GiveWave

WEB aplikacija za doniranje

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 21.04.2023. | 1.0 | Inicijalna verzija | Stevan  Nikola  Dusan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 osnovnih podataka o laboratoriji 8

6.2.3 Pregled spiska članova 8

6.2.4 Pregled podataka o određenom članu laboratorije 8

6.2.5 Pregled publikacija po autoru 8

6.2.6 Pregled publikacija po tipu 8

6.2.7 Pregled publikacija po godini 8

6.2.8 Pregled spiska projekata 8

6.2.9 Pregled podataka o određenom projektu 8

6.2.10 Prijavljivanje 9

6.2.11 Ažuriranje podataka o članu 9

6.2.12 Dodavanje nove publikacije 9

6.2.13 Brisanje postojeće publikacije 9

6.2.14 Ažuriranje osnovnih podataka o laboratoriji 9

6.2.15 Kreiranje novog člana 9

6.2.16 Brisanje postojećeg člana 9

6.2.17 Arhiviranje postojećeg člana 9

6.2.18 Kreiranje projekta i postavljanje vođe 9

6.2.19 Ažuriranje podataka o projektu 9

7. Pogled na logičku arhitekturu sistema 9

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 10

7.1.1 Korisnički interfejs 10

7.1.2 Aplikaciona logika 10

7.1.3 Pristup podacima 10

7.1.4 HTML 10

7.1.5 CSS 11

7.1.6 MySQL 11

8. Pogled na procese 11

8.1 Procesi 11

8.1.1 Web čitač 11

8.1.2 Web server 11

8.1.3 ASP.NET 11

8.1.4 MySQL Server 12

9. Pogled na raspoređivanje sistema 12

9.1 Klijent 12

9.2 Web server 12

9.3 DBMS server 12

10. Pogled na implementaciju sistema 12

10.1 Model domena 12

10.2 Šema baze podataka 13

10.3 Komponente sistema 14

10.3.1 Komponente korisničkog interfejsa 14

10.3.2 Komponente aplikacione logike 14

10.3.3 Komponente za pristup podacima 15

11. Performanse 16

12. Kvalitet 16

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture GiveWave aplikacije.

# Opseg dokumenta

Dokument se odnosi na GiveWave aplikaciju koja će biti razvijena od strane DNSTeam-a. Namena GiveWave aplikacije je efikasno doniranje ugrozenima.

# Reference

Spisak korišćene literature:

1. GiveWave – Predlog projekta, D01\_Predlog\_Projekta, V1.0, 2023, DNSTeam.
2. GiveWave – Vizija Sistema, D02\_Vizija\_Sistema, V1.0, 2023, DNSTeam.
3. GiveWave – Plan realizacije projekta, , D03\_Plan\_Realizacije, V1.0, 2023, DNSTeam.
4. GiveWave – Specifikacija zahteva, D04\_Spec\_zahteva,V1.0, 2023, DNSTeam.

# 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. GiveWave Web aplikacija će biti implementirana kao Web aplikacija zasnovana na Angular skripting jeziku i MySQL bazi podataka [4].
2. Klijentski deo GiveWave Web aplikacije će biti optimizovan za sledeće Web čitače: Internet Explorer 11.0 i noviji, Opera 43.0 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.

# Pogled na slučajeve korišćenja

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

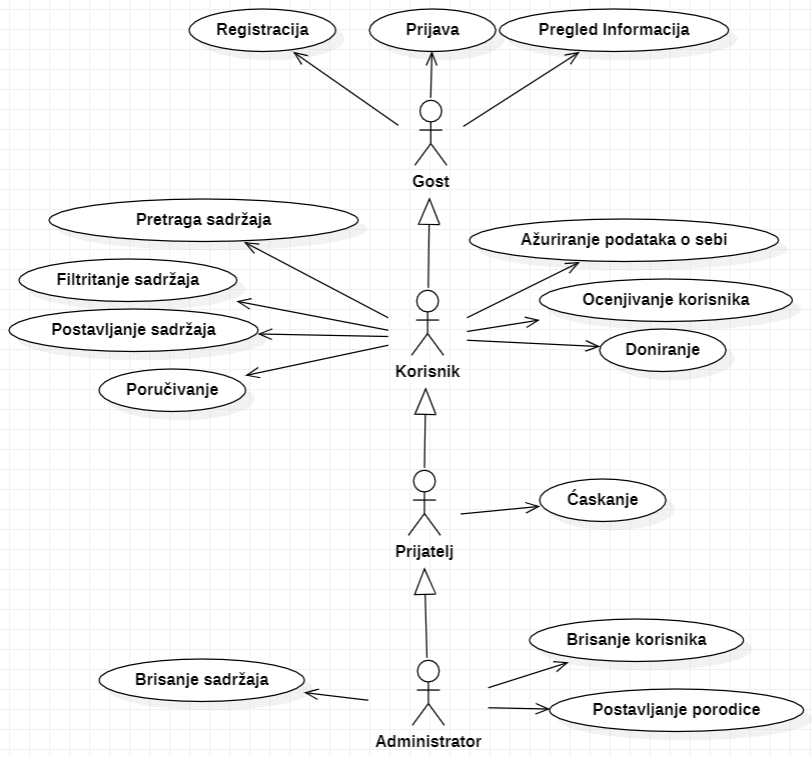
Slučajevi korišćenja GiveWave Web aplikacije su:

* Prijavljivanje
* Pregled podataka profila
* Izmena podataka profila
* Registracija
* Pregled pocetne stranice
* Pregled profila prijatelja
* Razmena poruka medju korisnicima
* Postavljanje donacije
* Izmena donacije
* Brisanje donacije

Ove slučajeve korišćenja mogu da iniciraju korisnik,prijatelj i administrator.

## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja GiveWave Web aplikacije prikazan je na sledećoj slici:

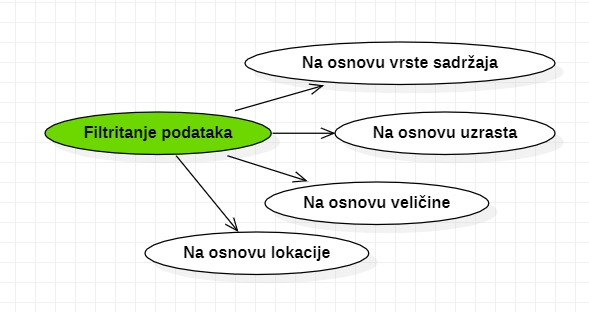


Slučajevi korišćenja *pregled informacija* i *kreiranje, brisanje i arhiviranje članova* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja

Detaljni UML dijagram za slučaj korišćenja registracije je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja filtritanje podataka je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Prijavljivanje

Kratak opis: Logovanje na sajt koristeći svoj mejl i lozinku.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Prijatelj, Administrator.

### Pregled podataka profila

Kratak opis: Prikaz stranice sa informacijama o korisniku.

Akteri koji iniciraju slučaj korišćenja: Korisnik.

### Izmena podataka profila

Kratak opis: Prikaz stranice sa formom za izmenu postojećih podataka korisnika.

Akteri koji iniciraju slučaj korišćenja: Korisnik.

### Registracija

Kratak opis: Prikaz stranice za registraciju novih korisnika.

Akteri koji iniciraju slučaj korišćenja: Gost.

### Pregled pocetne stranice

Kratak opis: Prikaz stranice sa opcijama za doniranje,i postavljanje proizvoda.

Akteri koji iniciraju slučaj korišćenja: Korisnik.

### Pregled podatala prijatelja

Kratak opis: Prikaz stranica sa detaljima o korisniku koji je prijatelj.

Akteri koji iniciraju slučaj korišćenja: Korisnik..

### Razmena poruka izmedju prijatelja

Kratak opis: Prikaz stranice sa dijalogom među dva korisnika.

Akteri koji iniciraju slučaj korišćenja: Prijatelj

### Postavljanje donacije

Kratak opis: Prikaz stranice sa mogucnoscu postavljanja donacije.

Akteri koji iniciraju slučaj korišćenja: Korisnik,Administrator

### Brisanje proizvoda

Kratak opis: Prikaz stranice sa mogucnoscu brisanja proizvoda.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator.

### Izmena donacije

Kratak opis: Prikaz stranice sa mogucnoscu izmene donacije.

Akteri koji iniciraju slučaj korišćenja: Korisnik,Administrator

# 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.

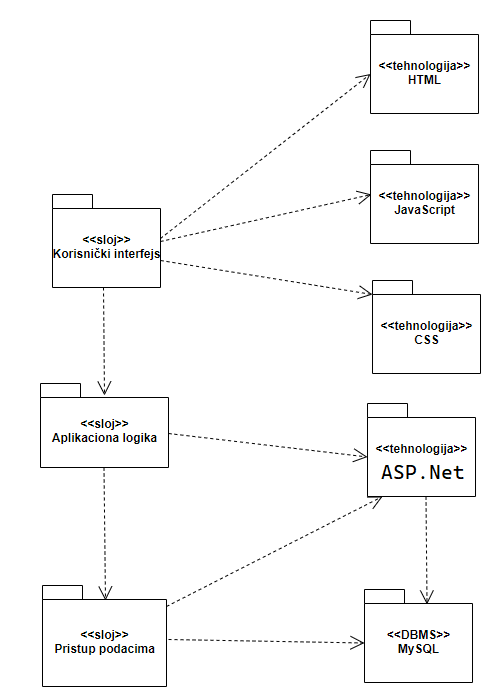
Logički pogled na GiveWave Web aplikaciju obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

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

Paket *Aplikaciona logika* predstavlja srednji sloj sistema koji sadrži ASP.Net fajlove zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži ASP.Net fajlove 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 web aplikacije. U njemu su sadržane sve HTML, multimedijalni sadržaji i Angular komponente koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

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

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi GiveWave Web aplikacije .Sadrži ASP.Net fajlove 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 se nalazi na dnu troslojne arhitekture i sadrži ASP.Net skripte zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u lokalnoj bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa ASP.Net i lokalne baze 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.

### 7.1.5 CSS

tehnologija

CSS (eng. Cascading Style Sheets) je jezik formatiranja pomoću kog se definiše izgled elemenata webstranice. Ove skripte će imati ulogu stilizovanja korisničkog interfejsa i nemaju veze sa funkcionalnošću aplikacije.

### 7.1.6 JavaScript – jQuery

tehnologija

Tehnologija JavaScrpt, odnosno jQuery pisan u JS-u, obezbeđuje mehanizam za pisanje i izvršavanje skripti na strani klijenta. Ove skripte mogu da generišu HTML kod koji realizuje korisnički interfejs i komuniciraju sa bazom podataka preko servera u cilju pribavljanja, unosa i ažuriranja podataka.

### ASP.Net

tehnologija

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

### SQL LocalDB

DBMS

SQL LocalDB predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju GiveWave 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 dotnetu imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta .Net 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 GiveWave Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju GiveWave Web aplikacije. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na ASP.Net-u i MySQL bazi podataka.

### 

### Web čitač

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

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

### Web server

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

### ASP.Net

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

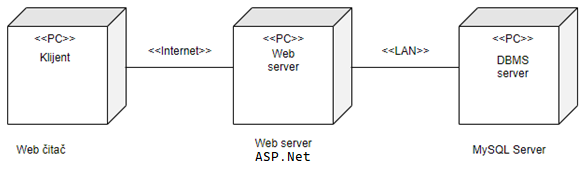
### MySQL Server

MySQL Server je proces koji izvršava funkcionalnost MySQL 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 GiveWave aplikacije.



## Klijent

Pristup GiveWave 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 ASP.Net koji vrše obradu zadatih ASP.Net 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 MySQL 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).

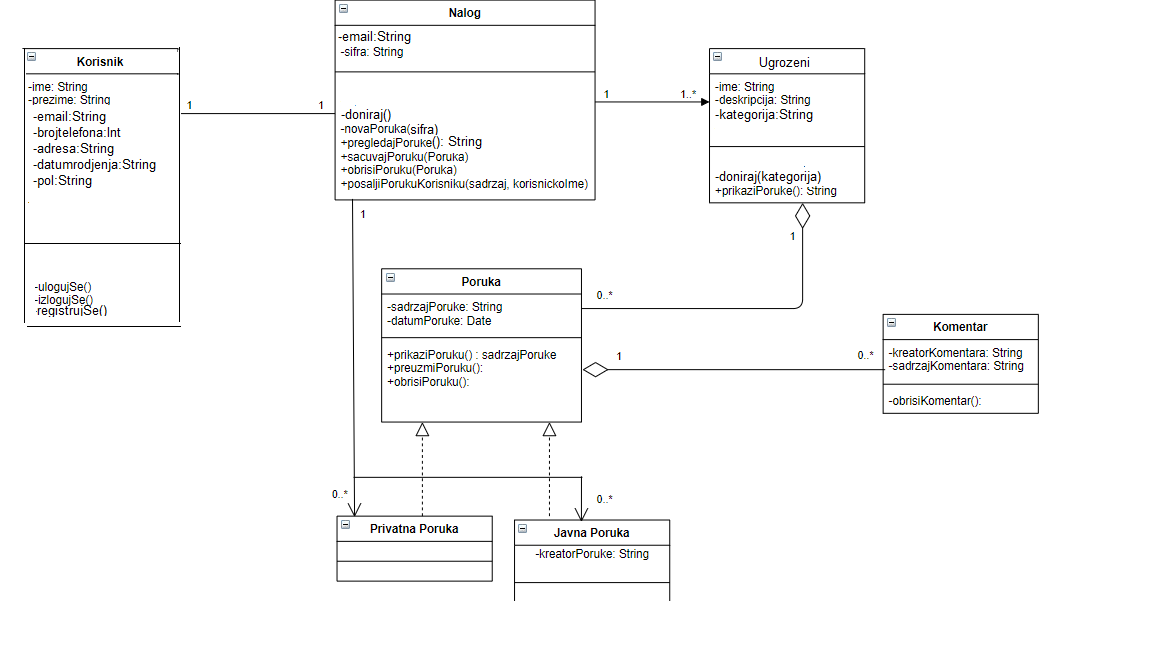
# Pogled na implementaciju sistema

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

## Model domena

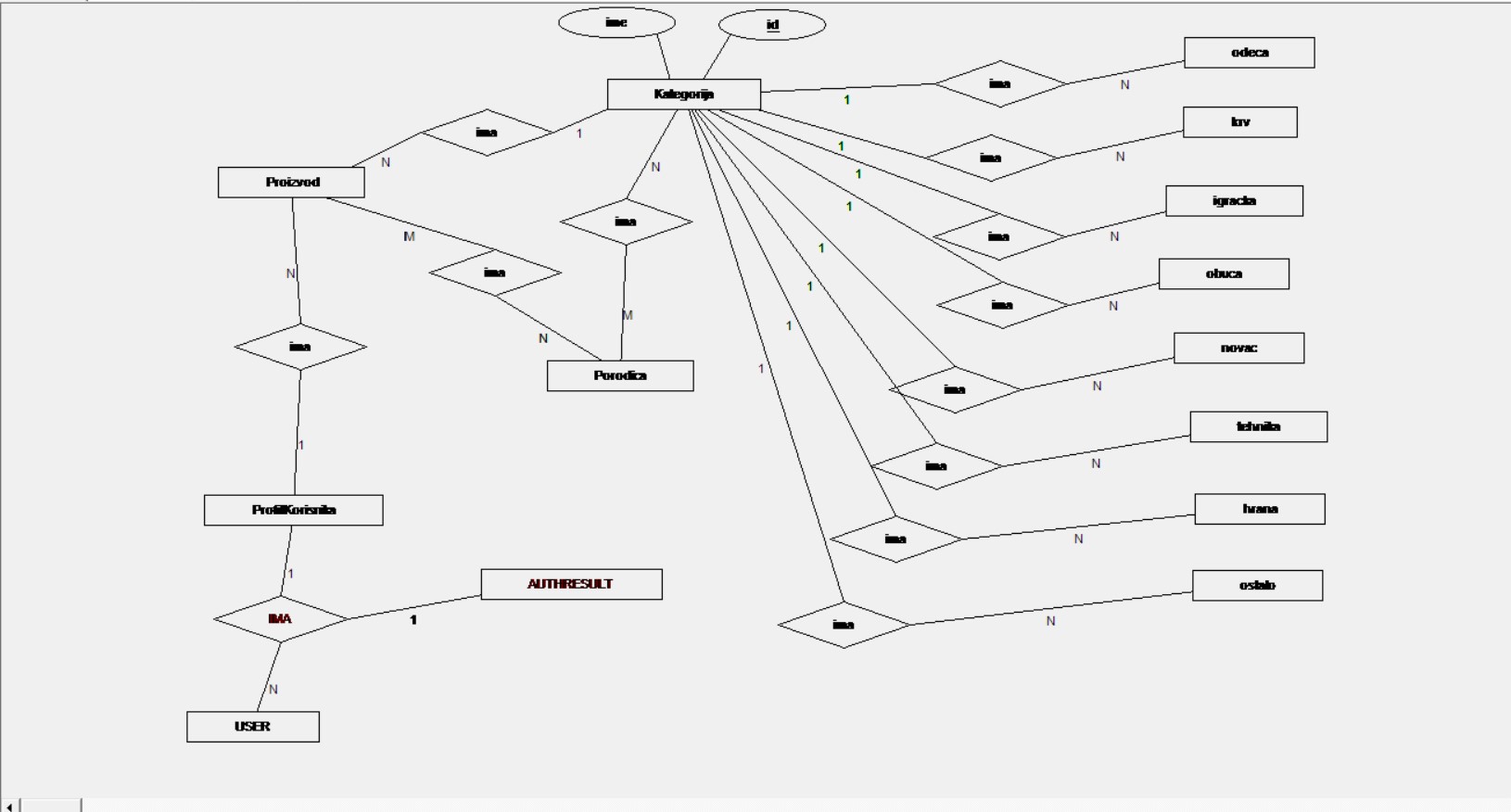
Model domena za koji se GiveWave aplikacija projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

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



## Komponente sistema

Komponente sistema GiveWave Web aplikacije su ASP.Net skripti čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je ASP.Net 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

**Uvodna napomena:** Projekat se radi u Angularu, koristi se webpack i razni css fajlovi komponenti koji ce biti kompajlirani u jedan main.css. Ostali fajlovi će biti podeljeni po modulima i biće ubačeni, po potrebi, u jedan glavni index.html fajl.

Dizajn korisničkog interfejsa za sve korisnike je obuhvaćen tipom komponenti: main.css.

Dizajn korisničkog interfejsa nakon otvaranja web stranice:



Komponenta **login.component.html** implementira stranicu sa koje korisnik moze da se uloguje.

Komponenta **navigation-bar.component.html** implementira navigacioni bar i navigacioni panel.

Dizajn korisničkog interfejsa nakon logovanja korisnika:

  
Sa ovog interfejsa korisnik može da ode:

1. Na svoj profil



1. Na proizvode



1. Na donacije



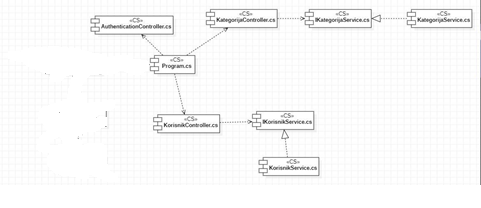
Korisnik može kroz **products.component.html** da filtrira i pretražuje podatke kako bi brže pretraživao. Na njemu se nalaze proizvodi koji su postavljeni

Korisnički interfejs **profile.component.html** ce prikazati lični profil ili profil nekog drugog korisnika, zajedno sa postavljenim proizvodim i istorijom donacija. Korisnik će moći da menja podatke na svom profilu.

Korisnički interfejs **donate.component.html** ce prikazati listu ugroženih.

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **program.cs** 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:



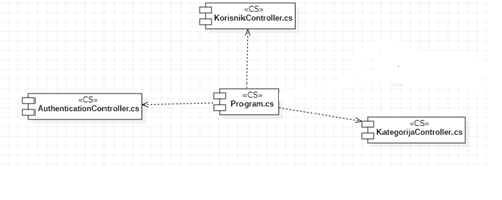
Kratak pregled funkcija svake komponente:

* LogIn– stranica za prijavljivanje ili kreiranje profila korisnika
* profil korisnika – stranica za prikaz podataka o ulogovanom korisniku
* Uredi-stranica za uredjivanje profila.
* Products- prikaz svih kategorija proizvoda koje mogu da se doniraju.
* Donate – stranica za doniranje odredjenog proizvoda.

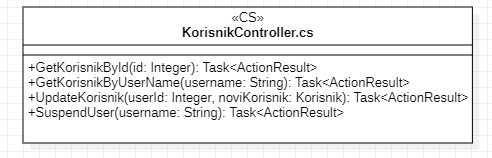
### Komponente za pristup podacima

Za realizaciju projekta koristi se Microsoft SQL Server baza podatakaPristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru EF Core framework-a .Pomenuti framework olaksava rad sa bazama podataka i nudi bogatu funkcionalnost za upravaljanje relacijskim podacima.

Na sledećem UML dijagramu klasa pobrojane su funkcije za pristup podacima iz baze. Uz pomoc GiveWaveDbContext klase, koja predstavlja sesiju sa bazom podataka izvrsvamo upite nad bazom podataka. Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru cs fajlova KorisnikController.cs, AuthenticationController.cs, KategorijaController.cs. Pomenuti fajlovi se uključuju na početku program.cs-a, tako da su sve funkcije za pristup podacima dostupne svim komponentama.



Na sledećim UML dijagramima klasa pobrojane su funkcije za pristup podacima iz baze:

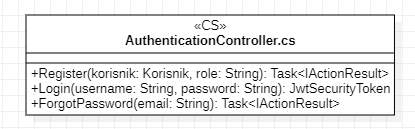


• GetKorisnikById – pribavljanje entiteta Korisnika na osnovu id-a

• GetKorisnikByUserName-pribavljanje entiteta Korisnika na osnovu userName-a

• UpdateKorisnik-ažuriranje podataka o korisniku

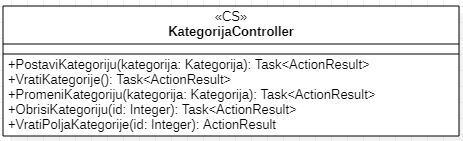
• SuspendKorisnik-blokiranje korisničkog naloga



• Register-kreiranje korisničkog naloga

• Login-loguje korisnika i vraća token

• ForgotPassword-promena šifre od strane korisnika



* PostaviKategoriju - Postavljanje entiteta kategorija
* VratiKategorije – Vraća sve kategorije sa njihovim podkategorijama
* PromeniKategoriju – Ažurira kategoriju
* ObrisiKategoriju – Briše se kategorija iz baze
* VratiPoljaKategorije – Vraća specifična polja za datu kategoriju na osnovu id-a

# 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. GiveWave aplikacija 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.