

Priprema za završni test

Korišćenjem radnih okvira Spring Boot, AngularJS i Bootstrap razviti Web aplikaciju za vođenje evidencije u okviru autobuske stanice.

1.1 Aplikacija treba da obezbedi rad sa sledećim entitetima:

Prevoznik:

- Id - identifikator
- Naziv - tekstualna vrednost (jedinstvena, obavezna)
- Adresa – tekstualna vrednost
- PIB – tekstualna vrednost (jedinstvena, obavezna)

Linija:

- Id - identifikator
- Broj mesta – celobrojna vrednost (obavezna)
- Cena karte – numerička vrednost
- Vreme polaska – tekstualna vrednost
- Destinacija – tekstualna vrednost (obavezna)
- Prevoznik - veza sa instancom klase prevoznik (jedna linija može pripadati samo jednom prevozniku, a jedan prevoznik može imati više linija). Veza je dvosmerna.

1.2 Pomoću radnog okvira Spring Boot implementirati sledeći REST API:

- GET /api/prevoznici - preuzimanje svih prevoznika
- POST /api/prevoznici – dodavanje novog prevoznika
- GET /api/linije - preuzimanje svih linija (paginirano)
- GET /api/linije/{id} - preuzimanje linija po zadatom id-u
- POST /api/linije - dodavanje nove linije
- PUT /api/linije/{id} - izmena postojeće linije
- DELETE /api/linije/{id} – brisanje postojeće linije

Autobuska stanica Subotica

Dodavanje linija

Broj mesta

Cena karte

Destinacija

Vreme polaska

Prevoznik

Dodaj

Pretraga linija

Destinacija

Prevoznik

Maksimalna cena

Traži

					Prethodna	Sledeća
Naziv prevoznika	Destinacija	Broj mesta	Vreme polaska	Cena karte		
Subotica trans	Novi Sad	48	08:00	640 din.	Rezerviši	Izmeni
Niš ekspres	Beograd	51	09:00	1200 din.	Rezerviši	Izmeni
Lasta	Tavanikut	36	10:00	120 din.	Rezerviši	Izmeni
Farlang	Kraljevo	44	11:00	1100 din.	Rezerviši	Izmeni

1.3 Na nivou API-ja validirati sledeće stavke:

- Broj mesta nije negativna vrednost
- Destinacija je neprazan, ne null tekst

2.1) Obezbediti unos nove linije. Nakon unosa vrednosti u polja prikazana na slici 1. i klika na dugme dodaj, linija se preko API-ja dodaje u aplikaciju i potom se prikazuje u tabeli na slici 3. Polja forme za unos se nakon unosa prazne.

2.2) Obezbediti izmenu linije na zasebnoj stranici. Nakon klika na dugme izmeni, podaci o liniji se prikazuju u zasebnoj stranici koja omogućava čuvanje izmena. Ukoliko korisnik odabere da sačuva podatke, izmenjeni podaci se čuvaju u aplikaciji i prelazi se na stranicu za unos i prikaz linija. Dizajn zasebne stranice za izmenu linije prepušten je polazniku kursa.

2.3) Obezbediti pretragu linija, preko forme za filtriranje (slika 2). Prilikom pretrage korisnik može da unese destinaciju, prevoznika (preko opcije za selekciju (dropdown)), i gornju granicu cene. Pronalaze se i prikazuju linije koje zadovoljavaju sve kriterijuma. Ukoliko korisnik neko polje ne unese, vrednost tog polja se ignoriše u pretrazi. Filtriranje se vrši na back-end delu aplikacije.

2.4) Obezbediti paginirani prikaz podataka. Dugme prethodno i dugme sledece iznad tabele za prikaz linija (slika 3) omogućuju promenu stranice. Ukoliko se korisnik nalazi na prvoj stranici onemogućiti dugme prethodno, a, ukoliko se nalazi na poslednjoj onemogućiti dugme sledeće. Paginacija se vrši na na back-end delu aplikacije.

3. Implementirati rezervaciju karata za odgovarajuću liniju klikom na dugme Rezerviši, prikazano na slici 3. To podrazumeva:

- da se broj slobodnih mesta za određenu liniju smanji, i da se u bazi zabeleži informacija o rezervaciji. Stoga je potrebno kreirati zaseban entitet Rezervacija, koji sadrži vezu ka liniji za koju su rezervisane karte. Takođe, potrebno je prilagoditi I entitet Linija.
- Implementacija rezervacije bi trebala da bude na nivou servisa na bekendu.
- Nakon rezervacije, na neki način prikazati korisniku da je rezervacija uspela.

Rezervaciju za liniju nije moguće izvršiti ako je ostalo manje mesta nego što je zahtevano.

Dizajn REST API-ja za rezervaciju karata se prepušta polazniku.