

RAF Food Ordering System

Projekat, januarski ispitni rok

1. Pregled zadatka

Cilj rada je implementacija WEB aplikacije koja će predstavljati simulaciju naručivanja hrane. Aplikacija treba da omogući korisnicima da poručuju hranu i prate status svojih porudžbina (priprema, dostava, isporučeno, itd.).

Projekat će biti podeljen po glavnim domenskim objektima koje je potrebno implementirati: upravljanje korisnicima i upravljanje porudžbinama.

Upravljanje korisnicima je definisano u trećem domaćem zadatku. Dodatak su nove permisije za upravljanje porudžbinama.

Takođe, potrebno je omogućiti zakazivanje porudžbina (tj. mogućnost da korisnik zakaže porudžbinu za određeno vreme) kao i evidenciju grešaka koje su se desile tokom obrade zakazanih porudžbina.

2. Zahtevi

Porudžbina kao entitet sadrži sledeće obavezne atribute:

Ime	Tip	Opis
id	Long	Jedinstvena identifikacija entiteta
status	Enum(ORDERED, PREPARING, IN_DELIVERY, DELIVERED, CANCELED)	Trenutni status porudžbine
createdBy	FK_UserTable(id)	Referenca na korisnika koji je napravio porudžbinu
active	Boolean	Da li je porudžbina aktivna ili otkazana
items	List <Dish>	Lista jela koja sačinjavaju porudžbinu

Napomena: Implementacija kataloga jela (**Dish**) koja mogu da se naruče je prepuštena proizvoljnoj implementaciji. Možete koristiti jednostavan skup jela ili kreirati detaljniji meni po želji.

Svaka porudžbina pripada korisniku koji ju je kreirao. **Običan korisnik** ima uvid samo u svoje porudžbine (aktivne i prošle), dok **administrator** ima uvid u sve porudžbine svih korisnika (aktivne i prošle).

Akcije koje se mogu izvršiti nad porudžbinom:

- **SEARCH**
- **PLACE_ORDER**
- **CANCEL**
- **TRACK**
- **SCHEDULE**

Svaka akcija je ispraćena permisijom. Potrebno je proširiti skup permisija i podržati još 5 novih, za svaku od mogućih akcija nad porudžbinom: **can_search_order**, **can_place_order**, **can_cancel_order**, **can_track_order**, **can_schedule_order**. Shodno tome, korisnik bez specifične permisije nema pristup niti mogućnost za izvršavanje odgovarajuće akcije.

Operacije **PREPARING**, **IN_DELIVERY** i **DELIVERED** nisu momentalne i za njihovo izvršavanje je potreban određen vremenski period. Dok je porudžbina u nekoj od tih operacija, obezbediti da nije moguće izvršiti druge operacije nad tom istom porudžbinom.

Svaki uspešni poziv ka nekoj od funkcionalnosti mora da vrati odgovor odmah sa statusom 2xx, a ne nakon X sekundi — operacije **PREPARING**, **IN_DELIVERY** i **DELIVERED** nakon poziva odmah vraćaju odgovor, a njihovo izvršavanje na odabranoj porudžbini će se nastaviti u pozadini.

Akcije/operacije nad porudžbinama:

SEARCH

- Iz baze vraća niz porudžbina.
 - **Običan korisnik** može da vidi samo svoje porudžbine (aktivne i prošle).
 - **Administrator** može da vidi sve porudžbine svih korisnika (aktivne i prošle).
- Sledeća tabela ne predstavlja entitet već moguće parametre pretrage. Svi parametri su opcioni.

Ime	Tip	Opis
status	List<String>	Vratiti porudžbine čiji je status u jednom od poslatih
dateFrom	Date	Predstavlja početak vremenskog opsega. Samo ukoliko je i dateTo definisan onda će vratiti porudžbine čiji je momenat kreiranja između dateFrom i dateTo
dateTo	Date	Predstavlja kraj vremenskog opsega. Pročitati dateFrom
userId	Long	(Samo za administratore) ID korisnika čije porudžbine se pretražuju. Ako nije navedeno, pretražuju se porudžbine svih korisnika.

PLACE_ORDER

- Ova akcija je momentalna i služi za kreiranje nove porudžbine. Prilikom kreiranja porudžbine potrebno je proslediti sve obavezne parametre, uključujući listu jela koja se naručuju.
- Rezultat je porudžbina koja je u stanju **ORDERED**.
- Prilikom kreiranja porudžbine u sistemu potrebno je povezati ulogovanog korisnika sa njom.

CANCEL

- Akcija je momentalna i može se izvršiti samo nad porudžbinama koje su u stanju **ORDERED**.
- Ova akcija samo označava porudžbinu kao otkazanu, ali je ne briše iz baze.

TRACK

- Ova akcija omogućava korisniku da dobije trenutni status svoje porudžbine.
- *Napomena:* Na frontend-u je potrebno implementirati mehanizam koji će automatski osvežavati prikaz statusa porudžbine bez potrebe za ručnim osvežavanjem strane ili ručnog pokretanja akcije **TRACK**. To može biti realizovano korišćenjem WebSocket-a, long-polling-a ili periodičnim slanjem zahteva (polling).

SCHEDULE

- Porudžbine se mogu zakazati — korisnik može da odredi datum i vreme kada će željena porudžbina da se kreira.
- U odgovarajuće vreme, potrebno je da sistem samostalno pokuša da kreira porudžbinu, ukoliko su uslovi za njeno kreiranje ispunjeni.
- Ukoliko porudžbina nije uspela da se kreira (zbog prekoračenja maksimalnog broja istovremenih porudžbina), potrebno je to evidentirati u novoj tabeli — **ErrorMessage**.
- Entitet **ErrorMessage** mora da sadrži datum, id porudžbine, operaciju koja je bila zakazana i poruku o grešci koja se dogodila.

Statusi porudžbine i njihovo automatsko menjanje:

- Porudžbina u stanju **ORDERED** će se automatski prebaciti u stanje **PREPARING** nakon **10+ sekundi** sa implementiranom vremenskom devijacijom po nahođenju.
- Porudžbina u stanju **PREPARING** će se automatski prebaciti u stanje **IN_DELIVERY** nakon **15+ sekundi** sa implementiranom vremenskom devijacijom.
- Porudžbina u stanju **IN_DELIVERY** će se automatski prebaciti u stanje **DELIVERED** nakon **20+ sekundi** sa implementiranom vremenskom devijacijom.
- Operaciju **CANCEL** je moguće izvršiti samo nad porudžbinom koja je u stanju **ORDERED**.

Ograničenje maksimalnog broja istovremenih porudžbina:

- Sistem može da podrži maksimalno **3 istovremene porudžbine** koje su u procesu pripreme ili dostave (stanja **PREPARING** i **IN_DELIVERY**).
- Ukoliko se pokuša kreiranje nove porudžbine (bilo trenutne ili zakazane) koja bi dovela do prekoračenja ovog limita, potrebno je da sistem odbije kreiranje te porudžbine i evidentira grešku u tabeli **ErrorMessage**.
- Greška treba da sadrži poruku da je maksimalan broj istovremenih porudžbina dostignut.

Frontend

1. Stranica za pretragu porudžbina

- Implementirati da se upotrebom **SEARCH** akcije prikažu sve porudžbine, bez prosleđenih parametara kada se stranica učitava.
- **Običan korisnik** treba da vidi samo svoje porudžbine (aktivne i prošle).
- **Administrator** treba da ima mogućnost da vidi sve porudžbine svih korisnika.
- Pored toga, potrebno je implementirati formu iznad tabele koja sadrži potrebna polja da pokrije sve funkcionalnosti za pretragu koje nudi Backend. Filteri će se primeniti submitom te forme.
 - **Administrator** može da pretražuje porudžbine po korisniku (**userId** ili nekom drugom identifikatoru).
- **Automatsko ažuriranje statusa:** Prikaz porudžbina na ovoj stranici treba da reflektuje trenutni status svake porudžbine bez potrebe za ručnim osvežavanjem strane. To može biti ostvareno korišćenjem WebSocket-a, Server-Sent Events (SSE) ili periodičnim slanjem zahteva ka serveru (polling).

2. Stranica za kreiranje porudžbine

- Napraviti jednostavnu formu koja će omogućiti korisniku da odabere jela koja želi da poruči i da kreira novu porudžbinu na Backend strani.
- Ova forma neće sadržati id, status, niti korisnika; to će Backend zaključiti sam.
- **Napomena:** Implementacija kataloga jela je prepuštena vama. Možete koristiti statički definisanu listu jela, učitavati je sa servera ili omogućiti dinamičko dodavanje jela.

3. Stranica sa istorijom grešaka

- Tabela u kojoj je potrebno prikazati greške koje su se dogodile pri izvršavanju zakazanih porudžbina.
- **Običan korisnik** treba da vidi samo greške koje su se dogodile nad njegovim porudžbinama.
- **Administrator** može da vidi greške svih korisnika. Ovde je potrebno implementirati paginaciju, kako na frontendu, tako i na backendu.

Korisnik bez specifične permisije nema pristup niti mogućnost za izvršavanje odgovarajuće akcije na frontend-u.

Ukoliko korisnik nema nijednu permisiju, nakon uspešnog logovanja, obavestiti ga alertom.

3. Tehnologije

Backend:

- Spring ili JBoss
- Relaciona baza podataka

Frontend:

- Angular 2+, Vue, ili React

4. Predaja projekta

Rok za predaju projekta je **TBD**. Datum i raspored termina odbrane će biti naknadno definisani nakon isteka roka za predaju projekta.

Domaći se šalje na:

- Imitrovic@raf.rs za grupe: 404, 405, 423
- nredzic@raf.rs za grupu: 406

Subject mail-a mora da bude u obliku: "NWP projekat <Ime> <Prezime> <Indeks>". Npr. "NWP projekat Student Studentić RN 3/21". Loš format subject-a rezultuje sa -50% od ostvarenih poena. U telu mejla navesti grupu kojoj pripadate po zvaničnom spisku kao i sam rad koji može biti u obliku:

- Zip
- Link ka Drive-u
- Link ka GitHub privatnom repozitorijumu (Iukamitrovic99@gmail.com ili nikolajr93@gmail.com).

5. Odbrana i bodovanje

Odbrana projekta je obavezna. Ukoliko neko uradi pre definisanog roka, uz dogovor, moguće je odbraniti i ranije. Termin za odbranu projekta će biti u ispitnom roku, par dana pre dana predviđenog za Napredno web programiranje. Tačan termin odbrane za svakog studenta će biti objavljen ubrzo nakon isteka roka za predaju projekta.

Svrha odbrane je da se pokaže autentičnost zadatka. Ovo podrazumeva odgovaranje na pitanja u vezi načina izrade zadatka ili izvršavanje neke izmene nad zadatkom na licu mesta.

Projekat nosi **40 poena**, a uslov za polaganje predmeta je minimum **20 poena** (uz ostale uslove definisane na početku kursa). Boduje se na sledeći način:

- **PLACE_ORDER/SEARCH/CANCEL porudžbine:** 5 poena
- **Automatsko menjanje statusa porudžbine (PREPARING, IN_DELIVERY, DELIVERED) sa automatskim osvežavanjem na frontend-u:** 10 poena
- **Implementacija permisija:** 5 poena
- **Zakazivanje porudžbine i evidencija grešaka usled prekoračenja maksimalnog broja istovremenih porudžbina:** 8 poena
- **Evidentiranje i prikaz grešaka koje su se dogodile na zakazanim porudžbinama:** 7 poena
- **Celokupna funkcionalnost i ponašanje aplikacije:** 5 poena

Svaka stavka podrazumeva potpunu funkcionalnost na backend-u i frontend-u. Ukoliko neka stavka nije u potpunosti završena, moguće je dobijanje parcijalnih poena.

Srećno! :)