

Univerzitet u Beogradu  
Fakultet organizacionih nauka  
Laboratorija za elektronsko poslovanje



## Web aplikacija za rezervaciju letova

Seminarski rad iz Internet Tehnologija

Profesor:  
Zorica Bogdanović

Studenti:  
Marija Bjelopetrović 396/16  
Nina Vuković 352/16

Beograd, 2020.god

## Sadržaj

1. Korisnički zahtev .....	4
1. Opis sistema .....	7
1.1 Slučajevi korišćenja .....	7
2.1.1. SK Prijava korisnika .....	7
2.1.2. SK Unos leta .....	8
2.1.3 SK Pretraživanje rezervacija .....	9
2.1.4 SK Izmena leta .....	10
2.1.5 SK Unos nove slike u galeriju .....	11
2.1.6 SK Unos nove rezervacije .....	12
2.1.7 SK Brisanje rezervacija .....	13
2.1.8 Prikaz grafika .....	14
2.1.9 SK Prikaz letova .....	15
2.1.10 SK Registrovanje korisnika .....	16
1.2 Opis strukture sistema .....	17
1.3 Opis procesa – dijagram sekvenci .....	18
2.3.1 DS Prijava korisnika .....	18
2.3.2 DS Unos leta .....	20
2.3.3 DS Pretraživanje rezervacija .....	22
2.3.4 DS Izmena leta .....	23
2.3.5 DS Unos nove slike u galeriju .....	25
2.3.6 DS Unos nove rezervacije .....	25
2.3.7 DS Brisanje rezervacije .....	26
2.3.8 DS Prikaz grafika .....	28
2.3.9 DS Prikaz letova .....	29
2.3.10 DS Registrovanje korisnika .....	29
1.4 Model podataka .....	31
2.5. Specifikacija REST APIja .....	32
2.5.1 Funkcija sviLetovi() .....	32
2.5.2 Funkcija sveKlase() .....	33
2.5.3 Funkcija ubaciKorisnika(data) .....	34
2.5.4 Funkcija unosLeta(data) .....	35
3. Korišćene tehnologije .....	36

4.Uputstvo za korisnike.....	38
5. Delovi koda seminarskog rada “HappyAirline” .....	46
5.1 Prikaz svih letova.....	46
Prikaz vremenske prognoze .....	52
Unos novog leta .....	55
Unos nove slike u galeriju .....	55
Prikaz grafika.....	57
Registracija .....	59
Login.....	62
Rezervacije .....	63
Prikaz svih rezervacija i brisanje rezervacije .....	67
Sopstveni servisi.....	69

## 1.Korisnički zahtev

Potrebno je napraviti web aplikaciju za rezervaciju avio karata. Sistem treba da podržava rad sa dve vrste korisnika. Prvu vrstu čine obični korisnici koji mogu da se registruju i uloguju na sistem, pregledaju letove iz ponude i rezervišu letove. Korisnicima je potrebno omogućiti logovanje, a ukoliko korisnici ne postoje u bazi, omogućiti im registrovanje i upisivanje u bazu podataka.

Druga vrsta su korisnici koji imaju ulogu administratora i oni imaju mogućnost da unose nove letove (odnosno destinacije putovanja sa dodatnim atributima o samom letu).

Pored navednog slučaja korišćenja, administratori imaju mogućnost i izmene podataka o prethodno unetom letu, omogućeno im je i brisanje rezervacija iz tabele rezervacije u bazi, kao i prikaz grafika, na kojem se prikazuju, procentualno, rezervacije po klasi leta.

Administratori, kao i obični korisnici imaju mogućnost rezervacija i pregleda svih letova. Administratori mogu i da ubacuju nove slike u galeriju slika. Na početnoj strani je potrebno omogućiti prikaz svih dostupnih letova.

Pristup imaju **dve vrste korisnika**:

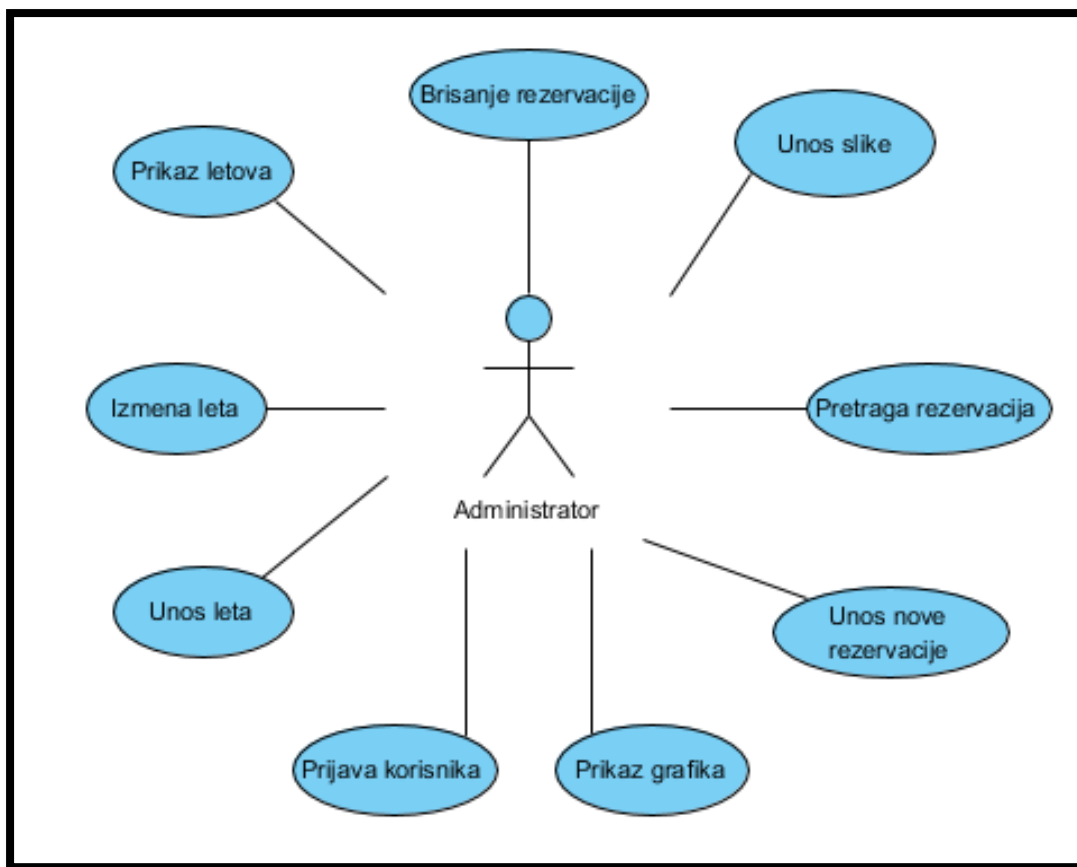
- administrator i
- korisnik

**Administratoru** omogućiti sledeće funkcionalnosti:

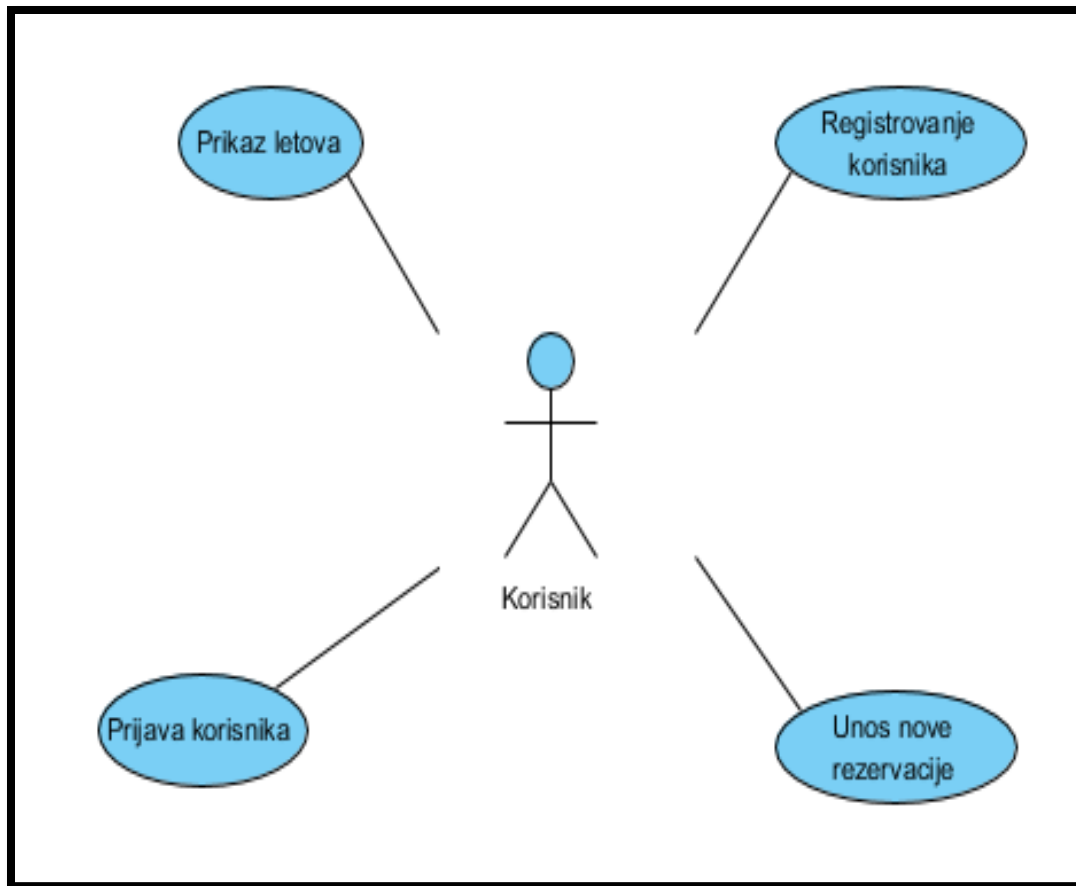
1. Prijava korisnika
2. Unos leta
3. Izmena leta
4. Pretraga rezervacija
5. Unos nove slike u galeriju
6. Unos nove rezervacije
7. Brisanje rezervacije
8. Prikaz grafika
9. Prikaz letova

**Korisniku** omogućiti sledeće funkcionalnosti:

1. Prijava korisnika
2. Registrovanje korisnika
3. Unos nove rezervacije
4. Prikaz letova



Slika 1 Slučajevi korišćenja za administratora



Slika 2 Slučajevi korišćenja za korisnika

# 1. Opis sistema

## 1.1 Slučajevi korišćenja

### 2.1.1. SK Prijava korisnika

#### **Naziv SK**

*Prijavljivanje na sistem*

#### **Aktori Sk**

*Korisnik*

#### **Učesnici SK**

*Korisnik i sistem*

#### **Preduslov**

*Sistem je uključen*

#### **Osnovni scenario SK**

1. *Korisnik unosi korisničko ime i lozinku (APUSO)*
2. *Korisnik poziva sistem da se uloguje (proveri podatke) (APSO)*
3. *Sistem proverava podatke o korisniku (SO)*
4. *Sistem prikazuje korisniku poruku „Uspešno ulogovan korisnik“ (IA)*

#### **Alternativni scenario**

- 4.1. *Ako sistem ne može da nađe korisnika, prikazuje poruku „Neuspešno prijavljivanje.“ (IA)*

### **2.1.2. SK Unos leta**

#### **Naziv SK**

*Unos leta*

#### **Aktori Sk**

*Administrator*

#### **Učesnici SK**

*Administrator i sistem*

#### **Preduslov**

*Sistem je uključen i administrator je ulogovan. Sistem prikazuje formu za rad sa letovima.*

#### **Osnovni scenario SK**

- 1. Administrator unosi podatke o letu(APUSO)*
- 2. Administrator poziva sistem da zapamti podatke o letu(APSO)*
- 3. Sistem pamti podatke o letu (SO)*
- 4. Sistem prikazuje administratoru poruku „Uspešno unet novi let.“(IA)*

#### **Alternativni scenario**

- 4.1. Ako sistem ne može da zapamti podatke o letu, administratoru prikazuje poruku „Neuspešno unošenje novog leta.“ (IA)*



### 2.1.3 SK Pretraživanje rezervacija

#### **Naziv SK**

*Pretraživanje rezervacija*

#### **Aktori Sk**

*Administrator*

#### **Učesnici SK**

*Administrator i sistem*

#### **Preduslov**

*Sistem je uključen i administrator je ulogovan.*

#### **Osnovni scenario SK**

1. *Administrator unosi vrednost po kojoj pretražuje rezervacije(APUSO)*
2. *Administrator poziva sistem da nađe rezervacije po zadatoj vrednosti(APSO)*
3. *Sistem traži rezervacije po zadatoj vrednosti (SO)*
4. *Sistem prikazuje administratoru listu rezervacija (IA)*

#### 2.1.4 SK Izmena leta

##### **Naziv SK**

*Izmena leta*

##### **Aktori Sk**

*Administrator*

##### **Učesnici SK**

*Administrator i sistem*

##### **Preduslov**

*Sistem je uključen i administrator je ulogovan. Učitana je lista letova (mesta polaska i mesta dolaska).*

##### **Osnovni scenario SK**

- 1. Administrator bira destinacije leta čiju klasu ili cenu karte želi da menja (APUSO)*
- 2. Administrator unosi novu klasu ili cenu leta (APUSO)*
- 3. Administrator poziva sistem da zapamti nove podatke o letu (APSO)*
- 4. Sistem pamti nove podatke o letu (SO)*
- 5. Sistem prikazuje administratoru poruku „Uspešno izmenjen let“ (IA)*

##### **Alternativni scenario**

- 5.1. Ako sistem ne može da zapamti podatke o letu, prikazuje poruku „Neuspešno izmenjen let.“ (IA)*

### 2.1.5 SK Unos nove slike u galeriju

**Naziv SK**

*Unos nove slike*

**Aktori Sk**

*Administrator*

**Učesnici SK**

*Administrator i sistem*

**Preduslov**

*Sistem je uključen i administrator je ulogovan.*

**Osnovni scenario SK**

1. *Administrator bira novu sliku (APUSO)*
2. *Administrator poziva sistem da ubaci novu sliku u galeriju (APSO)*
3. *Sistem pamti novu sliku (SO)*
4. *Sistem prikazuje administratoru novu sliku u galeriji (IA)*

### 2.1.6 SK Unos nove rezervacije

#### **Naziv SK**

*Unos nove rezervacije*

#### **Aktori Sk**

*Korisnik(ili administrator)*

#### **Učesnici SK**

*Korisnik(ili administrator) i sistem*

#### **Preduslov**

*Sistem je uključen I korisnik je ulogovan. Sistem prikazuje formu za rad sa rezervacijama. Učitane su lista relacija i lista klasa .*

#### **Osnovni scenario SK**

1. *Korisnik unosi podatke o rezervaciji (APUSO)*
2. *Korisnik poziva sistem da zapamti podatke o rezervaciji (APSO)*
3. *Sistem pamti podatke o rezervaciji (SO)*
4. *Sistem prikazuje korisniku poruku „Uspešno rezervisano“(IA)*

#### **Alternativni scenario**

- 4.1. *Ako sistem ne može da zapamti podatke o rezervaciji, korisniku prikazuje poruku „Neuspešna rezervacija“ (IA)*

### 2.1.7 SK Brisanje rezervacija

#### **Naziv SK**

*Brisanje rezervacija*

#### **Aktori Sk**

*Administrator*

#### **Učesnici SK**

*Administrator i sistem*

#### **Preduslov**

*Sistem je uključen i administrator je ulogovan. Sistem prikazuje formu za rad sa rezervacijama.*

#### **Osnovni scenario SK**

1. *Administrator unosi vrednost po kojoj pretražuje rezervacije (APUSO)*
2. *Administrator poziva sistem da pronade rezervacije po zadatoj vrednosti (APSO)*
3. *Sistem traži rezervacije po zadatoj vrednosti (SO)*
4. *Sistem prikazuje administratoru listu rezervacija (IA)*
5. *Administrator bira rezervaciju koju će obrisati (APUSO)*
6. *Administrator poziva sistem da obriše izabranu rezervaciju (APSO)*
7. *Sistem briše rezervaciju (SO)*

#### **Alternativni scenario**

- 7.1 *Ako sistem ne može da obriše rezervaciju, administratoru prikazuje poruku „Došlo je do greške prilikom brisanja“ (IA)*

### 2.1.8 Prikaz grafika

**Naziv SK**

*Prikaz grafika*

**Aktori Sk**

*Administrator*

**Učesnici SK**

*Administrator i sistem*

**Preduslov**

*Sistem je uključen i administrator je ulogovan.*

**Osnovni scenario SK**

1. *Administrator poziva sistem da prikaže grafik (APSO)*
2. *Sistem obrađuje podatke i kreira grafik (SO)*
3. *Sistem prikazuje administratoru grafik (IA)*

### 2.1.9 SK Prikaz letova

**Naziv SK**

*Prikaz letova*

**Aktori Sk**

*Korisnik(ili administrator)*

**Učesnici SK**

*Korisnik(ili administrator)i sistem*

**Preduslov**

*Sistem je uključen.*

**Osnovni scenario SK**

1. *Korisnik poziva sistem da učitava listu letova (APSO)*
2. *Sistem učitava listu letova (SO)*
3. *Sistem prikazuje korisniku listu letova (IA)*

## 2.1.10 SK Registrovanje korisnika

### **Naziv SK**

*Registrovanje korisnika*

### **Aktori Sk**

*Korisnik*

### **Učesnici SK**

*Korisnik i sistem*

### **Preduslov**

*Sistem je uključen. Sistem prikazuje formu za registrovanje novog korisnika.*

### **Osnovni scenario SK**

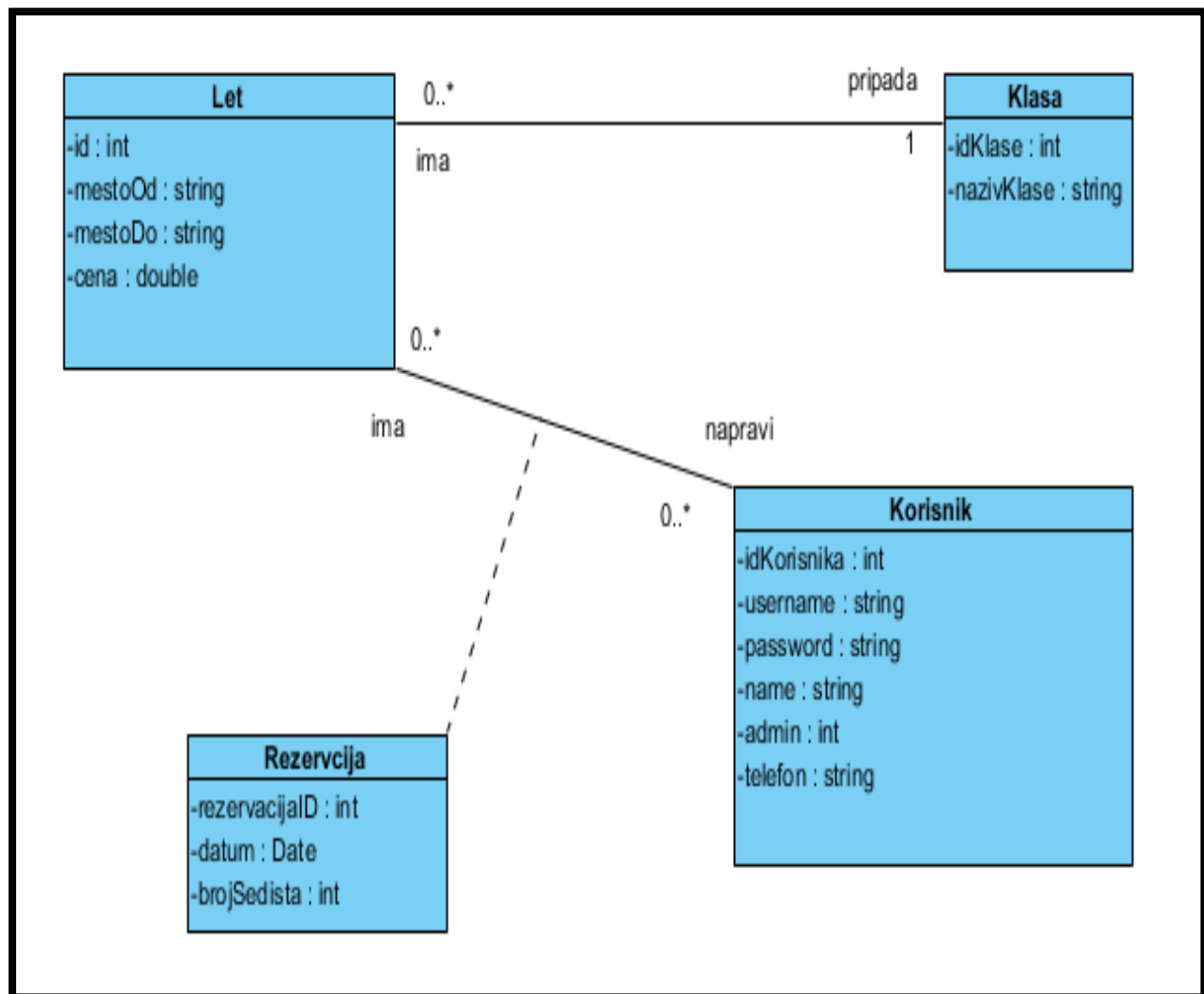
1. *Korisnik unosi podatke (APUSO)*
2. *Korisnik poziva sistem da zapamti podatke (APSO)*
3. *Sistem pamti podatke (SO)*
4. *Sistem prikazuje korisniku poruku „Uspešno registrovan korisnik“(IA)*

### **Alternativni scenario**

- 4.1. *Ako sistem ne može da zapamti podatke o korisniku, prikazuje poruku „Neuspešna registracija korisnika.“(IA)*



## 1.2 Opis strukture sistema



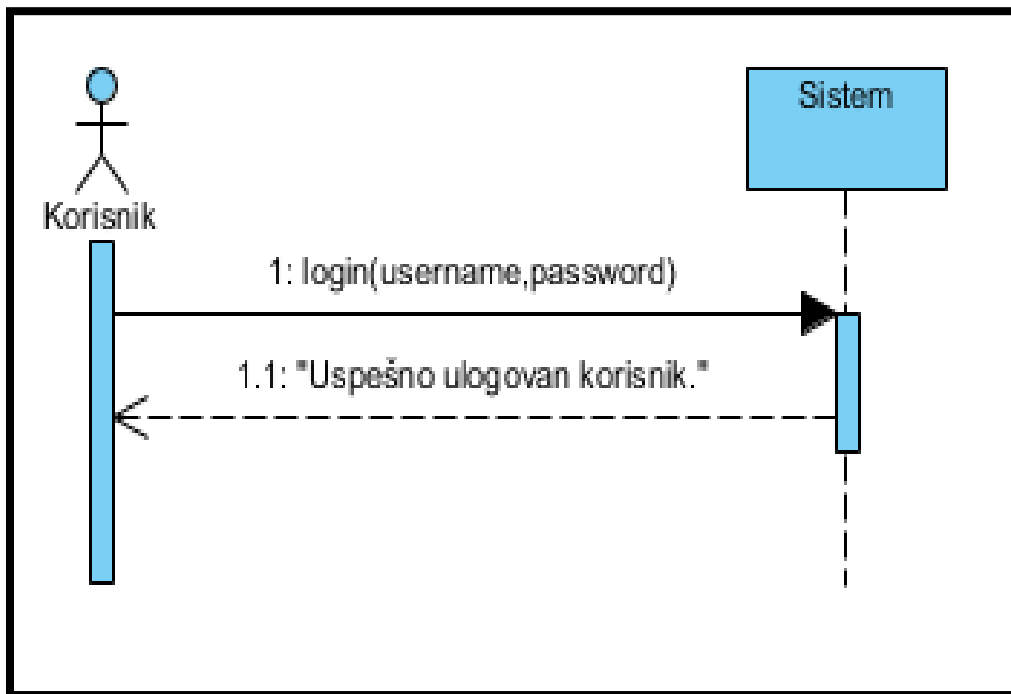
Slika 3 Dijagram klasa

## 1.3 Opis procesa – dijagram sekvenci

### 2.3.1 DS Prijava korisnika

#### Osnovni scenario SK

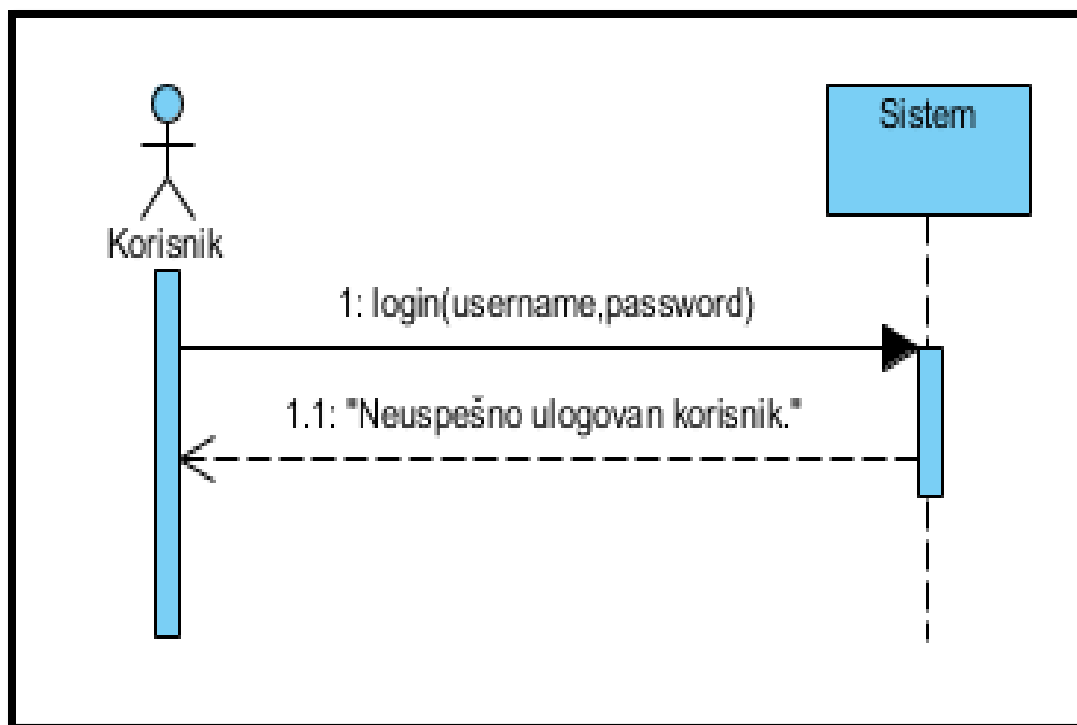
1. Korisnik poziva sistem da se uloguje (proveri podatke) (APSO)
2. Sistem prikazuje korisniku poruku „Uspešno ulogovan korisnik“ (IA)



Slika 4 DS uspešno logovanje

#### Alternativni scenario

- 2.1. Ako sistem ne može da nađe korisnika, prikazuje poruku „Neuspešno logovanje korisnika.“(IA)

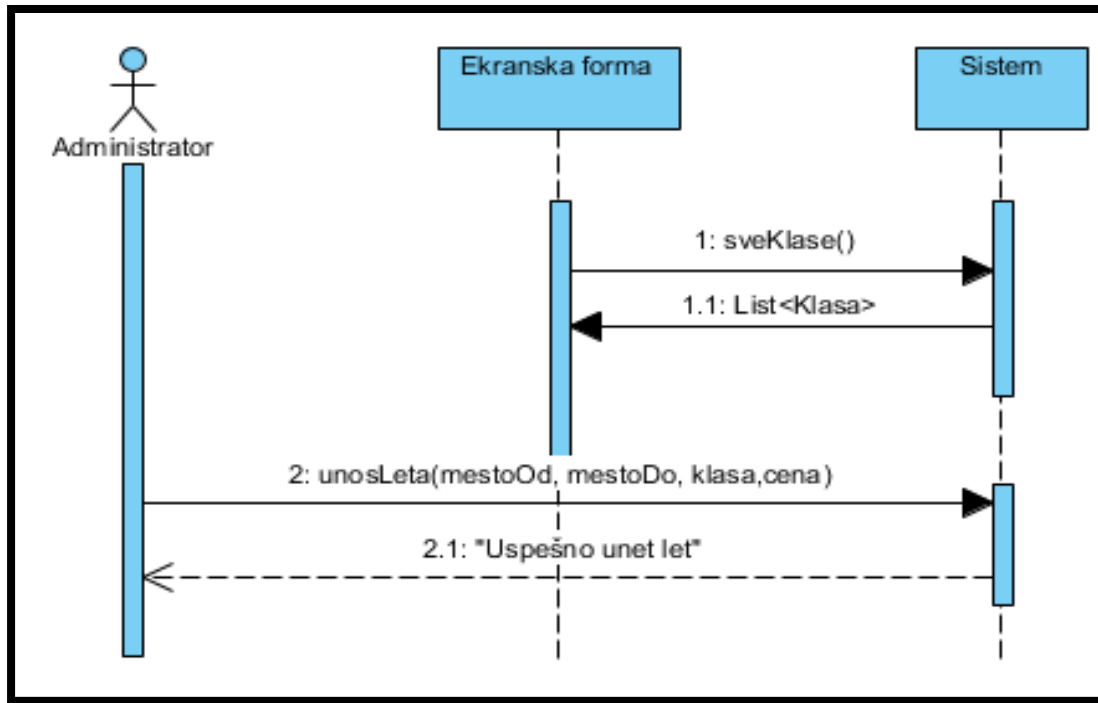


Slika 5 DS neuspešno logovanje

### 2.3.2 DS Unos leta

#### Osnovni scenario SK

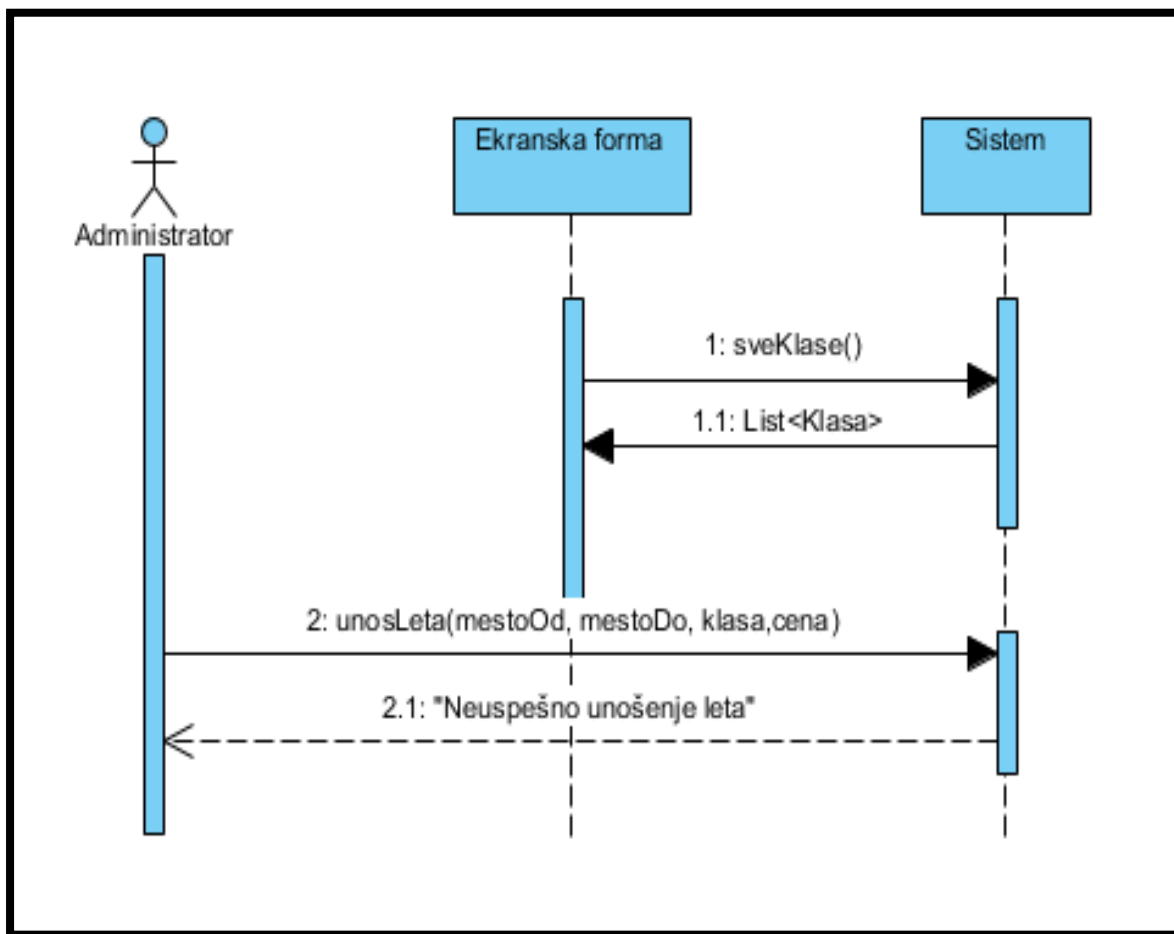
1. Administrator poziva sistem da zapamti podatke o letu(APSO)
2. Sistem prikazuje administratoru poruku „Uspešno unet let“(IA)



Slika 6DS unos leta

## Alternativni scenario

- 2.1. Ako sistem ne može da zapamti podatke o letu, administratoru prikazuje poruku „Neuspešno unošenje novog leta“ (IA)

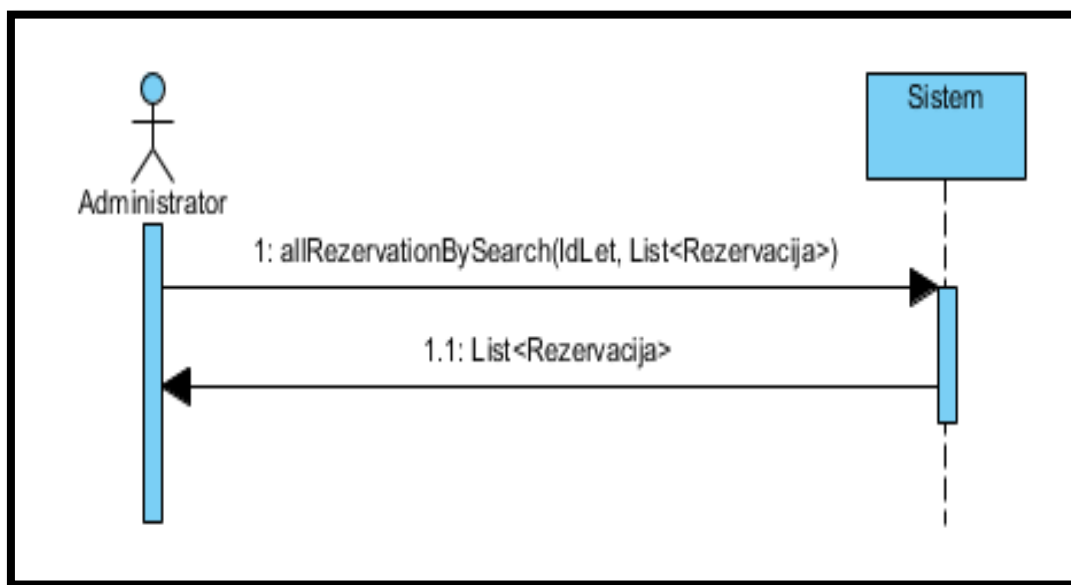


Slika 7 DS neuspesno unosenje leta

### 2.3.3 DS Pretraživanje rezervacija

#### Osnovni scenario SK

1. Administrator poziva sistem da nađe rezervacije po zadatoj vrednosti(APSO)
2. Sistem prikazuje administratoru listu rezervacija (IA)

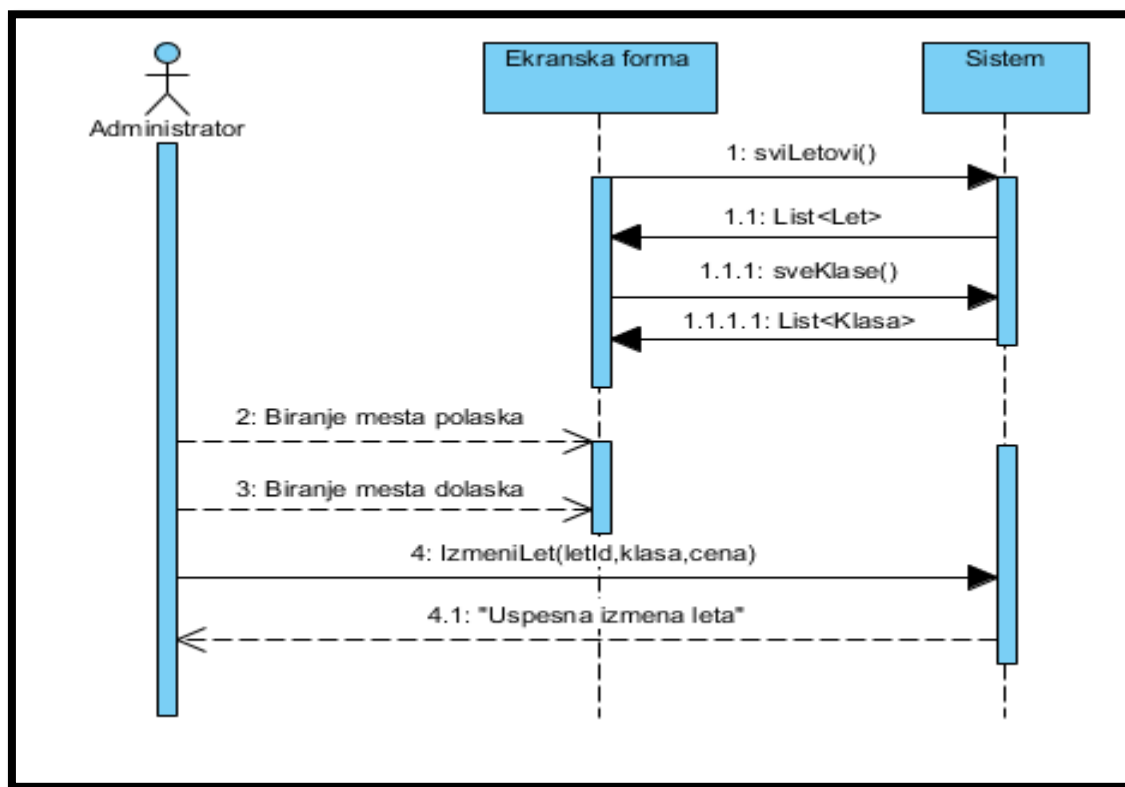


Slika 8 DS pretrazivanje rezervacija

### 2.3.4 DS Izmena leta

#### Osnovni scenario SK

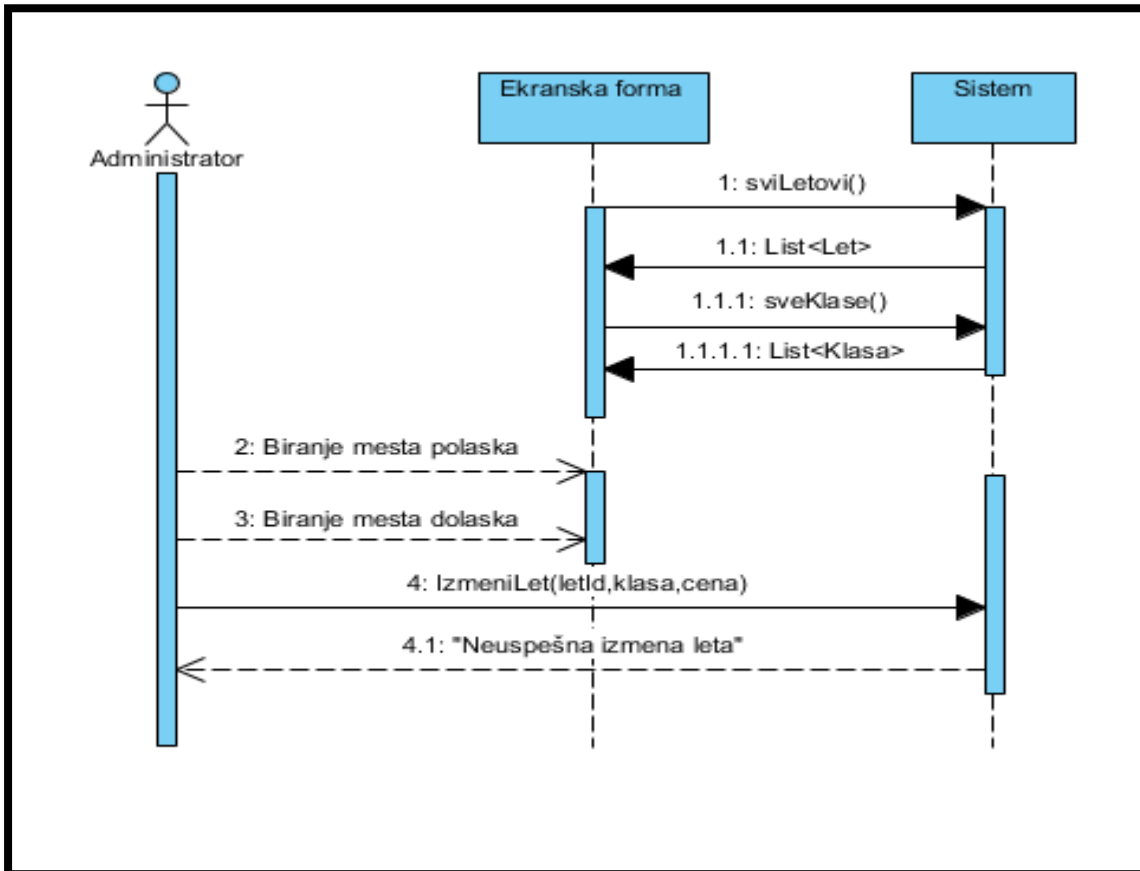
1. Ekranska forma poziva sistem da učitá listu letova, tačnije njihovih destinacija i klasa (APSO)
2. Sistem prikazuje gore navedene liste (IA)
3. Administrator poziva sistem da prikaže izabranu relaciju (APSO)
4. Sistem prikazuje izabranu relaciju (IA)
5. Administrator poziva sistem da zapamti nove podatke o letu (APSO)
6. Sistem prikazuje administratoru poruku „Uspešna izmena leta“ (IA)



Slika 9 DS uspesna izmene leta

### Alternativni scenario

- 6.1. Ako sistem ne može da zapamti podatke o letu, prikazuje poruku „Neuspešna izmena leta.“(IA)



Slika 10 DS neuspesna izmena leta



### 2.3.5 DS Unos nove slike u galeriju

#### Osnovni scenario SK

1. Administrator poziva sistem da ubaci novu sliku u galeriju (APSO)
2. Sistem prikazuje administratoru novu sliku u galeriji (IA)

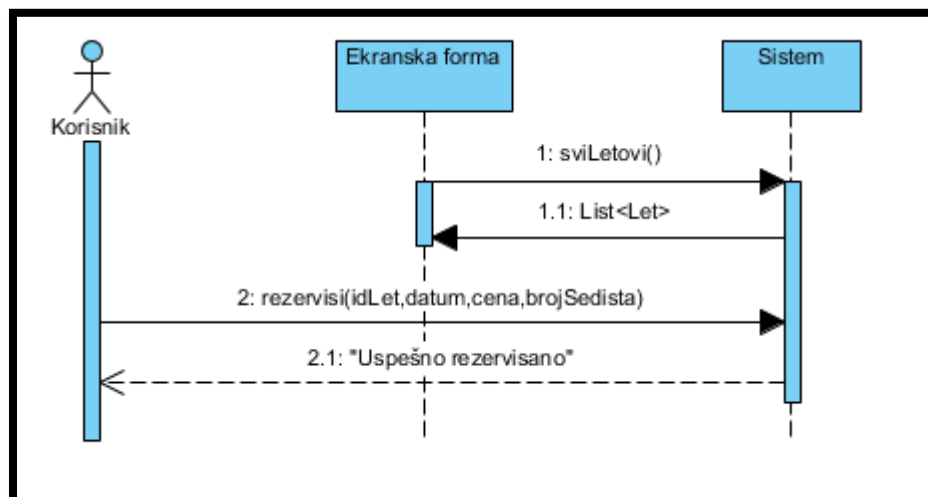


Slika 11 DS prikaz slike

### 2.3.6 DS Unos nove rezervacije

#### Osnovni scenario SK

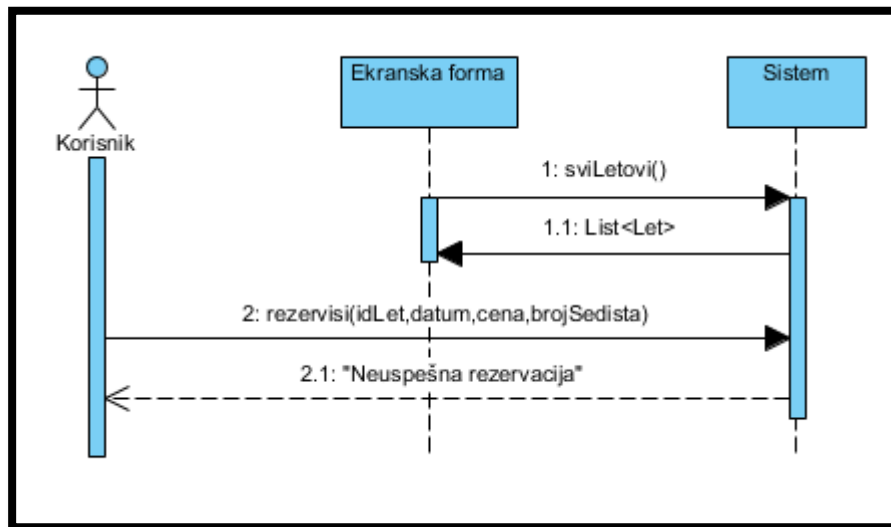
1. Korisnik poziva sistem da zapamti podatke o rezervaciji (APSO)
2. Sistem prikazuje korisniku poruku „Uspešno rezervisano“(IA)



Slika 12 DS uspesno rezervisano

### Alternativni scenario

- 2.1. Ako sistem ne može da zapamti podatke o rezervaciji, korisniku prikazuje poruku „Neuspešna rezervacija“ (IA)

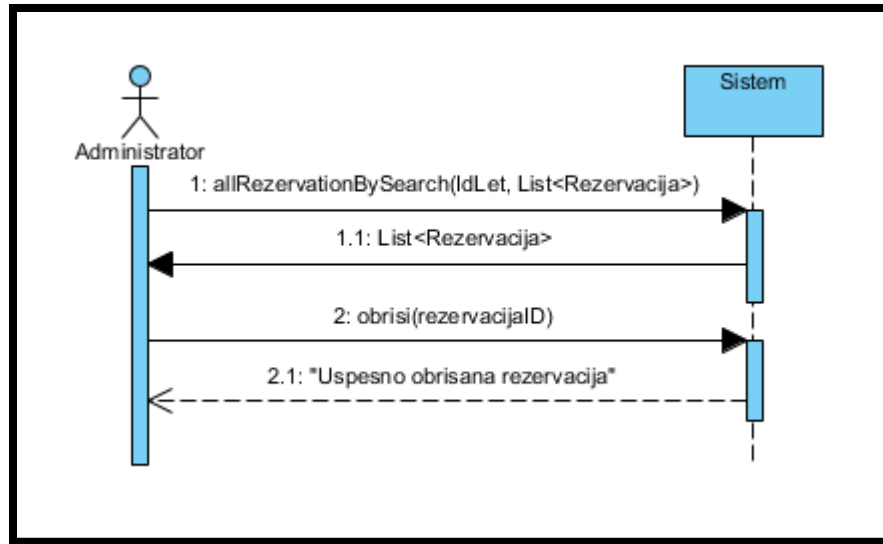


Slika 13 DS neuspesno rezervisano

### 2.3.7 DS Brisanje rezervacije

#### Osnovni scenario SK

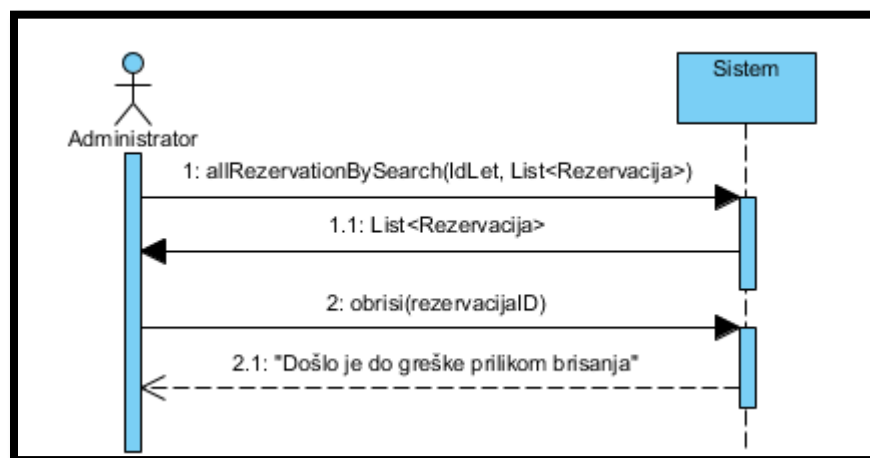
1. Administrator poziva sistem da pronade rezervacije po zadatoj vrednosti (APSO)
2. Sistem prikazuje administratoru listu rezervacija (IA)
3. Administrator poziva sistem da obriše izabranu rezervaciju (APSO)



Slika 14 DS uspesno obrisana rezervacija

### Alternativni scenario

- 3.1. Ako sistem ne može da obriše rezervaciju, administratoru prikazuje poruku „Došlo je do greške prilikom brisanja“ (IA)

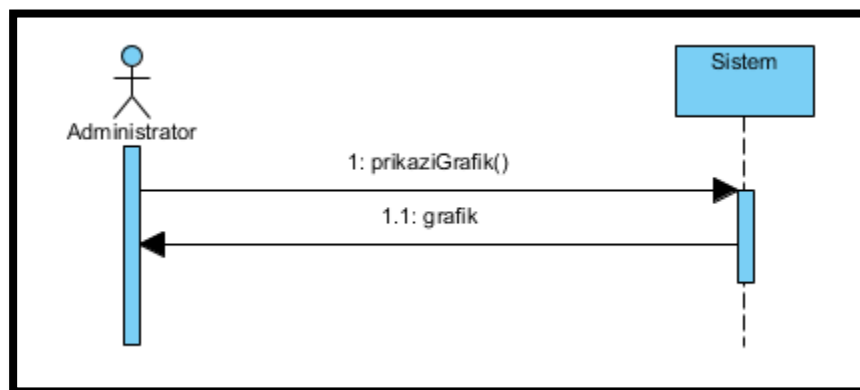


Slika 15 DS neuspesno brisanje

### 2.3.8 DS Prikaz grafika

#### Osnovni scenario SK

1. Administrator poziva sistem da prikaže grafik (APSO)
2. Sistem prikazuje administratoru grafik (IA)

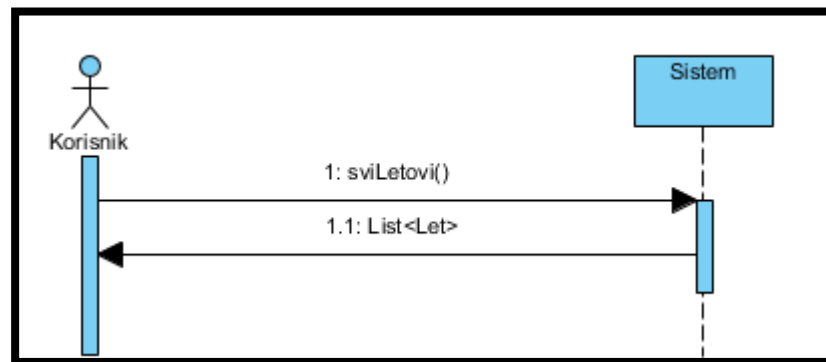


Slika 16 DS prikaz grafika

### 2.3.9 DS Prikaz letova

#### Osnovni scenario SK

1. Korisnik poziva sistem da učitava listu letova (APSO)
2. Sistem prikazuje korisniku listu letova (IA)

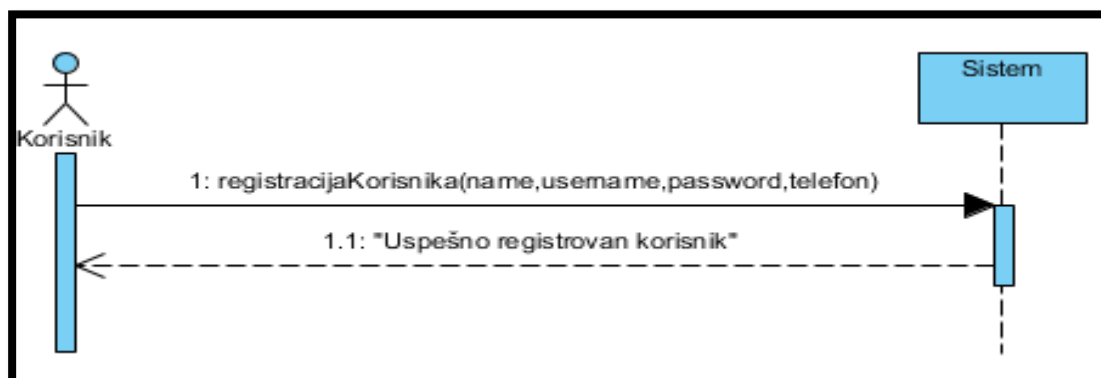


Slika 17 DS prikaz letova

### 2.3.10 DS Registrovanje korisnika

#### Osnovni scenario SK

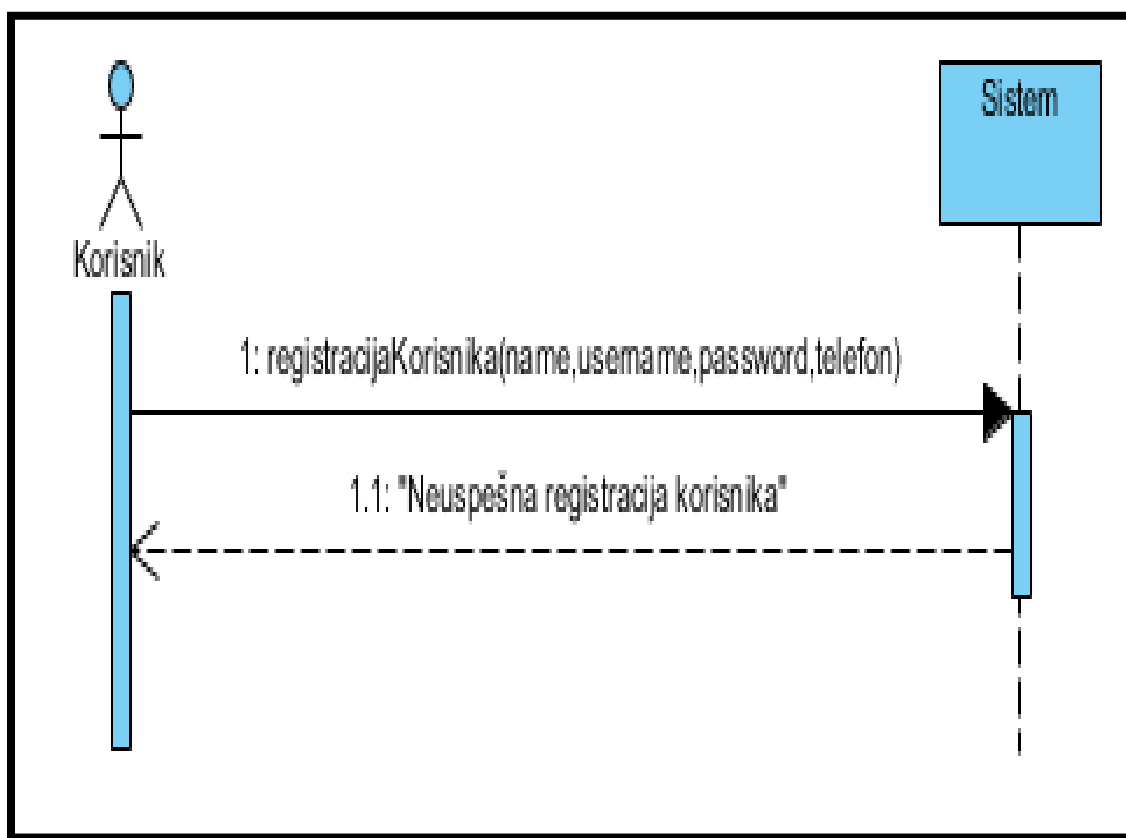
1. Korisnik poziva sistem da zapamti podatke (APSO)
2. Sistem prikazuje korisniku poruku „Uspešno registrovan korisnik“(IA)



Slika 18 DS uspesno registrovan korisnik

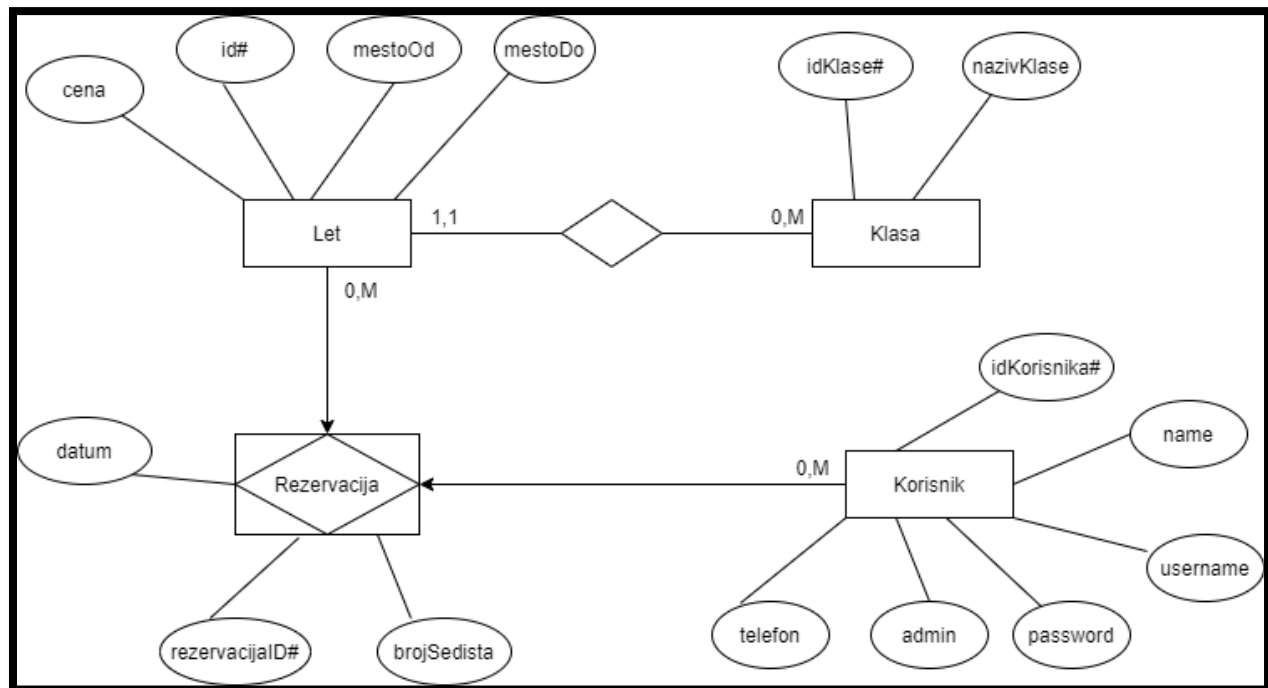
### Alternativni scenario

- 2.1. Ako sistem ne može da zapamti podatke o korisniku, prikazuje poruku „Neuspešna registracija korisnika.“(IA)



Slika 19 DS neuspesna registracija korisnika

## 1.4 Model podataka



Slika 20 PMOV

## 2.5. Specifikacija REST APIja

### 2.5.1 Funkcija sviLetovi()

OPIS FUNKCIJE	VRAĆANJE SVIH VRSTA LETOVA
HTTP METODA	GET
URL	/letovi.json
URL PARAMETRI	(nema)
HTTP BODY PARAMETRIC	(nema)
FORMAT HTTP BODY PARAMETARA	(nema)
IZLAZNI PARAMETRIC	JSON objekat sa atributima id[int],mestoOd[varChar],mestoDo[varchar(50)],klasa[int], cena[double]  Primer:  <pre>{     "id": "7",     "mestoOd": "Beograd",     "mestoDo": "Pariz",     "klasa": "3",     "cena": "36014" }</pre>
FORMAT IZLAZNIH PARAMETARA	application/json



### 2.5.2 Funkcija sveKlase()

OPIS FUNKCIJE	VRAĆANJE SVIH TIPOVA KLASA
HTTP METODA	GET
URL	/klase.json
URL PARAMETRI	(nema)
HTTP BODY PARAMETRI	(nema)
FORMAT HTTP BODY PARAMETARA	(nema)
IZLAZNI PARAMETRI	JSON objekat sa atributima idKlase[int],nazivKlase[varchar]  Primer: <pre>{     "idKlase": "3",     "nazivKlase": "biznis" }</pre>
FORMAT IZLAZNIH PARAMETARA	application/json

### 2.5.3 Funkcija ubaciKorisnika(data)

OPIS FUNKCIJE	DODAVANJE NOVOG KORISNIKA
HTTP METODA	POST
URL	/ubaciKorisnika.json
URL PARAMETRI	(nema)
HTTP BODY PARAMETRI	JSON objekat sa atributima idKorisnik[int], name[varchar], username[varchar], password[varchar], admin[int], telefon[varchar], Primer: { "idKorisnika": "1", "name": "Marija Bjelopetrovic", "username": "marija", "password": "marija97", "admin": "1", "telefon": "063418418" }
FORMAT HTTP BODY PARAMETARA	application/json
IZLAZNI PARAMETRI	JSON objekat sa jednim atributom, poruka[string].  Primer: { "poruka": "Uspesno!" }
FORMAT IZLAZNIH PARAMETARA	application/json

#### 2.5.4 Funkcija unosLeta(data)

OPIS FUNKCIJE	DODAVANJE NOVOG LETA
HTTP METODA	POST
URL	/unosLeta.json
URL PARAMETRI	(nema)
HTTP BODY PARAMETRI	JSON objekat sa atributima id[int], mestoOd[varChar], mestoDo[varChar], klasa[int], cena[double]  Primer: <pre>{   "id": "8",   "mestoOd": "Beograd",   "mestoDo": "Rim",   "klasa": "2"   "cena": "47600" }</pre>
FORMAT HTTP BODY PARAMETARA	application/json
IZLAZNI PARAMETRI	JSON objekat sa jednim atributom, poruka[string].  Primer: <pre>{   "poruka": "Uspesno!" }</pre>
FORMAT IZLAZNIH PARAMETARA	application/json

### 3. Korišćene tehnologije

Kod izrade sajta korišćene susledeće tehnologije:

- PHP
- HTML
- CSS
- Java Script
- Bootstrap
- AJAX
- DataTables

U par rečenica, prokomentarišaćemo svaku od navedenih tehnologija.

- I. **PHP** je open source jezik koji se koristi za razvoj dinamičkog web sadržaja i izvodi se na strani servera. PHP se koristi za upravljanje dinamičkim sadržajem, praćenje sesija, takođe tu je i interakcija se velikim brojem relacionih baza podataka, konkretno PHP vam omogućava da napravite krajnje e-commerce sajtove. Rasmus Lerdorf je izbacio prvu verziju PHP-a davne 1994. godine. PHP je stekao popularnost zbog svoje jednostavnosti i sintakse nasleđene iz programskog jezika C.
- II. **HTML** je akronim za **HyperText Markup Language** i predstavlja standardizovan jezik koji se koristi prilikom struktuiranja tekstova, medija i ugradjenih objekata u web stranice. Prikaz hiperteksta omogućava web pretraživač. Prvi HTML se spominje 1991, a originalna verzija je izašla juna 1993. i kreirana je od strane Tim Berners-Lee. Sam jezik sastoji se od tagova i teksta koji omogućavaju da internet pretraživač prepozna strukturu dokumenta i ispravno je prikaže korisniku.
- III. **CSS -Cascading HTML StyleSheets** je jezik za formatiranje pomoću kog se definiše izgled samih elemenata web stranice. Konkretno HTML je služio da definiše kompletan izgled, strukturu i sadržaj web strnice, i to je činio sve do HTML verzije 4.0. Od ove verzije uveden je CSS koji bi definisao konretnan izgled, dok je HTML ostao u funkciji definisanja strukture i sadržaja.
- IV. **JavaScript** je programski jezik HTML-a i web-a. Bitan deo je web aplikacija jer omogućava interaktivnost aplikacija. Java Script je skriptni jezik, odnosno sastoji se od niza komandi koje se očitavaju u interepertoaru, bez prethodnog kompajliranja i podržava objektno-orjentisano programiranje.

- V. **Bootstrap**- Bootstrap je interfejs između serverske strane i korisnika, što ga čini front-end frejmworkom. Trenutno je najpopularniji frejmwork za izradu web sajtova i web aplikacija, ispred Zurb Foundation-a. Sadrži HML i CSS šablone za dizajn, kao i JavaScript ekstenzije. Jedna od najvažnijih uloga jeste omogućavanje responzivnosti sajta. Takođe podrazumeva identičan pregled sadržaja na svim web-browserima (Mozilla Firefox, Chrome, Opera, Internet Explorer,...). Jednostavan je za primenu, međutim jedna od njegovih mana jeste ta što korišćenjem njegovih elemenata može doći do toga da su pojedini web sadržaji sličnog izgleda. Pored toga, zahteva više resursa pa može otežati očitavanje stranica
- VI. **AJAX** je akronim za **Asynchronous JavaScript And XML**. AJAX predstavlja web razvojnu tehniku za kreiranje interaktivnih web sadržaja. AJAX tehnologija se zasniva na razmeni malih količina podataka sa serverom, kako korisnik ne bi stalno morao da osvežava stranicu. Na taj način se omogućava potpuna interaktivnost, efikasnost i poboljšava funkcionisanje stranice uopšte. AJAX nije tehnologija sam po sebi, već termin koji se odnosi na korišćenje grupe tehnologija.
- VII. **REST web servisi**: REST (Representational State Transfer) predstavlja stil arhitekture aplikacije, koji se zasniva na često korišćenim tehnologijama i protokolima, od kojih je najvažniji HTTP u kombinaciji sa URL-om. Koristi osnovne HTTP-ove metode kao što su GET,POST,PUT,DELETE. Osnovni koncept REST-a je resurs, koji može biti informacija, dokument, slika, itd. Svaki resurs ima svoj identifikator, koji je URL, pomoću koga se pristupa i manipuliše tim resursom. Pri kreiranju sopstvenog REST servisa korišćen je framework Flight. Takođe korišćeni su i dostupni API-ji za prikazivanje vremenske prognoze i generisanje lozinke pri registraciji novog korisnika
- VIII. **DataTables** predstavlja besplatan JQuery plugin koji služi za tabelarni prikaz različitih podataka. Korišćenjem ovog plugina omogućena je pretraga podataka bez većih dodatnih podešavanja, kao i paginacija. Ovo je fleksibilno rešenje i omogućava dodavanje naprednih kontrola svakoj HTML tabeli. DataTables plugin može prikazivati podatke iz HTML tabele (korišćenjem HTML DOM-a) ili podatke sa servera, uz upotrebu AJAX tehnologije. Osim paginacija, DataTables plugin omogućava i brza pretraga i sortiranje po različitim kolonama, podržava različite izvore podataka kao što su HTML DOM,JavaScript, Ajax i procesiranje podataka na serverskoj strani. Izgled tabele može se lako urediti primenom: DataTables tema, JQuery UI tema, ili BootStrap-a.

## 4.Uputstvo za korisnike

Učitavanjem web aplikacije, najpre nam se prikazuje početna strana index.php, koja neulogovanom korisniku omogućava da se uloguje ili da pristupi početnoj stranici na kojoj se nalazi korisnički meni. Neulogovan korisnik ne može da vrši rezervaciju letova.

Ukoliko stranici pristupa neulogovani korisnik, ona sadrži meni, odakle korisnik može da izabere preostale stranice naše aplikacije, među kojima su:

- POČETNA
- O NAMA
- GALERIJA
- VREMENSKA PROGNOZA
- REGISTRACIJA
- PRIJAVA NA SISTEM

Kartica O NAMA nudi prikaz svih dodatnih informacija ukoliko se neko od naših potencijalnih korisnika odluči da baš nama ukaže poverenje. Kartica GALERIJA sadrži vizuelni prikaz naših aviona. U kartici VREMENSKA PROGNOZA dobićete informacije o tome kakvo nas vreme očekuje u Beogradu u narednih 10 dana. Ukoliko se odlučite da preko nas rezervišete karte za Vaše omiljene destinacije ono što je neophodno je da se najpre registrujete na našu aplikaciju u kartici REGISTRACIJA. Kada se jednom registrujete na našu aplikaciju, svaki sledeći put našoj ponudi moći ćete da pristupite tako što se ulogujete na kratici PRIJAVA NA SISTEM.




Slika 21 - Prikaz početne stranice HOME

U okviru početne stranice imamo još i ponudu trenutno dostupnih letova.

HAPPY AIRLINE					POČETNA	O NAMA	GALERIJA	VREMENSKA PROGNOZA	REGISTRACIJA	PRIJAVA NA SISTEM
LETOWI										
SHOW 10 ENTRIES		SEARCH: <input type="text"/>								
ID	MESTO POLASKA	MESTO DOLASKA	KLASA	CENA						
1	BEOGRAD	AMSTERDAM	PRVA	6866						
2	BEOGRAD	ATINA	EKONOMSKA	5500						
3	BEOGRAD	BARSELONA	BIZNIS	58722						
4	BEOGRAD	FRANKFURT	EKONOMSKA	18744						
5	BEOGRAD	HELSINKI	PRVA	30452						
6	BEOGRAD	KOPENHAGEN	EKONOMSKA	10052						
7	BEOGRAD	PARIZ	BIZNIS	36014						
8	BEOGRAD	RIM	PRVA	47600						
9	BEOGRAD	DIZELDORF	BIZNIS	26810						
10	BEOGRAD	ISTANBUL	PRVA	15682						
SHOWING 1 TO 10 OF 20 ENTRIES					PREVIOUS 1 2 NEXT					

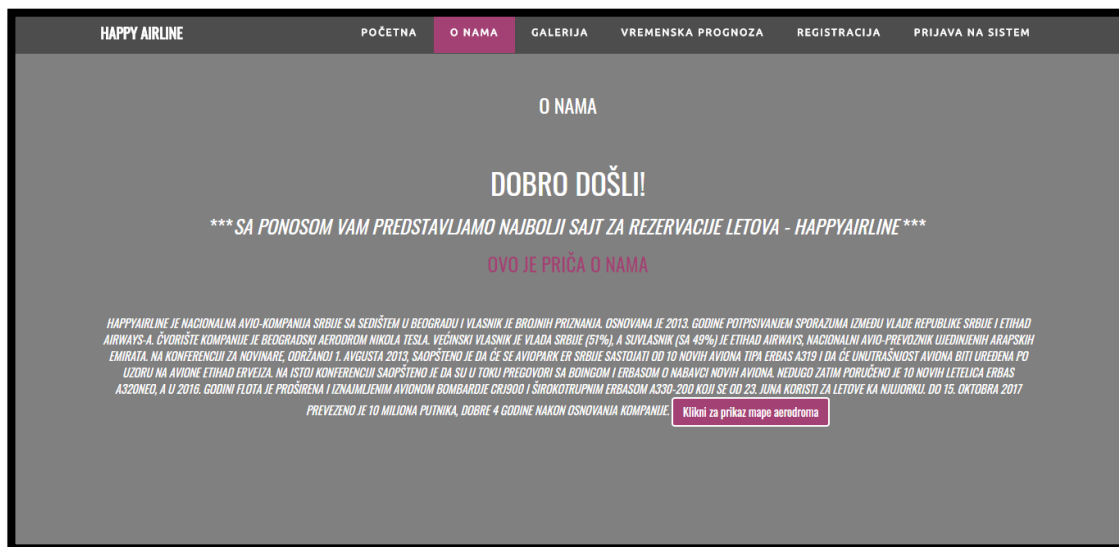
Slika 22 – Ponuda letova

Na dnu stranice možemo videti osnovne podatke vezane za našu avio-kompaniju, kao i datum, vreme i vremensku prognozu za današnji dan u Beogradu:

PODACI	
<div><h3>Belgrade Weather</h3><p>Friday 1:29 pm 7th February, 2020 Overcast Clouds  -1°C-5.56°C Humidity: 85 % Wind: 8.7 km/h</p></div>	<div><h3>Kontakt i adresa</h3><p>E-mail : happyairline@gmail.com Broj telefona: +381 11/311-21-23 Aerodrom Beograd 59 Srbija © 2020</p></div>

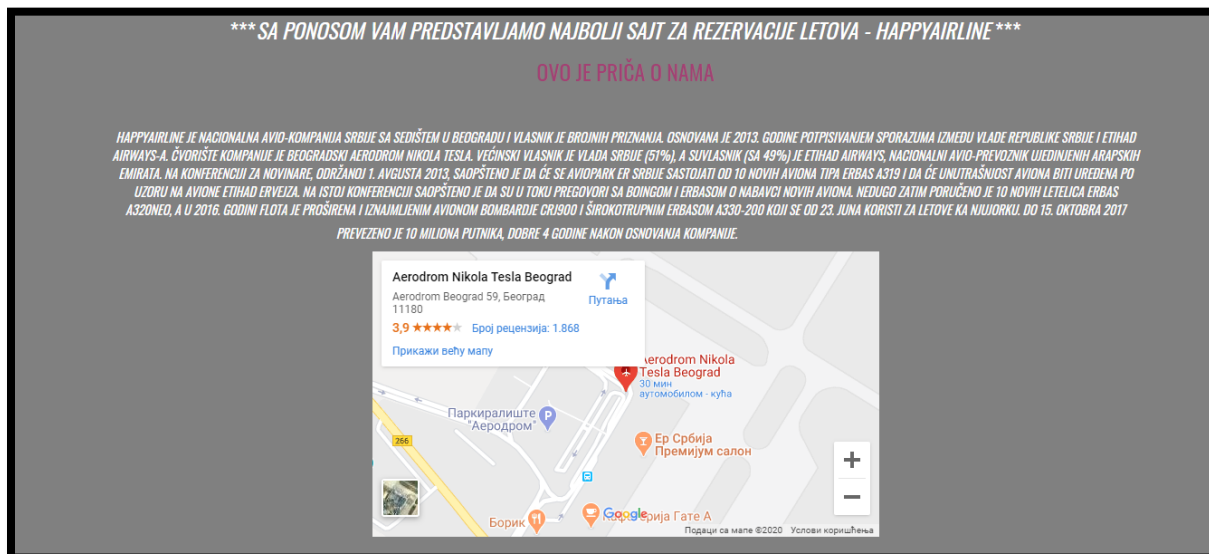
Slika 23– Osnovni podaci

Odabirom kartice O NAMA, na ekranu nam se pored već objašnjenog footera i ponude prikazuju i dodatne informacije.



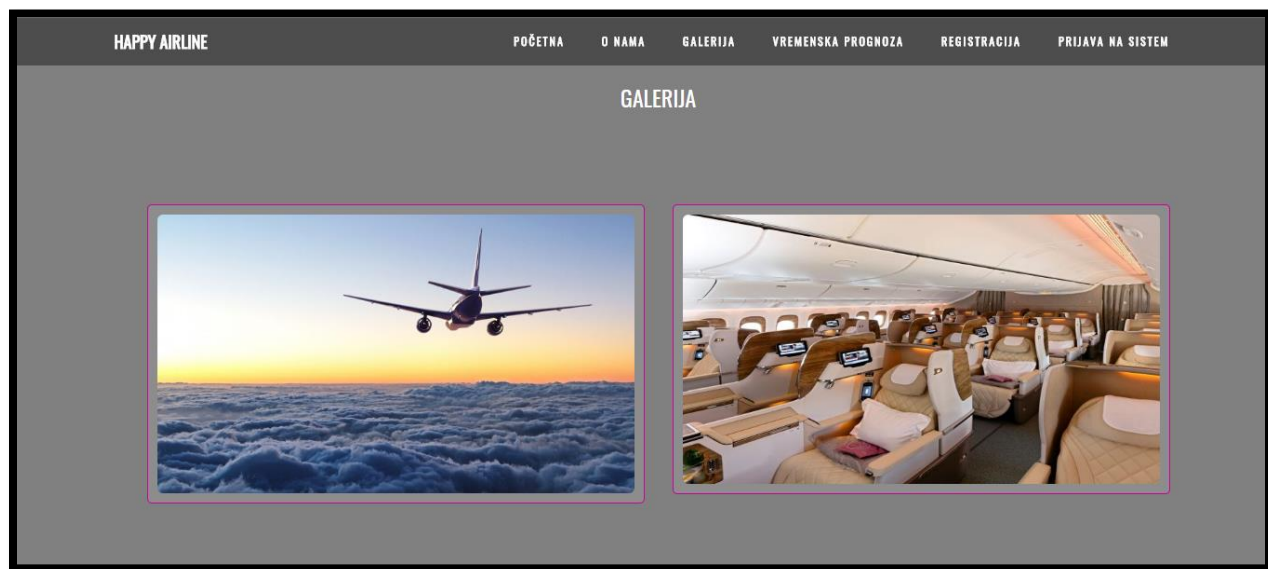
Slika 24 – Prikaz stranice O NAMA

Klikom na dugme “Klikni za prikaz mape aerodroma” dobijate tačnu lokaciju naše kompanije.





Klikom na GALERIJA u okviru menija dobijamo izgled prozora kao na slici:



Slika 25 – Prikaz stranice GALERIJA

Kartica VREMENSKA PROGNOZA prikazuje za narednih 10 dana za Beograd minimalnu i maksimalnu dnevnu temperaturu.



Slika 26- Prikaz stranice VREMENSKE PROGNOZE

Kartica REGISTRACIJA omogućava novim korisnicima da se prijave na sistem i pamti ih.

Korisnik unosi svoje ime i prezime, korisnicko ime i šifru koje će koristiti za logovanje, kao i svoj broj telefona. Neophodno je da korisnik unese sva zahtevana polja kako bi bio uspešno registrovan u bazi i to je obezbeđeno validacijom. Korisnik može izabrati svoju šifru ili kliknuti na dugme Generiši lozinku koja mu kreira novu lozinku.

Klikom na dugme Registruj se pamti se novi korisnik. U slučaju da je novi korisnik uspešno zapamćen i registrovan, na ekranu će se iznad forme za registraciju ispisati poruka “Uspesno registrovan korisnik”.

HAPPY AIRLINE

POČETNA O NAMA GALERIJA VREMENSKA PROGNOZA REGISTRACIJA PRIJAVA NA SISTEM

REGISTRACIJA

IME I PREZIME

Ime i prezime

KORISNICKO IME

Korisnicko ime

LOZINKA

Lozinka

Generiši lozinku

BROJ TELEFONA

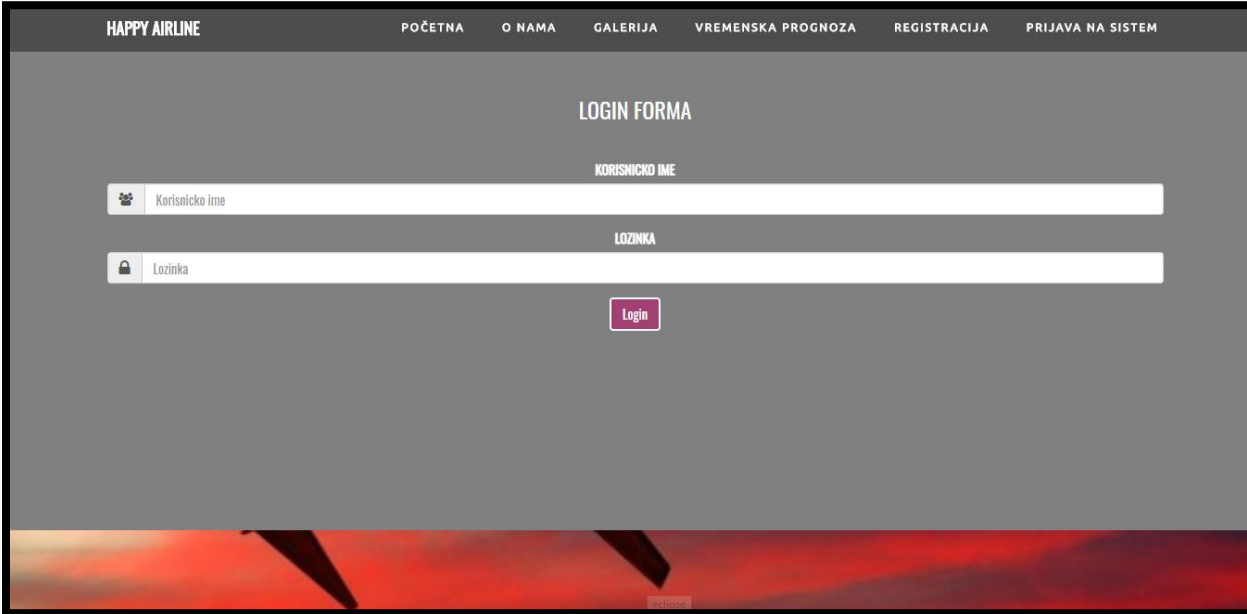
Broj telefona

Registruj se

Slika 27 – Prikaz strnice REGISTRACIJA

Kartica PRIJAVA NA SISTEM omogućava korisnicima koji su u bazi da se prijave, bilo da su administratori ili obični korisnici.

Potrebno je upisati korisničko ime i šifru i kliknuti dugme *Login*. Klikom na dugme "Login" sistem proverava da li u bazi postoji korisnik sa tim korisničkim imenom i šifrom i ako postoji korisnik je uspešno ulogovan.



Slika 28 – Prikaz stranice PRIJAVA NA SISTEM

Ulogovanom korisniku se umesto kartica REGISTRACIJA i PRIJAVA NA SISTEM prikazuju kartica ODJAVA SA SISTEMA za odjavljivanje, nakon čega se ponovo prikazuje početna strana, i kartica REZERVISI LET, koja omogućava kreiranje nove rezervacije. Ukoliko je korisnik - administrator, prikazuje mu se i kartica ADMINISTRATOR.

Ukoliko korisnik želi da rezerviše let, potrebno je da iz padajuće liste izabere mesto poletanja i mesto sletanja, zatim da iz kalendara izabere datum i da unese broj sedišta. Klikom na dugme **Rezerviši** pamt se nova rezervacija.

Slika 29 – Prikaz stranice REZERVISI LET

Kartica ADMINISTRATOR omogućava administratorima da izmene postojeću destinaciju za neki let, prikaz svih rezervacija za jedan let, unos nove slike u galeriju, unos novog leta i prikaz grafika rezervisanosti za određenu klasu.

U slučaju da želimo da izmenimo destinaciju, administrator bira iz padajuće liste mesto poletanja i mesto sletanja, a zatim bira klasu i unosi novu cenu karte. Validacijom je obezbeđeno da korisnik mora da unese novu cenu leta, odnosno ne sme da ostavi to polje prazno.

Klikom na dugme **Izmeni** let pamti se novi let.

BROJ REZERVACIJE	RELACIJA	KORISNIK	DATUM REZERVACIJE	CENA KARTE	BROJ SEDISTA	BRISANJE
10	BEOGRAD-ATINA	STEFAN JELUC	2020-06-25	5500	3	<a href="#">OBRIŠI</a>

Slika 30- Prikaz strnaice IZEMNA DESTINACIJE LETA

REZERVACIJE						
Beograd - Pariz						
BROJ REZERVACIJE	RELACIJA	KORISNIK	DATUM REZERVACIJE	CENA KARTE	BROJ SEDISTA	BRISANJE
9	BEOGRAD-PARIZ	STEFAN JELIC	2020-02-15	36014	6	<a href="#">OBRISI</a>
14	BEOGRAD-PARIZ	BOJAN PERIC	2020-02-08	36014	3	<a href="#">OBRISI</a>
16	BEOGRAD-PARIZ	MARIJA BJELOPETROVIC	2020-02-29	36014	2	<a href="#">OBRISI</a>

Slika 31 – Prikaz svih REZERVACIJA za destinaciju Beograd-Pariz

NOVA SLIKA ZA GALERIJU:

Choose File No file chosen

Ubaci

UNOS LETA

GRAD POLASKA

mesto polaska

GRAD DOLASKA

mesto dolaska

KLASA

ekonomska

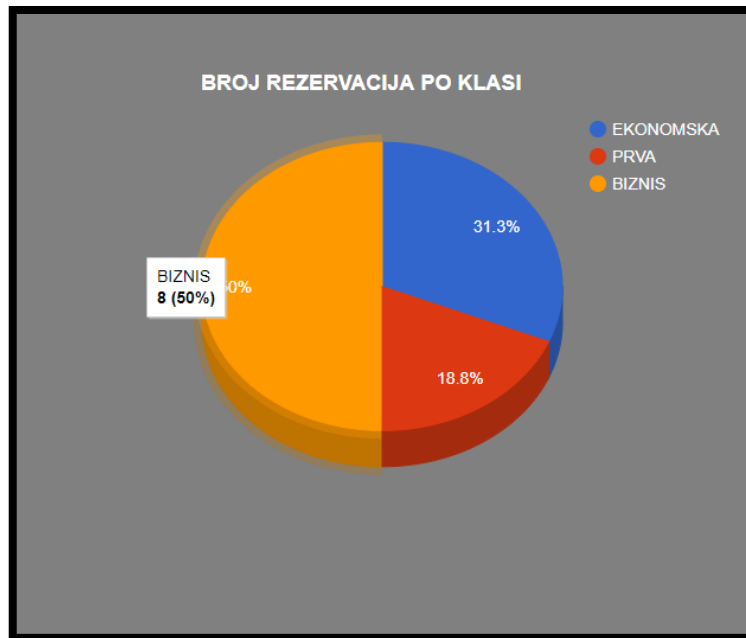
CENA KARTE

cena karte

Dodaj let

Slika 32 – Prikaz preostalih funkcija sa ADMINISTRATORA

Koristeći rezervacije dobijamo grafik broja rezervacija po klasi, prikazan na slici ispod:



Slika 33- Prikaz grafika - Broj rezervacija po klasi

## 5. Delovi koda seminarskog rada “HappyAirline”

### 5.1 Prikaz svih letova

Za prikaz letova korišćeni su PHP fajlovi:

- index.php koji prikazuje podatke,
- init.php koji omogućava konekciju na bazu odakle se čitaju podaci i
- letClass.php koji predstavlja klasu i sadrži atribute leta i funkcije.

```
<?php
    include("letClass.php");
    $let = new Let($mysqli);
    $let->sviLetovi();
    $result = $let->getResult();

    if(count($result) >0) {
        ?>
<table class="table">
<thead>
```

```

<tr>
<th class="text-center">ID</th>
<th class="text-center">Mesto polaska</th>
<th class="text-center">Mesto dolaska</th>
<th class="text-center">Klasa</th>
<th class="text-center">Cena</th>
</tr>
</thead>
<tbody>
<?php
foreach ($result as $red ) {
    $id = $red['id'];
    $mestoOd = $red['mestoOd'];
    $mestoDo= $red['mestoDo'];
    $klasa= $red['nazivKlase'];
    $cena= $red['cena'];
    ?>
<tr style="color: #fff; background: black;">
<td><?php echo $id; ?></td>
<td><?php echo $mestoOd; ?></td>
<td><?php echo $mestoDo; ?></td>
<td><?php echo $klasa; ?></td>
<td><?php echo $cena; ?></td>
</tr>
<?php
    }
?>

```

- U ovom delu koda smo koristili DataTables za prikaz podataka u tabeli i pretragu.
- DataTables koristi AJAX tehnologiju

```

<script type="text/javascript" src="https://cdn.datatables.net/v/dt/dt-1.10.16/datatables.min.js"></script>
<script>

```

```
$(document).ready(function() {  
    $('#table').DataTable();  
});  
</script>  
</body>
```

- init.php fajl za konekciju i pokretanje sesije

```
<?php  
error_reporting(E_ALL | E_STRICT);  
    ini_set("display_errors", 0);  
    ini_set("log_errors", 1);  
    ini_set("error_log", "logs.log");  
  
//instance of mysqli  
$mysqli = mysqli_connect("localhost", "root", "", "letovi");  
  
//sesija korisnik sistema, inicijalizovanje user sesije  
session_start();  
if(!isset($_SESSION['user'])){  
    $_SESSION['user'] = array();  
}  
?>
```

- **letClass.php**, za prikaz svih letova poziva se funkcija sviLetovi()

```
<?php
```



```

class Let {

    private $conn;
    private $result;

    public function __construct($conn) {
        $this->conn = $conn;
    }

    public function getResult(){
        return $this->result;
    }

    public function setResult($res){
        $this->result = $res;
    }

    public function sviLetovi() {

        $curl_zahtev = curl_init("http://localhost/rezervacije/api/letovi.json");

        curl_setopt($curl_zahtev, CURLOPT_RETURNTRANSFER, 1);
        $curl_odgovor = curl_exec($curl_zahtev);
        $json_objekat=json_decode($curl_odgovor, true);
        curl_close($curl_zahtev);
        $this->setResult($json_objekat);

    }

    public function sveKlase() {

        $curl_zahtev = curl_init("http://localhost/rezervacije/api/klase.json");

        curl_setopt($curl_zahtev, CURLOPT_RETURNTRANSFER, 1);
        $curl_odgovor = curl_exec($curl_zahtev);
        $json_objekat=json_decode($curl_odgovor, true);
        curl_close($curl_zahtev);

        $this->setResult($json_objekat);

    }
}

```

```

public function allReservationBySearch($id) {
    $id = mysqli_real_escape_string($this->conn,$id);

    $q = mysqli_query($this->conn, "SELECT * FROM rezervacija as r join let as l on
(r.let=l.id ) join klasa as k on (l.klasa=k.idKlase) join korisnik as u on (u.idKorisnik=r.korisnik)
where r.let = $id");
    $this->setResult($q);
}

public function rezervacija($letId,$datum,$brSedista) {

    $letId = mysqli_real_escape_string($this->conn,$letId);

    $datum = mysqli_real_escape_string($this->conn,$datum);
    $brSedista = mysqli_real_escape_string($this->conn,$brSedista);

    $userID=$_SESSION["user"]["idKorisnik"];
    $timestamp = date('Y-m-d H:i:s', strtotime($datum));
    if($brSedista<0 or $brSedista>60)
    {
        $this->setResult(false);
        return;
    }
    $sql = "INSERT INTO rezervacija (datum,let,korisnik,brojSedista) VALUES
('$timestamp',$letId,$userID,$brSedista)";

    if(mysqli_query($this->conn, $sql)){
        $this->setResult(true);
    }else{
        $this->setResult(false);
    };
}

public function unosLeta($gradOd,$gradDo,$klasa,$scena) {

    $gradOd = mysqli_real_escape_string($this->conn,$gradOd);
    $gradDo = mysqli_real_escape_string($this->conn,$gradDo);
    $klasa = mysqli_real_escape_string($this->conn,$klasa);
    $scena = mysqli_real_escape_string($this->conn,$scena);

    $data = Array (
"gradOd" => $gradOd,
"gradDo" => $gradDo,

```

```

    );

    $zaSlanje = json_encode($data);

    $curl_zahtev =
curl_init("http://localhost/rezervacije/api/unosLeta.json");
    curl_setopt($curl_zahtev, CURLOPT_POST, TRUE);
    curl_setopt($curl_zahtev, CURLOPT_POSTFIELDS, $zaSlanje);
    curl_setopt($curl_zahtev, CURLOPT_RETURNTRANSFER, 1);
    $curl_odgovor = curl_exec($curl_zahtev);
    $curl_odgovor = curl_exec($curl_zahtev);
    $json_objekat=json_decode($curl_odgovor, true);
    curl_close($curl_zahtev);

    if($json_objekat == "Uspesno!") {
        $this->setResult(true);
    }
    else {
        $this->setResult(false);
    }

}

public function izmenaLeta($mestoOd,$klasa,$cena) {

    $mestoOd = mysqli_real_escape_string($this->conn,$mestoOd);
    $mestoDo = mysqli_real_escape_string($this->conn,$mestoDo);
    $klasa = mysqli_real_escape_string($this->conn,$klasa);
    $cena = mysqli_real_escape_string($this->conn,$cena);

    if(mysqli_query($this->conn, "UPDATE let SET klasa=$klasa,cena='$cena' WHERE
id=$mestoOd")){
        $this->setResult(true);
    }else{
        $this->setResult(false);
    };

}

}

?>

```

## Prikaz vremenske prognoze

Za prikaz vremenske prognoze, kod koji smo pisali nalazi se u fajlu prognoza.php, a sam fajl koristi javni API. U nastavku sledi deo koda **prognoza.php** koji omogućava korišćenje API-ja.

```
<script>

    $( document ).ready(function() {

        $.getJSON('http://api.worldweatheronline.com/premium/v1/weather.ashx?key=611b337e360348da9ff120030191811&q=Belgrade&num_of_days=10&format=json')

            .done(function(data) {

                var text = '<ul>';

                $.each(data.data.weather, function(i, val) {

                    text += '<li>'+val.date+' - maksimalna temperatura: '+val.maxtempC+', minimalna temperatura: '+val.mintempC+'</li>';

                });

                text+='</ul>';

                $('#podaci').html(text);

            });

    });

</script>
```

```
    })  
    .error(function(err) { alert('Greska');  
    });  
  
});  
  
</script>
```

### Izmena podataka o letu

Za izmenu podataka o letu potrebni su:

- admin.php i
- letClass.php.

Admin.php fajl prikuplja podatke i kreira novu klasu leta koja omogućava pamćenje izmene.

Iz letClass.php fajla pozivaju se funkcije sviLetovi() i izmenaLeta(mestoOd, mestoDo, klasa, novaCena).

### **Delovi koda admin.php za izmenu podataka o letu**

```
<select class="form-control" name="letSearch" id="letSearch" onchange="search()">
<?php

        $let = new Let($mysqli);
        $let->sviLetovi();
        $result = $let->getResult();

        foreach ($result as $red ) {
            $id = $red['id'];
            $mestoOd = $red['mestoOd'];
            $mestoDo = $red['mestoDo'];
            ?>
<option value="<?php echo $id;?>"><?php echo $mestoOd.' - '.$mestoDo;?></option>
<?php
        }
        ?>
</select>

if(isset($_POST['izmeni'])) {

        $let = new Let($mysqli);

        $let->izmeniLeta(trim($_POST['mestoOd']),trim($_POST['type']),trim($_POST['novacena']));

        if($let->getResult()){

            $msg="Uspesno izmena leta";

        }else{

            $msg="Neuspesna izmena leta , takav let ne postoji u bazi.";

        }

    }
```

## Unos novog leta

Za unos leta potrebni su takođe:

- admin.php i
- letClass.php fajl.

Iz letClass.php fajla se poziva funkcija unosLeta(gradOd, gradDo, tip, cena)

### Deo admin.php fajla koji omogućava unos novog leta

```
• if(isset($_POST['unosLeta'])) {  
    $let = new Let($mysqli);  
    $let->unosLeta(trim($_POST['gradOd']),trim($_POST['gradDo']),trim($_POST['type']),trim($_POST['cena']  
));  
    if($let->getResult()){  
        $msg="Uspesno unet novi let ";  
    }else{  
        $msg="Neuspesno unosenje novog leta ";  
    }  
}
```

## Unos nove slike u galeriju

Za unos nove slike potrebni su nam sledeći fajlovi admin.php, novaSlikaGalerija.php i fajl koji služi za prikaz slika.

### Deo koda admin.php

```
<form name="galerija" method="post" action="novaSlikaGalerija.php"  
enctype="multipart/form-data">  
<div class="form-group">  
<label for="file" class="cols-sm-2 control-label">Nova slika za galeriju:</label>  
<div class="cols-sm-10">  
<div class="input-group">  
<span class="input-group-addon"><i class="fa fa-magic fa" aria-hidden="true"></i></span>  
<input type="file" class="form-control" name="file" placeholder="Ubacite sliku za galeriju"/>  
</div>  
</div>  
</div>  
<div class="form-group ">  
<input type="submit" name="file" class="btn btn-danger btn-lg " value="Ubaci">  
</div>  
</form>
```

## novaSlikaGalerija.php

```
<?php
    ini_set('display_errors', 1);

    include("init.php");
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {

        $name    = $_FILES['file']['name'];
        $tmpName  = $_FILES['file']['tmp_name'];
        $error    = $_FILES['file']['error'];
        $size     = $_FILES['file']['size'];
        $ext      = strtolower(pathinfo($name, PATHINFO_EXTENSION));

        $targetPath = dirname( __FILE__ ) . DIRECTORY_SEPARATOR. 'assets'.
        DIRECTORY_SEPARATOR. 'galerija' . DIRECTORY_SEPARATOR. $name;
        move_uploaded_file($tmpName,$targetPath);
        header( 'Location: galerija.php' );
        exit;
    }

?>
```



## Prikaz grafika

Grafik treba procentualno da prikaže broj rezervacija i popunjenost svakog leta pojedinačno. Za to su nam potrebni fajlovi:

- admin.php i
- podaciGrafik.php i
- ajax tehnologija.

**U nastavku sledi deo koda iz fajla admin.php**

```
<script>
$( function() {
    $( "#datepicker" ).datepicker({ dateFormat: 'yy-mm-dd' });
});
</script>
<script type="text/javascript" src="https://www.google.com/jsapi"></script>

<script type="text/javascript">
    google.load('visualization', '1', {'packages':['corechart']});
    google.setOnLoadCallback(grafik);

    function grafik() {
        var jsonData = $.ajax({
            url: "podaciGrafik.php",
            dataType:"json",
            async: false
        }).responseText;
        var data = new google.visualization.DataTable(jsonData);
        var options = {'title':'Broj rezervacija po klasi',
            backgroundColor: { fill:'transparent' },
            titleTextStyle: {
                textAlign: 'center',
                color: 'white',
                                fontSize: 18},
            'width':800,
            'height':500,
            is3D:true,
            legend: {
                textStyle: {
                    color: 'white'
                }
            },
        };
    }
```

```

var chart = new google.visualization.PieChart(document.getElementById('divGrafik'));
function selectHandler() {
    var selectedItem = chart.getSelection()[0];
    if (selectedItem) {
        alert( data.getValue(selectedItem.row,0));
    }
}
google.visualization.events.addListener(chart, 'select', selectHandler);

chart.draw(data, options);

}
</script>

```

### Kod iz fajla podaciGrafik.php

```

<?php
include("init.php");

$array['cols'][] = array('label' => 'Kalsa','type' => 'string');
$array['cols'][] = array('label' => 'Broj letova po klasi', 'type' => 'number');

$sql = "SELECT nazivKlase, COUNT(l.klasa) as broj FROM let l JOIN klasa k ON (l.klasa=k.idKlase)
join rezervacija r on (r.let=l.id) GROUP BY k.idKlase";
if (!$q=$mysqli->query($sql)){
    echo '{"greska":"Nastala je greška pri izvršavanju upita."}';
    exit();
} else {
    if ($q->num_rows>0){

        $niz[] = array();
        while ($red=$q->fetch_object()){
            $array['rows'][] = array('c' => array( array('v'=>$red->nazivKlase),array('v'=>(int)$red-
>broj)) );
        }

        $niz_json = json_encode ($array);
        print ($niz_json);
    } else {
        //ako nema rezultata u bazi
        echo '{"greska":"Nema rezultata."}';
    }
}

?>

```

## Registracija

Registracija nam pruža mogućnost da upamtimo novog korisnika. Fajlovi koji su nam potrebni su :

- register.php,
- userClass.php koji predstavlja klasu User-a i
- lozinka.php koji omogućava korišćenje javnog API-ja za generisanje lozinke.

**U nastavku sledi deo fajla register.php**

```
$msg = "";

if(isset($_POST['register'])) {
    require('userClass.php');
    $user = new User($mysqli);
    $user->
>registracijaKorisnika(trim($_POST['name']),trim($_POST['username']),trim($_POST['password']),trim($_
POST['telephone']));
    if($user->getResult()){
        $msg="Uspesno registrovan korisnik";
    }else{
        $msg="Neuspesna registracija korisnika ";
    }
}

//generisanje lozinke
<script>

$(document).ready(function() {

    $('#generisiLozinku').click(function() {

        $.ajax({
            url: 'lozinka.php',
            dataType: 'json',
            success: function(json) {

                $('#password').val(json);

            },
```

```

        error: function(json) {
            console.log(json,password +' error');

        }
    });

});

});
</script>

```

### Kod iz fajla lozinka.php

```

<?php
$ch = curl_init();

curl_setopt($ch, CURLOPT_URL, "https://helloacm.com/api/random/?n=10");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);

$output = curl_exec($ch);
$password = "";
for($i=1;$i<9;$i++){
    $password[$i]=$output[$i];
}
curl_close($ch);

echo json_encode($password);
?>

```

## Kod iz fajla userClass.php

```
<?php

class User {

    private $conn;
    private $result;

    public function __construct($conn) {
        $this->conn = $conn;
    }

    public function getResult(){
        return $this->result;
    }

    public function setResult($res){
        $this->result = $res;
    }

    public function registracijaKorisnika($name,$username,$password,$telephone) {

        $name = mysqli_real_escape_string($this->conn,$name);
        $username = mysqli_real_escape_string($this->conn,$username);
        $password = mysqli_real_escape_string($this->conn,$password);
        $telephone = mysqli_real_escape_string($this->conn,$telephone);

        $data = Array (
            "name" => $name,
            "username" => $username,
            "password" => $password,
            "telephone"=> $telephone
        );

        $zaSlanje = json_encode($data);

        $curl_zahtev = curl_init("http://localhost/rezervcije/api/ubaciKorisnika.json");
        curl_setopt($curl_zahtev, CURLOPT_POST, TRUE);
        curl_setopt($curl_zahtev, CURLOPT_POSTFIELDS, $zaSlanje);
        curl_setopt($curl_zahtev, CURLOPT_RETURNTRANSFER, 1);
        $curl_odgovor = curl_exec($curl_zahtev);
        $json_objekat=json_decode($curl_odgovor, true);
        curl_close($curl_zahtev);

        $this->setResult($json_objekat);
    }
}
```

```

    }
    public function login($username,$password) {

        $username = mysqli_real_escape_string($this->conn,$username);
        $password = mysqli_real_escape_string($this->conn,$password);
        $q = mysqli_query($this->conn, "select * from korisnik where username='$username'
and password='$password' limit 1");

        if(mysqli_num_rows($q)>0){
            while($red = mysqli_fetch_assoc($q)) {
                $_SESSION['user'] = $red;
            }
            $this->setResult(true);
        }else{
            $this->setResult(false);
        }
    }
}

?>

```

## Login

Omogućava prijavu korisnika koji postoje u bazi. Fajlovi koji su nam potrebni su:

- login.php i
- userClass.php koji poziva funkciju login(username, password).

### Deo koda iz fajla login.php

```

$msg = "";

if(isset($_POST['login'])) {
    require('userClass.php');
    $user = new User($mysqli);
    $user->login(trim($_POST['username']),trim($_POST['password']));
    if($user->getResult()){
        $msg="Uspesno ulogovan korisnik";
    }else{
        $msg="Neuspesan login korisnika ";
    }
}

```

## Rezervacije

Rezervacije omogućavaju korisnicima kreiranje nove rezervacije. Potrebni fajlovi su:

- rezervacija.php i
- letClass.php koji poziva funkcije sviLetovi() za prikaz ponuđenih letova i sveKlase() za prikaz svih klasa ponudi.

Na kraju poziva se funkcija iz letClass.php rezervacija(letId, datum i brSedista) koja pamti novu rezervaciju.

### Kod iz fajla rezervacija.php

```
<?php

include("init.php");
require('letClass.php');
$msg = '';

if(isset($_POST['rezervisi'])) {

    $let = new Let($mysqli);
    $let->
>rezervacija(trim($_POST['letId']),trim($_POST['datum']),trim($_POST['brSedista']));
    if($let->getResult()){
        $msg="Uspesno rezervisano";
    }else{
        $msg="Neuspesna rezervacija ! Broj sedista mora biti u opsegu 0-80";
    }
}
?>

<!DOCTYPE html>
<html lang="en" class="no-js" >
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" />
<meta name="description" content="" />
<meta name="author" content="" />

<title>Flightl booking</title>
<link href="assets/css/bootstrap.css" rel="stylesheet" />
```

```

<link href="assets/css/ionicons.css" rel="stylesheet" />
<link href="assets/css/font-awesome.css" rel="stylesheet" />
<link href="assets/js/source/jquery.fancybox.css" rel="stylesheet" />
<link href="assets/css/animations.min.css" rel="stylesheet" />
<link href="assets/css/style-solid-black.css" rel="stylesheet" />
<link href="assets/css/jquery-ui.css" rel="stylesheet" />
</head>
<body data-spy="scroll" data-target="#menu-section">

<?php

    include("meni.php");
    ?>
<?php

    include("slider.php");
    ?>

<!-- Glavna sekcija -->
<section>
<div class="container">
<div class="row text-center header">
<h3>Rezervacija</h3>
<hr />
<?php
    if($msg!=""){
        echo("<h3> ".$msg."</h3>");
    }
    ?>
<div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">
<div class="services-wrapper">
<form name="rezervacija" method="post" action="">
<div class="form-group">
<label for="let" class="cols-sm-2 control-label"> Relacije </label>
<div class="cols-sm-10">
<div class="input-group">
<span class="input-group-addon"><i class="fa fa-plane fa" aria-hidden="true"></i></span>
<select class="form-control" name="letId" >
<?php

                $let = new Let($mysqli);
                $let->sviLetovi();

```



```

        $result = $let->getResult();

        foreach ($result as $red ) {
            $id = $red['id'];
            $mestoOd = $red['mestoOd'];
            $mestoDo = $red['mestoDo'];

            ?>
<option value="<?php echo $id;?>"><?php echo $mestoOd.' - '.$mestoDo;?></option>
<?php
            }
        ?>
</select>
</div>
</div>
</div>

<div class="form-group">
<label for="date" class="cols-sm-2 control-label">Datum</label>
<div class="cols-sm-10">
<div class="input-group">
<span class="input-group-addon"><i class="fa fa-calendar fa" aria-hidden="true"></i></span>
<input id="datepicker" type="text" class="form-control" name="datum" id="datum"
placeholder="Datum"/>
</div>
</div>
</div>

                <div class="form-group">
<label for="drzava" class="cols-sm-2 control-label">Broj sedista</label>
<div class="cols-sm-10">
<div class="input-group">
<span class="input-group-addon"><i class="fa fa-magic fa" aria-hidden="true"></i></span>
<input type="text" class="form-control" name="brSedista" id="brSedista" placeholder="Broj
sedista"/>
</div>
</div>
</div>
<div class="form-group ">
<input type="submit" name="rezervisi" class="btn btn-danger btn-lg " value="Rezervisi">
</div>
</form>

```

```
</div>
</div>
</div>
</div>
</section>

<?php
    include("footer.php");
?>
<script src="assets/js/jquery-1.11.1.js"></script>
<script src="assets/js/bootstrap.js"></script>
<script src="assets/js/vegas/jquery.vegas.min.js"></script>
<script src="assets/js/jquery.easing.min.js"></script>
<script src="assets/js/source/jquery.fancybox.js"></script>
<script src="assets/js/jquery.isotope.js"></script>
<script src="assets/js/appear.min.js"></script>
<script src="assets/js/animations.min.js"></script>
<script src="assets/js/custom-solid.js"></script>
<script src="assets/js/jquery-ui.js"></script>
<script>
    $( function() {
        $( "#datepicker" ).datepicker({ dateFormat: 'yy-mm-dd' });
    });
</script>
</body>

</html>
```

## Prikaz svih rezervacija i brisanje rezervacije

U okviru prikaza svih rezervacija I brisanja imamo:

- admin.php,
- letClass.php fajl koji poziva funkciju allReservationBySearchId) koji pronalazi rezervacije za određeni let,
- generisiPodatkeSearch.php koji kreira klasu leta i prikuplja podatke i
- obriši.php koji omogućava asinhrono brisanje rezervacije.

### Deo koda iz fajla admin.php

<script>	
	function search(){
<pre>                var event = \$("#letSearch").val();                 \$.ajax({                     url: "generisiPodatkeSearch.php",                     data: "id="+event,                     success: function(result){                         var text = '&lt;table class="table"&gt;&lt;thead&gt;&lt;tr&gt;&lt;th&gt;Broj rezervacije&lt;/th&gt;&lt;th&gt;Relacija&lt;/th&gt;&lt;th&gt;Korisnik&lt;/th&gt;&lt;th&gt;Datum rezervacije&lt;/th&gt;&lt;th&gt;Cena karte&lt;/th&gt;&lt;th&gt;Broj sedista&lt;/th&gt;&lt;th&gt;Brisanje&lt;/th&gt;&lt;/tr&gt;&lt;/thead&gt;&lt;tbody&gt;';                         \$.each(\$.parseJSON(result), function(i, val) {                             text += '&lt;tr&gt;';                             text += '&lt;td&gt;'+val.rezervacijaID+'&lt;/td&gt;';                             text += '&lt;td&gt;'+val.mestoOd+'-'+val.mestoDo+'&lt;/td&gt;';                             text += '&lt;td&gt;'+val.name+'&lt;/td&gt;';                             text += '&lt;td&gt;'+val.datum+'&lt;/td&gt;';                             text += '&lt;td&gt;'+val.cena+'&lt;/td&gt;';                             text += '&lt;td&gt;'+val.brojSedista+'&lt;/td&gt;';                             text += '&lt;td&gt;&lt;a href="obrisi.php?id='+val.rezervacijaID+'"&gt;Obrisi&lt;/a&gt;&lt;/td&gt;';                             text += '&lt;/tr&gt;';                         });                         text+='&lt;/tbody&gt;&lt;/table&gt;';                         \$('#tabela').html(text);                     }                 });             }</pre>	
</script>	

#### Kod iz fajla generisiPodatke.php

```
<?php
include("init.php");
require('letClass.php');
$id=$_GET['id'];
$let = new Let($mysqli);
$let->allReservationBySearch($id);
$result = $let->getResult();
$niz = array();
$iterator = 0;
while($red = mysqli_fetch_assoc($result)) {
    $niz[$iterator] = $red;
    $iterator++;
}

echo(json_encode($niz));

?>
```

#### Kod iz fajla obrisi.php

```
<?php
include("init.php");

$id=mysqli_real_escape_string($mysqli,$_GET["id"]);

if(mysqli_query($mysqli, "delete from rezervacija where rezervacijaID=$id")){
    header("Location: admin.php");
}else{
    echo("Doslo je do greske prilikom brisanja");
};

?>
```

## Sopstveni servisi

### Kod iz fajla database.php

```
<?php
class Database {
    private $hostname = "localhost";
    private $username = "root";
    private $password = "";
    private $dbname = "letovi";
    private $dblink;
    private $result = true;
    private $records;
    private $affectedRows;

    function __construct($dbname)
    {
        $this->$dbname = $dbname;
        $this->Connect();
    }

    public function getResult()
    {
        return $this->result;
    }

    function __destruct()
    {
        $this->dblink->close();
    }

    function Connect()
    {
        $this->dblink = new mysqli($this->hostname, $this->username, $this->password, $this-
>dbname);
        if($this->dblink->connect_errno)
        {
            printf("Konekcija neuspesna: %s\n", $mysqli->connect_error);
            exit();
        }
        $this->dblink->set_charset("utf8");
    }

    function noviLet($data) {
        $mysqli = new mysqli("localhost", "root", "", "letovi");
```

```

        $mestoOd = mysqli_real_escape_string($mysqli,$data["mestoOd"]);
        $mestoDo = mysqli_real_escape_string($mysqli,$data["mestoDo"]);
        $klasa = mysqli_real_escape_string($mysqli,$data["klasa"]);
        $cena = mysqli_real_escape_string($mysqli,$data["cena"]);
        $sql = "INSERT INTO let (mestoOd,mestoDo,klasa,cena) VALUES
('$mestoOd','$mestoDo','$klasa','$cena')";

        if($mysqli->query($sql))
        {
            $this->result = true;
        }
        else
        {
            $this->result = false;
        }
        $mysqli->close();
    }

    function ubaciKorisnika($data) {
        $mysqli = new mysqli("localhost", "root", "", "letovi");
        $cols = '(name, username, password, admin, telephone)';

        $name = mysqli_real_escape_string($mysqli,$data['name']);
        $username = mysqli_real_escape_string($mysqli,$data['username']);
        $password = mysqli_real_escape_string($mysqli,$data['password']);
        $telephone = mysqli_real_escape_string($mysqli,$data['telephone']);

        $query = "INSERT into user (name, username, password, admin, telefon)
VALUES ('$name','$username','$password','0','$telephone')";

        if($mysqli->query($query))
        {
            $this->result = true;
        }
        else
        {
            $this->result = false;
        }
        $mysqli->close();
    }

    function sviLetovi() {
        $mysqli = new mysqli("localhost", "root", "", "letovi");
        $q = 'SELECT * FROM let l join klasa k on (l.klasa=k.idKlase)';
        $this->result = $mysqli->query($q);
        $mysqli->close();
    }

```

```

    }

    function sveKlase() {
        $mysqli = new mysqli("localhost", "root", "", "letovi");
        $q = 'SELECT * FROM klasa';
        $this->result = $mysqli->query($q);
        $mysqli->close();
    }

    function ExecuteQuery($query)
    {
        if($this->result = $this->dblink->query($query)){
            if (isset($this->result->num_rows)) $this->records      = $this->result-
>num_rows;
                                if (isset($this->dblink->affected_rows)) $this->affected      = $this-
>dblink->affected_rows;
                                return true;
        }
        else{
            return false;
        }
    }
}
?>

```

## Kod iz fajla index.php

```
<?php
require 'flight/Flight.php';
require 'jsonindent.php';

Flight::route('/', function(){

    echo('Popis ruta: <br>');
    echo('----- <br>');

    echo('GET rooms.json <br>');
    echo('GET letovi.json <br>');
    echo('POST ubaciKorisnika.json <br>');
    echo('POST unosLeta.json <br>');

});

Flight::register('db', 'Database', array('niz'));

Flight::route('GET /klase.json', function()
{
    header("Content-Type: application/json; charset=utf-8");
    $db = Flight::db();
    $db->sveKlase();

    $niz = array();
    $iterator = 0;
    while ($red = $db->getResult()->fetch_object())
    {
        $niz[$iterator] = $red;
        $iterator += 1;
    }

    echo indent(json_encode($niz));
});

Flight::route('GET /letovi.json', function()
{
    header("Content-Type: application/json; charset=utf-8");
    $db = Flight::db();
    $db->sviLetovi();

    $niz = array();
    $iterator = 0;
    while ($red = $db->getResult()->fetch_object())
    {
        $niz[$iterator] = $red;
        $iterator += 1;
    }
}
```



```

        echo indent(json_encode($niz));
    });

Flight::route('POST /ubaciKorisnika.json', function()
{
    header("Content-Type: application/json; charset=utf-8");
    $db = Flight::db();
    $post_data = file_get_contents('php://input');
    $json_data = json_decode($post_data,true);
    $db->ubaciKorisnika($json_data);
    if($db->getResult())
    {
        $response = "Uspesno!";
    }
    else
    {
        $response = "Greska!";
    }
    echo indent(json_encode($response));
});

Flight::route('POST /unosLeta.json', function()
{
    header("Content-Type: application/json; charset=utf-8");
    $db = Flight::db();
    $post_data = file_get_contents('php://input');
    $json_data = json_decode($post_data,true);
    $db->noviLet($json_data);
    if($db->getResult())
    {
        $response = "Uspesno!";
    }
    else
    {
        $response = "Neuspesno";
    }

    echo indent(json_encode($response));
});

Flight::start();
?>

```