Dušan Stojiljković 5835 SE201 – PZ

Informacioni sistem za supermarket

Metropolitan Univerzitet 20.12.2023.

Sadržaj

[1. Agilni metod 5](#_Toc157456950)

[1. Vizija proizvoda 6](#_Toc157456951)

[1.1. Vizija proizvoda za zaposlene na kasi 6](#_Toc157456952)

[1.2. Vizija proizvoda za zaposlene u magacinu 6](#_Toc157456953)

[1.3. Vizija proizvoda za korisnika 6](#_Toc157456954)

[1.4. Vizija proizvoda za dostavljače 6](#_Toc157456955)

[2. Funkcionalni zahtevi 7](#_Toc157456956)

[2.1. Registracija korisnika 7](#_Toc157456957)

[2.2. Prijavljivanje korisnika 7](#_Toc157456958)

[2.3. Upravljanje ličnim informacijama 7](#_Toc157456959)

[2.4. Upravljanje proizvodima 7](#_Toc157456960)

[2.5. Korpa i plaćanje 7](#_Toc157456961)

[2.6. Narudžbine i isporuka 7](#_Toc157456962)

[3. Nefunkcionalni zahtevi 8](#_Toc157456963)

[3.1. Performanse 8](#_Toc157456964)

[3.2. Sigurnost 8](#_Toc157456965)

[3.3. Dostupnost 8](#_Toc157456966)

[3.4. Pouzdanost 8](#_Toc157456967)

[3.5. Održivost 8](#_Toc157456968)

[3.6. Integracija 8](#_Toc157456969)

[4. Korisničke price 9](#_Toc157456970)

[4.1. Korisničke price za zaposlene na kasi 9](#_Toc157456971)

[4.2. Korisničke priče za zaposlene u magacinu 9](#_Toc157456972)

[4.3. Korisničke priče kupca 10](#_Toc157456973)

[4.4. Korisničke priče dostavljača 10](#_Toc157456974)

[5. Persone 11](#_Toc157456975)

[5.1. Marko Marković – kasir 11](#_Toc157456976)

[5.2. Jovana Jovanović - kupac 11](#_Toc157456977)

[5.3. Stefan Nikolić – magacioner 11](#_Toc157456978)

[5.4. Igor Petrović – dostavljač 11](#_Toc157456979)

[6. Scenariji 12](#_Toc157456980)

[6.1. Korisnički scenariji za Marka - kasira 12](#_Toc157456981)

[6.1.1. Porudžbina proizvoda 12](#_Toc157456982)

[6.1.2. Brza i precizna obrada transakcije 12](#_Toc157456983)

[6.1.3. Korišćenje kupona 12](#_Toc157456984)

[6.2. Korisnički scenariji za Jovanu – kupca 12](#_Toc157456985)

[6.2.1. Brza i jednostavna navigacija proizvoda 12](#_Toc157456986)

[6.2.2. Kupovina od kuće 13](#_Toc157456987)

[6.2.3. Personalizovana ponuda na osnovu prethodnih kupovina 13](#_Toc157456988)

[6.3. Korisnički scenariji za Stefana – magacionera 13](#_Toc157456989)

[6.3.1. Upravljanje zalihama i narudžbinama 13](#_Toc157456990)

[6.3.2. Praćenje isteka roka trajanja 13](#_Toc157456991)

[6.4. Korisnički scenariji za Igora – dostavljača 13](#_Toc157456992)

[6.4.1. Optimizacija rute za brzu dostavu 13](#_Toc157456993)

[6.4.2. Rešavanje problema na terenu 13](#_Toc157456994)

[6.4.3. Feedback kupca 14](#_Toc157456995)

[7. Softverske arhitekture 15](#_Toc157456996)

[7.1. Monolitna arhitektura 15](#_Toc157456997)

[7.2. Mikroservisna arhitektura 16](#_Toc157456998)

[7.3. Serverless arhitektura 17](#_Toc157456999)

[8. Klase 18](#_Toc157457000)

[9. Baza podataka 19](#_Toc157457001)

[10. Komponente koje mogu biti ponovo upotrebljene 20](#_Toc157457002)

[11. Delovi sistema koji može biti računarstvo u oblaku 20](#_Toc157457003)

[12. Komponente sistema koje mogu biti mikroservisi 21](#_Toc157457004)

[13. Demonstracija Brute Force napada 22](#_Toc157457005)

[14. Atributi kvaliteta 23](#_Toc157457006)

[14.1. Performanse 23](#_Toc157457007)

[14.2. Bezbednost 23](#_Toc157457008)

[14.3. Pouzdanost 23](#_Toc157457009)

[14.4. Upotrebljivost 23](#_Toc157457010)

[14.5. Etika 23](#_Toc157457011)

[15. Koncepti pouzdanog programiranja 24](#_Toc157457012)

[15.1. Validacija input-a 24](#_Toc157457013)

[15.2. Šabloni (Pattern) 24](#_Toc157457014)

[15.3. Regularni izrazi (RegEx) 25](#_Toc157457015)

[16. Scenario testiranja 26](#_Toc157457016)

[16.1. Magacioner pretražuje proizvod 26](#_Toc157457017)

[16.2. Magacioner dodaje novi proizvod 27](#_Toc157457018)

[16.3. Korisnik završava kupovinu 28](#_Toc157457019)

[17. JUnit testiranje 29](#_Toc157457020)

[18. Trello 30](#_Toc157457021)

[19. Git 30](#_Toc157457022)

# Agilni metod

Agilni metod koji ću koristiti za izradu ovog projekta je **kanban**.

Kanban je vizuelni sistem za upravljanje projektima koji koristi tablu sa kolonama koje predstavljaju faze razvoja projekta kao što su „Backlog“, „In progress“ i „Done“. Svaki zadatak je predstavljen karticom koja počinje u Backlog-u i cilj je da se taj zadatak završi i ubaci u kolonu Done ili u Discontinued ako je razvojni tim odustao od implementacije tog zadatka.

Odabrao sam ovaj metod zato što kanban za razliku od drugih tehnologija nema fiksne iteracije I to može biti od pomoći jer sistem još uvek nije u potpunosti definisan tako da ću pomoću kanbana moći lakše da dodam nove zahteve za sistem, dodelim I promenim prioritete zahtevima. Kanban nudi vizuelni radni tok što bi moglo da mi pomogne da lakše pratim napredak projekta.

Na ovom linku možete da pratiti razvoj projekta na aplikaciji Trello:

https://trello.com/b/sfcgulF5/retailease

# Vizija proizvoda

## Vizija proizvoda za zaposlene na kasi

Za potrebe zaposlenih u supermarketu koji žele da efikasnije upravljaju svakodnevnim operacijama u supermarketu, kreiran je sistem RetailEase koji pruža real-time analitiku o prodaji, direktnu komunikaciju sa kupcima i intuitivni korisnički interfejs. Za razliku od tradicionalnih metoda, naš proizvod poboljšava iskustvo zaposlenih na kasi, sistem za upravljanje inventarom sinhronizuje podatke što zaposlenima omogućava da se više fokusiraju na pružanje kvalitetne usluge klijentima.

## Vizija proizvoda za zaposlene u magacinu

Za potrebe zaposlenih u magacinu koji žele da efikasnije upravljaju svakodnevnim operacijama u magacinu, kreiran je sistem RetailEase koji pruža zaposlenima napredne alate za praćenje, organizaciju i upravljanje inventarom i automatizaciju generisanja porudžbenica. Za razliku od tradicionalnih metoda, naš proizvod donosi visok nivo pouzdanosti praćenja inventara, čime se smanjuju greške i nedostaci.

## Vizija proizvoda za korisnika

Za potrebe korisnika koji žele poboljšati iskustvo kupovine kreiran je sistem RetailEase koji omogućava korisnicima jednostavno naručivanje proizvoda, personalizovane opcije isporuke, automatske prenosive liste i real-time praćenje porudžbine. Sistem je dizajniran sa fokusom na korisničko iskustvo nudeći intuitivni interfejs, jednostavnu navigaciju i brze odgovore. Za razliku od tradicionalnih metoda kupovina, naš proizvod donosi viši nivo komfora I efikasnije planiranje vremena.

## Vizija proizvoda za dostavljače

Za potrbe dostavljača koji žele ubrzati isporuku dostave kreiran je sistem RetailEase koji omogućava dostavljačima optimizaciju ruta za brže isporuke, real-time praćenje saobraćaja i automatsko ažuriranje rute. Za razliku od tradicionalnih metoda dostave, naš sistem nudi viši nivo efikasnosti, smanjenje troškova goriva, efikasno planiranje dostava I povećanje kapaciteta isporuka.

# Funkcionalni zahtevi

## Registracija korisnika

* Korisnici mogu kreirati nalog na sistemu unosom osnovnih informacija uključujući korisničko ime, lozinku, ime, prezime, email, ulicu, broj stana i broj telefona.
* Sistem treba validirati unete podatke tokom procesa registracije.

## Prijavljivanje korisnika

* Registrovani korisnici mogu se prijaviti na sistem koristeći svoje korisničko ime i lozinku.

## Upravljanje ličnim informacijama

* Korisnici mogu u svakom trenutku izmeniti svoje lične informacije.

## Upravljanje proizvodima

* Magacioneri mogu dodavati nove proizvode u sistem sa informacijama o šifri, nazivu, ceni, količini, tipu, opisu, roku trajanja i dobavljaču proizvoda.
* Magacioneri mogu menjati informacije nekog proizvoda ili brisati proizvode.
* Korisnici i magacioneri mogu pretraživati proizvode na osnovu različitih kriterijuma kao što su naziv, kategorija, tip ili cena.

## Korpa i plaćanje

* Korisnici mogu dodavati proizvode u korpu, pregledati sadržaj i završiti proces kupovine.
* Sistem treba obezbediti siguran proces plaćanja putem različitih metoda (kreditna kartica, plaćanje pouzećem).

## Narudžbine i isporuka

* Korisnici mogu praviti narudžbine proizvoda i pratiti status isporuke.
* Dostavljači mogu potvrditi i procesuirati narudžbine, ažurirati status isporuke i označiti ih kao isporučene.

# Nefunkcionalni zahtevi

## Performanse

* Sistem treba da omogući brz pristup podacima o proizvodima, sa vremenom odgovora ispod 2 sekunde
* Vreme trajanja procesa registracije novih korisnika ne sme preći 5 sekundi.

## Sigurnost

* Lozinke korisnika trebaju da se čuvaju u singurnom format, korišćenjem hash algoritma SHA256.

## Dostupnost

* Sistem treba da bude dostupan 99.9% vremena , uz mogućnost redovnog održavanja od najmanje 2 sata nedeljno.

## Pouzdanost

* Sigurnosne kopije baze podataka treba da se prave automatski svakodnevno i čuvaju na odvojenom server.

## Održivost

* Kod sistema treba biti dobro dokumentovan kako bi se olakšalo održavanje i buduće proširenje koda.

## Integracija

* Sistem treba da omogući integraciju sa drugim sistemima kao što su sistem za plaćanje ili sisltem za praćenje isporuka.

# Korisničke price

## Korisničke price za zaposlene na kasi

* Kao kasir, želim da se prijavim na sistem sa svojim korisničkim imenom i lozinkom, kako bih započeo radnu smenu.
* Kao kasir, želim mogućnost dodavanja proizvoda ručno ukoliko ne mogu skenirati bar-kod ili ukoliko proizvod nema bar-kod.
* Kao kasir, želim da pregledam cenu svakog proizvoda pre nego što ga dodam na račun, kako bih izbegao greške u fakturisanju.
* Kao kasir, želim da imam mogućnost da prihvatim različite vrste plaćanja (gotovina, kartica) i izdam potvrdu o transakciji.
* Kao kasir, želim mogućnost brisanja proizvoda iz korpe ukoliko se korisnik predomisli pre nego što završi kupovinu.
* Kao kasir, želim da sistem automatski oduzme kupljeni proizvod iz inventara kako bi se održala tačnost zaliha.

## Korisničke priče za zaposlene u magacinu

* Kao magacioner, želim da mogu prijaviti dolazak novih proizvoda u magacin, unosom informacija kao što su količina, rok trajanja i šifra proizvoda.
* Kao magacioner, želim mogućnost pretrage proizvoda u magacinu po različitim kriterijumima, kao što su šifra, naziv ili količina.
* Kao magacioner, želim mogućnost označavanja proizvoda koji su istekli ili su blizu isteka roka trajanja, kako bih ih lako identifikovao prilikom pretrage.
* Kao magacioner, želim mogućnost ažuriranja informacija o proizvodima, kao što su cena, količina i kategorija, kako bih održavao tačnost podataka u sistemu.
* Kao magacioner, želim da sistem automatski smanjuje količinu proizvoda u magacinu prilikom svake prodaje, kako bih održavao tačnost stanja zaliha.

## Korisničke priče kupca

* Kao kupac, želim da mogu pretraživati proizvode po kategorijama, brendovima ili drugim filterima, kako bih lako pronašao željeni proizvod.
* Kao kupac, želim da mogu dodati proizvode u korpu i nastaviti s kupovinom, kako bih mogao kupiti više proizvoda u jednoj sesiji.
* Kao kupac, želim mogućnost pregleda i uređivanja sadržaja svoje korpe pre nego što završim kupovinu.
* Kao kupac, želim pratiti status isporuke moje narudžbine i dobiti obaveštenje o vremenu isporuke.
* Kao kupac, želim da budem obavešten o posebnim ponudama, rasprodajama ili novim proizvodima putem obaveštenja na platformi.

## Korisničke priče dostavljača

* Kao dostavljač, želim pregledati listu narudžbina koje treba isporučiti, uključujući informacije o adresi isporuke i vremenu naručivanja.
* Kao dostavljač, želim imati navigaciju do adrese isporuke putem integrirane mape, kako bih pronašao najbrži put do odredišta.
* Kao dostavljač, želim moći kontaktirati korisnika (kupca) radi dodatnih informacija ili obaveštenja o isporuci.
* Kao dostavljač, želim imati mogućnost označavanja narudžbine kao isporučenu nakon što je uspešno predam korisniku.
* Kao dostavljač, želim biti obavešten o posebnim uputstvima ili zahtevima korisnika (npr. ostaviti paket na vratima).
* Kao dostavljač, želim imati evidenciju svojih prethodnih isporuka i pregled napojnica ili ocena koje sam dobio od korisnika.
* Kao dostavljač, želim biti obavešten o promenama u vremenu isporuke ili eventualnim problemima tokom isporuke.

# Persone

## Marko Marković – kasir

Marko ima 23 godine i on je iskusan prodavac sa petogodišnjim iskustvom u maloprodaji. Marko je završio srednju Ekonomsku školu u Nišu i dobro poznaje proizvode u svojoj prodavnici.

Marko je ključna karika u pružanju kvalitetne usluge kupcima i voleo bi da koristi novi softverski sistem kako bi poboljšao svoje prodajne veštine i organizaciju rada u prodavnici. Brz pristup proizvodima u prodavnici bi mu pomogao da kupcima preporuči proizvode koji bi im se dopali. Dodatno, sistem bi mogao da mu pomogne u održavanju zaliha i organizaciji proizvoda na policama.

## Jovana Jovanović - kupac

Jovana ima 19 godina i redovno posećuje supermarket. Jovana je student Građevinskog Fakulteta u Nišu.

Jovana ceni kvalitet proizvoda, pristojne cene i brzu uslugu. Sistem bi mogao da joj poboljša iskustvo brzom navigacijom kroz proizvode ili kupone i pružanju tačnih informacija o cenama i karakteristikama. Takođe, sistem bi mogao da joj omogući kupovinu od kuće i da joj šalje obaveštenja kada su njeni omiljeni proizvodi na popustu. Mogućnost kontaktiranja prodavca za preporuke može dodatno povećati zadovoljstvo Jovane tokom kupovine.

## Stefan Nikolić – magacioner

Stefan ima 25 godina i radnik je u magacinu. Ima višegodišnje iskustvo u organizaciji proizvoda i upravljanju inventarom i završio je kurs logistike.

Stefanu bi novi informacioni sistem bio ključan alat za poboljšanje organizacije i efikasnosti rada u magacinu. Automatizacija informacija o zalihama će mu pomoći da efikasnije upravlja obnavljanjem proizvoda a briz pristup podacima o lokaciji proizvoda olakšati process pripreme narudžbina.

## Igor Petrović – dostavljač

Igor ima 20 godina i radi kao dostavljač. Igor je student na Mašinskom fakultetu u Nišu a u slobodno vreme obavlja dostave proizvoda korisnicima supermarketa.

Svakodnevno se suočava sa izazovima brze i efikasne dostave proizvoda. Kao neko ko je već nekoliko godina u ovoj branši, Igor vidi vrednost pouzdanog i optimizovanog sistema dostave kako bi unapredio kako svoje, tako i iskustvo kupaca. Sistem bi mu omogućio da lakše upravlja narudžbinama i da brže i efikasnije pronalazi puteve do kupaca, čime doprinosi boljem korisničkom iskustvu.

# Scenariji

## Korisnički scenariji za Marka - kasira

### Porudžbina proizvoda

Marko je na početku radnog vremena I primetio je da je jedan od proizvoda rasprodat I da će biti potrebno obnoviti zalihe. Marko koristi novi sistem kako bi proverio stanje zaliha I napravio porudžbinu za proizvode čije su zalihe pri kraju. Sistem mu omogućava da brzo I lako pronađe sve proizvode koje bi trebao da poruči I generiše narudžbenicu koju šalje dobavljaču.

### Brza i precizna obrada transakcije

Marko je na kasi kada kupac donosi korpu sa proizvodima. Marko koristi novi softverski sistem da brzo skenira bar-kodove i automatski izračuna ukupan iznos. Kupac plaća karticom, a Marko završava transakciju efikasno, pružajući brzu uslugu.

### Korišćenje kupona

Kupac dolazi na kasu sa korpom. Marko koristi novi softverski sistem da brzo skenira bar-kodove. Kupac želi da iskoristi kupone za popust. Marko unosi identifikacioni broj kupca u sistem čime se primenjuju kuponi za popust i saopštava kupcu ukupan iznos uštede koji su ostvarili korišćenjem kupona. Kupac je zadovoljan dobijenim popustom a Marko završava transakciju.

## Korisnički scenariji za Jovanu – kupca

### Brza i jednostavna navigacija proizvoda

Jovana želi brzo pronaći sveže voće. Koristi aplikaciju supermarketa koja joj omogućava da pretraži proizvode u supermarket i pronađe tačnu lokaciju voća. Brza i efikasna navigacija joj pomaže da obavi kupovinu.

### Kupovina od kuće

Jovana uči za ispit i nema vremena da poseti supermarket. Koristi aplikaciju kako bi odabrala željene proizvode i napravi porudžbinu. Sistem joj omogućava da lako pregleda proizvode, doda ih u korpu i prati aktuelne popuste. Nakon završenog odabira, Jovana bira opciju dostave na kućnu adresu i završava porudžbinu.

### Personalizovana ponuda na osnovu prethodnih kupovina

Jovana dobija obaveštenje od sistema o trenutnom popustu na njene omiljene grickalice. Sistem koristi informacije o njenim prethodnim kupovinama kako bi napravio personalizovane ponude. Jovana zadovoljno koristi popuste na proizvode koje voli.

## Korisnički scenariji za Stefana – magacionera

### Upravljanje zalihama i narudžbinama

Stefan je primetio da su određeni proizvodi pri kraju zaliha. Stefan koristi novi informacioni sistem da proveri trenutno stanje količine i automatski generiše narudžbenicu za obnavljanje zaliha. Sistem mu odgovara potvrdom o uspešnoj narudžbini.

### Praćenje isteka roka trajanja

Stefan je primetio da se neki proizvodi približavaju isteku roka trajanja. Sostem ga automatski obaveštava o proizvodima koji su blizu isteka roka trajanja, omogućavajući mu da pripremi porudžbinu za te proizvode ako oni nisu dostupni u zalihama.

## Korisnički scenariji za Igora – dostavljača

### Optimizacija rute za brzu dostavu

Igor preuzima nekoliko narudžbina za dostavu. Sistem automatski optimizuje rutu, uzimajući u obzir trenutni saobraćaj i kreira najbržu rutu za dostavu. Igor koristi sistem za praćenje optimalne rute, čime značajno smanjuje vreme putovanja.

### Rešavanje problema na terenu

Igor stiže na adresu dostave, ali kupac nije dostupan. Igor koristi sistem za obaveštavanje kupca o dostavi i čeka dodatne instrukcije. Ako kupac nije dostupan, koristi opciju za vraćanje proizvoda u supermarket.

### Feedback kupca

Nakon završene dostave, kupac daje ocenu i komentar dostavljaču. Igor prima povratne informacije o o svojoj usluzi putem sistema. Sistem beleži ocene i komentare pomažući Igoru da poboljša svoj rad i pruži bolje iskustvo kupovine kupcima.

# Softverske arhitekture

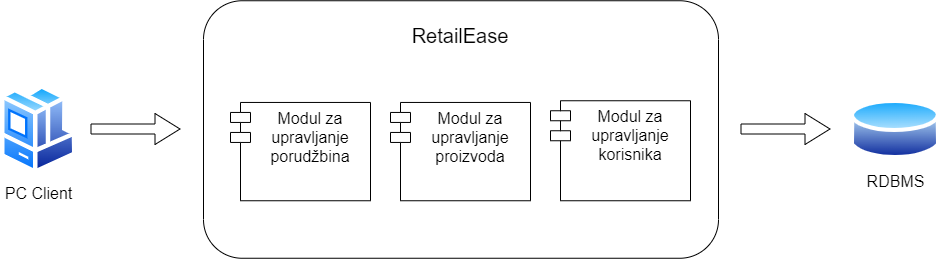
## Monolitna arhitektura

Monolitna arhitektura predstavlja softverski dizajn koji se odnosi na pristup u kojem se cela aplikacija razvija kao jedan integrisani sistem, gde su sve komponente i moduli smeštene u istu aplikaciju i izvršavaju se unutar istog procesa.

Prednosti monolitne aplikacije su te što je jednostavna za razvoj, testiranje i održavanje i lake su za upravljanje.

Mane monolitne aplikacije su te što nije skalabilna, praćenje performansi i optimizacija je jako teška i može doći do domino efekta kada se javi greška u jednom delu sistema.

U moja aplikacija jeste monolitna jer je ceo sistem centralizovan i obuhvata sve funkcionalnosti, uključujući kasirski sistem, magacinski sistem, sistem za korisnike i sistem za upravljanje dostavom.

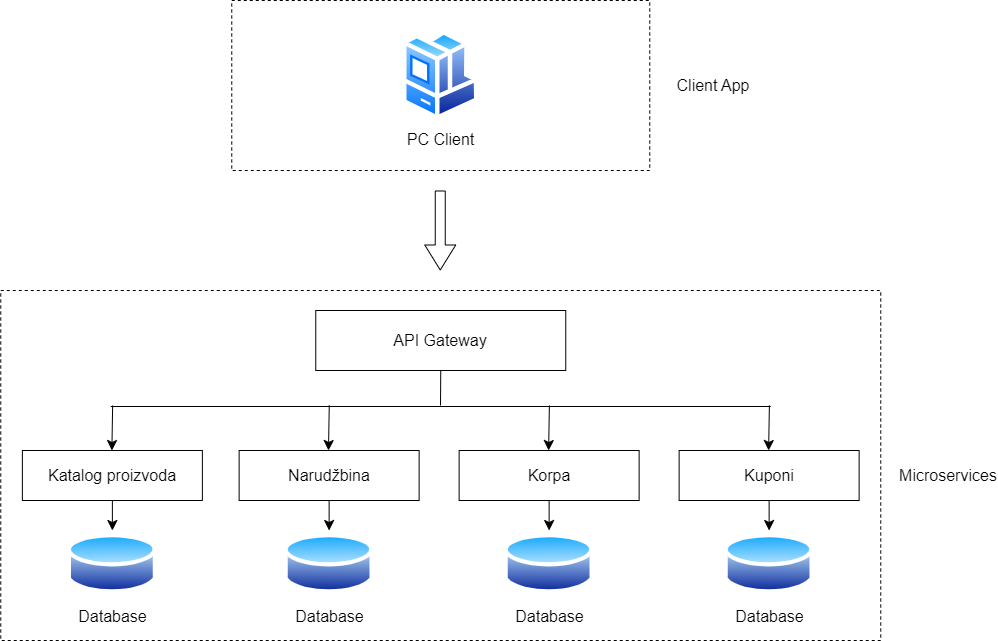


## Mikroservisna arhitektura

Mikroservisna arhitektura predstavlja softverski dizajn u kojem se aplikacija razvija kao skup manjih, nezavisnih servisa. Svaki mikroservis obavlja specifičan posao i može se razvijati i testirati nezavisno od drugih mikroservisa u sistemu.

Prednosti mikroservisne arhitekture su lakše skaliranje, nezavisno upravljanje servisima i povećana otpornost od greške u jednom delu sistema.

Mane mikroservisne arhitekture može biti velika složenost u upravljanju međusobnom komunikacijom između mikroservisa, zahtevaju dodatnu infrastrukturu za upravljanje i kontrolu i održavanje doslednosti podataka.

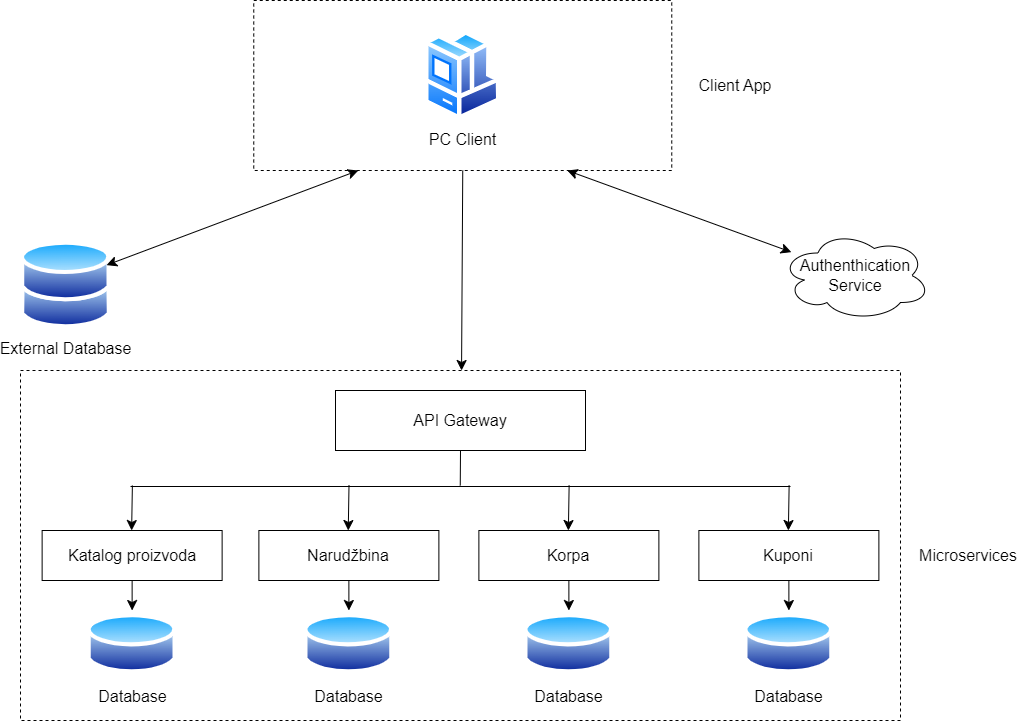


## Serverless arhitektura

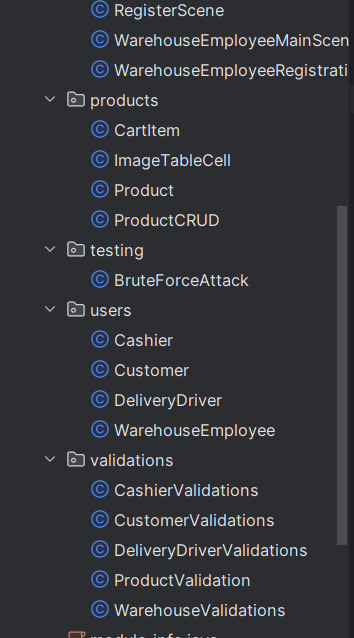
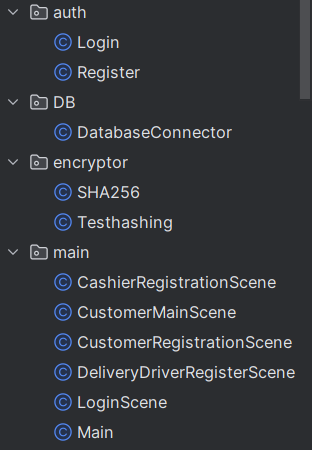
Serverless arhitektura ili FaaS (Function-as-a-Service) je model razvoja softvera u kojem se programeri fokusiraju na kreiranje funkcija umesto na upravljanje infrastrukturom.

Prednosti ove arhitekture su te što može da se skalira automatski zavisno od opterećenja, smanjuje potrebu za upravljanje infrastrukturom i plaća se samo za stvarno korišćenje resursa.

Mane ove arhitekture su te što zbog prirode serverless arhitekture imamo manje kontrole nad infrastrukturom i performansama,



# Klase



# Baza podataka



# Komponente koje mogu biti ponovo upotrebljene

* Klasa “DatabaseConnector” koja se koristi za povezivanje sa bazom podataka i sadrži CRUD metode za sve persone.
* Paket “validations” koji sadrži validacije za sve ulaze. ovaj paket uključuje validaciju za klase: Customer, Cashier, WarehouseEmployee, DeliveryDriver i Product.
* Klasa SHA256 koja se koristi za šifriranje lozinke.
* Možemo izdvojiti korpu u koju se dodaju proizvodi jer uz malu izmenu koda može da se koristi i u drugim projektima.
* Sistem za rad sa narudžbinama.

# Delovi sistema koji može biti računarstvo u oblaku

Komponente koje mogu biti računarstvo u oblaku i tipovi:

* Cela aplikacija može biti SaaS ako se prevede u Web aplikaciju koja će biti dostupna putem Web pretraživača.
* Baza podataka u oblaku može biti IaaS ili PaaS zavisno o tome koliku kontrolu možemo imati nad infrastrukturom.

Neke komponente koje mogu biti zamenjene već postojećim rešenjima su autentifikacija jer umesto samostalne implementacije te komponente, možemo koristiti Firebase Authentication. Takođe možemo koristiti Zendesk kao sistem za praćenje korisnika i analizu performansi podrške.

Komponente za koje računarstvo u oblaku nije najbolje rešenje može biti sistem za upravljanje transakcijama jer je lokalni server sa striktnom kontrolom pristupa bolja opcija od cloud-a.

# Komponente sistema koje mogu biti mikroservisi

Mikroservisi su samostalne komponente sistema sa spostvenom bazom podataka koje se mogu razvijati, implementirati i održavati nezavisno od ostatka sistema.

Neke komponente iz mog sistema koje mogu biti mikroservisi su:

* Mikroservis za upravljanje proizvodima, skladištem i praćenjem inventara.
* Mikroservis za upravljanje informacijama o dostavljačima, praćenje dostave i rute dostave.
* Mikroservis za upravljanje korisničkim nalozima, njihovom registracijom i autentifikacijom.
* Mikroservis za obradu narudžbina, praćenje statusa i potvrdu plaćanja.
* Mikroservis za prijem i upravljanje korisničkim zahtevima, podršku i rešavanje problema.

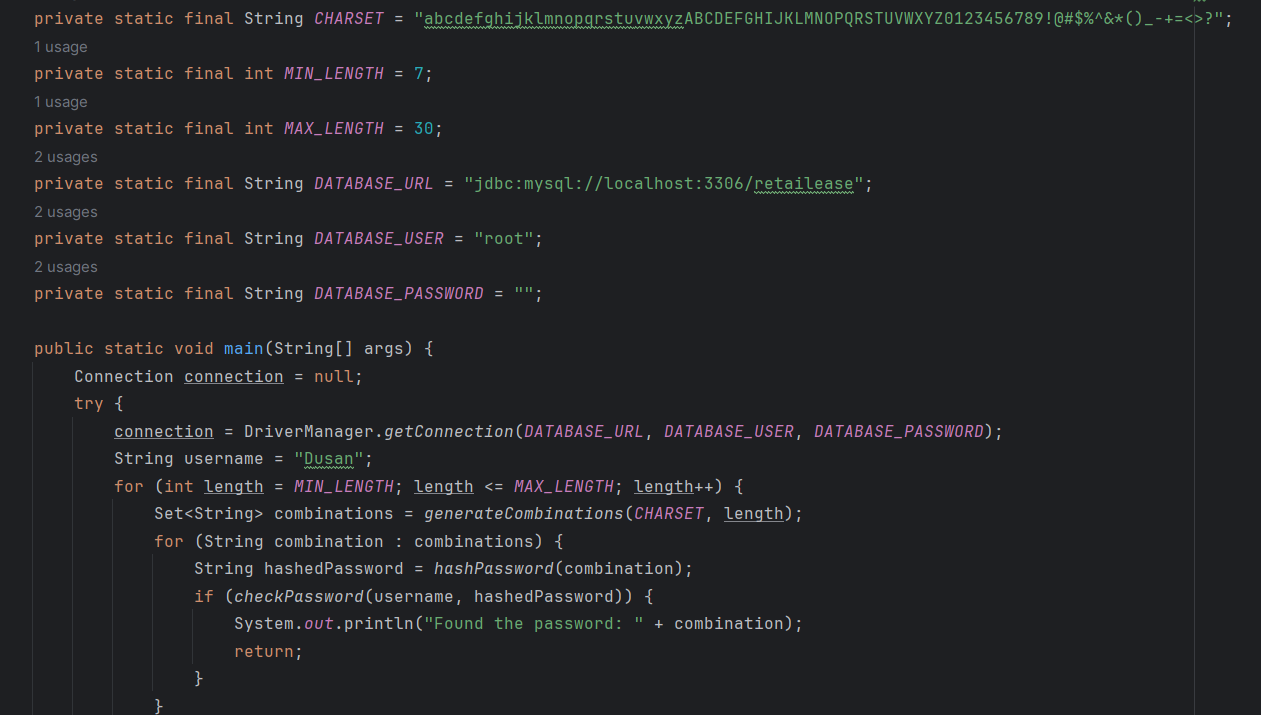
# Demonstracija Brute Force napada

Brute force napad je vrsta napada u kojoj napadač pokušava da pronađe pravu vrednost određenog podatka kao sto je lozinka ili kod. Napadač uglavnom koristi neki alat koji će moći da generiše kombinacije karaktera dok ne pronađe pravu vrednost podatka.

Kako bismo se odbranili od brute force napada, trebamo koristiti jake i složene lozinke, ograničiti broj pokušaja unošenja lozinke i implementirati dodatne sigurnosne mere kao što je dvofaktorska autentifikacija.

U mom projektu, primer brute force napada se nalazi u paketu „testing”. Lozinke korisnika su šifrovane algoritmom SHA256 i na lozinke je dodat salt kako bismo izbegli slučaj da dva različita korisnika sa istom šifrom imaju istu hash vrednost lozinke.

Probao sam da napadaču maksimalno olakšam pokušaj napada tako što sam mu omogućio da vidi kako se lozinke validiraju, takođe sam mu dao validnog korisnika kao i salt vrednost koju koristi njegova lozinka ali nisam uspeo da probijem šifru i redovno dobijam OutOfMemoryError.



# Atributi kvaliteta

## Performanse

* Vreme od kada korisnik inicira zahtev do trenutka kada sistem pruža odgovor je jako kratko, kao i vreme koje je potrebno da se učitaju stranice ili resursi

## Bezbednost

* Lozinke svih korisnika su šifrirane SHA256 algoritmom uz dodavanje salt-a.
* Sistem ima mogućnost da proverava identitet korisnika i pruža odgovarajuće dozvole

## Pouzdanost

* Sistem je sposoban da radi bez neočekivaanih prekida ili padova.

## Upotrebljivost

* Korisnički interfejs je intuitivan i lak za korišćenje.

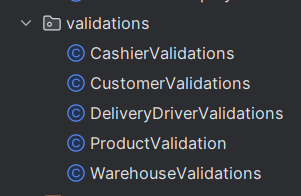
## Etika

* Sistem se pridržava važećim zakonskim propisima i poštuje etičke principe u vezi korišćenja podataka i interakcija sa korisnikom.

# Koncepti pouzdanog programiranja

## Validacija input-a

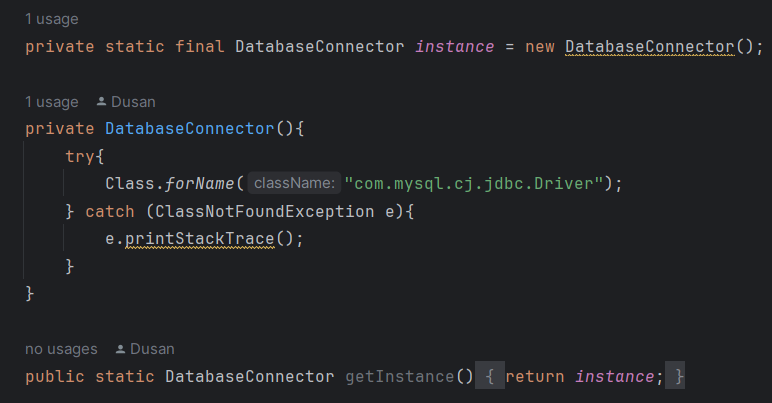
Validacija inputa se nalazi u paketu validations. Sve informacije koje neko može da unese kao input su validirane.



## Šabloni (Pattern)

Singleton pattern je oblik projektovanja softvera koji se koristi kako bi se obezbedilo da klasa može da ima samo jednu instancu. Ovaj obrazac se koristi kada imamo klase koje koordinišu rad sistema ili klase koje pružaju pristup resursima kao što su baze podataka.

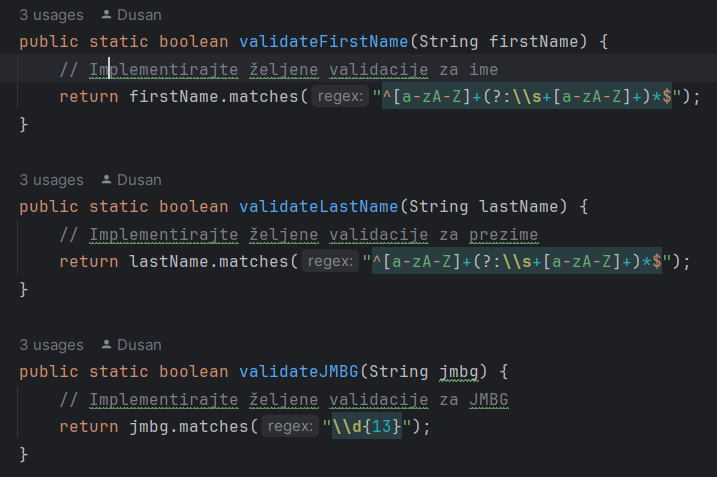
U projektu sam iskoristio singleton pattern u klasi DatabaseConnector kako bih napravio samo jednu instancu te klase.



## Regularni izrazi (RegEx)

Regex je alat koji se koristi za pretragu i manipulaciju tekstualnih podataka. To je skup pravila koja definišu šablone za prepoznavanje zadatih znakova u tekstu. Sastoji se od različitih znakova i specijalnih karaktera koji se koriste za definisanje šablona

U projektu sam koristio regularne izraze u kombinaciji sa validacijom na input-u kako bih proverio da li input korisnika zadovoljava zadata pravila i samim tim obezbedio čuvanje podataka koji su uspešno validirani.



# Scenario testiranja

## Magacioner pretražuje proizvod

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naslov | Pretraga proizvoda | Rev 01 | Autor | Dušan Stojiljković | Datum | | 29.01.2024. |
| Cilj | Podaci o pretraženom proizvodu | | Reference | / | | | |
| Test uslovi | / | Vreme neophodno za izradu test slučaja | 5 min | Neophodno vreme za izvršenje test slučaja | | 3 min | |

|  |  |
| --- | --- |
|  | Opis postavke za testiranje |
| \* | Magacioner ulogovan u aplikaciju |
| \* | Magacioner bira opciju da pretrazi proizvod |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Definicija testa | | | | Izvršenje testa | |
|  | Uslovi | Ulazni podaci | Očekivani rezultat | Aktuelni rezultat | Broj problema |
| \* | Magacioner je uneo sedam karaktera za šifru koji postoji u bazi | Šifra proizvoda | Prikazuju se svi proizvodi koji sadrže deo unešene šifre | Prikazuju se svi proizvodi koji sadrže deo unešene šifre | 0 |
| \* | Magacioner je uneo petnaest karaktera za šifru koja postoji u bazi | Šifra proizvoda | Uspešno se prikazuje traženi proizvod | Uspešno se prikazuje traženi proizvod | 0 |
| \* | Magacioner je uneo petnaest karaktera za šifru koja ne postoji u bazi | Šifra proizvoda | Pretraga nije moguća, šifra ne postoji | Pretraga nije moguća | 0 |

|  |  |
| --- | --- |
|  | Opis post uslova |
|  | Magacioner je uspešno pretražio proizvod na osnovu šifre |

## Magacioner dodaje novi proizvod

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naslov | Dodavanje proizvoda | Rev 01 | Autor | Dušan Stojiljković | Datum | | 29.01.2024. |
| Cilj | Uspešno dodat proizvod | | Reference | / | | | |
| Test uslovi | / | Vreme neophodno za izradu test slučaja | 5 min | Neophodno vreme za izvršenje test slučaja | | 3 min | |

|  |  |
| --- | --- |
|  | Opis postavke za testiranje |
| \* | Magacioner ulogovan u aplikaciju |
| \* | Magacioner bira opciju da doda novi proizvod |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Definicija testa | | | | Izvršenje testa | |
|  | Uslovi | Ulazni podaci | Očekivani rezultat | Aktuelni rezultat | Broj problema |
| \* | Magacioner je uneo sve validne informacije za dodavanje novog proizvoda | Šifra proizvoda, naziv, cena, količina, tip, opis, rok trajanja, dostavljač, slika | Proizvod uspešno dodat u bazu i tabelu sa proizvodima | Proizvod uspešno dodat u bazu i tabelu sa proizvodima | 0 |
| \* | Magacioner je uneo sve informacije osim cene proizvoda | Šifra proizvoda, naziv, količina, tip, opis, rok trajanja, dostavljač, slika | Proizvod nije dodat u bazu, polje cena je prazno | Proizvod nije dodat u bazu | 0 |
| \* | Magacioner je uneo sve informacije ali šifra nema 15 karaktera | Šifra proizvoda, naziv, cena, količina, tip, opis, rok trajanja, dostavljač, slika | Proizvod nije dodat u bazu, šifra nije prošla validaciju | Proizvod nije dodat u bazu | 0 |

|  |  |
| --- | --- |
|  | Opis post uslova |
|  | Magacioner je uspešno dodao proizvod u bazu podataka |

## Korisnik završava kupovinu

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naslov | Kupovina proizvoda | Rev 01 | Autor | Dušan Stojiljković | Datum | | 29.01.2024. |
| Cilj | Uspešno obavljena kupovina | | Reference | / | | | |
| Test uslovi | / | Vreme neophodno za izradu test slučaja | 5 min | Neophodno vreme za izvršenje test slučaja | | 3 min | |

|  |  |
| --- | --- |
|  | Opis postavke za testiranje |
| \* | Korisnik ulogovan u aplikaciju |
| \* | Korisnik dodaje proizvode u korpu |
| \* | Korisnik završava kupovinu |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Definicija testa | | | | Izvršenje testa | |
|  | Uslovi | Ulazni podaci | Očekivani rezultat | Aktuelni rezultat | Broj problema |
| \* | Korisnik je ubacio sve proizvoje u korpu i završio kupovinu | Proizvod, količina proizvoda | Kupovina uspešno završena | Kupovina uspešno završena | 0 |
| \* | Korisnik je završio kupovinu a nije dodao ni jedan proizvod u korpu | / | Kupovina neuspešno završena, korpa je prazna | Kupovina neuspešno završena | 0 |
| \* | Korisnik je ubacio proizvod u korpu a količinu podesio na 0 i završio kupovinu | Proizvod, količina proizvoda | Kupovina neuspešno završena, količina ne može biti 0 | Kupovina neuspešno završena | 0 |

|  |  |
| --- | --- |
|  | Opis post uslova |
|  | Korisnik je uspešno završio kupovinu |

# JUnit testiranje

JUnit je framework u Javi koji se koristi za pisanje i izvršavanje automatizovanih jediničnih testova u Java aplikacijama. JUnit testovima možemo osigurati da svaki deo koda radi ispravno i daje očekivane rezultate.

JUnit testovi se sastoje od **anotacije** koje označavaju metode koje će biti izvršene kao testovi , postavljanje pretpostavki ili čišćenje resursa nakon završetka testa.

**Asertacije** su izrazi koji se koriste u testovima kako bi se proverilo da li su određeni uslovi ili očekivanja ispunjeni. Test se smatra uspešnim ako je asertacija ispunjena tj. neuspešnim ako asertacija nije ispunjena.

Svi testovi moje aplikacije se nalaze u paketu testing.

# Trello

Trello je online alat za upravljanje projektima koji koristi koncept table sa karticama. Ova platforma je namenjena pojedincima ili timovima da bolje organizuju svoje zadatke i projekte i da sve to prikažu na vizueal način.

Trello se sastoji iz table koja predstavlja radni prostor i na kojoj se nalaze kartice.

Kartice predstavljaju pojedinačne zadatke koji su detaljno opisani i koji sadrže dodatne informacije kao što je grupa kome je dodeljen taj zadatak, rok dostave, dodatni komentari i sl.

Sve kartice se nalaze u listama koje predstavljaju različite faze u razvoju projekta.

Kartice se pomeraju iz jedne liste u drugu u zavisnosti od napretka razvoja svakog zadatka. Svi zadaci počinju u listi “Backlog” i trebaju da se završe u “Done” listi.

Razvoj projekta možete pratiti na linku:

https://trello.com/b/sfcgulF5/retailease

# Git

Git je alat otvorenog koda koji se koristi za upravljanje izvornim kodom i praćenje promena tokom vremena. Svi korisnici imaju kompletnu kopiju repozitorijuma a razvoj se odvija kreiranjem, spajanjem i upravljanjem grana pre nego što se spoji u glavnu granu (master).

Projekat se nalazi na GitHub-u. Imao sam problema sa commit porukama jer iz nekog razloga nisam mogao da promenim njihov sadržaj.

Link do repozitorijuma:

https://github.com/work-dusan/RetailEase.git