**Internet tehnologije**

**Tema: Društvena mreža**

Student:

Marinković Danko 34/2016

Fakultet Organizacionih Nauka, Beograd

Contents

[Verbalni opis sistema 3](#_Toc45309555)

[Model slučaja korišćenja 4](#_Toc45309556)

[Opis zahteva pomoću modela slučaja korišćenja 5](#_Toc45309557)

[SK1: Slučaj korišćenja – Prijavljivanje na sistem 5](#_Toc45309558)

[SK2: Slučaj korišćenja – Kreiranje novog naloga 6](#_Toc45309559)

[SK3: Slučaj korišćenja – Pretraživanje korisnika 7](#_Toc45309560)

[SK4: Slučaj korišćenja – Slanje zahteva za prijateljstvo 8](#_Toc45309561)

[SK5: Slučaj korišćenja – Prihvatanje ili odbijanje zahteva 9](#_Toc45309562)

[Analiza 10](#_Toc45309563)

[DS1: Dijagram slučaja korišćenja – Prijavljivanje na sistem 10](#_Toc45309564)

[DS2: Dijagram slučaja korišćenja – Kreiranje novog naloga 12](#_Toc45309565)

[DS3: Dijagram slučaja korišćenja – Pretraživanje korisnika 17](#_Toc45309566)

[DS4: Dijagram slučaja korišćenja – Slanje zahteva za prijateljstvo 19](#_Toc45309567)

[DS5: Dijagram slučaja korišćenja – Prihvatanje ili odbijanje zahteva 21](#_Toc45309568)

[Struktura baze 23](#_Toc45309569)

[PMOV 24](#_Toc45309570)

[Specifikacija REST API-ja 25](#_Toc45309571)

[Opis sistema i tehnologija korišćenih u razvoju aplikacije: 30](#_Toc45309572)

[GitHub 33](#_Toc45309573)

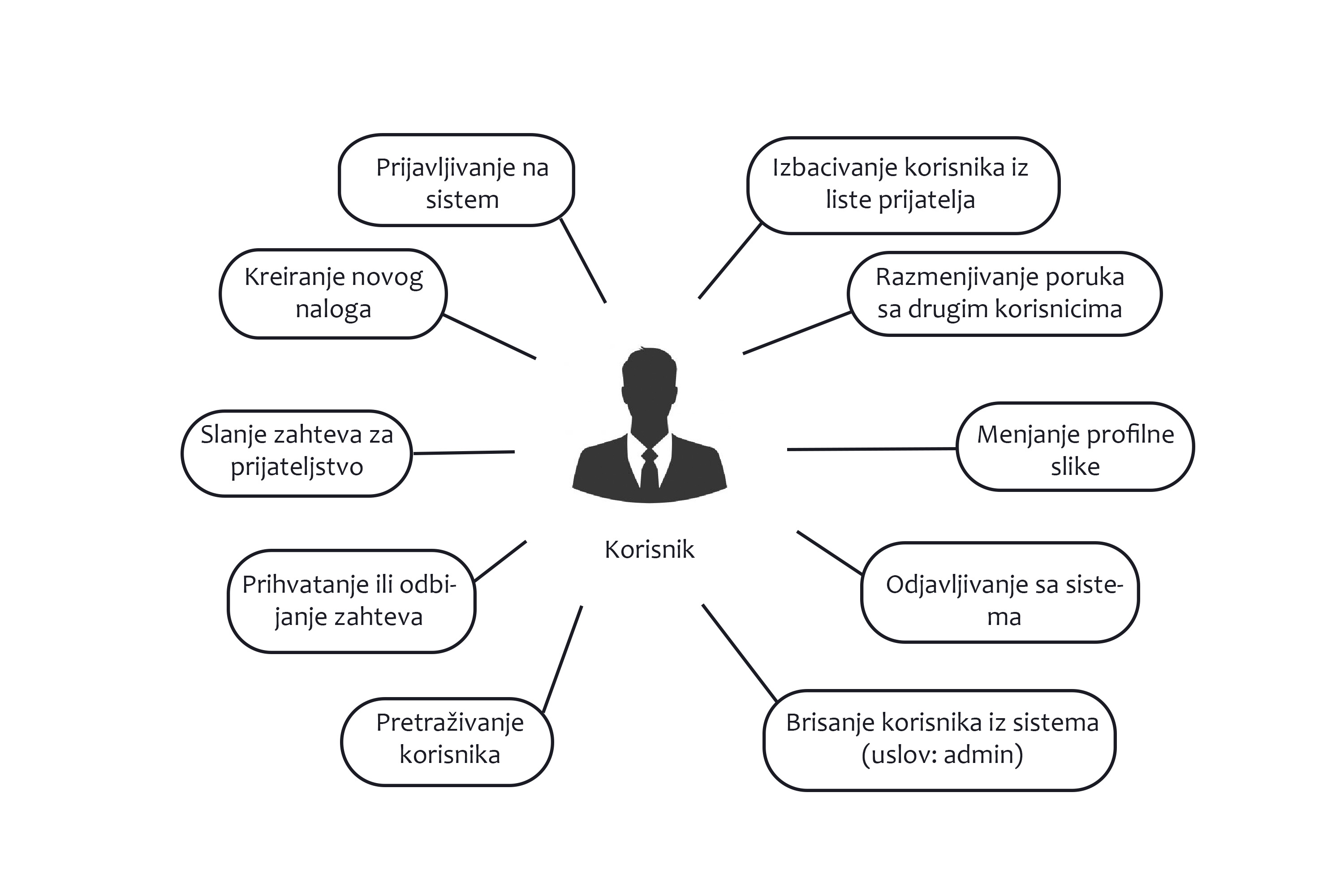
# Verbalni opis sistema

Ova aplikacija predstavlja drustvenu mrežu na kojoj je korisnik u mogućnosti da kreira svoj profil, postavi svoju sliku i konektuje se sa drugim korisnicima. Sistem korisniku treba da obezbedi dodavanje prijatelja i razmenu poruka sa njima. Takođe treba da obezbedi izbacivanje drugih korisnika iz prijatelja. Svaki korisnik može da pregeda nove slike prilikom prijavljivanja ili ponovnim učitavanjem stranice. Korisnicima koji imaju status admina treba omogućiti brisanje profila sa sistema.

# Model slučaja korišćenja

Korisniku je potrebno omogućiti sledeće funkcionalnosti:

1. Prijavljivanje na sistem
2. Kreiranje novog naloga
3. Slanje zahteva za prijateljstvo
4. Prihvatanje ili odbijanje zahteva
5. Pretraživanje korisnika
6. Izbacivanje korisnika iz liste prijatelja
7. Razmenjivanje poruka sa drugim korisnicima
8. Menjanje profilne slike
9. Odjavljivanje sa sistema
10. Brisanje korisnika iz sistema (uslov: admin)



# Opis zahteva pomoću modela slučaja korišćenja

### SK1: Slučaj korišćenja – Prijavljivanje na sistem

**Naziv sk**

Prijavljivanje na sistem

**Autori SK**

Korisnik

**Učesnici SK**

Korisnik i sistem

**Preduslov**

Sistem je uključen i otvorena je forma za logovanje i registrovanje

**Osnovni scenario SK**

1. Korisnik **unosi** korisničko ime i lozinku. (АПУСО)
2. Korisnik **kontroliše** da li je korektno uneo korisničko ime i lozinku. (АНСО)
3. Korisnik **poziva** sistem da se uloguje (proveri podatke). (АПСО)
4. Sistem **proverava** podatke o korisniku. (СО)
5. Sistem **učitava** formu „home page“. (ИА)

**Alternativna scenarija**

5.1. Ukoliko sistem ne može da nađe korisnika prikazuje korisniku poruku: „Wrong Username or Password, try again ”. (ИА)

### SK2: Slučaj korišćenja – Kreiranje novog naloga

**Naziv sk**

Kreiranje novog naloga

**Autori SK**

Korisnik

**Učesnici SK**

Korisnik i sistem

**Preduslov**

Sistem je uključen i otvorena je forma za logovanje i registrovanje

**Osnovni scenario SK**

1. Korisnik **unosi** korisničko ime i lozinku. (АПУСО)
2. Korisnik **kontroliše** da li je korektno uneo korisničko ime i lozinku. (АНСО)
3. Korisnik **poziva** sistem da zapamti podatke o novom korisniku. (АПСО)
4. Sistem **proverava** podatke o korisniku. (СО)
5. Sistem **prikazuje** korisniku poruku: „Registration successful“. (ИА)

**Alternativna scenarija**

5.1. Ukoliko sistem vidi da je lozinka manja od 8 karaktera prikazuje korisniku poruku: „Password must be at least 8 characters long ”. (ИА)

5.2. Ukoliko sistem vidi da lozinka ne sadrži ni jedno slovo prikazuje korisniku poruku: „Password must contain at least one letter”. (ИА1)

5.3. Ukoliko sistem vidi da lozinka ne sadrži ni jednu cifru prikazuje korisniku poruku: „Password must contain at least one num [1-9] ”. (ИА2)

5.4. Ukoliko sistem vidi da lozinka ne sadrži ni jednan poseban karakter [\'^£$%&\*()}{@#~?><>,|=\_+¬-] prikazuje korisniku poruku: „Password must contain at least one symbol: \'^£$%&\*()}{@#~?><>,|=\_+¬- ”. (ИА3)

### SK3: Slučaj korišćenja – Pretraživanje korisnika

**Naziv sk**

Pretraživanje korisnika

**Autori SK**

Korisnik

**Učesnici SK**

Korisnik i sistem

**Preduslov**

Sistem je uključen i korisnik je ulogovan pod svojim šifrom. Učitana je forma „home page“.

**Osnovni scenario SK**

1. Korisnik **unosi** slovo po slovo u input polje. (АПУСО)
2. Sistem **prikazuje** pronađene vrednosti. (СО)
3. Korisnik **poziva** sistem da pošalje korisnika na odabrani profil. (АПУСО)
4. Sistem **učitava** formu za odabrani profil. (ИА)

**Alternativna scenarija**

2.1. Ukoliko sistem ne može da nađe profil sa unešenim početnim slovima prikazuje korisniku poruku: „No user with this username ”. (ИА)

### SK4: Slučaj korišćenja – Slanje zahteva za prijateljstvo

**Naziv sk**

Slanje zahteva za prijateljstvo

**Autori SK**

Korisnik

**Učesnici SK**

Korisnik i sistem

**Preduslov**

Sistem je uključen i korisnik je ulogovan pod svojim šifrom. Učitana je forma odabranog profila („finduser“).

**Osnovni scenario SK**

1. Korisnik **poziva** sistem da pošalje zahtev za prijateljstvo. (АПУСО)
2. Sistem **pamti** da je poslat zahtev. (СО)
3. Sistem **učitava** formu za prikaz statusa zahteva. (ИА)
4. Sistem **prikazuje** korisniku poruku: „poslali ste friend request :)”

**Alternativna scenarija**

4.1. Ukoliko sistem primeti da je već poslat zahtev za prijateljstvo prikazuje korisniku poruku: „Vec ste poslali friend request ! ! ! ”. (ИА)

### SK5: Slučaj korišćenja – Prihvatanje ili odbijanje zahteva

**Naziv sk**

Prihvatanje ili odbijanje zahteva

**Autori SK**

Korisnik

**Učesnici SK**

Korisnik i sistem

**Preduslov**

Sistem je uključen i korisnik je ulogovan pod svojim šifrom. Učitana je forma „home page“.

**Osnovni scenario SK**

1. Sistem **proverava** da li je poslat zahtev korisniku sa nekog profila.(СО)
2. Sistem **učitava** formu za prihvatanje ili odbijanje zahteva. (ИА)
3. Korisniku **poziva** sistem da prihvati drugog korisnika za prijatelja.
4. Sistem **učitava** formu za prikaz statusa prijateljstva. (ИА)
5. Sistem **prikazuje** korisniku poruku: „Vi i {korisnik koji je poslao zahtev} ste sada prijatelji :)”

**Alternativna scenarija**

3.1. Korisniku **poziva** sistem da odbije drugog korisnika za prijatelja.

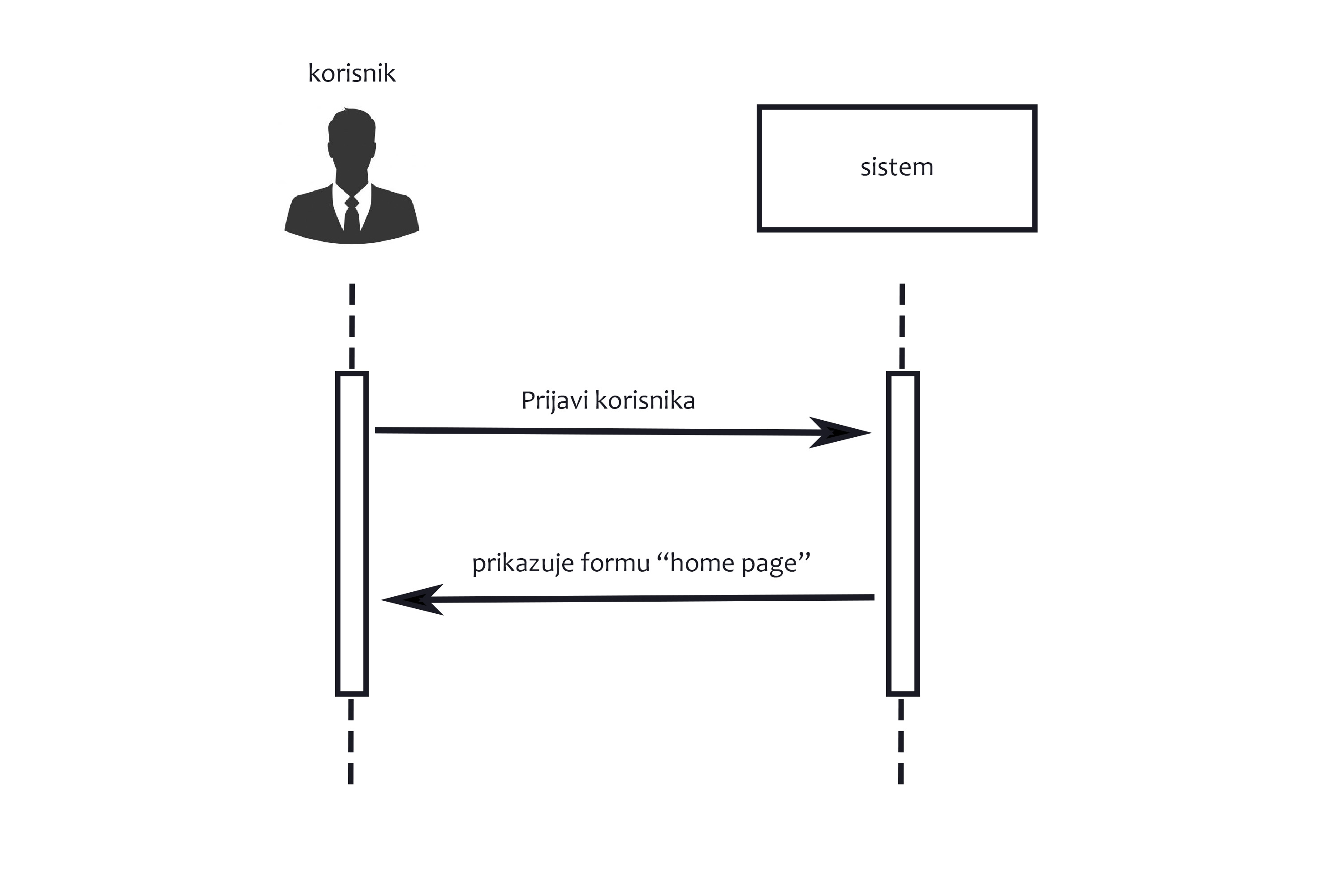
5.1. Sistem **prikazuje** korisniku poruku: „Vi i {korisnik koji je poslao zahtev} niste prijatelji :(”

# Analiza

**Ponašanje softverskog sistema – Dijagram sekvenci**

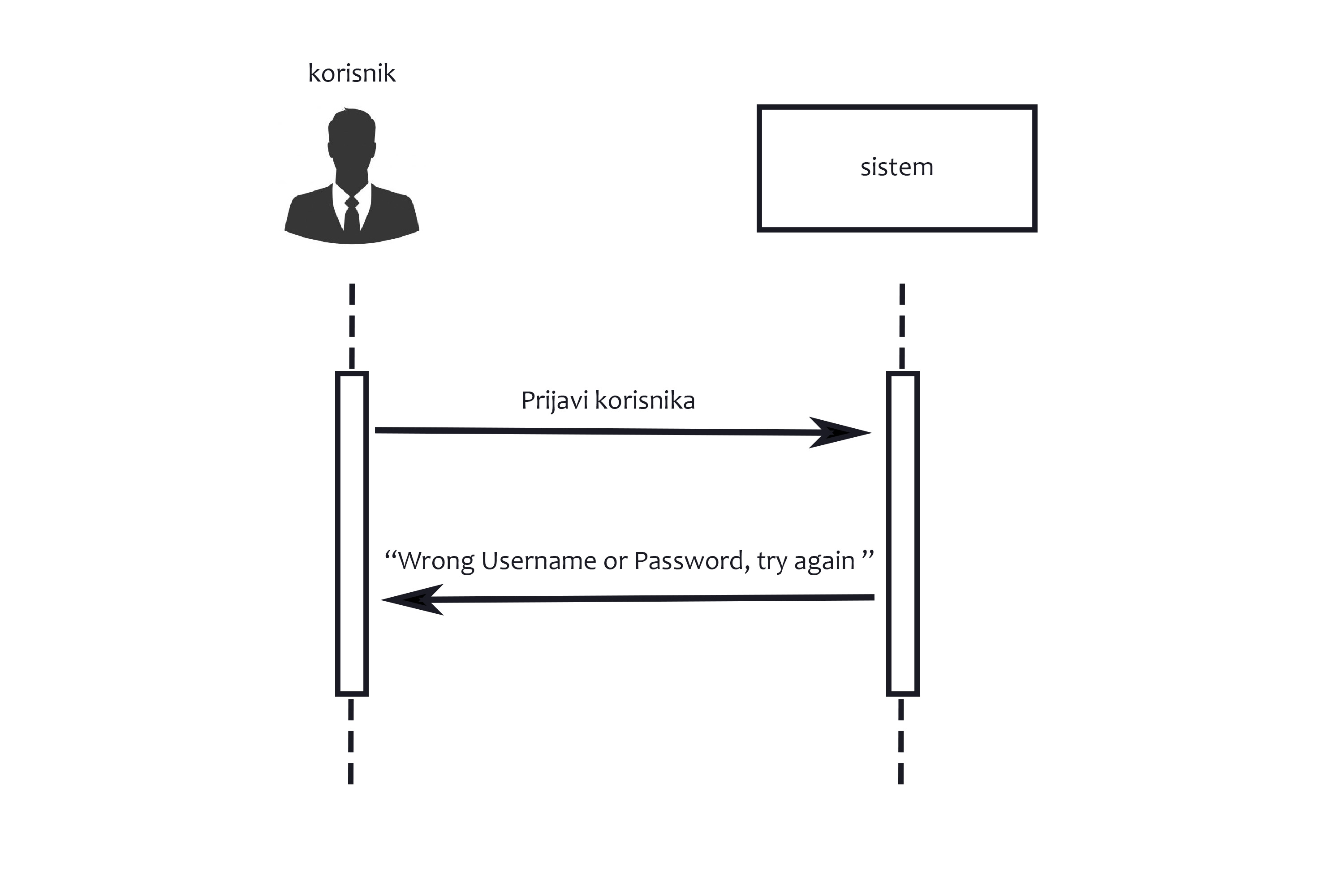
### DS1: Dijagram slučaja korišćenja – Prijavljivanje na sistem

1. Korisnik **poziva** sistem da se uloguje (proveri podatke). (АПСО)
2. Sistem **učitava** formu „home page“. (ИА)



**Alternativna scenarija**

2.1. Ukoliko sistem ne može da nađe korisnika prikazuje korisniku poruku: „Wrong Username or Password, try again ”. (ИА)

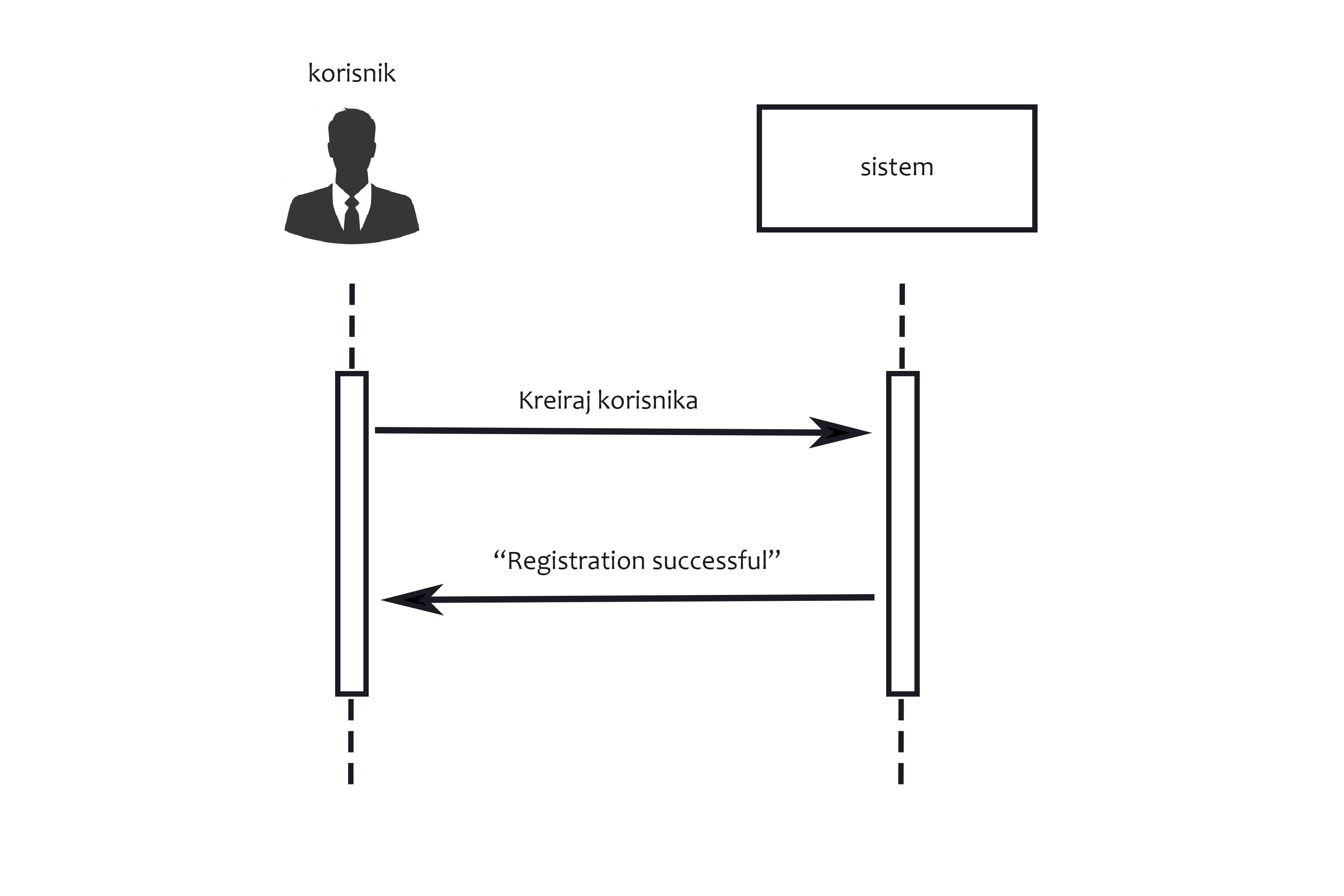


Sa navedemih dijagrama sekvenci uočavaju se sledeće sistemske operacije koje treba projektovati:

1. signal Prijavi korisnika

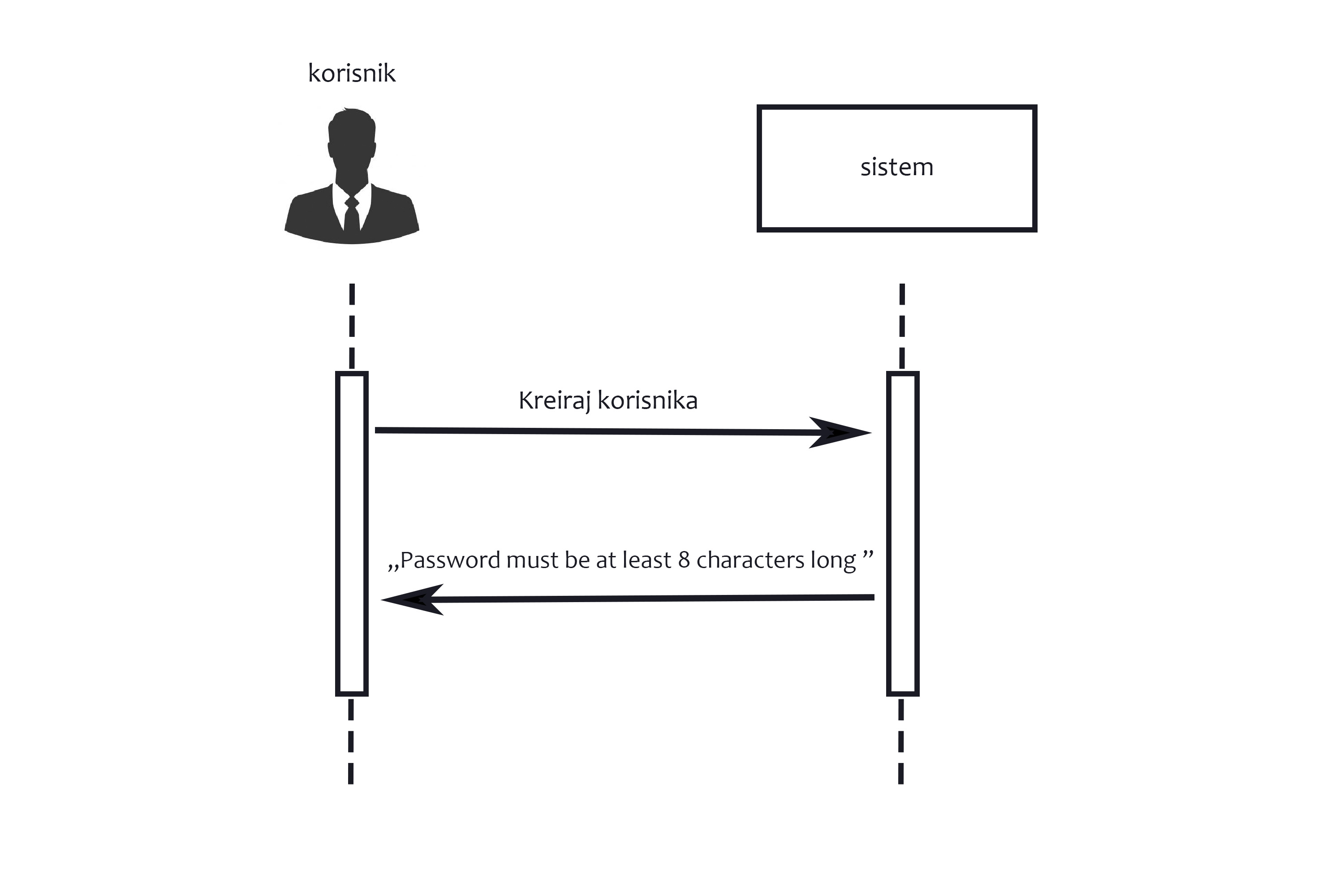
### DS2: Dijagram slučaja korišćenja – Kreiranje novog naloga

1. Korisnik **poziva** sistem da zapamti podatke o novom korisniku. (АПСО)
2. Sistem **prikazuje** korisniku poruku: „Registration successful“. (ИА)

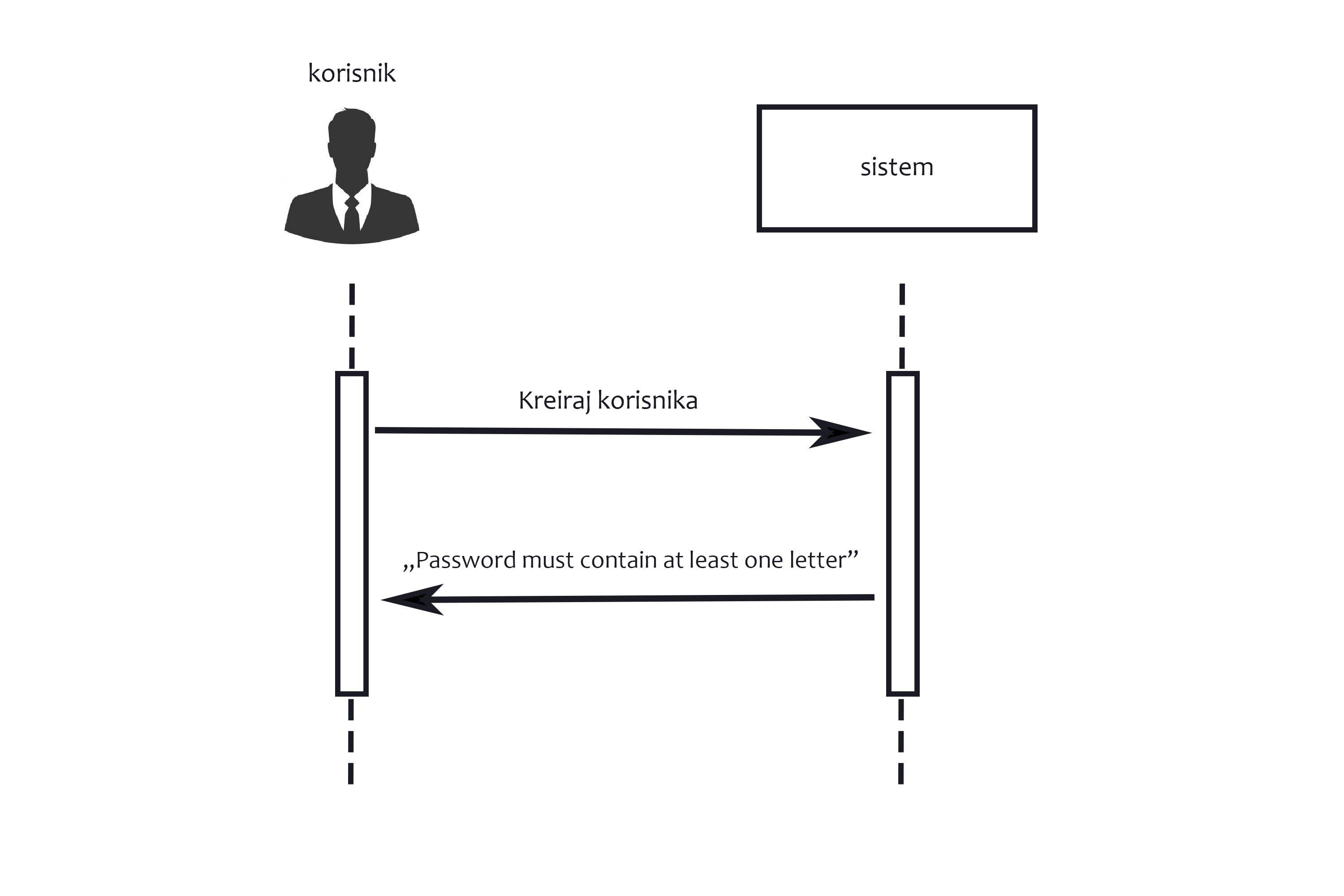


**Alternativna scenarija**

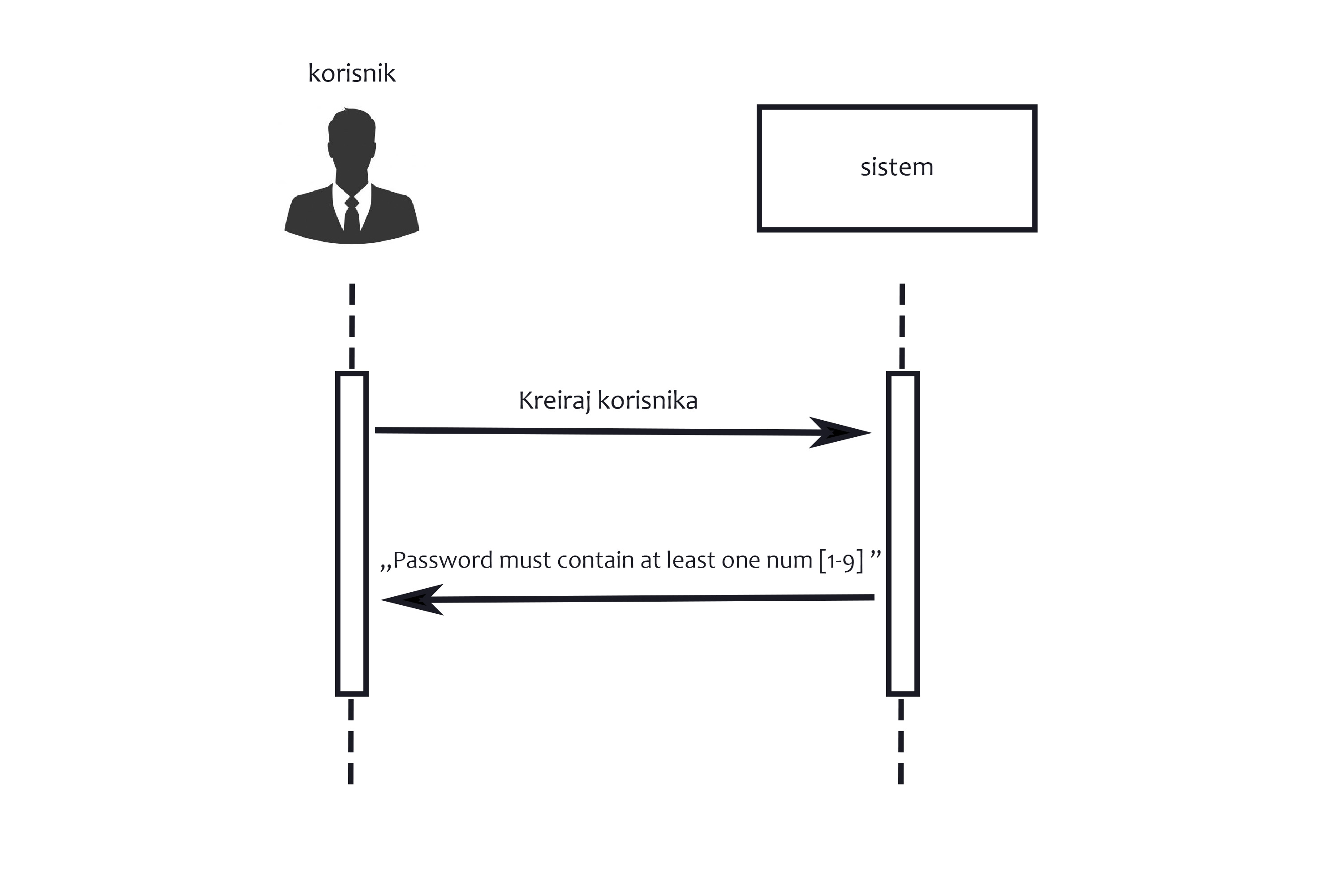
5.1. Ukoliko sistem vidi da je lozinka manja od 8 karaktera prikazuje korisniku poruku: „Password must be at least 8 characters long ”. (ИА)



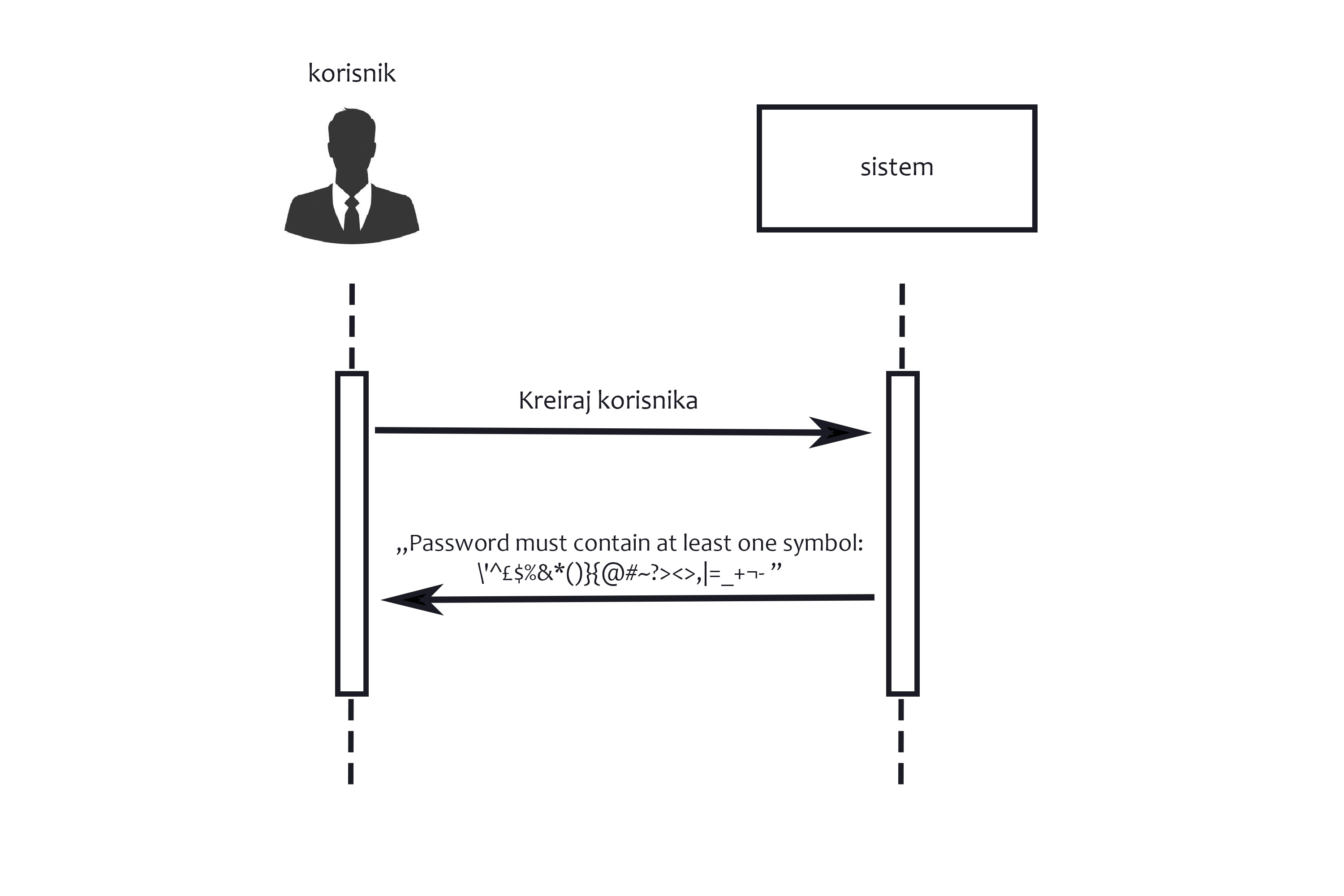
* 1. Ukoliko sistem vidi da lozinka ne sadrži ni jedno slovo prikazuje korisniku poruku: „Password must contain at least one letter”. (ИА1)



5.3. Ukoliko sistem vidi da lozinka ne sadrži ni jednu cifru prikazuje korisniku poruku: „Password must contain at least one num [1-9] ”. (ИА2)



5.4. Ukoliko sistem vidi da lozinka ne sadrži ni jednan poseban karakter [\'^£$%&\*()}{@#~?><>,|=\_+¬-] prikazuje korisniku poruku: „Password must contain at least one symbol: \'^£$%&\*()}{@#~?><>,|=\_+¬- ”. (ИА3)

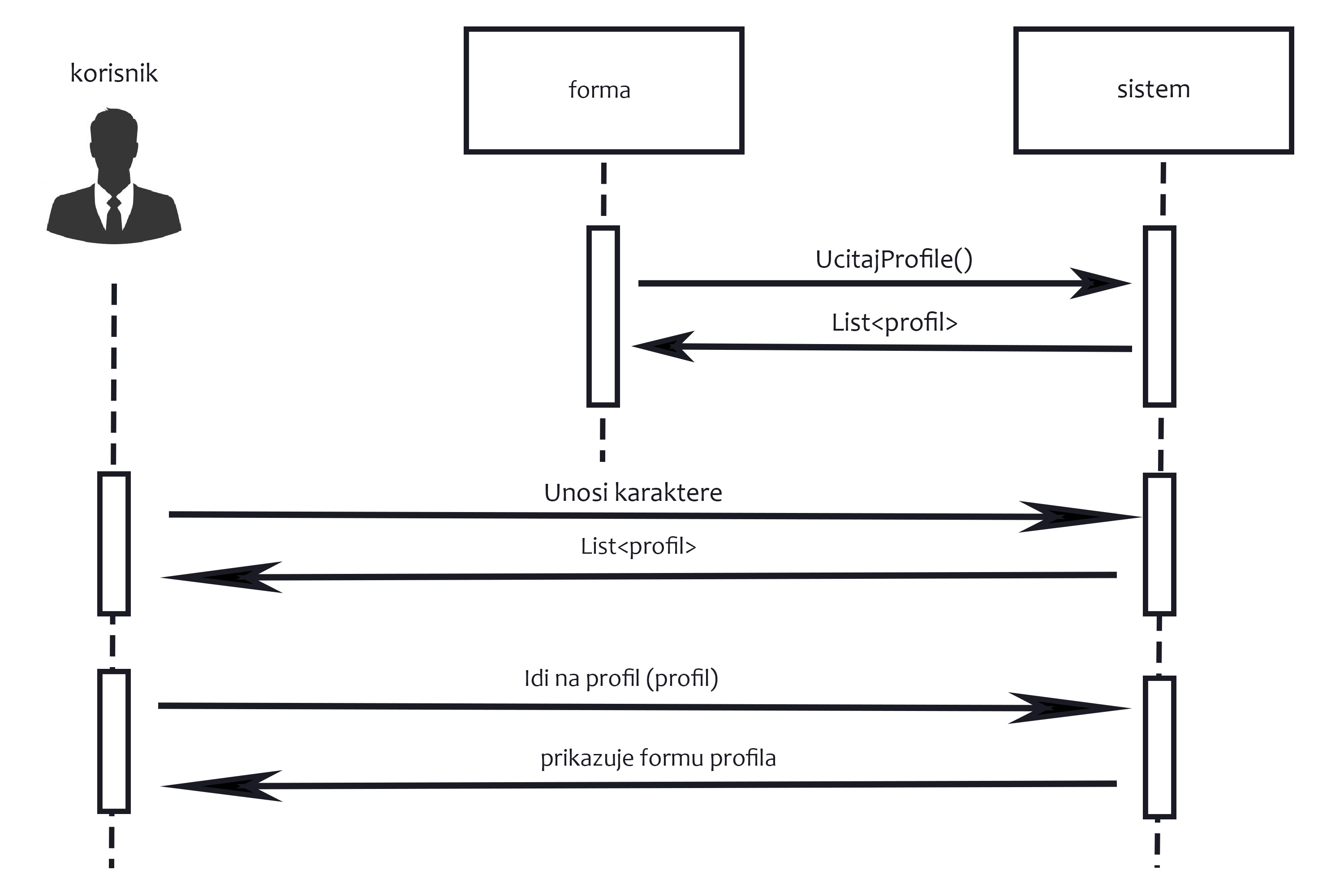


Sa navedemih dijagrama sekvenci uočavaju se sledeće sistemske operacije koje treba projektovati:

1. signal Kreiraj korisnika

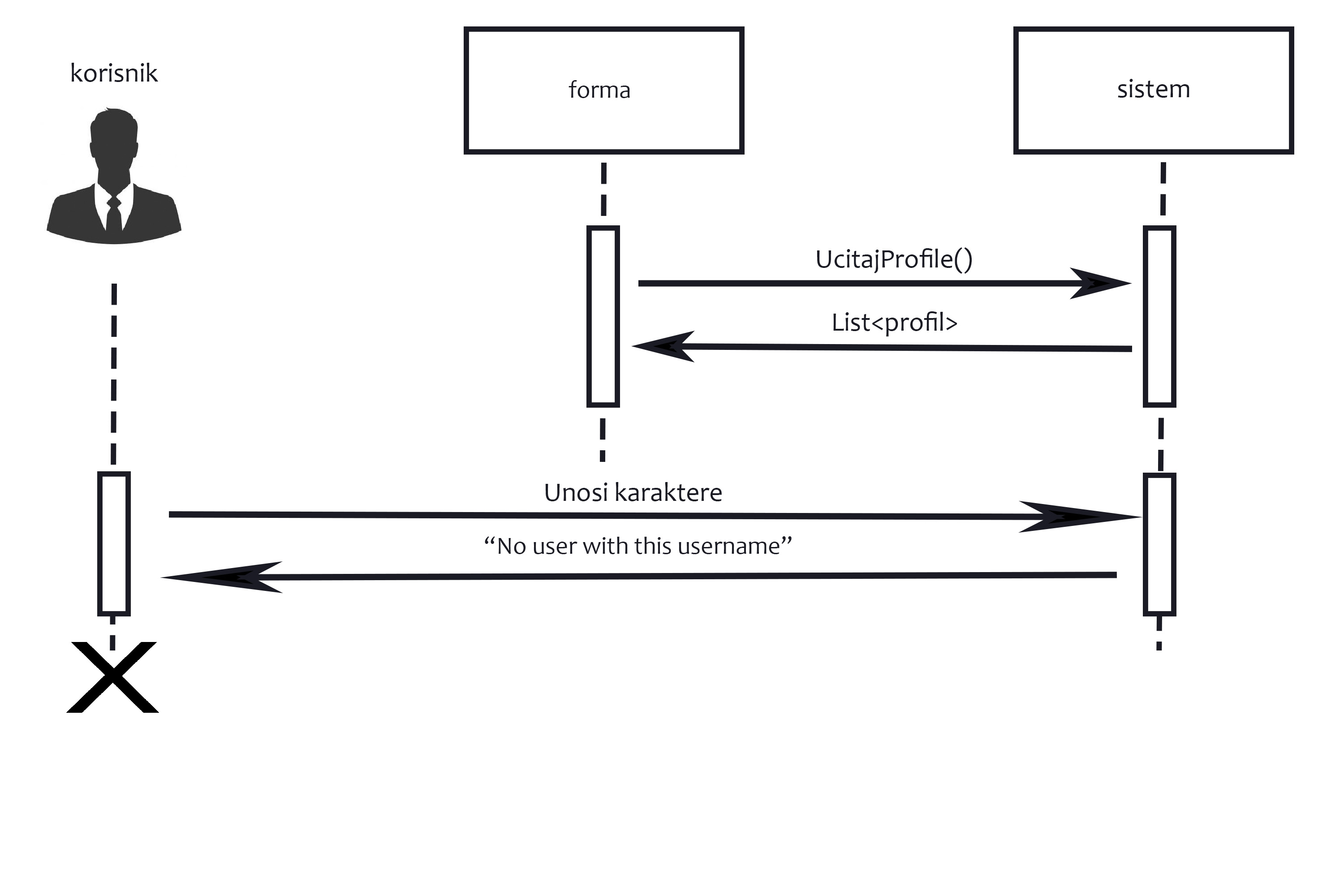
### DS3: Dijagram slučaja korišćenja – Pretraživanje korisnika

1. Korisnik **unosi** slovo po slovo u input polje. (АПУСО)
2. Forma poziva sistem da prikaze listu profila.
3. Sistem prikazuje formi listu profila.
4. Sistem **prikazuje** pronađene vrednosti. (СО)
5. Korisnik **poziva** sistem da pošalje korisnika na odabrani profil. (АПУСО)
6. Sistem **učitava** formu za odabrani profil. (ИА)



**Alternativna scenarija**

4.1. Ukoliko sistem ne može da nađe profil sa unešenim početnim slovima prikazuje korisniku poruku: „No user with this username ”. Prekida se izvršenje scenarija. (ИА)

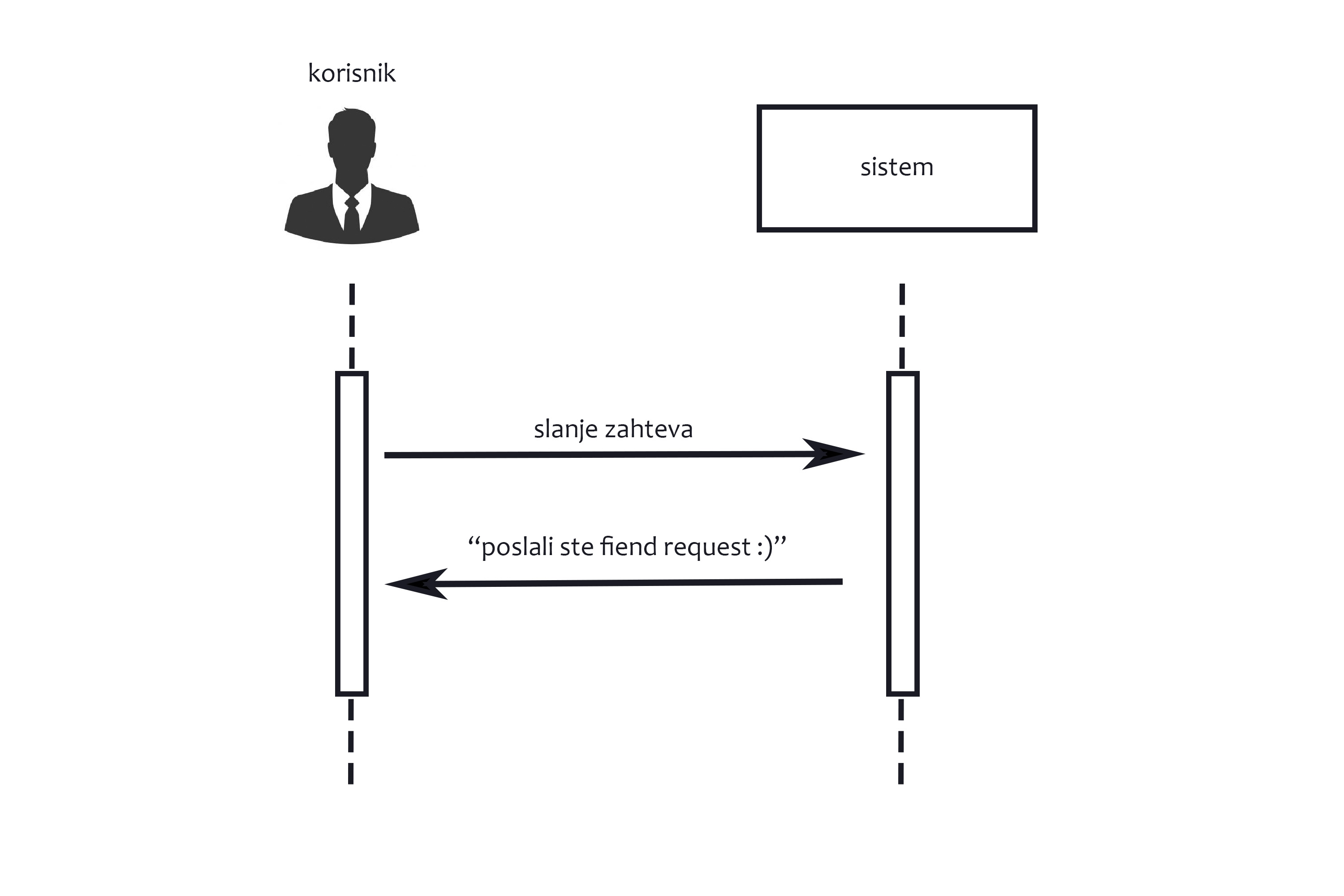


Sa navedemih dijagrama sekvenci uočavaju se sledeće sistemske operacije koje treba projektovati:

1. signal Unosi karaktere
2. signal UcitajProfile()
3. signal idi na profil (profil)

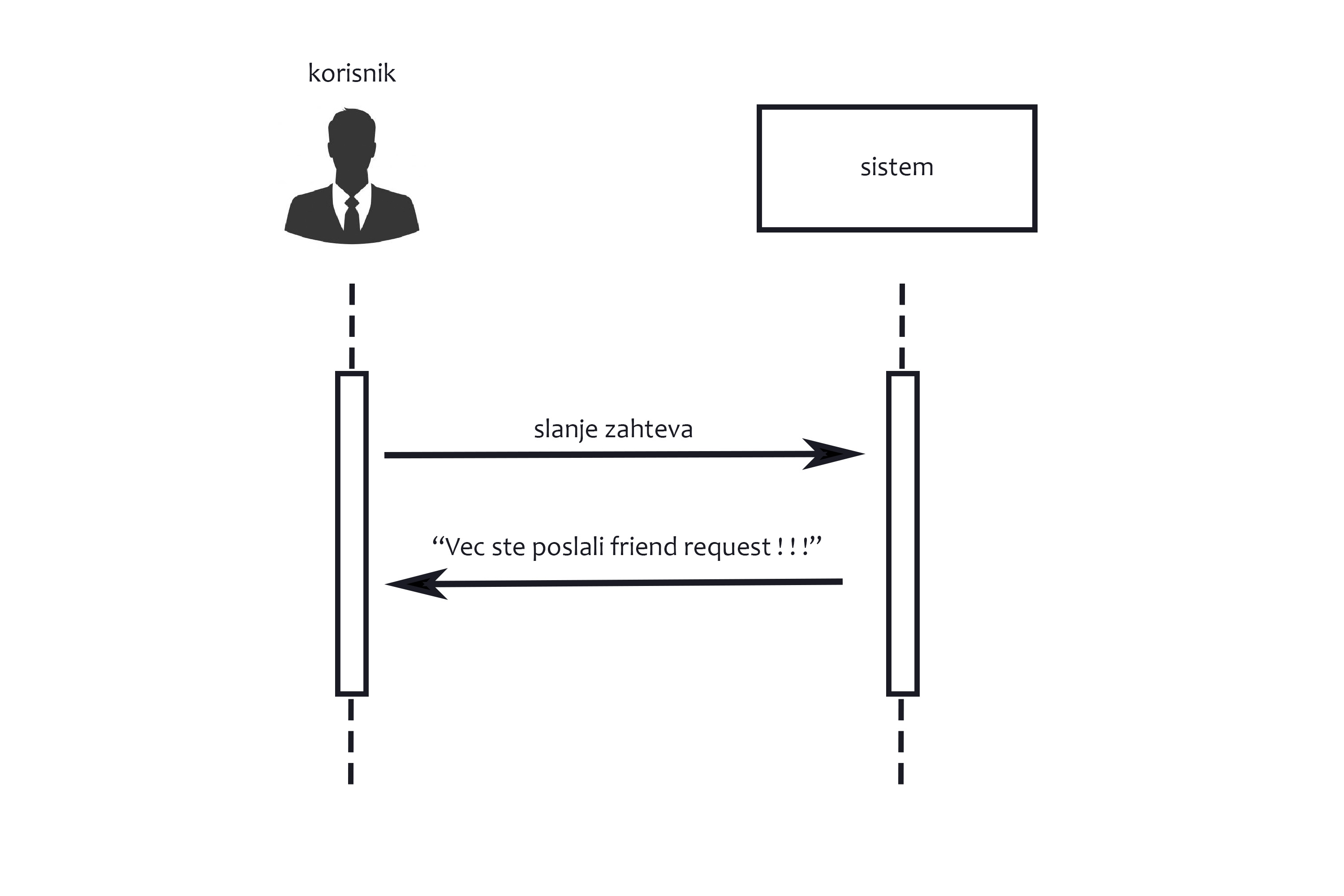
### DS4: Dijagram slučaja korišćenja – Slanje zahteva za prijateljstvo

1. Korisnik **poziva** sistem da pošalje zahtev za prijateljstvo. (АПУСО)
2. Sistem **prikazuje** korisniku poruku: „poslali ste friend request :)”



**Alternativna scenarija**

2.1. Ukoliko sistem primeti da je već poslat zahtev za prijateljstvo prikazuje korisniku poruku: „Vec ste poslali friend request ! ! ! ”. (ИА)

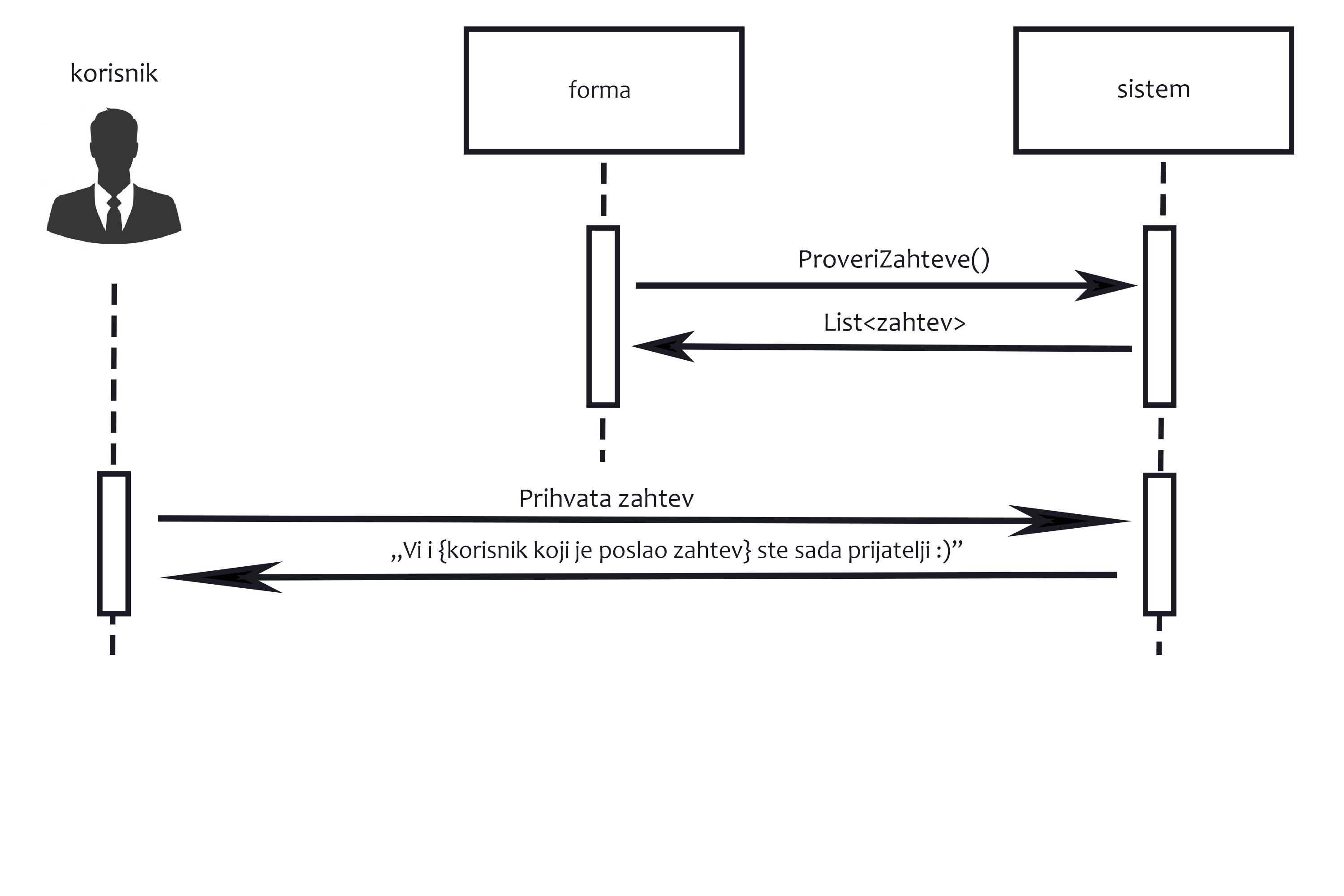


Sa navedemih dijagrama sekvenci uočavaju se sledeće sistemske operacije koje treba projektovati:

1. signal slanje zahteva

### DS5: Dijagram slučaja korišćenja – Prihvatanje ili odbijanje zahteva

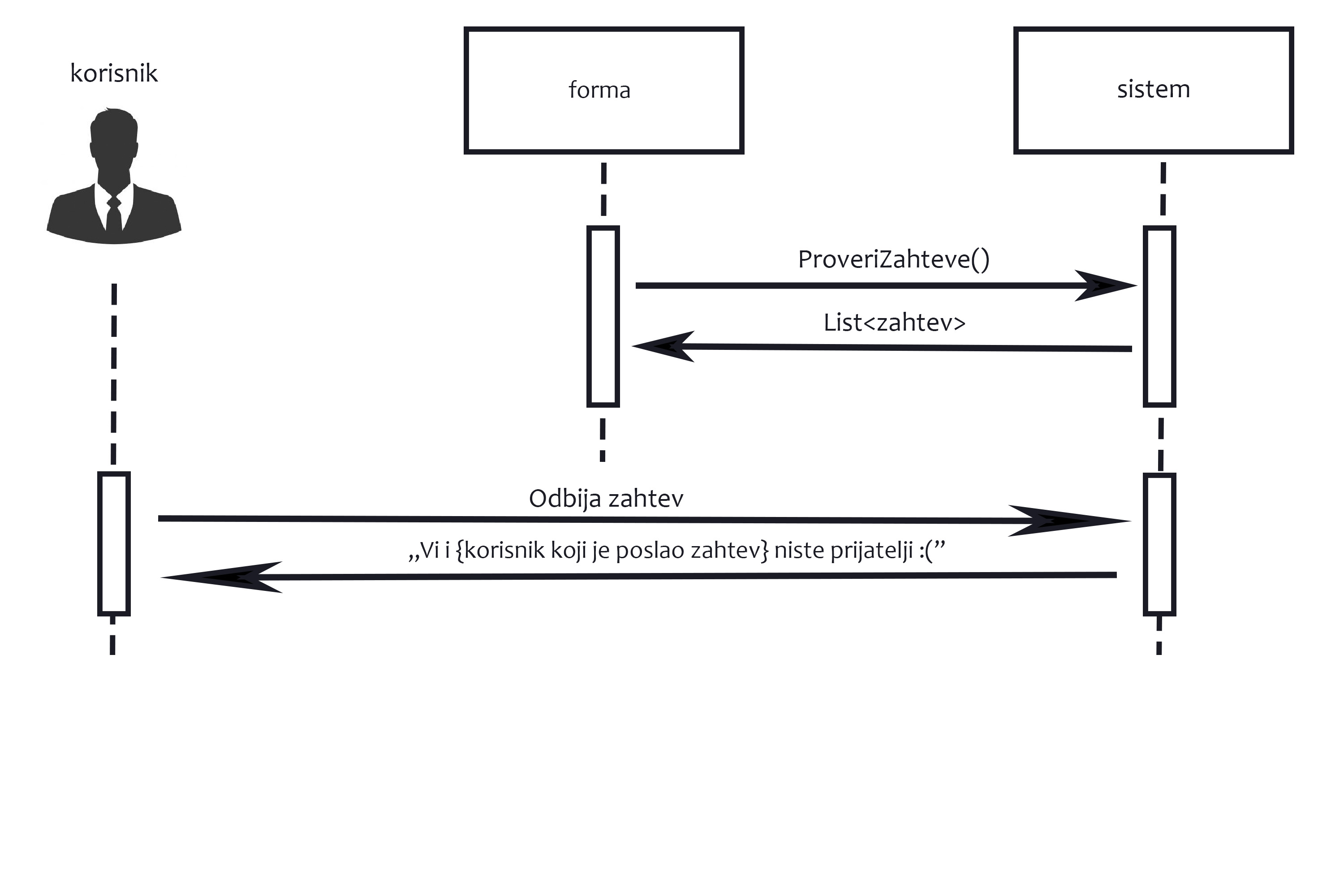
1. Forma poziva sistem da proveri zahteve
2. Sistem **prikazuje** listu zahteva. (ИА)
3. Korisniku **poziva** sistem da prihvati drugog korisnika za prijatelja.
4. Sistem **prikazuje** korisniku poruku: „Vi i {korisnik koji je poslao zahtev} ste sada prijatelji :)”



**Alternativna scenarija**

3.1. Korisniku **poziva** sistem da odbije drugog korisnika za prijatelja.

5.1. Sistem **prikazuje** korisniku poruku: „Vi i {korisnik koji je poslao zahtev} niste prijatelji :(”



Sa navedemih dijagrama sekvenci uočavaju se sledeće sistemske operacije koje treba projektovati:

1. signal ProveriZahteve()
2. signal Prihvata zahtev
3. signal Odbija zahtev

# Struktura baze

Baza: „userregistration“

Tabela 1: „usertable“

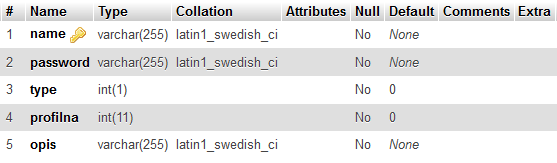


Tabela 2: „friends“

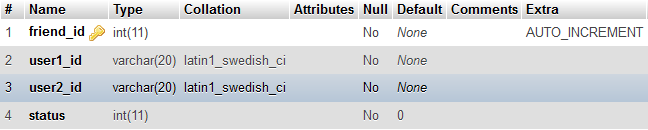
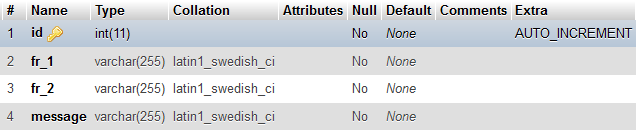
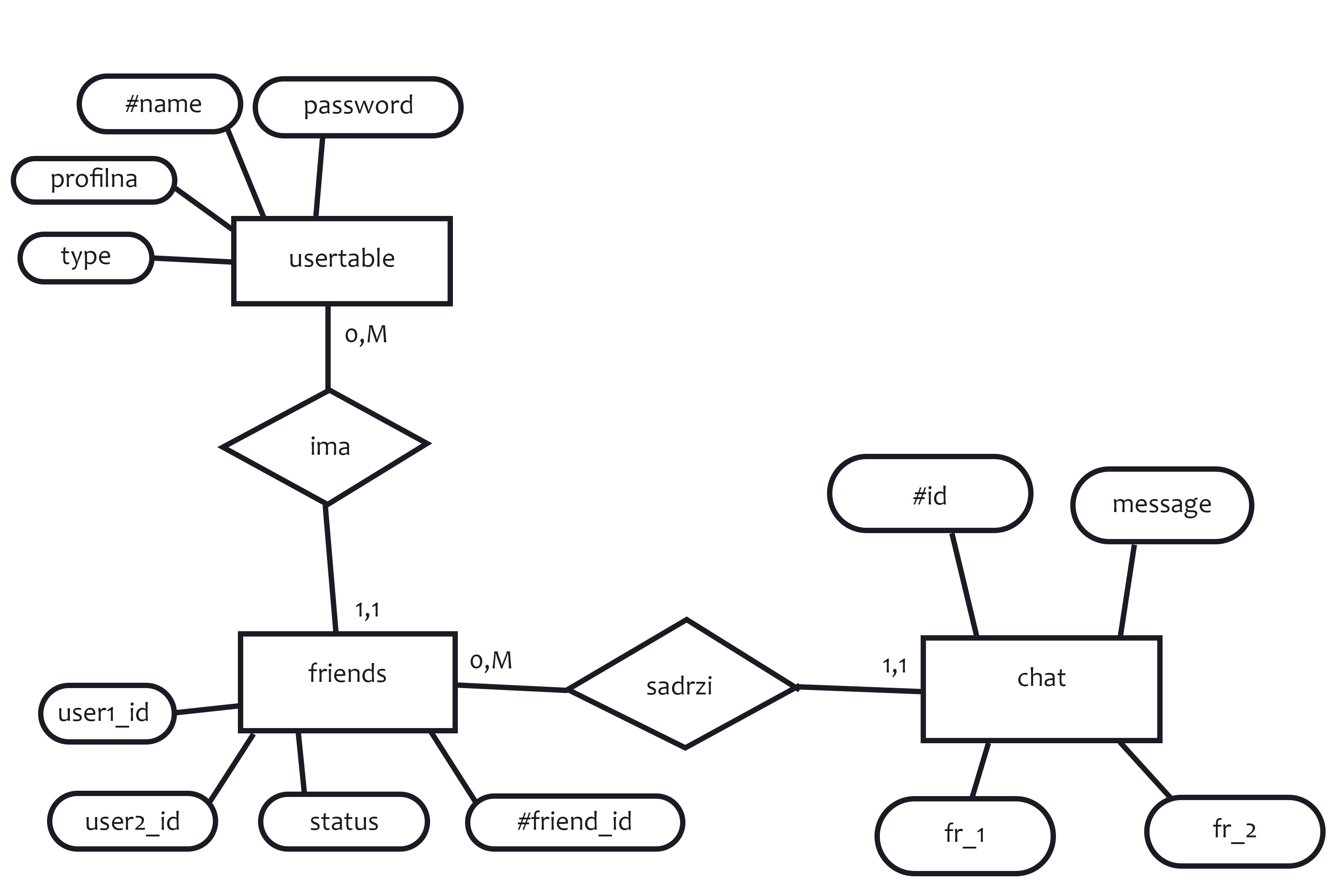


Tabela 3: „chat“



# PMOV



# Specifikacija REST API-ja

|  |  |
| --- | --- |
| Opis funkcije | Pregled profila |
| HTTP metoda | GET |
| URL | /usertable.json |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Niz JSON objekata. Svaki element niza ima atribut name[string]  Primer:  [  {'name':'danko'},  {'name':'danko1'},  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Pregled prijatelja i zahteva |
| HTTP metoda | GET |
| URL | /friends/:id.json |
| URL parametri | id[int] – status izmedju 2 profila |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Niz JSON objekata. Svaki element niza ima atribut user1\_id [string] i user2\_id [string]  Primer:  [  {'user1\_id':'danko','user2\_id':'danko1'},  {'user1\_id':'danko','user2\_id':'danko2'},  {'user1\_id':'danko4','user2\_id':'danko'}  ] |
| Format izlaznih parametara | application/json |

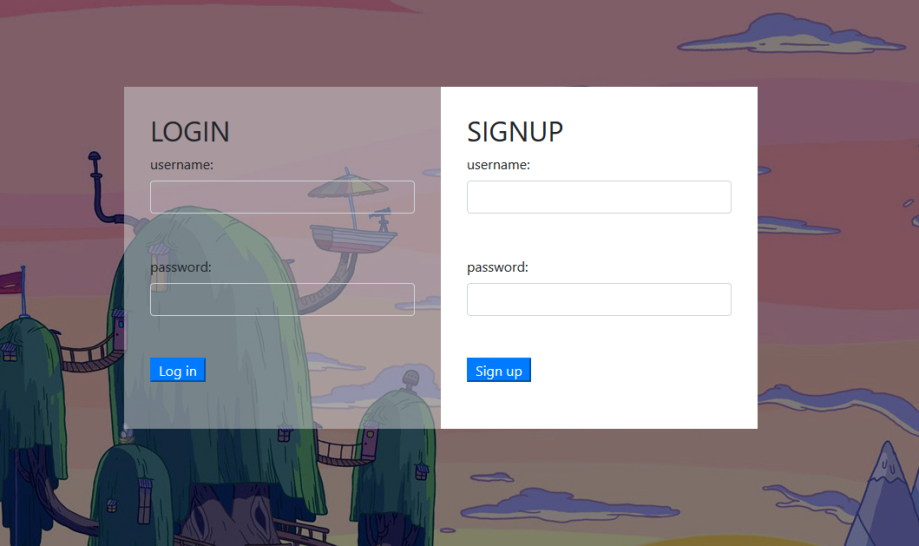
|  |  |
| --- | --- |
| Opis funkcije | Dodavanje novog profila |
| HTTP metoda | POST |
| URL | /usertable |
| URL parametri | (nema) |
| HTTP body parametri | JSON objekat sa atributima name[string], password[string]  Primer:  {  “name”:”aleksandra”,  “password”:”a#452dFg”  } |
| Format HTTP body parametara | application/json |
| Izlazni parametri | Kratak JSON objekat sa jednim atributom, poruka[string].  Primer:  {  "poruka":"Korisnik je uspešno ubačen"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Menjanje poruke iz chat-a |
| HTTP metoda | PUT |
| URL | /chat/:id |
| URL parametri | id[string] – koju poruku želimo da izmenimo |
| HTTP body parametri | JSON objekat sa atributiom novaporuka[string]  Primer:  {  “novaporuka”:”hello”,  } |
| Format HTTP body parametara | application/json |
| Izlazni parametri | Kratak JSON objekat sa jednim atributom, poruka[string].  Primer:  {  „poruka“:“Poruka je uspešno izmenjena“  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje profila |
| HTTP metoda | DELETE |
| URL | /usertable/:id |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | Kratak JSON objekat sa jednim atributom, poruka[string].  Primer:  {  „poruka“:“Korisnik je uspešno izbrisan“  } |
| Format izlaznih parametara | application/json |

# **Opis sistema i tehnologija korišćenih u razvoju aplikacije**:

1. Na prvoj stranici login.php korišćen je bootsrap za kreiranje login-a i registracije. Tu se nalaze dva division-a jedan za prijavljivanje i jedan za registraciju i svaki u sebi sadrži formu koja klikom na dugme (“log in” ili “sign up”) koristi POST metodu i šalje vrednosti iz input polja (“username” i “password”).



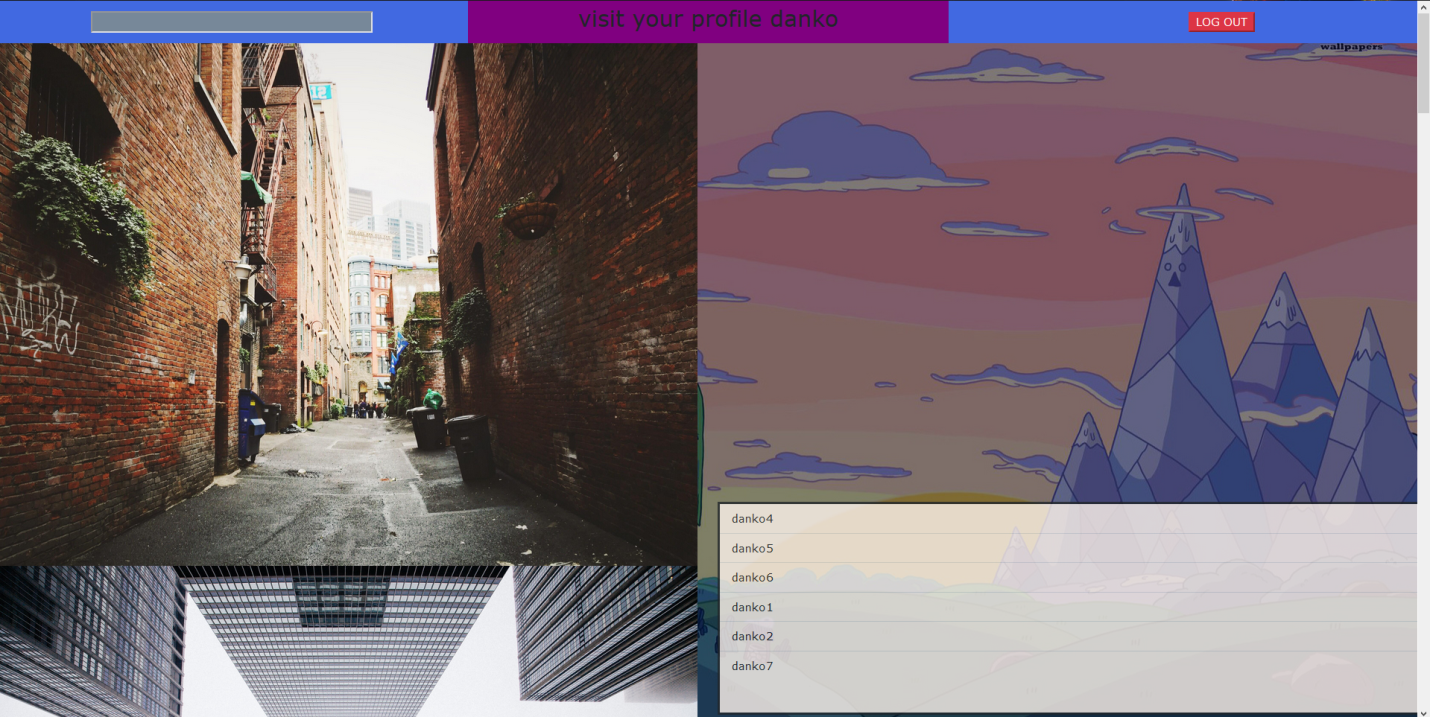
registration.php uzima POST promenljive (“username” i “password”), ostvaruje konekciju sa bazom userregistration i proverava da li u tabeli usertable postoji korisnik sa datim imenom.

Ako ne postoji, u tabelu usertable se dodaje novi red sa datim vrednostima za username i password i kreira novi direktorijum koji ima isti naziv kao i username.

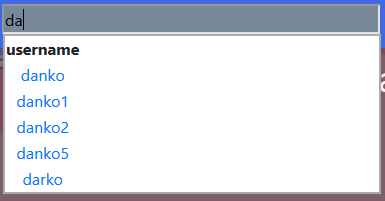
validation.php uzima POST promenljive (“username” i “password”), ostvaruje konekciju sa bazom userregistration i proverava da li u tabeli usertable postoji korisnik sa datim imenom i šifrom. Takođe započinje sesiju i pamti SESSION varijable za ime profila, broj profilne slike i tip profila (0 – običan korisnik, 1 – admin).

1. Log in-ovanjem dolazimo do home page-a.

Na vrhu stranice se nalaze 3 division-a.



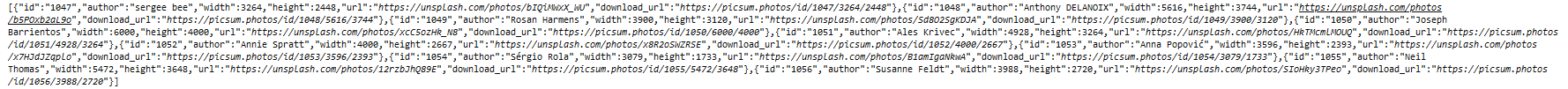
U prvom se nalazi input polje koje koristi AJAX tehnologiju da izlista korisnike:



ajax.js koristi GET metodu da dobije listu imena koja počinju sa slovima koja su upisana u input polju. U people.php vidimo da se ispisuju imena kao href i da pritiskom na neko ime prelazimo na nečiji profil.

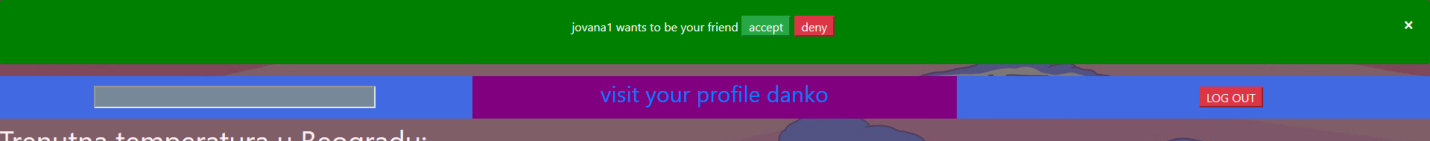
U drugom division-u se nalazi href koji nas šalje na profil korisnika koji je ulogovan, a u trećem log out dugme koje prekida session i otvara log in formu ponovo.

Ispod ovoga vidimo izlistane slike koje smo dobili korišćenjem spoljnjeg API-ja koji nam vraca JSON koji izgleda ovako:



Ovaj API nam daje nasumično slike, u ovom zahtevu smo u URL-u prosledili broj slika koje želimo “limit” i dodali da nam vrati slike sa “random” stranice.

Takođe, na home page-u dobijamo obaveštenja kada je korisniku poslat friend request ili kada je odbijen friend request koji je on poslao. Ako je dobio friend request ima mogućnost ta prihvati ili odbije.

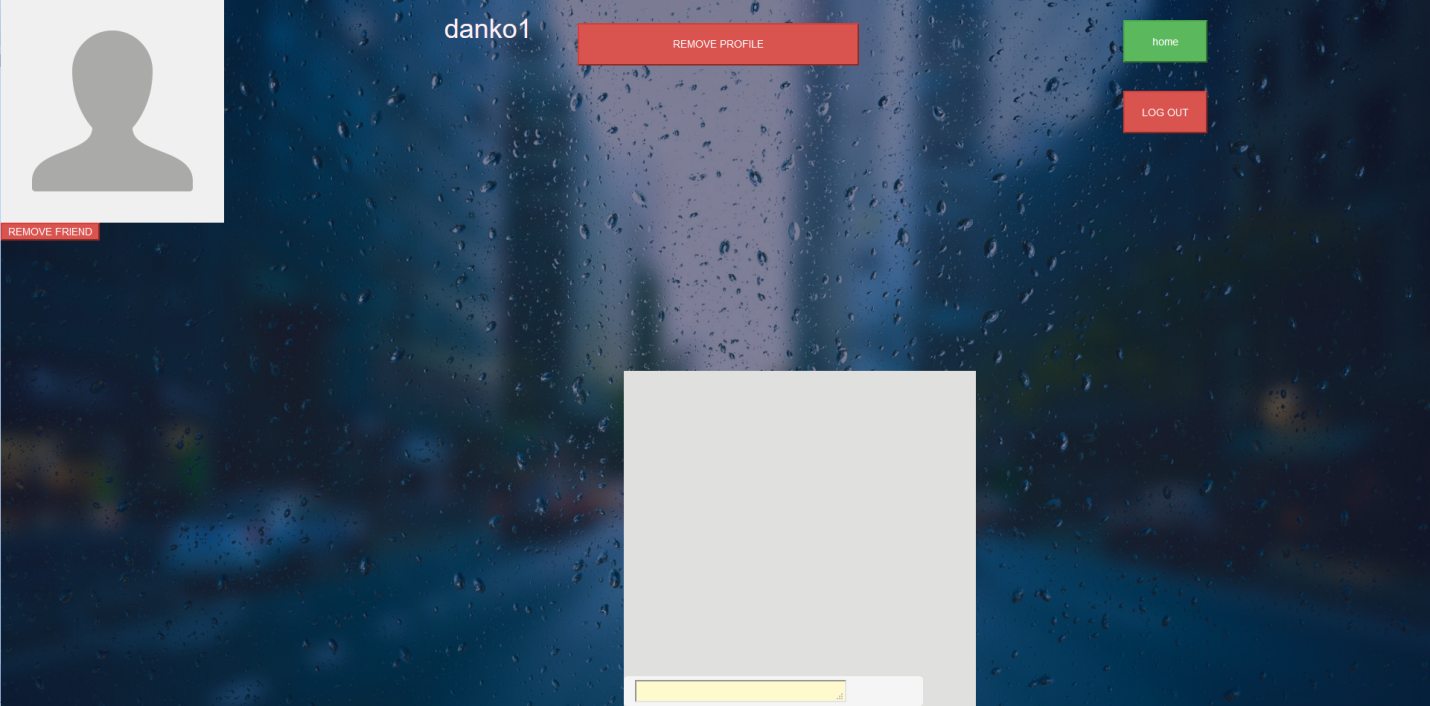


Prihvatanjem se poziva accept.php koji pravi konekciju sa bazom userregistration i u tabeli friends menja vrednost „status“ kolone iz 0 u 1 i ispisuje slededcu poruku:



U suprotnom se poziva deny.php koji menja vrednost „status“ kolone iz 0 u 2 i slično ispisuje poruku samo o tome da je odbijen zahtev.

1. Kada korisnik pristupi profilu koji nije njegov izgleda ovako:



U ovom slučaju korisnik je admin tako da je prisutno dugme „remove profile“, pritiskom na to dugme iz tabele usertable se briše red koji sadrži id ovog profila.

Ispod profilne slike se nalazi dugme „add friend“ i ono poziva addFriend.php.

Prvo se proverava da li u tabeli „friends“ baze „userregistration“ postoji kombinacija gde su oba profila u istom redu i status je 0 (poslat je friend request), ako postoji ispisuje se odgovarajuća poruka. I ako je profil na kome se korisnik nalazi prvi poslao friend request u tabeli se menja status na 1 (prijatelji su).

Nakon toga se proverava da li postoji kombinacija gde su oba profila u istom redu i status je 1 (prijatelji su), ako postoji izbacuje se poruka da su već prijatelji.

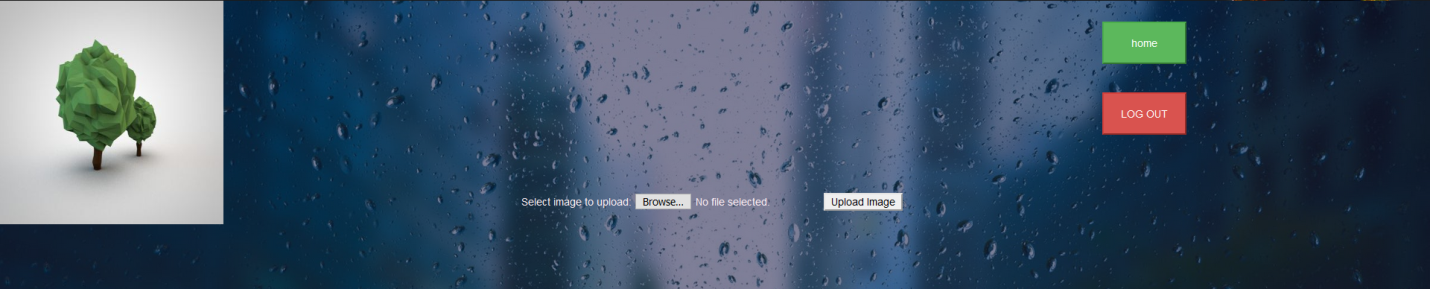
A ako nema kombinacije ova 2 profila u jednom redu, dodaje se novi red u tabelu sa statusom 0 (poslat friend request).

Druga situacija (kao na slici) je da su ova dva profila već prijatelji i onda imamo dugme „remove friend“ koje poziva removeFriend.php.

Na dnu stranice vidimo chat koj koristi jQuery POST metodu da pošalje sasdržaj textarea tag-a sa ID-em „poruka“ u ajax1.php i u zavisnosti od vrednosti za ključ „action“ se ili dodaje novi red u tabeli „chat“ (baze „userregistration“) ili se ispisuje istorija dopisivanja između ova 2 profila.

Ako se ulogujemo na 2 različita profila i pošaljemo poruku vidimo da se poruka ispisuje sa odstupanjem od 1 sekunde.

1. Kada se korisnik nalazi na svom profilu ima mogućnost da promeni sliku.



Kada se slika odabere, klikom na „upload image“ se poziva change\_profile\_pic.php koji premešta sliku u direktorijum, koji je kreiran kada smo registrovali profil, i menja joj ime tako da ima naziv profila sa brojem (koji označava redni broj profilne slike). Takodje u tabeli „usertable“, kolona „profile“ se menja tako što se dodaje 1 na trenutni broj. Ovim je obezbeđeno da je uvek poslednja upload-ovana slika prikazana na svakom profilu.

# GitHub

GitHub link : <https://github.com/danko112233/ITEH>