
WeSketch Team

Stevica Stojković, 15426

Stefan Stanković, 15382

WeSketch

Online alat za kolaborativno vektorsko crtanje

Sadržaj

Sadržaj	0
Uvod	1
Funkcionalnosti	1
Nefunkcionalni zahtevi i ograničenja	1
Arhitekturni zahtevi	2
Arhitektura sistema	3
Biblioteke i aplikacioni okviri	3
Arhitekturni dizajn	4
Pogled komponenti	4
Fizički pogled	4
Pogled na procese	5
Slučajevi korišćenja	5

Uvod

U ovom dokumentu dat je opis arhitekture sistema, domena aplikacije, funkcionalnih i nefunkcionalnih zahteva.

Funkcionalnosti

WeSketch je online alat za kolaborativno vektorsko crtanje, osnovne funkcionalnosti sistema su:

- Izrada vektorskih crteža pomoću 2D grafičkih primitiva (elipsa, pravougaonik, linija, tekst)
- Grupisanje grafičkih elemenata
- Istovremeni rad više korisnika na jednom crtežu (u jednom trenutku jedan korisnik uređuje)
- Kreiranje i upravljanje dokumentima
- Dodavanje i uklanjanje kolaboratora
- Časkanje sa kolaboratorima
- Pregled crteža koji su korisnici označili kao javni
- Dodavanje crteža u listu omiljenih
- Snimanje dokumenata u relacionu bazu podataka
- Ograničen rad u *offline* modu
- Snimanje u pdf

Nefunkcionalni zahtevi i ograničenja

WeSketch sistem treba da odgovori na sledeće nefunkcionalne zahteve (atribute kvaliteta):

- Reupotrebljivost (reusability)
- Proširljivost (extensibility)
- Pouzdanost (reliability)
- Lakoća korišćenja (usability)
- Lakoća testiranja (testability)
- Otpornost na greške (fault tolerance)
- Performanse

- Otvoren kod

Ograničenja sistema su:

- Izvšenje klijentske aplikacija zahteva Windows 7+ operativni sistem kao i .NET framework 4.5+. Za rad aplikacije je potrebna internet konekcija
- Serverski deo sistema se obavlja na Windows platformi

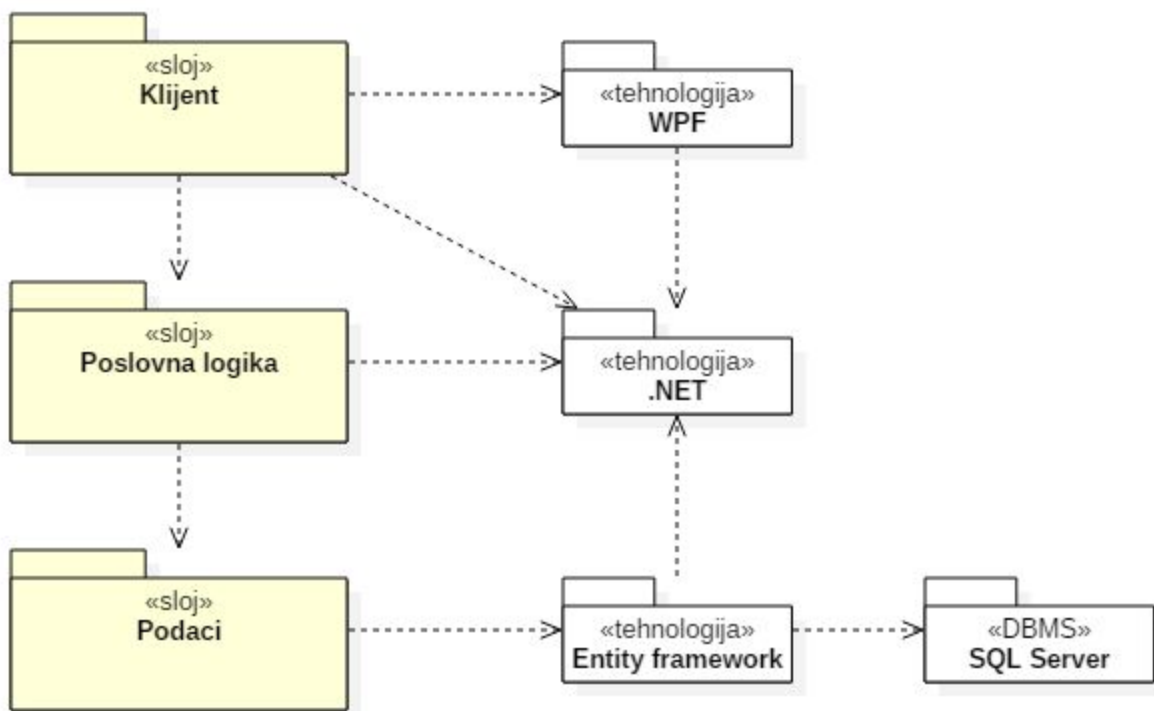
Arhitekturni zahtevi

Na osnovu navedenih nefunkcionalnih zahteva i ograničenja, imamo sledeće arhitekturne zahteve:

- Reupotrebljivost: komponente sistema su slabo spregnute kako bi mogle da se koriste za druge sisteme ili zamene
- Proširljivost: Arhitektura aplikacije mora da bude takva da omogući lako dodavanje novih funkcionalnosti na bazi trenutne aplikacije.
- Pouzdanost: Aplikacija mora da pokaže poslednje stanje dokumenta (crteža)
- Lakoća korišćenja: Aplikacija treba da obezbedi prijatan i intuitivan interfejs
- Lakoća testiranje: S obzirom da su komponente slabo spregnute moguće je brzo i lako lokalizovati greške, odnosno proveriti ispravnost.
- Otpornost na greške: U slučaju loše akcije korisnika ili u slučaju da je sistem nedostupan treba korisniku prikazati odgovarajuću poruku. Deo aplikacije može da funkcioniše i bez internet konekcije.
- Performanse: Aplikacija treba novonastale promene na crtežu da propagira korisnicima koji nadgledaju crtež u roku od 3 sekunde.
- Otvoren kod: Kod aplikacije treba da bude dostupan na javnom repozitorijumu na GitHub-u.

Arhitektura sistema

WeSketch sistem implementira *Layered* arhitekturni obrazac zbog toga što je standardno rešenje za ovakav tip aplikacije. Čine ga tri sloja: klijentski sloj, sloj poslovne logike i sloj sa podacima.



Klijentski sloj: Uloga ovog sloja je da omogući korisniku interakciju sa sistemom. U pitanju je C# .NET desktop aplikacija gde je za interfejs zadužena MahApps.Metro biblioteka za Windows Presentation Foundation (WPF). Klijentska aplikacija je izrađna po MVC arhitekturnom obrascu što omogućuje jasno izdvajanje modela i obradu podataka od prezentacionog dela aplikacije.

Sloj poslovne logike: U ovom sloju je realizovana logika za obradu zahteva klijenata i obradu podataka. Usluga klijentima je omogućena putem API-a izrađenog u ASP.NET-u. Ovaj sloj koristi usluge sloja podataka za pristup podacima.

Sloj podataka: Sloj je zadužen za komunikaciju sa bazom podataka. Samo u ovom sloju se direktno pristupa podacima u bazi. Objektno-relaciono mapiranje je izvršeno pomoću Entity Framework-a.

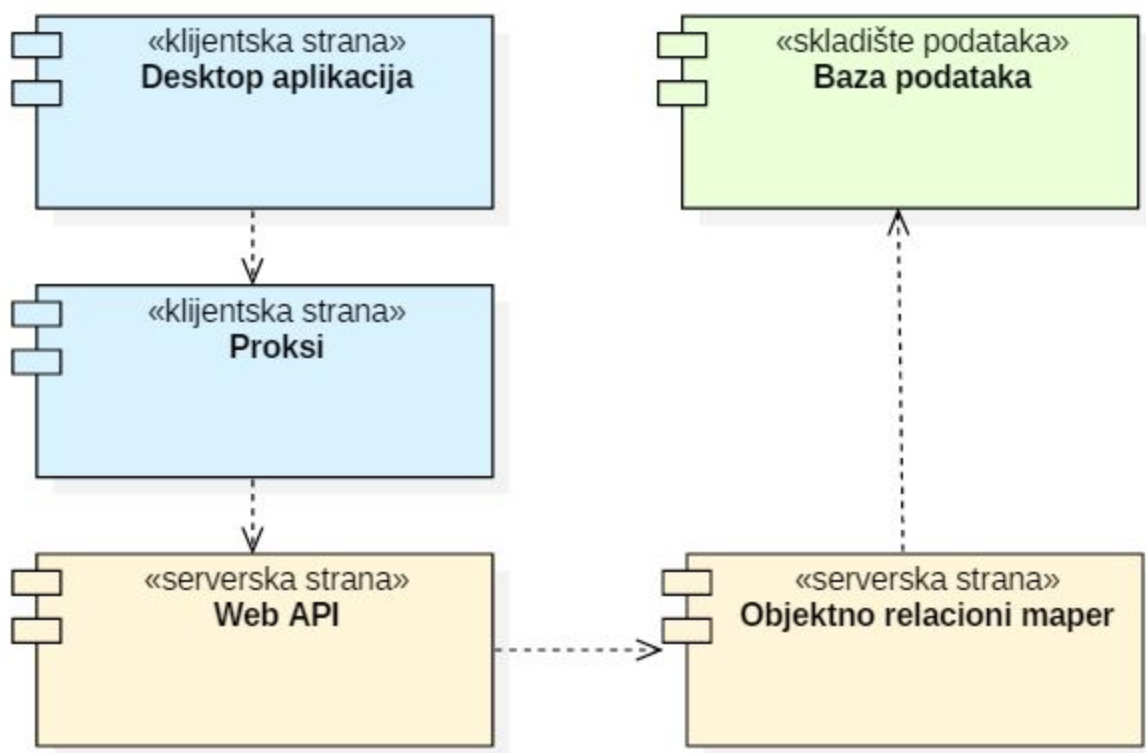
Biblioteke i aplikacioni okviri

Sistem koristi:

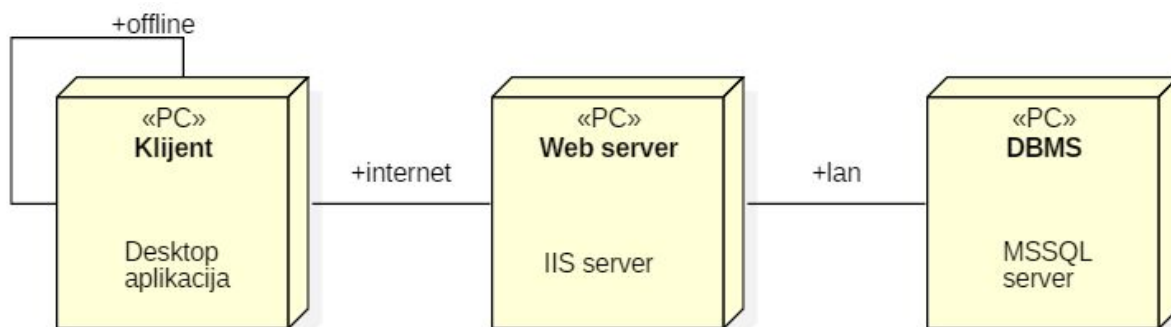
- .NET: Koristi se kao osnovni aplikacioni okvir na kome se izvršava i klijent i server
- Entity framework: Služi za ORM
- SignalR: Biblioteka zadužena za asinhronu komunikaciju između klijenta i servera
- ASP.NET WebApi: Služi za realizaciju API-a (serverski deo aplikacije)
- MahApps.Metro: *Toolkit* za korisnički interfejs namenjen za WPF grafički podsistem

Arhitekturni dizajn

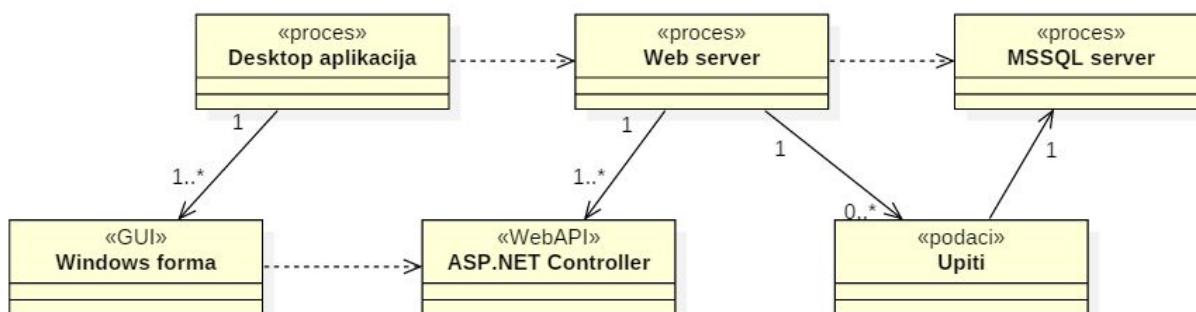
Pogled komponenti



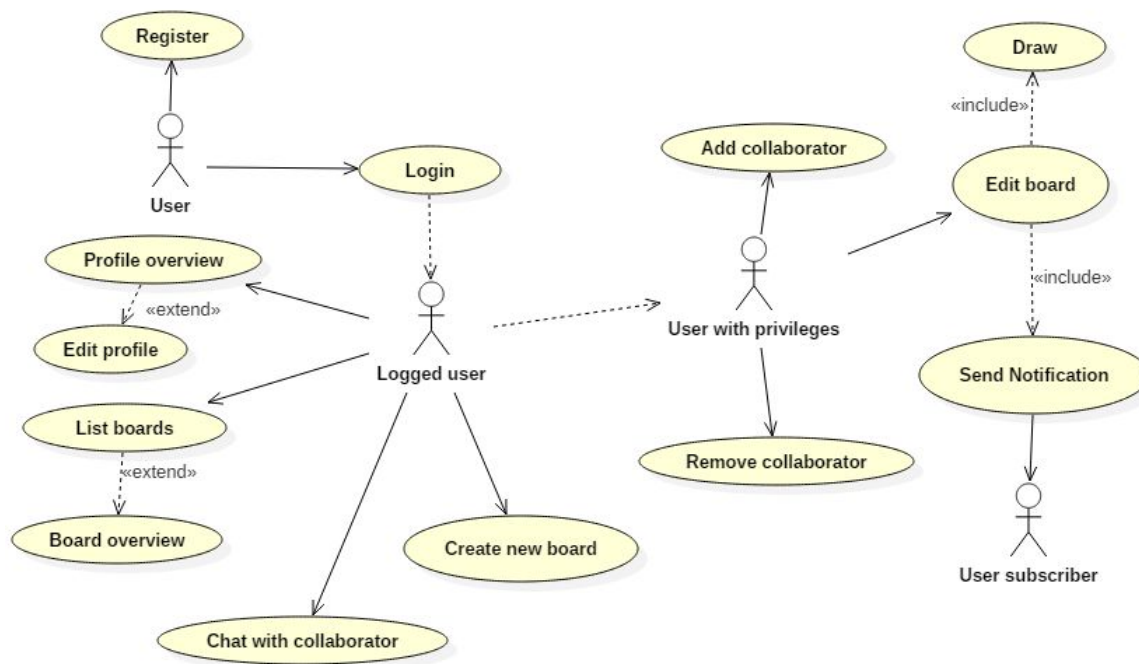
Fizički pogled



Pogled na procese



Slučajevi korišćenja



Registracija(Register) - potrebno je da se korisnici koji nemaju nalog registruju na sistem.

Prijava(Login) - neophodno je prijaviti se kako bi se koristile usluge sistema.

Pregled i uređivanje profila(Profile overview, Edit profile) - korisnik može da pogleda podatke o svom nalogu a onda ih i uređuje.

Pregled liste tabli i konkretne table(List board, Board overview) - korisnik ima mogućnost da pregleda sve table(dokumente) na kojima radi kao i da pogleda detalje o određenoj tabli.

Ćaskanje(Chat with collaborators) - korisnik može da šalje i prima poruke od drugih korisnika koji uređuju tablu.

Kreiranje table(Create board) - korisnik ima mogućnost da kreira tablu(dokument).

Dodavanje kolaboratora(Add collaborator) - korisnik može da dodaje saradnike i sa njima uređuje dokument.

Uklanjanje kolaboratora(Remove collaborator) - korisnik može da ukloni kolaboratora sa dokumenta.

Uređivanje table(Edit board) - korisnik može da uređuje tablu. U uređivanje table spada crtanje(**Draw**) i obaveštavanje ostalih saradnika na dokumentu o promenama(**Send Notification**).