

**Univerzitet Singidunum**

**Tehnički fakultet**

**Naslov projektnog rada: PetrolPlus aplikacija upotrebom Design Thinking modelovanja**

**- Projektni rad -**

Predmet: **Metodologija razvoja softvera**

**Profesor:** **Student/i: Poeni**

Prof. dr Angelina Njeguš 1. Marija Arsenijević 2022201592

**Asistent:**  2. Mihajlo Stanković 2022201592

Prof. Petar Biševac

Beograd, 2024. Godine

Sadržaj:

[**1. Uvod** 3](#_Toc184733691)

[**2.Metodologija razvoja softvera** 4](#_Toc184733692)

[**3. Specifikacija korisničkih zahteva** 6](#_Toc184733693)

[**4. Analiza softvera** 12](#_Toc184733694)

[**4.1 Arhitektura sistema** 12](#_Toc184733695)

[**4.2. Analiza slučajeva korišćenja** 13](#_Toc184733696)

[**5. Projektovanje softvera** 18](#_Toc184733697)

[**6. Zaključak** 24](#_Toc184733698)

[**7. Literatura** 25](#_Toc184733699)

## **1. Uvod**

Aplikacija “PetrolPlus” je aplikacija koja omogućava jasan i jednostavan pregled sadržaja benzinske pumpe. Omogućava izvršavanje više zadataka.

U današnje vereme, većina ljudi poseduje mobilne telefone na kojem provode veliki deo slobodnog vremena. Zato je vlasnicima benszinskih pumpi u interesu da svoju benzinsku pumpu mogu predstaviti u obliku aplikacije koju korisnik može instalirati.

Web aplikacije su dostupne na internet pregledima i mogu se pregledati na ličnim računarima i na mobilnim telefonima.

Cilj je da korisnik može instalirati aplikaciju na mobilnom telefonu i da to bude mobilna aplikacija, a ne Web stranica.

Moja aplikacija PetrolPlus služi da korisnicima pruži informacije gde se nalazi objekat i da na njemu mogu da sipaju motorni benzin i dizel. Takođe, mogu da obave i druge aktivnosti i to:

* kupovinu opreme za automobile (motorno ulje, antifriz, tečnost za vetrobrane, sijalice i drugo),
* pumpanje guma,
* pranje vetrobrana,
* kupovinu hrane, pića, novina i drugih artikala.

## **2.Metodologija razvoja softvera**

U toku kreiranja aplikacija “PetrolPlus” primenili smo metoldologiju Design Thinking. Design Thinking je korisnički usmern pristup razvoju aplikacije koja se fokusira na razumevanje korisničkih potreba I kreiranje inivativnih rešenja.

Ova metodologija se sastoji od pet faza:

* Empatija (Emphatize),
* definicija (Define),
* ideacija (Ideate),
* prototipizacija (Prototype) i
* testiranje (Test).

Design Thinking vrši detaljna istraživanja, revizije i pristupaju problemima iz više uglova. Ova metodologija nam je omogućila da koristimo kreativne alate za rešavanje širokog spektra problema, takođe nas je navela na različite načine razmišljanja. Ono što je bitno da smo za kreiranje i ispitivanje aplikacije nismo koristili neke tajne recepte već smo koristili ljude iz odgovarajućih delatnosti kroz faze kako bi olakšali put do inovativnih rešenja.

U nastavku ćemo predstaviti svaku fazu koja je deo metodologije Design Thinkig-a.

1. *Empatija (Emphatize):*

* **Razumevanje korisnika**: Prikupljanje uvida kroz intervjue, ankete i posmatranje korisnika benzinskih pumpi.
* **Mapiranje korisničkog iskustva**: Kreiranje korisničkih persona i korisničkog putovanja (journey map) kako bi se bolje razumele njihove potrebe i bolne tačke.

1. *Definisanje (Define):*

* **Identifikovanje problema**: Analiza prikupljenih podataka kako bi se definisali ključni problemi i izazovi s kojima se korisnici suočavaju.
* **Postavljanje ciljeva**: Jasno definisanje ciljeva koje aplikacija treba da postigne, na primer, poboljšanje korisničkog iskustva na pumpama ili optimizacija procesa plaćanja.

1. *Ideacija (Idea):*

* **Brainstorming**: Generisanje što više ideja za rešenja, bez kritike ili filtriranja. Fokusiranje na inovativne pristupe.
* **Odabir najboljih ideja**: Evaluacija i selekcija najperspektivnijih ideja koje najbolje rešavaju definisane probleme.

1. *Prototipizacija (Prototype):*

* **Kreiranje prototipova**: Razvijanje brzih i jednostavnih prototipova koji vizualizuju ključne funkcionalnosti aplikacije. Ovo može uključivati wireframe-ove, mockup-ove ili interaktivne prototipove.
* **Iterativno poboljšavanje**: Uzimanje povratnih informacija od korisnika i kontinuirano unapređivanje prototipova.

1. *Testiranje (Test):*

* **Testiranje sa korisnicima**: Prikupljanje povratnih informacija od stvarnih korisnika tokom interakcije sa prototipovima.
* **Iterativno prilagođavanje**: Prilagođavanje i unapređivanje aplikacije na osnovu povratnih informacija dok se ne postigne optimalno rešenje.

Primena Design Thinking metodologije na aplikaciji PetrolPlus:

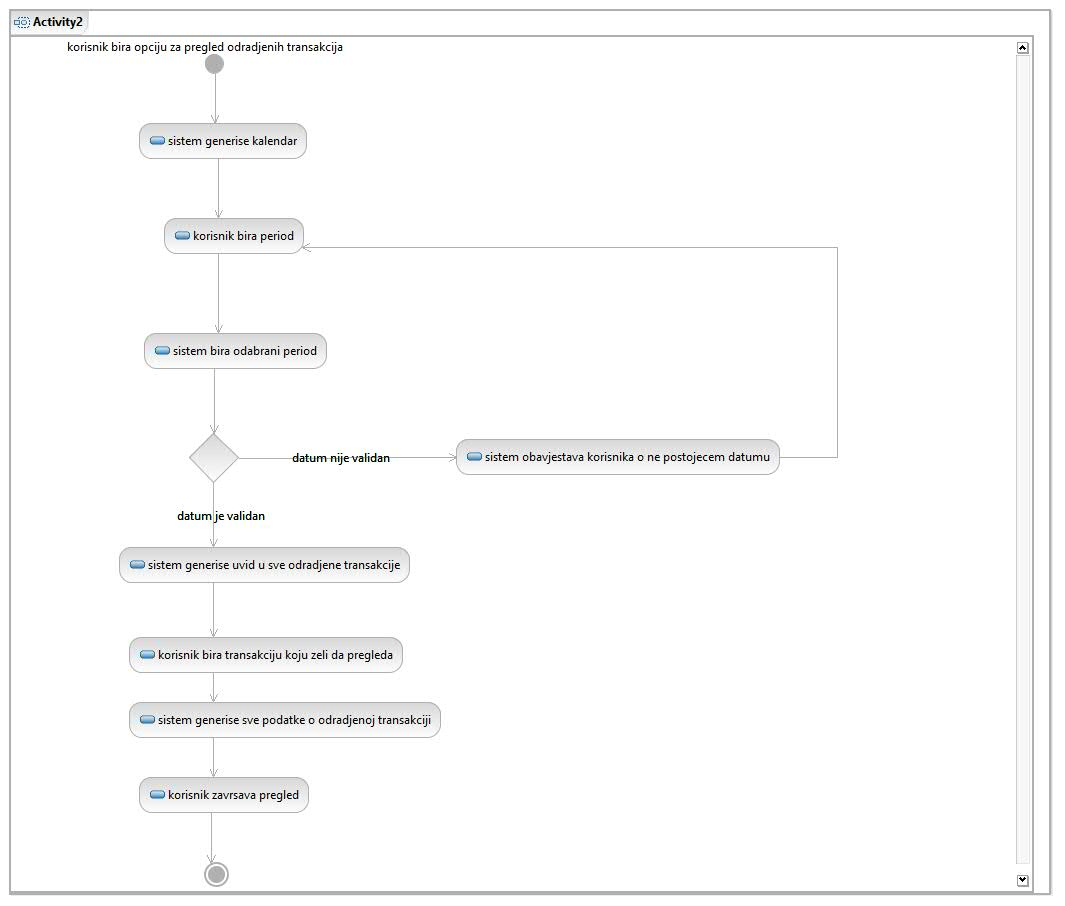
* **Empatija**: Posetite benzinske pumpe, razgovarajte sa korisnicima i radnicima, pratite njihovo ponašanje i potrebe. Kreirajte korisničke persone kao što su "žurni vozač", "redovni kupac", "vozač kamiona", itd.
* **Definisanje**: Identifikujte glavne bolne tačke, kao što su dugi redovi, sporo plaćanje, nedostatak informacija o uslugama, itd.
* **Ideacija**: Generišite ideje kao što su unapredjenje mobilne aplikacije za brže plaćanje, integracija sa navigacionim sistemima za lociranje najbližih pumpi, ili dodavanje lojalnosti programa.
* **Prototipizacija**: Napravite wireframe-ove i interaktivne prototipove aplikacije. Na primer, možete kreirati prototip za brze transakcije plaćanja ili mapu pumpi sa informacijama o trenutnoj dostupnosti goriva.
* **Testiranje**: Prikupite povratne informacije od korisnika kroz testiranje prototipova na stvarnim pumpama. Prilagodite i unapređujte aplikaciju na osnovu tih informacija.



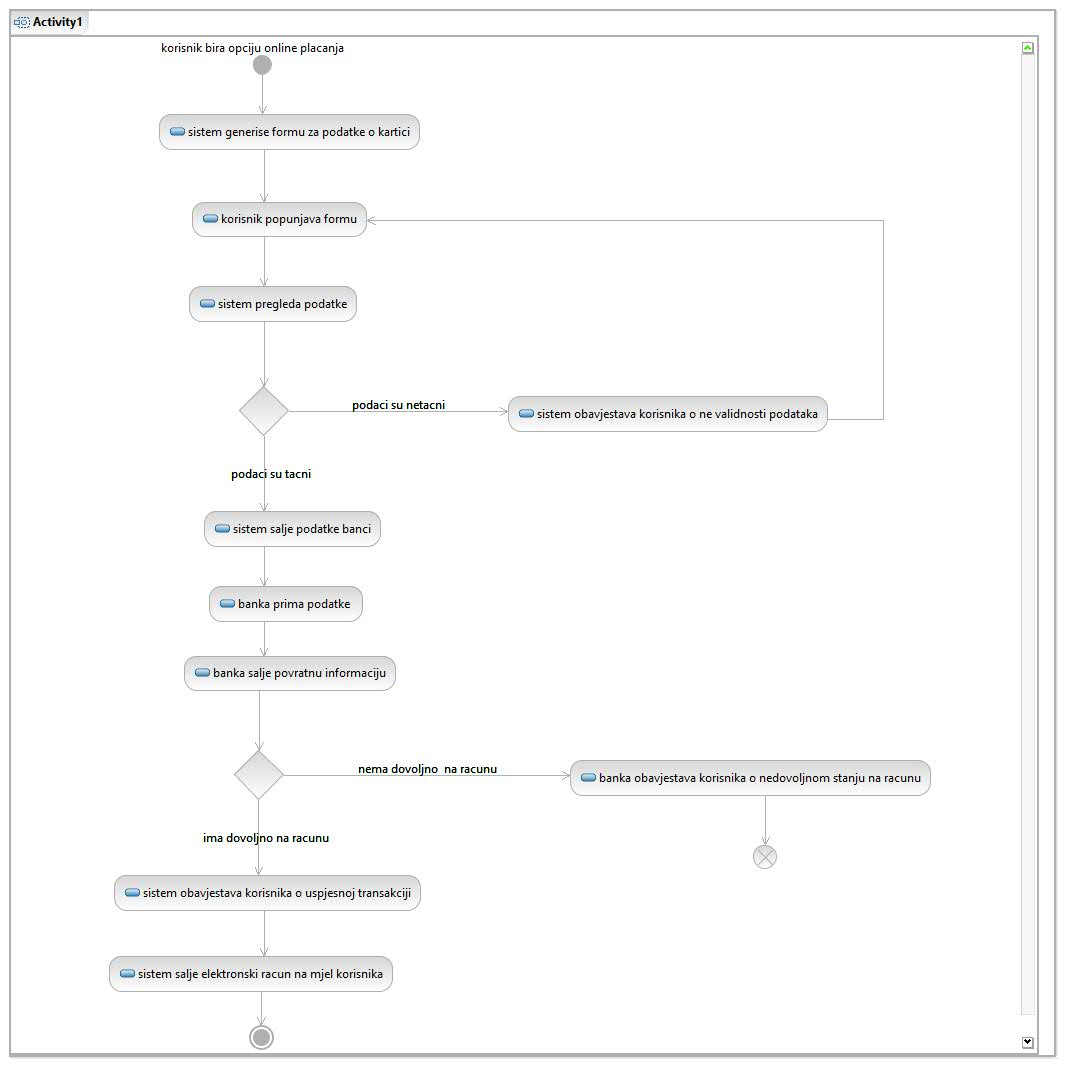
## **3. Specifikacija korisničkih zahteva**



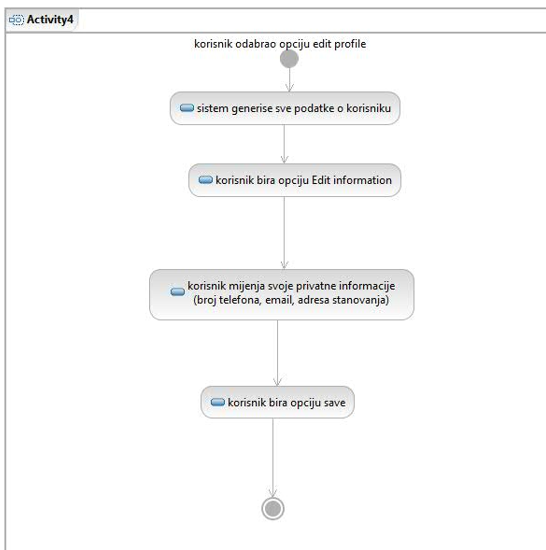
Slika 1. Use case dijagram



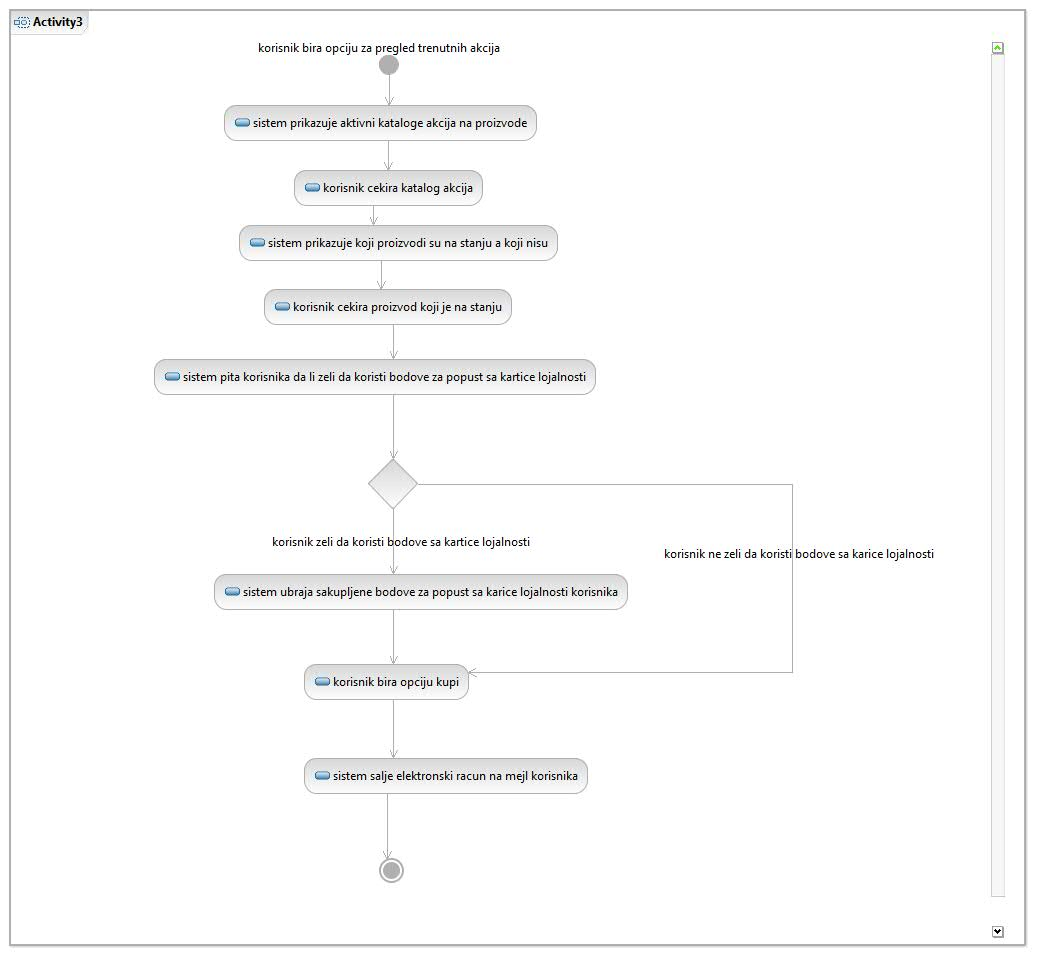
Slika 2. Activity diagram za use case “pregled transakcija”



Slika 3. Activity diagram za use case “online plaćanje “



Slika 4. Activity diagram za use case “edit profile“



Slika 5. Activity diagram za use case “pregled trenutnih akcija “

## **4. Analiza softvera**

## **4.1 Arhitektura sistema**

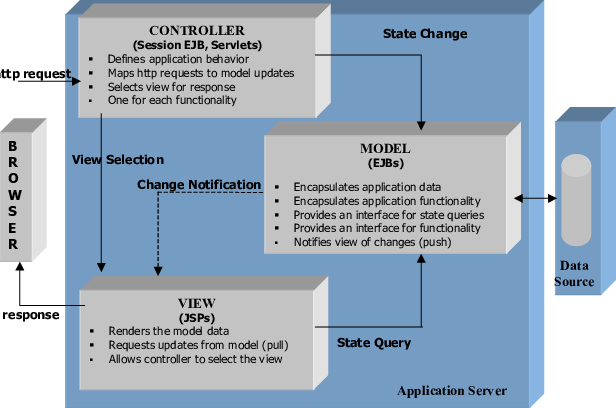
MVC (Model-View-Controller) je arhitektonski/dizajnerski obrazac koji razdvaja aplikaciju u tri glavne logičke komponente:

1. Model (predstavlja poslovnu logiku ili operaciju koju aplikacija treba da obavlja) ,
2. Prikaz ( je komponenta koja je odgovorna za prikaz sadržaja korisniku) i
3. Kontroler( komponenta koja određuje interakcije sa korisnikom , radi sa modelom i bira prikaz koji će izvršiti u aplikaciji).

Svaka arhitektonska komponenta je napravljena za rukovanje specifičnim razvojnim aspektima aplikacije. Izoluje poslovnu logiku i sloj prezentacije jedni od drugih.

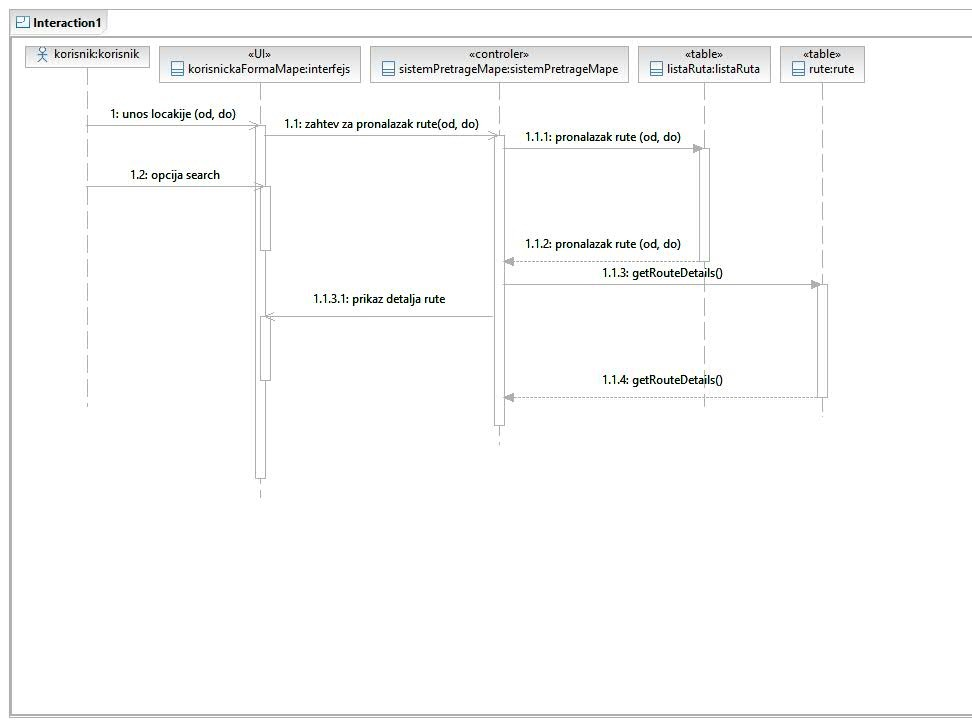
Tradicionalno se koristi za grafičke interfejse radne površine (GUIs). U današnje vreme, MVC je jedan od najčešće korišćenih industrijskih standardnih web razvojnih okvira za kreiranje skalabilnih i ekstenzibilnih projekata. Koriste se i za dizajniranje mobilnih aplikacija.

Tvorac MVC je Trygve Reeskaug. Glavni cilj je rešavanje problema korisnika koji kontrolišu velike i složene podatke postavljene deljenjem velike aplikacije na određene odeljke koja svaka ima svoju namenu.

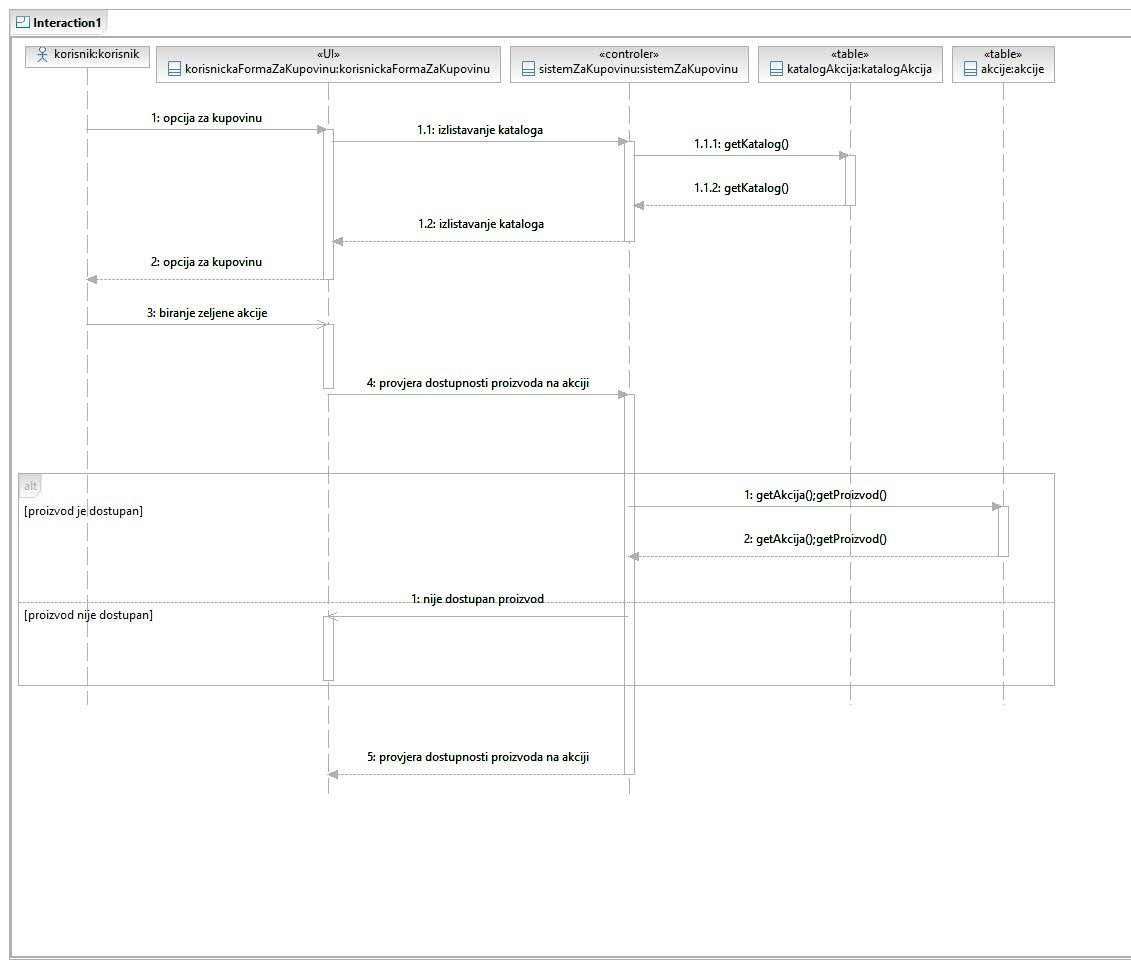
****

Slika 6. MVC Arhitektura sistema

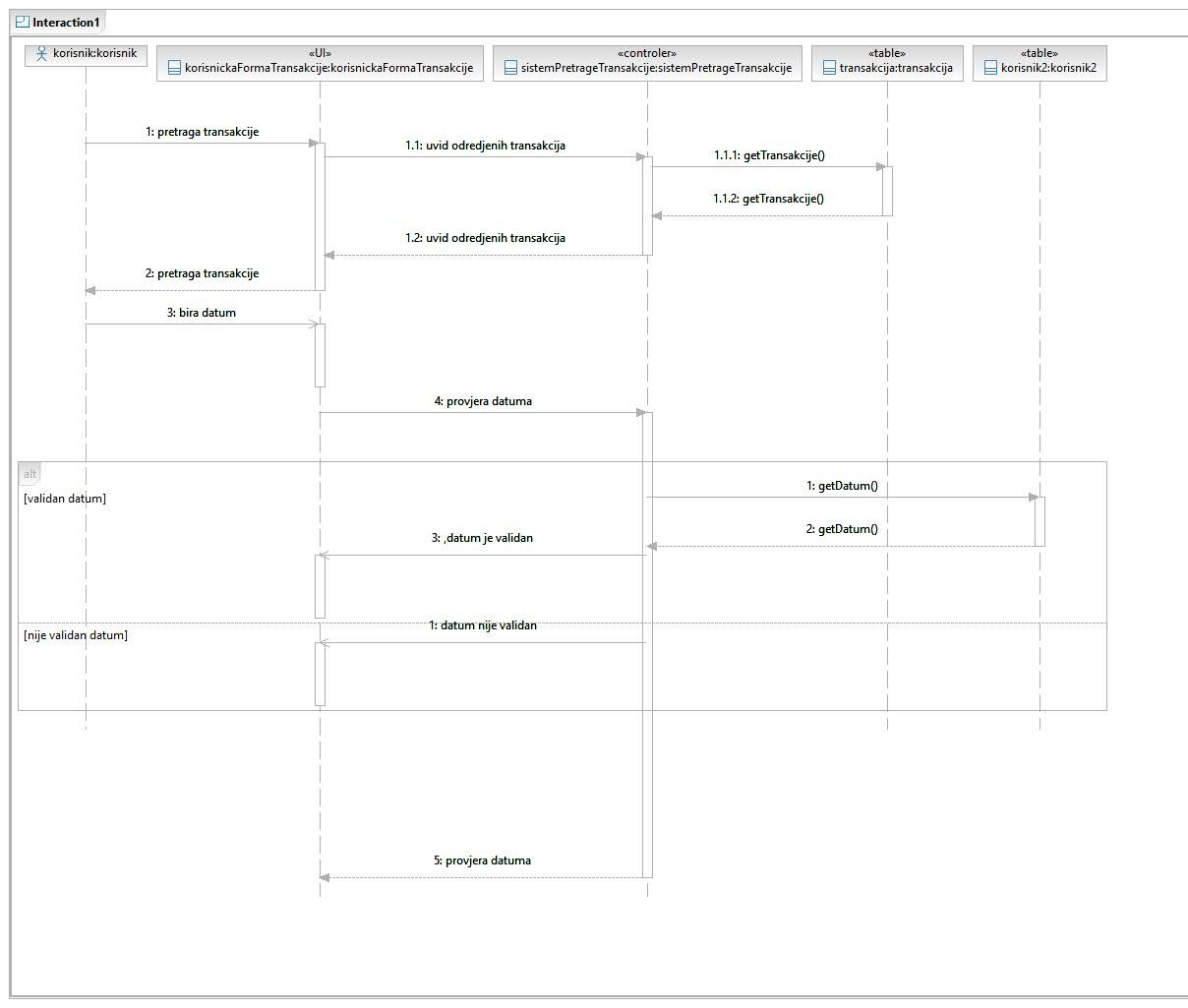
## **4.2. Analiza slučajeva korišćenja**



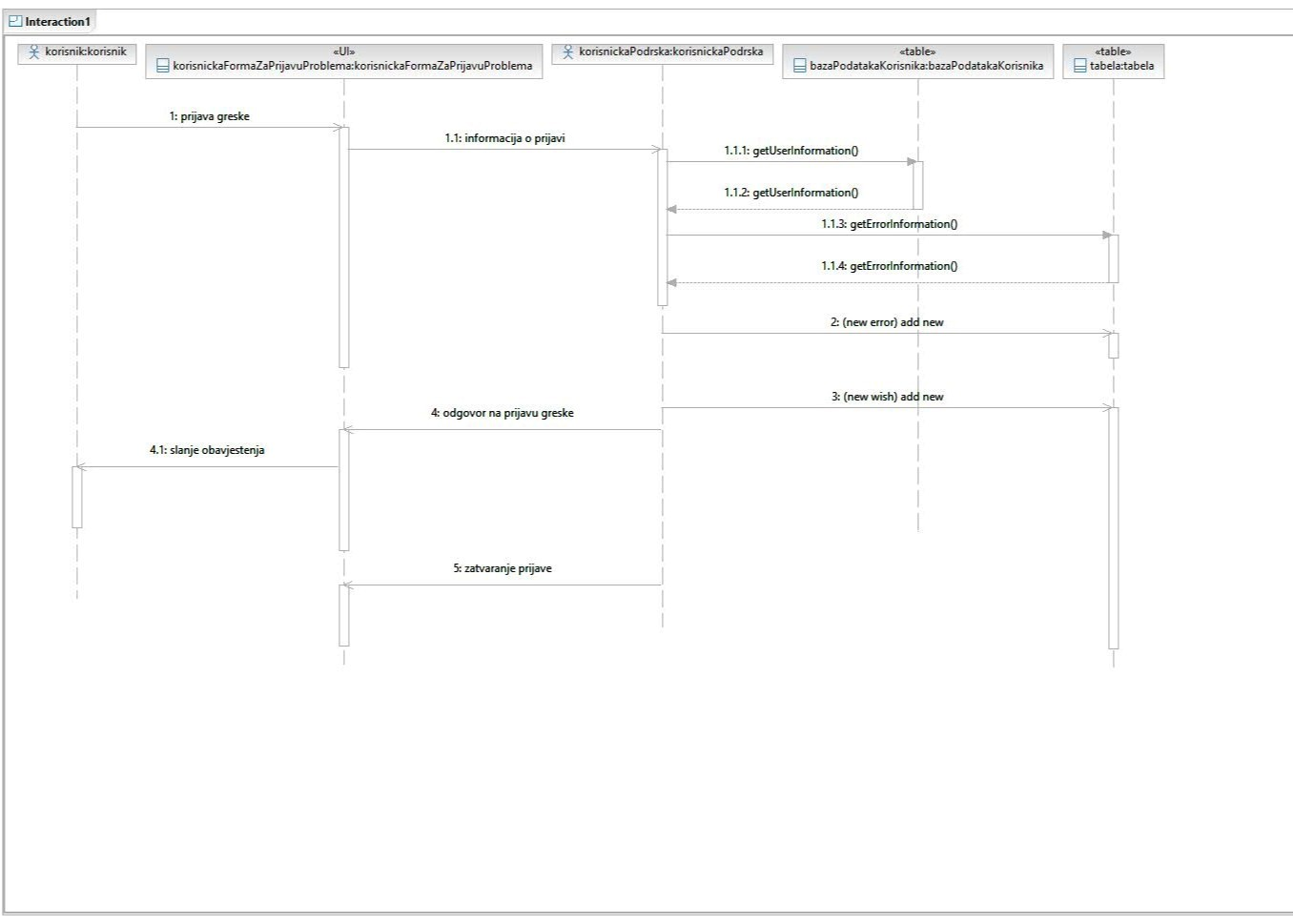
Slika 7. Diagram sequence za use case “pregled mape”



Slika 8. Diagram sequence za use case “katalog akcije”



Slika 9. Diagram sequence za use case “transakcije”



Slika 10. Diagram sequence za use case “edit profile”

Slika 11. Logički dijagram klase

## **5. Projektovanje softvera**

Design Pattern je noviji pokret u softverskom inženjerstvu. Osnovni koncept je ponovno korišćenje znanja o dizajnu. Dizajnerski obrazac je žanr kojim se standardno, stručno testirano rešenje kanonskih problema dizajna rigorozno dokumentuje. Primena dizajn paterna u aplikaciji PetrolPlus može značajno poboljšati strukturu koda, održavanje i ekstenzibilnost.

Termin dizajn obrazac bi mogao da se odnosi i na samo rešenje. Takvi obrasci dizajna obično rešavaju problem dizajna nametanjem nove organizacije u softveru i rasporedom određenog protokola u kojem bi pojedinačni računarski entiteti trebalo interakciju.

Usklađivanjem sa takvim ograničenjem, cilj program bi postao lakši za održavanje. Iteratori se mogu se naći u svim objektno orijentisane biblioteke klasa kontejnera. Definiše protokol pomoću kojeg klijentski potprogrami mogu pristupiti unutrašnjem delu agregatnog objekta.

Dizajnerski obrasci mogu se neformalno prikazati kao kutije sa vezama među njima. Popularna notacija je dijagram klasa/objekat poznat kao OMT dijagram (Tehnika modeliranja objekata).

**Bridge method**

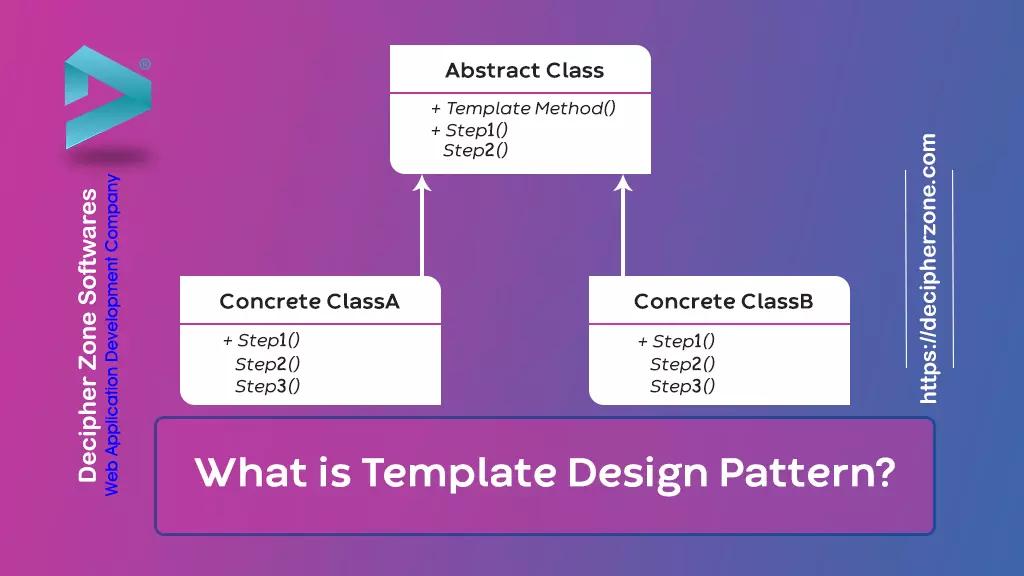
The bridge method je dobar primer primene principa razdvajanja zabrinutosti u ovom kontekstu. Motivisan je željom da se odvoji interfejs, i zatim implementacija interfejsa, iz apstrakcije.Dizajn koji je rezultat primene šablona za dizajn mosta je dva ortogonalna hijerarhije klasa koje mogu da variraju nezavisno.

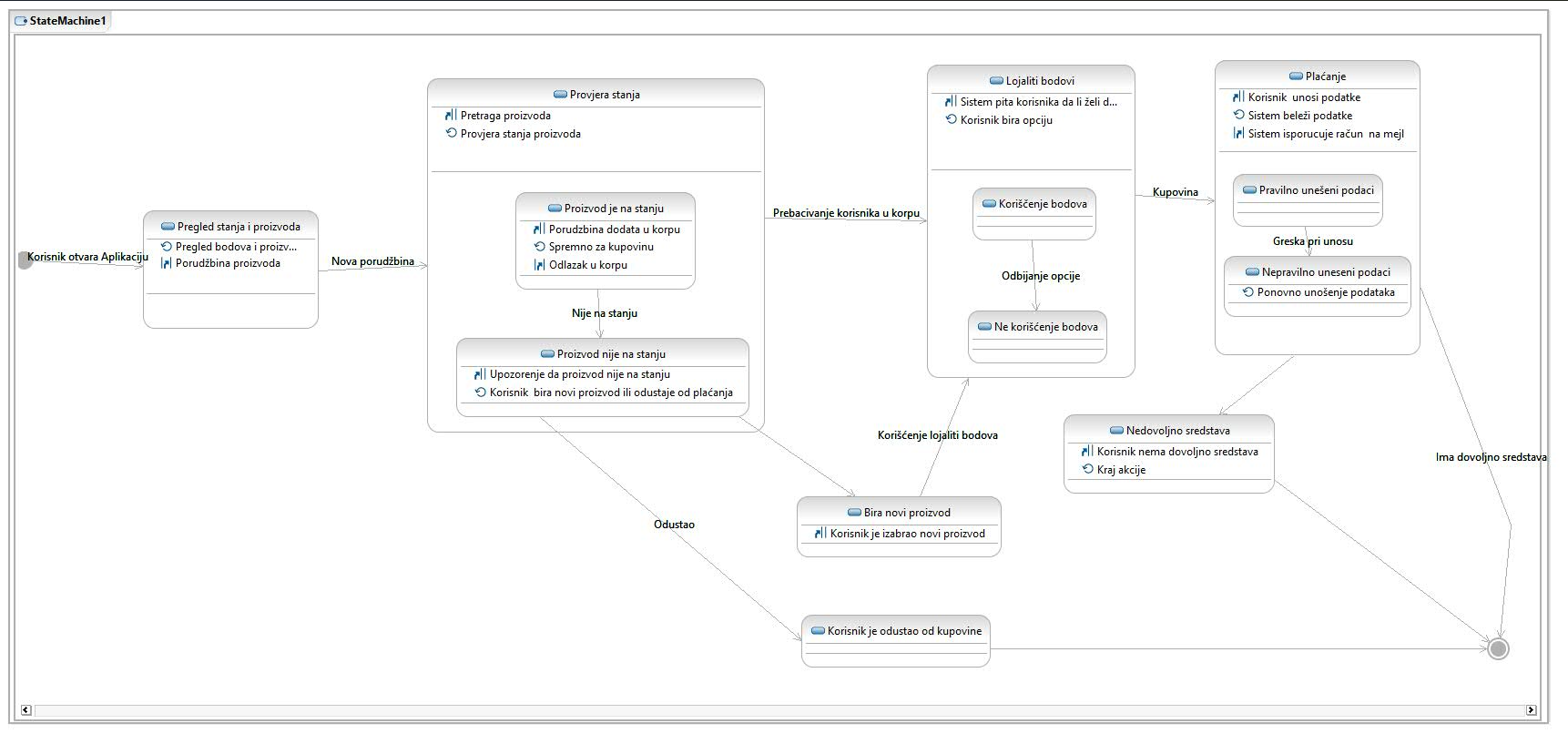


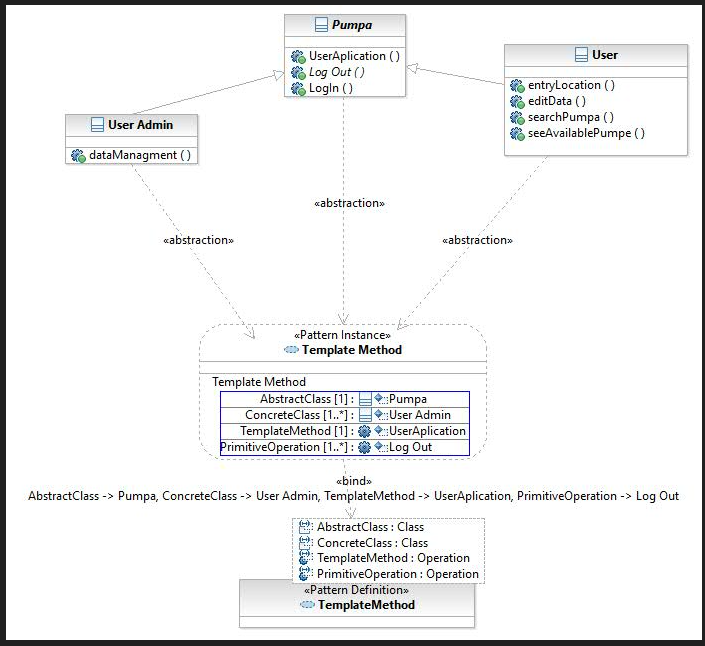
**Template method**

Templete method vam omogućava da definišete skelet analgoritam u operaciji i odložiti neke od koraka na podklase. Metod šablona omogućava redefinisanje podklasa određene korake algoritma bez promene struktura algoritma.

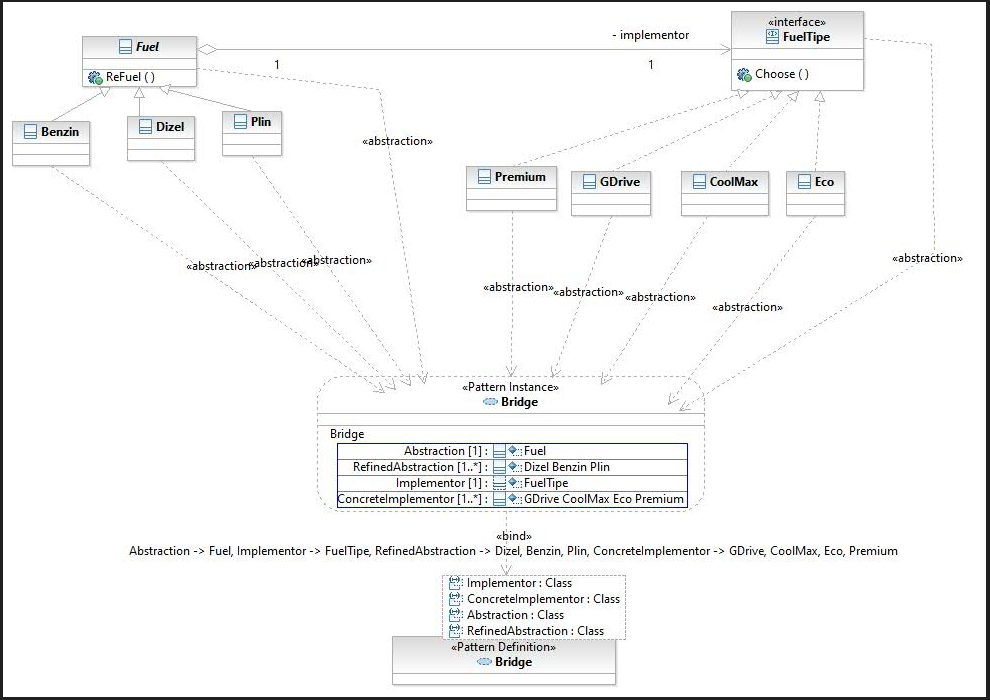
Imate apstraktnu klasu koja je osnovna , a hijerarhiju i ponašanje koje je zajedničko svim objektima u hijerarhiji se implementira u apstraktnoj klasi i ostali detalji su prepušteni pojedinim podklasama.



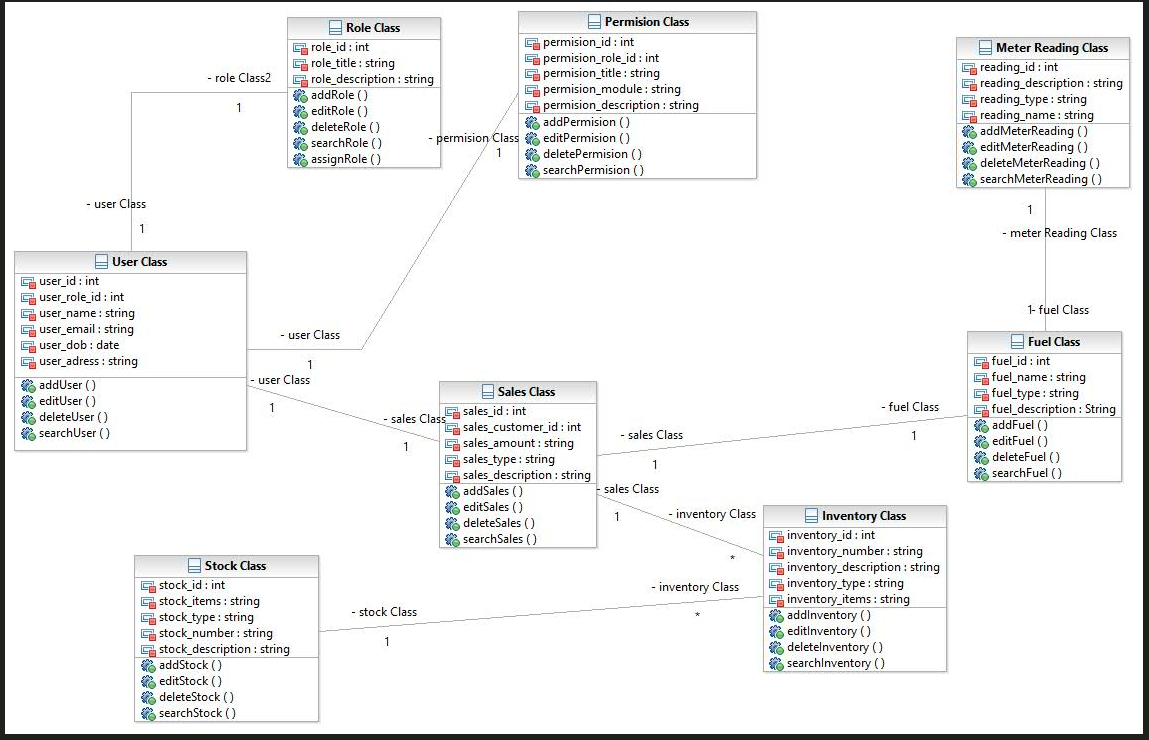
*Slika 12. Dijagram stanja*



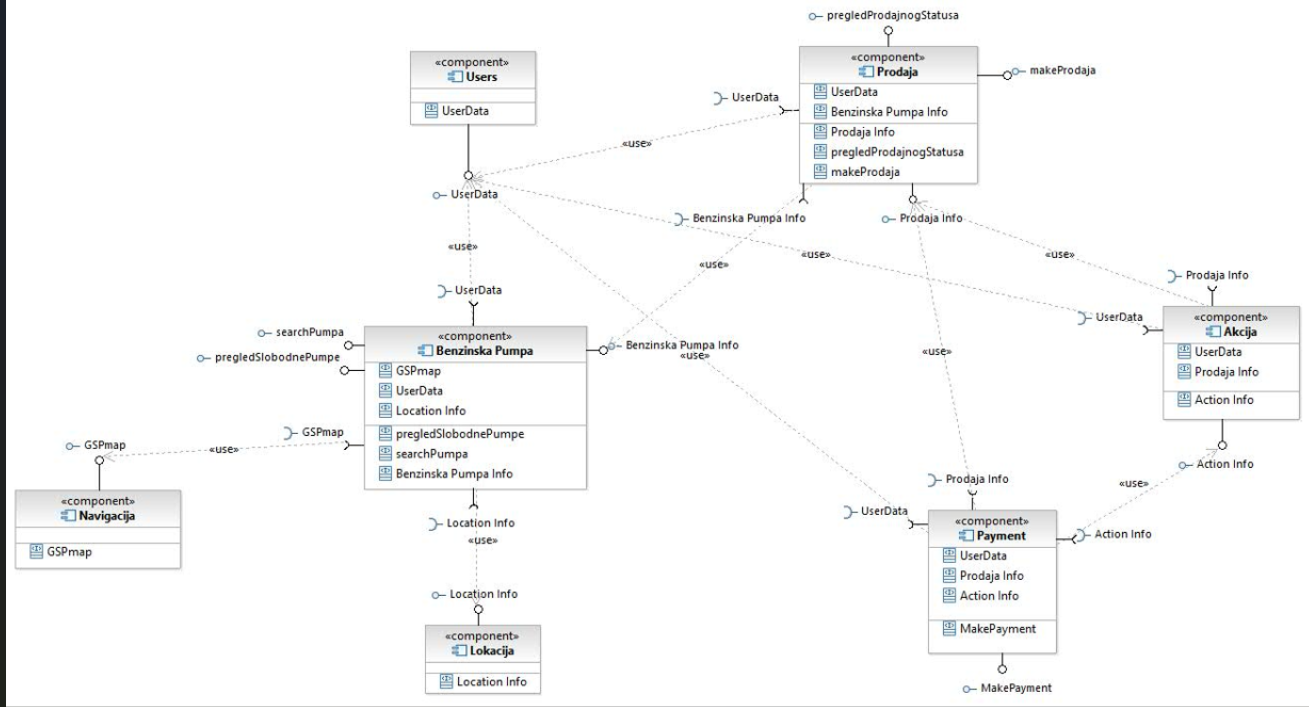
Slika 13. Design pattern – Templete Method



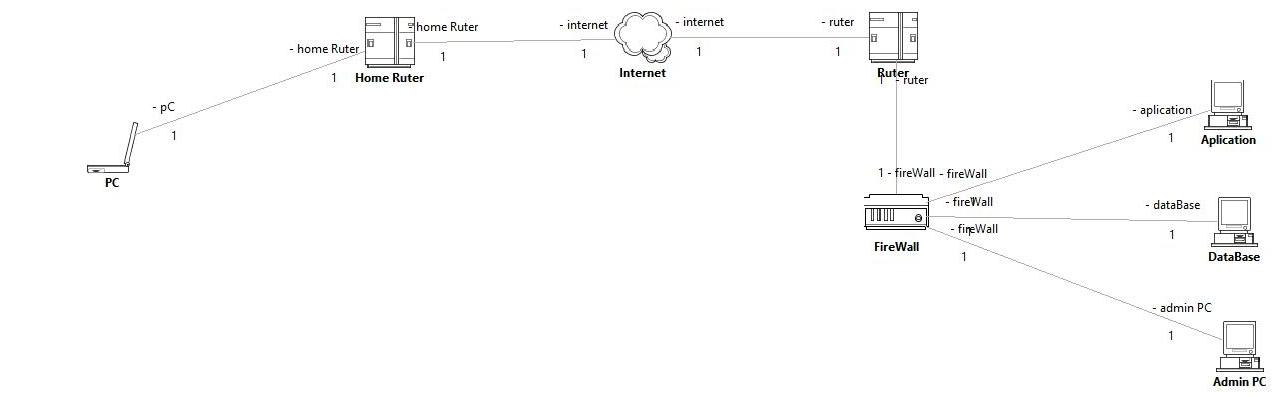
Slika 14. Design pattern – Bridge



Slika 15. Dijagram klasa



Slika 16. Dijagram komponenti



Slika 17. Dijagram uvođenja

## **6. Zaključak**

Glavna prednost aplikacije “PetrolPlus” se odnosi na brzinu i dostupnost informacija. Potrošačima je omogućeno da sami biraju, planiraju i rezervišu svoje usluge

Takodje prednost mobilnog poslovanja je što podrazumeva niže troškove poslovanja i može se odvijati bilo gde uz primenu interneta.

Ova aplikacija je dizajnirana na način da bude dostupna svim korisnicima, i da omogućava lakše i brže korišćenje aplikacije.

Moguće je optimizovati i ubrzati radne procese i smanjiti mogućnost ljudske greške, jer ceo process je automatizovan i zasniva se na elektronskim sklopovima.

Razlikujemo nekoliko glavnih prednosti koji proizilaze iz upotebe naše aplikacije: brzina, sakupljanje informacija o pojedinačnim korisnicima, uloga potrošača u promociji proizvoda i usluga, mogućnost izbora pojedinaca koji pristupaju informacijama i kreiranje profila potrošača, mogućnost da se informacije pružaju u svakom trenutku kada ih potrošač zatraži…

Nedostatak kod aplikacije bio bi taj što se korisnici prilikom planiranja putovanja mogu previše osloniti na aplikaciju, što bi kasnije dovelo do razočarenja prilikom putovanja i samim tim krivac bi bila aplikacija.

## **7. Literatura**

1. <https://www.geeksforgeeks.org/mvc-design-pattern/>

2. <https://jbf.ekofis.ues.rs.ba/images/2021/ZR_JPF2021/S2_Ilic.pdf>

3. <https://cdn.aaai.org/Workshops/1998/WS-98-03/WS98-03-011.pdf>

4. <https://ima.udg.edu/~sellares/EINF-ES1/TemplateMethodToni.pdf>