

Dokumentacija

Projekat iz predmenta OOAD

Grupa: 10

Tim: TimSMTH

Demonstrator: Jasmina Bajramović

Članovi:

- Mašović Haris
- Salihić Samra
- Šabović Dženana

Sadržaj

Opis teme	3
Funkcionalnosti	3
Dizajn projekta (dijagrami).....	4
Use Case dijagrami	4
Dijagram klasa (MVVM)	7
Dijagrami aktivnosti	8
Dijagrami sekvenci i komunikacije	12
Dijagrami komponenti, paketa i raspoređivanja	15
Izvještaj o radu	16
Tabela očekivanih i implementiranih funkcionalnosti	19

Opis teme

Vicinator¹ je aplikacija koja omogućava korisniku (user-u) da na jednostavan način pronađe mjesto/objekat koje servira/prodaje hranu kao npr. fast food/restorani itd. pri čemu korisnik bira objekat koji mu se sviđa iz liste ponuđenih objekata.

Često korisnik dolazi u situaciju gdje ne može odlučiti gdje jesti, te pomoću ove aplikacije ima izbor restorana/fast food-ova (generalno food places) u određenom radijusu u odnosu na trenutnu lokaciju. Korisnik može izabrati određeni restoran i/ili dodati u listu svojih omiljenih restorana ponuđene restorane od strane aplikacije ili restorane u odnosu na trenutnu lokaciju definisane određenim radijusom kojeg je korisnik zadao.

Nakon toga aplikacija daje uputstva kako doći na određenu lokaciju koju je korisnik izabrao.

Funkcionalnosti

Korisnik:

- Mogućnost izbora/promjene radijusa pretrage restorana u odnosu na lokaciju korisnika
- Mogućnost pregleda restorana u vidu liste
- Mogućnost izbora restorana
- Mogućnost dobijanja uputa do odabranog restorana
- Mogućnost dobijanja broja za poziv restorana/objekta, prilikom izbora restorana
- Mogućnost izbora iz skupa restorana preporučenih od strane aplikacije

Registrovani korisnik:

- Mogućnost uređivanja korisničkog profila
- Mogućnost pravljenja liste želja (lista omiljenih restorana)

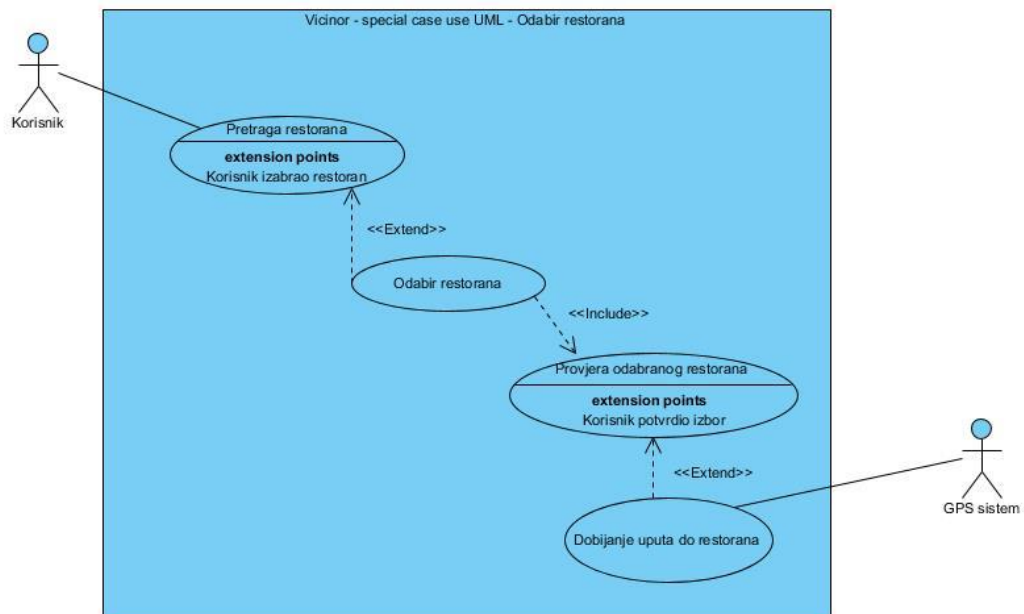
Neregistrovani korisnik:

- Mogućnost registracije

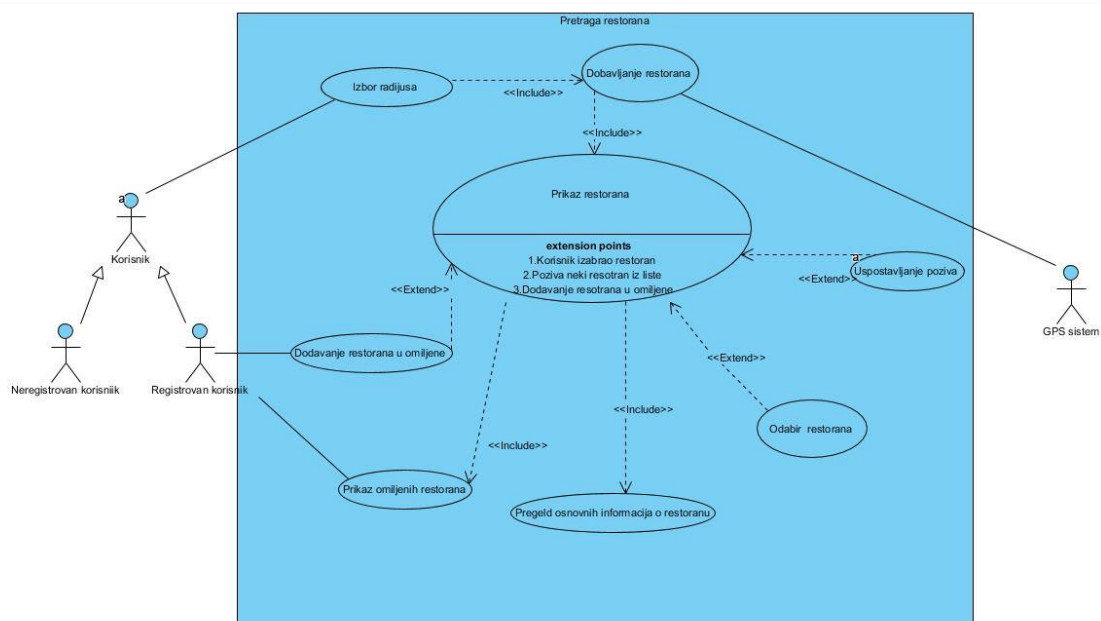
Administrator:

- Prikaz statistike
- Praćenje registrovanih korisnika sistema te banovanje korisnika po potrebi
- Uređivanje profila

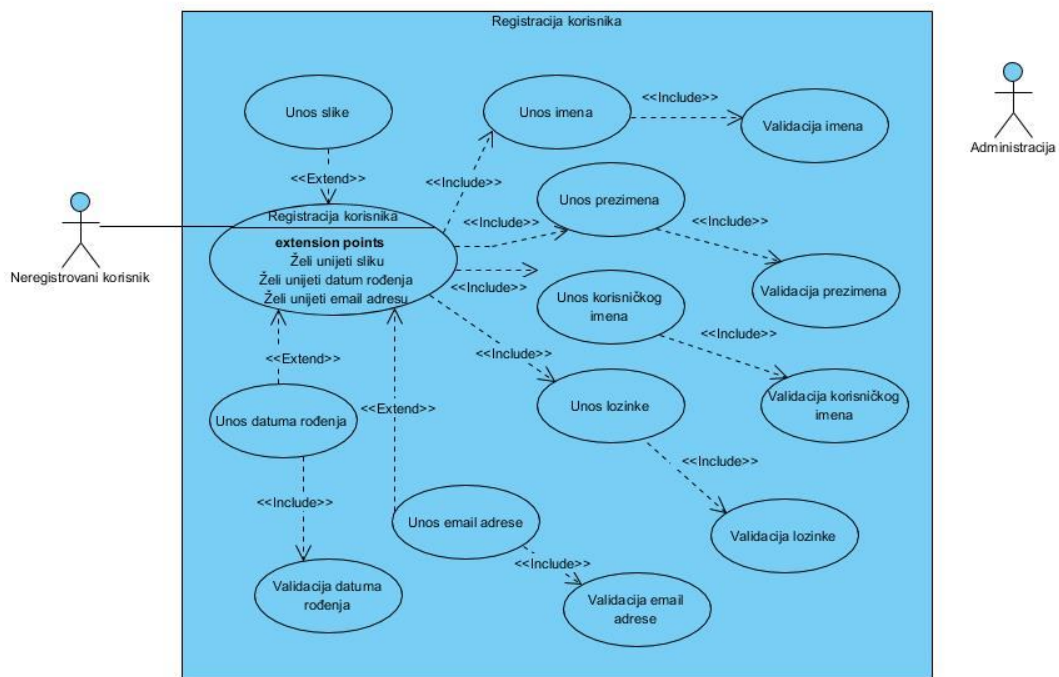
¹ vicino - ital. znači nearby/close - blizu, a r kao restorani).



Slika 2. Odabir restorana

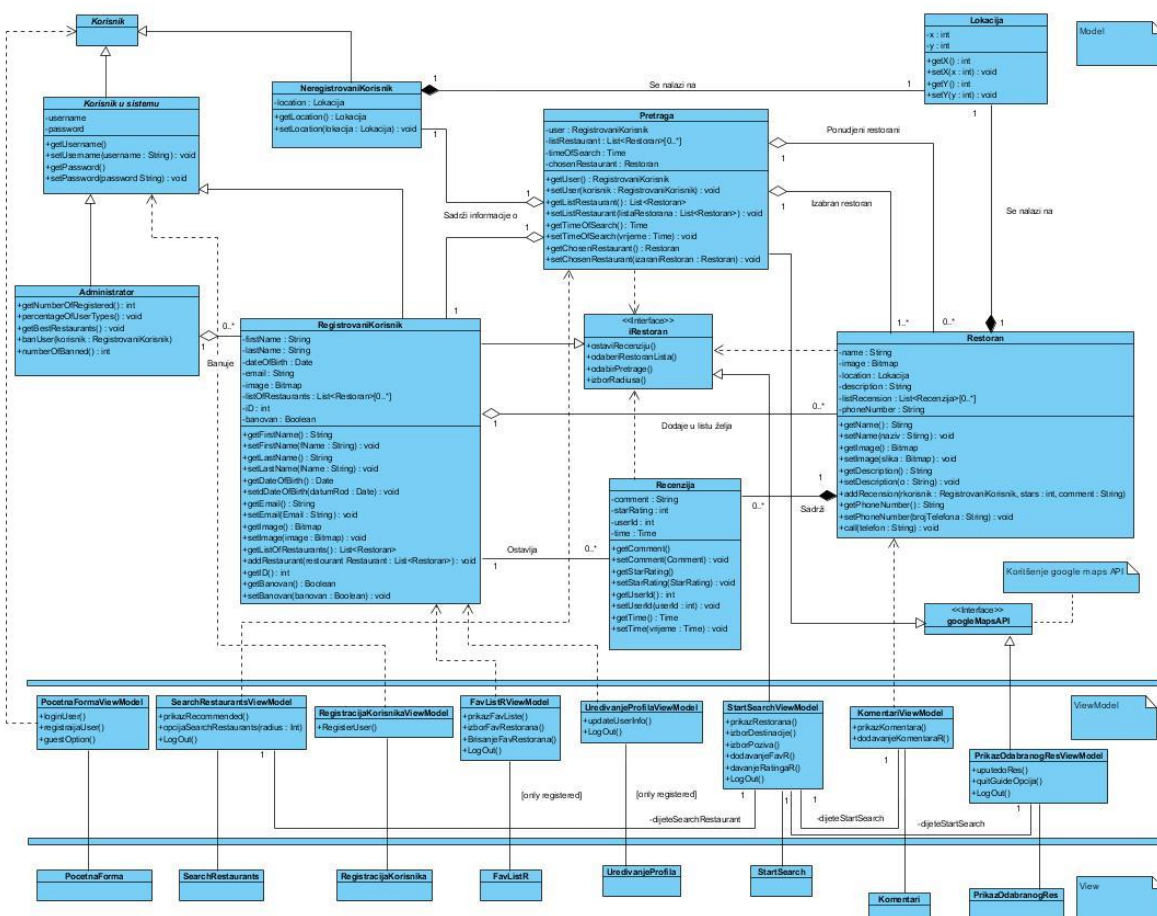


Slika 3. Pretraga restorana



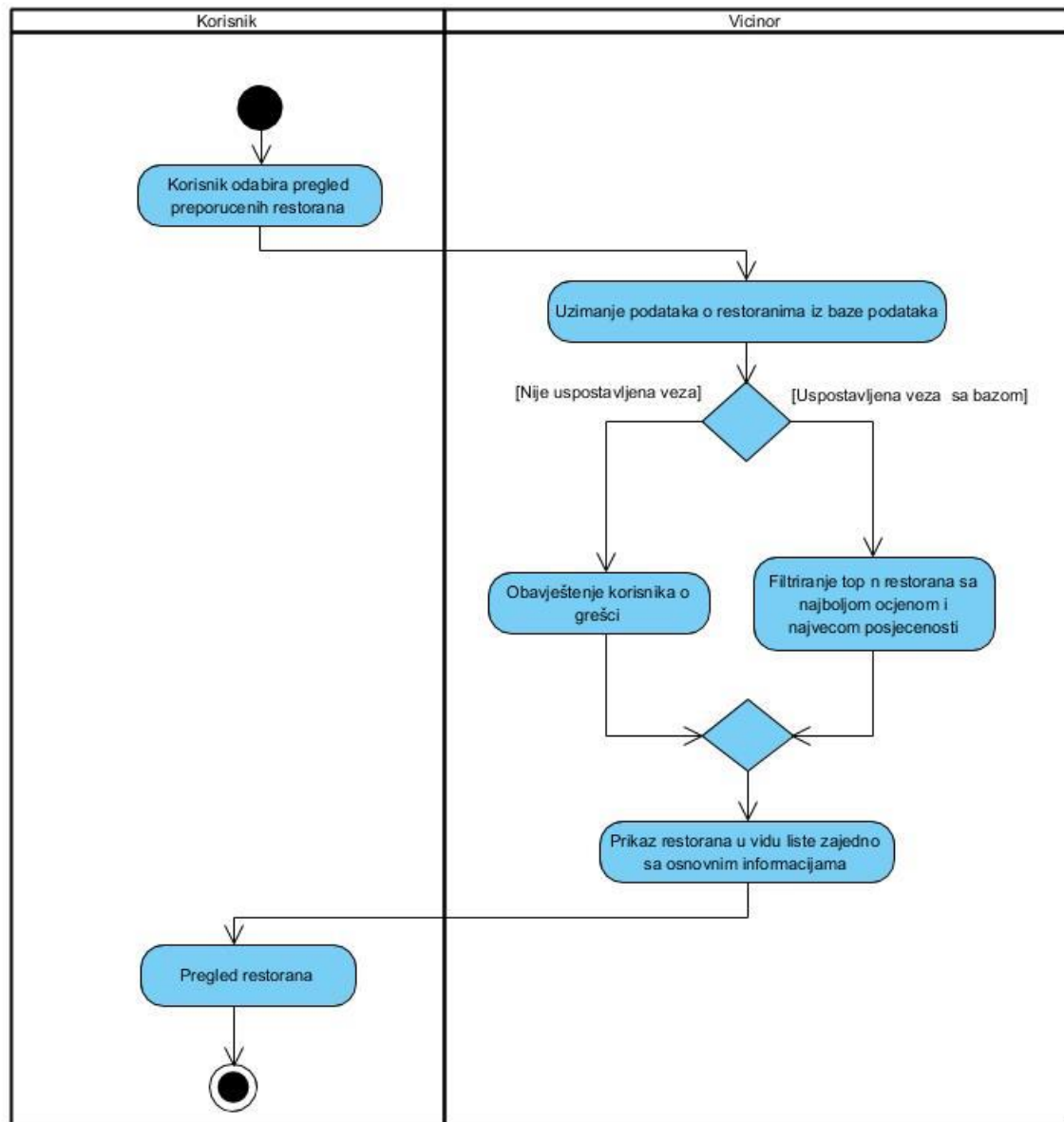
Slika 4. Registracija korisnika

Dijagram klasa (MVVM)

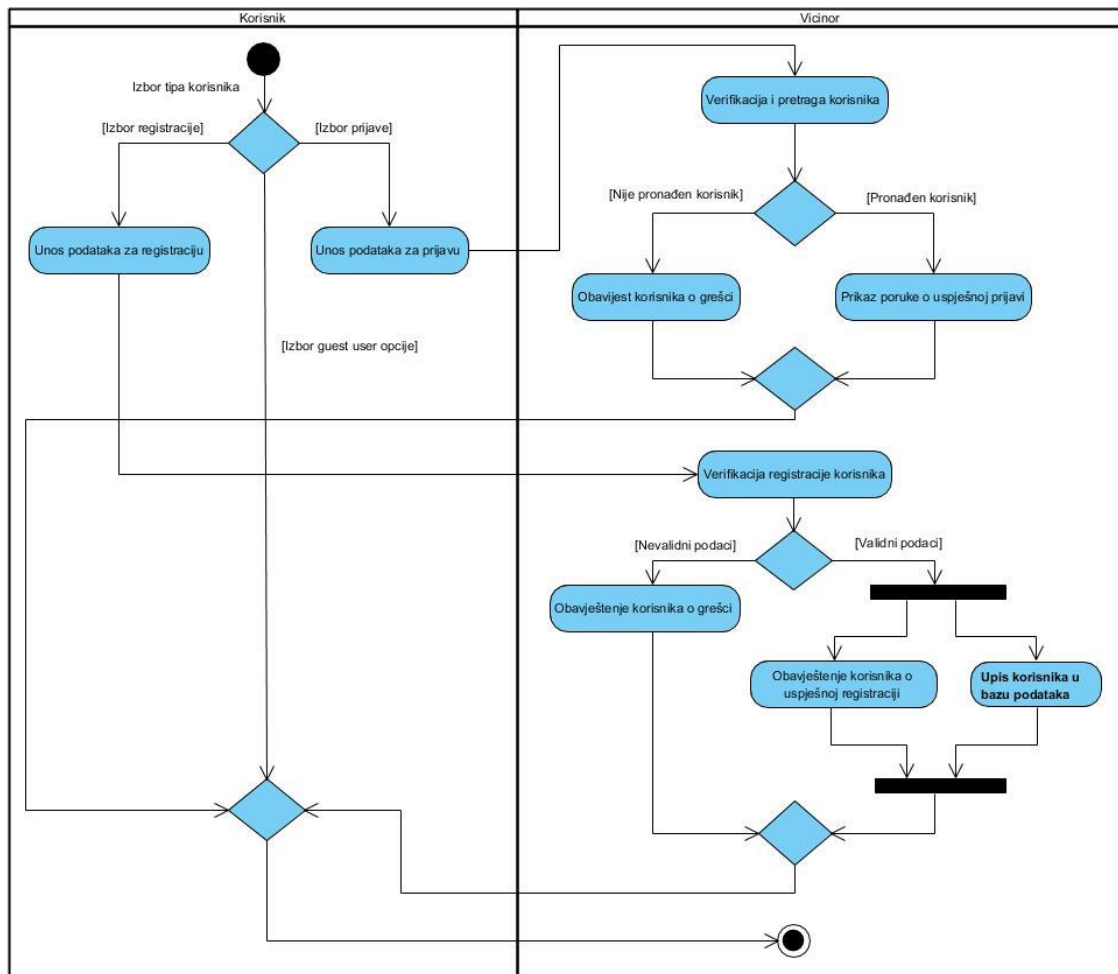


Slika 5. Dijagram klasa

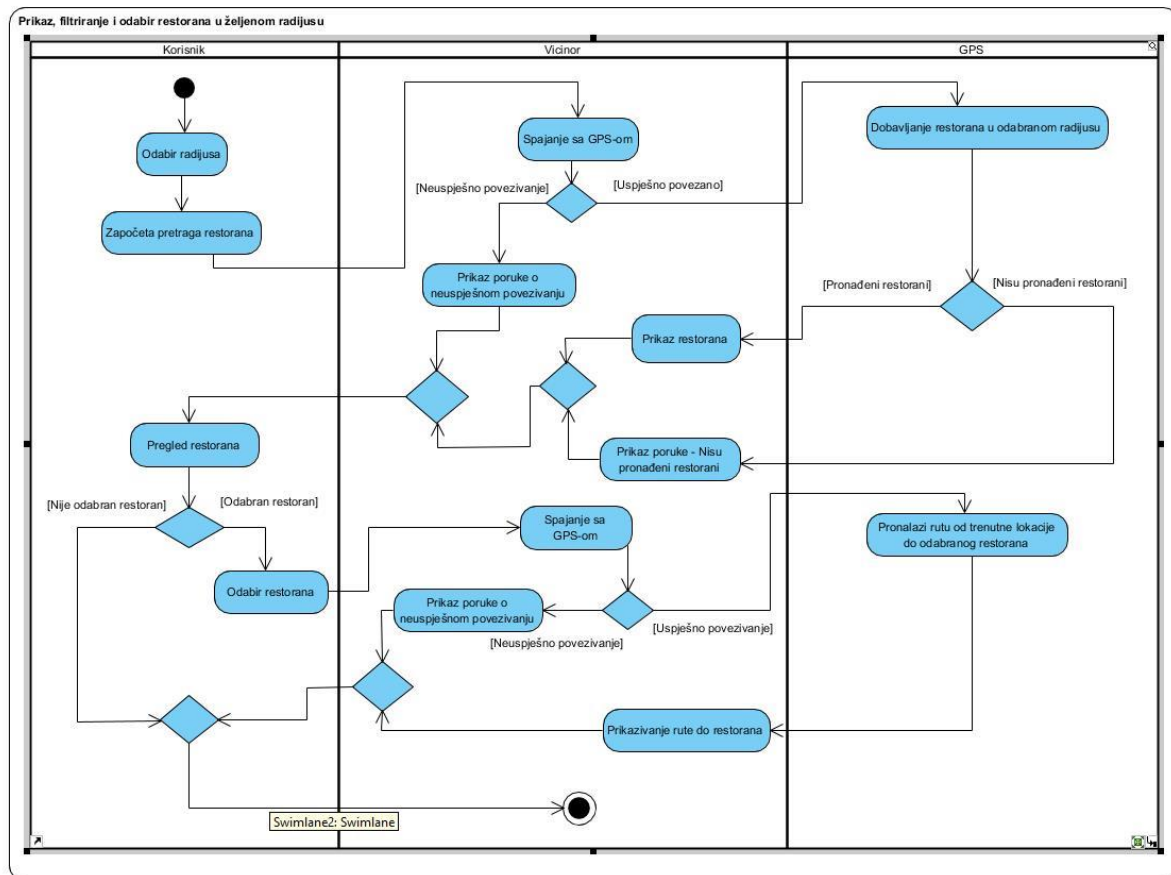
Dijagrami aktivnosti



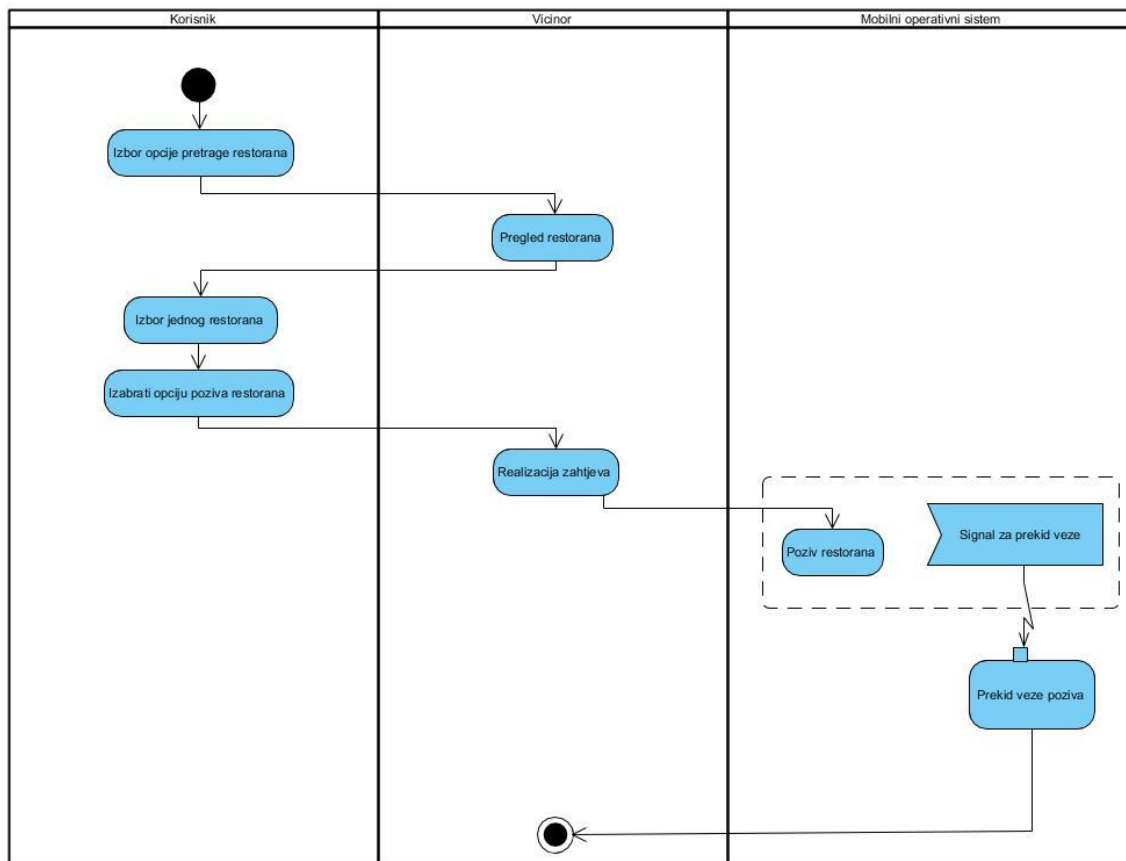
Slika 6. Generisanje preporučenih restorana



Slika 7. Prijava i registracija

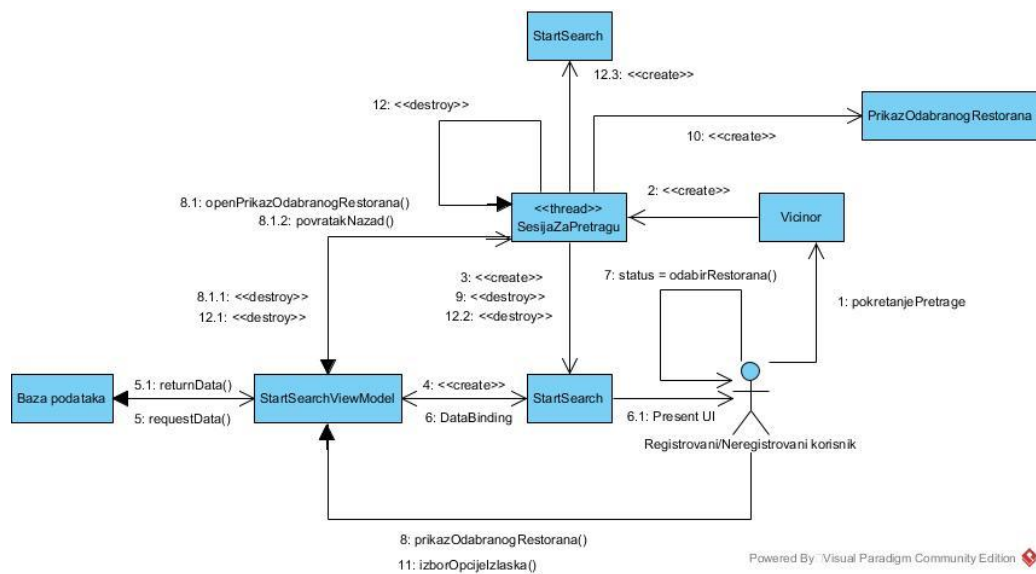
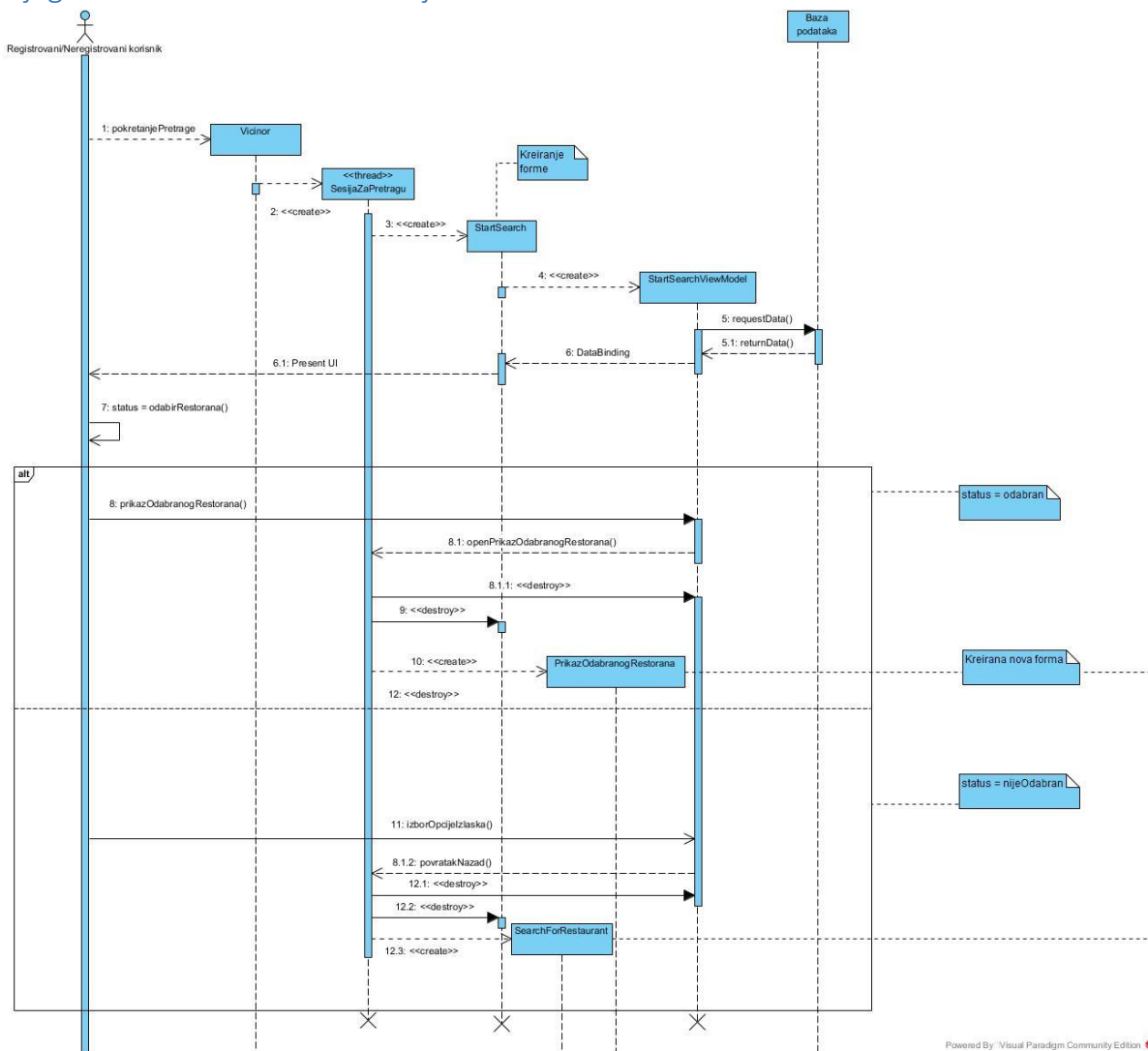


Slika 8. Prikaz, filtriranje i odabir restorana

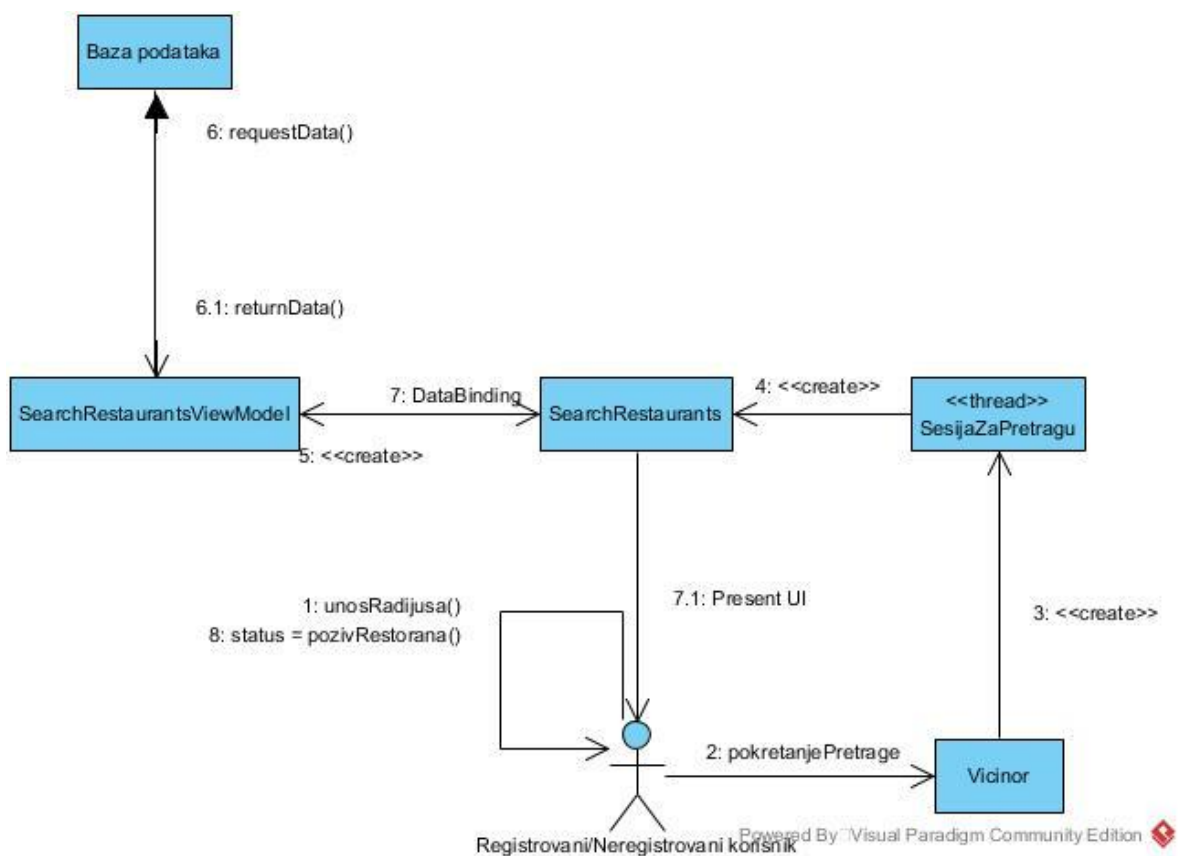
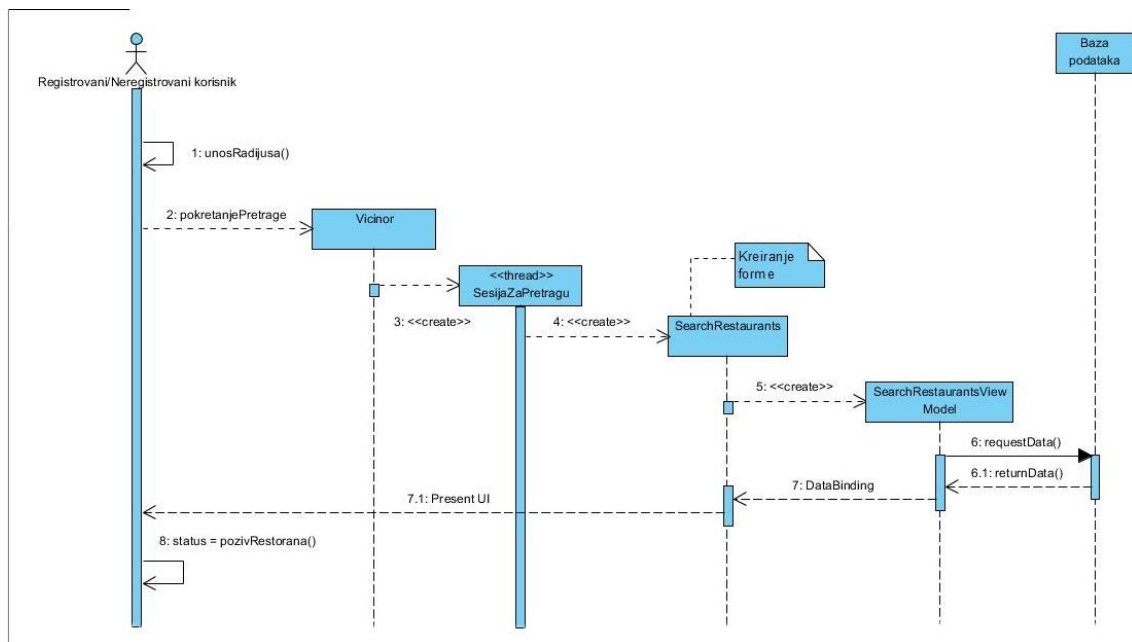


Slika 9. Odabir pazara

Dijagrami sekvenci i komunikacije



Slika 10. i 11. Odabir restorana

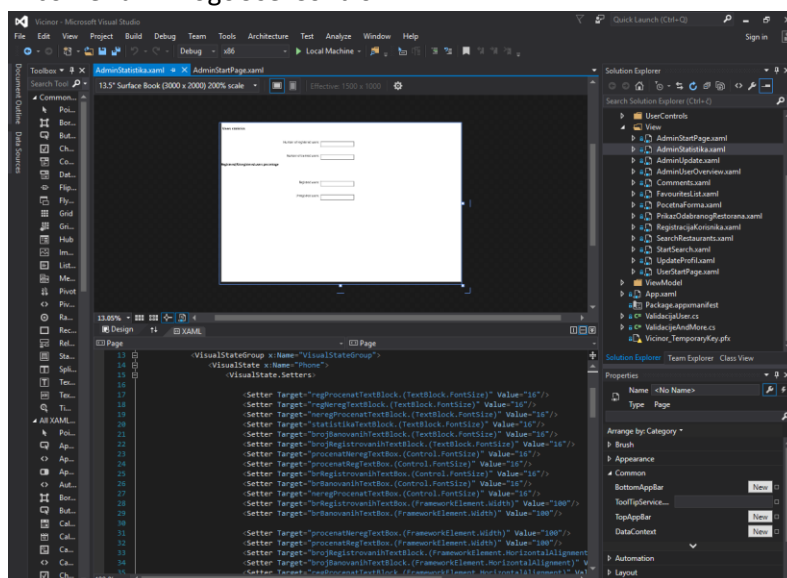


Slika 12. i 13. Pretraga restorana

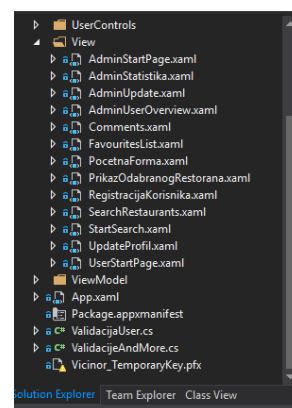
Dijagrami komponenti, paketa i raspoređivanja

Izveštaj o radu

- Odrađen dizajn u UWP-u i implementacija adaptivnog korisničkog Interfejsa (View) za forme administracije AdminStartPage, AdminStatistika, AdminUpdate i AdminUserOverview. Svaki član tima implementirao po jednu formu te po jednu korisničku kontrolu. Korisničke kontrole koje su implementirane: loginUserController, procenaUserController i banUserController. Naknadno dodane i korisničke kontrole RatingControl, welcomeLogoUserController i welcomeAdminLogoUserController.

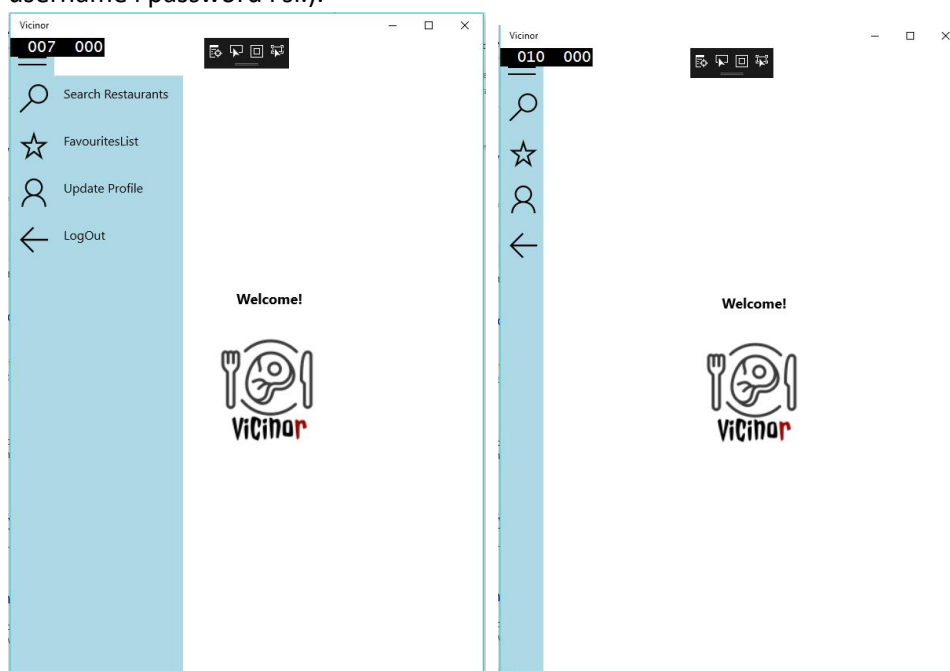


Slika 16. Adaptivni interfejs

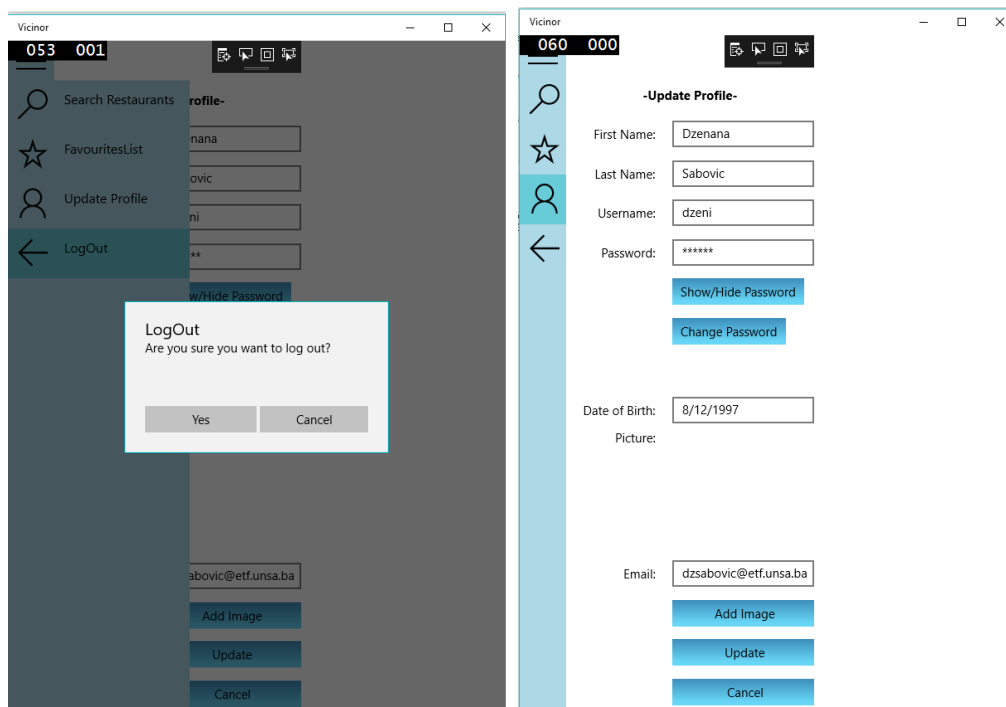


Slika 17. View klase

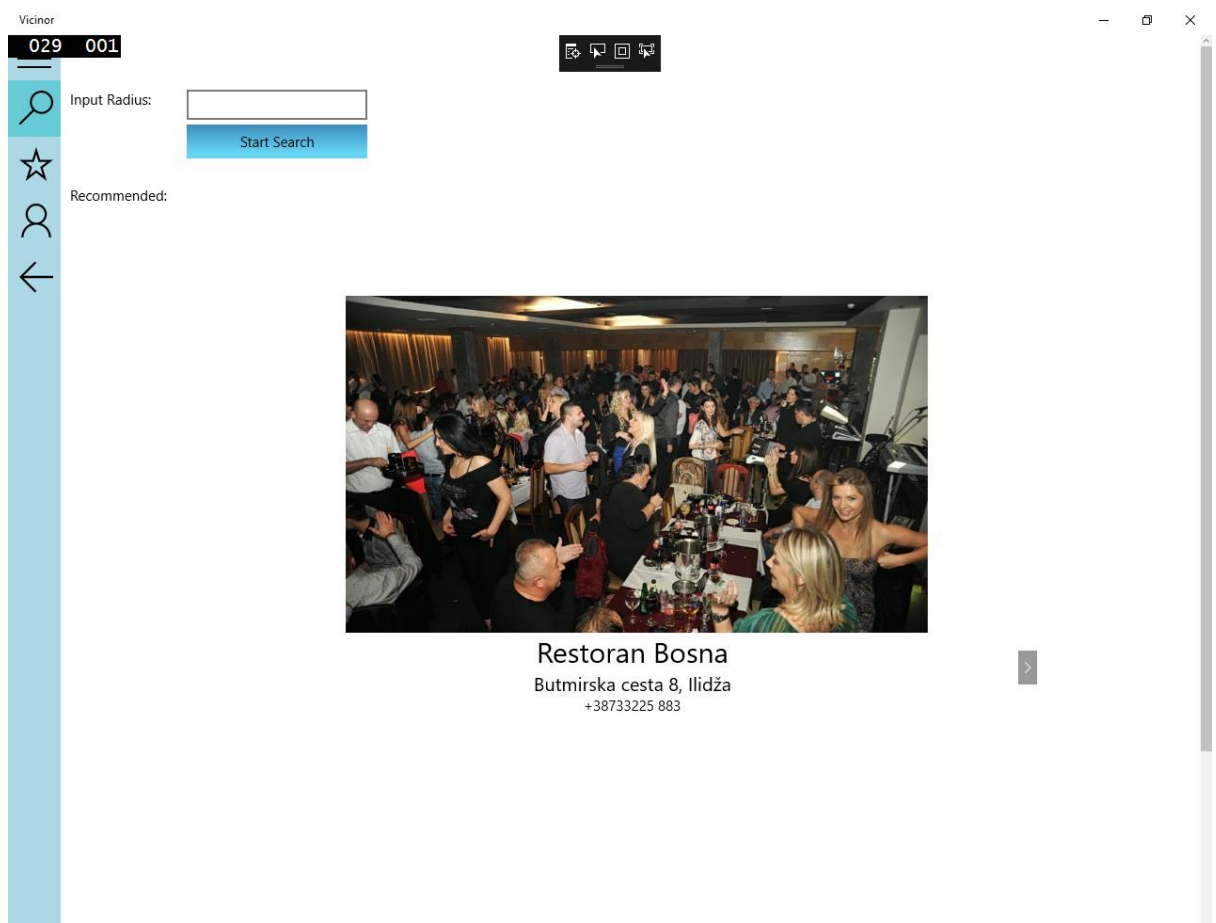
- Implementiran model – funkcionalnosti u UWP-u za forme administracije. Korišten MVVM arhitekturni patern. Svaki član tima implementirao po jednu klasu ViewModel za odgovarajući View. Implementirane klase modela. Odrađena validacija podataka (ispravan username i password i sl.).



Slika 18. i 19. Početna forma za korisnika (analogno urađeno i za administratora)

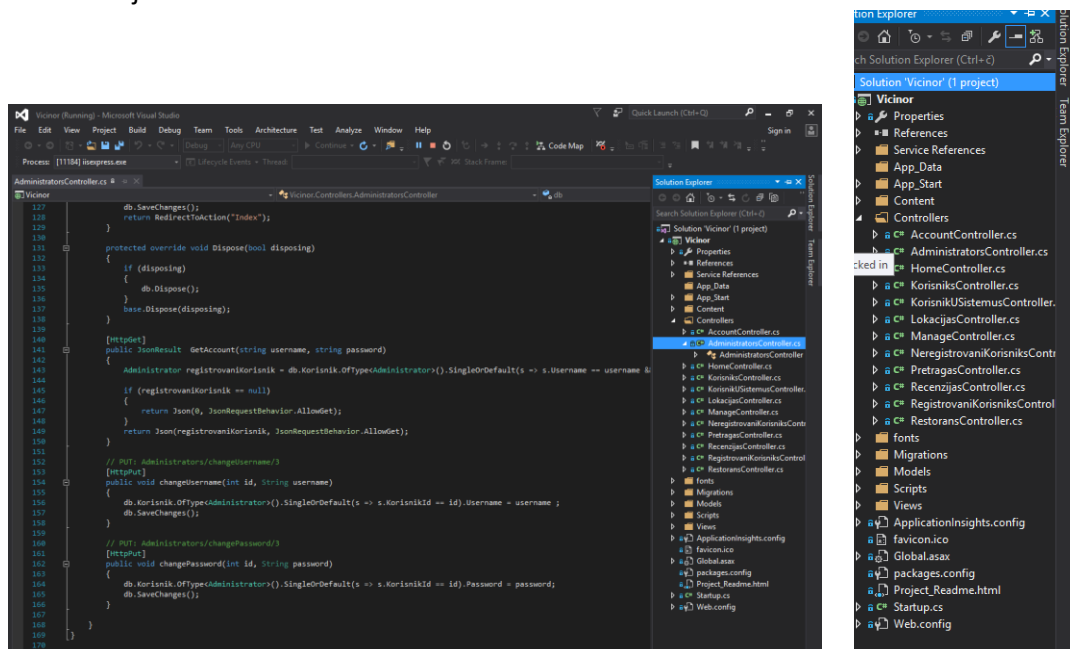


Slika 20. i 21. LogOut korisnika i UpdateProfile forma (Phone view)



Slika 22. Forma za pretragu restorana (Desktop View)

3. Implementirana perzistencija podataka – rad sa bazom podataka na cloudu. Povezani backend i frontend dijelovi projekta. Implementirano spašavanje podataka za administraciju. Svaki član tima implementirao upis podataka ili čitanje podataka iz baze.
4. Kreiran ASP.NET projekat pri čemu je arhitektura aplikacije MVC. Implementiran model. Rađeno zajedno.



Slika Kontroleri

5. Kreirani odgovarajući kontroleri i Views te izvršeno povezivanje sa modelom. Radili svi članovi zajedno.
6. Rađeno sa Azure bazom podataka.
7. Implementiran rad sa vanjskim uređajima (GPS za uzimanje trenutne lokacije korišten na više mjesta – za uzimanje restorana u određenom radijusu u odnosu na trenutnu lokaciju te za prikaz rute do odabranog restorana). Korištene mape za prikaz rute. Svaki član tima je implementirao rad sa vanjskim uređajem ili jednu specifičnu funkcionalnost.
8. Implementiran ASP.NET WEB API servis i REST API za određene funkcionalnosti koje koristi MVC aplikacija. Radili svi članovi skupa.
9. Odrađen poziv API servisa iz ASP.NET MVC aplikacije. Radili svi članovi skupa.
10. Odrađena dokumentacija.
11. Urađen refaktoring koda na osnovu kataloga refaktoringa. Analizirana mogućnost primjene 6 refaktoringa te svaki član tima implementirao po jedan refaktoring. Analizirana mogućnost refaktoringa pomoću dizajn paterna i implementirana tri dizajn paterna (Svaki član tima implementirao po jedan). Napisana dokumentacija za refaktoring.

Tabela očekivanih i implementiranih funkcionalnosti

Akter	Funkcionalnost	Implementirano
Administrator	Izmjena profila (lozinka i password)	Da
Administrator	Pregled statistike	Da
Administrator	Banovanje korisnika	Da
Neregistrovani korisnik	Registracija korisnika	Da
Registrovani korisnik	Dodavanje restorana u listu omiljenih	Da
Registrovani korisnik	Uređivanje profila	Da
Korisnik	Pretraga restorana u željenom radijusu	Da
Korisnik	Pregled osnovnih informacija o restoranu	Da
Korisnik	Pregled recenzija restorana	Da
Korisnik	Dobijanje uputa do restorana	Da