WinScout

Portal za kupovinu građevinske stolarije

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 14.04.2021. | 1.0 | Inicijalna verzija | Andrija Đorđević |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 Izbor jezika 8

6.2.2 Pregled sadržaja 8

6.2.3 Prikazivanje informacija edukativnog karaktera 8

6.2.4 Prikazivanje podataka o proizvodu 8

6.2.5 Upoređivanje proizvoda 8

6.2.6 Konvertovanje valuta 8

6.2.7 Ocenjivanje sadržaja 9

6.2.8 Dodavanje komentara 9

6.2.9 Brisanje komentara 9

6.2.10 Izmena komentara 9

6.2.11 Čuvanje sadržaja 9

6.2.12 Prikaz notifikacija 9

6.2.13 Prikazivanje podataka o prodavcu 9

6.2.14 Filtriranje 9

6.2.15 Pretraživanje 9

6.2.16 Dodavanje sadržaja 9

6.2.17 Ažuriranje sadržaja 9

6.2.18 Brisanje sadržaja 9

6.2.19 Registrovanje 10

6.2.20 Prijavljivanje 10

6.2.21 Ažuriranje sopstvenog naloga 10

7. Pogled na logičku arhitekturu sistema 10

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 11

7.1.1 Korisnički interfejs 11

7.1.2 Aplikaciona logika 11

7.1.3 Pristup podacima 11

7.1.4 HTML, ReactJS, BootstrapReact 11

7.1.5 ASP.NET Core 12

7.1.6 MS SQL Server 12

8. Pogled na procese 12

8.1 Procesi 12

8.1.1 Web čitač 12

8.1.2 IIS Web Server 13

8.1.3 ASP.NET HTTP Runtime 13

8.1.4 MS SQL Server 13

9. Pogled na raspoređivanje sistema 13

9.1 Klijent 13

9.2 Web server i DBMS 13

10. Pogled na implementaciju sistema 13

10.1 Model domena 14

10.2 Šema baze podataka 15

10.3 Komponente sistema 15

10.3.1 Komponente korisničkog interfejsa 15

10.3.2 Komponente aplikacione logike 18

11. Performanse 19

12. Kvalitet 19

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture WinScout portala.

# Opseg dokumenta

Dokument se odnosi na WinScout portal koji će biti razvijen od strane EESTeam-a. Namena sistema je efikasno prezentovanje, pretraživanje i objavljivanje sadržaja vezanih za kupovinu proizvoda građevinske stolarije.

# Reference

Spisak korišćene literature:

1. WinScout – Predlog projekta, EES-WinScout-01, V1.0, 2021, EESTeam.
2. WinScout – Vizija sistema, EES-WinScout-02, V1.0, 2021, EESTeam.
3. WinScout – Plan realizacije, EES-WinScout-03, V1.0, 2021, EESTeam.
4. WinScout – Raspored aktivnosti, V1.0, 2021, EESTeam.
5. WinScout – Specifikacija zahteva, EES-WinScout-04, V1.0, 2021, EESTeam.

# 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. WinScout portal će biti implementiran kao Web aplikacija zasnovana na REACT biblioteci, ASP.NET Core framework-u i MySQL bazi podataka.
2. Klijentski deo WinScout portala će biti optimizovan za sledeće Web čitače: Google Chrome, Opera 8.0 i noviji, kao i Firefox (Mozilla).
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.

# Pogled na slučajeve korišćenja

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

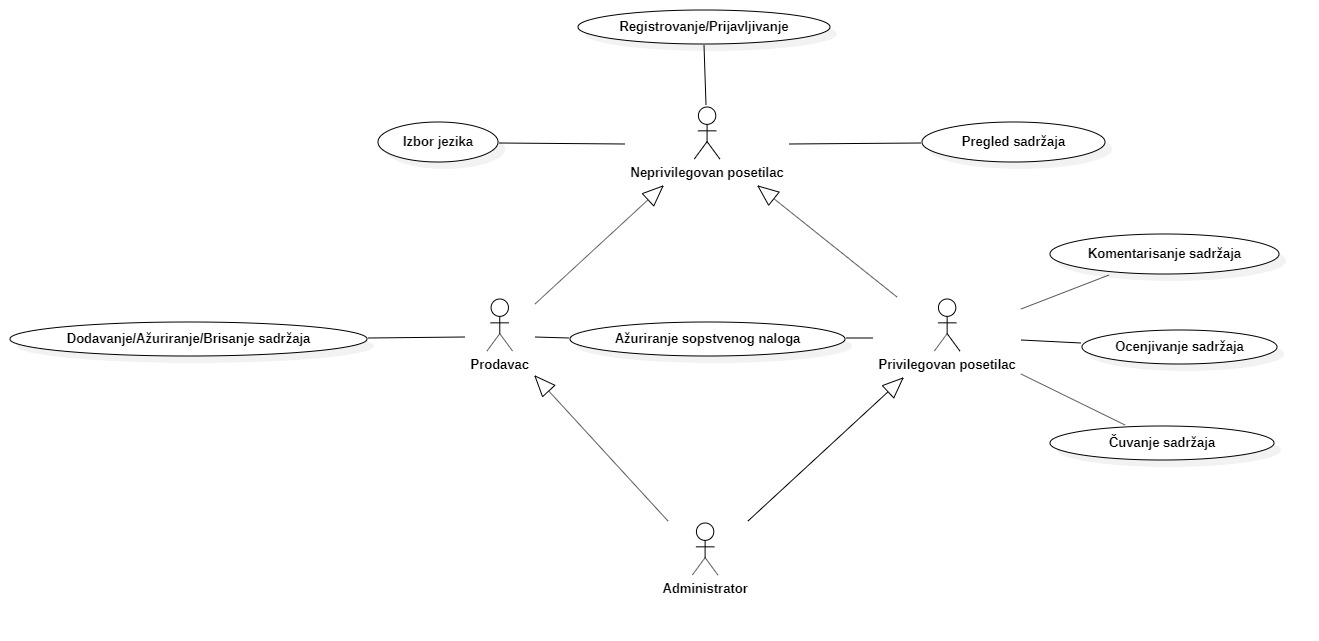
Slučajevi korišćenja WinScout portala su:

* Izbor jezika
* *Pregled sadržaja*
  + Prikazivanje informacija edukativnog karaktera
  + *Prikazivanje podataka o proizvodu*
    - Upoređivanje proizvoda
    - Konvertovanje valuta
    - Ocenjivanje sadržaja
    - Komentarisanje sadržaja
      * Dodavanje komentara
      * Izmena komentara
      * Brisanje komentara
    - Čuvanje sadržaja
  + Prikaz notifikacija
  + Prikazivanje podataka o prodavcu
  + Filtriranje
  + Pretraživanje
* *Dodavanje/Ažuriranje/Brisanje sadržaja*
  + Dodavanje sadržaja
    - Dodavanje slike
    - Unošenje cene proizvoda
    - Unošenje karakteristika proizvoda
  + Ažuriranje sadržaja
    - Promena cene proizvoda
  + Brisanje sadržaja
* Registrovanje/Prijavljivanje
  + Registrovanje
  + Prijavljivanje
* Ažuriranje sopstvenog naloga

Ove slučajevi korišćenja mogu da iniciraju neprivilegovan posetilac portala, privilegovan posetilac portala, prodavac ili administrator.

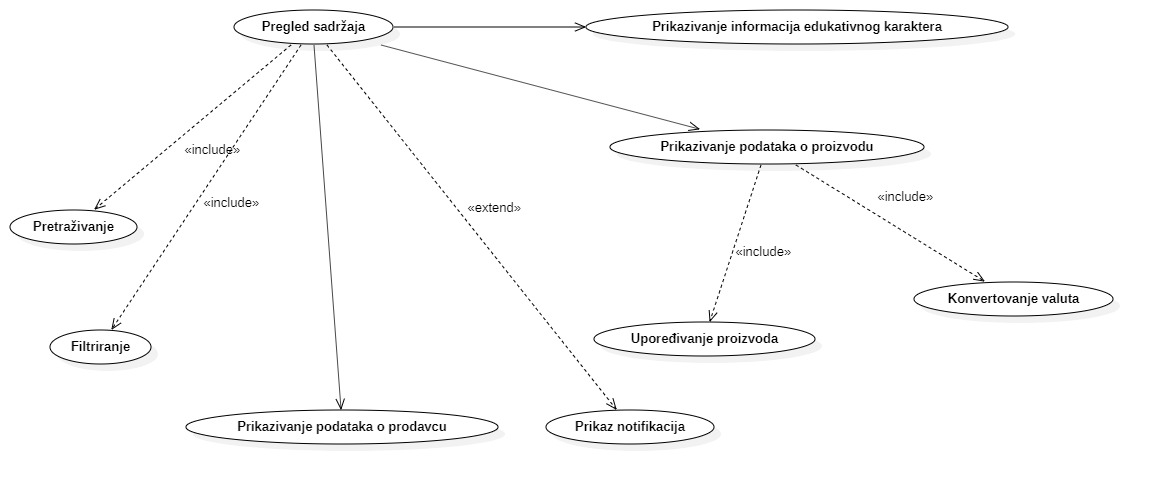
## Dijagrami slučajeva korišćenja

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

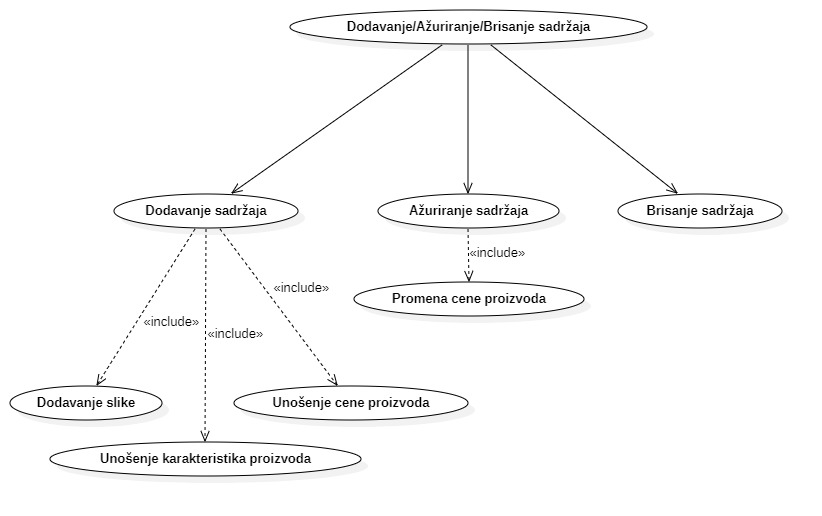


Slučaj korišćenja *Pregled sadržaja*, *Dodavanje/Ažuriranje/Brisanje sadržaja, Komentarisanje sadržaja* i *Registrovanje/Prijavljivanje* obuhvataju složenije radnje koje se mogu razložiti dalje na pojedinačne slučajeve korišćenja.

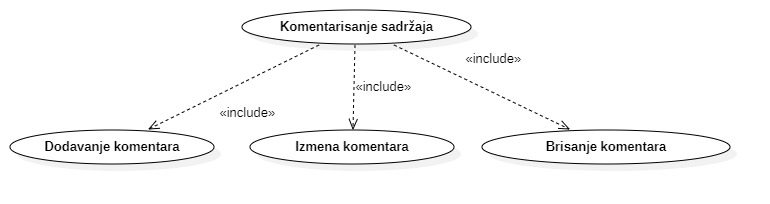
Detaljni UML dijagram za slučaj korišćenja *Pregled sadržaja* je prikazan na sledećoj slici:



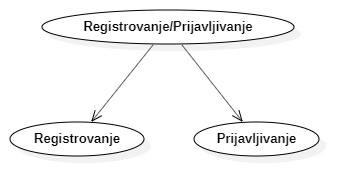
Detaljni UML dijagram za slučaj korišćenja *Dodavanje/Ažuriranje/Brisanje sadržaja* je prikazan na sledećoj slici:



Detaljni UML dijagrami za slučaj korišćenja *Komentarisanje sadržaja* je prikazan na sledećoj slici:



Detaljni UML dijagrami za slučaj korišćenja *Registrovanje/Prijavljivanje* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Izbor jezika

Kratak opis: Izbor između srpskog i engleskog jezika za prikaz informacija na portalu.

Akteri koji iniciraju slučaj korišćenja: Neprivilegovan posetilac portala, Privilegovan posetilac portala, Prodavac, Administrator.

### Pregled sadržaja

Kratak opis: Prikaz stranice portala sa izlistanim proizvodima.

Akteri koji iniciraju slučaj korišćenja: Neprivilegovan posetilac portala, Privilegovan posetilac portala, Prodavac, Administrator.

### Prikazivanje informacija edukativnog karaktera

Kratak opis: Prikaz stranice sa tekstovima bitnim za kupca prilikom kupovine proizvoda.

Akteri koji iniciraju slučaj korišćenja: Neprivilegovan/privilegovan posetilac portala, Prodavac, Administrator.

### Prikazivanje podataka o proizvodu

Kratak opis: Prikaz stranice portala sa detaljnijim karakteristikama proizvoda.

Akteri koji iniciraju slučaj korišćenja: Neprivilegovan/privilegovan posetilac portala, Prodavac, Administrator.

### Upoređivanje proizvoda

Kratak opis: Prikaz stranice sa izlistanim karakteristikama više proizvoda.

Akteri koji iniciraju slučaj korišćenja: Neprivilegovan/privilegovan posetilac portala, Prodavac, Administrator.

### Konvertovanje valuta

Kratak opis: Prikaz cena proizvoda na stranici u različitim valutama.

Akteri koji iniciraju slučaj korišćenja: Neprivilegovan/privilegovan posetilac portala, Prodavac, Administrator.

### Ocenjivanje sadržaja

Kratak opis: Korisnik ima mogućnost ocenjivanja proizvoda i prodavaca.

Akteri koji iniciraju slučaj korišćenja: Privilegovani posetilac.

### Dodavanje komentara

Kratak opis: Dodavanje komentara na određen proizvod.

Akteri koji iniciraju slučaj korišćenja: Privilegovan posetilac, Prodavac.

### Brisanje komentara

Kratak opis: Brisanje postavljenih komentara sa portala.

Akteri koji iniciraju slučaj korišćenja: Privilegovan posetilac, Prodavac, Administrator.

### Izmena komentara

Kratak opis: Vršenje izmena nad vlastitim već postavljenim komentarima.

Akteri koji iniciraju slučaj korišćenja: Privilegovan posetilac, Prodavac.

### Čuvanje sadržaja

Kratak opis: Korisnik dodaje izabrani proizvodi u sekciju omiljenih proizvoda.

Akteri koji iniciraju slučaj korišćenja: Privilegovani posetilac.

### Prikaz notifikacija

Kratak opis: Pregled pristiglih obaveštenja o promenama vezanih za sadržaj portala.

Akteri koji iniciraju slučaj korišćenja: Privilegovan posetila, Prodavac, Administrator.

### Prikazivanje podataka o prodavcu

Kratak opis: Prikaz stranice sa informacijama o prodavcu.

Akteri koji iniciraju slučaj korišćenja: Neprivilegovan/privilegovan posetilac portala, Prodavac, Administrator.

### Filtriranje

Kratak opis: Biranje filtera za prikaz sadržaja po određenim kriterijumima.

Akteri koji iniciraju slučaj korišćenja: Neprivilegovan/privilegovan posetilac portala, Prodavac, Administrator.

### Pretraživanje

Kratak opis: Opcija pretrage sadržaja portala (proizvoda/prodavaca).

Akteri koji iniciraju slučaj korišćenja: Neprivilegovan/privilegovan posetilac portala, Prodavac, Administrator.

### Dodavanje sadržaja

Kratak opis: Korisnik dodaje novi proizvod.

Akteri koji iniciraju slučaj korišćenja: Prodavac.

### Ažuriranje sadržaja

Kratak opis: Korisnik ažurira podatke proizvoda.

Akteri koji iniciraju slučaj korišćenja: Prodavac.

### Brisanje sadržaja

Kratak opis: Korisnik briše postojeći proizvod.

Akteri koji iniciraju slučaj korišćenja: Prodavac.

### Registrovanje

Kratak opis: Registrovanje posetioca portala u cilju dobijanja pristupa dodatnim funkcionalnostima sistema.

Akteri koji iniciraju slučaj korišćenja: Neprivilegovan/Privilegovan posetilac portala, Prodavac, Administrator.

### Prijavljivanje

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

Akteri koji iniciraju slučaj korišćenja: Privilegovan posetilac, Prodavac, Administrator.

### Ažuriranje sopstvenog naloga

Kratak opis: Vršenje izmena nad podacima koji se tiču naloga.

Akteri koji iniciraju slučaj korišćenja: Privilegovani posetilac, Prodavac.

# 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.

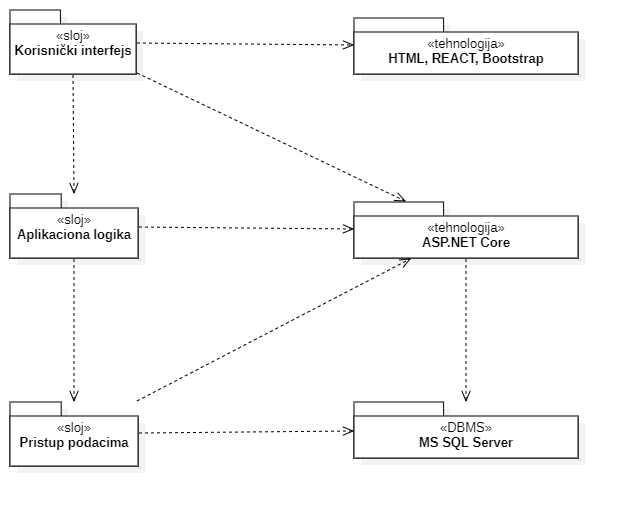
Logički pogled na WinScout portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket Korisnički interfejs sadrži web stranicu preko kojih korisnici sistema komuniciraju sa aplikacionim sistemom i prikazuje podatke iz baze podataka. Korisnički interfejs je realizovan preko React JavaScript biblioteke,HTML i CSS.

Paket *Aplikaciona logika* predstavlja srednji sloj sistema koji sadrži C# kod zadužen za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* je Microsoft SQL Server DBMS koji predstavlja 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 se nalazi HTML,CSS,React JavaScript biblioteke koje generišu web stranicu i sve njene funkcionalnosti.Web stranica je tipa SPA (Single-page application) koja komunicira sa korisnikom,dinamičkim prepisivanjem podataka koji su dobijeni pristupanjem pri web serveru sve u cilju pruženja lakše i brže interakcije korisnika sa interfejsom. Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i HTML i JavaScript biblioteka..

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi WinScout portala. Sadrži C# objekte 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.

### Pristup podacima

sloj

Sloj za pristup podacima je MS SQL Server koji se nalazi na dnu troslojne arhitekture i ima funkciju pribavljanja, dodavanja i ažuriranja podataka koji se čuvaju u bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali su entiteti u bazi zavisni od srednjeg sloja jer se koristi code-first pristup prilikom njihovog kreiranja.

### HTML, ReactJS, BootstrapReact

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.

Tehnologija ReactJS je front-end JavaScript biblioteka za izradu korisničkog interfejsa ili korisničkih komponenti.React je zabrinut za održavanje stanja i prikazivanjem stanja DOM elementima. React aplikacije zahtevaju dodatne biblioteke za rutiranje stranica,kao i za odredjene klijentske funkcionalnosti.

Tehnologija BootstrapReact omogućava pristup velikom UI ekosistemu već napravljenih React komponenti i tabeli različitih stilova pomoću kojih se definiše izgled React komponenti.

### ASP.NET Core

tehnologija

Tehnologija ASP.NET Core obezbeđuje mehanizam za pisanje i izvršavanje klasa i metoda na strani servera. Ove metode mogu da opsluže zahteve od strane klijenta i pristupaju bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

### MS SQL Server

DBMS

MS SQL Server predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju WinScout portala.

# Pogled na procese

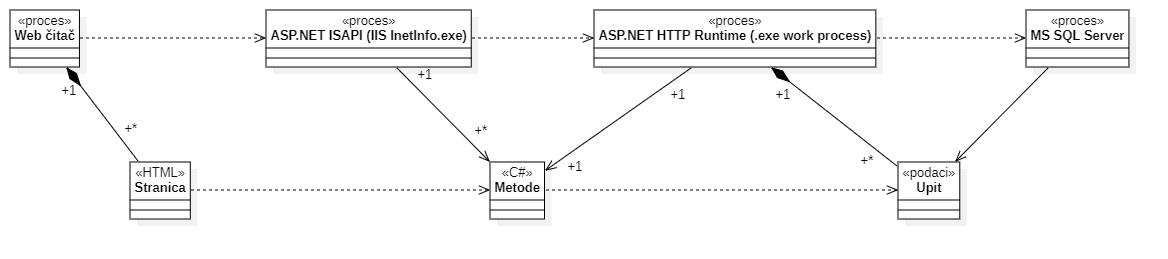
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

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

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

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju WinScout portala. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na ASP.NET Core i Microsoft SQL Server bazi podataka.



### Web čitač

Web čitač je proces koji izvršava funkcionalnost aplikacije za prikaz web stranica dobijenih od nekog Web servera.

Web stranica je tipa SPA(Single-page application) kod koje se sav ptreban kod HTML,JavaScript i CSS se preuzima u okviru učitavanja jedne stranice,ili se odgovarajući resursi učitavaju dinamički i po potrebi dodaju stranici,obično kao posledica neke akcije od strane korisnika.Zahtevi serveru obično rezultuju vraćanjem sirovih podataka (JSON).Kada server vrati sirove podatke na korisničkoj strani JavaSript šablon prevodi sirove podatke u HTML,koji se dalje korisiti za ažuriranje DOM-a.

Web čitač zavisi od IIS Web servera koji generiše i vraća odgovarajuće JSON podatke na zahtev.

### IIS Web server

IIS Web server sadrži proces inetinfo.exe koji direktno opslužuje klijenta tako što osluskuje HTTP zahteve na portu 80. Pristigli zahtevi se smeštaju u red (queue) i ASP.NET ISAPI ih prosledjuje ASP worker procesu. Zadužen je za vraćanje odgovora na klijentove zahteve. Izvršava se u user mode-u.

### ASP.NET HTTP Runtime

ASP worker proces koji izvršava ASP.NET web aplikaciju, pristigli klijentov zahtev predaje ASP.NET HTTP runtime okruženju, gde se izvršavaju metode za obradu zahteva, obraćanje DBMS serveru radi obrade podataka, i generišu odgovori za klijenta koji je inicirao komunikaciju.

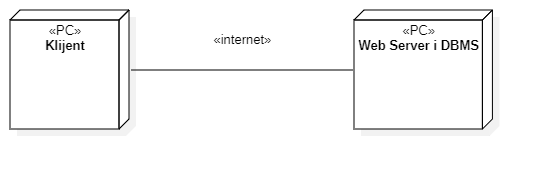
### MS SQL Server

MSSQLSERVER Service za upravljanje bazama podataka, u okviru Microsoft SQL Server-a, sadrži proces sqlservr.exe koji ima ulogu interfejsa ka bazi podataka. Prihvata i izvršava upite nad bazom, i vraća rezultate procesu koji ih je 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 WinScout portala.



## Klijent

Pristup WinScout portalu 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 i DBMS

Računar na kome se izvršava Web server i DBMS simultano. Web server opslužuje više klijenata koji pristupaju preko Interneta, i u okviru njega, izvršavaju se procesi za razmenu HTTP zahteva i odgovora (komunikaciju sa klijentom), kao i proces za izvršavanje metoda u cilju obrade zahteva i generisanje odgovora. Takože, na istoj mašini se u okviru DBMS-a izvršavaju i procesi koji su zaduženi za rad sa bazom. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom u lokalnoj mreži (LAN), ali za inicijalne potrebe WinScout portala, ustanovljeno je da je jedna mašina dovoljna za izvršavanju obe funkcionalnosti.

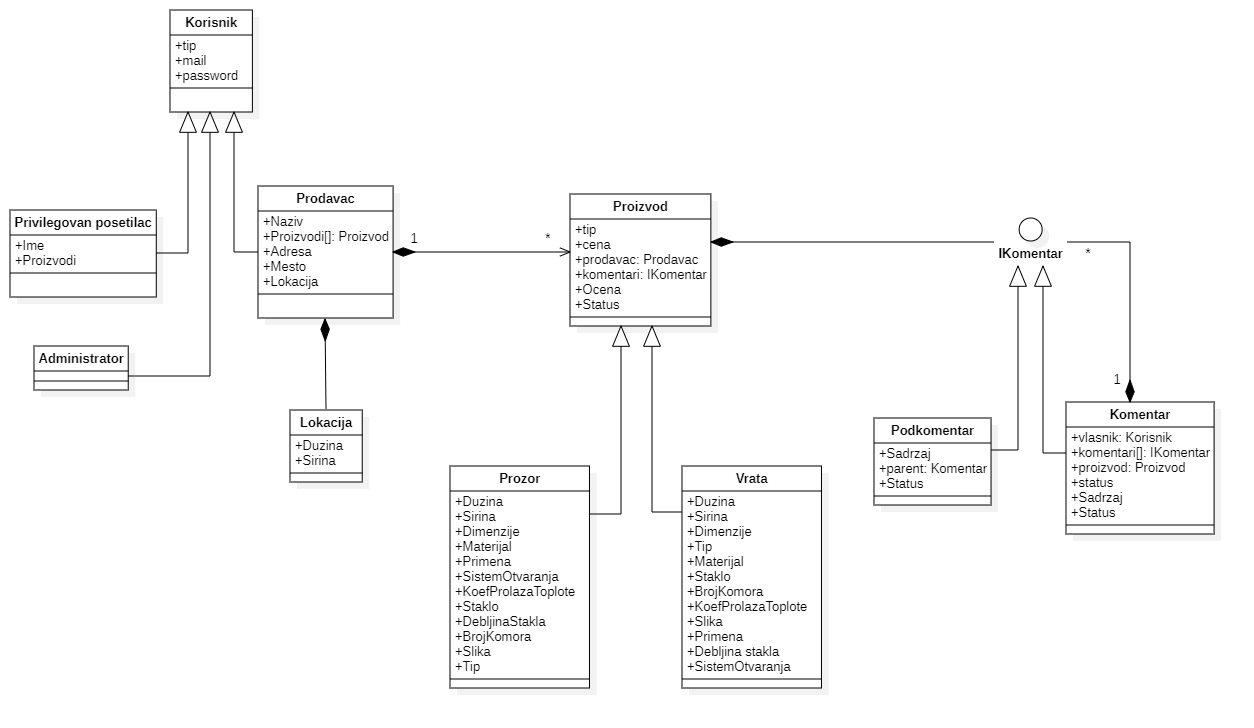
# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju WinScout portala ovaj odeljak sadrži model domena, šemu baze podataka i prikaz komponenti sistema razvrstanih u ranije identifikovane pakete.

## Model domena

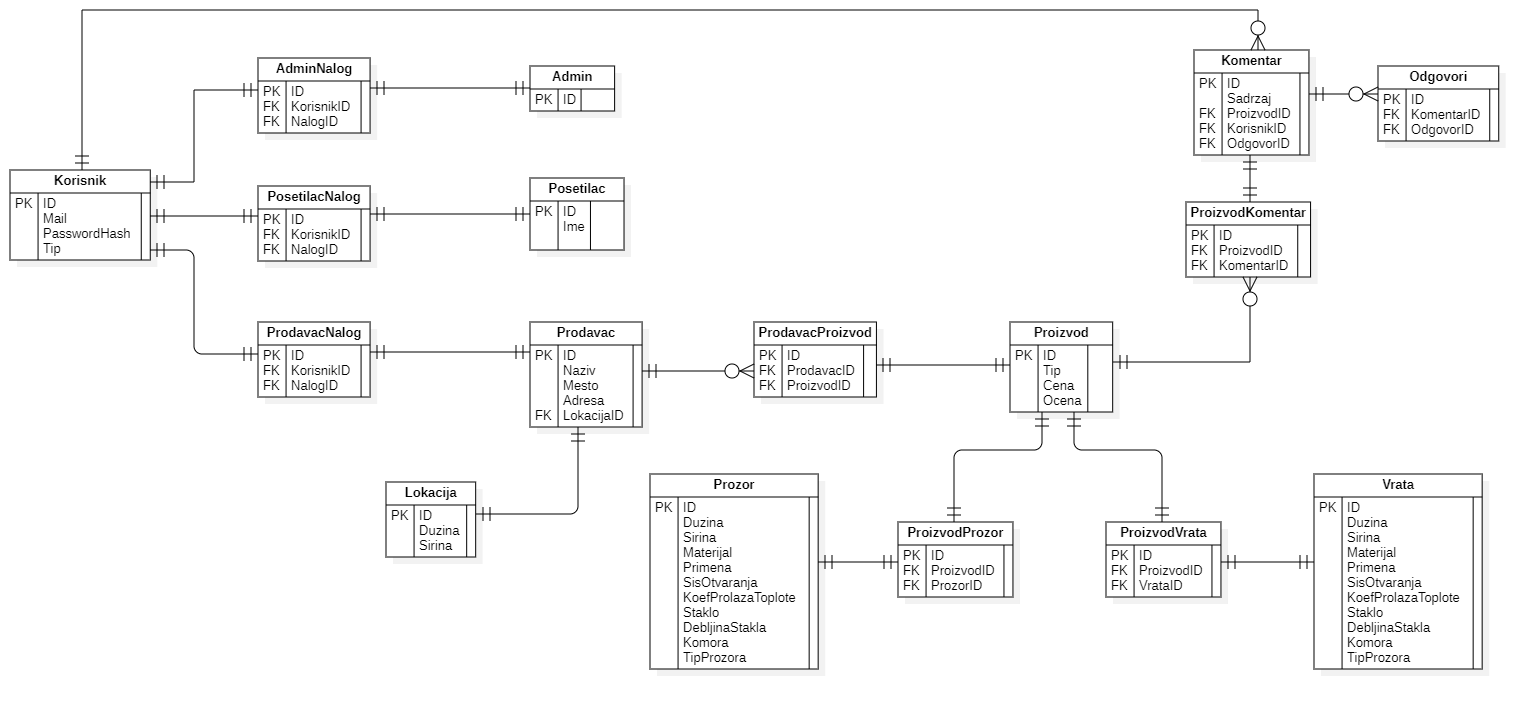
Model domena za koji se WinScout portal projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti (klasa i metoda) koje će biti implementirane.



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Dijagram baze je kreiran korišćenjem *StarUML*-a, dok je migracija na MS SQL Server obavljena pomoću *Microsoft Entity Framework*-a.



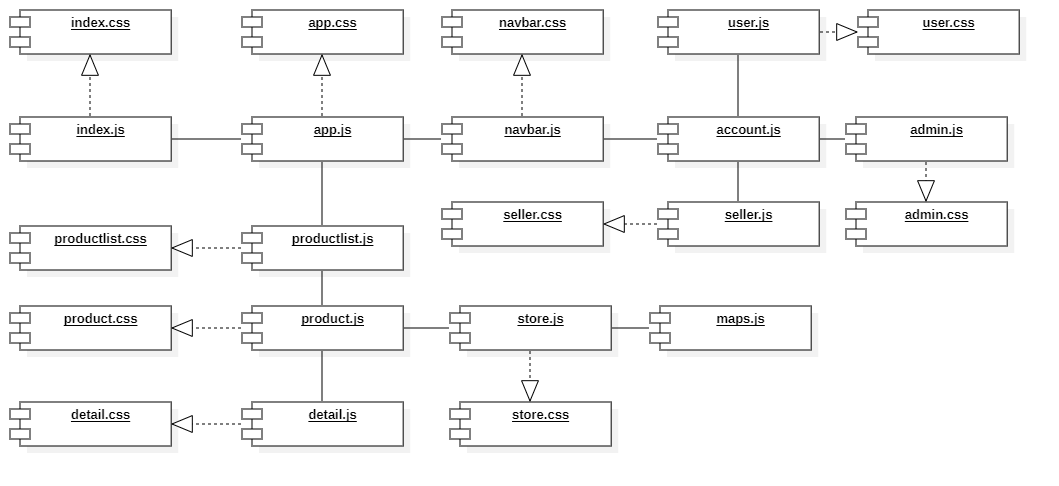
## Komponente sistema

Komponente sistema WinScout portala su .js, .css i .cs fajlovi čiji će pregled biti dat po arhitekturnim

slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti i klasni dijagrami.

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvanaest komponentama:



Komponenta **index.js** implemenentira stranicu portala i prikazuje DOM objekte čiji sadržaj šalje app.js komponenta.

Komponenta **index.css** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na različitim stranicama.

Komponenta **app.js** isključivo se bavi usmeravanjem(routhing) ostalih komponenti u zavisnosti od url linka.

Komponenta **app.css** predstavlja opis stilova pojedinih HTML komponeti koji se javljaju.

Komponenta **navbar.js** implementira osnovni navigacioni sistem u web portalu dok komponenta navbar.css stilizuje navigacioni meni.

Komponenta **productlist.js** implementira prikaz svih proizvoda na web stranici.Takođe ima mogućnost filtriranjem proizvoda po željenom upitu i pretraživanjem proizvoda.

Komponenta **productlist.css** predstavlja opis prikaza tih proizvoda.

Komponenta **product.js** implementira prikaz kartica proizvoda sa njenim nazivom,tipom i cenom dok komponenta **product.css** predstavlja opis stilova HTML elemenata.

Komponenta **details.js** implementira detaljniji prikaz proizvoda a komponenta **details.css** predstavlja opis stilova HTML elemenata.

Komponenta comments.js implementira odeljak za prikazivanje i dodavanje komentara dok komponenta **comments.css** predstavlja opis stilova HTML elemenata.

Komponenta **store.js** implementira prikaz prodanice proizvoda i njene osnovne informacije o njoj kao i sve proizvode te prodavnice.

Komponenta **store.css** realizuje opis HTML elemenata.

Komponenta **maps.js** implementira prikaz interaktivne mape kao i tačno određenu lokaciju prodavnice.

Komponenta **accounts.js** implementira prikaz prijave i registracije na nalog web portala.

Komponenta **accounts.css** predstavlja opis stilova HTML elementa.

Komponenta **user.js** implementira prikaz privilegovanog korisiničkog naloga ,njegove notifikacije kao i proizvode na koje se pretplatio.

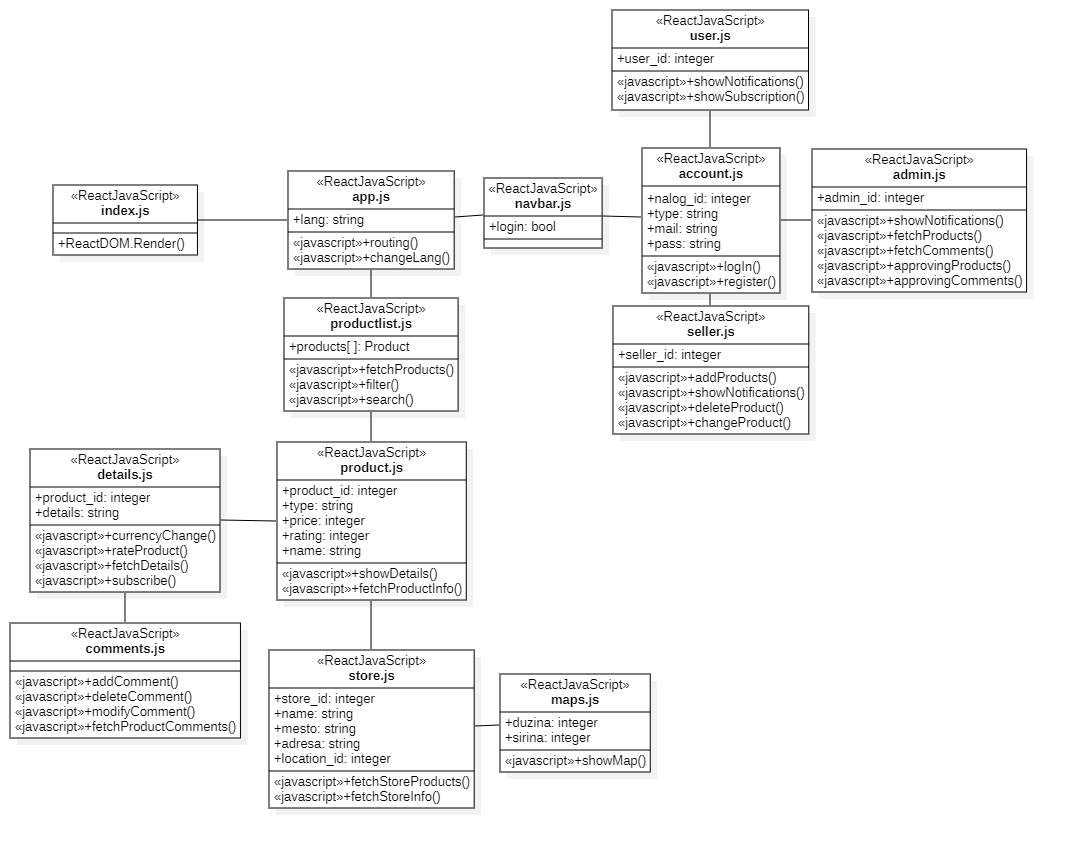
Komponenta **user.css** predstavlaj opis stilova HTML elemenata.

Komponenta **admin.js** implementira prikaz naloga administratora i prikaz svih proizvoda i komentara koji čekaju na odobrenje.

Komponenta **admin.css** predstavlja opis stilova HTML elemenata.

Komponenta **seller.js** implementira prikaz naloga prodavca,njegovih proizvoda i odeljka za dodavanje,brisanje i izmenu proizvoda.

Parametri komponenti za prikaz stranice su ilustrovani na sledećim dijagramima klasa:



Značenje atributa za navbar.js:

• login - definiše da li je korisnik prijavljen

Značenje atributa za productlist.js:

• products[ ] - niz svih proizvoda

Značenje atributa za product.js:

• product\_id - identifikator proizvoda

• type - tip proizvoda

• price - cena proizvoda

• rating - ocena proizvoda

• name - ime proizvoda

Značenje atributa za details.js:

• product\_id - identifikator proizvoda

• details - detalji proizvoda

Značenje atributa za store.js:

• store\_id - identifikator prodavnice

• name - ime prodavnice

• mesto - mesto prodavnice

• adresa - adresa prodavnice

• location\_id - identifikator lokacije

Značenje atributa za maps.js:

• duzina - koordinate geografske dužine

• širina - koordinate geografske širine

Značenje atributa za accounts.js:

• nalog\_id - identifikator korisničkog naloga

• type - tip korisničkog naloga

• mail - email adresa korisničkog naloga

• pass - lozinka korisničkog naloga

Značenje atributa za seller.js:

• seller\_id - identifikator prodavca

Značenje atributa za user.js:

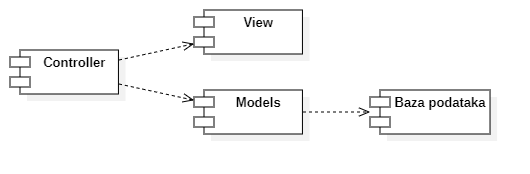
• user\_id - identifikator privilegovanog korisnika

Značenje atributa za admin.js:

• admin\_id - identifikator administratora

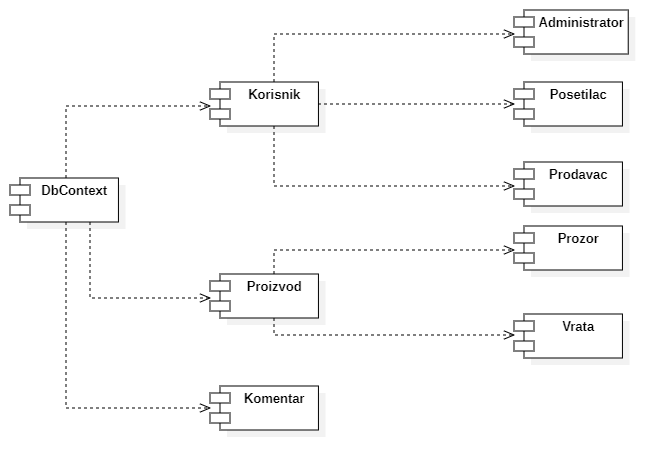
### Komponente aplikacione logike

Komponente koje realizuju aplikacioni sloj predstavljene su na sledećem dijagramu:



Iz dijagrama se vidi da je za izradu web aplikacije korišćena MVC arhitektura.

Models komponenta sadrži sve potrebne modele koji služe za pristup, unos i izmene podataka u bazi. Sledi detaljniji dijagram komponenti vezan za modele:



DbContext klasa je apstrakcija baze podataka, preko koje se vrši pristup bazi.

Controller komponenta ima namenu u komunikaciji sa korisnikom. Sadrži CRUD metode koje se izvršavaju kao odgovor na HTTP zahteve pristigle od korisnika. Komponenta koristi definisane modele podataka preko koje vrši pribavljanje i izmene nad podacima iz baze.

View komponenta ima ulogu u prezentaciji sadržaja korisniku dobijenih iz baze podataka korišćenjem modela. Napomena: View i Models komponente ne interaguju direktno, već se njihova komunikacija vrši preko Controller-a koji njima upravlja.

# 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 portalu.
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. WinScout portal će biti dostupan 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.