

Reši problem smeća u prirodi

Arhitekturni projekat

Verzija 2.0

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

Pregled izmena

Datum	Verzija	Opis	Autor
23.04.2020.	1.0	Inicijalna verzija	MGMTeam
02.07.2020.	2.0	Finalna verzija	MGMTeam

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

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
7.	Dijagrami slučajeva korišćenja	6
8.	Kratak opis slučajeva korišćenja	7
8.1	Pregled sadržaja	7
8.2	Dodavanje organizatora	7
8.3	Brisanje korisnika	7
8.4	Kreiranje naloga	7
8.5	Prijavljivanje za učešće na akciji	7
8.6	Pisanje žalbe	7
8.7	Postavljanje slika	7
8.8	Obeležavanje lokacija	7
8.9	Prijavljivanje	7
8.10	Potvrđivanje učesnika akcija	8
8.11	Postavljanje slika završenog posla	8
8.12	Pregled slika	8
8.13	Određivanje preporučenog broja učesnika	8
8.14	Određivanje datuma događaja	8
9.	Pogled na logičku arhitekturu sistema	8
9.1	Pregled arhitekture – organizacija paketa i podsistema u slojeve	9
9.1.1	Korisnički interfejs	9
9.1.2	Aplikaciona logika	9
9.1.3	Pristup podacima	9
9.1.4	HTML	9
9.1.5	JavaScript	9
9.1.6	Bootstrap	9
9.1.7	SQL Server	10
10.	Pogled na procese	10
10.1	Procesi	10
10.1.1	Web čitač	10
10.1.2	Web server	10
10.1.3	SQL Server	11

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

11.	Pogled na raspoređivanje sistema	11
11.1	Klijent	11
11.2	Web server	11
11.3	DBMS server	11
12.	Pogled na implementaciju sistema	11
12.1	Model domena	11
12.2	Šema baze podataka	12
12.3	Komponente sistema	13
12.3.1	Komponente korisničkog interfejsa	13
12.3.2	Komponente aplikacione logike	17
12.3.3	Komponente za pristup podacima	18
13.	Performanse	22
14.	Kvalitet	22

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

Arhitekturni projekat

1. Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Trashee aplikacije.

2. Opseg dokumenta

Dokument se odnosi na Trashee aplikaciju koji će biti razvijena od strane MGMTeam-a. Namena aplikacije je organizovanje akcija čišćenja smeća.

3. Reference

Spisak korišćene literature:

1. SWE_01_Predlog_Projekta, V1.0, 2020, MGMTeam
2. SWE_02_Vizija_Sistema, V2.0, 2020, MGMTeam.
3. SWE_03_Plan_Realizacije_Projekta, V2.0, 2020, MGMTeam.
4. SWE_Trashee_Raspored_Aktivnosti, V1.0, 2020, MGMTeam
5. SWE_04_Specifikacija_Zahteva, V2.0, 2020, MGMTeam.

4. 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.

5. Ciljevi i ograničenja arhitekture

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

1. Trashee će biti implementirana kao Web aplikacija zasnovana na ASP.NET Core, Bootstrap i SQL Server bazi podataka [4].
2. Klijentski deo Trashee portala će biti optimizovan za sledeće Web čitače: Microsoft Edge, Google Chrome i Mozilla Firefox. [4].
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.

6. Pogled na slučajeve korišćenja

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

Slučajevi korišćenja Trashee aplikacije su:

- Pregled sadržaja
- Dodavanje organizatora
- Brisanje korisnika
- Kreiranje naloga
- Prijavljivanje za učešće u akciji
- Pisanje žalbe
- Postavljanje slika
- Obeležavanje lokacija
- Prijavljivanje

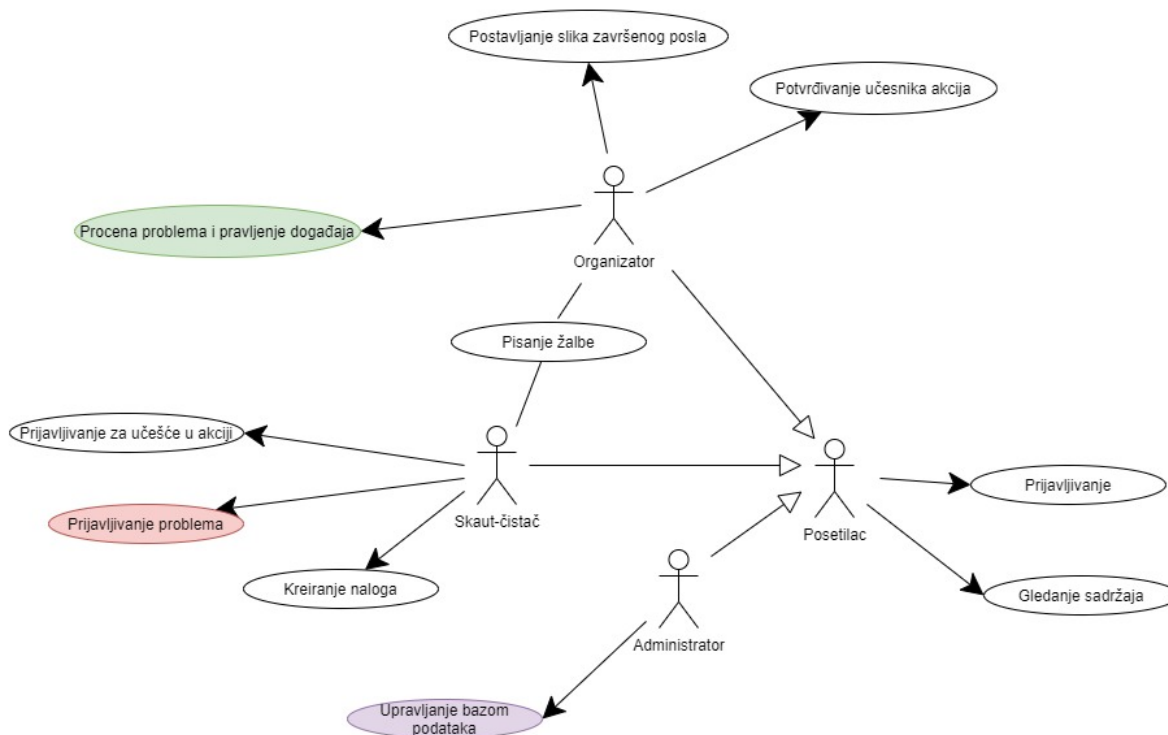
Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

- Potvrđivanje učesnika akcija
- Postavljanje slika završenog posla
- Pregled slika
- Određivanje preporučenog broja učesnika
- Određivanje datuma događaja

Ove slučajeve korišćenja mogu da iniciraju posetilac, organizator, skaut-čistač, vođa projekta ili administrator.

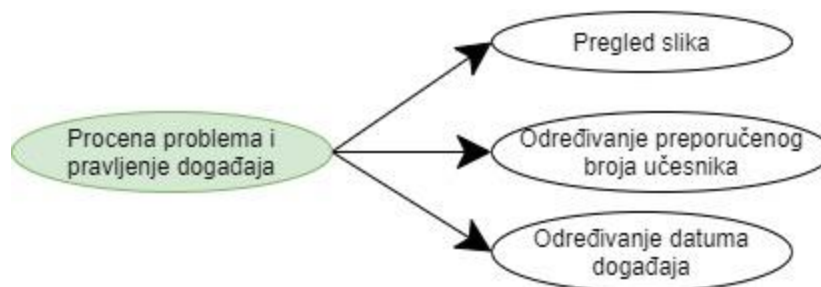
7. Dijagrami slučajeva korišćenja

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



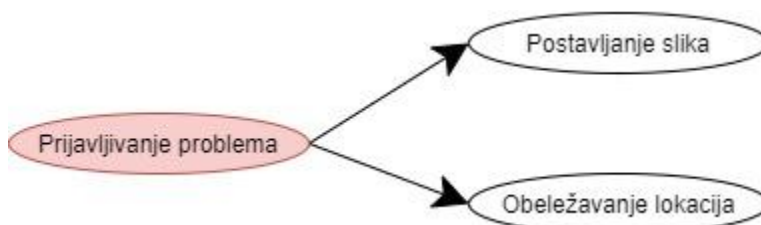
Slučajevi korišćenja *procena problema i pravljenje događaja*, *prijavljivanje problema* i *upravljanje bazom podataka* obuhvataju složenije radnje koje se dalje mogu razložiti na pojedinačne slučajeve korišćenja.

Detaljni UML dijagram za slučaj korišćenja *procena problema i pravljenje događaja* je prikazan na sledećoj slici:

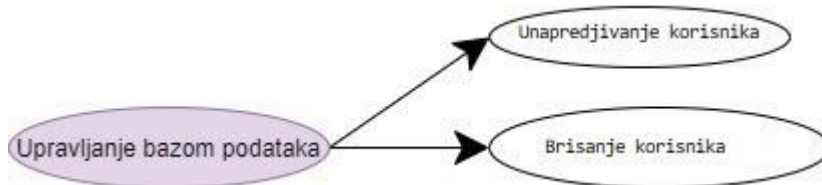


Detaljni UML dijagram za slučaj korišćenja *prijavljivanje problema* je prikazan na sledećoj slici:

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	



Detaljni UML dijagram za slučaj upravljanje *bazom podataka* je prikazan na sledećoj slici:



8. Kratak opis slučajeva korišćenja

8.1 Pregled sadržaja

Kratak opis: Prikaz stranice portala.

Akteri koji iniciraju slučaj korišćenja: Posetilac, Administrator, Organizator i Skaut-čistač.

8.2 Dodavanje organizatora

Kratak opis: Unapređivanje skauta-čistača u organizatore.

Akteri koji iniciraju slučaj korišćenja: Administrator

8.3 Brisanje korisnika

Kratak opis: Brisanje korisnika iz baze podataka.

Akteri koji iniciraju slučaj korišćenja: Administrator.

8.4 Kreiranje naloga

Kratak opis: Kreiranje novog naloga za skauta-čistača

Akteri koji iniciraju slučaj korišćenja: Skaut-čistač.

8.5 Prijavljivanje za učešće na akciji

Kratak opis: Skaut-čistač se prijavljuje na kreirani događaj za akciju.

Akteri koji iniciraju slučaj korišćenja: Skaut-čistač

8.6 Pisanje žalbe

Kratak opis: Ukoliko neki članovi nisu zadovoljni ponašanjem drugih članova, omogućeno je pisanje žalbe organizatorima.

Akteri koji iniciraju slučaj korišćenja: Skaut-čistač i Organizator.

8.7 Postavljanje slika

Kratak opis: Skaut – čistač može postavljati slike na obeleženom mestu na mapi.

Akteri koji iniciraju slučaj korišćenja: Skaut-čistač.

8.8 Obeležavanje lokacija

Kratak opis: Obeležavanje lokacija na kojima se nalazi smeće.

Akteri koji iniciraju slučaj korišćenja: Skaut-čistač.

8.9 Prijavljivanje

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

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

Akteri koji iniciraju slučaj korišćenja: Posetilac, Administrator, Organizator i Skaut-čistač.

8.10 Potvrđivanje učesnika akcija

Kratak opis: Po završetku akcije organizator vodi evidenciju o prisutnim korisnicima.

Akteri koji iniciraju slučaj korišćenja: Organizator.

8.11 Postavljanje slika završenog posla

Kratak opis: Organizator postavlja slike posle završene akcije.

Akteri koji iniciraju slučaj korišćenja: Organizator.

8.12 Pregled slika

Kratak opis: Organizator pregleda slike koje su skaut-čistači postavljali.

Akteri koji iniciraju slučaj korišćenja: Organizator.

8.13 Određivanje preporučenog broja učesnika

Kratak opis: Na osnovu slika površina koja trebaju biti očišćena, organizator procenjuje broj učesnika potrebnih za učešće na akciji.

Akteri koji iniciraju slučaj korišćenja: Organizator.

8.14 Određivanje datuma događaja

Kratak opis: Pri kreiranju događaja, određuje se datum kada će biti održana akcija.

Akteri koji iniciraju slučaj korišćenja: Organizator.

9. 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 za Trashee portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

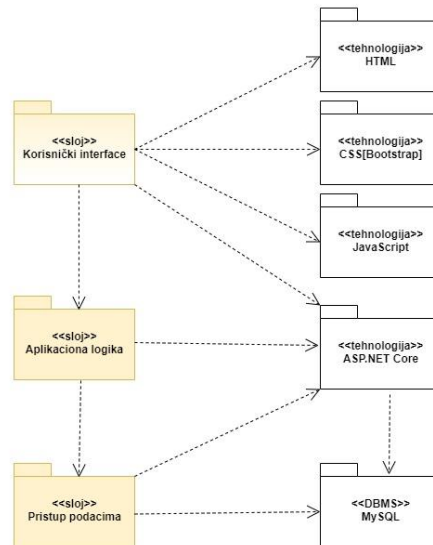
Paket *Korisnički interfejs* sadrži Web stranice, JS skripte, Bootstrap komponente i multimedijalni sadržaj koji realizuju grafički dizajn i forme preko kojih korisnici sistema komuniciraju sa sistemom.

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

Paket *Pristup podacima* sadrži ASP.NET Core fajlove koji predstavljaju interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

9.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve



9.1.1 Korisnički interfejs

Ovaj sloj realizuje korisnički interfejs web aplikacije. U njemu je sadržan HTML, multimedijalni sadržaji, CSS u vidu Bootstrapa, .NET Core i JavaScript koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom. Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML, JavaScript i CSS.

9.1.2 Aplikaciona logika

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Trashee web aplikacije. Sadrži Razor stranice koje realizuju funkcionalnost karakterističnu za domen primene portala i uspostavlja vezu između korisničkog interfejsa i sloja za pristup podacima. Ovaj sloj zavisi od sloja za pristup podacima i ASP.Net Core paketa.

9.1.3 Pristup podacima

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži .NET Core zadužen za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u SQL Server bazi podataka. Ovaj sloj ne zavisi od drugih slojeva, ali je zavistan od paketa ASP.NET i SQL Server baze podataka.

9.1.4 HTML

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.

9.1.5 JavaScript

Tehnologija JavaScript, 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.

9.1.6 Bootstrap

Bootstrap je poznat CSS framework usmeren ka front-end razvoju. Bootstrap sadrži HTML i JavaScript šeme za različite elemente jedne web stranice.

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

9.1.7 SQL Server

DBMS

SQL Server predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju Trashee aplikacije.

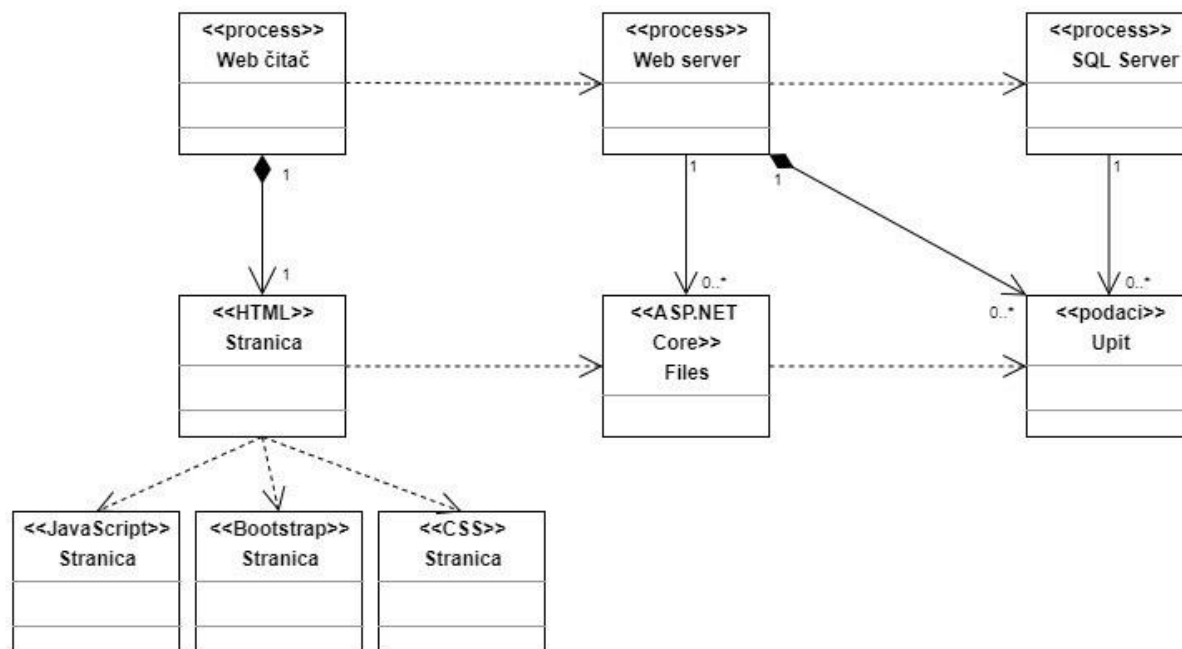
10. Pogled na procese

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.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje Trashee Web aplikacije.

10.1 Procesi

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



10.1.1 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.

10.1.2 Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača. Web server dobija zahteve od više Web klijenata, zatim se obraća Aplikacionom serveru(ASP.NET) koji sadrži logiku koja obrađuje zahteve. Kada se ovi procesi završe, vraćaju se odgovarajući podaci za dati zahtev Web serveru koji generiše HTML, JavaScript i CSS fajlove kako bi se generisala stranica na klijentovoj strani.

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

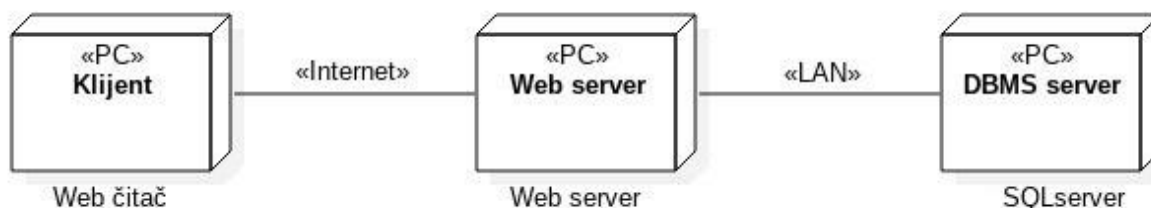
10.1.3 SQL Server

SQL Server je proces koji izvršava funkcionalnost SQL 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.

11. 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 Trashee aplikacije.



11.1 Klijent

Pristup Trashee 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.

11.2 Web server

Računar na kome se izvršava Web server opslužuje više klijenata koji pristupaju preko Interneta. U najopštijoj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom u lokalnoj mreži (LAN).

11.3 DBMS server

DBMS server je računar na kome se izvršava SQL 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).

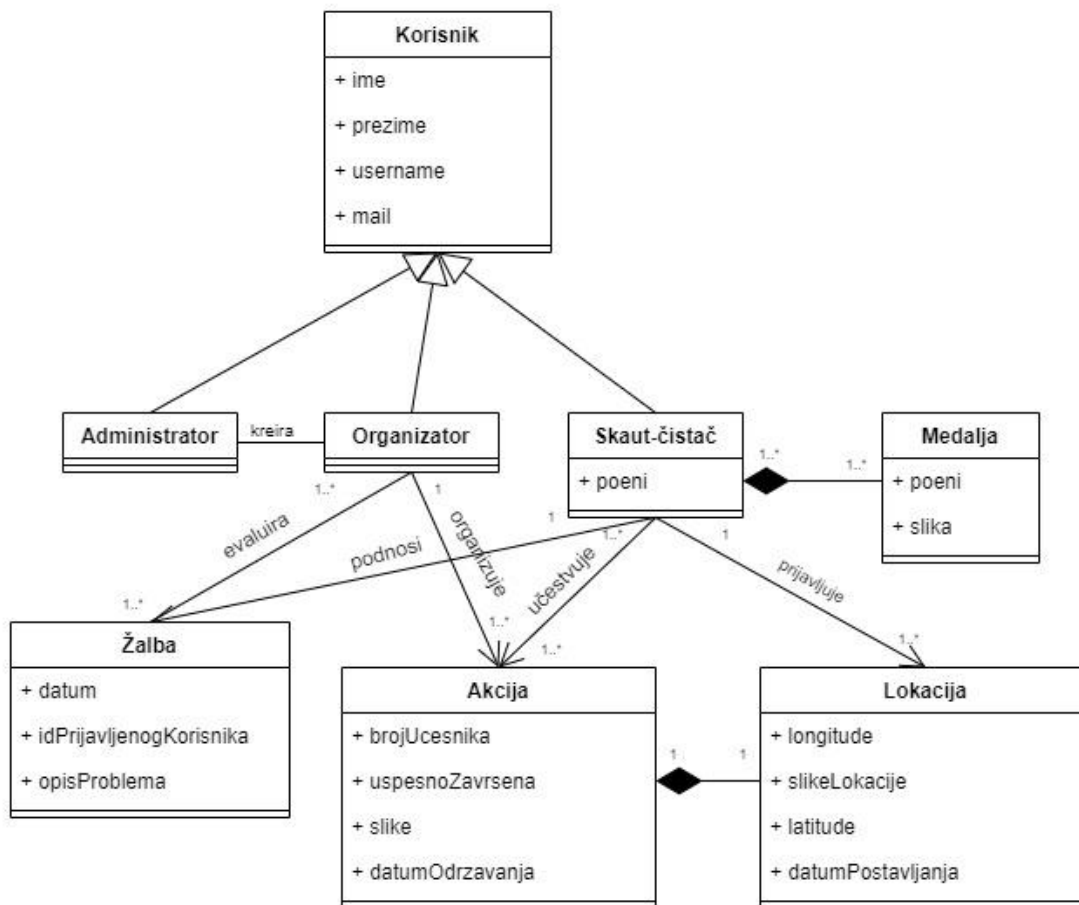
12. Pogled na implementaciju sistema

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

12.1 Model domena

Model domena za koji se Trashee Web 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 koje će biti implementirane.

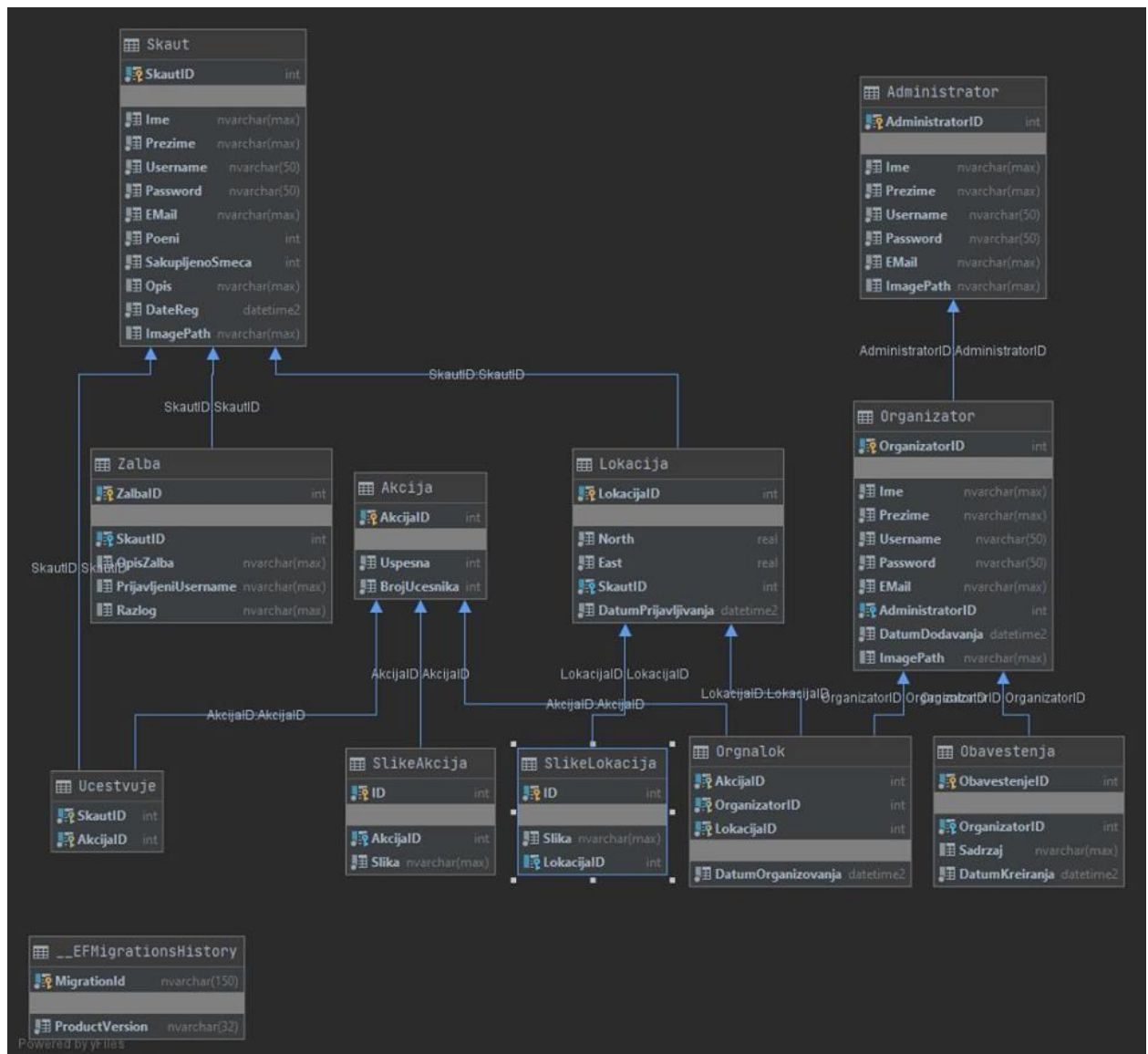
Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	



12.2 Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem *Datagrip*, dok je migracija na SQL Server obavljena pomoću EF Core-a.

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	



12.3 Komponente sistema

Komponente Web aplikacije Trashee su Razor stranice čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je Razor strana prikazana kao klasa atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije definisane u okviru skripta.

12.3.1 Komponente korisničkog interfejsa

Fajlovi sa ekstenzijom .cshtml koriste princip Separation of Concerns što znači da u jednom mogu da se nalaze HTML, CSS, JavaScript and C#.

Folder **wwwroot** sadrži fajlove koji su zaduženi za generisanje statičkog sadržaja ove aplikacije. Ovaj folder sadrži foldere **css**, **javascript** i **lib**. **css** sadrži stilove koji se mogu iskoristiti u aplikaciji, **javascript** sadrži razne skripte koje se mogu uključiti u stranicu i **lib** sadrži bootstrap komponente, fontove i ostale biblioteke koje se koriste za dizajn stranice. U ovom folderu će biti smeštene i slike koje će korisnici postavljati.

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

Svaki Razor page ima dve komponente: Page View i Page Model. Page View je zadužen za prikazivanje i formatiranje sadržaja stranica i ima ekstenziju .cshtml. Page Model je zadužen za pribavljanje podataka iz baze podataka i za čuvanje podataka koji su pribavljeni iz baze, a koje je potrebno prikazati. Page Model ima ekstenziju .cshtml.cs

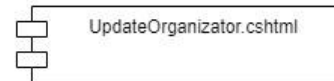
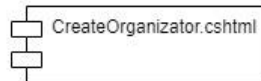
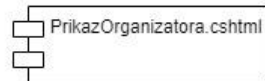
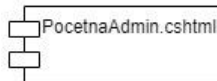
Dizajn korisničkog interfejsa za *sve korisnike* je obuhvaćen stranicama sa ekstenzijom .cshtml

Komponenta
index.html



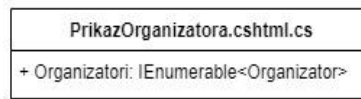
implementira početnu stranicu koju će videti svi neregistrovani i neulogovani korisnici koji dodju na sajt. Ova stranica će voditi na dve druge stranice dostupne ovim korisnicima: **login.cshtml** i **register.cshtml**. Komponenta **login.cshtml** služi za logovanje na sajt. U zavisnosti od tipa korisnika, prikazaće se odgovarajuće početne stranice za taj tip. Komponenta **register.cshtml** je stranica koja će sadržati formu za kreiranje naloga. Posle kreiranja naloga, ova stranica će preusmeriti korisnika na **login.cshtml**

Dizajn korisničkog interfejsa za korisnika Administrator je obuhvaćen stranicama sa ekstenzijom .cshtml.



Komponenta **PocetnaAdmin.cshtml** prikazuje dobrodošlicu korisniku nakon logovanja i sadrži elemente koji vode na druge stranice koje se tiču korisničkog interfejsa administratora.

Komponenta **PrikazOrganizatora.cshtml** prikazuje tabelu sa svim postojećim organizatorima. Podaci potrebni da bi se ova stranica prikazala su predstavljeni sledecim klasnim dijagramom:



Komponenta **CreateOrganizator.cshtml** prikazuje formu za kreiranje organizatora sa odgovarajućim poljima. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledecim klasnim dijagramom:

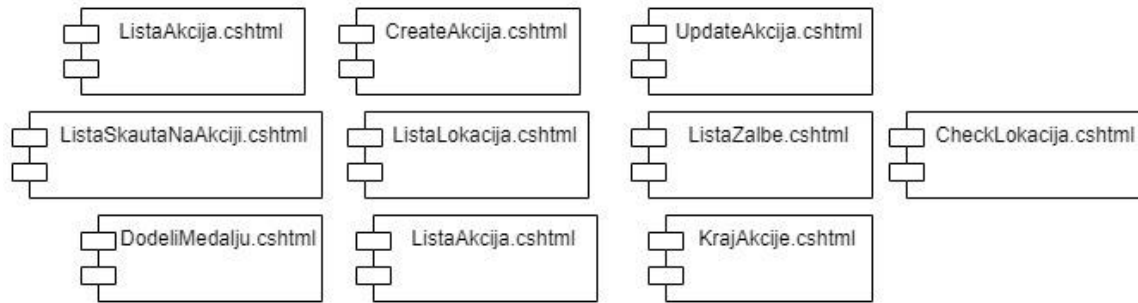


Komponenta **UpdateOrganizator.cshtml** prikazuje formu za ažuriranje podataka o organizatoru od strane admina. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledecim klasnim dijagramom:

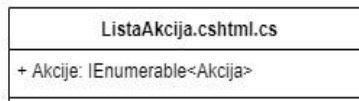


Dizajn korisničkog interfejsa za *Organizator* je obuhvaćen stranicama sa ekstenzijom .cshtml

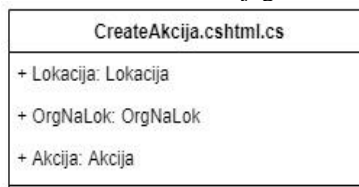
Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	



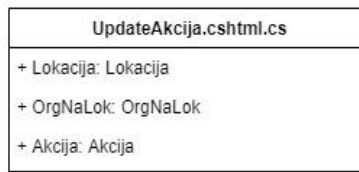
Komponenta **ListaAkcija.cshtml** prikazuje listu svih organizovanih akcija. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledecim klasnim dijagramom:



Komponenta **CreateAkcija.cshtml** prikazuje formu za kreiranje akcija. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledecim klasnim dijagramom:



Komponenta **UpdateAkcija.cshtml** prikazuje formu za ažuriranje podataka o akciji. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledecim klasnim dijagramom:



Komponenta **ListaSkautaNaAkciji.cshtml** prikazuje listu prijavljenih skauta na akciji. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledecim klasnim dijagramom:



Komponenta **ListaLokacija.cshtml** prikazuje spisak svih prijavljenih lokacija sa smećem. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledecim klasnim dijagramom:



Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

Komponenta **CheckLokacija.cshtml** prikazuje stranu koja sadrži osnovne podatke o akciji. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledecim klasnim dijagramom:



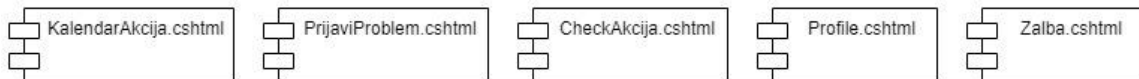
Komponenta **ListaZalbe.cshtml** prikazuje spisak svih nerešenih žalbi. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledecim klasnim dijagramom:



Komponenta **KrajAkcije.cshtml** prikazuje stranicu koja sadrži formu za dodavanje slika kao i tabelu prijavljenih učesnika na akciju gde će organizator moći da potvrdi prisustvo skauta-čistača na akciji. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledecim klasnim dijagramom:



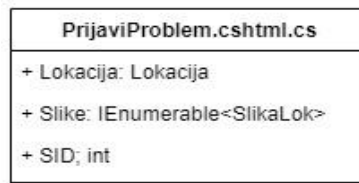
Dizajn korisničkog interfejsa za *Skaut-čistač* je obuhvaćem stranicama sa ekstenzijom .cshtml



Komponenta **KalendarAkcija.cshtml** prikazuje kalendar sa predstojećim i završenim akcijama. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledećim klasnim dijagramom:

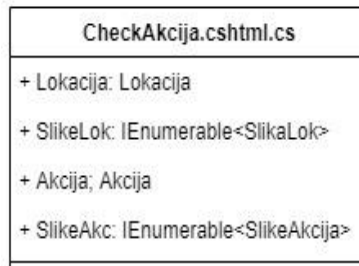


Komponenta **PrijaviProblem.cshtml** prikazuje formu za prijavljivanje lokacije sa problemom. Na ovoj stranici se nalazi mapa na kojoj će korisnik moci da obeleži lokaciju, kao i forma za unos slika. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledećim klasnim dijagramom:

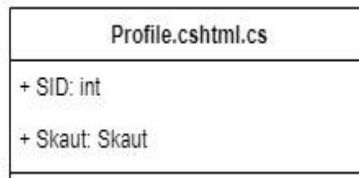


Komponenta **CheckAkcija.cshtml** prikazuje sve informacije o akciji i omogućava skautu da se prijavi na akciju. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledećim klasnim dijagramom:

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	



Komponenta **Profile.cshtml** prikazuje stranicu na kojoj će korisnik moći da menja svoje podatke. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledecim klasnim dijagramom:



Komponenta **Zalba.cshtml** prikazuje stranicu sa formom za kreiranje žalbe. Podaci potrebni da bi se namena ove stranice ostvarila prikazani su sledećim klasnim dijagramom:



12.3.2 Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **index.cshtml** komponente korisničkog interfejsa. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

PrikazOrganizatora.cshtml.cs
+ OnGetAllOrganizatori(): async Task
+ OnPostDeleteOrganizator(int id): async Task<IActionResult>

Navedene funkcije obavljaju sledeće zadatke:

- OnGetOrganizatori() – vraća sve organizatore iz baze podataka
- OnPostDeleteOrganizator(int id) – briše organizatora sa odgovarajućim id-em

CreateOrganizator.cshtml.cs
+ OnPostCreateOrganizator(): async Task<IActionResult>

Navedene funkcije obavljaju sledeće zadatke:

- OnPostCreateOrganizator() – kreira organizatora

UpdateOrganizator.cshtml.cs
+ OnPostUpdateOrganizator(int id): async Task<IActionResult>

Navedene funkcije obavljaju sledeće zadatke: OnPostUpdateOrganizator(int id) – ažurira organizatora novim podacima

- OnPostUpdateOrganizator(int id) – ažurira organizatora novim podacima

ListaAkcija.cshtml.cs
+ OnGetListaAkcija(): async Task

Navedene funkcije obavljaju sledeće zadatke: OnGetListaAkcija(int id) – vraća listu akcija

- OnGetListaAkcija(int id) – vraća listu akcija

CreateAkcija.cshtml.cs
+ OnGetLokacija(int lid): async Task
+ OnPostOrgNaLok(): async Task<IActionResult>
+ OnPostAkcija(): async Task<IActionResult>

Navedene funkcije obavljaju sledeće zadatke: OnGetLokacija(int lid) – vraća lokaciju iz baze podataka

- OnGetLokacija(int lid) – vraća lokaciju iz baze podataka

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

- OnPostOrgNaLok() – upisuje u tabelu OrgNaLok(ternarna veza izmedju organizatora, lokacije i akcije) podatke
- OnPostAkcija() – kreira akciju sa prosleđenim podacima

ListaSkautaNaAkciji.cshtml.cs
+ OnGetListaSkautaNaAkciji(int aid): async Task<IActionResult>

Navedene funkcije obavljaju sledeće zadatke:

- OnGetListaSkautaNaAkciji(int aid) – vraća podatke o skautima na odabranoj akciji

ListaLokacija.cshtml.cs
+ OnGetListaLokacija(): async Task

Navedene funkcije obavljaju sledeće zadatke: OnGetListaLokacija() – vraća sve prijavljene lokacije

- OnGetListaLokacija() – vraća sve prijavljene lokacije

ListaZalbe.cshtml.cs
+ OnGetListaZalbe(): async Task

Navedene funkcije obavljaju sledeće zadatke:

- OnGetListaZalbe() – vraća listu svih žalbi

CheckLokacija.cshtml.cs
+ OnGetLokacija(int id): async Task
+ OnGetSlikeLokacije(): async Task
+ OnGetSkaut(): async Task

Navedene funkcije obavljaju sledeće zadatke: OnGetLokacija(int id) – vraća lokaciju iz baze sa zadatim id-em

- OnGetLokacija(int id) – vraća lokaciju iz baze sa zadatim id-em
- OnGetSlikeLokacije() – vraća slike iz baze podataka za datu lokaciju
- OnGetSkaut() – vraća skauta iz baze podataka koji ima vrednost sid iz lokacije

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

KrajAkcije.cshtml.cs
+ OnPostDodajSlike(int sid): async Task<ActionResult>
+ OnPostPotvrdiUcesnike(): async Task

Navedene funkcije obavljaju sledeće zadatke: OnPostDodajSlike(int oid) – slike završene akcije se upisuju u bazu podataka

- OnPostDodajSlike(int oid) – slike završene akcije se upisuju u bazu podataka
- OnPostPotvrdiUcesnike() – u bazu podataka se upisuje informacija o potvrdi učesnika akcije

KalendarAkcija.cshtml.cs
+ OnGetListaAkcija(): async Task
+ OnGetVremeOdrzavanja(int akid): async Task

Navedene funkcije obavljaju sledeće zadatke: OnGetListaAkcija() – vraća listu akcijama

- OnGetListaAkcija() – vraća listu akcijama
- OnGetVremeOdrzavanja(int akid) – vraća datum i vreme održavanja akcije iz baze podataka

PrijaviProblem.cshtml.cs
+ OnPostLokacija(int sid): async Task<ActionResult>
+ OnPostSlike(): async Task<ActionResult>

Navedene funkcije obavljaju sledeće zadatke:

- OnPostLokacija(int sid) – u tabelu lokacija se upisuju unešeni podacima
- OnPostSlike() – u tabelu slikeLok se upisuju slikeLok

CheckAkcija.cshtml.cs
+ OnGetLokacija(int lid): async Task
+ OnGetAkcija(int aid): async Task

Navedene funkcije obavljaju sledeće zadatke: OnGetLokacija(int lid) – vraća sve podatke o lokaciji iz baze podataka

- OnGetLokacija(int lid) – vraća sve podatke o lokaciji iz baze podataka
- OnGetAkcija(int aid) – vraća sve podatke o akciji iz baze podataka. Vraća i slike ukoliko je akcija uspešno završena, ukoliko nije samo vraća osnovne informacije.

Trashee	Verzija: 1.0
Arhitekturni projekat	Datum: 23.04.2020. godine
SWE-Trashee-05	

Profile.cshtml.cs
+ OnPostUpdateProfile(): async Task<IActionResult>
+ OnGetSkaut(int sid): async Task

Navedene funkcije obavljaju sledeće zadatke: OnPostUpdateProfile() – ažurira podatke u bazi podataka unetim podacima

- OnPostUpdateProfile() – ažurira podatke u bazi podataka unetim podacima
- OnGetSkaut(int sid) – vraća skauta iz baze podataka

Zalba.cshtml.cs
+ OnPostCreateZalba(int Sid): async Task<IActionResult>

Navedene funkcije obavljaju sledeće zadatke: OnPostCreateZalba(int sid) – kreira žalbu sa unetim podacima i upisuje u bazu podataka

- OnPostCreateZalba(int sid) – kreira žalbu sa unetim podacima i upisuje u bazu podataka

Register.cshtml.cs
+ OnPostRegister(): async Task<IActionResult>

Navedene funkcije obavljaju sledeće zadatke: OnPostRegister() – upisuje u bazu podataka iz forme registracija

- OnPostRegister() – upisuje u bazu podataka iz forme registracija

13. Performanse

Izabrana arhitektura softvera podržava zahteve u pogledu broja korisnika koji mogu simultano pristupiti 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].

14. Kvalitet

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

1. Web aplikacija Trashee 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.