



UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET SARAJEVO



DOKUMENTACIJA PROJEKTA

PREDMET: OBJEKTNO ORIJENTISANA ANALIZA I DIZAJN
STUDENTI: ILMA SPAHIĆ
SELMA VUČIJAK
FARUK SMAJLOVIĆ

GRUPA: 10

MENTOR: JASMINA BAJRAMOVIĆ



OPIS TEME:

Pošta je ustanova u kojoj se stvaraju velike gužve zbog mnogo interakcije sa ljudima. Zbog toga smo osmislili softversko rješenje koje će simulirati njen rad, prije svega da bi se olakšala organizacija i ubrzale njene funkcionalnosti, ali i da bismo modernizovali njene usluge i bili u korak sa vremenom.

Kako bi se to ostvarilo potrebna je aplikacija koja će istovremeno biti efikasna i jednostavna za korištenje. Cilj je olakšati potrošačima plaćanje računa, slanje paketa, te kupovinu razglednica i markica. Poštariima će sve informacije o paketima i ostaloj pošti biti na dohvat ruke, što će im uveliko olakšati i ubrzati posao.

Uposlenicima će rad biti automatizovan što će smanjiti mogućnost greške.



PROTOTIP

Planirano je korištenje adaptivnog layouta, te implementacija Viewova korištenjem XAML.

Posta

001 000

E-mail

Password

Log in

Nemate account?

Registruj se

003 002

Pregled racuna

Pracenje paketa

Shop

Log out

Posta



006 003

Nazad

Podaci o racunima:

ID: 1 Cijena: 1 Stanje: Placeno

ID: 2 Cijena: 2 Stanje: Nije placeno

Posta



003 001

Razglednice Markice



Adresa za dostavu:

Kupi

004 002

Nazad

Uposlenik:

Ime

ilma

Datum
rodjenja:

6/6/2018



Prezime:

spahic

Adresa:

malta

E-mail:

ispahic

Password:

...

Tip posla:



Postar



Salter

REGISTRACIJA

004 001

Nazad

Pretraga potrosaca:

2307997175013

Pretraga

Prikaz racuna

Obrisi potrosaca

Podaci potrosaca

Posta

002 001

Nazad

Ime:

Selma

Prezime:

Vucijak

Adresa:

Logavina

E-mail:

selmav

JMBG:

2307997175013

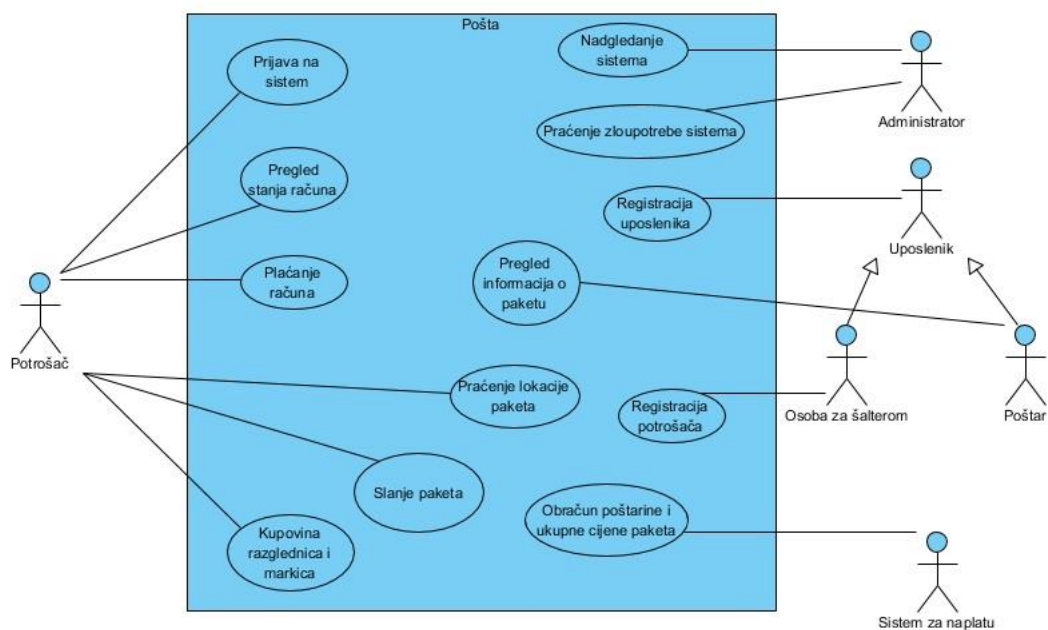
Broj telefona:

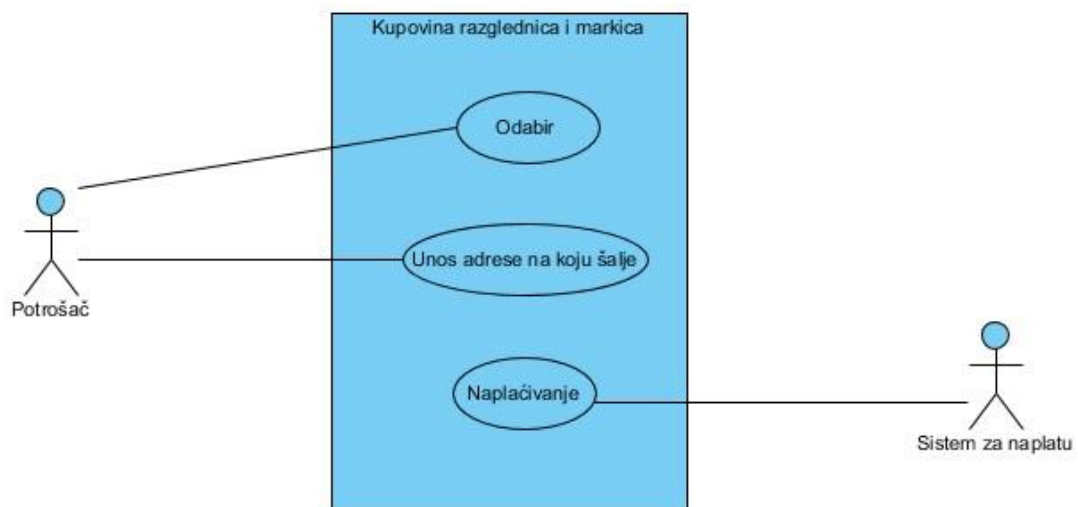
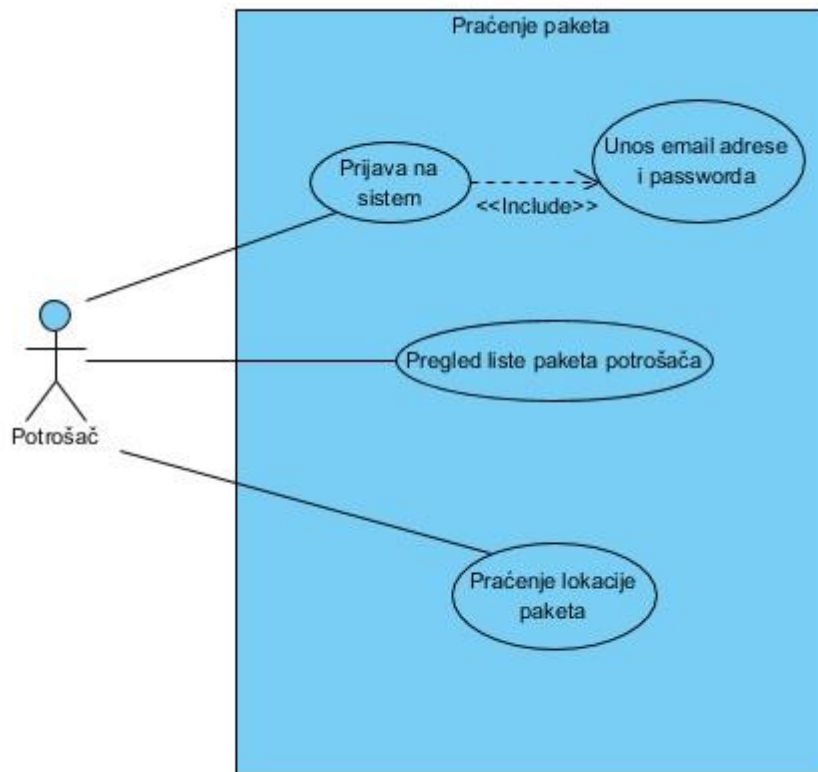
062316446

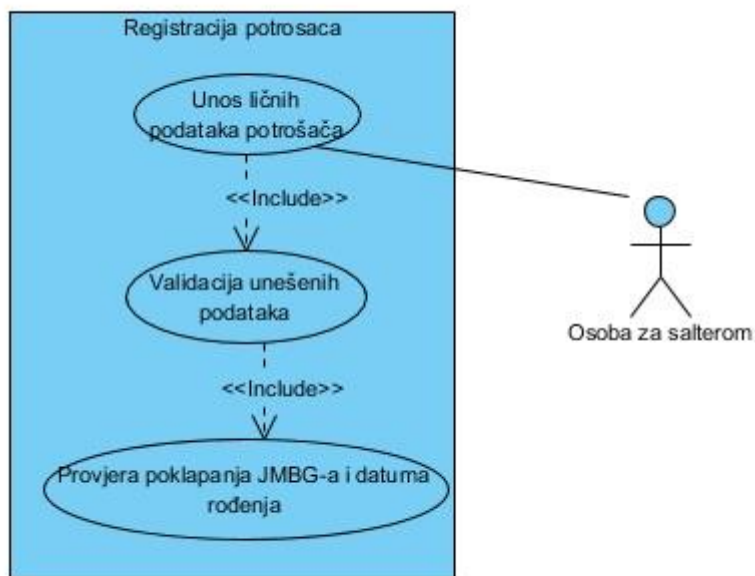
DIJAGRAMI

Dijagrami su rađeni u Visual Paradigm.

USE CASE

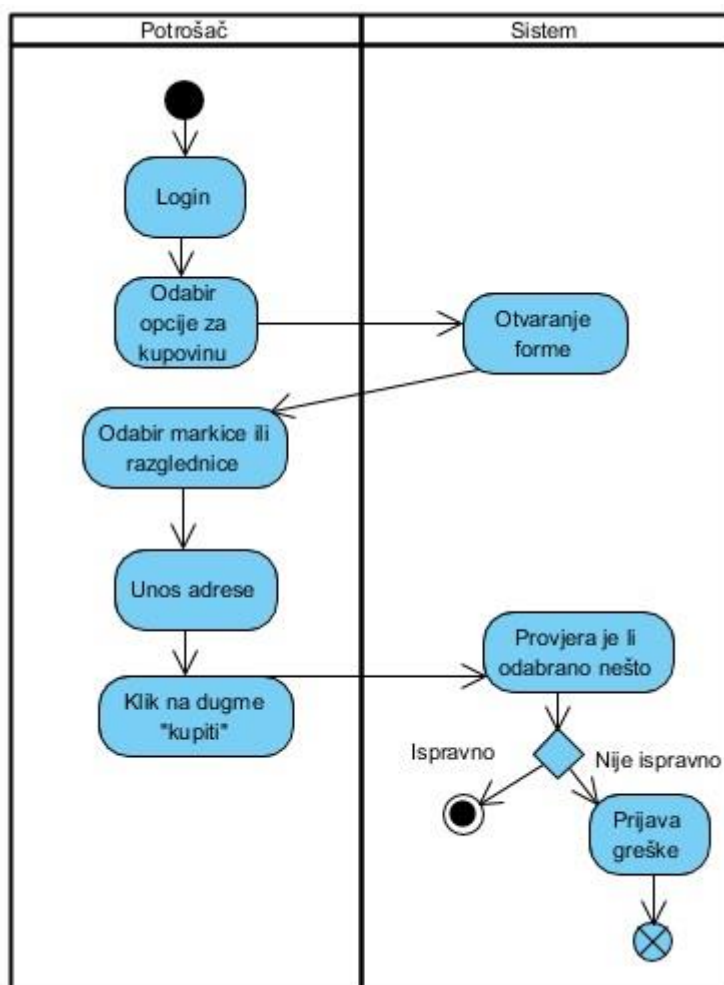


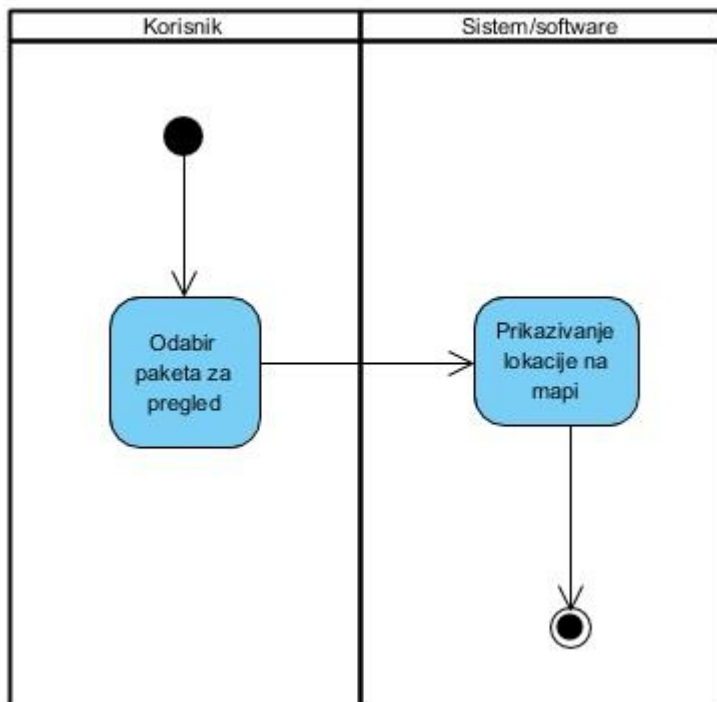
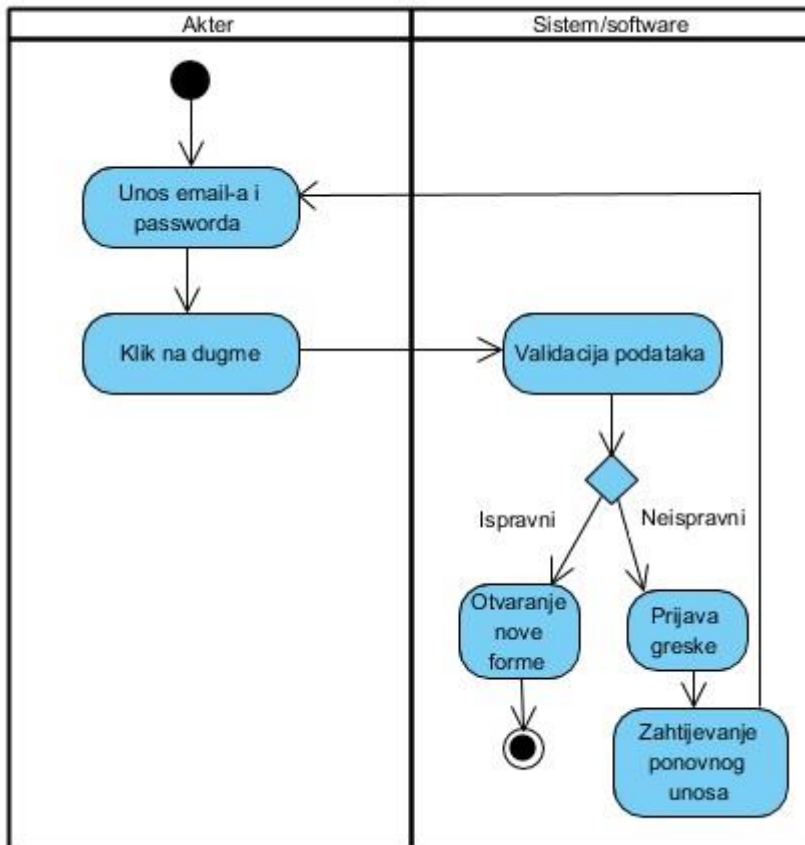


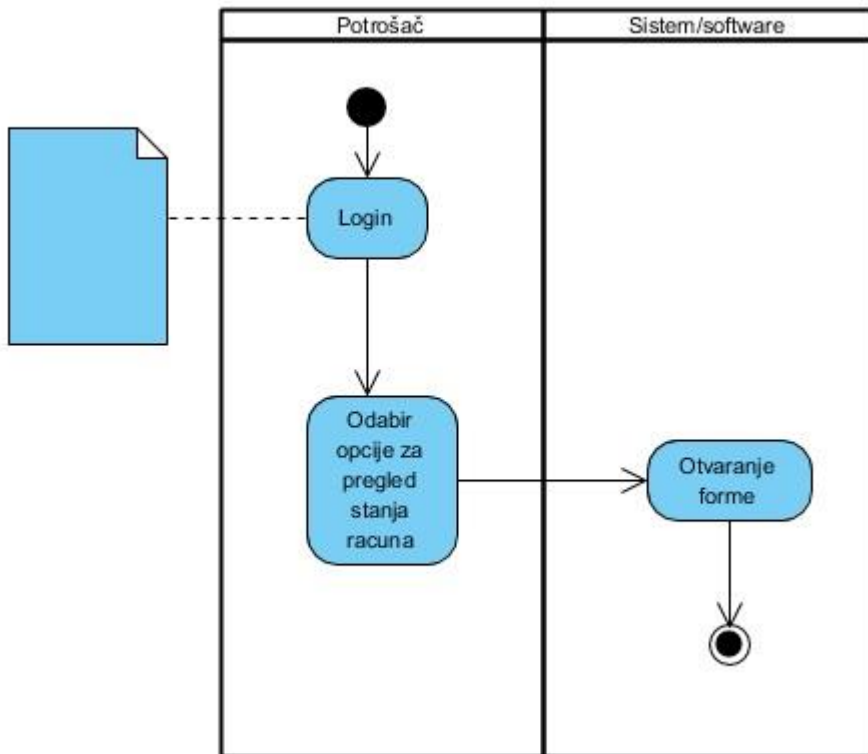
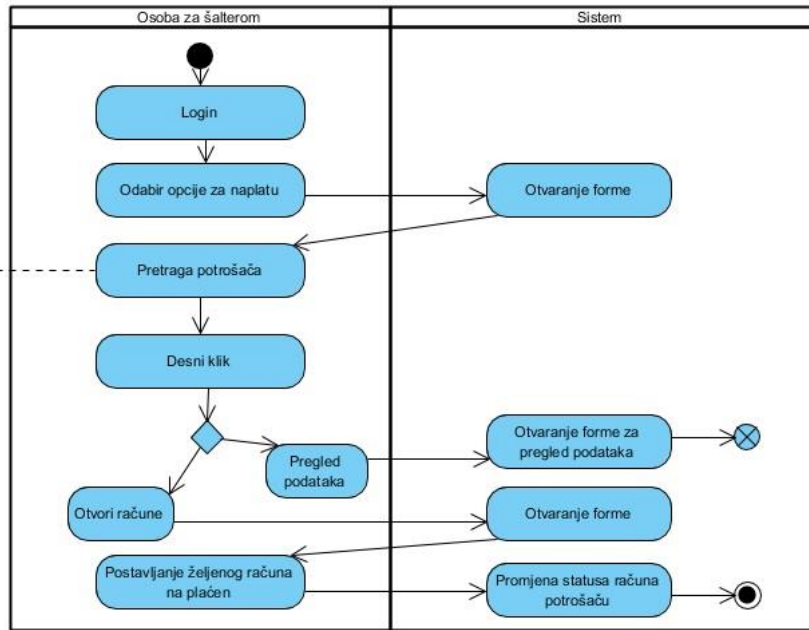
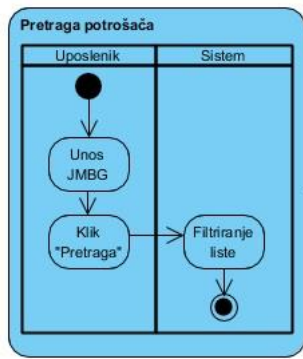


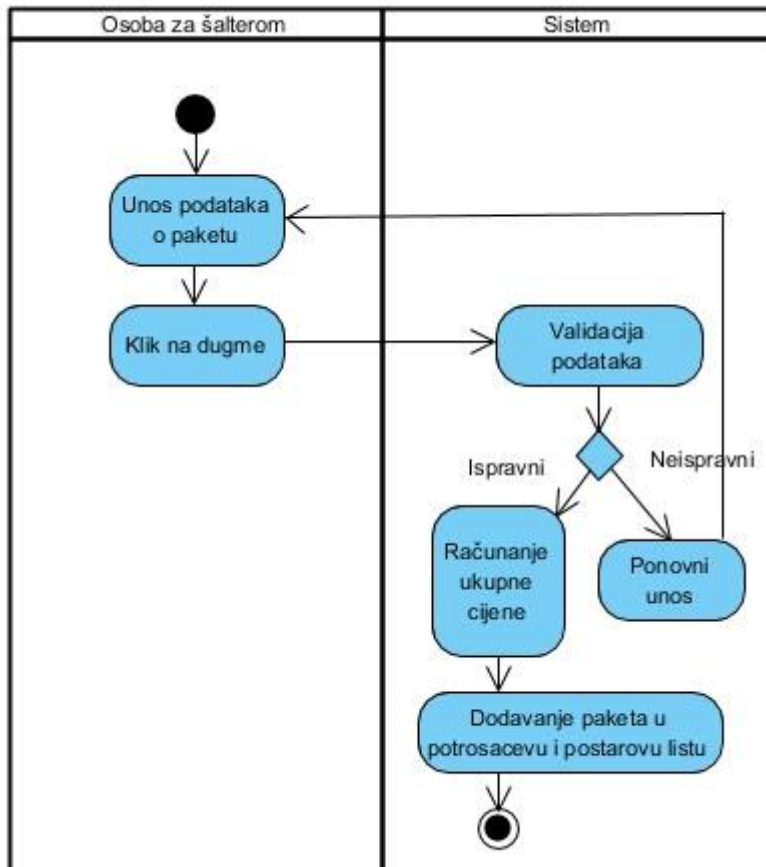
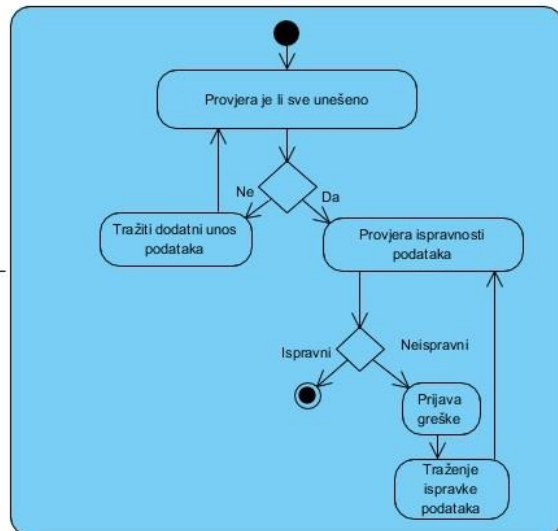
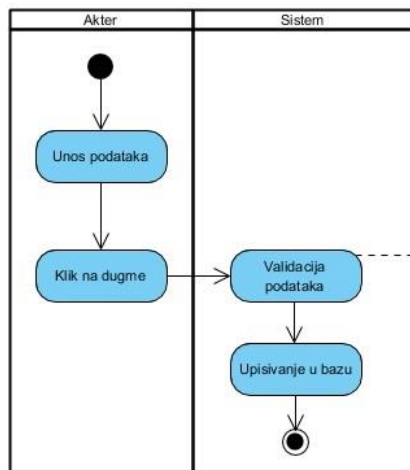
Radili: Selma Vučijak, Ilma Spahić, Faruk Smajlović

ACTIVITY





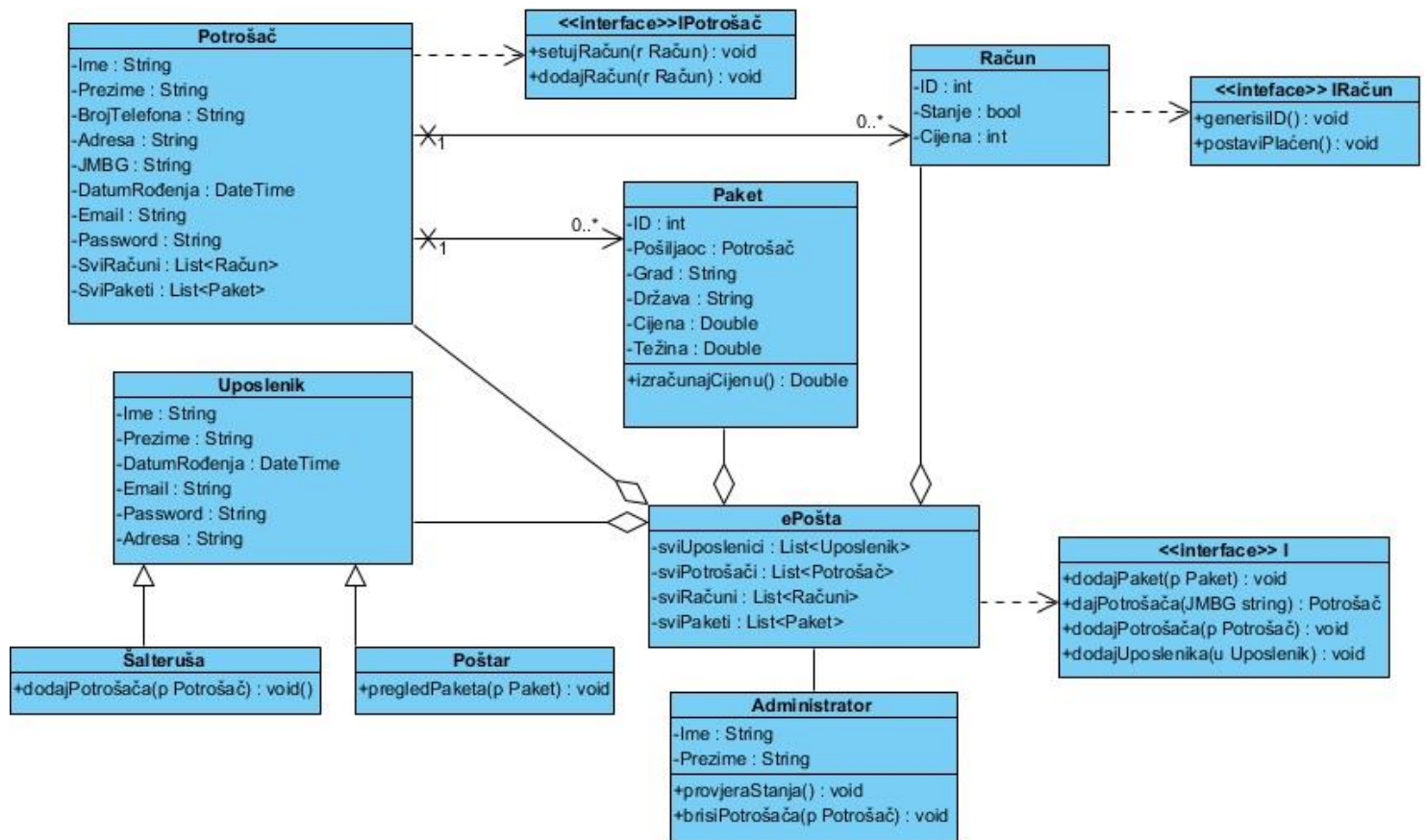




Radili: Selma Vučijak, Ilma Spahić, Faruk Smajlović



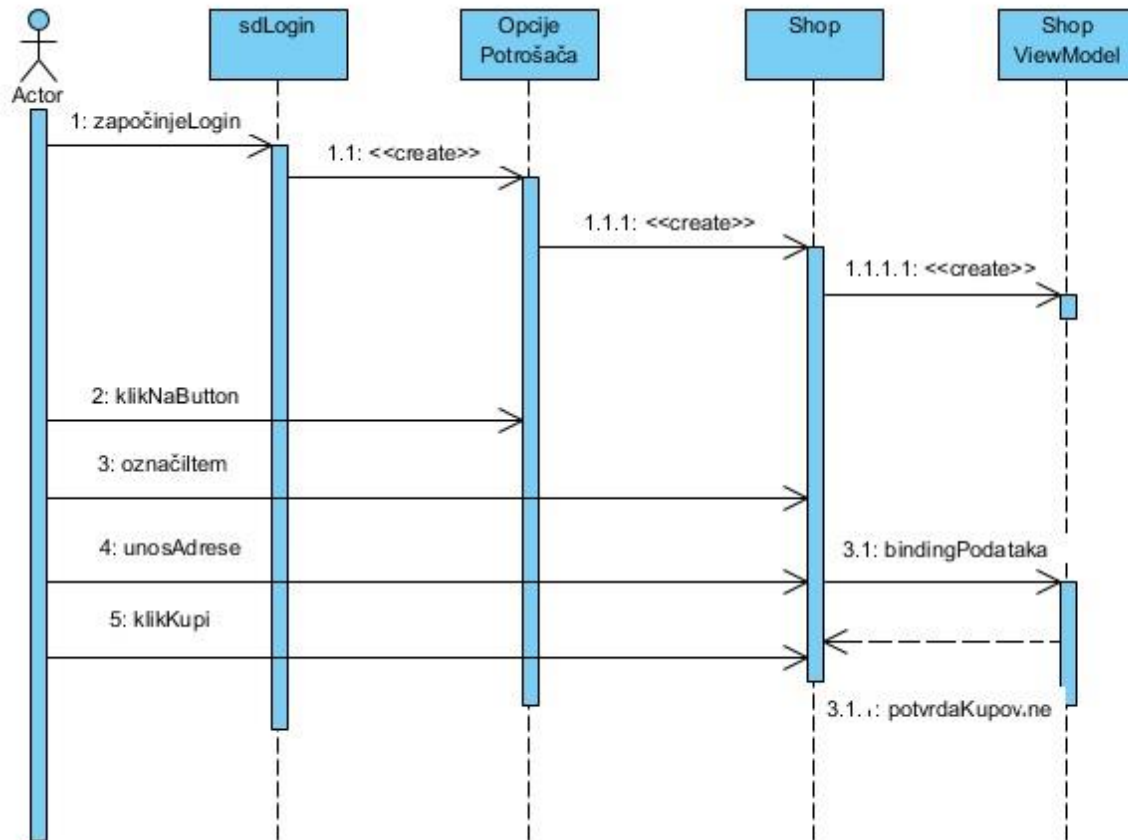
CLASS

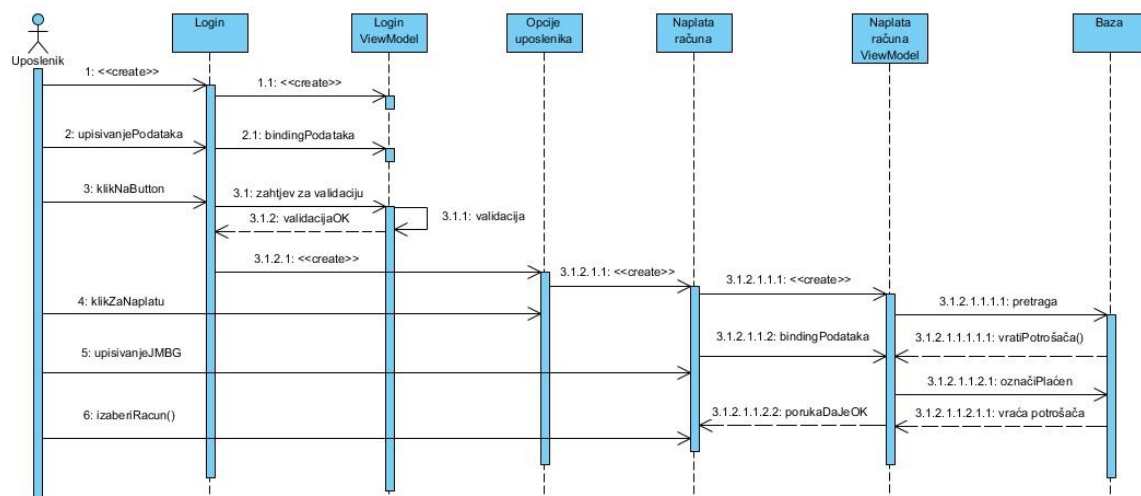
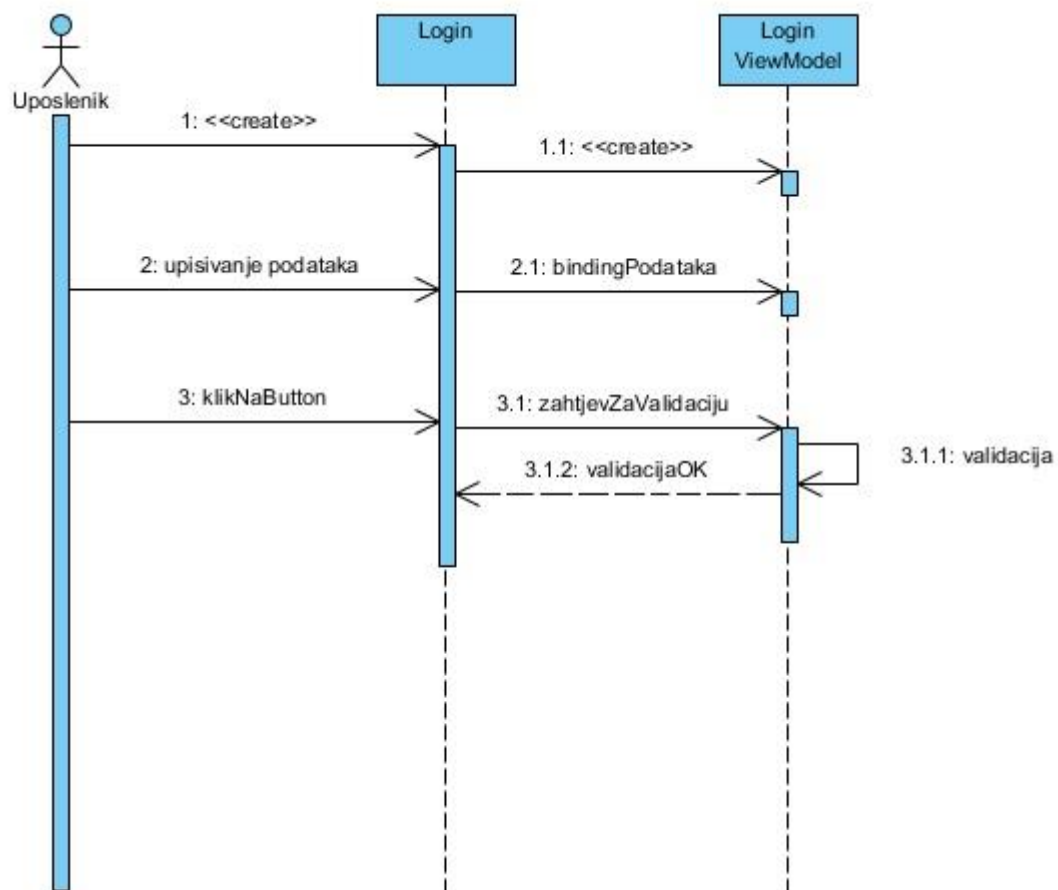


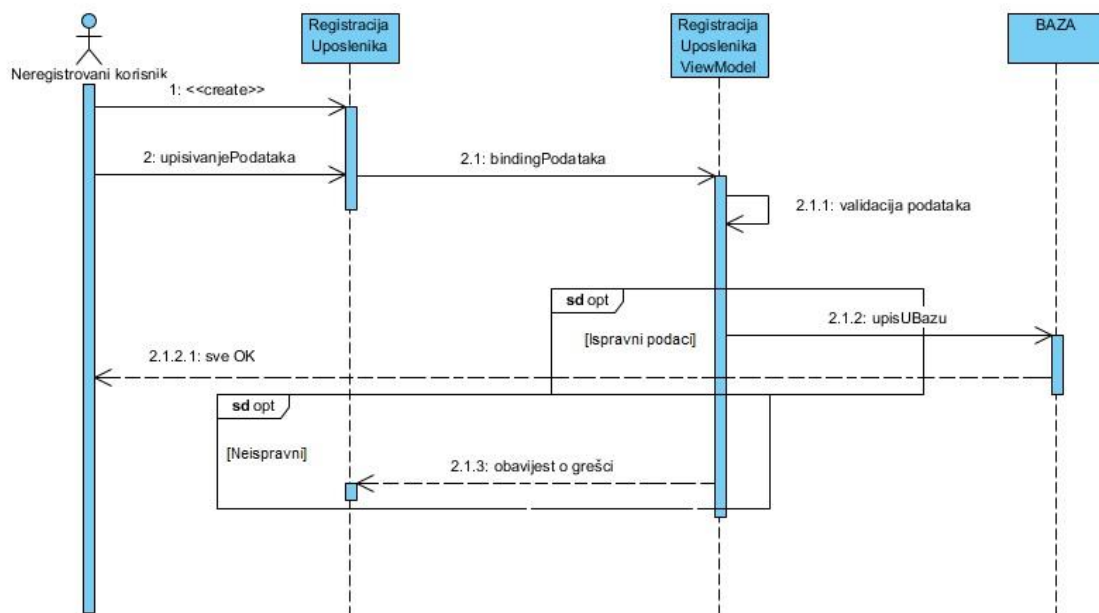
MVVM na [linku](#)

Radili: Selma Vučijak, Ilma Spahić, Faruk Smajlović

SEQUENCE

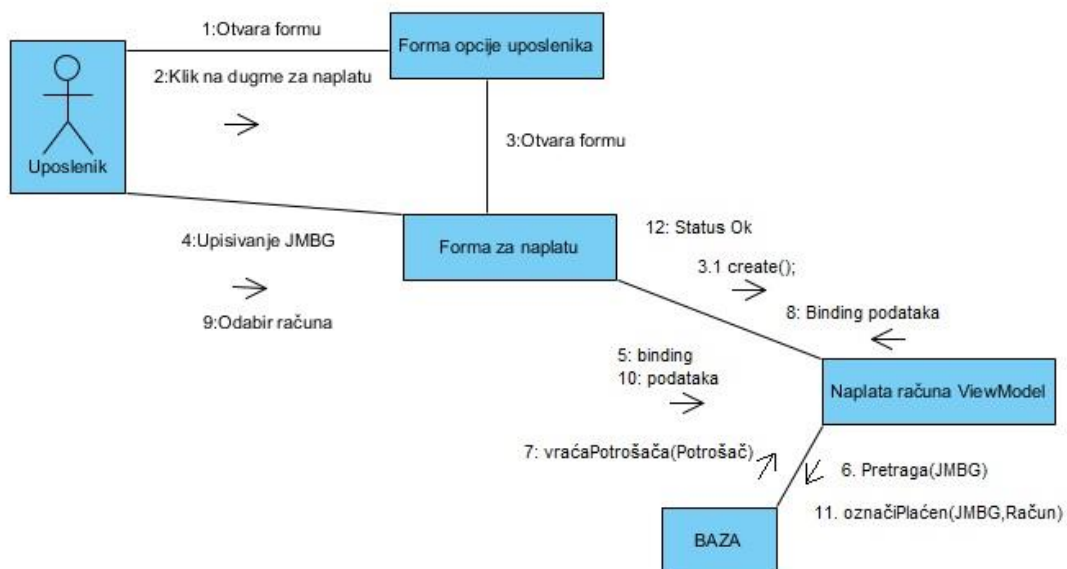


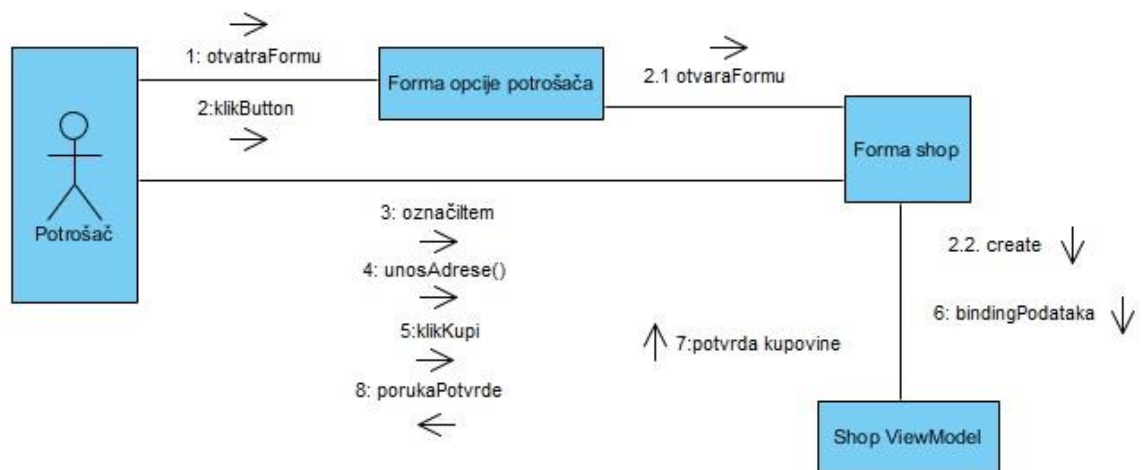
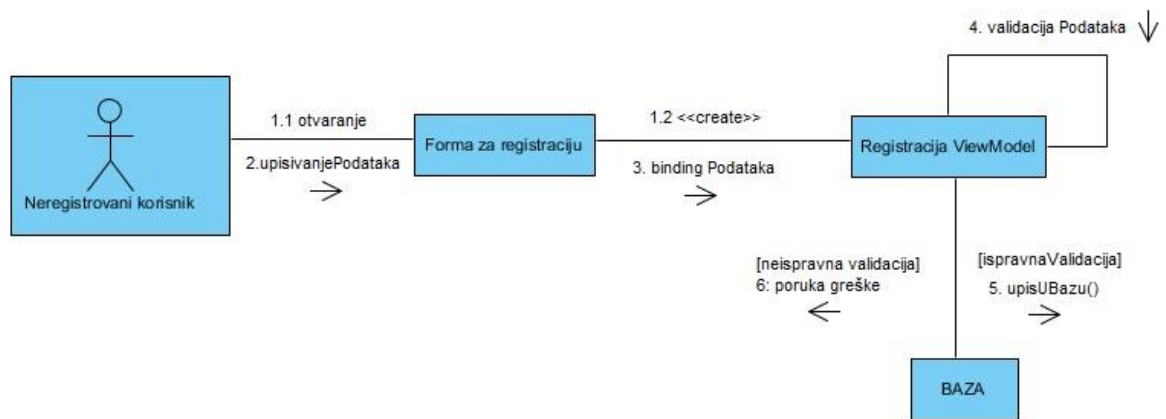




Radili: Selma Vučijak, Ilma Spahić, Faruk Smajlović

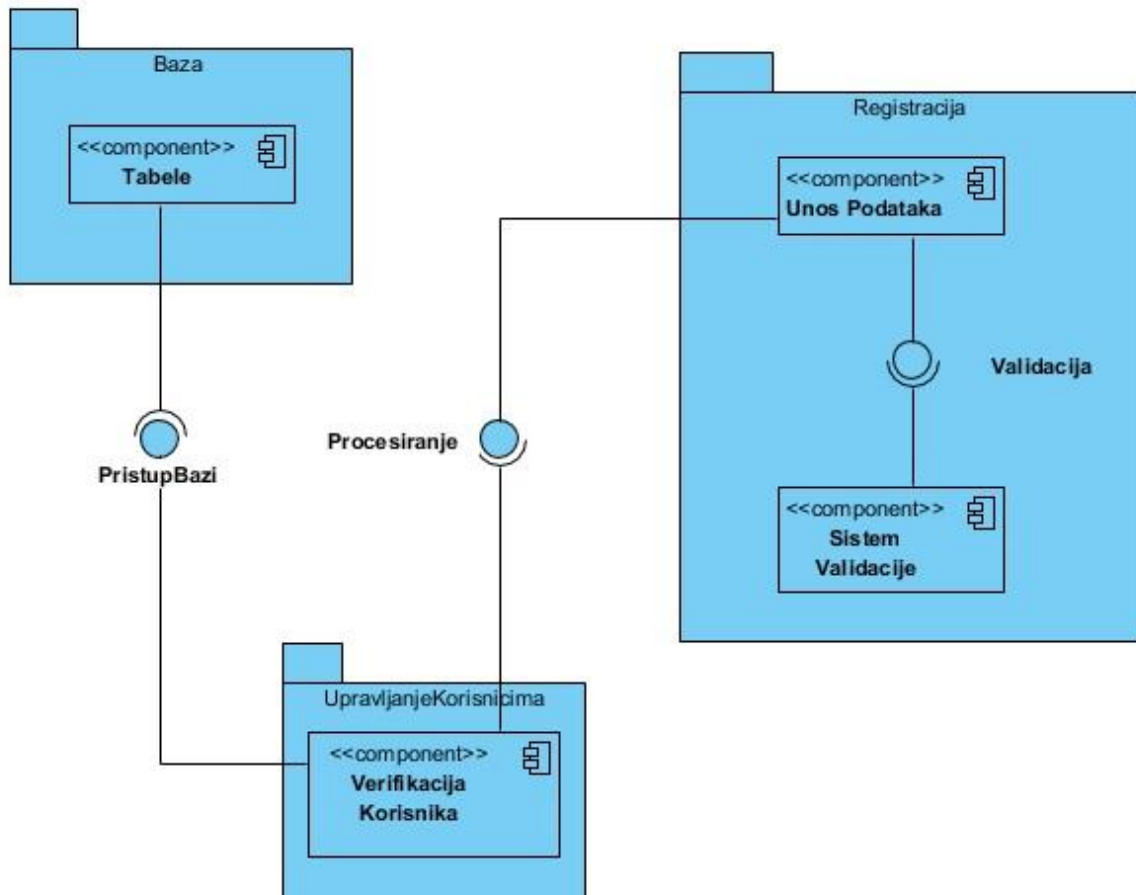
COMMUNICATION





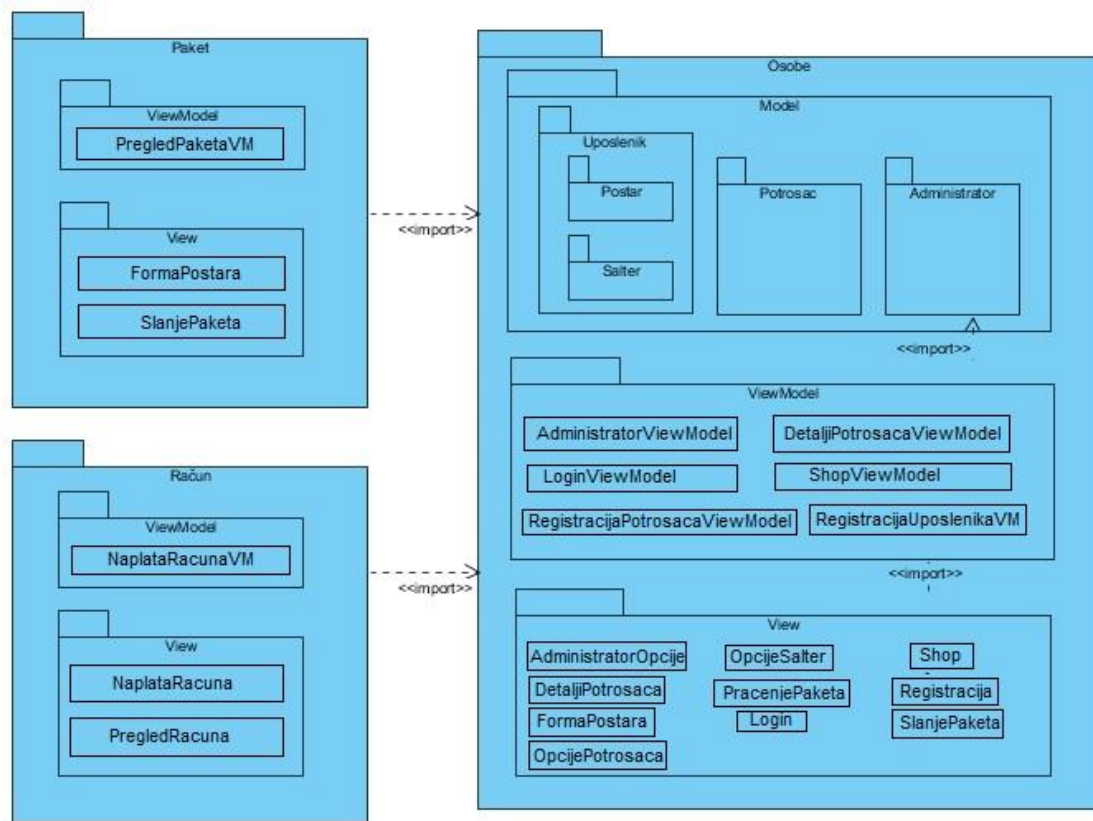
Radili: Selma Vučijak, Ilma Spahić, Faruk Smajlović

COMPONENT



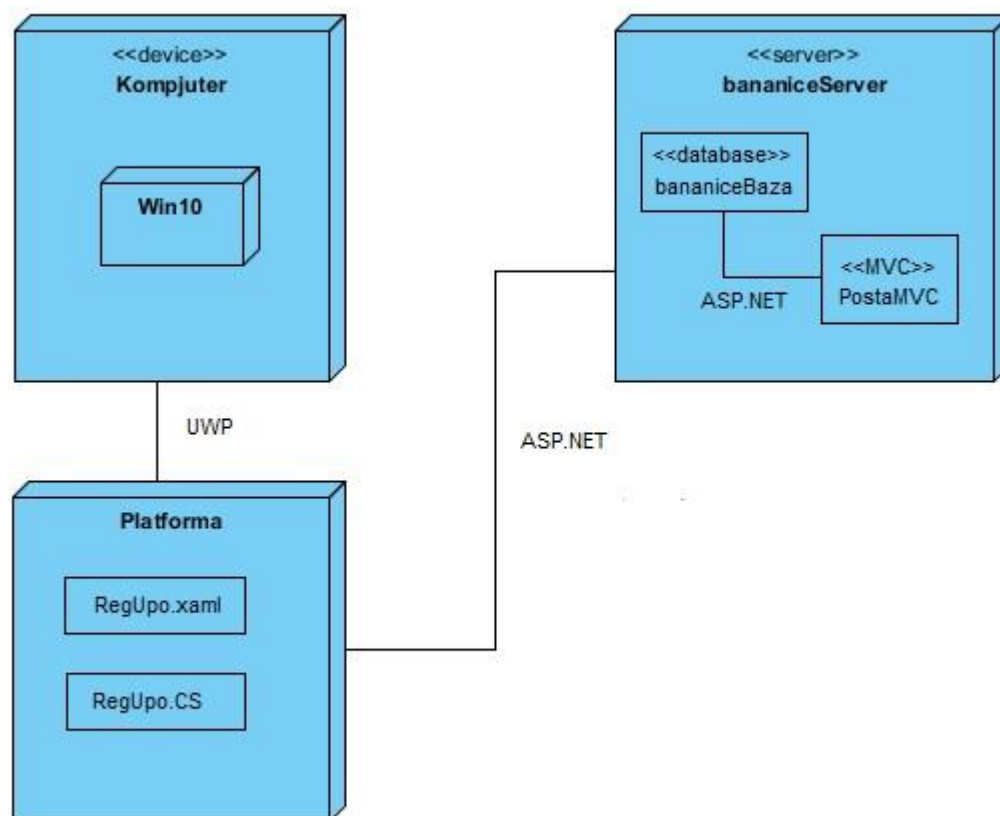
Radili: Selma Vučijak, Ilma Spahić, Faruk Smajlović

PACKAGE



Radili: Selma Vučijak, Ilma Spahić, Faruk Smajlović

DEPLOYMENT



Radili: Selma Vučijak, Ilma Spahić, Faruk Smajlović

ANALIZA I PRIJEDLOZI UPOTREBE PATERNA

Design Patterns

1. Singleton - Kreacijski pattern

Uloga Singleton patterna je da osigura da se klasa može instancirati samo jednom i da osigura globalni pristup kreiranoj instanci klase.

Postoji više objekata koje je potrebno samo jednom instancirati i nad kojim je potrebna jedinstvena kontrola pristupa.

Pattern smo iskoristili u našem projektu kod klase za rad sa bazom jer nam je potrebna samo jedna klasa u citavom projektu.

2. Factory Method - Kreacijski pattern

Uloga Factory Method patterna je da omogući kreiranje objekata na način da podklase odluče koju klasu instancirati. Različite podklase mogu na različite načine implementirati interfejs. Factory Method često se primjenjuje jer dopušta projektu da blisko slijedi SOLID principe.

Factory Method instancira odgovarajuću podklasu (izvedenu klasu) preko posebne metode na osnovu informacije od strane klijenta ili na osnovu tekućeg stanja.

Ovaj pattern nismo koristili u našem projektu, ali smo ga mogli iskoristiti prilikom registracije uposlenika obzirom da imamo dvije vrste uposlenika, osobu za salterom i postara.

3. Command - Pattern ponasanja

Razdvaja klijenta koji zahtjeva operaciju i omogućava slanje zahtjeva različitim primateljima. Komanda, to je objekt koji zna primaoca i poziva njegovu metodu.

Primalac (receiver) kada se pozove metoda execute() u komandi, izvršava. Pozivatelj zna kako izvršiti komandu, ali ne zna ništa o konkretnoj komandi, zna samo o njenom interfejsu.



Objekti pozivatelja i primaoca se trže u klijent objektu, a klijent odlučuje šta se iz primaoca šalje komandi, a koje komande se dodjeljuju pozivatelju, odlučuje koje komande se izvršavaju u kojem trenutku. Da bi izvršio komandu, mora proslijediti komandu pozivatelju. Ovaj tip patterna nismo iskoristili u našem projektu, ali smo mogli prilikom odjavljivanja korisnika.

4. Decorator pattern- Strukturalni pattern

Osnovna namjena Decorator patterna je da omogući dinamičko dodavanje novih elemenata i ponašanja (funktionalnosti) postojećim objektima. Objekat pri tome ne zna da je uradjena dekoracija što je veoma korisno za iskoristljivost i ponovnu upotrebu komponenti softverskog sistema. Ovaj tip patterna nismo iskoristili u našem projektu. Mogli smo ga iskoristiti prilikom registracije. Ukoliko se registruje uposlenik, da se doda odabir tipa posla: Osoba za salterom ili postar.

5. Observer Pattern

Uloga Observer patterna je da uspostavi relaciju između objekata tako kada jedan objekat

promijeni stanje, drugi zainteresirani objekti se obavještavaju..

Ovaj pattern nismo koristili u našem projektu, ali smo ga mogli koristiti prilikom implementacije funkcionalnosti praćenja paketa, ukoliko je paket stigao na adresu da korisnik dobije obavještenje.

6. Interpreter Pattern

Interpreter pattern služi za evaluiranje gramatike nekog jezika i jezičkih izraza.

Obično se koriste TerminalExpression i CompoundExpression klase koje služe za rekurzivno rješavanje problema interpretacije nekog jezičkog izraza gdje je CompoundExpression neki dio pravila koji poziva drugo pravilo za obradu izraza, dok je TerminalExpression bazni slučaj za neko pravilo.

Nismo koristili ovaj pattern u našem projektu. Možemo ga iskoristiti za provjeravanje validnosti JMBG broja i datuma rođenja, ili provjere dužine passworda.

7. Adapter Pattern

Osnovna namjena Adapter patterna je da omogućiti širu upotrebu već postojećih klasa. Adapter pattern kreira novu adapter klasu koja služi kao posrednik između originalne klase i željenog interfejsa. Tim postupkom se dobija željena funkcionalnost bez izmjena na originalnoj klasi i bez ugrožavanja integriteta cijele aplikacije.

Nismo koristili ovaj pattern unutar našeg projekta zbog već dobro osmišljenih klasa koje ne zahtijevaju promjene.

8. Proxy Pattern

Namjena Proxy patterna je da omogućiti pristup i kontrolu pristupa stvarnim objektima. Proxy je obično mali javni surogat objekat koji predstavlja kompleksni objekat čija aktivizacija se postiže na osnovu postavljenih pravila. Proxy pattern rješava probleme kada se objekat ne može instancirati direktno. Ovaj pattern nije iskoristen, ali se mogao iskoristiti za restrikciju brisanja potrošača svima osim administratoru sistema.

9. Bridge Pattern

Osnovna namjena Bridge patterna je da omogućiti odvajanje apstrakcije i implementacije neke klase tako da ta klasa može posjedovati više različitih apstrakcija i više različitih implementacija za pojedine apstrakcije. Nismo koristili ovaj pattern unutar našeg projekta, ali smo ga mogli iskoristiti za razmjenu poruka između korisnika i sistema.

- Mogućnost prijave na sistem
- Mogućnost registracije novih korisnika sistema
- Mogućnost kupovine razglednica i markica
- Mogućnost plaćanja računa
- Mogućnost pregleda stanja računa
- Mogućnost praćenja lokacije paketa
- Mogućnost podnošenja zahtjeva za slanje paketa
- Mogućnost pregleda svih informacija o paketu
- Mogućnost obračuna poštarine i ukupne cijene paketa

Projektni zadatak	Članovi tima	Implementirano
Adaptivni layout	Selma Vučijak, Ilma Spahić, Faruk Smajlović	Da
MVVM – Views	Selma Vučijak, Ilma Spahić, Faruk Smajlović	Da
MVVM – Models	Selma Vučijak, Ilma Spahić, Faruk Smajlović	Da
MVVM – ViewModels	Selma Vučijak, Ilma Spahić, Faruk Smajlović	Da
Entity framework	Selma Vučijak, Ilma Spahić, Faruk Smajlović	Da
Azure	Selma Vučijak, Ilma Spahić, Faruk Smajlović	Da
Vanjski uređaj	Selma Vučijak, Ilma Spahić, Faruk Smajlović	Ne

SCENARIJ

Scenarij mozete vidjeti na naredna tri linka [1](#), [2](#) , [3](#).