



*Jesenji semestar, 2021/22*

*PREDMET: SE311 PROJEKTOVANJE I ARHITEKTURA*  
*SOFTVERA*

## Projektni zadatak

Ime i prezime: **Tomislav Živadinović**

Broj indeksa: **3948**

Datum izrade: **u izradi**

Sadržaj: .....	2
----------------	---

## 1. Uvod

Razvija se aplikacija za firmu “Blinco FF” koja se bavi proizvodnjom ribolovačkog pribora. Firma trenutno kontrolu svojih porudžbina vrši pomoću zastarelog programa koji ne pruža neophodne mogućnosti. Sa druge strane, radnici i direktor firme celokupnu administraciju vezanu za zaposlene rade ručno preko papira. Iz tog razloga, često dolazi do greške i gubitka podataka. Pored toga, potrebno je mnogo vremena doći do neke bitne informacije i ovakav način rada nije efikasan. Zato se razvija nova aplikacija koja će otkloniti greške i olakšati rad, kao i poboljšati efikasnost rada.

Deo koji se tiče zahteva biće u dokumentu, ali je detaljnije opisan u SRS dokumentu koji će se kasnije pisati.

## 2. Korisnički zahtevi

U ovom delu biće opisani funkcionalni i nefunkcionalni zahtevi ovog sistema. Svi zahtevi prate istu konvenciju pisanja i po svom identifikatoru se mogu naći detaljnije opisanu u SRS dokumentu.

### 2.1 Funkcionalni zahtevi

REQ-1: Svakom korisniku treba zahtevati da se uz pomoć svojih kredencijala uloguje na sistem

REQ-2: Direktor treba omogućiti pregled svih zaposlenih u firmi

REQ-3: Direktor treba omogućiti unos podataka koji se traže za dodavanje novog zaposlenog. Direktor treba da unese ime, prezime, JMBG, fajl ugovora...

REQ-4: Direktor treba omogućiti izmenu podataka koji se traže pri izmeni zaposlenog. Direktor može da izmeni iste podatke kao i kod dodavanja novog zaposlenog.

REQ-5: Direktor treba omogućiti deaktiviranje zaposlenih na sistemu

REQ-6: Direktor treba omogućiti pregled svih zaposlenih u firmi

REQ-7: Direktor treba omogućiti unos podataka koji se traže za dodavanje novog proizvoda. Direktor treba da unese šifru, naziv, količinu na stanju i cenu.

REQ-8: Direktor treba omogućiti izmenu podataka koji se traže pri izmeni proizvoda. Direktor može da izmeni iste podatke kao i kod dodavanja novog proizvoda.

REQ-9: Direktor treba omogućiti brisanje proizvoda na sistemu

REQ-10: Direktor treba omogućiti pregled svih porudžbina u firmi

REQ-11: Direktor treba omogućiti generisanje(štampanje) svih porudžbina, faktura, profaktura i packing listi

REQ-12: Direktor treba omogućiti prihvatanje porudžbine koja je na čekanju. Direktor treba da unese dodatne informacije(vreme potrebno za dostavu) pri prihvatanju

REQ-13: Direktor treba omogućiti odbijanje porudžbine koja je na čekanju. Direktor treba da unese dodatne informacije(razlog odbijanja) pri odbijanju.

REQ-14: Sistem treba da obavesti kupca mail-om kada direktor prihvate ili odbije njegovu porudžbinu

REQ-15: Direktor treba omogućiti pregled svih izveštaja koje je moguće generisati

REQ-16: Direktor treba omogućiti generisanje izveštaja koje izabere

REQ-17: Kupcu treba omogućiti pregled svih proizvoda koje je moguće proizvesti u datom trenutku. Korisnik treba da izabere kategoriju proizvoda koje želi.

REQ-18: Kupcu treba omogućiti da kreira porudžbinu. Kupac kreira porudžbinu dodavanjem željenog proizvoda u takozvanu korpu i unošenjem količine za svaki od proizvoda

REQ-19: Kupcu treba omogućiti izbor da u već kreiranoj porudžbini doda željeni proizvod

REQ-20: Kupcu treba omogućiti potpunu kontrolu svih proizvoda koje je dodao u korpu. Kupac može da ukloni neki proizvod iz korpe ili promeni količinu proizvoda koje je naručio

REQ-21: Kupcu treba omogućiti potpuni pregled svih porudžbina koje je imao do sada na sistemu

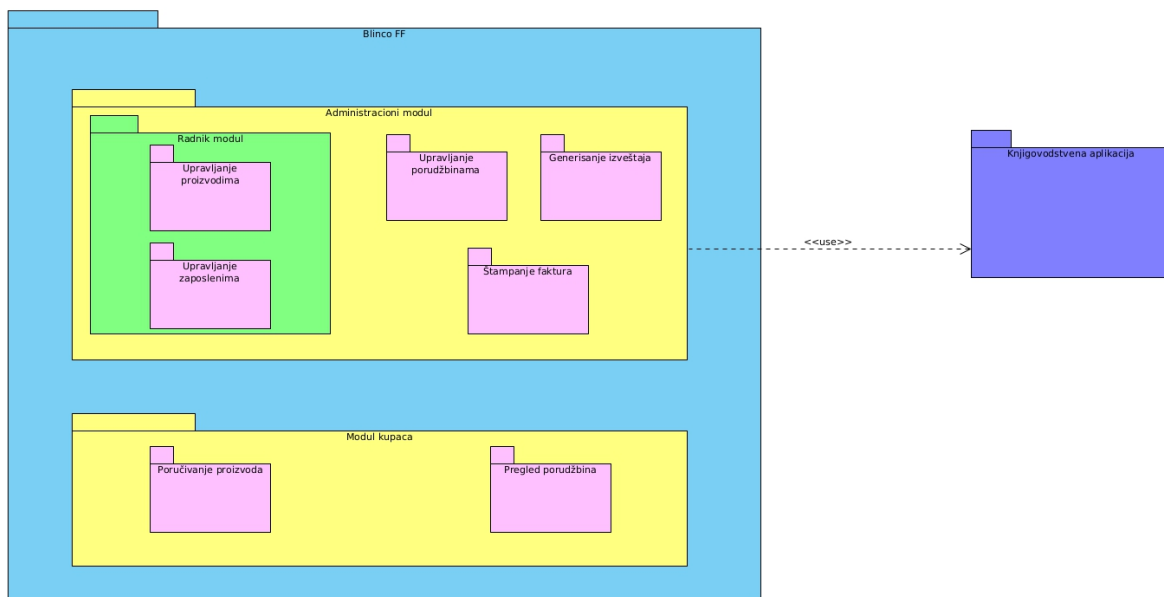
### 3. Arhitektura sistema

#### Stil razlaganja

Stilom razlaganja, sistem se deli na nezavisne module koji međusobno komuniciraju jedni sa drugima kako bi ispunili funkcionalne zahteve. Ovaj stil odgovara stilu softverskih modula.

Sistem je podeljen u dva modula: administracioni koji u sebi sadrži i radnik modul i modul kupaca.

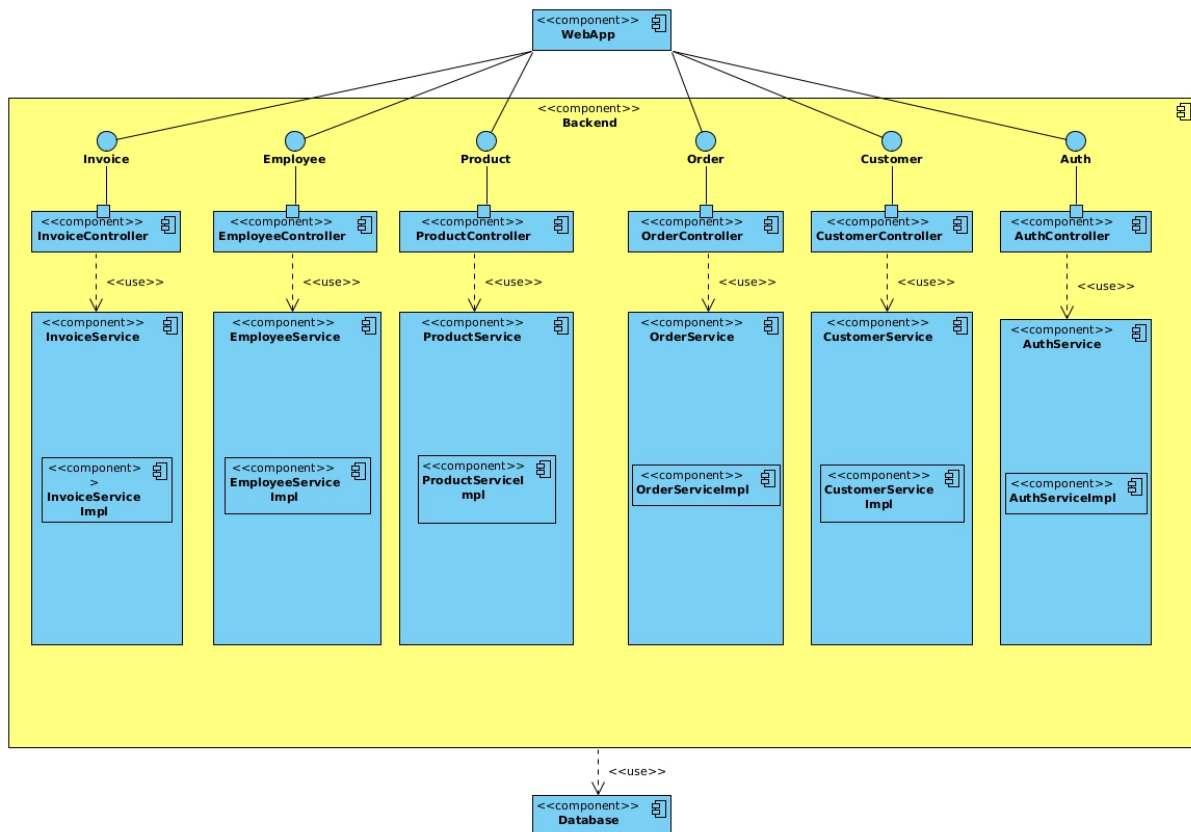
Pored modula na dijagramu se nalazi komponenta “knjigovodstvena aplikacija” koja predstavlja gotovu komponentu koju ne treba razvijati.



## Servisno-orijentisani stil

Servisno-orijentisani stil odgovara stilu povezivanja softverskih komponenata.

Ovim dijagramom je predstavljen servisno-orijentisani stil. Web aplikacija komunicira sa serverom preko odgovarajućih RESTful servisa. U zavisnosti od podataka koje zahteva korisnik, aplikacije će pozivati odgovarajući endpoint(servis). Server komunicira sa bazom podataka.



## Stil raspoređivanja

Ovim stilom predstavljena je povezanost elemenata softverskog okruženja i elemenata softverske arhitekture. Sistem će biti postavljen na web serveru. Komunicira sa serverom baze podataka. Server baze podataka komunicira sa backup bazom koja je postavljena na odvojenom serveru gde se vrši sam **backup** podataka. Pristup sistemu imaju web browseri preko **HTTP** ili **HTTPS** protokola. Ukoliko je pristup preko browsera na uređaju koji nije desktop, servisi koje daje knjigovodstvena aplikacija neće biti dostupni.

Stil raspoređivanja odgovara stilu alokacije.

