GPS

Projekt iz kolegija Računarski praktikum 3 – C#

Studentice: Mihaela Gamulin, Valentina Gavranić, Dajana Jerončić

Zagreb, 2019.

1. OPIS APLIKACIJE

Aplikacija sadrži plan grada Zagreba. U planu se nalaze čvorovi koji predstavljaju križanja, trgove ili značajna mjesta, te bridovi koji predstavljaju ulice (jednosmjerne ili dvosmjerne). U plan je moguće dodavati nove čvorove preko koordinata i nove linije tako da se unesu početni i završni čvor. Svaki čvor i linija mogu imati neke karakteristike kao što su: trgovina, restoran, kafić, pošta, ljekarna, garaža i benzinska postaja.

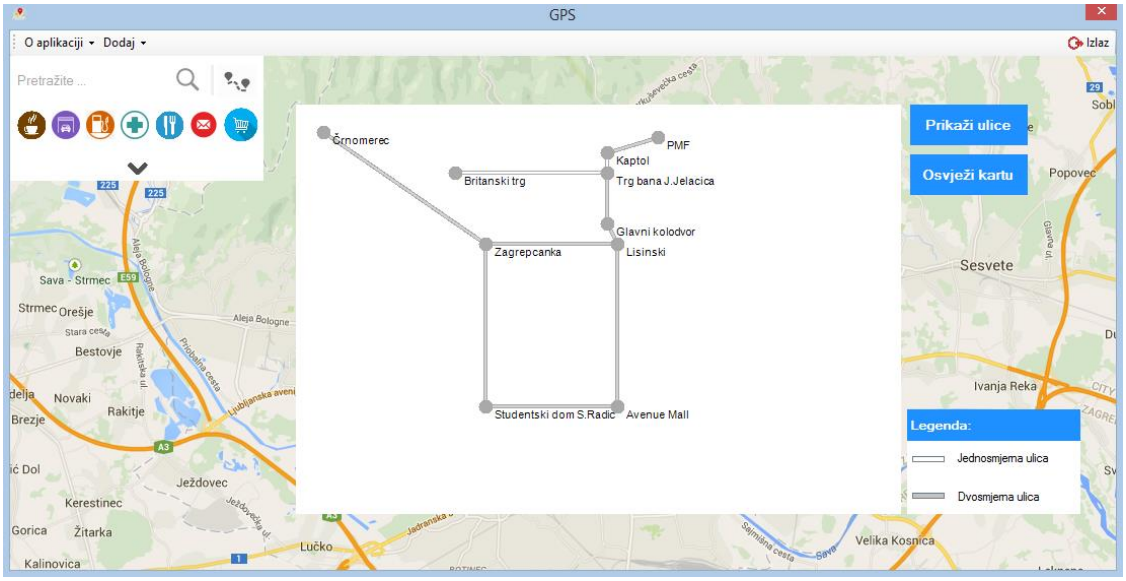
Detalji o čvoru, odnosno bridu pamte se u bazi i, ukoliko korisnik to želi, mogu se prikazati. Kao osnovni dio, prilikom pretraživanja nekog objekta, prikazuje se lokacija na karti, a zatim se mogu dobiti detaljnije informacije o karakteristikama koje ta lokacija sadrži.

Korisnik može nadalje pronaći najkraći put od lokacije A do lokacije B uz detaljne upute o duljini puta i nazivima ulica kojima mora proći. Također može pronaći put od te dvije lokacije tako da obavi neki posao, tj. svrati u jednu od karakteristika koju naznači. Na zahtjev korisnika na planu se mogu istaknuti sve ulice i čvorovi koji sadrže zadanu karakteristiku.

2. FUNKCIONALNOSTI I NJIHOV OPIS

2.1. Iscrtavanje mape

Prilikom ulaska, aplikacija iscrtava plan grada Zagreba, gdje je kružićem označen čvor, a linijom označena ulica.

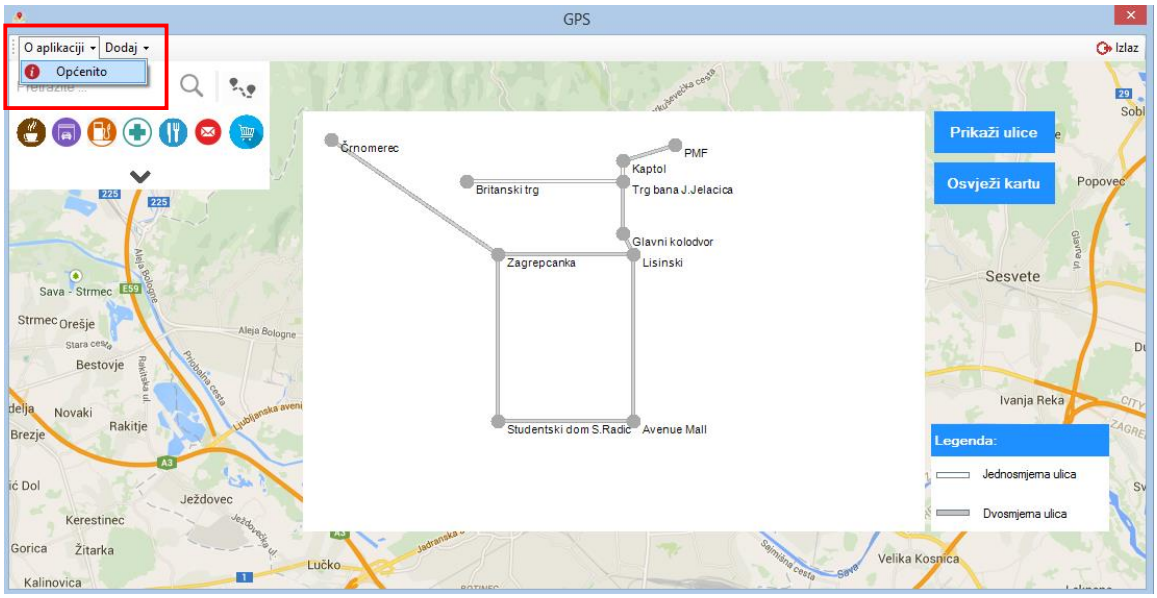


Kao što je vidljivo na legendi u donjem desnom kutu, jednosmjerne ulice su naznačene bijelim linijama, dok dvosmjerne sivim linijama. Na karti su također naznačena imena čvorova.

2.2. Prikaz informacija o aplikaciji

Korisnik može saznati kratki opis aplikacije tako da u gornjem lijevom kutu ode na:

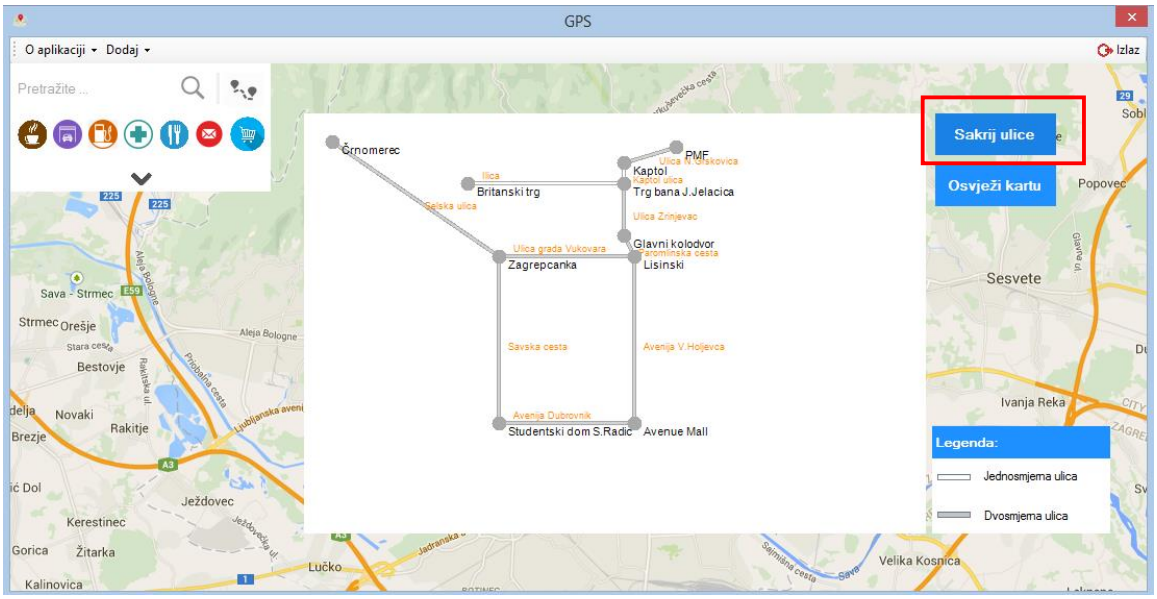
O aplikaciji -> Općenito



2.3. Prikazivanje ulica na zahtjev korisnika

Inicijalno se na mapi vide samo imena čvorova, ali na zahtjev korisnika mogu se prikazati i ponovno sakriti nazivi ulica klikom na gumb:

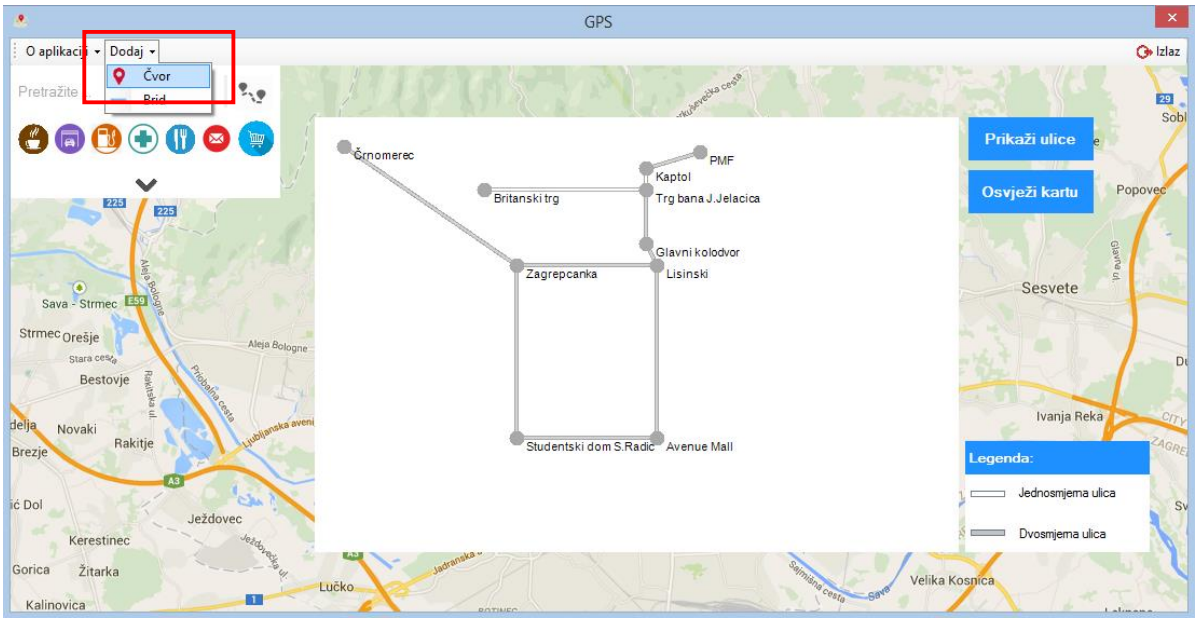
Prikaži ulice / Sakrij ulice



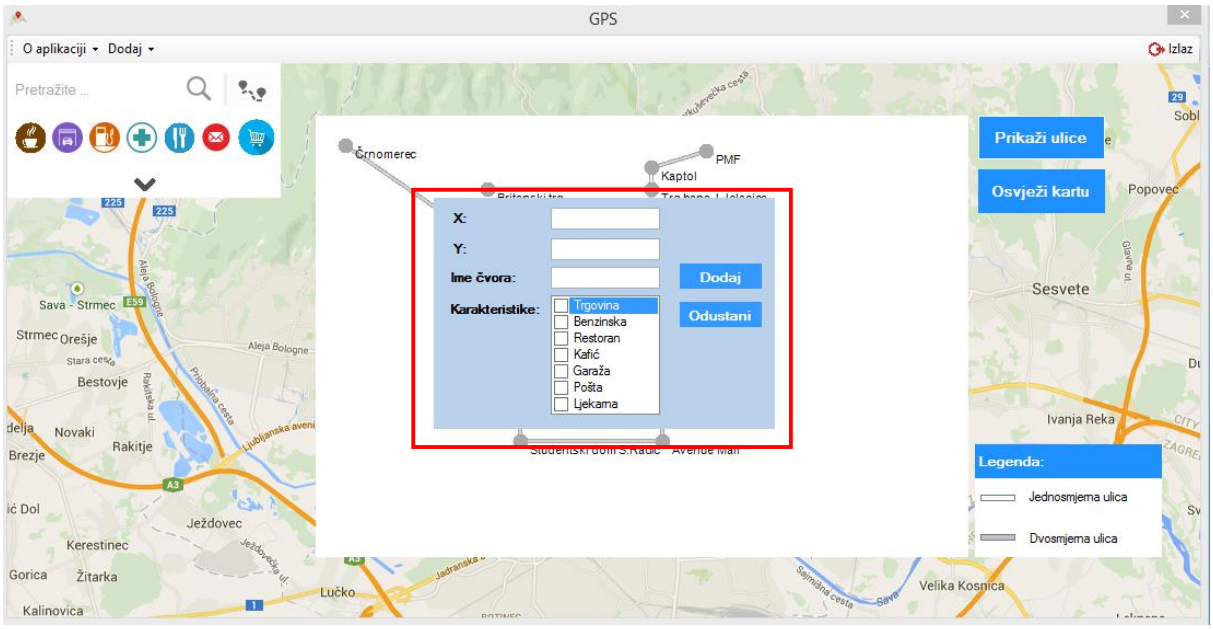
2.4. Dodavanje novog čvora

Novi čvorovi se mogu dodavati tako da u gornjem lijevom kutu korisnik ode na:

Dodaj -> Čvor



Tada mu se otvara forma za unos novog čvora:



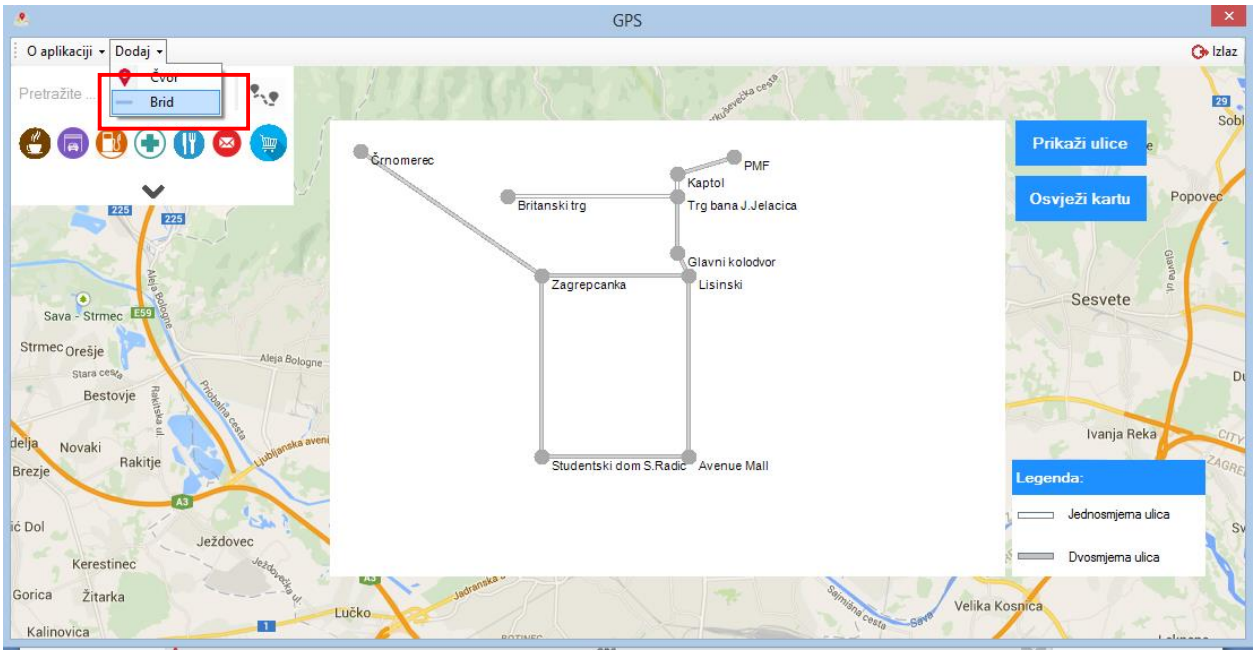
Ovdje korisnik unosi X i Y koordinatu, ime novog čvora i karakteristike koje želi da taj čvor ima.

Ukoliko korisnik ne unese ili krivo unese X, Y ili ime čvora, aplikacija javlja poruku o grešci.

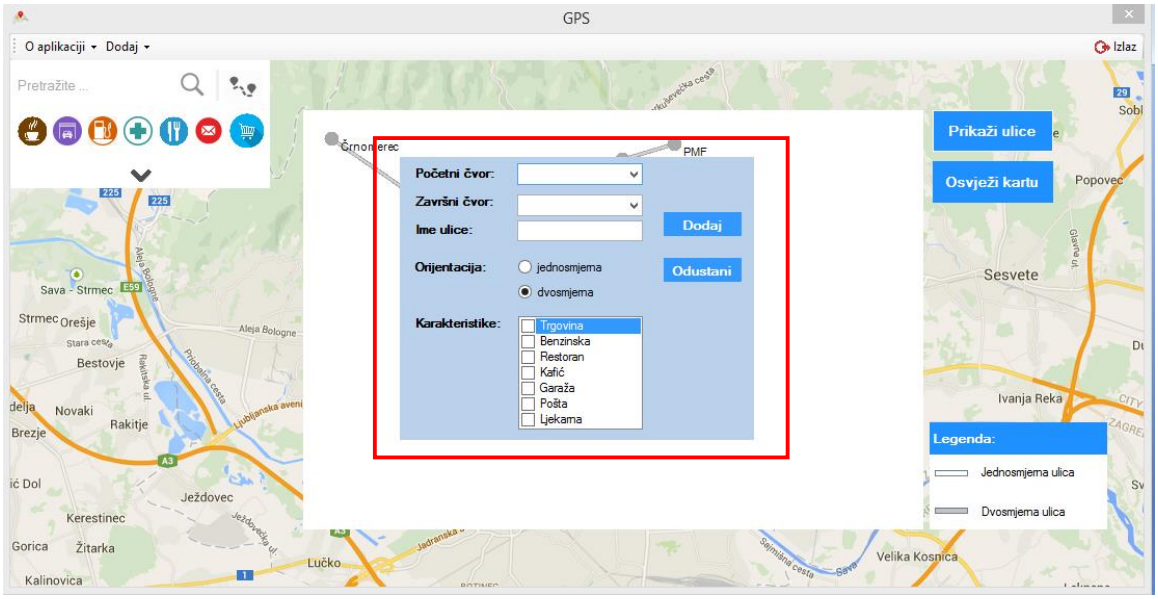
2.5. Dodavanje novog brida

Novi bridovi se mogu dodavati tako da u gornjem lijevom kutu korisnik ode na:

Dodaj -> Brid



Tada mu se otvara forma za unos novog brida:

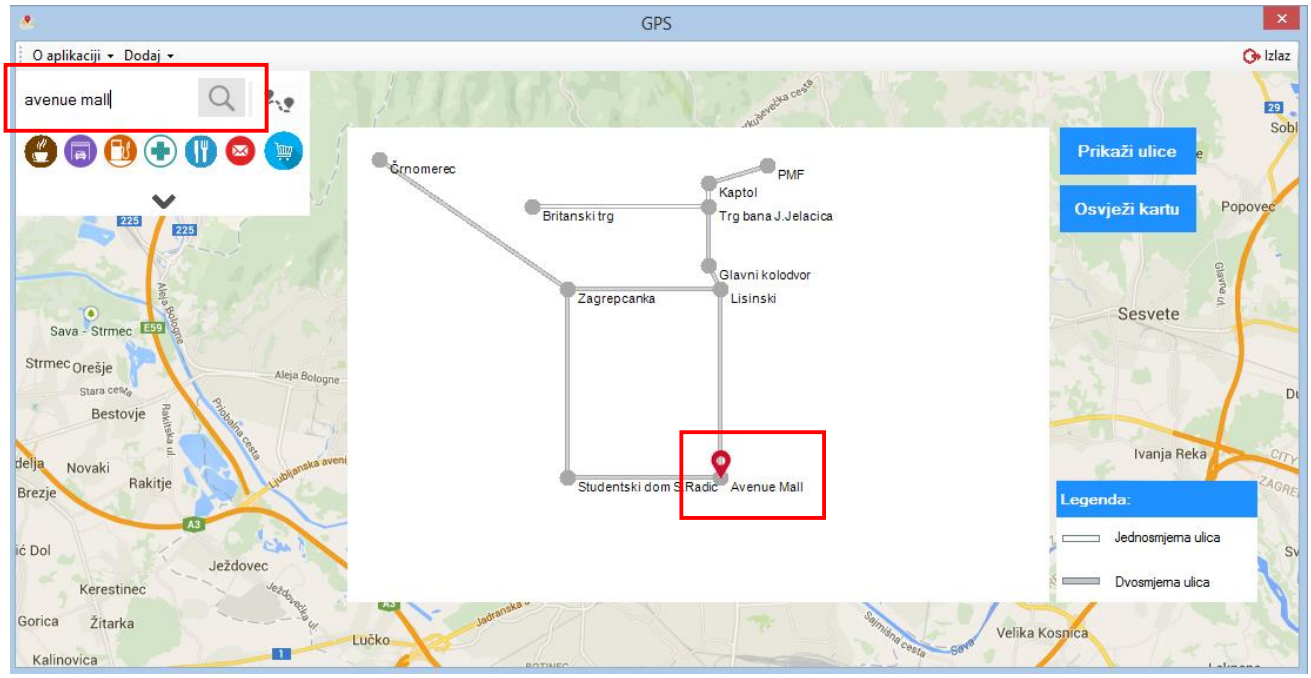


Ovdje korisnik bira početni i završni čvor (ponuđeni oni koji su već na karti), ime ulice, orijentaciju (po defaultu dvosmjerna) i karakteristike koje želi da ta ulica ima.

Ukoliko korisnik ne unese ili krivo unese ime čvora, aplikacija javlja poruku o grešci.

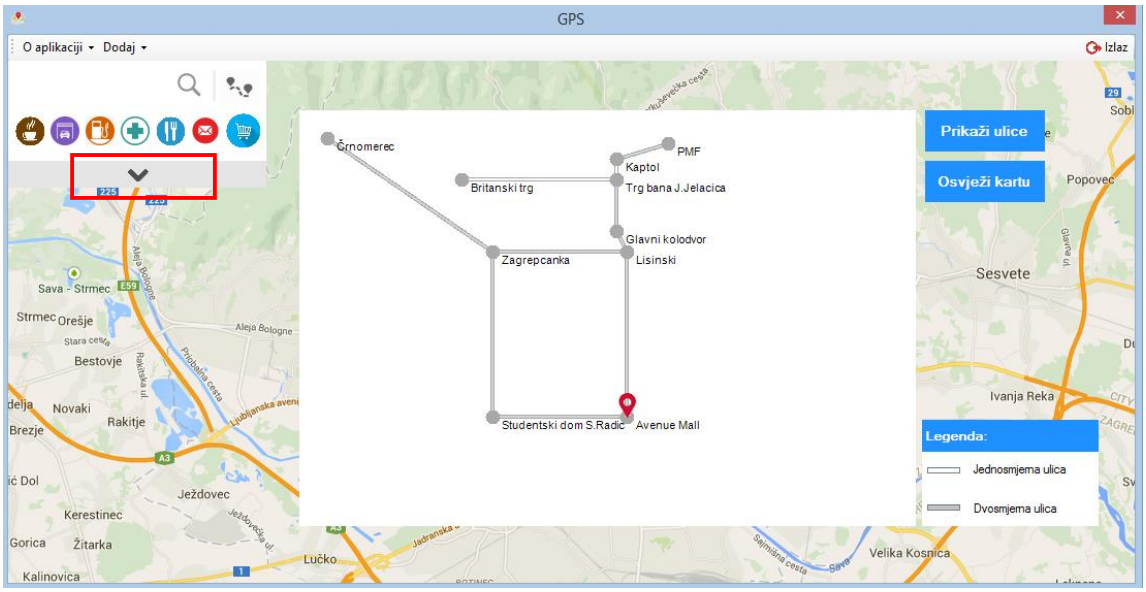
2.6. Pronalazak lokacije

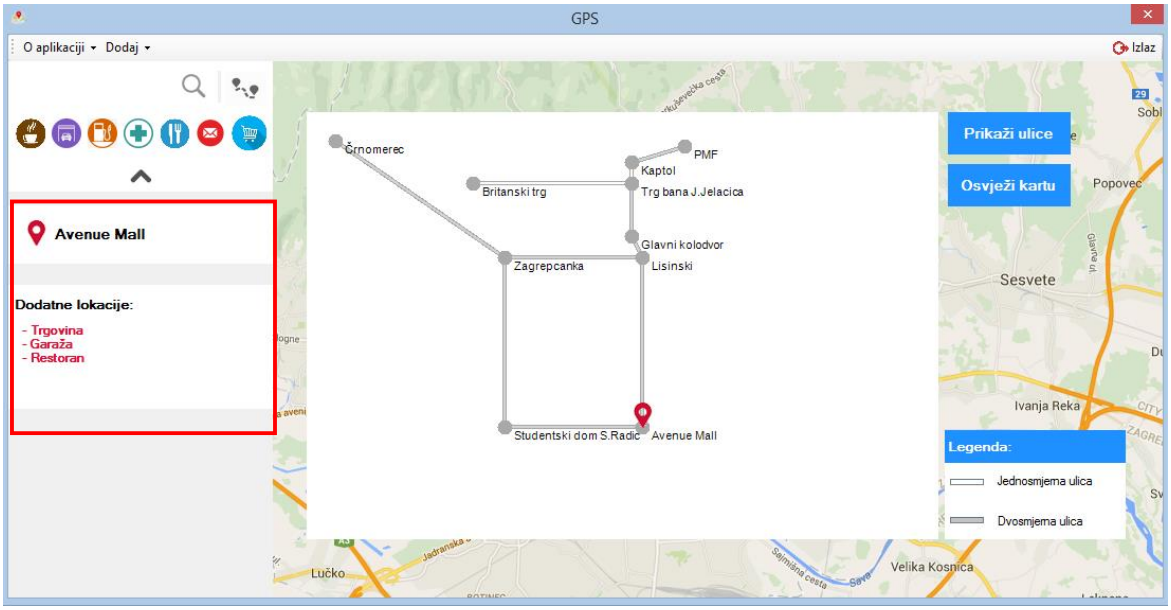
Korisnik može pronaći lokaciju na karti upisom imena te lokacije u pretraživač i pritiskom na gumb traži (povećalo). Tada se na karti pojavljuje oznaka gdje se nalazi navedena lokacija.



2.7. Prikaz detalja o lokaciji

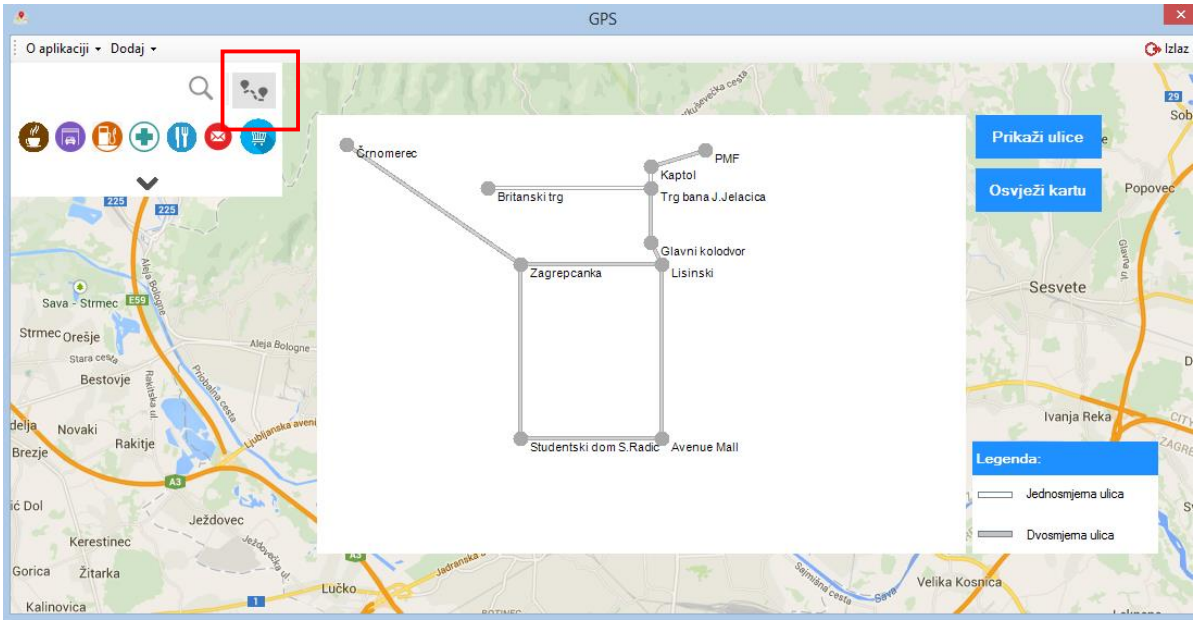
Nakon pronalaska lokacije moguće je dobiti detaljne informacije o karakteristikama koje sadrži ta lokacija kao što je prikazano na slikama.



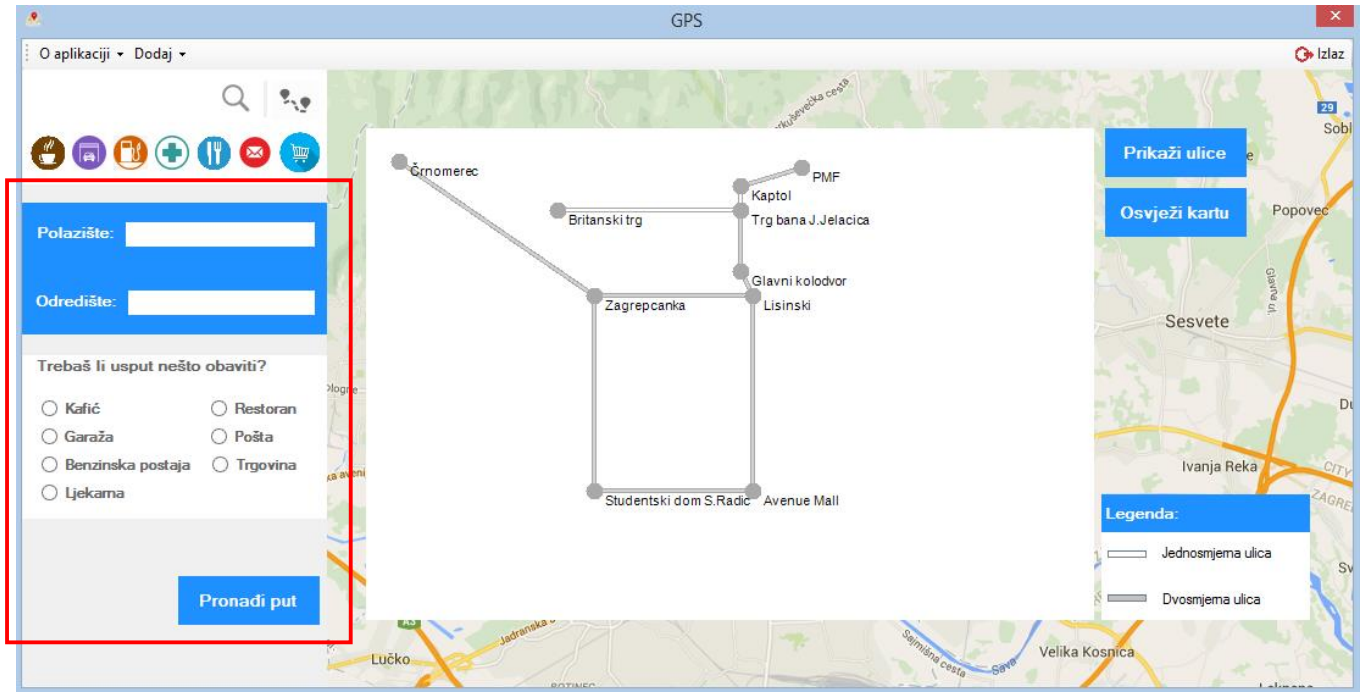


2.8. Traženje puta od lokacije A do lokacije B + uz dodatan posao te prikaz informacija o putu

Korisnik može pronaći najkraći put od lokacije A do lokacije B pritiskom na gumb naznačen na sljedećoj slici:

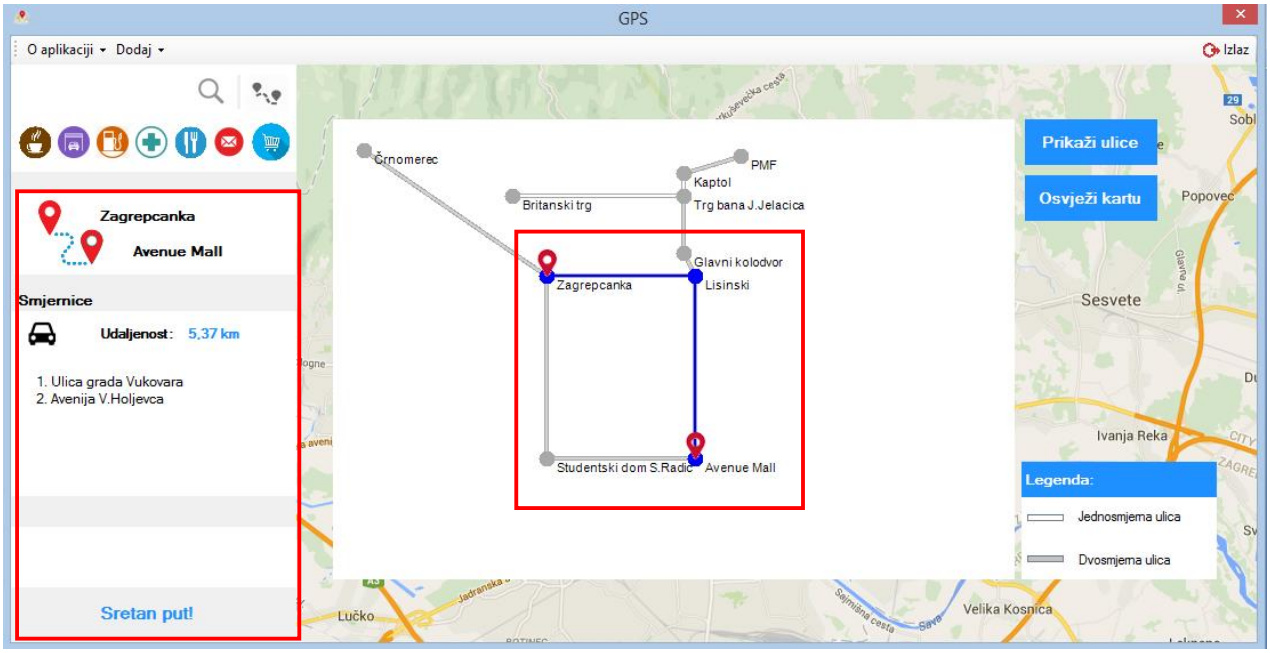


Tada mu se otvara sljedeća forma za unos:



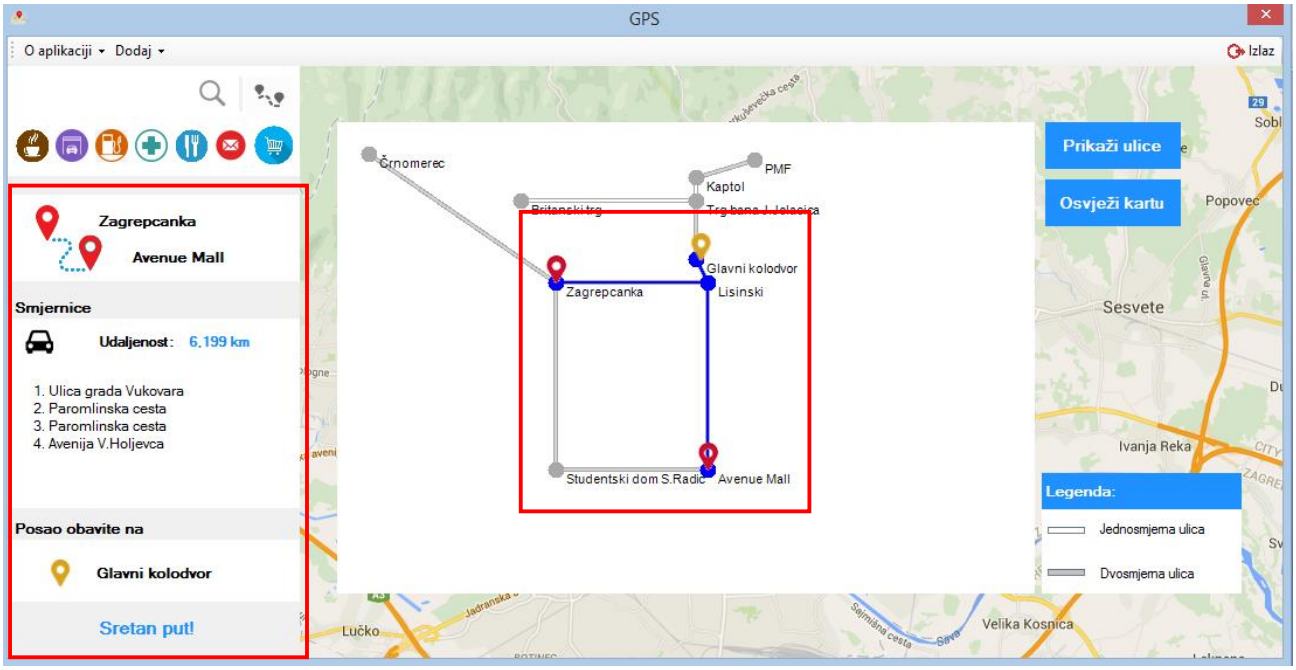
Ovdje korisnik unosi ime polazišta i ime odredišta ukoliko samo treba dobiti najkraći put između tih dviju lokacija ili dodatno još odabere posao koji mora usput obaviti (dozvoljen samo jedan). Klikom na gumb *Pronađi put* dobiva se sljedeći prikaz:

- Ako samo traži najkraći put od lokacije A do lokacije B:



Ovdje dobiva naznačenu putanju na karti te informacije o duljini puta i ulicama kojima mora proći.

- Ako traži najkraći put od lokacije A do lokacije B uz obavljanje nekog posla:



Ovdje dobiva naznačenu putanju na karti, informacije o duljini puta i ulicama kojima mora proći te lokaciju na kojoj najbliže na tom putu može obaviti posao koji je zadao.

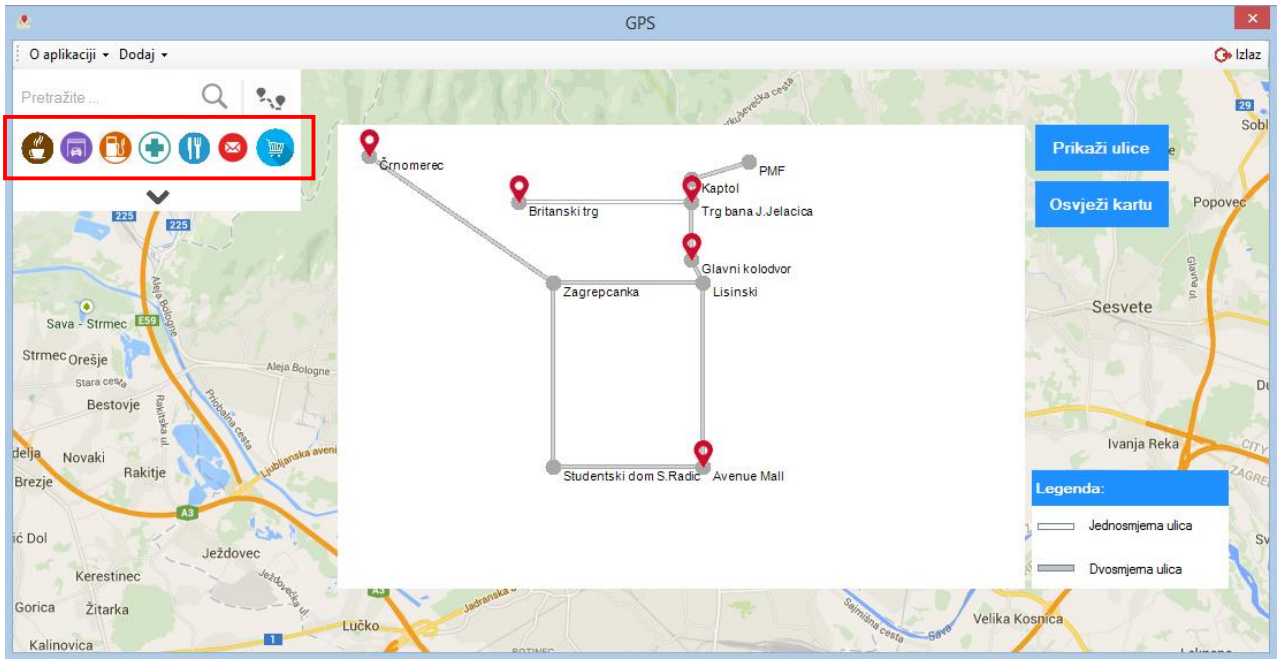
Ukoliko se unese krivo ili nepostojeće ime polazišta/odredišta, aplikacija javlja poruku o grešci.

Napomena:
Ukoliko korisnik traži put koji uključuje jednosmjernu ulicu tada aplikacija može ili ne mora dati rutu puta ovisno u kojem smjeru je orijentirana jednosmjerna ulica.
Npr: ukoliko se pretražuje put od *Kaptol* do *Trg bana J.Jelacica*, tada aplikacija javlja da ne postoji put, dok ako navedemo obratno javit će da put postoji. To se događa jer aplikacija gleda na ulice kao prometnice pa dopušta u jednosmjernim ulicama prolazak samo u jednom smjeru!

2.9. Pretraživanje lokacija sa zadanom karakteristikom

Korisnik može pronaći sve lokacije (čvorove i bridove) sa zadanom karakteristikom i to pritiskom na gumbe koji označavaju određenu karakteristiku (oznake redom: kafić, garaža, benzinska pumpa, ljekarna, restoran, pošta, trgovina).

Sljedeći prikaz se dobije pritiskom na gumb dućan.



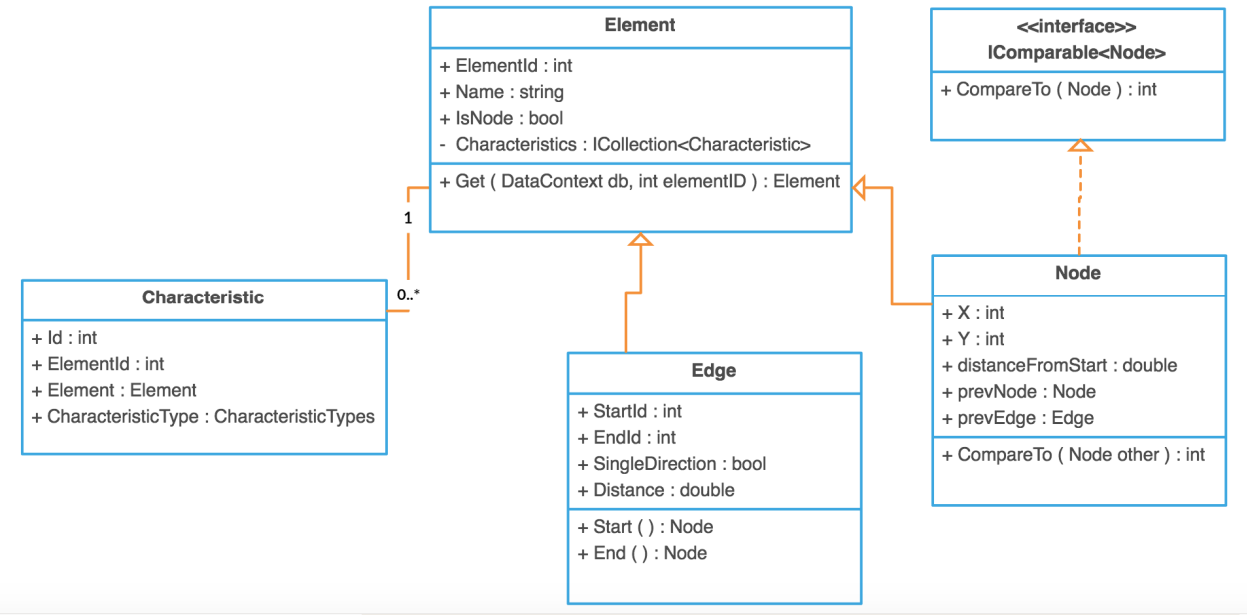
2.10. Osvježavanje aplikacije

Pritiskom na gumb *Osvježi kartu* na desnoj strani aplikacije, prikaz se vraća na početni prikaz bez ikakvih oznaka (kao pri startanju aplikacije).

3. IMPLEMENTACIJA

3.1. Baza podataka

Na sljedećem UML class dijagramu su prikazane klase koje se odnose na podatke spremljene u bazu podataka. Svaki element može biti **Node** ili **Edge**, te u ovisnosti o tipu ima dodatne atribute. Node predstavlja križanja ulica (trgove, znamenitosti, ...), dok Edge predstavlja ulice koje spajaju dva



čvora tipa Node određena atributima StartId te EndId.

Također svaki element može imati 0 ili više karakteristika tipa **Characteristic**. Svaka karakteristika ima svoj tip koji je određen enumeracijom **CharacteristicTypes** prikazanoj na sljedećoj slici.

```
public enum CharacteristicTypes
{
    Trgovina = 0,
    Benzinska = 1,
    Restoran = 2,
    Kafić = 3,
    Garaža = 4,
    Pošta = 5,
    Ljekarna = 6
}
```

Kako bismo mogli direktno stvoriti u bazi tablice koje odgovaraju navedenim klasama, koristili smo **EntityFramework** te smo prvo naredbom **enable-migrations** u Package Manager Console-i omogućili migracije, a zatim automatski stvorili migraciju pomoću naredbe **add-migration AddTables.** U metodi Seed(GPSContext) file-a **Configuration.cs** dodali smo početne bridove i čvorove te karakteristike kako bi aplikacija imala neko početno stanje. Početna baza se nalazi na sljedeće dvije slike.

ElementId	Name	IsNode	StartId	EndId	SingleDirection	Distance	X	Y	Discriminator
24	Glavni...	True	NULL	NULL	NULL	NULL	300	110	Node
25	Trg ba...	True	NULL	NULL	NULL	NULL	300	60	Node
26	Kaptol	True	NULL	NULL	NULL	NULL	300	40	Node
27	PMF	True	NULL	NULL	NULL	NULL	350	25	Node
28	Britan...	True	NULL	NULL	NULL	NULL	150	60	Node
29	Lisinski	True	NULL	NULL	NULL	NULL	310	130	Node
30	Zagre...	True	NULL	NULL	NULL	NULL	180	130	Node
31	Stude...	True	NULL	NULL	NULL	NULL	180	290	Node
32	Avenu...	True	NULL	NULL	NULL	NULL	310	290	Node
33	Ulica ...	False	25	24	False	50	NULL	NULL	Edge
34	Kaptol...	False	25	26	True	20	NULL	NULL	Edge
35	Ulica ...	False	26	27	False	52.2015325...	NULL	NULL	Edge
36	Ilica	False	25	28	True	150	NULL	NULL	Edge
37	Parom...	False	24	29	False	22.3606797...	NULL	NULL	Edge
38	Ulica ...	False	29	30	False	130	NULL	NULL	Edge
39	Savsk...	False	30	31	False	160	NULL	NULL	Edge
40	Avenij...	False	31	32	False	130	NULL	NULL	Edge
41	Avenij...	False	29	32	False	160	NULL	NULL	Edge

Id	ElementId	CharacteristicType
17	25	0
18	25	2
19	25	6
20	28	0
21	24	0
22	24	5
23	24	1
24	32	0
25	32	4
26	32	2

Stoga je kod prvog pokretanja aplikacije potrebno u konzoli naredbom **update-database** pokrenuti migraciju te seed metodu kako bi se stvorile tablice te navedeno početno stanje baze.

3.2. Najkraći put

Najkraći put je dobiven implementacijom Dijkstrinog algoritma u klasi **ShortestPath**. Klasa **GPSNavigation** se koristi za dobivanje nekih podataka kao što su brid između dva čvora, duljina toga brida te susjedni čvorovi. ShortestPath postavi duljinu najkraćeg puta, te listu čvorova i bridova koji se moraju proći kako bi se došlo od početnog čvora start do završnog čvora end. Ukoliko taj put ne postoji (jer npr. čvorovi nisu povezani ili jer su neki bridovi koji spajaju neke čvorove jednosmjerni), tada se varijabla exists postavi na false.

Kod pretrage najkraćeg puta uz uvjet obavljanja nekog posla, promatraju se svi elementi koji imaju na sebi traženi posao, tj. karakteristiku. Zatim se promatra zbroj duljine puta od početnog čvora do tog elementa i od tog elementa do završnog čvora te se traži minimalni takav. Ukoliko postoji, varijabla stop tipa Element se postavi na onaj element koji sadrži karakteristiku preko koje je put najkraći.