Niir

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 04.04.2023. | Nacrt | Inicijalna verzija | Dušan |
| 08.04.2023. | 1.0 | Dopunjena i prerađena verzija | Veljko |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Cilj dokumenta 5

2. Opseg dokumenta 5

3. Reference 5

4. Predstavljanje arhitekture 5

5. Ciljevi i ograničenja arhitekture 5

6. Pogled na slučajeve korišćenja 5

6.1 Dijagrami slučajeva korišćenja 6

6.2 Kratak opis slučajeva korišćenja 8

6.2.1 Izmena teme 8

6.2.2 Pregled artikla 8

6.2.3 Dodatne informacije o artiklu 8

6.2.4 Mogućnost plaćanja 8

6.2.5 Poručivanje artikla 8

6.2.6 Kreiranje potrošačke korpe 8

6.2.7 Prijavljivanje 8

6.2.8 Izmena potrošačkog artikla 8

6.2.9 Dodavanje inventara 8

6.2.10 Brisanje trenutne slike artikla 8

6.2.11 Ažuriranje slike artikla 8

6.2.12 Dodavanje artikla u bazu podataka 8

6.2.13 Brisanje artikla 8

7. Pogled na logičku arhitekturu sistema 9

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 9

7.1.1 Korisnički interfejs 9

7.1.2 Aplikaciona logika 9

7.1.3 Pristup podacima 10

7.1.4 HTML 10

7.1.5 React tsx 10

7.1.6 SQLite 10

8. Pogled na procese 10

8.1 Procesi 10

8.1.1 Web čitač 11

8.1.2 Web server 11

8.1.3 TSX-CGI 11

8.1.4 SQLite Server 11

9. Pogled na raspoređivanje sistema 12

9.1 Klijent 12

9.2 Web server 12

9.3 DBMS server 13

10. Pogled na implementaciju sistema 13

10.1 Model domena 13

10.2 Šema baze podataka 13

10.3 Komponente sistema 13

10.3.1 Komponente korisničkog interfejsa 13

10.3.2 Komponente aplikacione logike 13

10.3.3 Komponente za pristup podacima 13

11. Performanse 13

12. Kvalitet 13

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Niir portala.

# Opseg dokumenta

Dokument se odnosi na Niir web aplikaciju koja će biti razvijena od strane WD tima. Niir predstavlja skraćenicu za blizinu i nagoveštaj da kupovina namirnica može biti bliže dostavljena krajnjim kupcima. Namena sistema je efikasan odabir, priprema, poručivanje i na kraju dostavljanje željenih artikala.

# Reference

Spisak korišćene literature:

1. Niir – Predlog projekta, WD-Niir-01, V1.0, 2023, WD tim.
2. Niir – Planirani raspored aktivnosti na projektu, V1.0, 2023, WD tim.
3. Niir – Plan realizacije projekta, V1.0, 2023, WD tim.
4. Niir – Vizija sistema, V1.0, 2023, WD tim.
5. Niir – Specifikacija zahteva, V1.0, 2023, SWETeam.

# Predstavljanje arhitekture

Arhitektura sistema u dokumentu je prikazana kao serija pogleda na sistem: pogled na slučajeve korišćenja, pogled na logičku arhitekturu sistema, pogled na procese, pogled na razmeštaj komponenti sistema i pogled na implementaciju. Ovi pogledi su predstavljeni odgovarajućim UML dijagramima.

# Ciljevi i ograničenja arhitekture

Ključni zahtevi i sistemska ograničenja koja imaju značajan uticaj na izbor arhitekture i projektovanje sistema su:

1. Niir web aplikacija će biti implementirana kao Web aplikacija zasnovana na React skripting jeziku i SQLite bazi podataka [4].
2. Klijentski deo Niir web aplikacije će biti optimizovan za sledeće Web čitače: Internet Explorer 6.0 i noviji, Opera 8.0 i noviji, kao i Firefox (Mozilla) [4].
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.
4. Klasifikaciju publikacija treba standardizovati sa postojećom tipologijom propisanom od strane Ministarstva za nauku.

# Pogled na slučajeve korišćenja

U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva [5].

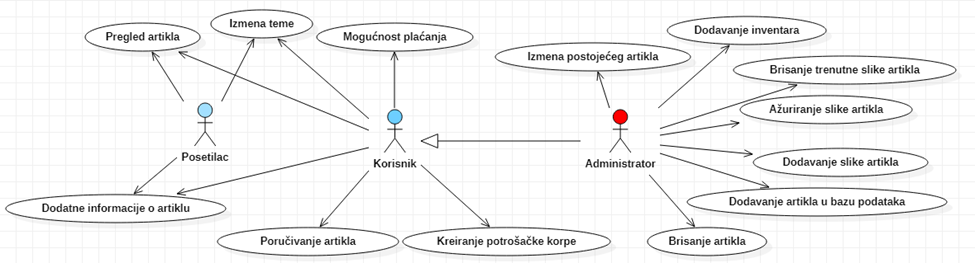
Slučajevi korišćenja Niir web aplikacije su:

* Izmena teme
* *Pregled artikla*
  + Dodatne informacije o artiklu
* Mogućnost plaćanja
* Poručivanje artikla
* Kreiranje potrošačke korpe
* Izmena postojećeg artikla
* Dodavanje inventara
* Brisanje trenutne slika artikala
* Ažuriranje slike artikala
* Dodavanje artikla u bazu podataka
* Brisanje artikla
* Prijavljivanje

Ove slučajevi korišćenja mogu da iniciraju posetilac, korisnik i administrator.

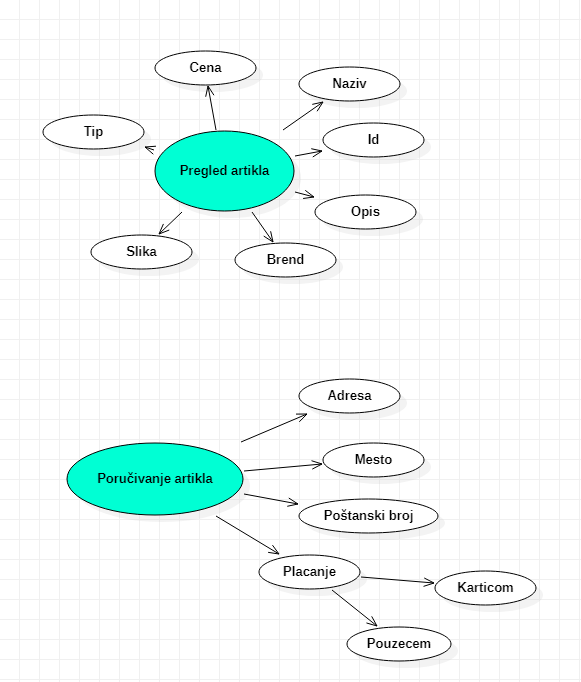
## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja PeNcIL portala prikazan je na sledećoj slici:



Slučajevi korišćenja *pregled artikla* i *poručivanje artikla* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

Detaljni UML dijagram za slučaj korišćenja *pregled informacija* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Izmena teme

Izbor između svetle i tamne teme na web portalu. Akteri mogu biti posetilac i korisnik.

### Pregled artikla

Prikaz velikog broja dostupnih artikala na web portalu. Akteri mogu biti posetilac i korisnik.

### Dodatne informacije o artiklu

Na odgovarajućoj route adresi web portala prikazuje se pregled osnovnih informacija o artiklu. Akteri mogu biti posetilac i korisnik.

### Mogućnost plaćanja

Prikaz stranice sa unosom svih potrebnih podataka o plaćanju. Akteri mogu biti korisnici samo.

### Poručivanje artikla

Kratak opis: Prikaz stranice sa spiskom svih proizvoda i mogućnosti popunjavanja adrese i ostalih ličnih informacija. Akteri mogu biti korisnici samo.

### Kreiranje potrošačke korpe

Kratak opis: Stek skladište koje omogućava čuvanje svih prethodno dodatih artikala. Akteri mogu biti samo korisnici.

### Prijavljivanje

Kratak opis: Prijavljivanje korisnika na portal u cilju pristupa specifičnim funkcijama koje zahtevaju autorizaciju.

Akteri mogu biti Korisnik i Administrator.

### Izmena postrojećeg artikla

Kratak opis: Funkcionalnost mogućnosti izmene postojećeg artikla tj. osnovnih informacija o njemu.

Akteri mogu biti administratori.

### Dodavanje inventara

Kratak opis: Izručivanje izveštaja o trenutnom stanju baze podataka.

Akteri koji mogu učestvovati su Administratori.

### Brisanje trenutne slike artikala

Kratak opis: Izmena slike (fotografije) artikala koji je u opticaju.

Akteri koji iniciraju slučaj korišćenja su Administratori.

### Ažuriranje slike artikala

Kratak opis: Izmena slike artikala koji je prethodno izabran.

Akteri koji mogu učestvovati su Administratori.

### Dodavanje artikla u bazu podataka

Kratak opis: Mogućnost dodavanja novih artikala u bazu podataka čineći ih vidljivim na katalogu svih artikala.

Akteri koji učestvuju su Administratori.

### Brisanje artikla

Kratak opis: Brisanje postojećeg artikla iz kataloga svih artikala..

Akteri koji učestvuju su Administratori.

# Pogled na logičku arhitekturu sistema

U ovom odeljku je dat pregled logičke arhitekture sistema. Ovaj pogled sadrži opis najznačajnijih klasa, njihove organizacije u pakete i podsisteme, i organizacija podsistema u slojeve. U cilju opisivanja dinamičkih aspekata arhitekture, ovaj odeljak može da uključi opise realizacije najznačajnijih slučajeva korišćenja. Da bi se ilustrovala veza između arhitekturno značajnih klasa, podsistema, paketa ili slojeva moguće je uključiti i odgovarajuće dijagrame klasa.

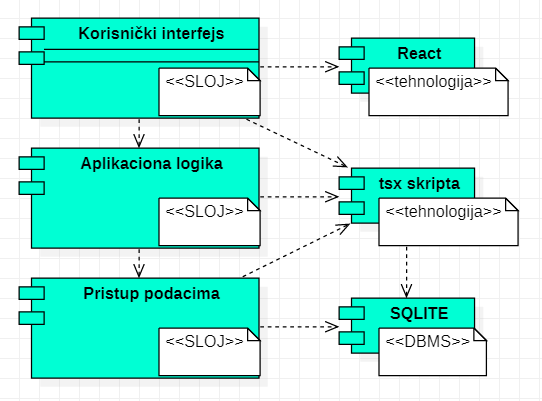
Logički pogled na Nirr web aplikaciju obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice, React tsx skripte i multimedijalni sadržaj koji realizuju grafički dizajn i forme preko kojih korisnici sistema komuniciraju sa sistemom.

Paket *Aplikaciona logika* predstavlja srednji sloj sistema koji sadrži React tsx skripte zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži React tsx skripte koje predstavljaju interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

sloj

Ovaj sloj realizuje korisnički interfejs portala. U njemu su sadržane sve HTML, multimedijalni sadržaji i tsx skripte koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML i tsx.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Niir web aplikacije. Sadrži tsx skripte koje realizuju funkcionalnost karakterističnu za domen primene portala i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima i tsx paketa.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži React tsx skripte zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u SQLite bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa React tsx i SQLite baza podataka.

### HTML

tehnologija

Tehnologija HTML-a definiše gradivne elemente stranica koje se prikazuju u Web čitaču i koje omogućavaju prikaz formatiranih informacija i realizaciju formi za unos i ažuriranje podataka.

### React tsx

tehnologija

Tehnologija React tsx-a obezbeđuje mehanizam za pisanje i izvršavanje skripti na strani servera. Ove skripte mogu da generišu typescript kod koji realizuje korisnički interfejs i pristupaju bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

### SQLite

DBMS

SQLite predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju Niir web aplikacije.

# Pogled na procese

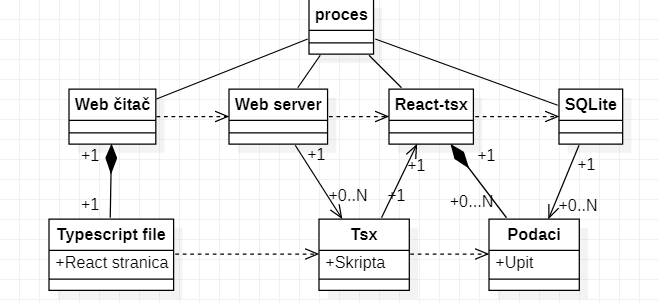
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe spada u opis procesne arhitekture.

Web aplikacije zasnovane na PHP-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta PHP Web aplikacije nije potrebno voditi računa o načinu rada Web servera i načinu izvršavanja skripti.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje Niir web portala kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Niir portala. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na tsx-u i SQLite bazi podataka.



### Web čitač

Web čitač je proces koji izvršava funkcionalnost aplikacije za prikaz Tsx stranica dobijenih od nekog Web servera. U najopštijem slučaju Web čitač u jednom trenutku može da prikazuje samo jednu React tsx stranicu.

Web čitač zavisi od Web servera koji generiše i vraća odgovarajuću tsx stranicu na zahtev.

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača. Ukoliko je zahtevana stranica tsx skript, Web server inicira izvršenje TSX-CGI procesa koji obrađuje odgovarajući skript i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj TSX-CGI procesa.

### TSX-CGI

TSX-CGI proces obavlja posao obrade zadatog Tsx skripta i generiše odgovarajući tekstualni sadržaj koji Web server šalje Web čitaču. Za izvršenje Tsx skripta ovaj proces može da zahteva usluge SQLite servera-a. Komunikacija između TSX-CGI procesa i SQLite servera se obavlja preko prosleđivanja upita i vraćanja rezultat.

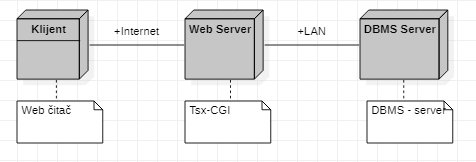
### SQLite Server

SQLite Server je proces koji izvršava funkcionalnost SQLite sistema za upravljanje bazama podataka. Ovaj proces može konkurentno da prihvati određen broj upita, izvrši ih nad bazom podataka i vrati rezultate procesu koji je upite postavio.

# Pogled na raspoređivanje sistema

Pogled na raspoređivanje sistema prikazuje različite fizičke čvorove za najopštiju konfiguraciju sistema. Fizičkim čvorovima koji predstavljaju procesore vrši se dodeljivanje identifikovanih procesa.

Na sledećoj slici dat je UML dijagram raspoređivanja Niir web aplikacije.



## Klijent

Pristup Niir web aplikaciji se obavlja preko klijentskih računara na kojima se izvršava Web čitač. Za povezivanje između klijenta i Web servera koristi se Internet infrastruktura tako da nema ograničenja u pogledu lokacije klijenta.

## Web server

Računar na kome se izvršava Web server opslužuje više klijenata koji pristupaju preko Interneta. Pored osnovnog procesa koji realizuje funkcionalnost Web servera, na ovom računaru mogu da se izvršavaju i procesi TSX-CGI koji vrše obradu zadatih TSX skripti. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom u lokalnoj mreži (LAN).

## DBMS server

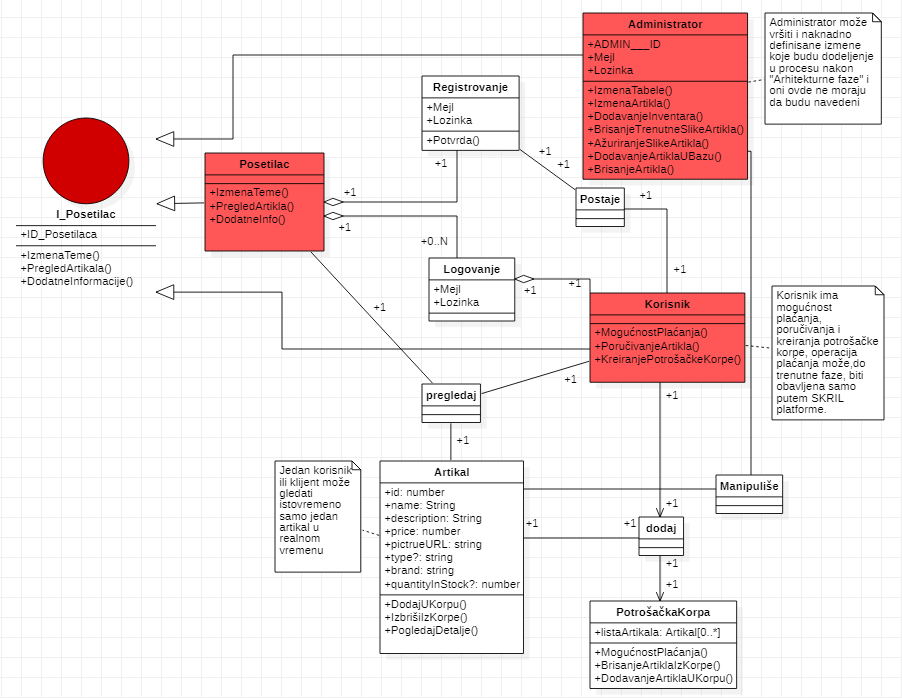
DBMS server je računar na kome se izvršava SQLite Server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka. Zbog sigurnosti podataka koji se na ovom računaru čuvaju pristup bazi je ograničen samo na računare iz lokalne mreže (LAN).

# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju Niir web aplikacije ovaj odeljak sadrži šemu baze podataka i prikaz komponenti sistema razvrstanih u ranije identifikovane pakete.

## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem *STARUML-a*, dok je migracija na SQLite obavljena pomoću *SQLite Migration Toolkit*-a.



## Komponente sistema

*NAPOMENA: Niir projekat nije kompletno implementiran, tj. implementiran je na nivou arhitekturnog prototipa. Zbog ovoga sledeću sekciju trebate shvatiti kao ilustraciju kako bi dokument trebao da izgleda. Opis komponenti sistema za vaše projekte mora biti kompletan! Procena složenosti vašeg rešenja će zavisiti od opisa sistema koji ovde date.*

Komponente sistema Niir web aplikacije su TSX skripte čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je TSX skript prikazan kao klasa atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije definisane u okviru skripta.

### Komponente korisničkog interfejsa

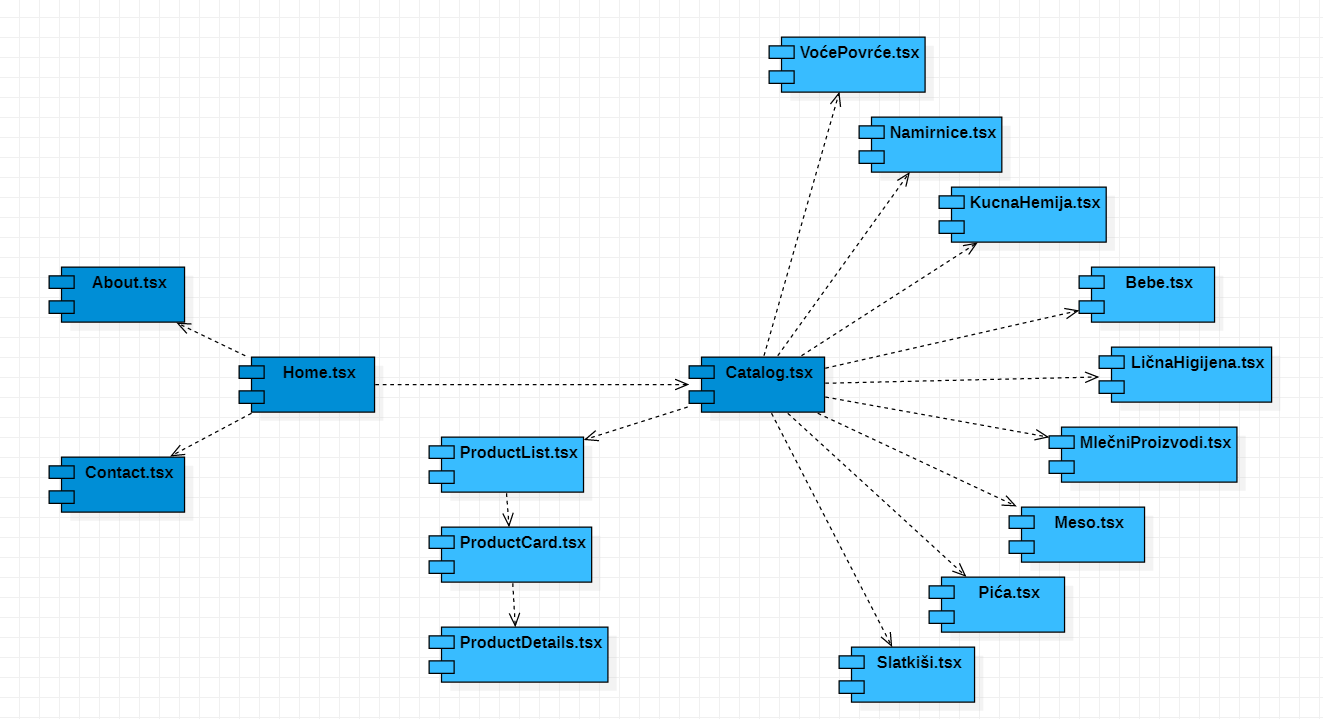
Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:

Komponenta **App.tsx** je implementira stranicu portala čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu.

Komponenta **material-ui** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na različitim stranicama.

### Komponente aplikacione logike

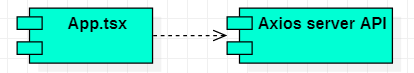
Komponente koje realizuju domen problema se uključuju isključivo preko **App.tsx** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u ovom skriptu. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



... opis svih komponenti sa dijagrama sa detaljnim dijagramima po potrebi.

### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru **Axios server** **poziva.** Ovaj poziv menja fatch funkciju u definisanom axios pozivu pomocu client.ts foldera gde je ostvarena direktna konekcija sa logikom baze podataka.



# Performanse

Izabrana arhitektura softvera podržava zahteve u pogledu broja korisnika koji mogu simultano pristupati sistemu i vremena odziva za pristup bazi podataka specificirane u zahtevima u pogledu performansi [5]:

1. Sistem će da podrži do 1000 simultanih pristupa korisnika web aplikaciji.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 5 sekundi.

Zahtevane performanse su zadovoljene izborom tehnologija na kojima će sistem biti razvijen i definisane hardverske platforme [5].

# Kvalitet

Izabrana arhitektura softvera podržava zahteve u pogledu dostupnosti i srednjeg vremena između otkaza specificirane u zahtevima u pogledu pouzdanosti [5]:

1. Niir web aplikacija će biti dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 10%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.