Elektronski Fakultet Niš Djordje Lazić 17722

Mladen Antić 17545

Arhitektura i projektovanje softvera

Projekat faza 1

1. Kontekst i cilj softverskog projekta

**Kontekst:**

Geometry Solver je aplikacija namenjena rešavanju geometrijskih problema na osnovu unetih parametara i iscrtavanju rešenja. Korisnici mogu uneti geometrijske figure i tražiti rešenja za različite probleme, kao i vizualizaciju tih rešenja.

**Cilj:**

Cilj projekta je stvoriti efikasan alat za rešavanje geometrijskih problema, omogućavajući korisnicima da brzo dobiju tačne rezultate I odgovarajući prikaz.

2. Arhitekturni zahtevi

### Funkcionalni Zahtevi:

Unos Geometrijskih Parametara:

Korisnici mogu unositi različite parametre za različite geometrijske oblike (npr. dužina, širina, radijus).

Rešavanje Geometrijskih Problema:

Sposobnost sistema da rešava različite vrste geometrijskih problema, uključujući izračunavanje površine, obima, uglova i slično.

Vizualizacija Rešenja:

Prikaz grafičkih reprezentacija geometrijskih rešenja na korisničkom interfejsu.

### Ne-funkcionalni Zahtevi (Atributi Kvaliteta):

Performanse:

Sistem mora pružati odgovor pravovremeno nakon svakog unosa korisnika.

Pouzdanost:

Stopa grešaka prilikom rešavanja problema treba biti što manja.

Korisničko Iskustvo:

Grafički interfejs mora biti intuitivan i lako razumljiv za širok spektar korisnika.

### Tehnička Ograničenja:

Jezici Programiranja:

Klijentska strana će koristiti JavaScript sa React.js, dok će serverska strana koristiti NodeJs.

Baza Podataka:

Korišćenje relationalne baze podataka (npr. MongoDB) za skladištenje korisničkih podataka i rezultata.

### Poslovna Ograničenja:

Besplatan Pristup Osnovnim Funkcionalnostima:

Osnovne funkcionalnosti aplikacije, uključujući unos i rešavanje problema za osnovne geometrijske oblike, trebaju biti besplatne za sve korisnike.

3. Arhitekturni Dizajn

### Arhitekturni Obrasci:

1) Model-View-Controller(MVC):

Model - Logika koja se koristi za rešavanje geometrijskih problema.

View - Interfejs za korisnike gde se unose ulazni parametri.

Controller – koordinacija/interakcija između Model i View komponenti.

2) Repository Obrasac:

Služi za rad sa bazom podataka i upravljanje podacima o geometrijskim figurama i omogućava jednostavan pristup istim podacima.

3) Observer obrazac:

Prati promene u podacima i ažurira ih. U MVC obrazcu uređuje vezu između View dela i Model-a . Na osnovu promena podataka ažurira se grafički prikaz na interfejsu.

??? Client/Server :

Klijentski deo radi sa interfejsom I korisničkim iskustvom , dok je serverski deo zadužen za logiku potrebnu za izračunavanje Površina i Obima tela i za njihovo iscrtavanje.

Generalna Arhitektura (Box-Line Dijagram):

### 

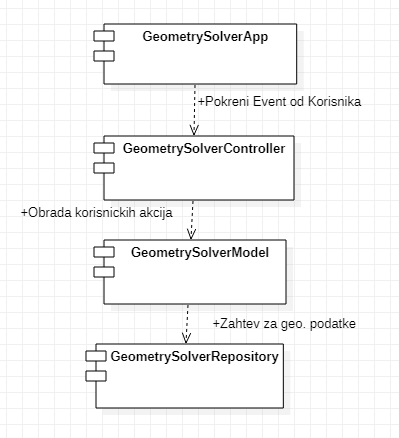
Klijentski interfejs – Komponenta koja je zadužena za komunikaciju sa korisnicima . Odnosi se na unos parametara za geometrijske probleme i prikaz rešenja.

Controller (Aplikaciona Logika) – Controller je posrednička komponenta koja upravlja komunikacijom između Klijentskog interfejsa i Modela i radi sa događajima.

Model (Logika za rešavanje) – Prima podatke od korisničkog interfejsa i obrađuje ih (rešava geometrijske probleme).

Repository (Rad sa podacima) – Čuva podatke o geometrijskim figurama. Služi kao veza između modela i skladišta podataka.

Strukturni Dijagram (Pogled):



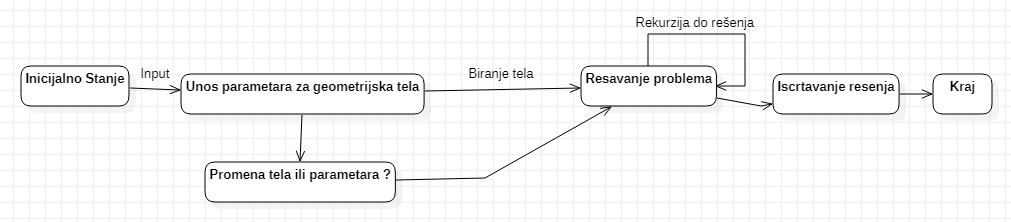
Geometry Solver App – Glavni modul aplikacije , sastoji se od View i Controller komponente.

Geometry Solver Controller – Kontroleri koji se bave obradom korisničkih akcija.

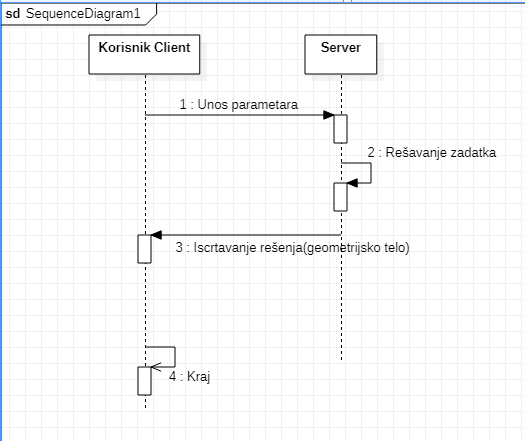
Geometry Solver Model – Logika za rešavanje geometrijskih modela.

Geometry Solver Repository – Upravljanje podacima , skladištenje podataka.

Bihejvioralni Dijagram (Pogled):



Dijagram aktivnosti za geometrijski problem – Na početku se unose parametri i bira se geometrijsko telo . U toku rada može se promeniti telo ili podaci o istom (visina , širina, prečnik ...). Prvo se reši problem pa se tek onda vrši iscrtavanje .



Dijagram sekvence – kao i kod dijagrama aktivnosti , na početku se unose parametri , na osnovu njih se rešava problem i nakon toga se iscrtava geometrijsko telo .

4. Biranje okvira (framework) i biblioteka (library)

Klijentski deo:

Biblioteka – React.js se koristi za obrazac MVC , komponente predstavljaju View sloj , a Redux se koristi kao Controller.

Serverska strana:

Framework – ExpressJs , NodeJS – Takođe se koristi za MVC

Biblioteka – MongoDB – Koristi se kao biblioteka za NodeJs .

MVC – ReactJs(client) ,Nodejs(server)

Observer – Express.js(server)

Repository – MongoDB(server)