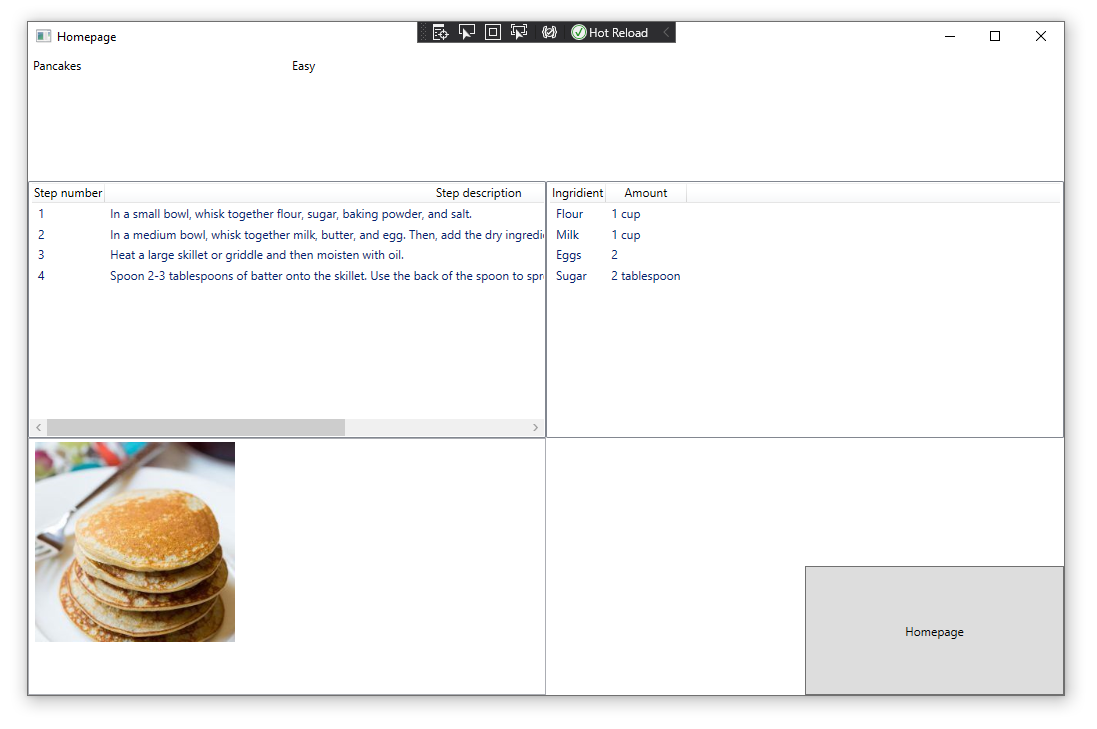


Figure 6 Detalji recepta za palačinke

Pritiskom na gumb *Homepage* korisnici se vraćaju na početnu stranicu.

Ako korisnici žele pretraživati recepte po naslovu najprije moraju kliknuti na karticu *Titles*, potom upisati naslov recepta koji ih zanima te naposljetku pritisnuti gumb *Search by title*.  
Sličan je pristup i pretraživanju recepta po autoru, samo što korisnici umjesto naslova recepta upisuju ime autora.

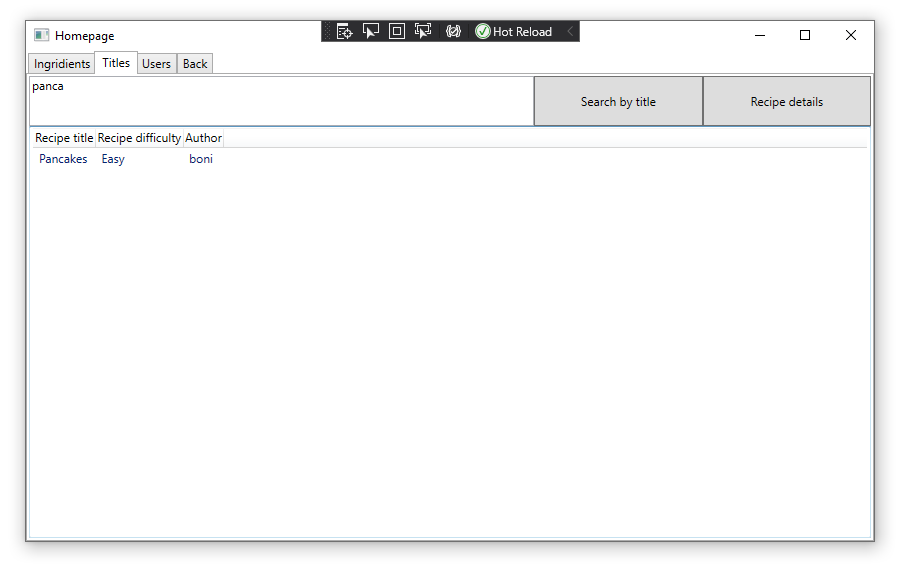


Figure 7 Pretraživanje recepta koji u naslovi sadrže „panca“

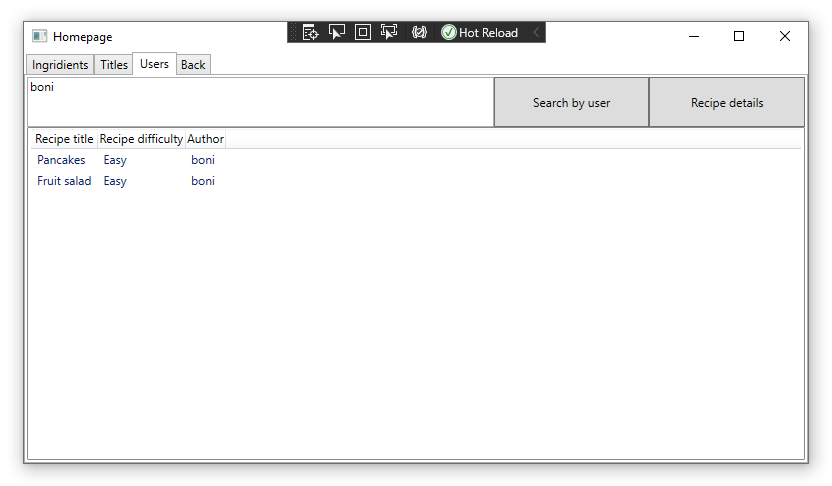


Figure 8 Pretraživanje recepta korisnika "boni"

Kako bi se vratio na početnu stranicu korisnik odabire karticu *Back* i pritiskom na gumb *Back to homepage* vraća se na početnu stranicu.

Ako korisnici žele pregledavati sve recepte u bazi podataka odabiru gumb *Explore recipes*, nakon čega dolaze na stranicu na kojoj se nalaze svi recepti u bazi podataka.

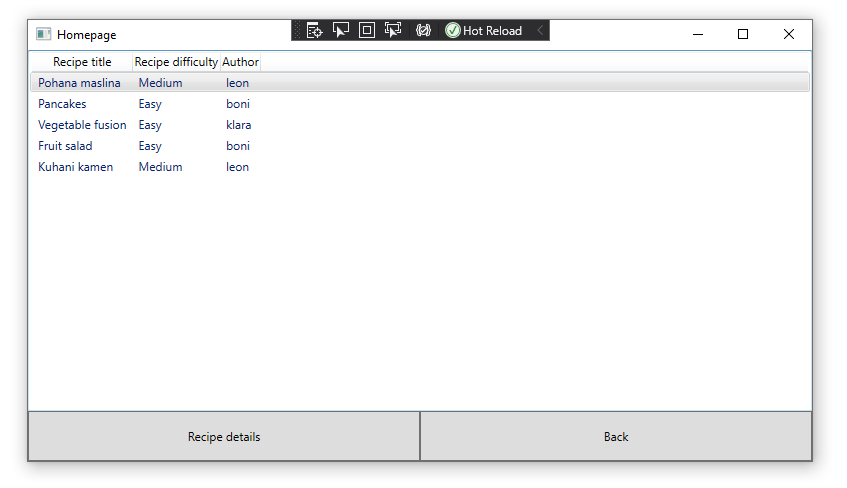


Figure 9 Pregledavanje svih recepata u bazi podataka

# WEB APLIKACIJA

## Uvod

Web aplikacija je implementirana pomoću ASP.NET MVC (*Model-View-Controller*) oblikovnog obrasca koji je korišten za razdvajanje korisničkog sučelja (*View*), podataka (*Model*) I aplikacijske logike (*Controller*). Model domene je ovdje također definiran u *model* dijelu obrasca, a preslikan je *O/R* mapiranjem preslikan pomoću *entityFrameworka* iz baze podataka u projekt.

dodaj stvari

## Screenshots

Nakon pokretanja aplikacije korisnici