



Univerzitet u Sarajevu
Elektrotehnički fakultet u Sarajevu
Odsjek za računarstvo i informatiku



Bazaar Marketplace Software Requirements Specification

Softverski inženjering

Grupe: 8, 9, 10
Profesor: Novica Nosović
Asistent: Hrnjić Tarik
Asistent: Halilović Kenan
Datum: 08.06.2025

Sadržaj

1	Uvod	2
1.1	Svrha	2
1.2	Konvencija dokumenta	2
1.3	Kome je namijenjeno	2
1.4	Obim projekta	2
1.5	Reference	3
2	Opis sistema	4
2.1	Perspektiva proizvoda	4
2.2	Entity Relationship Diagram	6
2.3	Korisničke klase i njihove karakteristike uz Activity i Use Case dijagrame	16
2.3.1	Activity dijagrami	19
2.4	Operativno okruženje	23
2.5	Ograničenja dizajna i implementacije	23
3	Karakteristike sistema	24
3.1	Opis i prioritet	24
3.2	Funkcionalni zahtjevi	24
3.3	Dijagram komponenti	27
4	Vanjski zahtjevi interfejsa	28
4.1	Korisnički interfejs	28
4.2	Hardverski interfejs	29
4.3	Softverski interfejs	29
4.4	Komunikacijski interfejs	30
4.5	Dijagram raspoređivanja	30
5	Nefunkcionalni zahtjevi	31
5.1	Upotrebljivost sistema	31
5.2	Sigurnost sistema	31
5.3	Dostupnost sistema	32
5.4	Održavanje sistema	32
5.5	Skalabilnost sistema	32
5.6	Performanse sistema	33
6	FURPS+	34
6.1	Funkcionalnosti	34
6.2	Upotrebljivost	36
6.3	Pouzdanost	37
6.4	Performanse	37
6.5	Podržanost	38
6.6	Implementacijski zahtjevi	38
6.7	Ograničenja Interfejsa	39
6.8	Fizički zahtjevi	39
6.8.1	Serverska Infrastruktura	40
6.8.2	Klijentske Aplikacije	40
6.8.3	Mrežna Povezanost	41
6.9	Ograničenja dizajna	41

1 Uvod

Ovaj dokument predstavlja specifikaciju softverskih zahtjeva (Software Requirement Specification - SRS) "Bazaar" Marketplace sistema. Za opis sistema biće korišten FURPS model.

1.1 Svrha

Svrha ovog dokumenta je da predstavi detaljan opis "Bazaar" Marketplace sistema. Objasniće svrhu i karakteristike sistema, interfejse sistema, šta sistem radi, koje su njegove funkcionalnosti, kao i na koji način će se upravljati sistemom. Ovaj dokument je referenca onima koji će koristiti sistem ali i onima koji su ga razvili i/ili će ga u budućnosti dodatno razvijati.

1.2 Konvencija dokumenta

Termin	Objašnjenje
FURPS	Akronim za model klasifikacije atributa kvaliteta softvera (funkcionalnih i nefunkcionalnih zahtjeva) - Functionality, Usability, Reliability, Performance i Supportability
UI	User interface – korisnički interfejs
UX	User experience – korisničko iskustvo
JWT	JSON web token - koristi se za autentikaciju i autorizaciju na svim platformama

1.3 Kome je namijenjeno

Sistem je specijalno kreiran za preduzeća i pojedince koji žele otkriti, usporediti i kupiti proizvode jednostavno i sigurno, kao i za prodavače koji imaju za cilj plasman svojih proizvoda većem broju kupaca.

1.4 Obim projekta

"Bazaar" Marketplace sistem je dizajniran kako bi omogućio brz i siguran način za prodaju, kupovinu i plasman različitih proizvoda. Sistem također posjeduje i administratorsku web stranicu namijenjenu za upravljanje korisnicima, narudžbama, tehničkom podrškom, cijenama proizvoda i reklamama. Prodavači i kupci imaju pristup namjenskim mobilnim aplikacijama (za iOS i Android) kako bi se osiguralo nesmetano i besprijekorno iskustvo za obje strane tržišta.

Aplikacija za kupce, Bazaar, bit će postavljena na engleski jezik, s opcijom lokalizacije jezika na bosanski, španski i njemački. Sve cijene bit će prikazane u lokalnoj valuti (KM). Kako bi se osigurala skalabilnost i proširenje, cijena će biti sistemska postavka. Ova aplikacija kupcima pruža jednostavan proces registracije i prijave, intuitivnu pretragu proizvoda grupiranih po kategorijama tipova, profile prodavača i korisničke recenzije vezane za njihove profile, jednostavno naručivanje uz organiziranu dostavu ili direktno preko prodavača, kao i tehničku podršku te mogućnost kontaktiranja prodavača.

Prodavači će koristiti zasebnu aplikaciju pod nazivom The Shop, koja će inicijalno biti na bosanskom jeziku. Ova aplikacija će imati jednostavan i intuitivan interfejs osmišljen da pomogne prodavačima da efikasno upravljaju svojim narudžbama. Osim toga, omogućit će prodavačima da lako ažuriraju cijene, količine i dostupnost proizvoda.

Administratori će upravljati platformom putem administratorskog web portala.

1.5 Reference

- <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-8.0>
- <https://learn.microsoft.com/en-us/ef/core/>
- <https://www.postgresql.org/docs/>
- <https://learn.microsoft.com/en-us/aspnet/core/signalr/introduction?view=aspnetcore-8.0>
- <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/social/?view=aspnetcore-9.0&tabs=visual-studio>
- <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/social/google-logins?view=aspnetcore-9.0>
- <https://developers.google.com/maps/documentation/directions/overview>
- <https://developers.google.com/maps/documentation/places/web-service/overview>
- <https://firebase.google.com/docs/cloud-messaging>
- <https://docs.aws.amazon.com/ec2/>
- <https://docs.aws.amazon.com/s3/>
- <https://reactjs.org/docs/getting-started.html>
- <https://reactnative.dev/docs/getting-started>
- <https://docs.expo.dev>

2 Opis sistema

2.1 Perspektiva proizvoda

Na ovom mjestu fokusirati ćemo se na to kako se naš proizvod uklapa u jedan širi kontekst, te kakvu interakciju ostvaruje sa drugim sistemima i korisnicima.

Prije svega recimo to da sistem "Bazaar" Marketplace predstavlja sveobuhvatnu e-commerce platformu dizajniranu da funkcioniše kao samostalan proizvod, ali sa sposobnošću integracije sa eksternim servisima radi proširenja funkcionalnosti. Platforma je koncipirana kao centralno mjesto koje povezuje tri ključne korisničke grupe: kupce, prodavce i administratore, nudeći im specifične interfejs i alate prilagođene njihovim potrebama.

Što se tiče korisničkih interfejsa, krenimo redom:

- **Bazaar mobilna aplikacija (za kupce):** Primarni interfejs za kupce, dostupan za iOS i Android platforme. Omogućava pretragu, pregled, poređenje i kupovinu proizvoda, kao i upravljanje korisničkim profilom i narudžbama.
- **The Shop mobilna aplikacija (za prodavače):** Namjenski interfejs za prodavače, također dostupan za iOS i Android. Služi za upravljanje trgovinom, proizvodima (inventar, cijene), narudžbama i komunikacijom s kupcima.
- **Administratorski web portal:** Web-bazirani interfejs dostupan administratorima. Omogućava sveobuhvatno upravljanje platformom, uključujući korisnike, trgovine, proizvode, narudžbe, kategorije, cijene i reklame.

Softverski interfejsi također predstavljaju važnu stavku za spomenuti, a vrijedi istaći sljedeće:

- **Backend API (RESTful):** Jezgro sistema koje omogućava komunikaciju između mobilnih aplikacija, administratorskog portala i baze podataka. Svi podaci i poslovna logika su centralizovani na backendu.
- **Baza podataka:** Sistem koristi PostgreSQL kao relacijsku bazu podataka za čuvanje svih podataka o korisnicima, proizvodima, narudžbama, trgovinama, reklamama i ostalim relevantnim informacijama.
- **Eksterni servisi (API integracije):** Ovdje možemo spomenuti Routing APIs za optimizaciju ruta dostave, Email servise, FCM za slanje push notifikacija...
- **SignalR:** Za omogućavanje real-time funkcionalnosti, kao što je praćenje statusa reklama.

Na kraju, što se tiče komunikacionih interfejsa, imamo:

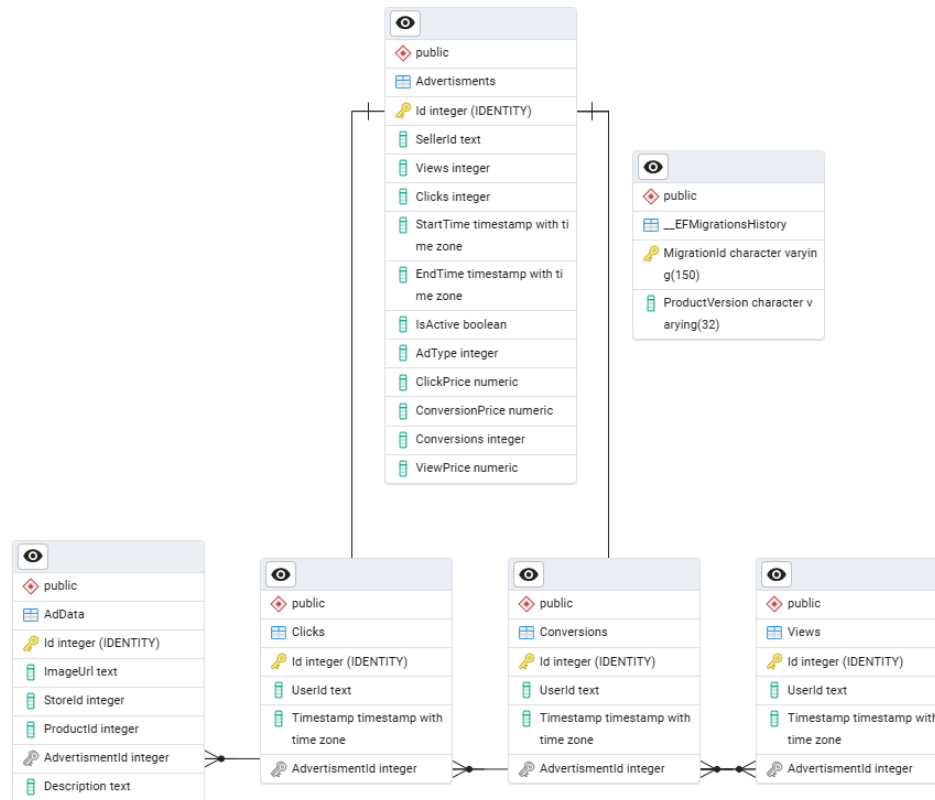
- **HTTPS (TLS/SSL):** Sva komunikacija između klijentskih aplikacija (mobilne i web) i backend API-ja odvija se preko sigurnog HTTPS protokola radi zaštite podataka prilikom prenosa.
- **WebSocket:** Koristi se za real-time komunikaciju putem SignalR-a.

Kada su u pitanju ograničenja i zavisnosti, sistem zahtijeva stabilnu internet konekciju za sve korisničke grupe kako bi se osigurala puna funkcionalnost. Mobilne aplikacije će zavisiti od operativnih sistema (iOS i Android) i njihovih prodavnica aplikacija za distribuciju i ažuriranja. Funkcionalnost integrisanih eksternih servisa (routing, analitika, email) zavisi od dostupnosti i ispravnosti njihovih API-ja.

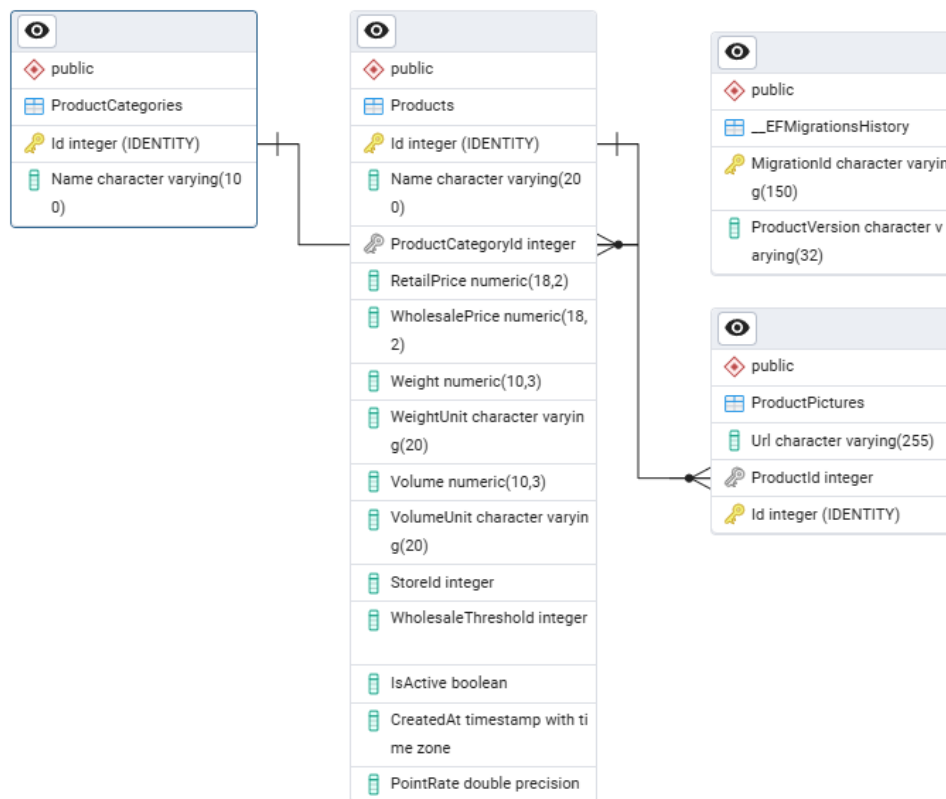
Sistem je dizajniran tako da bude skalabilan i prilagodljiv budućim potrebama tržišta i rastućem broju korisnika i transakcija. U nastavku ćemo spomenuti neke osnovne mogućnosti koje ovaj sistem nudi:

- **Za kupce:** Registracija, prijava, pretraga i filtriranje proizvoda, pregled detalja proizvoda i prodavnica, dodavanje proizvoda u korpu, proces naručivanja, praćenje statusa narudžbe, ostavljanje recenzija, komunikacija sa prodavačima i tehničkom podrškom.
- **Za prodavače:** Registracija, upravljanje profilom trgovine, dodavanje i ažuriranje proizvoda (opis, cijene, količine, slike), upravljanje narudžbama (potvrda, odbijanje, status dostave), komunikacija sa kupcima.
- **Za administratore:** Upravljanje korisničkim nalogima (kupci, prodavači), upravljanje trgovinama (odobravanje, deaktivacija), upravljanje kategorijama proizvoda, upravljanje proizvodima, pregled i upravljanje narudžbama, upravljanje cijenama, upravljanje reklamama, pristup analitičkim izvještajima i tehnička podrška.

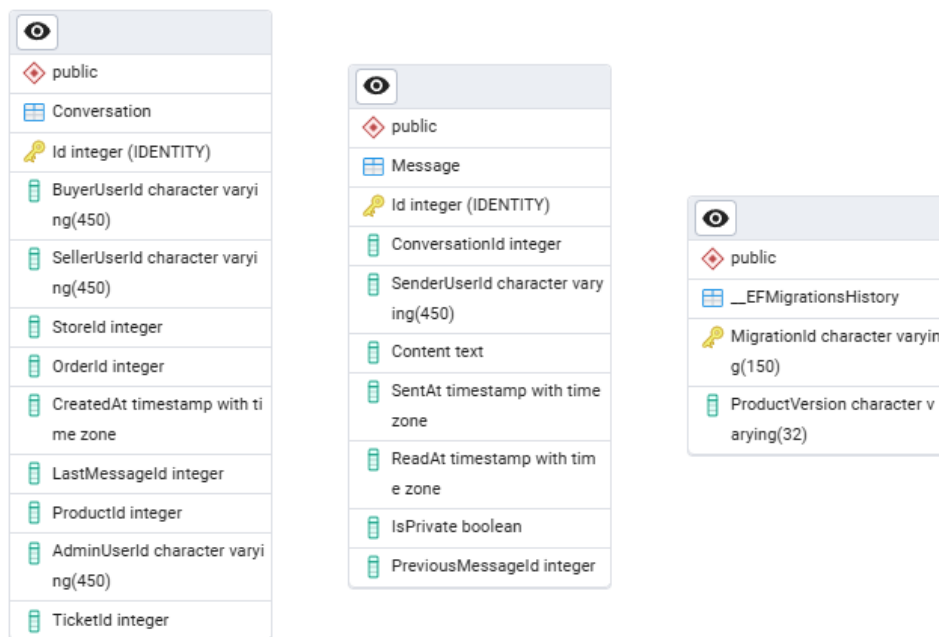
2.2 Entity Relationship Diagram



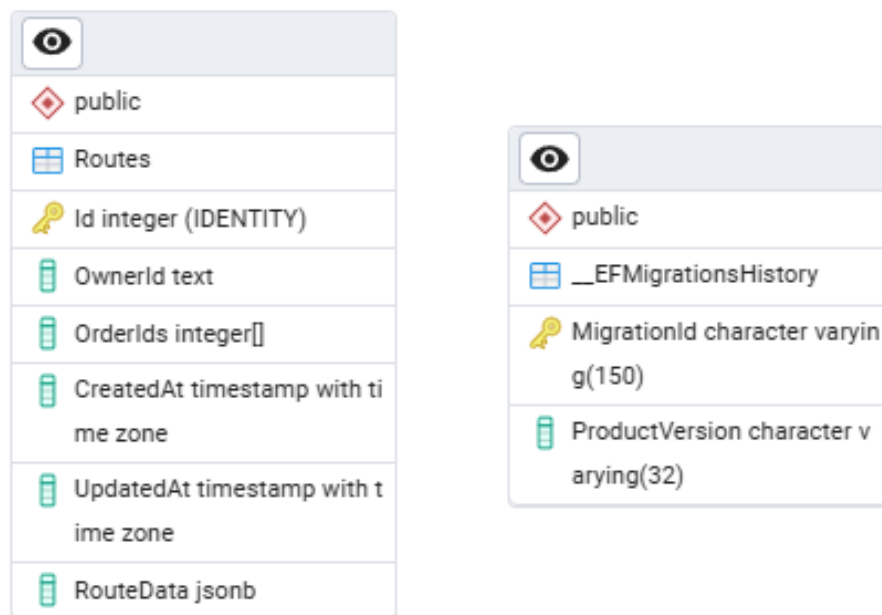
Slika 1. ERD: Modul za reklame (Advertisements, AdData, Clicks, Conversions, Views)



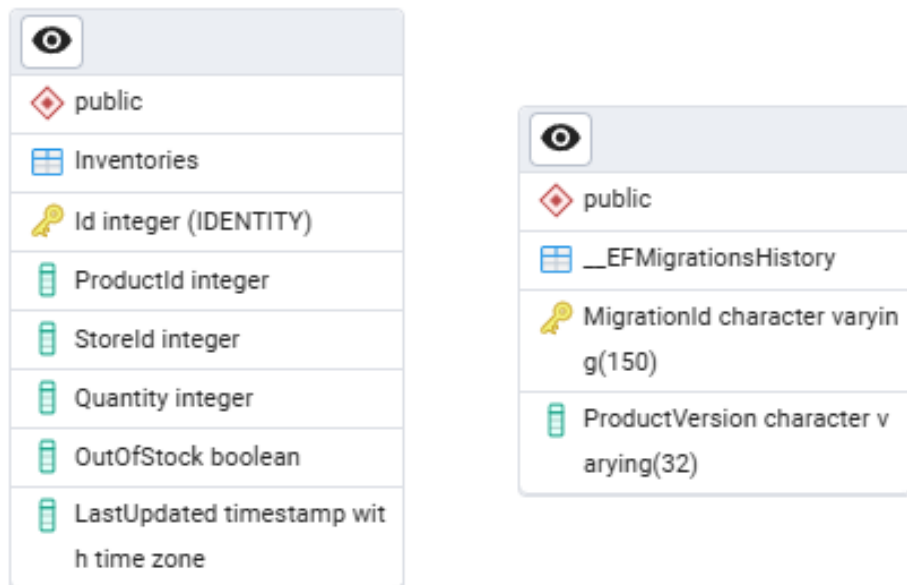
Slika 2. ERD: Modul za proizvode (ProductCategories, Products, ProductPictures)



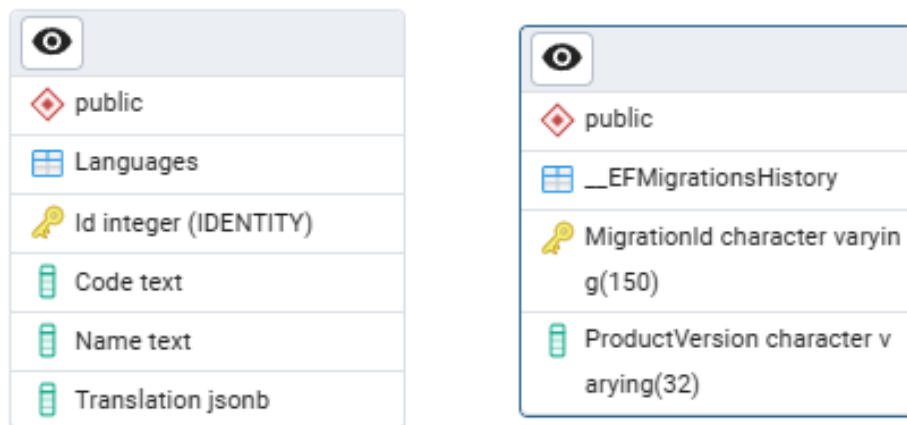
Slika 3. ERD: Modul za konverzacije (Conversation, Message)



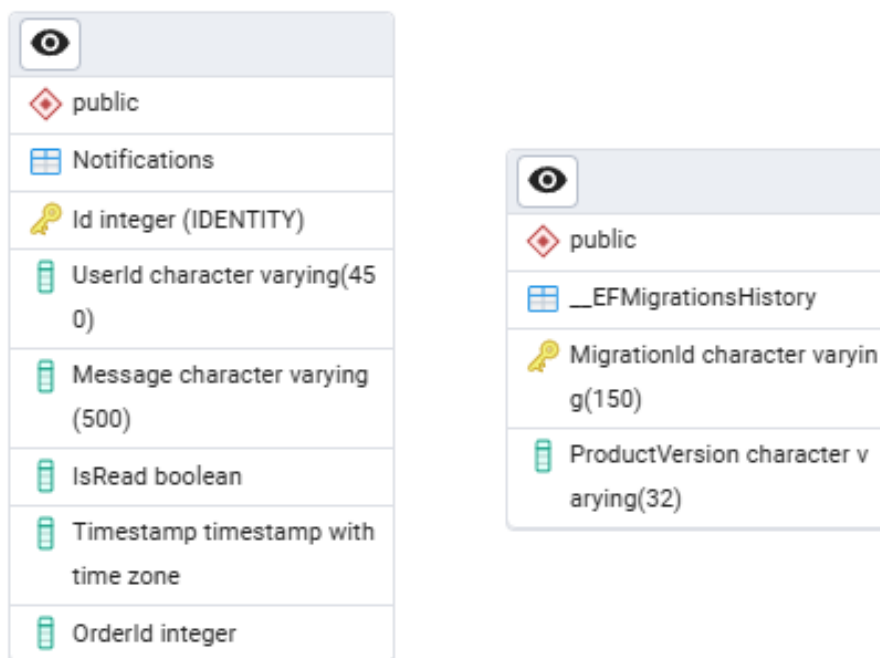
Slika 4. ERD: Modul za dostavu (Routes)



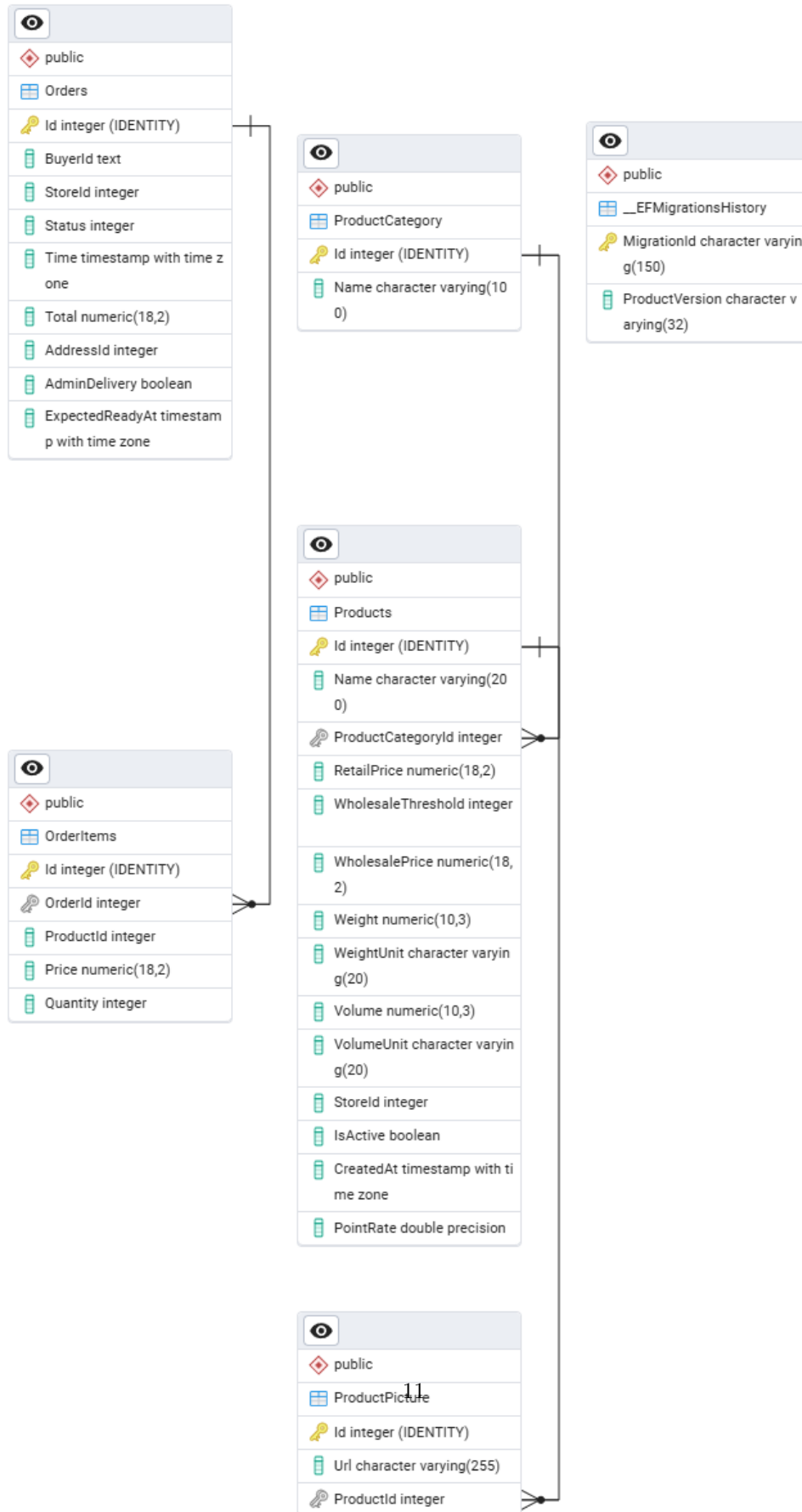
Slika 5. ERD: Modul za inventar (Inventories)

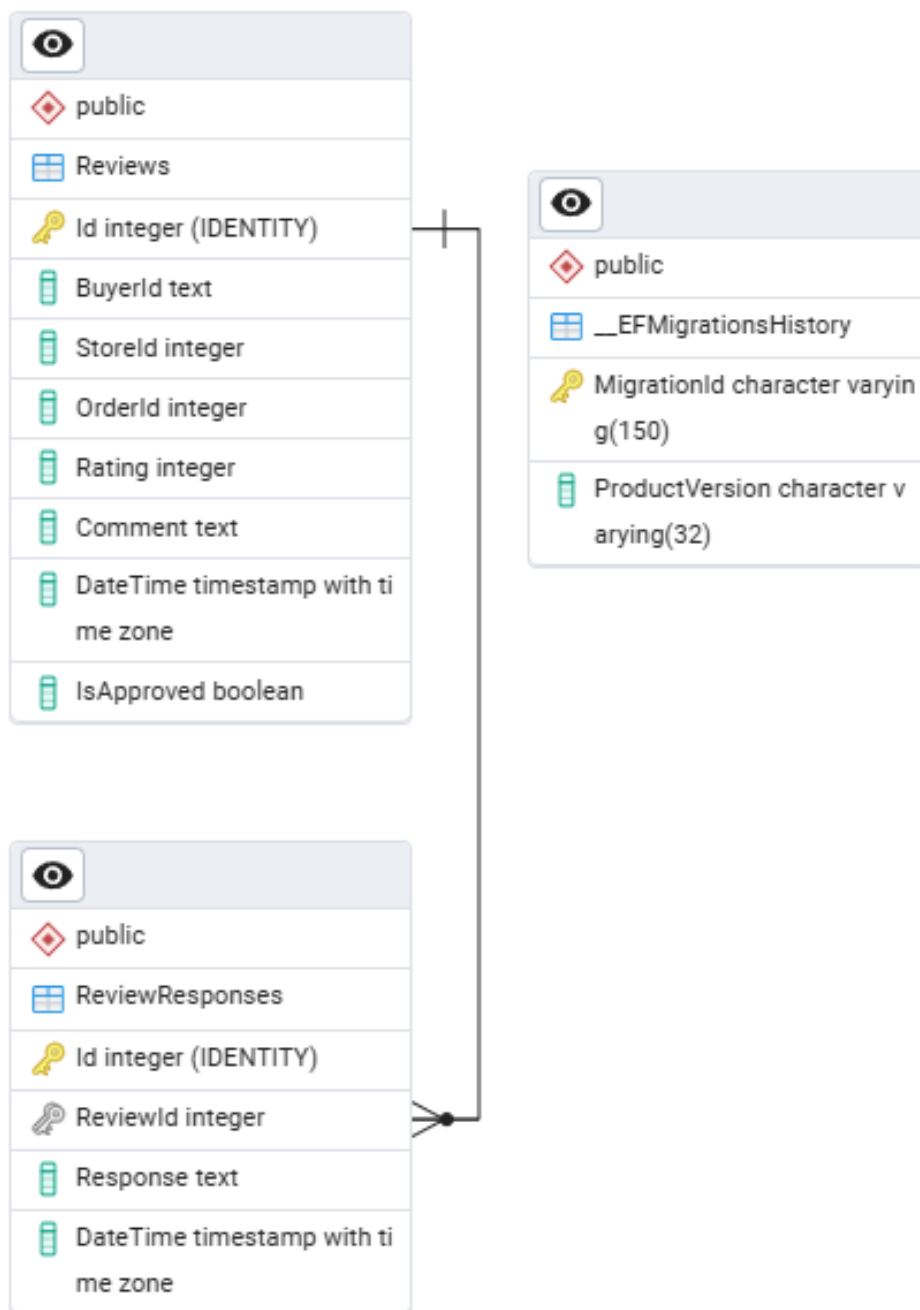


Slika 6. ERD: Modul za jezike (Languages)

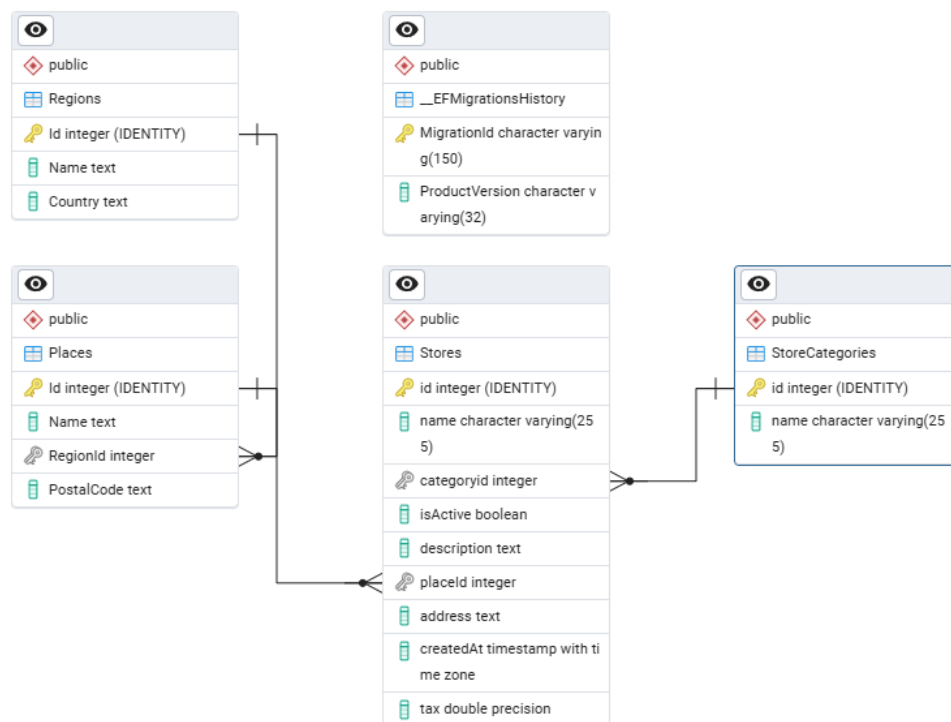


Slika 7. ERD: Modul za notifikacije (Notifications)

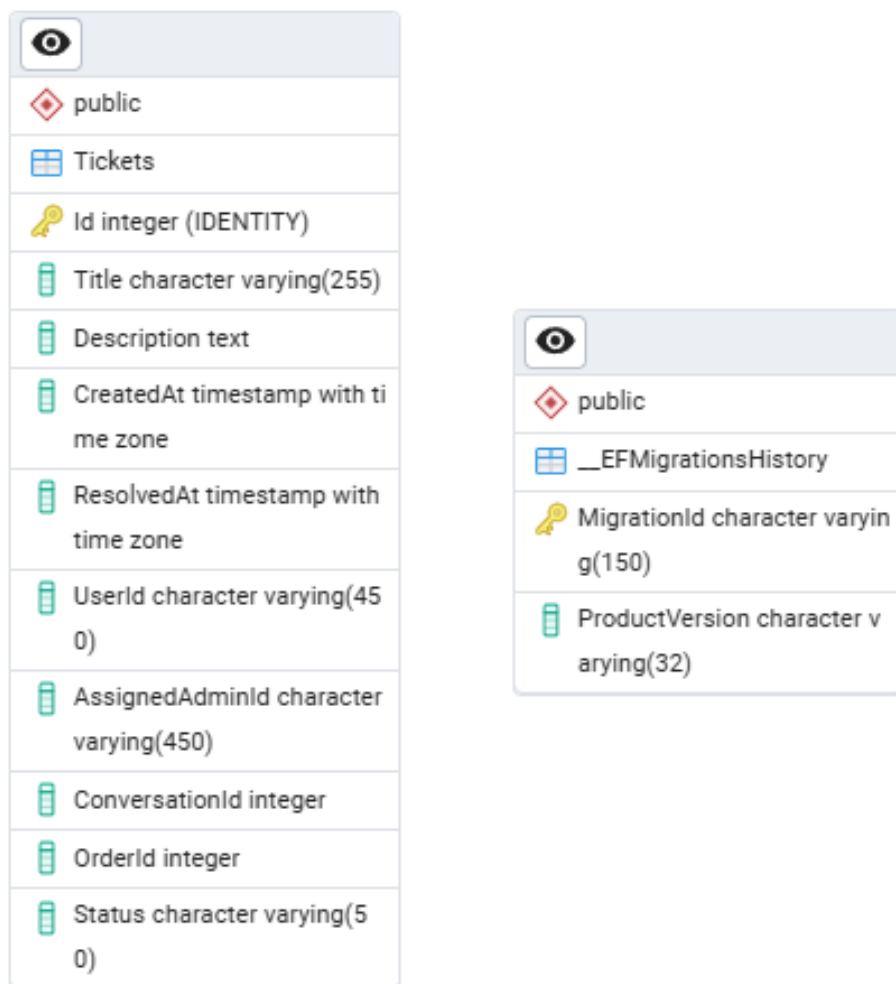




Slika 9. ERD: Modul za recenzije (Reviews, ReviewResponses)



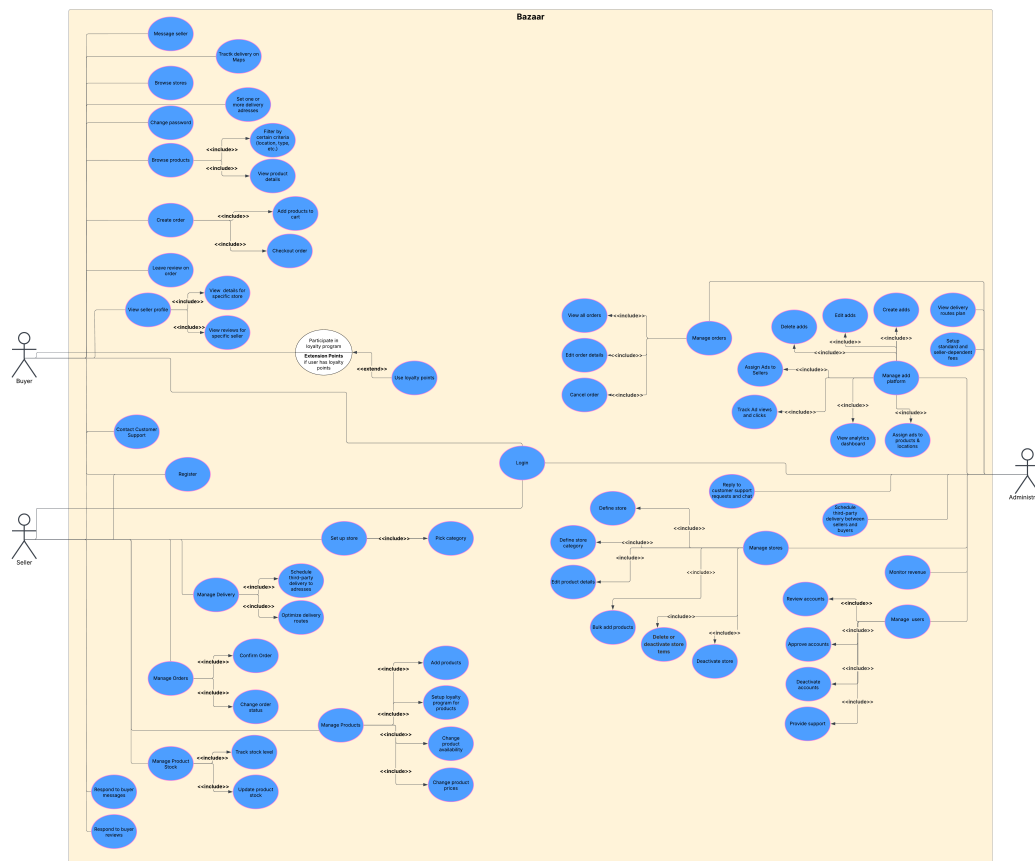
Slika 10. ERD: Modul za prodavnice (Regions, Places, Stores, StoreCategories)



Slika 11. ERD: Modul za tikete podrške (Tickets)



2.3 Korisničke klase i njihove karakteristike uz Activity i Use Case dijagrame

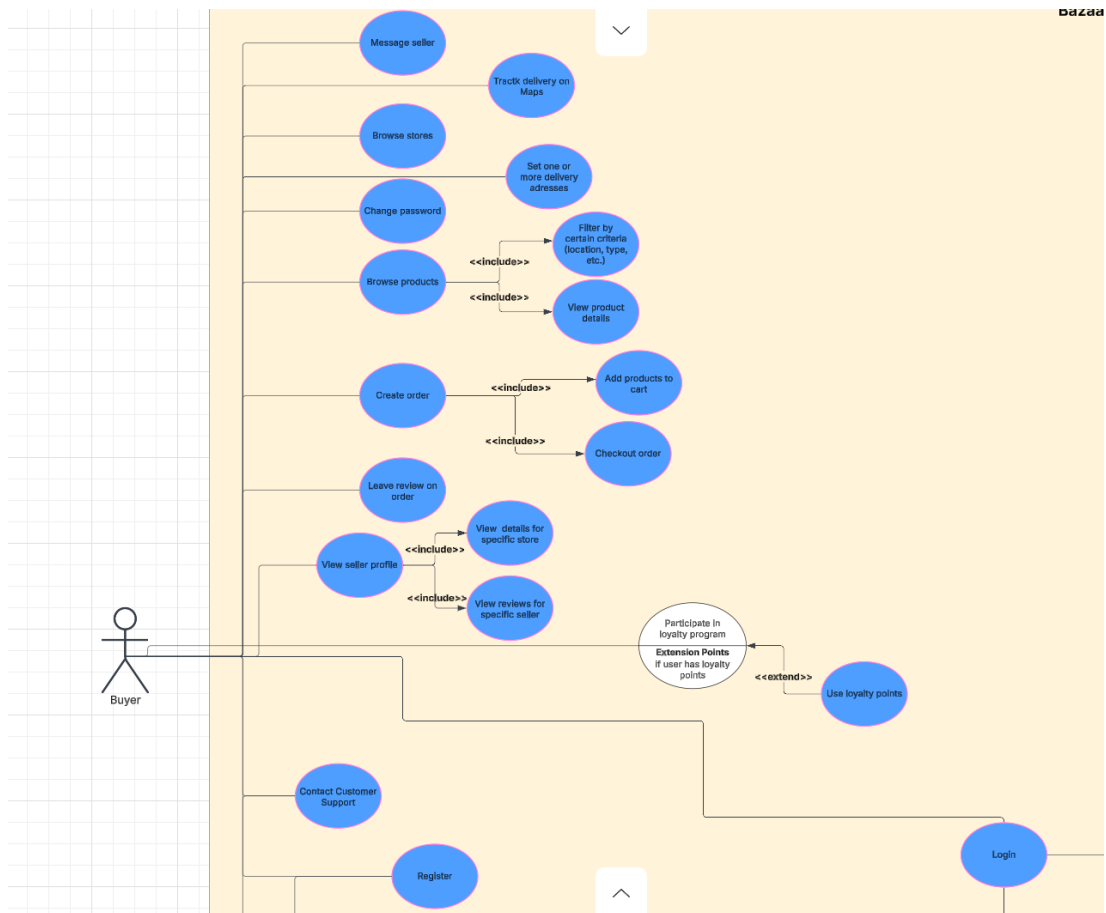


Slika 13. Opšti *Use Case* dijagram za sistem Bazaar Marketplace

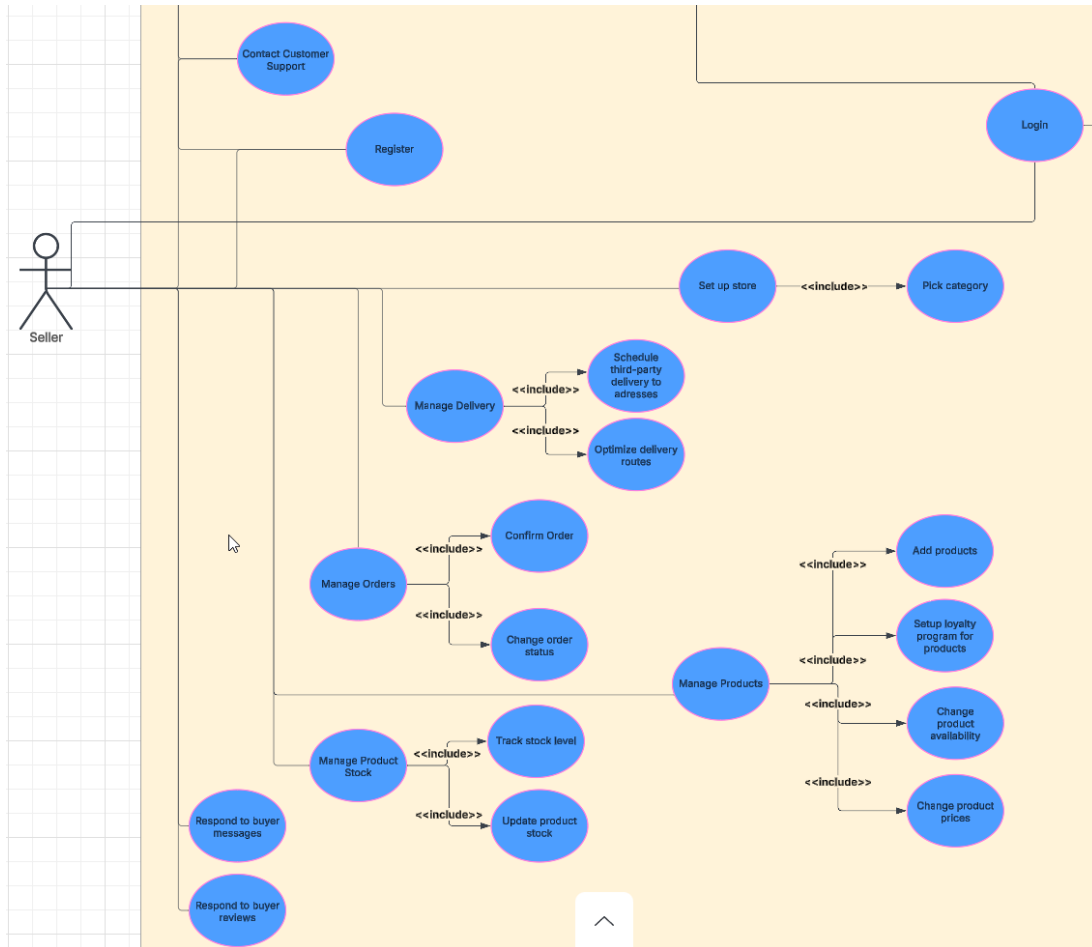
U okviru sistema **Bazaar Marketplace** definisane su tri glavne klase korisnika: *kupac* (**Buyer**), *prodavač* (**Seller**) i *administrator* (**Admin**). Svaka od ovih klasa posjeduje jasno definisanu ulogu i odgovarajuće funkcionalnosti u skladu sa svojom namjenom.

Kupci koriste platformu za pretragu i filtriranje proizvoda, pregled detalja artikala, upravljanje svojim narudžbama i komunikaciju s prodavačima. **Prodavači** imaju mogućnost dodavanja i uređivanja proizvoda, upravljanja zalihama i narudžbama, planiranja dostave i korištenja oglasnih alata. **Administratori** upravljaju svim aspektima sistema – korisnicima, narudžbama, prodavnicama, oglasima, naplatom i analitikom.

Dijagrami na slikama 14, 15 i 16 prikazuju uvećane segmente glavnog dijagrama iz slike 13, omogućujući pregledniju analizu funkcionalnosti po korisničkim ulogama.



Slika 14. Uvećani dio dijagrama *Use Case* dijagrama: funkcionalnosti kupca



Slika 15. Uvećani prikaz dijela *Use Case* dijagrama: funkcionalnosti prodavača



Slika 16. Uvećani prikaz dijela *Use Case* dijagrama: funkcionalnosti administratora

Use Case dijagram na slici 13 jasno prikazuje raspodjelu funkcionalnosti među korisnicima, s posebnim naglaskom na hijerarhiju procesa izraženu putem *include* i *extend* veza. Kupčev tok obuhvata sve faze korisničkog putovanja, počevši od pretrage i filtriranja proizvoda, preko odabira količine i potvrde narudžbe, do korištenja loyalty bodova i praćenja isporuke. Administratorske funkcionalnosti uključuju odobravanje korisnika, upravljanje oglasima uz praćenje metrika, te pristup analitičkom nadzoru sistema, čime se osigurava centralizovana kontrola nad svim operacijama. Prodavačev dijagram prikazuje njegove funkcionalnosti od postavljanja trgovine i upravljanja proizvodima, do obrade narudžbi i optimizacije isporuka.

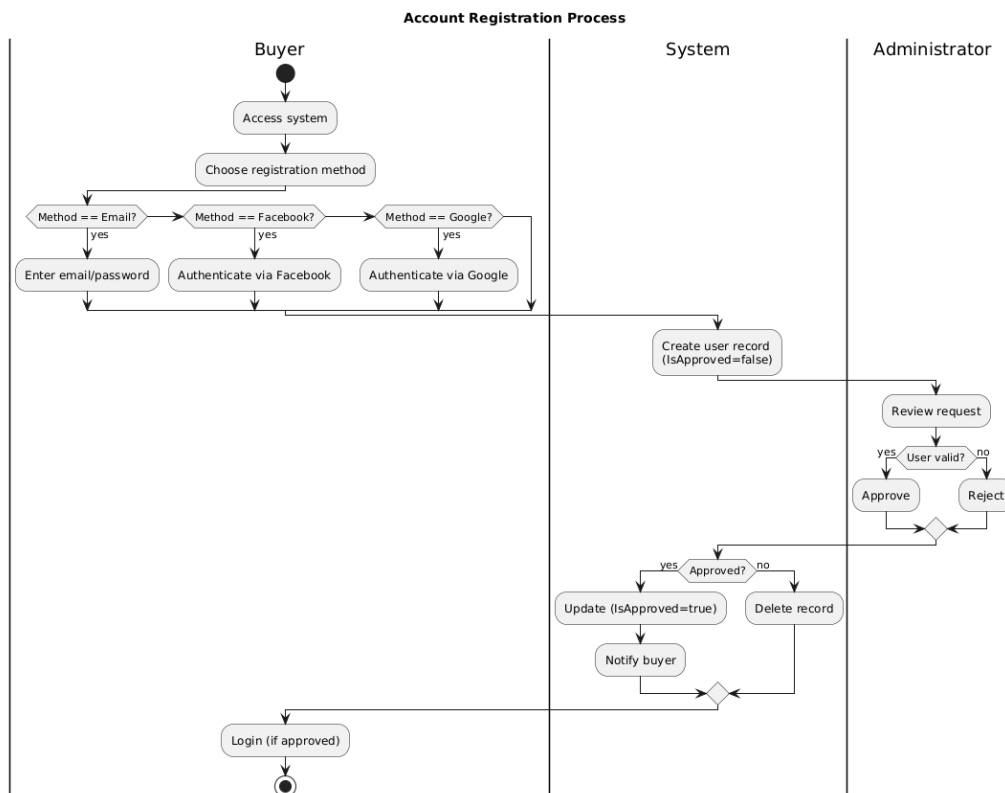
2.3.1 Activity dijagrami

Activity dijagrami se oslanjaju na prethodno definisane funkcionalnosti sistema i modeliraju ključne tokove aktivnosti za svaku korisničku ulogu. Fokus je stavljen na prikaz konkretnih koraka i međusobnih interakcija koje se odvijaju tokom najčešćih scenarija korištenja sistema.

Dijagrami prikazuju sljedeće scenarije:

- tok registracije korisničkog računa, koji uključuje interakciju kupca, sistema i adminis-

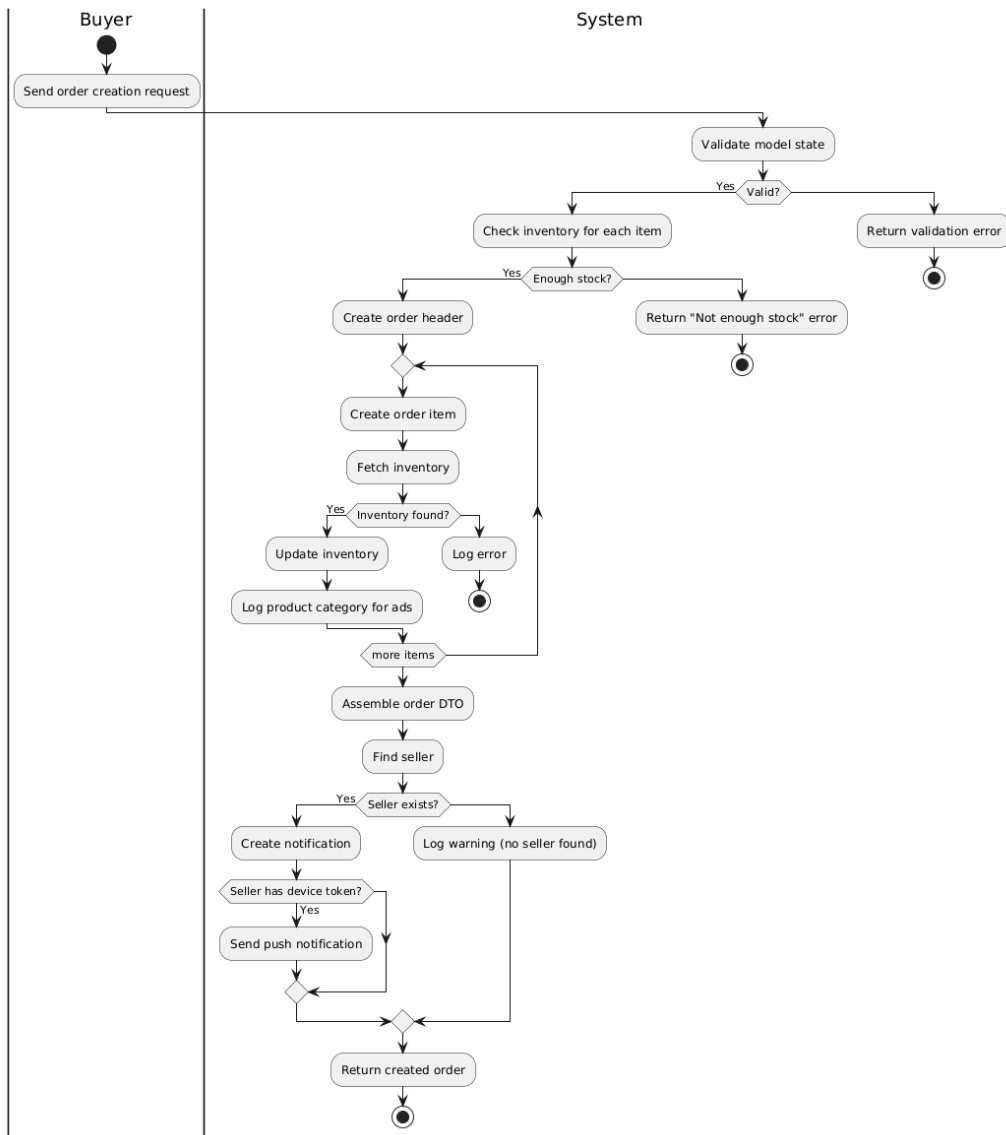
- tratora,
- proces kreiranja narudžbe od strane kupca, uključujući validaciju i obradu u sistemu,
 - postupak kreiranja prodavnice, koji obuhvata radnje prodavača, administratora i sistema.



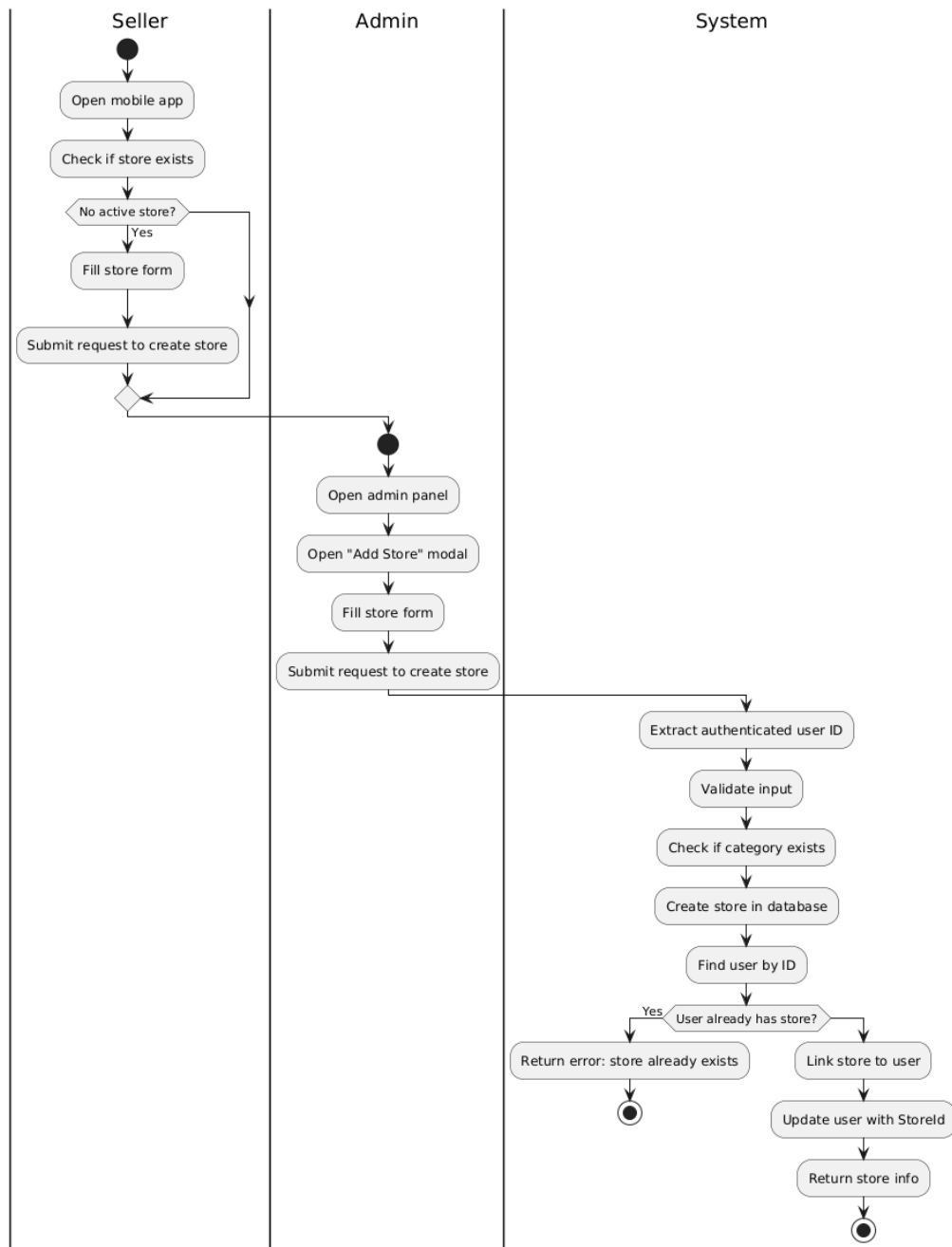
Slika 17. Activity dijagram registracije kupca

Dijagram 17 prikazuje proces registracije kupca kroz tri alternativna metoda: putem e-maila, Facebooka ili Google naloga. Nakon autentifikacije, sistem kreira korisnički zapis sa statusom *IsApproved = false*. Administrator potom pregledava zahtjev, te u zavisnosti od validnosti korisnika ili odobrava i ažurira status, ili odbija i briše zapis. Na kraju, sistem obavještava korisnika i omogućava prijavu ako je odobren.

Dijagram 18 modelira proces kreiranja narudžbe koji započinje zahtjevom kupca i obuhvata validaciju unosa, provjeru dostupnosti zaliha i formiranje stavki narudžbe. Ukoliko validacija ne uspije ili zaliha nije dovoljna, proces se prekida uz odgovarajuću poruku o grešci. Uspješan tok uključuje ažuriranje zaliha, kreiranje narudžbe i povezivanje s prodavačem. Ako prodavač postoji i ima aktivan uređaj, sistem šalje push notifikaciju. Na kraju, narudžba se vraća kupcu kao potvrđena.



Slika 18. Activity dijagram kreiranja narudžbe



Slika 19. Activity dijagram kreiranja narudžbe

Dijagram 19 prikazuje proces kreiranja prodavnice koji može pokrenuti i prodavač putem mobilne aplikacije i administrator putem administracijskog panela. U oba slučaja, korisnik popunjava formular s podacima o prodavnici i šalje zahtjev. Sistem zatim provjerava validnost unosa, postojanje kategorije i identitet korisnika. Ukoliko korisnik već ima aktivnu prodavnicu, zahtjev se odbacuje s odgovarajućom porukom. U suprotnom, prodavnica se kreira, povezuje s korisnikom i vraća se potvrda o uspješnom kreiranju.

2.4 Operativno okruženje

Bazaar predstavlja distribuirani e-commerce sistem zasnovan na klijent-server arhitekturi. Tri klijentske aplikacije povezane s ovim sistemom su: mobilna aplikacija za kupce, mobilna aplikacija za prodavače i web administratorski panel. Serverska strana sistema implementirana je u ASP.NET 8 koristeći C# programski jezik i Entity Framework Core za pristup PostgreSQL bazi podataka. Sve klijentske aplikacije komuniciraju sa serverom putem REST API-ja i SignalR protokola. Autentifikacija korisnika zasniva se na JSON Web Token (JWT) mehanizmu, dok se za notifikacije koriste Firebase Cloud Messaging (Android) i Apple Push Notification Service (iOS). Mobilne aplikacije razvijene su u React Native okviru, koristeći Expo platformu, i podržavaju Android i iOS operativne sisteme. Administratorski panel razvijen je korištenjem React biblioteke i Vite alata, te je prilagođen modernim web preglednicima. Razvoj aplikacije mora biti podržan na Windows, Linux i MacOS operativnim sistemima. Mobilne aplikacije podržavaju Android verzije 8 i više, te iOS verzije 15 i više. Sistem je dizajniran tako da u lokalnom okruženju koriste se resursi na lokalnom računaru, dok je za produkciju predviđena AWS infrastruktura. Automatizacija procesa isporuke web aplikacije obezbijedena je korištenjem GitHub Actions CI/CD servisa.

2.5 Ograničenja dizajna i implementacije

Bazaar sistem je funkcionalno ograničen na sljedeće definisane grupe mogućnosti, prema potrebama korisnika: Kupci imaju pristup isključivo mobilnoj aplikaciji: omogućeno im je pretraživanje proizvoda i prodavnica, upravljanje narudžbama, komunikacija s prodavačima i administratorima, učestvovanje u programu lojalnosti, te ostavljanje recenzija. Prodavači koriste odvojenu mobilnu aplikaciju sa tačno određenim mogućnostima: upravljanje proizvodima, narudžbama, cijenama i zalihama. Administratorski panel dostupan je isključivo putem web aplikacije: omogućava pregled i uređivanje svih korisnika, proizvoda, narudžbi, upravljanje oglasima i podrškom.

3 Karakteristike sistema

3.1 Opis i prioritet

Bazaar je e-commerce platforma koja omogućava direktnu komunikaciju i razmjenu proizvoda između krajnjih korisnika i malih prodavača putem mobilnih i web aplikacija. Sistem se sastoji od tri funkcionalna dijela: mobilne aplikacije za kupce, mobilne aplikacije za prodavače i web aplikacije za administratore.

Cilj sistema jeste omogućiti brz, jednostavan i siguran proces online kupovine i prodaje, uz administrativnu kontrolu nad oglasima, korisnicima i statistikom. Sistem rješava izazove povezane s neregulisanim tržištem, netransparentnom ponudom i otežanim upravljanjem korisnicima i sadržajem. Prioritet leži u potrebi za digitalizacijom maloprodaje, veleprodaje i omogućavanju efikasnog kanala za kupovinu i prodaju krajnjim korisnicima.

3.2 Funkcionalni zahtjevi

Buyer (Kupac)

- Kupac ima mogućnost registracije putem *email*-a, *Google* ili *Facebook* naloga, te *login* u aplikaciju, uključujući opciju promjene zaboravljene lozinke.
- Kupac može pretraživati i filtrirati proizvode i prodavnice po kategorijama, lokaciji, tipu i drugim kriterijima, uz pregled detalja proizvoda (naziv, opis, cijena, dostupnost i slike) na organizovanom i preglednom interfejsu.
- Kupac može dodavati proizvode u korpu, uređivati sadržaj iste i izvršiti *checkout*, uz primanje notifikacija o statusu narudžbe (potvrđeno, spremno, poslano, otkazano, dostavljeno, odbijeno).
- Kupac ima pristup historiji svojih narudžbi i mogućnost ostavljanja recenzija i ocjena za prodavnice, kao i pregled recenzija drugih korisnika.
- Kupac može pregledati profil prodavača i komunicirati s njim putem direktnog *chat*-a u vezi narudžbi, proizvoda ili općih informacija, te prima notifikacije kada primi poruku ili kada prodavac odgovori na recenziju.
- Kupac prima podsjetnike da ostavi recenziju nakon kupovine.
- Kupcu se prikazuju oglasi u skladu s *recommender* sistemom baziran na pojačanom učenju, a podaci o prikazima, klikovima i konverzijama se bilježe i dostavljaju administraciji.
- Kupac može kontaktirati korisničku podršku putem *ticketing* sistema i direktnog *chat*-a radi rješavanja tehničkih problema ili pitanja u vezi narudžbi.
- Kupac može dodavati i upravljati s više adresa za dostavu, kao i pregledati plan dostavne rute na mapi.
- Kupac učestvuje u programu lojalnosti – prikuplja bodove tokom kupovine, pregleda stanje svojih bodova i koristi ih za ostvarivanje popusta pri budućim narudžbama.

Seller (Prodavač)

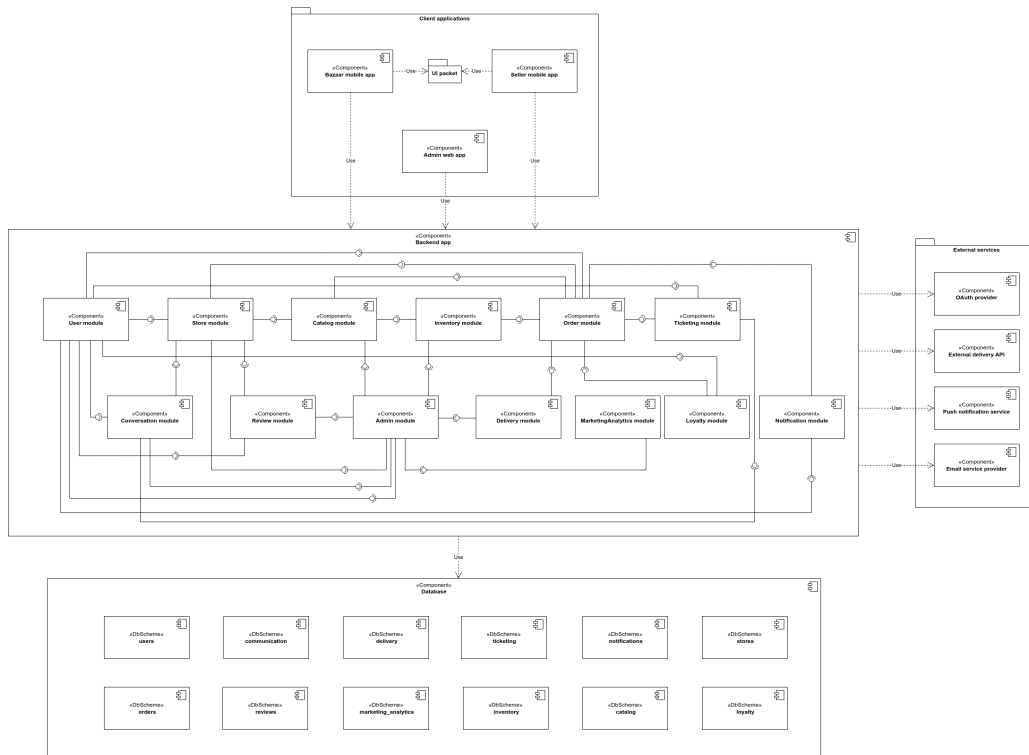
- Prodavač ima mogućnost registracije putem *email*-a, *Google* ili *Facebook* naloga, te pristupa aplikaciji nakon odobrenja od strane *admin*-a.
- Prodavač može kreirati i ažurirati informacije o svojoj prodavnici, uključujući naziv, opis, adresu i kategoriju (koja se dodjeljuje od strane *admin*-a).
- Upravljanje proizvodima uključuje:
 - Dodavanje, uređivanje i brisanje proizvoda.
 - Postavljanje maloprodajnih i veleprodajnih cijena sa definiranim pragovima količine.
 - Postavljanje naziva, kategorije, mase/volumena, slika visoke kvalitete i opisa proizvoda.
- Upravljanje zalihama podrazumijeva automatsko kreiranje zapisa prilikom dodavanja proizvoda, mogućnost praćenja količine na stanju, ručno ažuriranje zaliha te automatsko označavanje proizvoda kao *Out of Stock* kada zaliha dosegne nulu. Prodavač prima notifikaciju kada proizvod nije više dostupan.
- Prodavač ima uvid u sve pristigle narudžbe i mogućnost ažuriranja statusa narudžbi (potvrđeno, spremno, poslano, otkazano, dostavljeno, odbijeno), uz notifikacije o novim narudžbama.
- Prodavač može podesiti *loyalty program* za svoje proizvode, definisati broj bodova koje kupac dobija po kupovini i pratiti statistiku (dodijeljeni bodovi, iskorišteni bodovi i finansijski efekat).
- Komunikacija s kupcima omogućena je putem direktnog *chat*-a unutar aplikacije. Prodavač može odgovarati na poruke, primiti notifikacije o pristiglim porukama, te reagovati na recenzije koje su kupci ostavili za proizvode ili prodavnicu.
- Prodavač može kontaktirati korisničku podršku putem *ticketing* sistema i direktnog *chat*-a sa *admin*-om u vezi tehničkih problema.
- Za dostavu, prodavač može izabrati opciju zakazivanja isporuka putem treće strane (putem *admin*-a), ili označiti dostavu kao samostalno upravljano. Sistem generiše optimalnu rutu dostave za odabrane narudžbe, prikazanu putem *Google Maps*.

Administrator

- Administrator ima mogućnost prijave u sistem.
- Administrator ima mogućnost pregleda svih korisnika grupisanih po ulogama (kupac, prodavač), odobravanja registracija, ručnog dodavanja novih korisnika i deaktivacije korisničkih naloga.
- Administrator ima mogućnost upravljanja prodavnicama: kreiranje, uređivanje i deaktivacija, kao i definisanje kategorija proizvoda i prodavnica.
- Administrator ima mogućnost upravljanja proizvodima bilo koje prodavnice: masovni unos proizvoda putem Excel/CSV datoteka, uređivanje detalja, deaktivacija i brisanje proizvoda.

- Administrator ima mogućnost upravljanja narudžbama: pregled, uređivanje i otkazivanje narudžbi u cilju pružanja podrške korisnicima.
- Administrator ima mogućnost upravljanja korisničkom podrškom: pregled i odgovaranje na tikete, promjena statusa tiketa (*Requested*, *Open*, *Resolved*), te direktna komunikacija putem *chat*-a, uključujući uvid u međukorisničke poruke uz poštovanje privatnosti.
- Administrator ima mogućnost organizacije dostave; kreiranje i optimizacija ruta sa prikazom na mapi.
- Administrator ima mogućnost upravljanja reklamama: dodavanje reklama, povezivanje s proizvodima, prodavnicama i korisnicima, praćenje metrika (prikazi, klikovi, konverzije, trošak).
- Administrator ima pristup analitičkom *dashboard*-u za pregled performansi prodaje, aktivnosti korisnika i efikasnosti marketinga.
- Administrator ima mogućnost definisanja i praćenja standardnih i individualnih naknada za prodavače i ukupnog prihoda platforme.

3.3 Dijagram komponenti



Slika 20. Dijagram komponenti Bazaar E-commerce sistema

4 Vanjski zahtjevi interfejsa

4.1 Korisnički interfejs

Korisnički interfejs Bazaar E-commerce Platforme se sastoji od univerzalne login/registracijske funkcionalnosti te tri glavne klijentske aplikacije prilagođene ulogama korisnika: mobilna aplikacija za kupce, mobilna aplikacija za prodavače i web-bazirani administrativni panel. Iako korisnici sistema različitih uloga imaju različite funkcionalnosti kojima mogu pristupiti, njihovi interfejsi teže konzistentnom dizajnu unutar svake aplikacije. U nastavku su navedeni glavni dijelovi korisničkog interfejsa kao i zahtjevi koje moraju ispuniti:

Stranica za registraciju i prijavu korisnika:

- Omogućava unos emaila i lozinke za registrovane korisnike.
- Opcija prijave putem Google i Facebook naloga.
- Opcija “Zaboravili ste lozinku?”.
- Proces registracije za nove korisnike (kupce i prodavače), pri čemu registracija zahtijeva odobrenje administratora.

Aplikacija za Kupce (Android/iOS): Glavni interfejs nakon uspješne prijave, omogućava pristup sljedećim funkcionalnostima:

- Pregled i pretraga prodavnica i proizvoda (sa filterima).
- Detaljan prikaz proizvoda (slike, opisi, cijene, dostupnost).
- Korpa za kupovinu (dodavanje, pregled, ažuriranje).
- Proces naručivanja (odabir adrese za dostavu).
- Ostavljanje recenzija i ocjena za proizvode i prodavnice.
- Program lojalnosti (skupljanje i korištenje bodova, pregled stanja).
- Korisnička podrška (kreiranje tiketa, direktan chat sa prodavačima i administratorima).
- Upravljanje adresama za dostavu.

Aplikacija za Prodavače (Android/iOS): Glavni interfejs nakon uspješne prijave i odobrenja naloga, omogućava pristup sljedećim funkcionalnostima:

- Upravljanje prodavnicom (kreiranje, ažuriranje informacija).
- Upravljanje proizvodima (dodavanje, izmjena, brisanje, cijene, slike, opisi).
- Upravljanje zalihama (praćenje stanja, ažuriranje količina, notifikacije o niskim zalihama).
- Upravljanje narudžbama (pregled, ažuriranje statusa).
- Program lojalnosti (postavke bodova za proizvode, statistika).
- Komunikacija sa kupcima (chat).

- Korisnička podrška (kreiranje tiketa administratorima, chat sa administratorima).
- Upravljanje recenzijama (pregled, odgovaranje, notifikacije).
- Logistika dostave (opcija za zakazivanje preko treće strane ili samostalno upravljanje).

Administratorski Panel (Web): Glavni interfejs za administratore nakon uspješne prijave, sastoji se od narednih interfejsa/modula:

- Upravljanje korisnicima (pregled svih korisnika, odobravanje registracija, aktivacija/deaktivacija naloga).
- Upravljanje prodavnicama (pregled, izmjena detalja, deaktivacija).
- Upravljanje proizvodima (pregled, modifikacija, deaktivacija/brisanje).
- Upravljanje narudžbama (pregled svih narudžbi, praćenje statusa, modifikacija).
- Korisnička podrška (pregled tiketa, dodjela, chat sa korisnicima).
- Naplata od Prodavača (postavke provizija, praćenje prihoda).
- Analitika (pregled dashboard-a sa performansama prodaje, aktivnostima korisnika).
- Logistika dostave (centralizovano upravljanje, optimizacija ruta).
- Upravljanje oglasima (povezivanje sa entitetima, dodavanje, praćenje metrika).

4.2 Hardverski interfejs

Sve komponente sa serverske strane (centralni backend sistem) izvršavaju se putem AWS EC2 instance. Za krajnje korisnike aplikacija je dostupna na mobilnim uređajima sa Android ili iOS operativnim sistemima, dok je administratorski interfejs prilagođen za pristup putem *browsera*. Budući da aplikacija zahtjeva pristup internetu za rad, neophodno je korištenje odgovarajuće mrežne opreme kao što su modem, ruter, Ethernet kabl i slično.

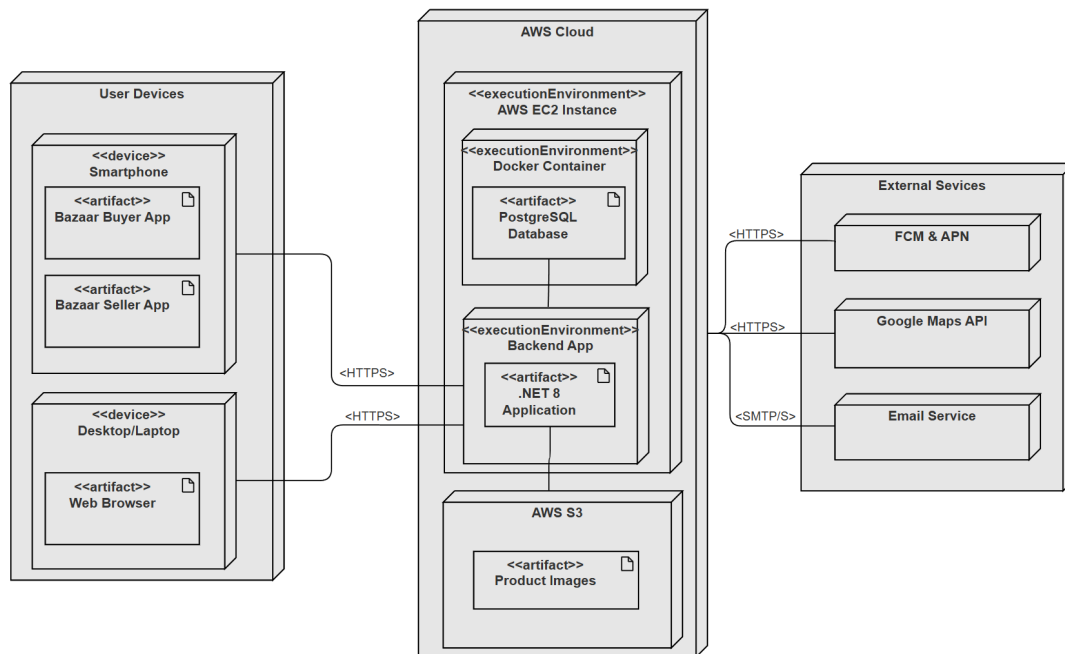
4.3 Softverski interfejs

Sve stranice/ekrani softverskog interfejsa trebaju imati jasno razdvojenu glavnu navigaciju/meni od ostatka sadržaja. Izgled glavnog menija i opšta struktura trebaju biti konzistentni unutar svake klijentske aplikacije. Sve stranice/ekrani interfejsa trebaju koristiti konzistentnu vizualnu temu unutar svake aplikacije. Softverski interfejs backend sistema mora imati mogućnost da se poveže i komunicira sa PostgreSQL bazom podataka. Sva komunikacija sa bazom podataka odvija se indirektno, putem API poziva koje generišu klijentske aplikacije odabirom odgovarajućih opcija na menijima, formama ili unosom podataka. Sva polja za unos podataka moraju biti praćena odgovarajućim labelama ili pružati jasne informacije o tome koje podatke je potrebno unijeti na nedvosmislen način (npr. placeholder tekst). Navigacija unutar aplikacija treba biti što manje zamorna za korisnike. Interfejse sa manjim sadržajem ili za specifične, fokusirane akcije (npr. potvrda brisanja, unos manje količine podataka) treba realizovati kao modalne prozore (pop-up).

4.4 Komunikacijski interfejs

Komunikacijska arhitektura slijedi klijent-server model. Backend sistem (.NET 8) služi kao centralni server, dok su mobilne aplikacije (React Native) i web administrativni panel (React) klijenti. Za komunikaciju između klijenta i servera koristi se HTTPS protokol zbog pojačane sigurnosti. Aplikacija komunicira sa bazom podataka (PostgreSQL) i različitim servisima (npr. Firebase Cloud Messaging, AWS S3) primarno pomoću REST API funkcijskih poziva. Za real-time chat funkcionalnosti koristi se SignalR. Većina klijent-server komunikacije putem REST API-ja je bez stanja (stateless), oslanjajući se na JWT (JSON Web Tokens) za autentifikaciju i autorizaciju svakog zahtjeva. SignalR konekcija za chat je perzistentna dok je korisnik aktivan u chat interfejsu. Uniformna ograničenja interfejsa definišu interfejs između klijenta i servera, omogućavajući da se klijentski i serverski dijelovi samostalno razvijaju. U Bazaar sistemu jasno su razdvojene klijentske i serverske uloge. Komunikacija sa bazom podataka, indirektno preko API-ja, omogućava klijentima da dobijaju i šalju podatke u obliku JSON formata. Korišten je e-mail servis za komunikaciju sa korisnikom, primarno za potvrdu prilikom izvršavanja funkcionalnosti ponovnog postavljanja lozinke. Za push notifikacije na mobilnim uređajima koriste se Firebase Cloud Messaging (FCM) za Android i Apple Push Notification service (APNs) za iOS, kojima upravlja backend servis.

4.5 Dijagram raspoređivanja



Slika 21. Dijagram Raspoređivanja

Dijagram raspoređivanja (Slika 21) prikazuje fizičku arhitekturu sistema Bazaar Marketplace. On ilustruje ključne hardverske i softverske čvorove na kojima se izvršavaju komponente sistema, kao i glavne komunikacione putanje između njih. Sistem je podijeljen na tri osnovne cjeline: korisničke uređaje, AWS cloud infrastrukturu i eksterne servise.

5 Nefunkcionalni zahtjevi

5.1 Upotrebljivost sistema

Sistem "Bazaar" će biti dizajniran s fokusom na intuitivnost i lakoću korištenja za sve tri korisničke grupe: kupce, prodavače i administratore.

- **Za kupce (Bazaar mobilna aplikacija):** Osigurat će se jednostavan proces registracije i prijave (uključujući opcije putem Email-a, Google-a, FB-a). Aplikacija će podrazumijevano koristiti bosanski jezik, s opcijom engleske lokalizacije. Navigacija, pretraga proizvoda, proces narudžbe i pregled historije narudžbi bit će logični i zahtijevat će minimalan broj koraka.
- **Za prodavače (The Shop mobilna aplikacija):** Interfejs će biti jednostavan i optimiziran za brzo upravljanje trgovinom, proizvodima (cijene, količine, dostupnost), narudžbama i komunikacijom s kupcima. Aplikacija će također podrazumijevano koristiti bosanski jezik.
- **Za administratore (Web portal):** Administratorski portal pružit će jasan pregled i efikasne alate za upravljanje korisnicima, trgovinama, proizvodima, narudžbama i kategorijama. Proces postavljanja trgovina i inicijalne registracije prodavaca od strane tehničkog tima bit će podržan efikasnim administratorskim alatima.

Sve aplikacije će pružati jasne povratne informacije korisnicima o njihovim akcijama i statusu sistema, te efikasno upravljanje greškama uz korisne poruke.

5.2 Sigurnost sistema

Sigurnost platforme "Bazaar" je prioritet i obuhvatit će zaštitu podataka i transakcija na svim nivoima.

- **Autentifikacija i autorizacija:** Implementirat će se robusni mehanizmi autentifikacije (npr. JWT za administratore, Email/Google/FB za korisnike mobilnih aplikacija) i autorizacije kako bi se osiguralo da korisnici imaju pristup samo onim podacima i funkcionalnostima koje odgovaraju njihovoj ulozi.
- **Zaštita podataka:** Svi osjetljivi podaci, uključujući lične podatke korisnika, detalje narudžbi i informacije o trgovinama, bit će zaštićeni enkripcijom kako u tranzitu (TLS/SSL) tako i u mirovanju.
- **Integritet podataka:** Osigurat će se integritet podataka kroz validacije na frontend i backend strani, te konzistentnost podataka u bazi.
- **Administratorski pristup:** Pristup administratorskom portalu bit će strogo kontroliran i zaštićen.
- **Treće strane:** Integracije s trećim stranama bit će realizirane preko sigurnih i provjerenih protokola.
- **Prevenција napada:** Sistem će biti zaštićen od uobičajenih web i mobilnih sigurnosnih prijetnji (npr. XSS, SQL injection, CSRF).

5.3 Dostupnost sistema

Cilj je osigurati visoku dostupnost svih komponenti sistema "Bazaar" kako bi korisnici mogli neometano koristiti platformu.

- **Visok uptime:** Težit će se ka 99.9% dostupnosti za backend servise i frontend aplikacije (mobilne aplikacije Bazaar i The Shop, te web administratorski portal).
- **Redundantnost i oporavak:** Infrastruktura na cloud platformi, uz korištenje kontejnerizacije i orkestracije, bit će dizajnirana da podrži redundantnost ključnih servisa i brz oporavak u slučaju otkaza.
- **Monitoring:** Kontinuirani monitoring sistema omogućit će proaktivno identificiranje i rješavanje potencijalnih problema koji bi mogli uticati na dostupnost.
- **Održavanje:** Planirana održavanja bit će najavljena unaprijed i izvođena u periodima najmanje aktivnosti korisnika kako bi se minimizirao utjecaj na dostupnost.

5.4 Održavanje sistema

Sistem "Bazaar" će biti razvijen s ciljem lakog održavanja, nadogradnje i otklanjanja grešaka.

- **Modularna arhitektura:** Jasno razdvojene komponente (React Native mobilne aplikacije, React Vite web aplikacija, .NET backend servisi) olakšat će nezavisno razvijanje, testiranje i ažuriranje dijelova sistema.
- **Kvalitet koda:** Korištenje modernih frameworka i pridržavanje najboljih praksi kodiranja osigurat će čitljiv, dobro strukturiran i dokumentiran kod.
- **CI/CD pipeline:** Automatizirani procesi za integraciju i isporuku (CI/CD) omogućit će brze i pouzdane izmjene i nadogradnje sistema.
- **Logging i monitoring:** Detaljno logiranje aktivnosti sistema i performansi, kao i alati za monitoring, pomoći će u brzom dijagnostici i rješavanju problema.
- **Konfigurabilnost:** Ključni parametri sistema (npr. valuta, kategorije proizvoda) bit će lako konfigurabilni kroz administratorski portal, smanjujući potrebu za izmjenama koda pri čestim promjenama.

5.5 Skalabilnost sistema

Platforma "Bazaar" mora biti sposobna da podrži rast broja korisnika (kupaca i prodavača), proizvoda, narudžbi i transakcija bez degradacije performansi ili dostupnosti.

- **Horizontalna skalabilnost:** Backend servisi razvijeni u .NET-u i postavljeni na cloud infrastrukturu bit će dizajnirani da omoguće horizontalno skaliranje (dodavanje više instanci servisa) prema potrebi.
- **Skalabilnost baze podataka:** Odabir i konfiguracija baze podataka vodit će računa o potrebama za skaliranjem kako bi se efikasno upravljalo rastućom količinom podataka.
- **Optimizacija resursa:** Efikasno korištenje resursa (CPU, memorija, mreža) na svim nivoima sistema.

- **Arhitektura servisa:** Mikroservisna ili servisno orijentirana arhitektura backend-a omogućit će nezavisno skaliranje pojedinačnih servisa (npr. servis za narudžbe, servis za proizvode).
- **Fleksibilnost:** Sistemske postavke, poput cijena, bit će dizajnirane imajući na umu skalabilnost i buduća proširenja.

5.6 Performanse sistema

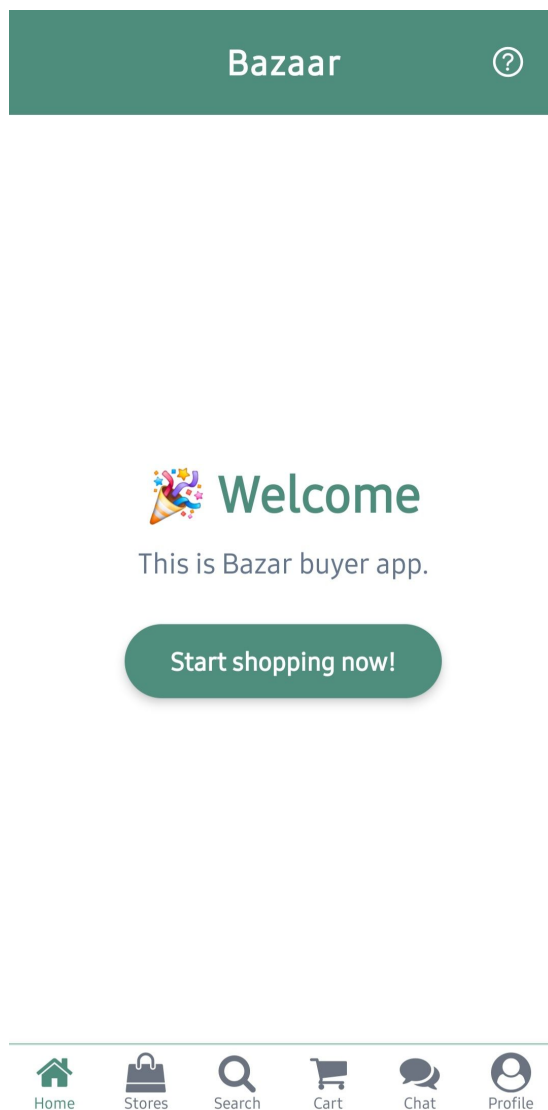
Sve komponente sistema "Bazaar" moraju pružati brz odziv i efikasno izvršavanje korisničkih akcija.

- **Vrijeme odziva aplikacija:**
 - Mobilne aplikacije (Bazaar i The Shop): Većina korisničkih interakcija (npr. učitavanje lista proizvoda, otvaranje detalja proizvoda, dodavanje u korpu, potvrda narudžbe, ažuriranje statusa narudžbe) treba se izvršiti unutar 2-3 sekunde.
 - Web administratorski portal: Učitavanje stranica i izvršavanje administrativnih akcija (npr. pretraga korisnika, upravljanje narudžbama) treba biti brzo i efikasno, s ciljanim vremenom odziva ispod 3 sekunde za većinu operacija.
- **Backend servisi:** API pozivi trebaju imati prosječno vrijeme odziva ispod 500ms pod normalnim opterećenjem.
- **Pretraga i filtriranje:** Pretraga proizvoda i filtriranje po različitim kriterijima (lokacija, cijena, kategorija) moraju biti brzi i vraćati relevantne rezultate u kratkom vremenskom roku.
- **Upravljanje zalihama:** Ažuriranje i provjera stanja zaliha moraju biti blizu realnog vremena kako bi se spriječile narudžbe nedostupnih proizvoda.
- **Optimizacija slika:** Slike proizvoda bit će optimizirane za brzo učitavanje na mobilnim uređajima bez značajnog gubitka kvalitete.
- **Istovremeni korisnici:** Sistem treba biti sposoban da podrži očekivani broj istovremenih korisnika bez degradacije performansi.

6 FURPS+

6.1 Funkcionalnosti

Aplikacija *Bazaar Marketplace* predstavlja sveobuhvatan sistem za digitalnu trgovinu, koji omogućava jednostavno, efikasno i sigurno povezivanje između kupaca, prodavača i administratora. Sistem je dizajniran kao distribuirana platforma koja funkcioniše putem mobilnih i web aplikacija, a podržava tri jasno definisane korisničke uloge: kupac, prodavač i administrator. Svaka od ovih uloga posjeduje specifične funkcionalnosti, prilagođene potrebama krajnjih korisnika i ciljevima sistema.





Slika 22. Prikaz Bazaar aplikacije (za kupce)

Kupci koriste mobilnu aplikaciju *Bazaar*, dostupnu na Android i iOS operativnim sistemima,

koja omogućava registraciju i prijavu putem više metoda (email, Google, Facebook). Aplikacija omogućava intuitivnu pretragu i filtriranje proizvoda po različitim kriterijima, pregled detalja artikala, dodavanje proizvoda u korpu, te kompletiranje procesa narudžbe uz praćenje njenog statusa u realnom vremenu. Dodatno, kupci mogu koristiti više adresa za dostavu, komunicirati s prodavačima putem direktnog chata, učestvovati u programu lojalnosti kroz prikupljanje i korištenje bodova, te ostavljati recenzije i ocjene za proizvode i prodavnice. Kroz aplikaciju se također mogu obratiti korisničkoj podršci putem sistema tiketa ili direktnog razgovora s administratorima.

20:52



Dobrodošli u The Shop

Prijavite se kako biste počeli prodavati


Email adresa*


Lozinka*

Nastavi

Nemate račun? [Registruj se](#)

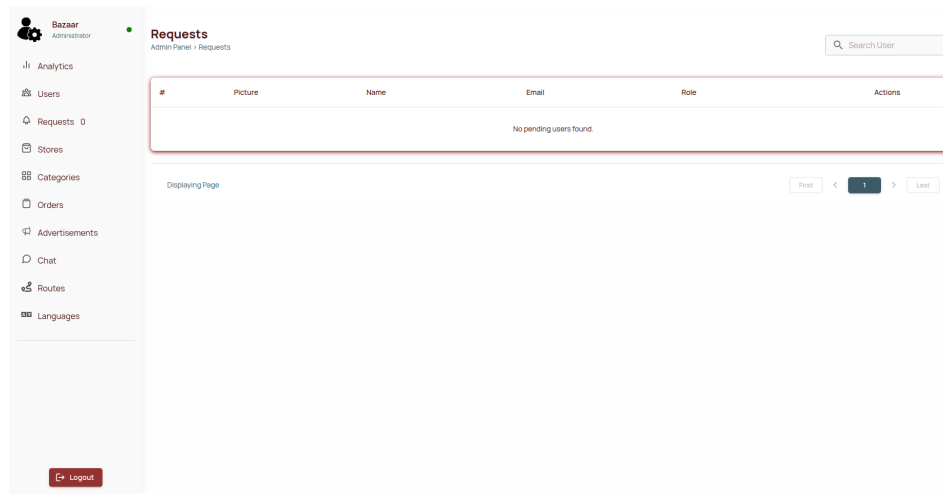
ILI

 Prijava putem Google-a

 Prijava putem Facebook-a

Slika 23. Prikaz The Shop aplikacije (za prodavače)

Prodavači imaju na raspolaganju namjensku mobilnu aplikaciju pod nazivom *The Shop*, putem koje upravljaju cjelokupnim poslovanjem svoje prodavnice. Aplikacija omogućava dodavanje, uređivanje i uklanjanje proizvoda, upravljanje zalihama i cijenama, kao i ažuriranje statusa narudžbi. Prodavači imaju mogućnost aktivne komunikacije s kupcima, odgovaranja na recenzije i podešavanja loyalty programa za svoje proizvode. Sistem im omogućava fleksibilno upravljanje logistikom, uključujući mogućnost ručnog upravljanja isporukama ili zakazivanja dostave putem eksternih kurirskih servisa, uz prikaz optimalnih ruta dostave kroz Google Maps integraciju.



Slika 24. Prikaz Admin panela

Administratori pristupaju sistemu putem sigurnog web portala, razvijenog korišćenjem savremenih tehnologija. Kroz administrativni interfejs, imaju mogućnost upravljanja korisnicima (verifikacija, deaktivacija, grupisanje po ulogama), prodavnicama i proizvodima (uključujući masovni unos putem Excel/CSV datoteka), kao i kompletnim procesom naručivanja. Takođe, upravljaju oglasnim kampanjama, loyalty sistemom, dostavnim logistikama i naplatnim modelima za prodavače. Administratori imaju pristup detaljnoj sistemskoj analitici koja uključuje praćenje performansi prodaje, korisničkih aktivnosti i efikasnosti marketinga.

Funkcionalnosti sistema dodatno su unaprijeđene implementacijom real-time mehanizama putem SignalR tehnologije, sigurnom komunikacijom kroz HTTPS protokol i autentifikacijom baziranom na JWT tokenima. Sistem omogućava integraciju s eksternim servisima, kao što su servisi za mapiranje i slanje notifikacija, čime se postiže visok nivo automatizacije i korisničkog iskustva. Cijeli sistem je skalabilan, podržava višejezičnost, i spreman je za rad u okruženju sa velikim brojem korisnika, proizvoda i transakcija.

6.2 Upotrebljivost

Prema ISO standardu, upotrebljivost se definiše kao:

“U kojoj mjeri određeni korisnici mogu koristiti proizvod za postizanje određenih ciljeva s učinkovitošću i zadovoljstvom u određenom kontekstu upotrebe.”

Aplikacija *Bazaar* razvijena je s ciljem da korisnicima omogući jednostavno i intuitivno

korištenje bez potrebe za prethodnim tehničkim znanjem. Interfejsi mobilnih i web aplikacija dizajnirani su prema savremenim standardima, uz jasnu navigaciju, preglednu strukturu i poznate obrasce korisničkog iskustva.

Validacije unosa, povratne informacije i vizuelna konzistentnost doprinose efikasnijoj upotrebi sistema i smanjenju grešaka. Poseban fokus je stavljen na jasnoću i funkcionalnost, čime se korisnicima omogućava brzo snalaženje i efikasno izvršavanje zadataka.

6.3 Pouzdanost

Prema ANSI standardu, pouzdanost softverskog sistema predstavlja “vjerovatnoću da softver radi bez grešaka tokom određenog vremenskog perioda u definisanom okruženju”. Platforma *Bazaar* je projektovana sa ciljem postizanja visoke stabilnosti i dostupnosti, oslanjajući se na fundamentalne principe softverskog inženjeringa i odabir zrelih tehnologija.

Na nivou aplikacije, pouzdanost se osigurava kroz nekoliko ključnih mehanizama. Backend sistem, razvijen u **ASP.NET Core**, implementira **strukturirano upravljanje greškama (exception handling)**, što sprečava da neočekivani problemi dovedu do pada čitavog servisa. Svi ulazni podaci od korisnika podliježu **strogoj validaciji** na serverskoj strani, čime se štiti integritet sistema i sprečava upis nekonzistentnih podataka u bazu. Korištenje **transakcijskog mehanizma** u radu sa bazom podataka garantuje da će složene operacije koje zahtijevaju više upisa biti izvršene u potpunosti ili nikako (atomicity), čuvajući time konzistentnost podataka.

Pouzdanost na nivou podataka dodatno je ojačana korištenjem **PostgreSQL** baze podataka, koja je poznata po svojoj robusnosti, stabilnosti i striktnoj **ACID** (Atomicity, Consistency, Isolation, Durability) saglasnosti. Korištenje Docker kontejnera za pokretanje baze podataka osigurava konzistentno i izolovano okruženje, što smanjuje rizik od problema uzrokovanih razlikama između razvojne i produkcione infrastrukture.

Za osiguranje dugoročne pouzdanosti i mogućnosti oporavka od katastrofalnih grešaka, za produkcijsko okruženje se postavlja zahtjev za implementacijom **strategije automatskog i periodičnog backupa baze podataka**. Također, za kontrolisano puštanje novih verzija i proaktivno praćenje stanja sistema, preporučuje se uspostavljanje **CI/CD procesa** i integracija **alata za monitoring i centralizovano logiranje**. Ovi mehanizmi trenutno nisu implementirani, ali predstavljaju ključan korak u evoluciji sistema ka maksimalnoj pouzdanosti.

6.4 Performanse

Performanse sistema *Bazaar* su temeljene na pažljivom odabiru tehnologija i arhitekturnih principa koji favorizuju brzinu i efikasnost. Backend aplikacija, razvijena na platformi **.NET 8**, koristi prednosti modernog, visoko-performantnog runtime-a. Njegov **just-in-time (JIT)** kompajler i efikasan garbage collector osiguravaju brzu obradu zahtjeva, dok se intenzivno korištenje **asinhronih programskih modela (async/await)** koristi za efikasno upravljanje I/O operacijama, čime se oslobađaju resursi servera i omogućava bolja podrška za konkurentni rad većeg broja korisnika.

Na nivou baze podataka, **PostgreSQL** pruža robusnu osnovu za brzo izvršavanje upita. Performanse su podržane strukturom baze gdje su primarni ključevi automatski indeksirani. Za ključne tabele i upite koji zahtijevaju često filtriranje i pretragu, implementirani su i dodatni, **specifični indeksi** kako bi se ubrzao dohvat podataka. **Entity Framework Core**

se koristi za prevođenje LINQ izraza u efikasne SQL upite, smanjujući potrebu za pisanjem kompleksnog, ručno optimizovanog SQL-a.

Za komunikaciju u realnom vremenu, specifično za funkcionalnost **chata**, koristi se **SignalR**. Ova tehnologija uspostavlja trajnu, nisko-latentnu konekciju između klijenta i servera, primarno putem **WebSocket** protokola. Na taj način se izbjegava overhead konstantnog HTTP pozivanja (polling), što rezultira trenutnom isporukom poruka i visokom responzivnošću. Važno je napomenuti da se za push notifikacije koriste namjenski servisi (FCM i APN), čime se postiže efikasna isporuka na mobilne uređaje bez nepotrebnog opterećenja backend servera.

6.5 Podržanost

Podržanost sistema *Bazaar* odnosi se na lakoću njegovog održavanja, testiranja i budućeg proširenja. Arhitektura sistema je dizajnirana sa modularnošću kao ključnim principom kako bi se olakšale izmjene i dodavanje novih funkcionalnosti bez narušavanja postojećih. Backend aplikacija prati jasnu projektnu strukturu sa razdvojenim slojevima odgovornosti (npr. sloj za API, servisni sloj, sloj za pristup podacima), što omogućava programerima da lakše lociraju kod i razumiju logiku sistema.

Ključan element koji doprinosi podržanosti je **automatski generisana API dokumentacija putem Swagger-a (OpenAPI)**. Svaka API ruta je detaljno opisana, uključujući obavezne parametre, primjere zahtjeva i moguće odgovore. Ovo ne samo da služi kao "živa" dokumentacija, već drastično olakšava testiranje i integraciju sa klijentskim aplikacijama, smanjujući mogućnost grešaka u komunikaciji između frontenda i backenda.

Sistem je dizajniran za prenosivost i konzistentnost. Backend aplikacija se može pokrenuti na **Linux i Windows okruženjima**. Da bi se osigurala uniformnost razvojnog okruženja, PostgreSQL baza podataka se pokreće unutar **Docker kontejnera**, čime se eliminišu problemi uzrokovani razlikama u lokalnim konfiguracijama. Za mobilne aplikacije, proces buildanja i distribucije je standardizovan i pojednostavljen korištenjem alata kao što su **Expo CLI i EAS Build**, što omogućava brzu i pouzdanu isporuku novih verzija. Konfiguracijski podaci sistema, poput konekcijskih stringova za bazu, odvojeni su od koda i upravljaju se putem **appsettings.json** fajlova, što omogućava laku promjenu postavki bez potrebe za ponovnim kompajliranjem aplikacije.

6.6 Implementacijski zahtjevi

Za uspješnu implementaciju, razvoj i održavanje sistema, neophodno je ispuniti niz softverskih i infrastrukturnih zahtjeva. Tehnološki stek je odabran tako da osigura skalabilnost, sigurnost i efikasnost, a čine ga moderne i provjerene tehnologije.

Zahtjevi za razvojno okruženje uključuju instalaciju **.NET 8 SDK** za razvoj backend servisa, uz preporučeno okruženje poput Visual Studija ili VS Code-a sa C# ekstenzijama. Za frontend komponente, neophodan je **Node.js** sa **npm** ili **yarn** package managerom za upravljanje zavisnostima React Native i React (Vite) projekata. Razvoj mobilnih aplikacija dodatno zahtijeva **Android Studio** (sa Android SDK) i/ili **Xcode** (sa iOS SDK). Lokalna instanca **PostgreSQL** baze podataka je neophodna za testiranje, a cjelokupan izvorni kod se upravlja putem **Git** sistema za verzionisanje.

Zahtjevi za produkciono (hosting) okruženje su strožiji. Backend aplikacija zahtijeva server sa instaliranim **.NET 8 Runtime**-om, sposoban da pokreće ASP.NET Core aplikacije (npr. na Linux serveru uz Nginx kao reverse proxy, ili na Windows Serveru uz IIS). Centralizovana baza podataka mora biti produkciona instanca **PostgreSQL** servera. Za sigurnu komunikaciju između klijenta i servera, neophodna je nabavka i ispravna konfiguracija **SSL/TLS certifikata** kako bi se omogućio **HTTPS** protokol. Autentifikacija korisnika se oslanja na **JWT**, što nameće zahtjev za sigurnim mehanizmom pohrane i upravljanja tajnim ključem (npr. putem environment varijabli ili namjenskog "secrets managera").

6.7 Ograničenja Interfejsa

Sistem se za isporuku ključnih funkcionalnosti oslanja na veći broj vanjskih servisa, pri čemu svaka integracija nameće specifična ograničenja koja se moraju poštovati. Za autentifikaciju korisnika putem Google i Facebook naloga, aplikacija koristi **OAuth 2.0** protokol. Ovo zahtijeva sigurnu pohranu i upravljanje API ključevima (Client ID i Client Secret) te sposobnost backend sistema da ispravno procesuirá povratne (callback) pozive od provajdera. Pored tehničkih, sistem se mora pridržavati i pravila o korištenju podataka i privatnosti koje propisuju Google i Facebook.

Interakcija sa korisnicima putem notifikacija odvija se na dva načina. Za slanje push notifikacija koriste se Firebase Cloud Messaging (FCM) za Android i Apple Push Notification service (APN) za iOS. Backend server stoga mora imati dozvoljenu odlaznu mrežnu konekciju prema Google i Apple serverima. Autentifikacija zahtijeva sigurno upravljanje serverskim ključevima i certifikatima, a sam sadržaj notifikacije (payload) podliježe strogim ograničenjima u pogledu formata i veličine. Za slanje email poruka, poput kodova za reset lozinke, koristi se Google SMTP servis. Ovo podrazumijeva odlaznu konekciju na portu **587 (STARTTLS)** i autentifikaciju pomoću validnih kredencijala. Ključno ograničenje ovog servisa je dnevni limit broja poslanih poruka (kvota), što predstavlja potencijalno usko grlo za skalabilnost aplikacije.

Geolokacijske funkcionalnosti, kao što su prikaz ruta i pretraga mjesta, implementirane su pomoću Google Maps API-ja (Routes & Places). Korištenje ovih servisa uslovljeno je posjedovanjem API ključa koji mora biti sigurno pohranjen i ograničen na specifične domene radi zaštite. Važno ograničenje predstavlja model naplate i limiti broja poziva (quotas); prekoračenje besplatnog nivoa korištenja može generisati značajne finansijske troškove.

Konačno, za komunikaciju u realnom vremenu između klijenta i servera koristi se SignalR. Optimalan rad ovog servisa zavisi o podršci za **WebSocket** protokol od strane cjelokupne infrastrukture (server, klijent, firewall). U slučaju da su WebSockets blokirani, performanse se degradiraju korištenjem alternativnih mehanizama poput Long Pollinga. Pored toga, SignalR održava trajne (persistent) konekcije, što direktno utiče na potrošnju serverske memorije i CPU resursa i predstavlja ključni faktor pri planiranju skaliranja sistema.

6.8 Fizički zahtjevi

Ovaj odjeljak definiše minimalne hardverske, softverske i mrežne preduvjete potrebne za uspješan rad svih komponenti sistema.

6.8.1 Serverska Infrastruktura

Aplikativni backend, web aplikacija i baza podataka su dizajnirani za hostovanje na skalabilnoj cloud infrastrukturi (npr. AWS EC2), ali su kompatibilni i sa drugim virtualizovanim ili fizičkim serverskim okruženjima. Fleksibilnost konfiguracije omogućava da se ove komponente hostuju na istom serveru ili na odvojenim mašinama, zavisno od zahtjeva za skalabilnošću i izolacijom.

- **Backend Server (API):**
 - **Runtime Okruženje:** .NET 8.0.
 - **Operativni Sistem:** Windows Server (npr. 2019+), Linux distribucije (npr. Ubuntu Server LTS, RHEL/CentOS) ili bilo koji OS sposoban za pokretanje .NET runtime-a. Iako trenutno nije kontejnerizovan, backend se može adaptirati za Docker deployment.
- **Web Server (React Aplikacija):**
 - **Softver:** Standardni web server (npr. Nginx, Apache, IIS) za serviranje statičkih React aplikacija.
 - **Operativni Sistem:** Bilo koji OS koji podržava odabrani web server (Linux distribucije, Windows Server).
- **Baza Podataka Server:**
 - **Tip Baze Podataka:** PostgreSQL verzija 15.x ili novija.
 - **Hostovanje:** Može biti instalirana direktno na server operativnom sistemu ili kontejnerizovana putem **Docker-a**.
- **Portovi (za izložene servise):**
 - Standardni HTTP port **80** i HTTPS port **443** moraju biti omogućeni za pristup backend API servisima i React web aplikaciji (bilo na istom ili odvojenim serverima).
 - Za internu komunikaciju sa bazom podataka, port **5432** mora biti dostupan unutar serverske mreže (npr. Docker bridge mreža ili privatna subnet), ali ne mora biti javno izložen.
- **Minimalni hardverski resursi (za početni deployment):** Preporučuje se ukupno minimalno 2 CPU jezgra, 4 GB RAM-a i 50 GB SSD skladišnog prostora za sve serverske komponente (Backend, Web, Baza Podataka) kada su hostovani zajedno. Ako su komponente razdvojene, pojedinačni zahtjevi će varirati.

6.8.2 Klijentske Aplikacije

Sistem se sastoji od web aplikacije (administratorski panel) i dvije mobilne aplikacije. Fizički zahtjevi za klijentske uređaje su sljedeći:

Web Aplikacija (Administratorski Panel) Web aplikacija je razvijena korišćenjem **React** tehnologije i zahtijeva pristup putem standardnog web preglednika.

- **Web Preglednici:** Posljednje dvije stabilne verzije Google Chrome, Mozilla Firefox, Microsoft Edge i Apple Safari.

- **Operativni Sistem:** Bilo koji desktop ili laptop operativni sistem (Windows 10+, macOS 11+, savremene Linux distribucije) sa podrškom za navedene web preglednike.
- **Minimalna Rezolucija Ekrana:** Preporučena rezolucija od 1280x768 piksela za optimalan prikaz i interakciju.

Mobilne Aplikacije Dvije mobilne aplikacije su razvijene kao **multiplatformske (React Native)** aplikacije.

- **Mobilni Operativni Sistem:**
 - **Android:** Verzija 8.0 (API Level 26 - Oreo) ili novija.
 - **iOS:** Verzija 15.0 ili novija.
- **Hardver Uređaja:** Bilo koji pametni telefon ili tablet sa minimalno 2GB RAM-a i procesorom srednje klase ili boljim, sposoban za pokretanje navedenih verzija operativnih sistema.
- **Slobodan Skladišni Prostor:** Minimalno 250 MB slobodnog prostora za instalaciju i keširanje podataka aplikacije.

6.8.3 Mrežna Povezanost

- Stabilna internet konekcija je neophodna za sve klijentske aplikacije (web i mobilne) kako bi se osigurala kontinuirana komunikacija sa backend sistemom.
- Preporučena minimalna brzina interneta za klijentske uređaje je 5 Mbps za ugodno korištenje.

6.9 Ograničenja dizajna

Tokom razvoja sistema susreli smo se s nekoliko tipičnih dizajnerskih ograničenja:

- Vremensko ograničenje: Svaki sprint imao je definisan rok, a krajnji rok za završetak projekta bio je unaprijed utvrđen.
- Tehnološka ograničenja: Na početku projekta je odlučeno da se koristi isključivo .NET 8 za backend i React/React Native za frontend komponente.
- Sastav tima: Članovi razvojnog tima dodijeljeni su nasumično, što je značilo da nije bilo prethodnog timskog iskustva među članovima, pa je trebalo dodatno vrijeme za uigravanje i koordinaciju.
- UI/UX standardi: Interfejs je morao biti konzistentan kroz pojedinačne aplikacije, te jednostavan i intuitivan za korištenje. Prilikom dizajniranja korisničkog interfejsa za sve aplikacije u sistemu, trebao je biti primijenjen princip segmentacije i fokusiranog prikaza za kompleksne interakcije i detaljne informacije. To znači da se manji, specifični dijelovi aplikacije koji zahtijevaju korisnički unos, prikaz detalja ili potvrdu akcije, ne prikazuju direktno unutar glavnog prikaza, već u izolovanim, namjenski dizajniranim kontekstima.