



SPECIFIKACIJA DIZAJNA SOFTVERA

PROJEKAT WEMUS

JOVAN JEROTIJEVIĆ
TEODORA MILJEVIĆ
LAZAR STEFANOVIĆ
DUŠAN MIJAILOVIĆ
ĐORĐE MARKOVIĆ
STRAHINJA MILOSAVLJEVIĆ

TIM: M3.JS

JUN 2016.

Sadržaj

Istorija dokumenta	4
1. Uvod	5
1.1. Namena dokumenta.....	5
1.2. Opseg proizvoda.....	5
1.3. Pregled sistema	5
1.4. Lista materijala	6
1.5. Definicije, akronimi i skraćenice.....	7
1.6. Pregled ostatka dokumenta	8
2. Razmatranje dizajna	9
2.1. Pretpostavke.....	9
2.2. Ograničenja.....	10
2.3. Sistemsko okruženje.....	11
3. Arhitektura sistema	12
3.1. Pregled sistema	12
3.2. Obrazloženje dizajna	13
4. Dizajn visokog nivoa	14
4.1. Konceptualni pogled.....	14
5. Dizajn baze podataka.....	15
6. Dizajn niskog nivoa	17
6.1. Dizajn funkcionalnosti aplikacije	17
6.1.1. Otvaranje stranice „O Wemus-u“	17
6.1.2. Registracija korisnika	18
6.1.3. Prijavljivanje korisnika	20
6.1.4. Unos ulaznih parametara (pokretanje i čuvanje)	22
6.1.5. Pregledanje stranice sa svim proračunima.....	24
6.1.6. Pregled stranice sa sačuvanim proračunima.....	26
6.1.7. Pregled stranice sa proračunima prijavljenog korisnika.....	27
6.1.8. Pregled stranice sa nepročitanim proračunima	28
6.1.9. Pregledanje rezultata proračuna	29
6.1.10. Pretraživanje tabele sa rezultatom	32
6.1.11. Čuvanje rezultata proračuna	35
6.1.12. Ažuriranje korisničkih podataka	37
6.1.13. Administracija korisnika	40
6.1.14. Promena jezika na aplikaciji	41

6.1.15. Odjavljivanje korisnika sa aplikacije	42
6.2. Zavisnosti između modula klijentskog dela.....	43
6.2.2. Zavisnosti između komponenti.....	46
6.3. Dijagrami klasa	50
6.4. Eksperiment REST API.....	62
6.5. Čuvanje proračuna	62
6.6. Brisanje proračuna	63
6.7. Čitanje svih proračuna	63
6.8. Čitanje korisnikovih proračuna.....	63
6.9. Čitanje nepročitanih proračuna.....	63
6.10. Pokretanje sačuvanog proračuna	64
6.11. Prekid pokrenutog proračuna	64
6.12. Proveravanje postojanja naslova proračuna	65
6.12.1. <i>URL i metod zahteva</i>	65
6.13. Uzimanje komada rezultata proračuna	65
6.5. ResultChunks API	66
6.5.1. Upisivanje rezultata proračuna u bazu.....	66
6.5.2. Brisanje rezultata proračuna iz baze	66
6.5.3. Čitanje komada rezultata iz baze.....	66
6.5.3.1. <i>Potpis funkcije:</i>	66
6.6. WemusUser REST API	67
6.6.1. Listanje svih korisnika:.....	67
6.6.2. Uzimanje podataka o korisniku	67
6.6.3. Provera da li je korisnik administrator	68
6.6.4. Menjanje informacija o korisniku	68
6.6.5. Menjanje role Korisnika.....	68
6.6.6. Brisanje korisnika.....	69
6.7. RunCModel API.....	69
6.7.1. Pokretanje proračuna.....	69
6.7.1.1. <i>Potpis funkcije:</i>	69
6.7.2. Slanje rezultata na wemus server.....	69
6.7.2.1. <i>Potpis funkcije:</i>	69
6.7.2.2. <i>Potpis callback funkcije:</i>	69
7. Dizajn korisničkog interfejsa.....	70
7.2. Stranica za prijavu i registrovanje korisnika	70

7.3.	Fiksirani delovi aplikacije	72
7.4.	Pokretanje proračuna	73
7.5.	Čuvanje i prikazivanje proračuna	78
7.6.	Pregledanje izvršenih proračuna	81
7.7.	Ažuriranje korisničkih podataka	83
7.8.	O projektu i aplikaciji Wemus.....	83
7.9.	Administrativna strana	84

Istorija dokumenta

Datum izmene	Ime člana	Opis izmene	Verzija dokumenta
6.6.2016.	Jovan Jerotijević	Napravljen template dokumenta	0.0
8.6.2016.	Jovan Jerotijević	Tačke 1.1, 1.2 i 1.4 su napisane	0.1
9.6.2016.	Dušan Mijailović	Napisane su tačke 3.1 i 3.2	0.2
9.6.2016.	Teodora Miljević	Napravljeni dijagrami Use Case	0.2
9.6.2016.	Strahinja Milosavljević	Ubačene slike aplikacije	0.2
10.6.2016.	Jovan Jerotijević	Tačke 1.3, 1.5 i 1.6 su napisane	0.3
10.6.2016.	Dušan Mijailović	Napravljeni dijagrami aktivnosti	0.3
11.6.2016.	Teodora Miljević	Napisani dizajn niskog i visokog nivoa	0.4
11.6.2016.	Strahinja Milosavljević	Ubačen prateći tekst za slike aplikacije	0.4
12.6.2016.	Lazar Stefanović	Napisana tačka 5 vezana za bazu	0.5
12.6.2016.	Teodora Miljević	Prilagođeni i ubačeni dijagrami aktivnosti	0.5
12.6.2016.	Jovan Jerotijević	Završena tačka 2 sa svim podnaslovima	0.5
13.6.2016.	Lazar Stefanović	Tabele vezane za bazu su ubačene	0.6
13.6.2016.	Jovan Jerotijević	Završno stilizovanje prve verzije SDS-a	1.0
14.6.2016.	Đorđe Marković	Prepravljene stvari vezane za D3	1.1
15.6.2016.	Jovan Jerotijević	Ispravljene zamerke vezane za tekst	1.2
15.6.2016.	Teodora Miljević	Prepravljene dijagrami Use Case	1.3
16.6.2016.	Teodora Miljević	Napravljeni dijagrami sekvenci	1.4
17.6.2016.	Dušan Mijailović	Napravljeni dijagrami klasa	1.5
17.6.2016.	Lazar Stefanović	Tačke 6.4 6.5 6.6 6.7 napisane	1.6
17.6.2016.	Jovan Jerotijević	Završni detalji za finalnu verziju	2.0

1. Uvod

1.1. Namena dokumenta

Ovaj dokument daje detaljan opis dizajna projekta WEMUS. Aplikacija WEMUS ima za cilj olakšano korišćenje postojećeg softverskog rešenja MUSICO čija je svrha simuliranje rada mišićnih ćelija. Kao takav dokument je namenjen da detaljno opiše način rešavanja zahteva koji su navedeni u dokumentu „Specifikacija softverskih zahteva za projekat WEMUS“.

1.2. Opseg proizvoda

Projekat WEMUS je softversko rešenje realizovano kao web aplikacija čiji je glavni cilj pojednostavljenje korišćenja programa MUSICO. Softver MUSICO je već postojeće softversko rešenje čija je namena simuliranje rada mišićnih ćelija sa zadatim parametrima.

Projekat WEMUS na prvom mestu treba da obezbedi određeni nivo apstrakcije programa MUSICO od njenih korisnika. Kao takav treba da obezbedi olakšano zadavanje parametara koji MUSICO koristi u svom radu. Pored zadavanja parametara potrebno je i da pokrene simulaciju koja će u svom radu koristiti zadate parametre. Po završetku simulacije neophodno je da dobijeni rezultati budu dostupni korisniku na prikazivanje. Pored toga korisniku je potrebno ponuditi opciju preuzimanja rezultata na računar.

1.3. Pregled sistema

WEMUS je dizajniran kao softversko rešenje koje treba biti napravljeno kao web aplikacija sa pratećom bazom podataka. U bazi će se čuvati lični podaci korisnika potrebni za korišćenje aplikacije kao i ulazni i izlazni podaci proračuna svih korisnika.

Aplikacija je prvenstveno namenjena već postojećim korisnicima programa MUSICO. Takođe moći će da je koriste i ostali koji imaju potrebu da vrše eksperimente o ponašanju mišićnih ćelija pod različitim uslovima. Njen glavni cilj je da obezbedi intuitivno rešenje za upotrebu programa MUSICO. Koristeći WEMUS korisnik više neće morati sam da pravi ulazne tekstualne fajlove i manuelno pokreće MUSICO, već će sistem WEMUS to za njega odraditi.

U okviru sistema potrebno je imati tri vrste korisnika:

1. Neregistrovan korisnik – moći će da pročita tekst o projektu WEMUS i da pogleda priloženi snimak o korišćenju aplikacije WEMUS i tome šta će mu ona ponuditi ukoliko se registruje.
2. Registrovan korisnik - imaće mogućnost zadavanja parametara za simulaciju, čuvanja parametara i pokretanja simulacije, pregleda rezultata svojih i tuđih proračuna, brisanja svojih proračuna, izmene svojih ličnih podataka i mogućnosti koje ima neregistrovan korisnik.
3. Administrator aplikacije – imaće sva prava kao i registrovan korisnik sa dodatkom administracije korisnika tj. dodeljivanja prava ostalim korisnicima kao i brisanje korisnika iz sistema i mogućnost brisanja proračuna koji ne pripadaju njemu.

Aplikacija mora da ispuni sledeću listu funkcionalnosti:

1. Obavezne funkcionalnosti
 - a. Mogućnost registracije i prijavljivanja korisnika na sistem i odjavljivanje sa sistema
 - b. Prikazivanje i mogućnost popunjavanja parametara za simulaciju rada mišića koje treba zadati kroz forme grupisane po svrsi parametara
 - c. Validacija zadatih parametara
 - d. Pokretanje simulacije rada mišića sa zadatim parametrima
 - e. Čuvanje rezultata formiranih od strane programa MUSICO u bazi podataka
 - f. Prikazivanje rezultata proračuna i to grafički i tabelarno sa numeričkim podacima
2. Poželjne funkcionalnosti
 - a. Vizuelizacija unešenih parametara na grafičkim prikazima delova sarkomere
 - b. Čuvanje prosleđenih podataka u bazi bez pokretanja proračuna
 - c. Vraćanje ulaznih parametara svojih ili tuđih proračuna iz baze na izmenu i njihovo ponovno čuvanje u bazu ili pokretanje proračuna sa tako izmenjenim podacima
 - d. Obaveštavanje korisnika o statusu izvršavanja simulacije
 - e. Obaveštavanje korisnika o broju nepročitanih rezultata
 - f. Mogućnost izbora pregleda rezultata nekog od svojih proračuna ili od proračuna drugih korisnika
 - g. Mogućnost odabira parametara za prikaz pri tabelarnom i grafičkom pregledanju rezultata proračuna
 - h. Mogućnost brisanja svojih proračuna iz baze podataka
3. Opcione funkcionalnosti
 - a. Opcija izbora između srpskog i engleskog jezika
 - b. Mogućnost izmene svojih ličnih podataka prosleđenih prilikom registracije
 - c. Pokretanje proračuna dok je u toku izvršavanje nekog drugog proračuna. Tada proračun čeka da se svi prethodno pokrenuti proračuni završe.
 - d. Zumiranje po obe ose, po X osi i po Y osi, skiciranje olovčicom, full screen prikaz grafičke reprezentacije rezultata proračuna
 - e. Mogućnost skidanja slike trenutno prikazanog grafika proračuna, skidanja tabele sa numeričkim podacima rezultata proračuna kao i skidanje svih izlaznih podataka odabranog proračuna koje je MUSICO izgenerisao
 - f. Pregledanje kratkog opisa aplikacije kao i snimka koji prikazuje opcije aplikacije i kako dođi do njih.

1.4. Lista materijala

- Specifikacija softverskih zahteva projekta WEMUS
- Opis projekta
- Reference tehnologija korišćene u izradi projekta navedene u dokumentu Specifikacija softverskih zahteva pod poglavljem 1.4. „Reference“

1.5. Definicije, akronimi i skraćenice

IMI – Institut za matematiku i informatiku.

PMF – Prirodno-matematički fakultet u Kragujevcu.

Hardware – opipljive komponente računara.

Software – programi računara koji mu pružaju funkcionalnost.

MUSICO (Muscle Simulation Code) – softver za simulaciju rada mišića na molekularnom nivou.

WEMUS (Web-based Muscle Simulation) – web aplikacija za intuitivniju upotrebu MUSICO-a.

MVC (Model View Controler) – softverski patern koji odvaja prikaz informacija od interakcije korisnika sa tim informacijama.

OO (Objektno orijentisano) – način programiranja zasnovan na objektima.

UML (Unified Modeling Language) – standardni grafički jezik za modelovanje objektno orijentisanog softvera.

CRUD (Create Read Update Delete) – kreiranje, čitanje, izmena i brisanje podataka.

WEB (World Wide Web) – svetska mreža računara uključujući i internet

HTTP (Hypertext Transfer Protocol) – aplikacioni protokol koji čini osnovu veba.

HTTPS (Hypertext Transfer Protocol Secure) – verzija HTTP protokola sa dodatnom zaštitom.

Socket – način povezivanja komponenti radi prebacivanja podataka.

Server i cluster – udaljeni računari koji vrše veliki broj kalkulacija.

RAID (Redundant Array of Independent Disks) – način konfigurisanja grupe hard diskova.

Windows – operativni sistem za računare koji razvija kompanija Microsoft.

Linux – operativni sistem otvorenog koda.

Front-end (klijentski deo aplikacije) – deo aplikacije koji se izvršava na računarima korisnika.

Back-end (serverski deo aplikacije) – deo aplikacije koji se izvršava na udaljenim računarima.

API (Application Program Interface) – interfejs za komunikaciju između aplikacija.

CSV (Comma-Separated Values) – fajl sa podacima koji su razdvojeni zarezom.

HTML (HyperText Markup Language) – standardni jezik za pravljenje web stranica.

CSS (Cascading Style Sheets) – jezik za definisanje izgleda elemenata web stranica.

Javascript – dinamički programski jezik koji se najčešće koristi za definisanje funkcionalnosti web strana na korisnikovom računaru.

Framework (okvir) – konceptualna struktura koja ima za cilj opsluživanje i učestvovanje u razvoju nečak korisnog.

Angular.js – javascript MVC okvir koji omogućava pravljenje dobro struktuiranih front-end aplikacija.

Node.js – serverska javascript platforma pogodna za izradu brzih aplikacija koje rade u realnom vremenu.

NPM (Node Package Manager) – paket menadžer koji koristi Node.js.

StrongLoop (LoopBack) – Node.js okvir.

D3.js (Data Driven Document) – javascript okvir za pravljenje dinamičkih i interaktivnih grafičkih elemenata web stranice.

NoSQL – vrste baza podataka koje ne koriste klasične SQL upite. Podaci se čuvaju u fajlovima a ne u tabelama kao u SQL bazama.

MongoDB – NoSQL baze podataka otvorenog koda.

Local storage – memorija koju pretraživači koriste za čuvanje podataka i nakon što su isključeni.

Sarkomera – strukturna i funkcionalna jedinica mišića koja omogućava kontrakciju.

Aktin – deo sarkomere koji se povlači.

Miozin – deo sarkomere koji vrši povlačenje.

SSZ – specifikacija softverskih zahteva.

SDS – specifikacija dizajna softvera.

Korisnik – svako lice koje koristi sistem.

Administrator – korisnik čiji je zadatak administracija ostalih korisnika i resursa.

1.6. Pregled ostatka dokumenta

Poglavlje 1 – „**Uvod**“, trenutna tačka koja daje opšti opis proizvoda, pregled sistema, opis dokumenta i skraćenice i definicije koje se mogu naći u dokumentu.

Poglavlje 2 – „**Razmatranje dizajna**“, opisuje pretpostavke o tome u kakvom okruženju je projekat namenjen da funkcioniše, softverska i hardverska ograničenja koje projekat mora ispuniti kao i tehnologije koje su korišćene u izradi projekta.

Poglavlje 3 – „**Arhitektura sistema**“, daje uvid o delovima projekta kao i obrazloženja tih delova u celokupnoj arhitekturi sistema.

Poglavlje 4 – „**Dizajn visokog nivoa**“, u kome su dati logički i fizički pogled projekta. U logičkom pogledu se opisuje povezanost komponenti dok se u fizičkom pogledu objašnjavaju funkcije komponenti.

Poglavlje 5 – „**Dizajn baze podataka**“, daje uvid u tabele koje baza podataka treba da ima, opis podataka koje će svaka tabela da čuva kao i tipove podataka koje svaka tabela čuva.

Poglavlje 6 – „**Dizajn niskog nivoa**“, koji daje detaljan opis za svaku od komponenti celog sistema uz prateće dijagrame i objašnjenja.

Poglavlje 7 – „**Dizajn korisničkog interfejsa**“, služi da prikaže kako gotov sistem zapravo treba da izgleda uz slike i prateća pojašnjenja.

2. Razmatranje dizajna

2.1. Pretpostavke

Kako je WEMUS web aplikacija, namenjeno je da joj se može pristupiti bilo kada, 24 časa dnevno, 7 dana u nedelji. Aplikacija treba da omogući istovremeni pristup više korisnika. Svaki od korisnika može da pokrene proračun koji staje u red izvršavanja i mora da sačeka sve prethodne proračune da završe sa izvršavanjem kako bi on stupio na red. Korisnik može da koristi sve ostale funkcionalnosti aplikacije dok čeka na izvršenje proračuna.

Odstupanje od pristupa aplikaciji je kvar web servera, nestanak struje na serveru ili kod korisnika, neaktivan priključak na internet kod servera ili korisnika. U slučaju da se dogodi takav događaj ili neka druga nepredviđena okolnost zbog koje server neće raditi, aplikaciji neće moći da pristupe korisnici.

Ukoliko se desi da klaster na kome se izvršavaju simulacije ne funkcioniše zbog gore navedenih okolnosti ili drugih nepredviđenih situacija, a web server koji služi za opsluživanje korisnikovih zahteva funkcioniše, korisnici će moći da koriste sve funkcionalnosti sajta ali se pokrenuti proračuni neće izvršavati sve dok klaster nije u funkciji.

Kako bi koristili aplikaciju korisnik mora imati registrovani nalog sa unetim podacima i validnom email adresom. Nakon registracije korisnik mora potvrditi identitet tako što će pratiti uputstvo poslato na unetu email adresu. U suprotnom neće imati pristup mogućnostima aplikacije. Odstupanje se može desiti ukoliko je email adresa blokirana usled preopterećenja pri čemu će korisnik biti obavešten, a nakon blokade korisnik će moći da se registruje.

Kada korisnik želi da promeni svoje lične podatke koje je predao prilikom registracije pored pristupa aplikaciji potrebno je da zna svoju lozinku. Ukoliko želi da promeni email adresu, pored šifre mora imati i pristup email adresi koju želi uneti kako bi mogao da prati uputstva koja su mu tamo poslata. U slučaju da zna lozinku sistema i promeni email na adresu kojoj nema pristup korisnik neće moći da verifikuje unetu email adresu samim tim ni da koristi sistem sa tim nalogom. Ukoliko je sistemska email adresa blokirana usled preopterećenja korisnik će biti obavešten, a nakon blokade korisnik će moći da nastavi sa korišćenjem sistema.

U slučaju da računar sa kojim korisnik pokušava da pristupi aplikaciji nema pristup internetu, stabilno napajanje strujom, ispravan operativni sistem sa web pretraživačem, server i klaster će raditi kao što je predviđeno ali korisnikov računar neće moći da komunicira sa serverom samim tim korisnik neće imati pristup aplikaciji. Odstupanje je ukoliko korisnik ima pokrenut server i klaster u lokalnoj mreži, tada nije neophodan pristup internetu.

Pretpostavlja se da korisnik ima dovoljan nivo znanja u korišćenju računara kao i u oblasti simulacije rada mišića softverom MUSICO kako bi pravilnije i bolje koristio WEMUS.

U slučaju promene ulaznih fajlova MUSICO softvera potrebno je izmeniti odgovarajuće forme WEMUS-a, prateće promene u bazi i u funkcijama za kreiranje fajlova na klasteru.

2.2. Ograničenja

Projektna ograničenja:

- Za front-end deo aplikacije treba koristiti Angular 2.0 javascript okvir,
- Za sve grafičke prikaze u aplikaciji neophodno je koristiti D3.JS javaskript okvir,
- Bazu podataka potrebno je razviti korišćenjem MongoDB pristupa,
- Klaster mora biti odvojen od servera za opsluživanje klijentskih računara i razvijen u proizvoljnom okviru Node.js javaskript tehnologije,
- Server za opsluživanje klijentskih računara mora biti razvijen korišćenjem proizvoljnog okvira Node.js javaskript tehnologije.

Minimalna hardverska konfiguracija za funkcionisanje serverske strane sistema je:

- Procesor sa radnim taktom od 2 GHz sa dva jezgra,
- Radna memorija od 4 GB,
- Memorija za trajno skladištenje podataka tipa hard disk ili solid state drive, a slobodnog kapaciteta 20 GB (zavisi od količine skladištenih podataka).

Softverska ograničenja sistema na kojoj će biti pokrenut server su sledeća:

- Linux operativni sistem distribucije Ubuntu 14.4,
- Instaliran i pokrenut MongoDB servis za baze podataka,
- Instaliran i pokrenut Node web servis i njegov npm paket menadžer,
- Instaliran process manager pm2 za pokretanje WEMUS servisa.

Klijentski deo aplikacije treba razvidi tako da se može izvršavati na sistemima koji zadovoljavaju sledeće uslove:

- Stabilno napajanje strujom i internet konekciju ili pristup lokalnoj mreži u slučaju da je sistem pokrenut lokalno,
- Instaliran i funkcionalan operativni sistem Windows XP/Vista/7/8/8.1/10 i na stabilnim distribucijama Linux-a,
- Instaliran Google Chrome pretraživač za najugodnije korišćenje. Aplikacija može raditi i na ostalim pretraživačima ali zbog razlika u podržanim tehnologijama neki delovi aplikacije neće funkcionisati kao što je namenjeno.

Hardverski i softverski zahtevi koji su navedeni iznad nisu jedini na kojima sistem može funkcionisati. Preporučeno je da se usklade hardverske komponente servera sa opterećenjem. Konfiguracije sa slabijim performansama nisu preporučljive iako bi možda sistem funkcionisao na takvim. Naime što je bolja konfiguracija više korisnika će moći da koriste istovremeno sistem. Korisnici će takođe imati ugodnije iskustvo u korišćenju aplikacije i manje su šanse za nepredviđenim događajima. Bolja konfiguracija klastera za izračunavanje će omogućiti bolje performanse u izvršavanju samih proračuna pa će korisnici manje čekati da njihove simulacije dođu na red za izvršavanje. Hardverske komponente računara sa koga korisnik pristupa sistemu nisu od ključne važnosti ali treba napomenuti da će moderniji računari sa normalnim ili većim performansama pružiti bolje iskustvo u korišćenju sistema.

2.3. Sistemsko okruženje

Server na kome će biti pokrenut sistem WEMUS se nalazi na Prirodno-matematičkom fakultetu Univerziteta u Kragujevcu. Server se nalazi u laboratorijskim uslovima i neophodno je da ispuni sve zahteve navedene u pretpostavkama i ograničenjima kako bi funkcionisao kao što je namenjeno.

Administriranje aplikacije kao i vođenje računa o funkcionalnosti celokupnog sistema zaduženi su eksperti informatike sa Instituta za matematiku i informatiku fakulteta na kom se server i klaster nalaze.

Kako bi sigurnost podataka bila pojačana preporučeno je da sistem bude instaliran na diskovima konfigurisanim u nekoj od RAID konfiguracija sa redundantnim čuvanjem podataka na sistemu. Ovim se sprečava da kvar jednog diska prouzrokuje pad sistema. Takođe je poželjno raditi dnevne udaljene bekap-ove podataka u slučaju kvara celog sistema ili nekih drugih nepogoda zbog kojih podaci mogu biti oštećeni. U slučaju pada sistema za povratak u funkcionalno stanje su zaduženi stručnjaci sa Instituta za matematiku i informatiku, Prirodno-matematičkog fakulteta Univerziteta u Kragujevcu i to u što kraćem roku u zavisnosti od potrebe za korišćenjem aplikacije.

Stručnim licima instituta je neophodno obezbediti pristup sistemu i njegovim resursima kako bi vršili provere, eventualne izmene i povratak u funkcionalno stanje pri padu sistema.

Korisnici aplikacije ne moraju biti eksperti u oblasti informatike ali se podrazumeva osnovni nivo znanja u korišćenju računara. Korisnici aplikacije moraju biti dovoljno stručna lica u oblasti simulacije mišića kako bi bili upoznati sa svim parametrima zahtevanim za pokretanje simulacije. Ovo je bitno kako bi podaci koje korisnik unosi bili validni i da ne bi došlo do situacije da se server preoptereći ili pokrene neispravna simulacija pri čemu se može blokirati red proračuna i ostali korisnici mogu ostati bez mogućnosti izvršavanja svojih simulacija.

Klijentski deo aplikacije je predviđen da se pokreće na desktop i prenosivim računarima. Iako se možda može koristiti sa pametnim telefonom, tabletom i sličnim hibridnim uređajima, aplikacija subjektivno neće pružiti isti doživljaj i lakoću korišćenja kao što je to na uređajima predviđenim za njeno korišćenje. Korisnik mora obezbediti da računar na kojem želi da koristi aplikaciju bude snabdeven konstantnim stabilnim napajanjem strujom i stabilnu konekciju na internet, a u slučaju da je server u lokalnoj mreži treba biti konektovan na tu mrežu. Računari preko kojih se pokreće aplikacija se mogu nalaziti bilo u kućnim ili laboratorijskim uslovima.

3. Arhitektura sistema

3.1. Pregled sistema

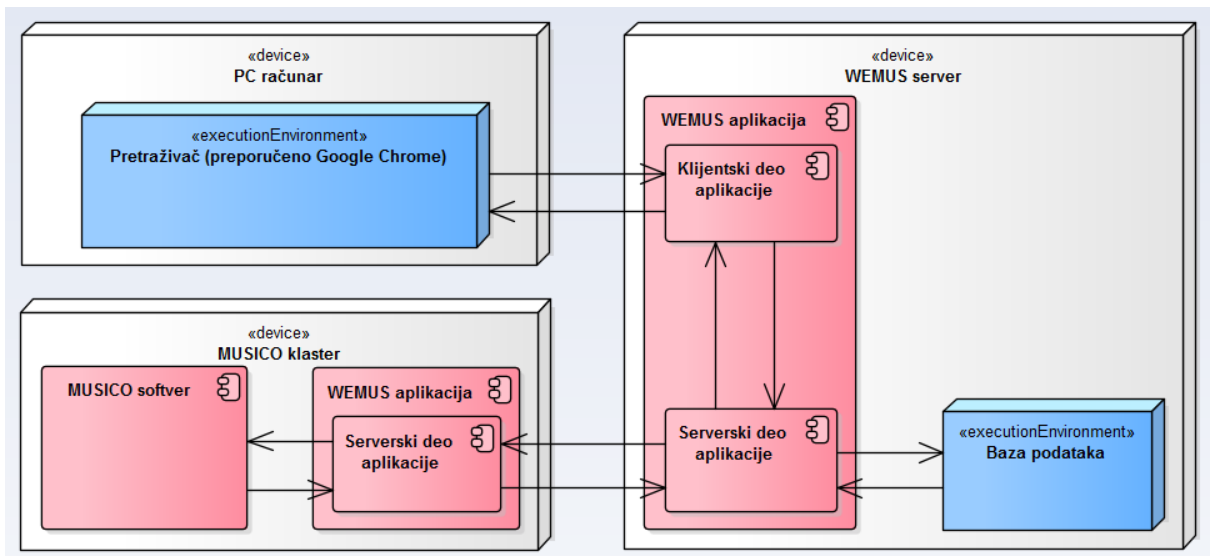
Sistem se sastoji od dve odvojene komponente:

- Web aplikacija – sastoji se sastoji od klijentskog i serverskog dela. Oni su predstavljeni pomoću tri sloja arhitekture. Aplikacija služi za zadavanje ulaznih parametara, pokretanje simulacije, obradu podataka dobijenih od MUSICO softvera i vizuelizaciju izlaznih podataka simulacije.
- MUSICO server – komponenta sistema koja se nalazi na zasebnom klasteru. Zadužena je za pokretanje MUSICO softvera, obradu informacija o trenutnom stanju i napretku proračuna i konačno obradu izlaznih podataka MUSICO softvera.

Komunikacija između, MUSICO servera i web aplikacije obavljaće se putem soketa, a posredstvom JSON objekata, kako bi web aplikacija uvek i pravovremeno bila obavještena o stanju MUSICO proračuna.

Web aplikacije će imati troslojnu arhitekturu pri čemu će slojevi činiti klijentski i serverski deo web aplikacije:

- Prezantacioni sloj
- Sloj poslovne logike
- Sloj zadužen za pristup podacima



Slika 1 - Komponente sistema

3.2. Obrazloženje dizajna

Usled dužine trajanja i kompleksnosti proračuna koje izvršava MUSICO softver potrebno je da taj softver, kao zasebna celina, bude odvojena od ostatka aplikacije, na sopstvenom klasteru. Na taj način, u slučaju eventualne nemogućnosti MUSICO programa za pokretanje proračuna korisnici će i dalje moći da vrše pregled svojih ranijih proračuna upravo zbog razdvojenosti ove dve komponente sistema, a samim tim ni klaster na kome se nalazi MUSICO softver neće biti opterećen opsluživanjem klijenata web aplikacije.

Razlog za korišćenje troslojne arhitekture leži u tome što su moduli jasno razdvojeni. Korisnički interfejs je odvojen od podataka i logike i obratno. Ovaj način arhitekture omogućava paralelno razvijanje modula, nezavisno jedan od drugog i njihovu laku izmenu.

Prezentacioni sloj predstavlja korisnički interfejs aplikacije. U njemu se nalaze komponente koje od korisnika prihvataju podatke i ulazne parametre, a sa druge strane i vrše prikaz podataka konačnim korisnicima. Prezentacioni sloj će biti implementiran uz pomoć html i javascript datoteka koje će biti kreirane u Angular 2.0 framework-u.

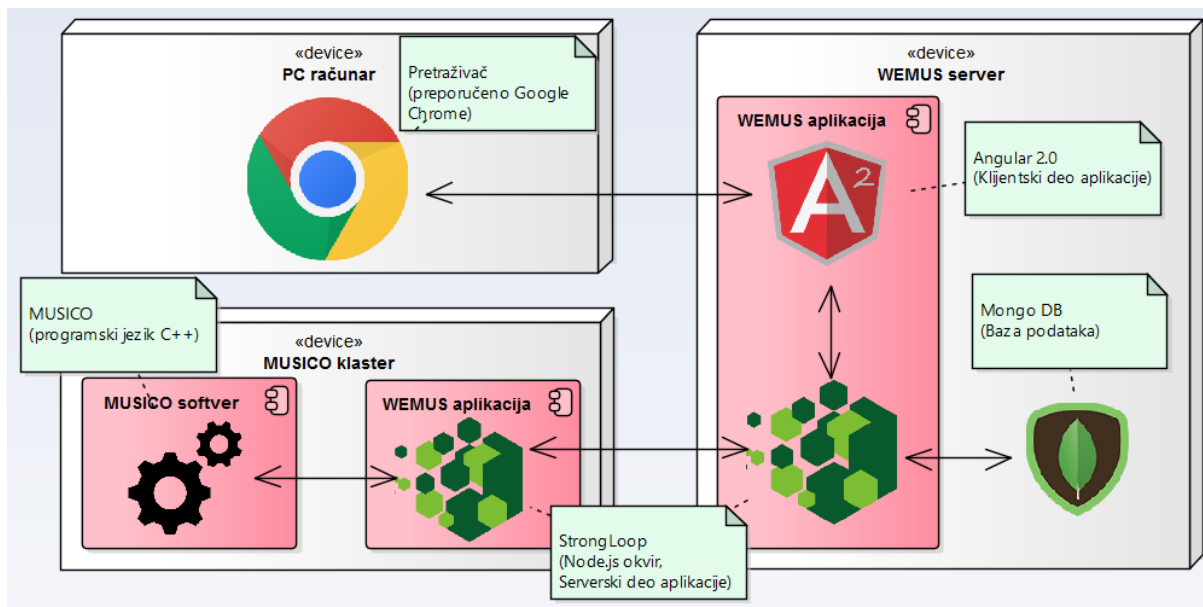
Sloj poslovne logike predstavlja posrednika između sloja prezentacije i sloja koji je zadužen za pristup i obradu podataka. Zadatak sloja poslovne logike jeste i komunikacija sa klasterom na kome će se nalaziti MUSICO.

Sloj pristupa podacima zadužen je za obradu podataka koje aplikacija obuhvata. Sadržaće module koji su zaduženi za izvršavanje upita nad MongoDB bazom podataka. Ovaj sloj će biti zadužen za obradu podataka koji se tiču MUSICO ulaznih parametara i rezultata MUSICO softvera, ali i za obradu podataka koji se tiču samih korisnika, poput njihovih ličnih podataka.

4. Dizajn visokog nivoa

4.1. Konceptualni pogled

Na priloženoj slici su predstavljene komponente WEMUS aplikacije i tehnologije u kojima se trebaju napraviti.



Slika 2 - Konceptualni dijagram sistema

1. Baza podataka koja će se koristiti biće NoSQL MongoDB baza.
2. Server koji opslužuje klijentski deo aplikacije kao i server za izvršavanje proračuna biće NodeJS serveri, pisani u StrongLoop framework-u za NodeJS.
3. Upiti ka bazi vršiće se korišćenjem predefinisanih StrongLoop metoda za rad sa podacima.
4. Server koji opslužuje klijentski deo aplikacije i komunicira sa bazom sastojaće se od REST API modela.
5. Klijent će sa REST API serverom komunicirati putem HTTP zahteva uz korišćenje soketa u određenim slučajevima.
6. Server za izvršavanje proračuna primaće zahtev za pokretanje proračuna i odgovarajuće fajlove nakon kojih će pokretati MUSICO softver.
7. Za komunikaciju između dva servera koristiće se soketi.
8. Za grafički prikaz rezultata biće korišćena D3 javaskript biblioteka. Za prikaz u punom ekranu će biti korišćen HTML5 Full Screen API.

5. Dizajn baze podataka

Zbog veliko količine podataka koja ce se čuvati u bazi, koristi se NoSQL MongoDB baza podataka.

Sama baza podataka ima 6 kolekcija:

1. Tabela **WemusUser** - Čuva podatke o registrovanim korisnicima.

WemusUser		
Naziv atributa	Tip podatka	Opis
id	String	Jedinstveni identifikator korisnika
username	String	Korisničko ime
firstName	String	Ime
lastName	String	Prezime
password	String	Enkriptovana sifra korisnika
Email	String	Email korisnika
emailVerified	Boolean	Označava da li je korisnik verifikovao svoj nalog
Type	String	Označava kog je tipa korisnik (admin/user)

2. Tabela **Experiment** - Čuva podatke o proračunu.

Experiment		
Naziv atributa	Tip podatka	Opis
userID	String	Id korisnika čiji je proračun
userName	String	Korisničko ime korisnika čiji je proračun
Title	String	Naziv proračuna
Description	String	Opis proračuna
Date	String	Vreme pokretanja proračuna
softwareVersion	String	Verzija musico programa na kom se poreće proračun
State	Number	Označava stanje proračuna (default 0 – pokrenut) 1 – završeno izračunavanje, nije pročitano 2 završeno izračunavanje, pročitano 3 – sačuvan za kasnije pokretanje
Parameters	JSON Object	Svi ulazni fajlovi za pokretanje proračuna

3. Tabela **resultChunks** - Čuva rezultat pokrenutog proračuna. Zbog velike količine podataka, oni se moraju deliti u komade.

resultChunks		
Naziv atributa	Tip podatka	Opis
experimentID	String	Ozračava kom proračunu pripada komad
chunkNumber	Number	Redni broj komada
Data	JSON Object	Sami podaci proračuna

4. Tabela **Role** – Čuva sve moguće uloge koje registvorani korsnik može da ima, u našem slučaju to su običan korisnik i administrator.

Role		
Naziv atributa	Tip podatka	Opis
id	String	Jedinsteni identifikator role
Name		Ime role koju kreiramo Na primer admin
Crated	Date	Kada je kreirana rola
Modified	Date	Kada je rola menjana

5. Tabela **RoleMapping** – Koristi se da rasporedi uloge na korisnike, slično šifarniku.

RoleMapping		
Naziv atributa	Tip podatka	Opis
ID	String	Jedinstveni identifikator
principalType	String	Tip uloge u sistemu
principalId	String	ID korisnika na kog se primenjuje rola
RoleId	String	ID role koja se primenjuje na korisnika

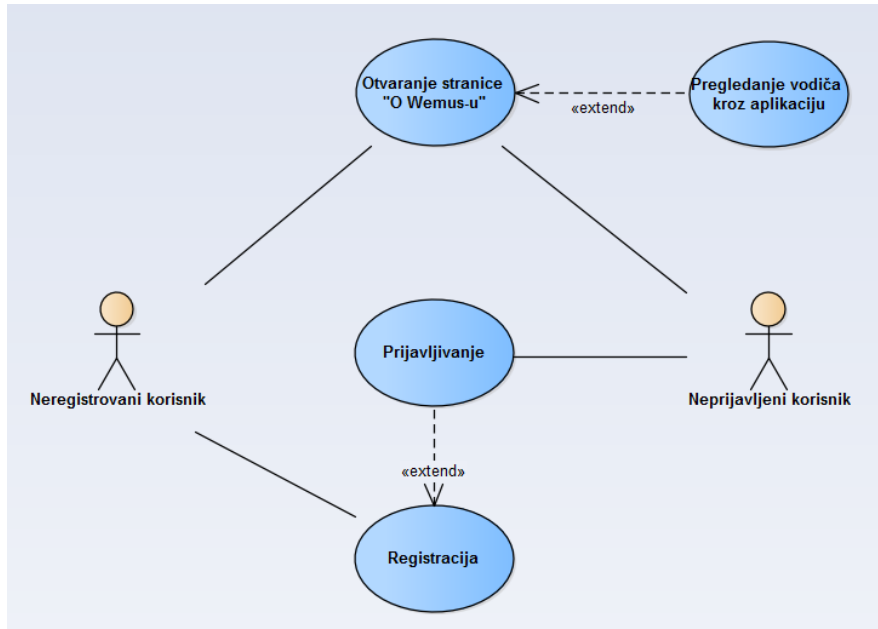
6. Tabela **AccessToken** – Čuva tokene koje korisnik dobija pri prijavljivanju na server, kako bi mu se omogućio pristup funkcionalnostima sistema.

AccessToken		
Naziv atributa	Tip podatka	Opis
ID	String	Jedinstveni identifikator
ttl	Number	Vreme trajanja access tokena, u sekundama
created	Date	Datum kada je access token kreiran
userId	String	Jedinstveni identifikator korisnika čiji je access token

6. Dizajn niskog nivoa

6.1. Dizajn funkcionalnosti aplikacije

U ovom odeljku priloženi su načini korišćenja WEMUS aplikacije, moguće greške i izuzeci u toku njenog korišćenja kao i potrebne preventive.



Slika 3 - Mogući slučajevi korišćenja pre prijavljivanja na sistem

6.1.1. Otvaranje stranice „O Wemus-u“

Kratak opis: Otvaranje i pregledanje stranice „O Wemus-u“.

Opis slučaja korišćenja:

Nakon pokretanja aplikacije korisnik će imati mogućnost pristupa ovoj stranici. Stranica „O Wemus-u“ omogućavaće korisniku kratak uvod u to šta je WEMUS aplikacija i čemu ona služi, kao i o samoj sarkomeri koja je predmet proračuna i izučavanja. Korisnik će ovoj stranici moći da pristupi klikom na polje „Nešto o wemus-u“ koje će se nalaziti u korisničkom meniju. Korisnik će klikom na linkove više ili manje moći da bira količinu teksta koji želi da pročita. Korisnik će moći da pogleda i vodič kroz aplikaciju, koji će mu pokazati moguće opcije korišćenja u slučaju da napravi svoj korisnički nalog.

Glavni tok događaja:

Korisnik pokušava da pristupi WEMUS aplikaciji. U procesu pokretanja aplikacije korisnika o toku istog obaveštava animirana slika za učitavanje. Ukoliko je aplikacija uspešno pokrenuta, korisnik vrši navigaciju na stranicu „O Wemus-u“ korišćenjem korisničkog menija. Korisnik preko ugrađenog YouTube player-a pregleda vodič kroz aplikaciju.

Slučaj korišćenja završava se gašenjem aplikacije ili redirekcijom na drugu stranicu.

Definisanje uslova:

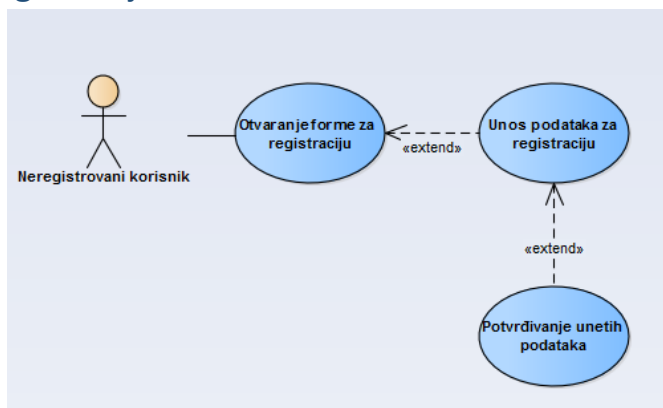
Pred-uslov: Uspešno pokretanje WEMUS aplikacije.

Alternativni tokovi događaja:

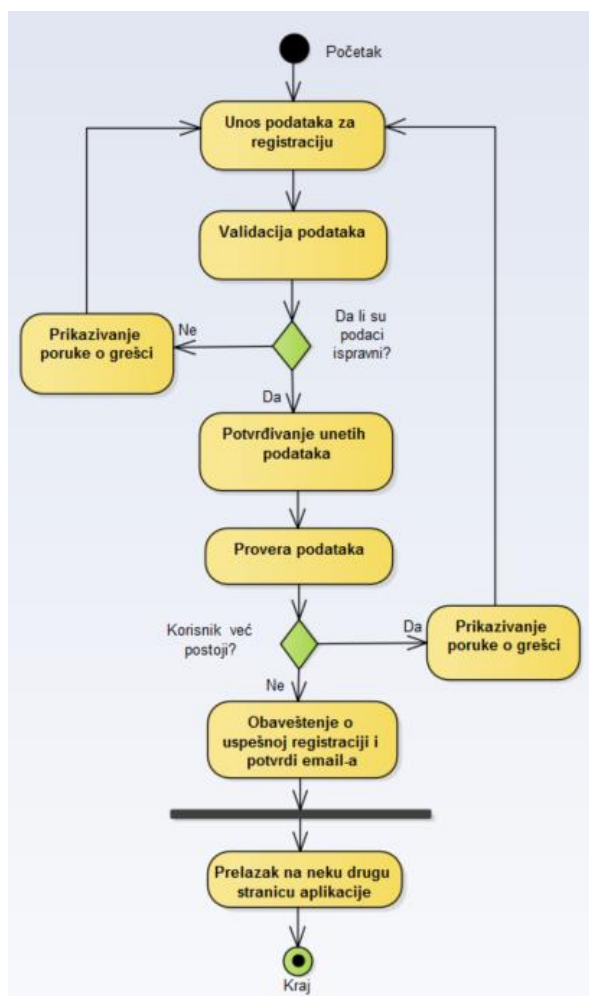
U slučaju da korisnik nije povezan na internet, neće biti u mogućnosti da pristupi aplikaciji. U slučaju da je server za opsluživanje korisničke aplikacije onesposobljen korisnik neće moći da pristupi aplikaciji.

Zahtevi: /

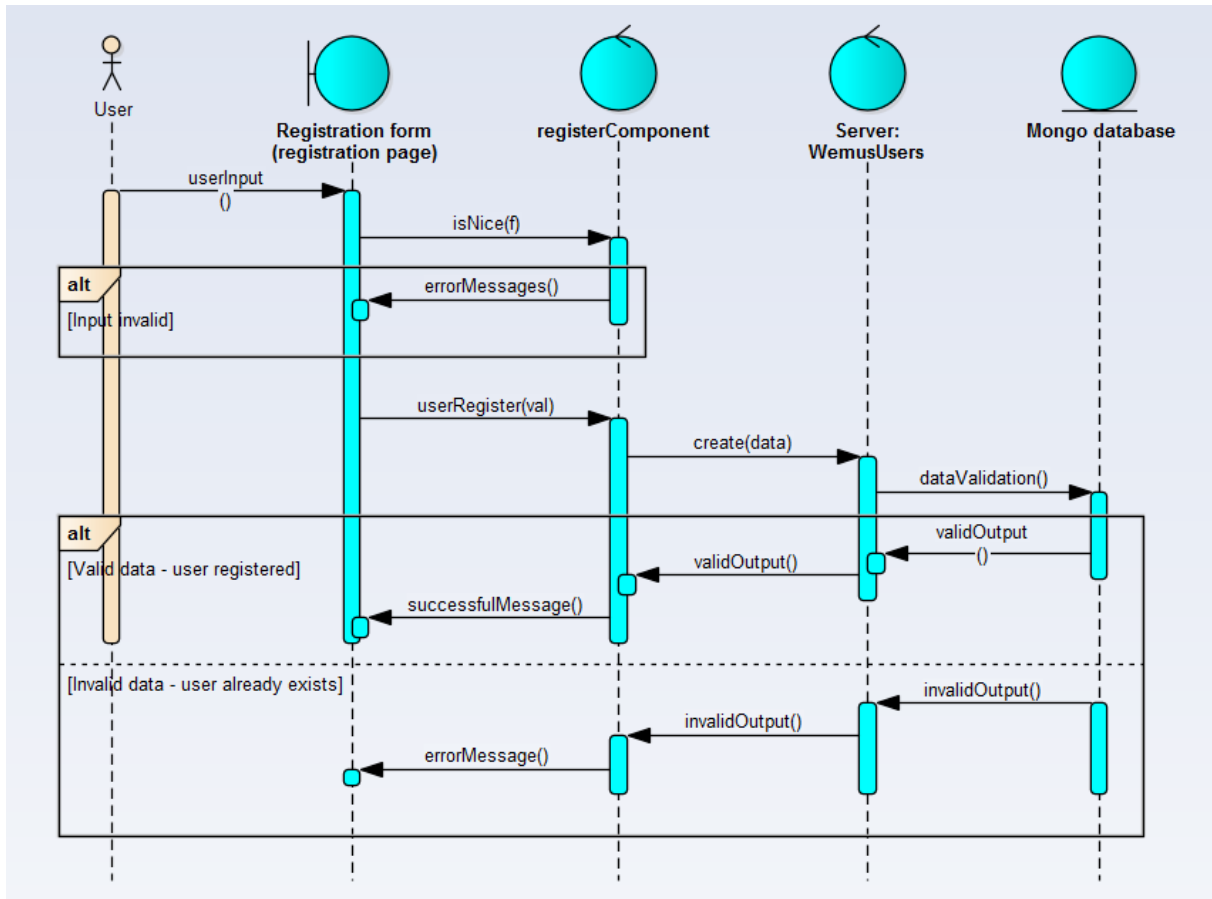
6.1.2. Registracija korisnika



Slika 4 - Slučaj korišćenja: registracija



Slika 5 - Dijagram aktivnosti registracije korisnika



Slika 6 - Dijagram sekvence za registraciju korisnika

Kratki opis: Korisnik će moći da kreira korisnički nalog.

Opis slučaja korišćenja:

Korisnik će imati mogućnost pravljenja korisničkog naloga. Nakon klika na polje „Registruj se“ u glavnom meniju otvoriće se forma za unos podataka. Korisnik mora da unese ime, prezime, elektronsku poštu i lozinku u ispravnom formatu, u suprotnom će porukom ispod polja za unos biti obavešten o grešci. Ukoliko je sve podatke uneo ispravno i registracija prođe uspešno, korisnik će biti obavešten porukom. Ukoliko korisnički nalog već postoji korisnik će biti obavešten porukom sa mogućnošću ponovnog registrovanja.

Glavni tok događaja:

Korisnik pristupa formi za registraciju. Korisnik unosi podatke. Pri unosu podataka u svako od polja korisnik biva obavešten o validnosti podataka koje je uneo. Nakon što je sve podatke uneo ispravno korisnik pokušava da se registruje klikom na dugme „Registruj se“. Ukoliko korisnik sa istim korisničkim imenom ili email adresom već postoji, korisnik će biti obavešten odgovarajućom porukom sa mogućnošću ponovnog registrovanja. Ukoliko je registracija završena uspešno korisnik će biti obavešten porukom o proveri email adrese radi završetka registracije.

Slučaj korišćenja se završava potvrdom email adrese, redirekcijom na drugu stranicu ili gašenjem aplikacije.

Definisanje uslova:

Pred-uslov: Pokretanje WEMUS aplikacije.

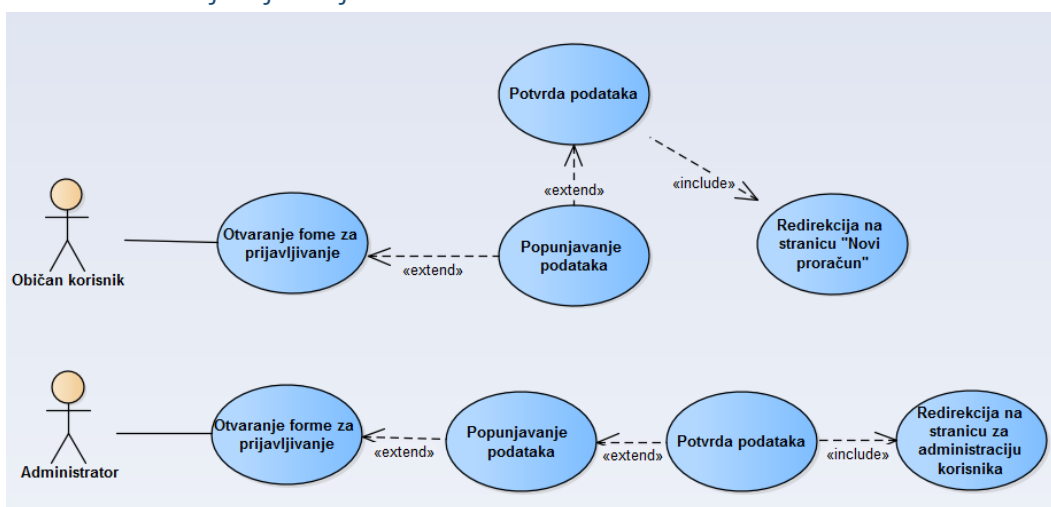
Post-uslov: Regularnost baze podataka.

Alternativni tokovi događaja:

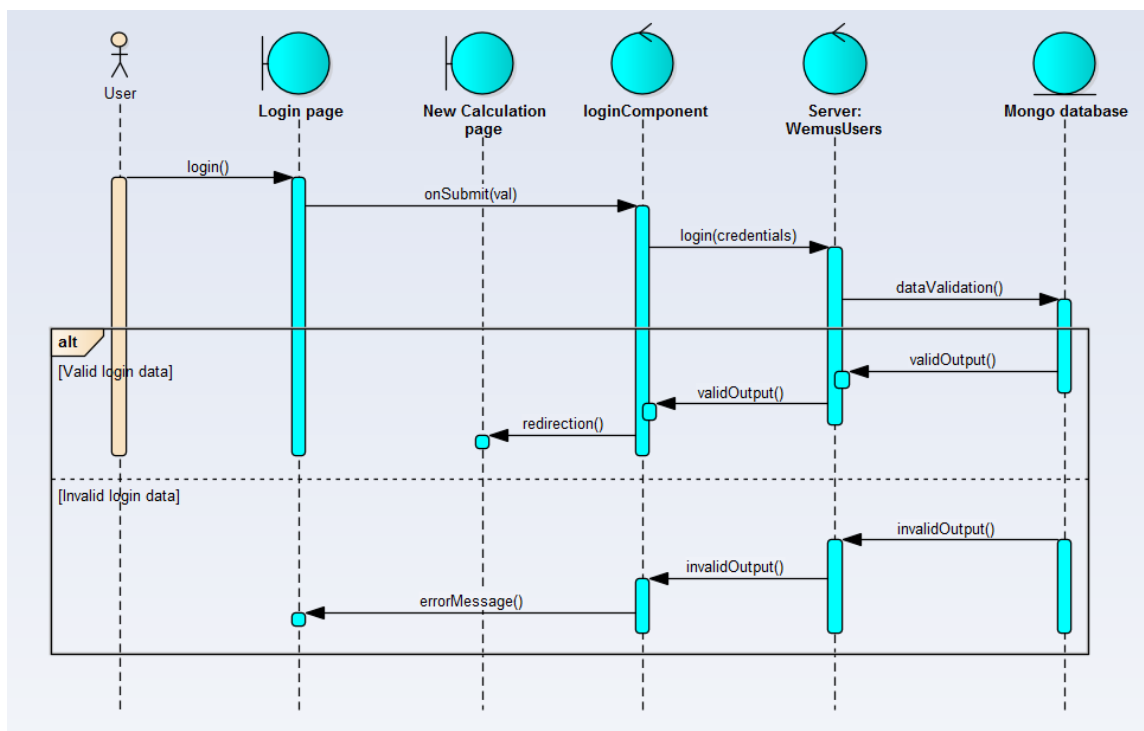
Ukoliko korisnik unese korisničko ime ili email adresu koji već postoje u sistemu, biće obavešten i dobija mogućnost ponovne registracije. Ukoliko nešto nije u redu prilikom slanja email-a za potvrdu registracije, korisnik će biti obavešten sa mogućnošću ponovne registracije.

Zahtevi: Korisnik pri administraciji ne dobija administratorska prava.

6.1.3. Prijavljivanje korisnika



Slika 7 - Slučaj korišćenja: Prijavljivanje



Slika 8 - Dijagram sekvence za prijavljivanje korisnika

Kratak opis: Korisnik će moći da se prijavi na sistem

Opis slučaja korišćenja:

Nakon klika na polje „Uloguj se“ u glavnom meniju ili pokretanja aplikacije, ukoliko nije prijavljen, otvoriće se forma za unos podataka. Ukoliko nije prijavljen a ima svoj korisnički nalog korisnik može da se prijavi koristeći svoje korisničko ime i lozinku. Korisnik će biti obavešten ukoliko pri prijavljivanju dođe do greške usled loše prosleđenih podataka. U suprotnom će biti prebačen na stranicu „Novi proračun“ ili stranicu za administraciju.

Glavni tok događaja:

Korisnik pristupa formi za prijavljivanje pokretanjem aplikacije ili klikom na polje „Uloguj se“ u glavnom meniju. Korisnik unosi podatke i pokušava da se prijavi klikom na dugme „Uloguj je“. Pri unosu pogrešnih podataka, korisnik će biti obavešten sa mogućnošću ponovnog prijavljivanja. Ukoliko je podatke uneo ispravno, korisnik će biti prebačen na stranicu „Novi proračun“ u slučaju da nema privilegije administratora, u suprotnom će biti prebačen na stranicu za administraciju korisnika.

Definisanje uslova:

Pred-uslov: Pokretanje WEMUS aplikacije.

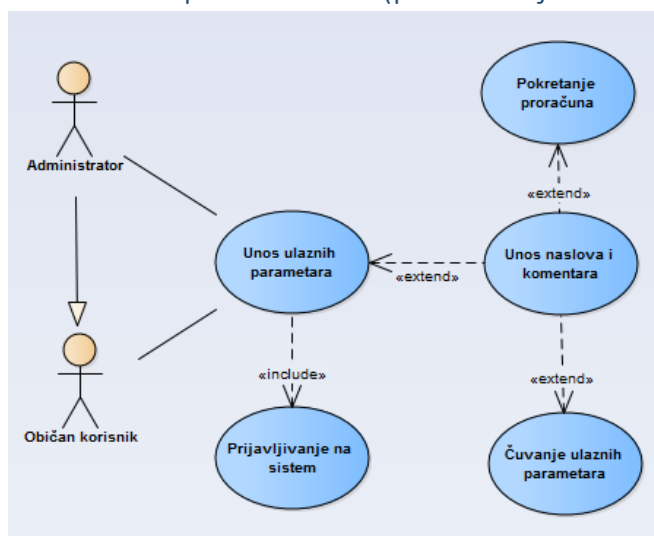
Post-uslov: Regularnost baze podataka.

Alternativni tokovi događaja:

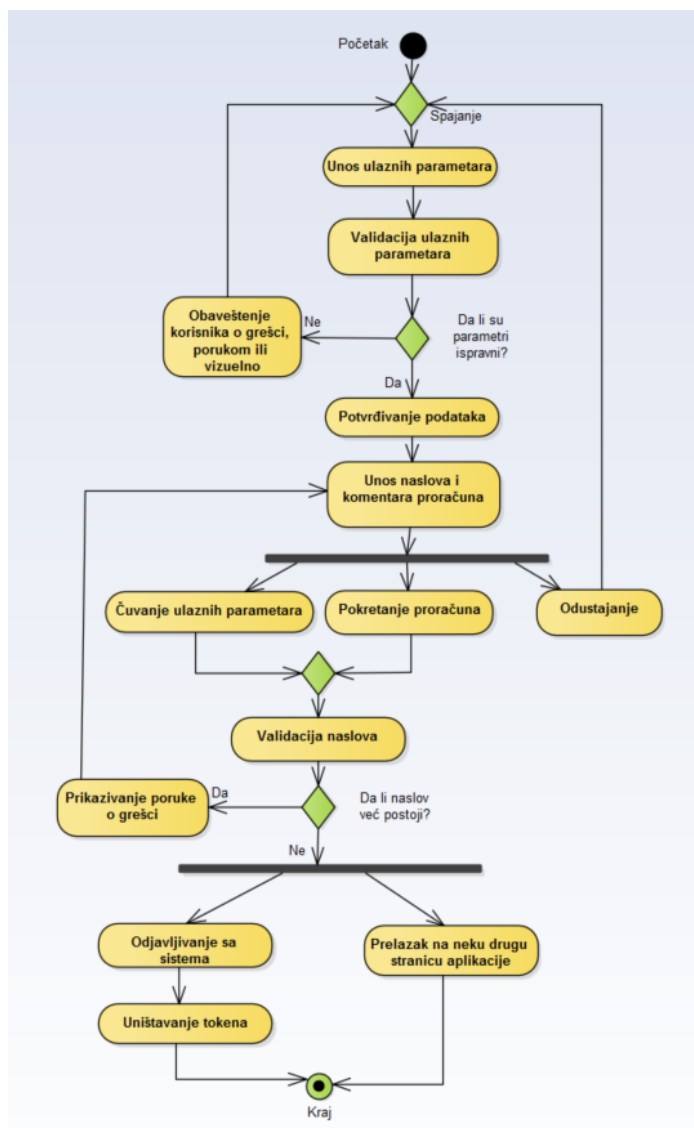
Ukoliko korisničko ime i lozinka nisu ispravni, korisnik dobija obaveštenje sa mogućnošću ponovnog prijavljivanja.

Zahtevi: /

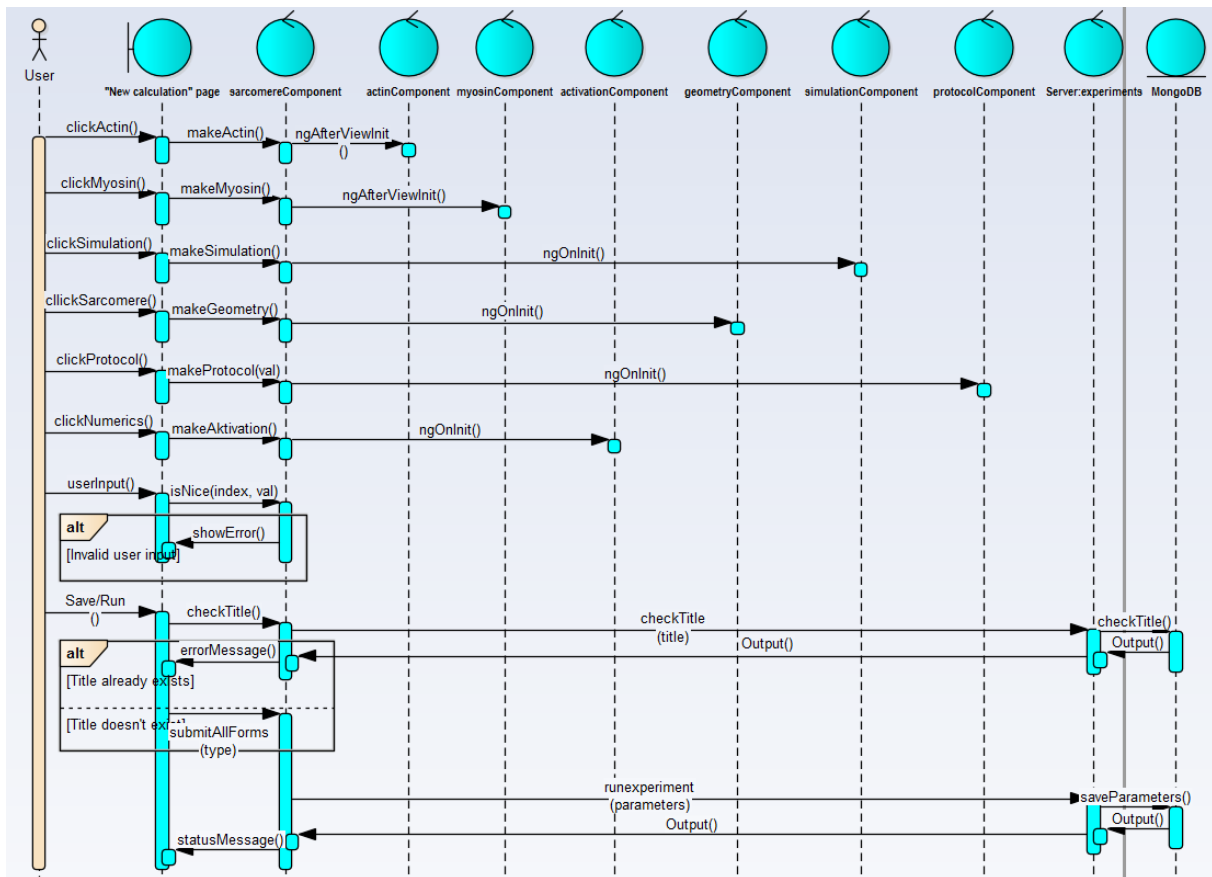
6.1.4. Unos ulaznih parametara (pokretanje i čuvanje)



Slika 9 - Slučaj korišćenja: Unos ulaznih parametara



Slika 10 - Dijagram aktivnosti unošenja parametara za simulaciju



Slika 11 - Dijagram sekvence unosa parametra

Kratak opis: Unos ulaznih parametara proračuna, uključujući pokretanje i čuvanje proračuna.

Opis slučaja korišćenja:

Otvaranjem stranice „Novi proračun“ korisnik će moći da pristupi formama za unos parametara potrebnih za pokretanje proračuna. Moći će da iskoristi predefinisane vrednosti upisane u polja ili da unese svoje. Izmenom nekih od polja za unos korisnik će moći da vidi izmenu i na grafičkom prikazu sarkomere. Ukoliko unese pogrešne podatke biće obavešten o grešci. Ukoliko je ispravno uneo podatke klikom na dugme „Pokreni proračun“ korisniku će biti ponuđeno da unese naslov proračuna i komentar. Korisnik će nakon ispravno unetih podataka moći da pokrene ili sačuva proračun.

Glavni tok događaja:

Korisnik otvaranjem stranice „Novi proračun“ pristupa formama za unos parametara potrebnih za pokretanje proračuna. Korisnik će moći da menja podatke u prikazanim poljama, ili da vrši otpremanje i preuzimanje fajlova sa podacima, a u toku unosa novih vrednosti vršice se proverava unetih podataka, te će korisnik biti obavešten o grešci ukoliko su podaci pogrešni. Izmenom nekih od polja za unos korisnik će moći da vidi izmenu i na grafičkom prikazu sarkomere. Ukoliko su svi podaci ispravno uneti korisniku će biti omogućeno da potvrdi podatke klikom na dugme „Pokreni proračun“ pri čemu se otvara prozor za unos naslova proračuna i komentara. Ukoliko korisnik klikne na dugme „Odbaci“ moći će da nastavi sa izmenom podataka.

Unos naslova proračuna će biti obavezan i korisnik će o tome biti obavešten, a klikom na dugme „Pokreni“ ili „Sačuvaj“ proveravaće se njegova jedinstvenost, pa ukoliko naslov nije jedinstven korisnik će biti obavešten sa mogućnošću ponovnog unosa. Klikom na dugme „Pokreni“, korisnik će moći da pokrene proračun, pri čemu će biti obavešten da je proračun u toku. Klikom na dugme „Sačuvaj“ korisnik će moći da sačuva ulazne parametre. Slučaj korišćenja se završava: Klikom na dugme „Pokreni“ ili „Sačuvaj,“ redirekcijom na drugu stranicu pri čemu korisnik mora da potvrdi da li zaista želi da napusti stranicu, odjavljivanjem, gašenjem aplikacije.

Definisanje uslova:

Pred-uslov: Prijavljivanje na sistem i pristup stranici „Novi proračun“.

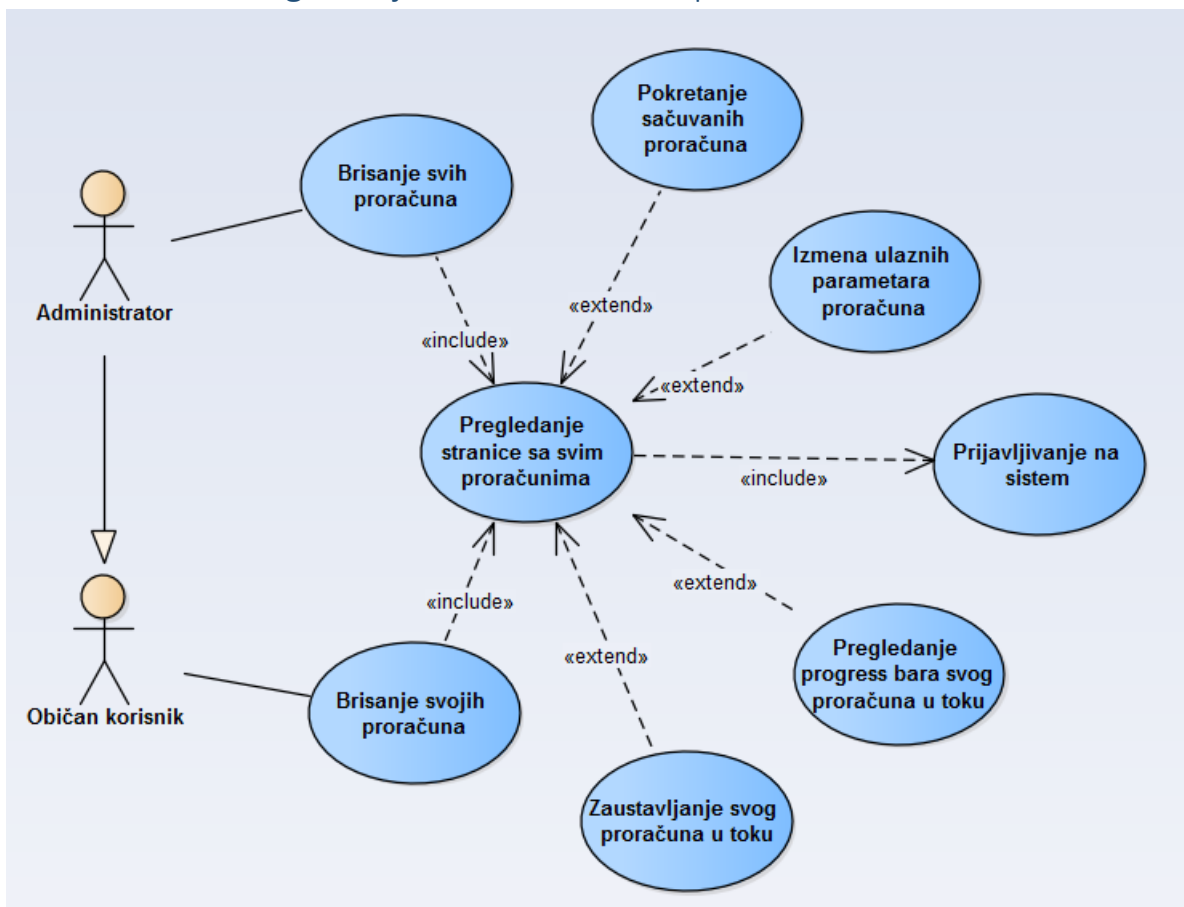
Alternativni tokovi događaja:

Napuštanje stranice pri čemu se gube svi podaci, pa će korisniku biti dato obaveštenje o tome sa izborom da li zaista želi da napusti stranicu.

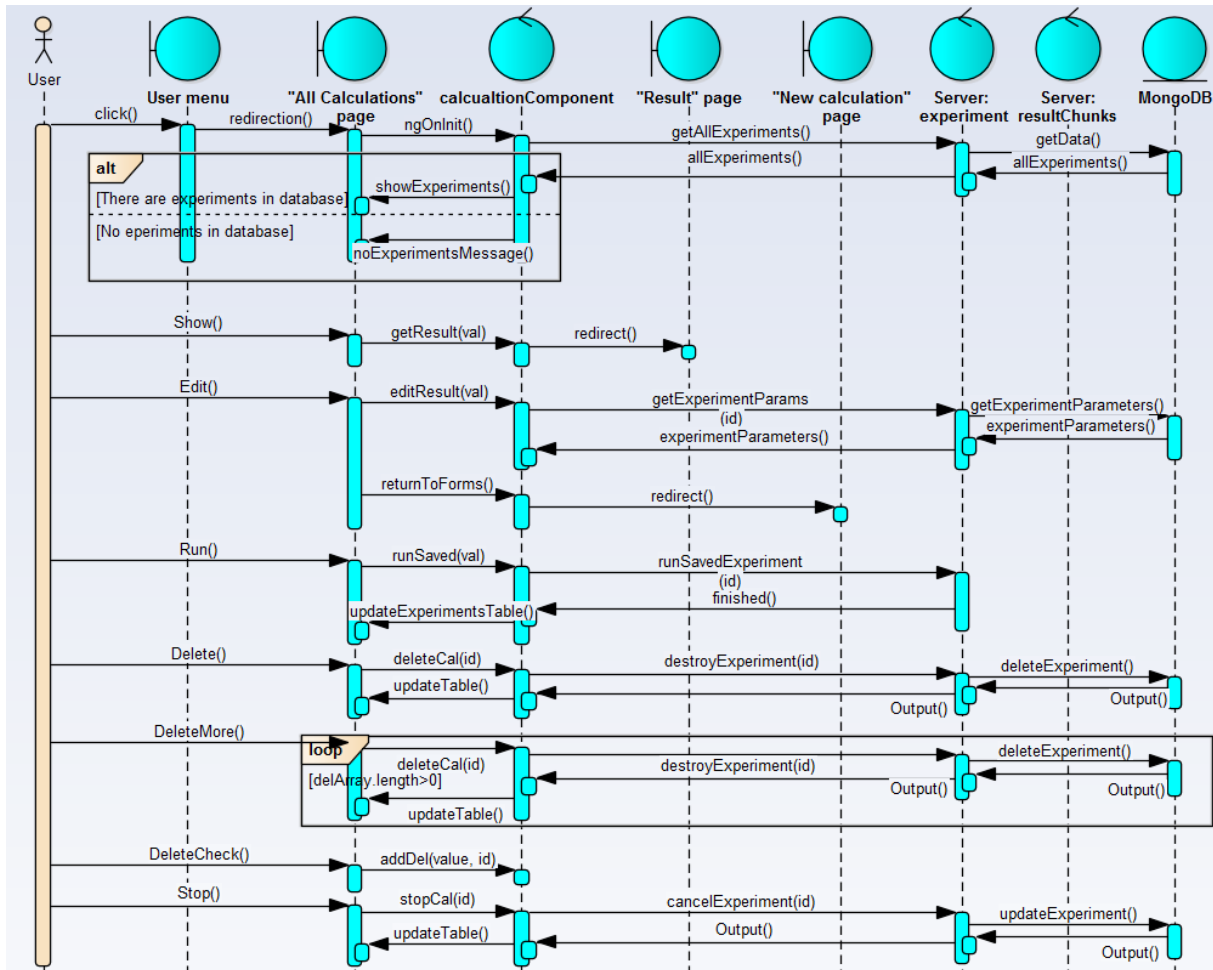
Zahtevi:

Samo prijavljen korisnik može da obavlja ovu aktivnost. Polja za unos moraju da budu ispravno popunjena. Neophodno je korisnika obaveštavati o greškama tako da on može u svakom trenutku da se snađe i ispravi greške.

6.1.5. Pregledanje stranice sa svim proračunima



Slika 12 - Slučaj korišćenja: Pregledanje stranice sa svim proračunima



Slika 13 - Dijagram sekvenci pregleda tabele sa proračunima

Kratak opis: Pregledanje stranica sa izlistanim proračunima svih korisnika i korišćenje njenih funkcionalnosti.

Opis slučaja korišćenja:

Otvaranjem stranice sa proračunima korisnika, biće prikazana tabela sa proračunima svih korisnika. Tabela će sadržati informacije o naslovu proračuna, komentaru na proračun kao i datumu njegovog upisa u bazu. Proračun se u tabeli može naći kao sačuvani proračun, nepročitani proračun čije će rezultate korisnik moći da pregleda, pročitani proračun čije će rezultat korisnik moći da pregleda i kao proračun koji je u trenutku pregledanja tabele u izvršavanju ili na čekanju. U zavisnosti od tipa proračuna odgovarajući red u tabeli biće obojen odgovarajućom bojom. Red u tabeli koji će prikazivati informacije o sačuvanom proračunu imaće i dve dodatne funkcionalnosti. Korisnik će imati mogućnost brisanja proračuna.

Glavni tok događaja:

Korisnik će pristupom stranici sa proračunima moći da pristupi listi svih proračuna. Korisnik će moći da pročita informacije o naslovu proračuna, komentaru na proračun i datumu upisa u bazu, kao i podatak o statusu proračuna.

Ukoliko je neki od njegovih proračuna prikazanih u tabelu u toku, korisnik će moći da vidi progress bar sa podatkom o tome koliko je preostalo do kraja.

Korisnik će moći da vidi i koji su od proračuna na čekanju u redu za izvršavanje. Korisnik će imati mogućnost brisanje određenog proračuna ili selektovanje i brisanje nekoliko njih. Ukoliko je korisnik administrator moći će da briše sve proračune, u suprotnom samo svoje. Klikom na dugme „Prikaži“ korisnik će biti redirektovan na stranicu za prikaz rezultata proračuna. Klikom na dugme „Izmeni“ korisnik će biti redirektovan na stranicu „Novi proračun“ sa ulaznim parametrima proračuna koji želi da izmeni. Klikom na dugme „Pokreni“ korisnik će moći da pokrene sačuvani proračun. Klikom na dugme „Zaustavi“ korisnik će moći da zaustavi svoj proračun, koji će se potom naći u tabeli kao sačuvani proračun.

Slučaj korišćenja se završava odjavljivanjem, gašenjem aplikacije ili redirektovanjem. Izvršavanjem neke od željenih akcija klikom na odgovarajuće dugme ili izlistavanjem željene stranice sa proračunima.

Definisanje uslova:

Pred-uslov: Prijavljivanje na sistem i pristup stranici sa proračunima korisnika

Post-uslov: Regularnost baze podataka

Alternativni tokovi događaja:

Ukoliko nema proračuna u bazi korisnik će biti obavešten porukom. Nemogućnost pokretanja proračuna usled pada servera za izvršavanje proračuna.

Zahtevi:

Ukoliko je se proračun izvršava potrebno je da kao polje u tabeli postoji progress bar koji će korisnika izveštavati o njegovom trenutnom stanju. Potrebno je da tabela bude pregledna i da jedna stranica tabele ne sadrži previše redova. Potrebna je mogućnost zaustavljanja proračuna, kao i stavljanje proračuna u red za čekanje. Potrebno je obavestiti korisnika porukom pri svakoj od bitnijih akcija, uz pitanje o potvrdi.

6.1.6. Pregled stranice sa sačuvanim proračunima

Kratak opis: Pregled stranice sa sačuvanim proračunima i korišćenje njenih funkcionalnosti.

Opis slučaja korišćenja:

Korisnik će moći da pristupi stranici sa sačuvanim proračunima svih korisnika koristeći korisnički meni. Ovaj slučaj korišćenja sličan je slučaju opisanom u tački 5.1.5., sa ograničenim mogućnostima, u smislu nedostupnosti nekih opcija. Korisnik će moći da izmeni, pokrene i obriše proračun. Ostale mogućnosti će biti nedostupne.

Glavni tok događaja:

Korisnik pristupa stranici korišćenjem korisničkog menija ili putem stranice „Novi proračun“ ukoliko odluči da sačuva proračun. Glavni tok događaja sličan je onom opisanom u tački 5.1.5. Razlika je u tome što će korisnik moći samo da izmeni, pokrene i obriše proračun (u zavisnosti od administratorskih prava).

Slučaj korišćenja se završava odjavljivanjem, gašenjem aplikacije ili redirektovanjem.

Definisanje uslova:

Pred-uslov: Prijavljivanje na sistem i pristup stranici sa sačuvanim proračunima korisnika.

Post-uslov: Regularnost baze podataka.

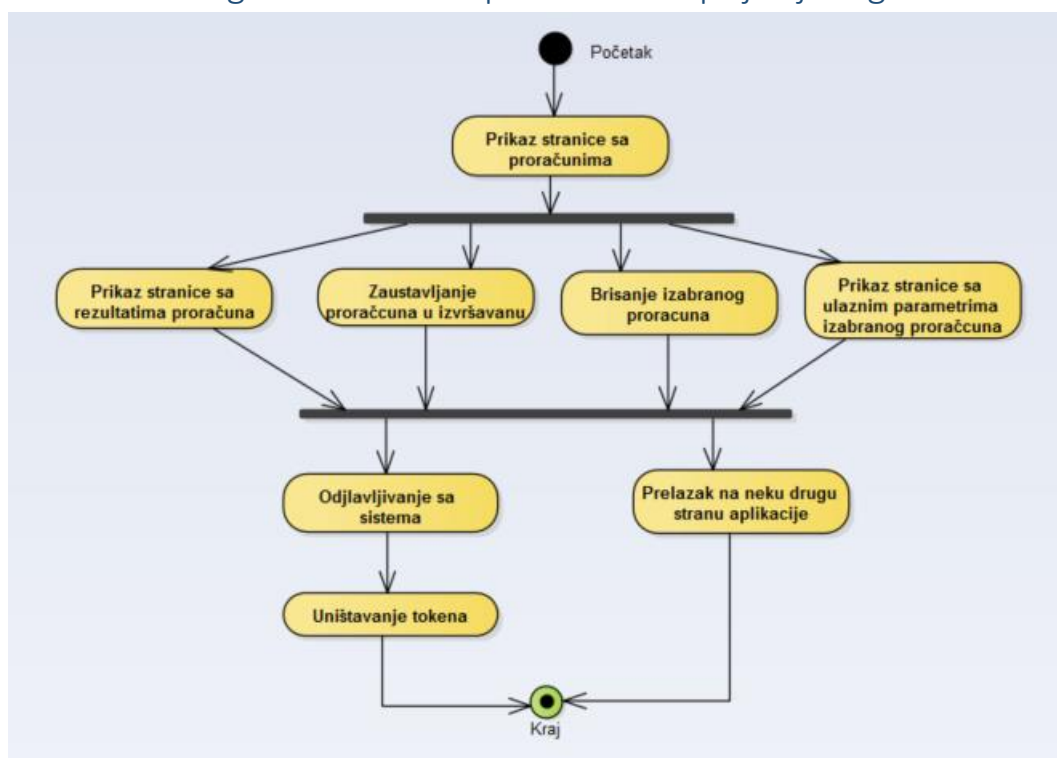
Alternativni tokovi događaja:

Ukoliko nema proračuna u bazi korisnik će biti obavešten porukom. Nemogućnost pokretanja proračuna usled pada servera za izvršavanje proračuna.

Zahtevi:

Potrebno je da tabela bude pregledna i da jedna stranica tabele ne sadrži previše redova. Potrebno je obavestiti korisnika porukom pri svakoj od bitnijih akcija, uz pitanje o potvrdi. Potrebno je prikazati sve proračune, a ne samo proračune trenutno ulogovanog korisnika.

6.1.7. Pregled stranice sa proračunima prijavljenog korisnika



Slika 14 - Dijagram aktivnosti prikazivanja stranice sa korisnikovim proračunima

Kratak opis: Pregled stranice sa proračunima prijavljenog korisnika i korišćenje njenih funkcionalnosti.

Opis slučaja korišćenja:

Korisnik će moći da pristupi stranici sa svojim proračunima koristeći korisnički meni. Ovaj slučaj korišćenja sličan je slučaju opisanom u tački 5.1.5. s tim što korisnik vidi samo svoje proračune.

Glavni tok događaja:

Korisnik pristupa stranici korišćenjem korisničkog menija ili putem stranice „Novi proračun“ ukoliko odluči da pokrene proračun. Glavni tok događaja sličan je onom opisanom u tački 5.1.5. Razlika je u tome što će korisnik moći da vidi samo svoje proračune.

Slučaj korišćenja se završava odjavljivanjem, gašenjem aplikacije ili redirektovanjem.

Definisanje uslova:

Pred-uslov: Prijavljivanje na sistem i pristup stranici sa svojim proračunima korisnika.

Post-uslov: Regularnost baze podataka.

Alternativni tokovi događaja:

Ukoliko nema proračuna u bazi korisnik će biti obavešten porukom. Nemogućnost pokretanja proračuna usled pada servera za izvršavanje proračuna.

Zahtevi:

Ukoliko je se proračun izvršava potrebno je da kao polje u tabeli postoji progress bar koji će korisnika izveštavati o njegovom trenutnom stanju. Potrebno je da tabela bude pregledna i da jedna stranica tabele ne sadrži previše redova. Potrebna je mogućnost zaustavljanja proračuna, kao i stavljanje proračuna u red za čekanje. Potrebno je obavestiti korisnika porukom pri svakoj od bitnijih akcija, uz pitanje o potvrdi.

6.1.8. Pregled stranice sa nepročitanim proračunima

Kratak opis: Pregled stranice sa nepročitanim proračunima i korišćenje njenih funkcionalnosti.

Opis slučaja korišćenja:

Korisnik će moći da pristupi stranici sa nepročitanim proračunima koristeći ikonicu sa obaveštenjima koja će se nalaziti u glavnom meniju. Ovaj slučaj korišćenja sličan je slučaju opisanom u tački 5.1.7. s tim što korisnik vidi samo svoje nepročitane proračune.

Glavni tok događaja:

Korisnik pristupa stranici korišćenjem ikonice za obaveštenja u glavnom meniju. Glavni tok događaja sličan je onom opisanom u tački 5.1.7. Razlika je u tome što će korisnik moći da vidi samo svoje nepročitane proračune.

Slučaj korišćenja se završava odjavljivanjem, gašenjem aplikacije ili redirektovanjem.

Definisanje uslova:

Pred-uslov: Prijavljivanje na sistem i pristup stranici sa nepročitanim proračunima korisnika.

Post-uslov: Regularnost baze podataka.

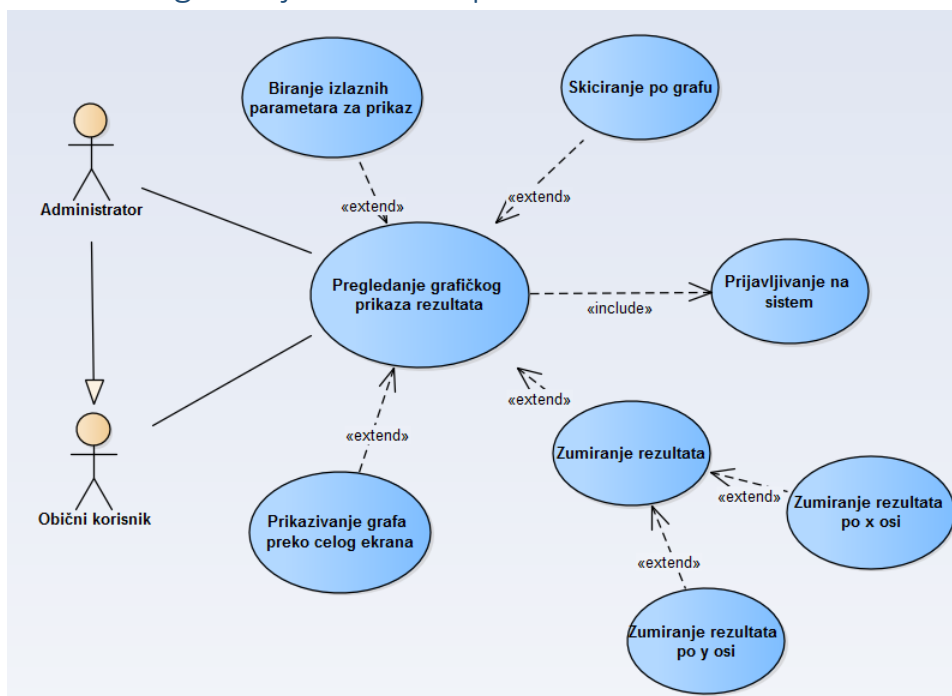
Alternativni tokovi događaja:

Ukoliko nema proračuna u bazi korisnik može da se redirektuje samo na stranicu sa svim svojim proračunima.

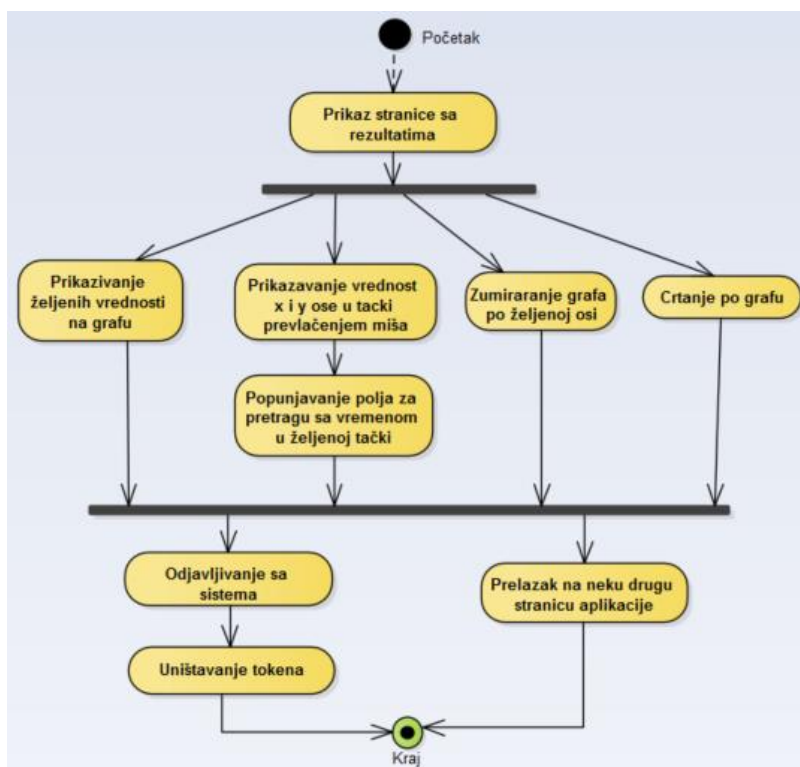
Zahtevi:

Potrebno je da tabela bude pregledna i da jedna stranica tabele ne sadrži previše redova. Potrebno je obavestiti korisnika porukom pri svakoj od bitnijih akcija, uz pitanje o potvrdi.

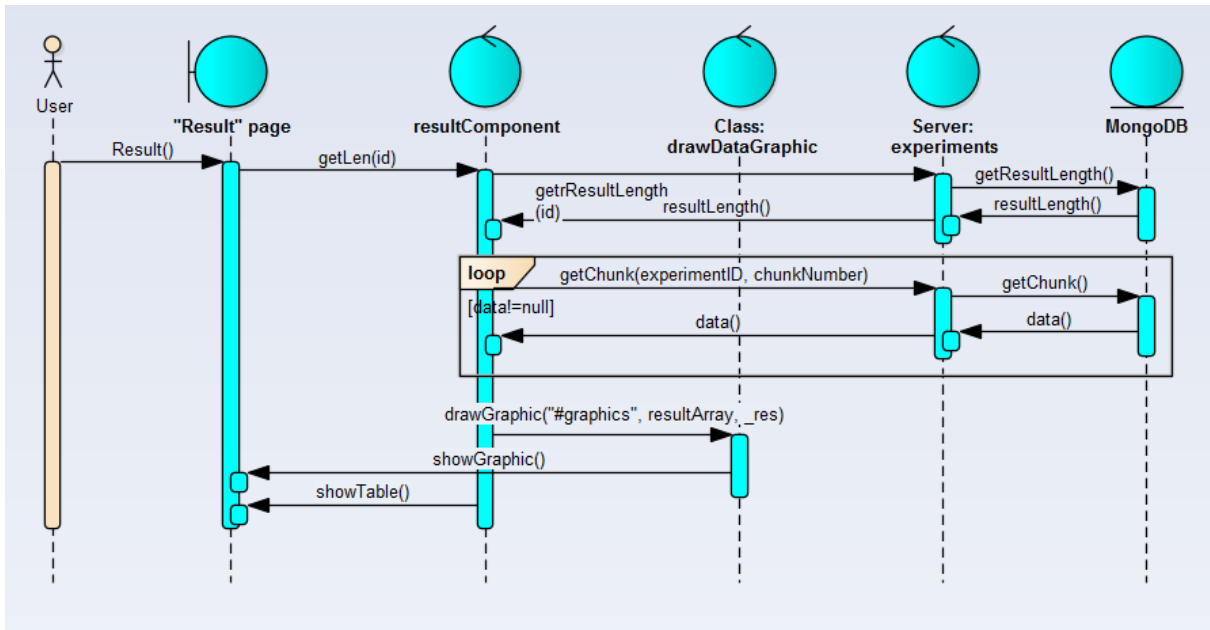
6.1.9. Pregledanje rezultata proračuna



Slika 15 - Slučaj korišćenja: Pregledanje grafičkog prikaza rezultata proračuna



Slika 16 - Dijagram aktivnosti interakcije sa grafom



Slika 17 - Dijagram sekvenci interakcije sa grafom

Kratak opis: Pregledanje grafičkog prikaza rezultata proračuna

Opis slučaja korišćenja:

Rezultati proračuna biće prikazani na grafiku u obliku linije. Korisnik će moći da bira parametre čije promene u toku vremena želi da posmatra. Korisnik će odabirom „Force/CB“ opcije moći da uporedi ponašanje više parametara u toku vremena. Prelaskom preko linije korisnik će dobiti obaveštenje o vrednostima parametara u toj tački, a klikom na liniju, vreme u toj tački upisivaće se u polje za pretragu tabele po vremenu. Korisniku će biti prikazan meni sa alatnama, a u zavisnosti od izbora alatke imaće neku od sledećih mogućnosti. Grafik će moći da se zumira, a korisnik će birati jedan od dostupnih načina za zumiranje. Dostupni načini za zumiranje biće zumiranje po x osi, zumiranje po y osi i po obe ose. Korisnik će moći da prikaže graf preko celog ekrana. Korisnik će odabirom alatke za crtanje moći da skicira po grafu. Korisnik će klikom na dugme „Sačuvaj sliku grafa“ moći da preuzme sliku grafa u njegovom trenutnom stanju.

Glavni tok događaja:

Korisnik stranici sa rezultatom proračuna pristupa preko bilo koje od stranica sa izlistanim proračunima klikom na dugme „Prikaži“ ili klikom na ikonicu za obaveštenja pri čemu se otvara padajući meni, u kojem korisnik može da izabere jedan od nepročitanih proračuna ukoliko ih ima. Pošto učitavanje podataka iz baze traje neko vreme, korisnik će biti obavešten o statusu putem progress bara.

Rezultati proračuna biće prikazani na grafiku u obliku linije. Korisnik će moći da bira parametre čije promene u toku vremena želi da posmatra. Korisnik će odabirom „Force/CB“ opcije moći da uporedi ponašanje više parametara u toku vremena. Prelaskom preko linije korisnik će dobiti obaveštenje o vrednostima parametara u toj tački, a klikom na liniju, vreme u toj tački upisivaće se u polje za pretragu tabele po vremenu. Korisniku će biti prikazan meni sa alatnama, a u zavisnosti od izbora alatke imaće neku od sledećih mogućnosti. Grafik će moći

da se zumira, a korisnik će birati jedan od dostupnih načina za zumiranje. Dostupni načini za zumiranje biće zumiranje po x osi, zumiranje po y osi i po obe ose. Korisnik će moći da prikaže graf preko celog ekrana. Korisnik će odabirom alatke za crtanje moći da skicira po grafu.

Slučaj korišćenja završava se gašenjem aplikacije, odjavljivanjem ,redirekcijom na drugu stranicu ili prikazom željenog rezultata.

Definisanje uslova:

Pred-uslov: Pokretanje aplikacije, prijavljivanje na sitem i pristup stranici sa rezultatom proračuna.

Post-uslov: Regularnost baze podataka.

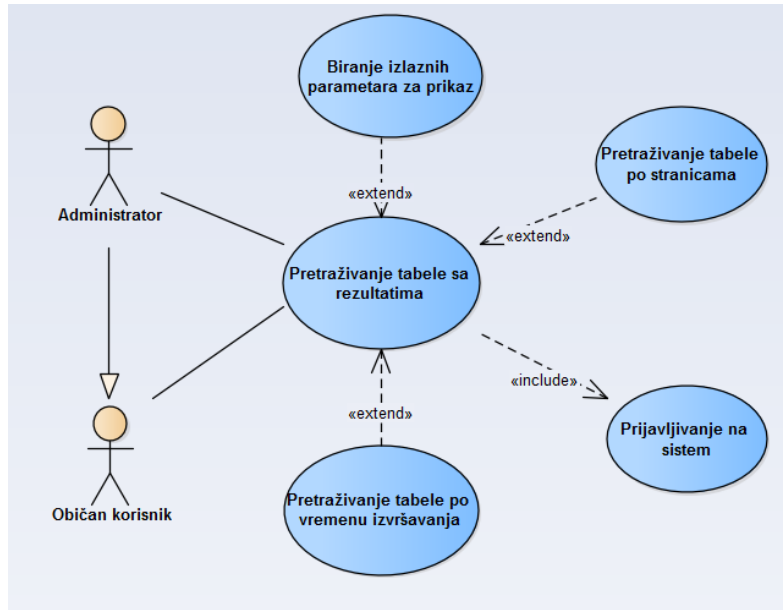
Alternativni tok događaja:

Nemogućnost učitavanja podataka iz baze.

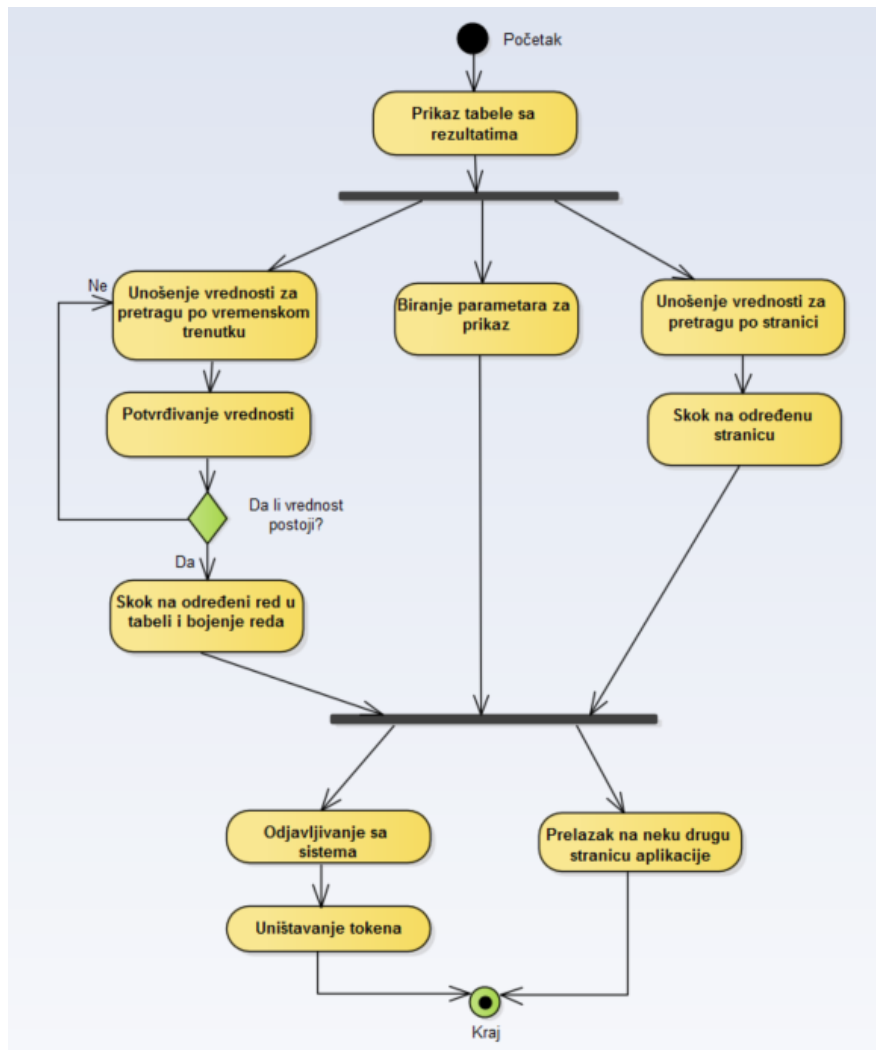
Zahtevi:

Zumiranje grafika. Zumiranje grafika po x i y osi. Korišćenje što više tačaka za iscrtavanje grafika. Mogućnost izbora više parametara za prikaz u odnosu na vreme.

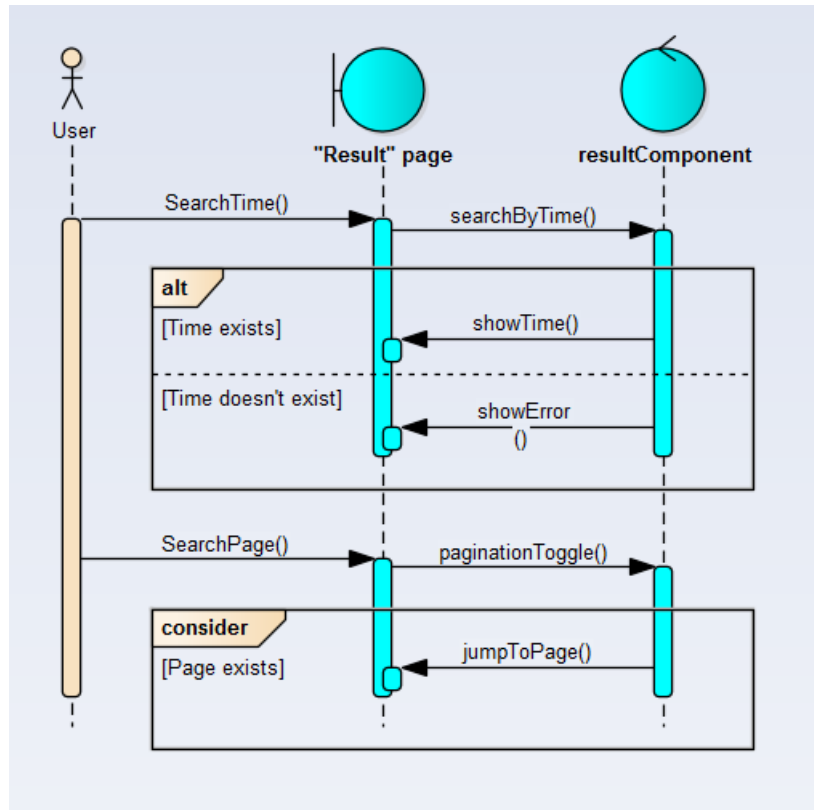
6.1.10. Pretraživanje tabele sa rezultatom



Slika 18 - Slučaj korišćenja: Pretraživanje tabele sa rezultatom



Slika 19 - Dijagram aktivnosti pretraživanje tabele



Slika 20 - Dijagram sekvenci pretraživanja tabele

Kratak opis: Pretraživanje tabele koja sadrži rezultat proračuna, po jedna linija za svaki vremenski trenutak.

Opis slučaja korišćenja:

Rezultati proračuna biće prikazani tabelarno. Redovi u tabeli sadržaće podatke o parametrima i o njihovim vrednostima koje se menjaju kroz vreme. Korisnik će moći da izabere koje parametre želi da pregleda a koje ne. Korisnik će moći da vrši pretragu tabele po vremenu unosom željenog vremena u polje za pretragu. Pritiskom na taster enter, ukoliko rezultat postoji, red u tabeli biće obojen u zeleno a korisnik prebačen na stranicu tabele na kojoj se red nalazi. Korisnik će takođe moći da vrši pretragu tabele odabirom stranice na koju želi da skoči.

Glavni tok događaja:

Korisnik stranici sa rezultatom proračuna pristupa preko bilo koje od stranica sa izlistanim proračunima klikom na dugme „Prikaži“ ili klikom na ikonicu za obaveštenja pri čemu se otvara padajući meni, u kojem korisnik može da izabere jedan od nepročitanih proračuna ukoliko ih ima. Pošto učitavanje podataka iz baze traje neko vreme, korisnik će biti obavešten o statutu putem progress bara.

Korisnik bira parametre za prikaz klikom na checkbox sa imenom parametra koji želi/ne želi da vidi u tabeli. Po učitavanju rezultata svi parametri su vidljivi. Korisnik vrši pretragu tabele unosom željenog vremena u input polje za pretragu. Klikom na enter, red u tabeli koji se poklapa sa unetim vremenom je prikazan i obojen u zeleno. Ukoliko ne postoji uneseno vreme korisnik je obavešten tako što se polje za unos boji u crveno. Korisnik vrši pretragu i odabirom stranice na koju želi da skoči.

Slučaj korišćenja se završava prikazivanjem željenog vremenskog rezultata ili skokom na željenu stranicu. Slučaj se može završiti i gašenjem aplikacije, odjavljivanjem ili redirekcijom na drugu stranicu.

Definisanje uslova:

Pred-uslov: Pokretanje aplikacije, prijavljivanje na sitem i pristup stranici sa rezultatom proračuna.

Post-uslov: Regularnost baze podataka.

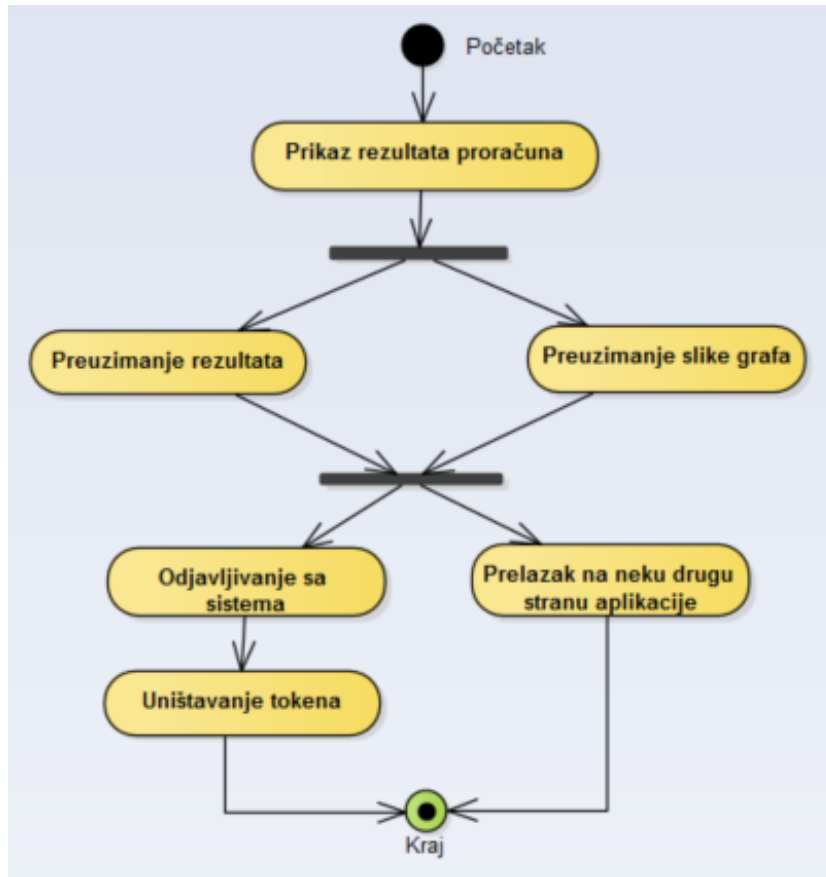
Alternativni tok događaja:

Nemogućnost učitavanja podataka iz baze. Ukoliko ne postoji vreme koje korisnik hoće da pretraži u tabeli korisnik je obavešten tako što se polje za pretragu zacrveni.

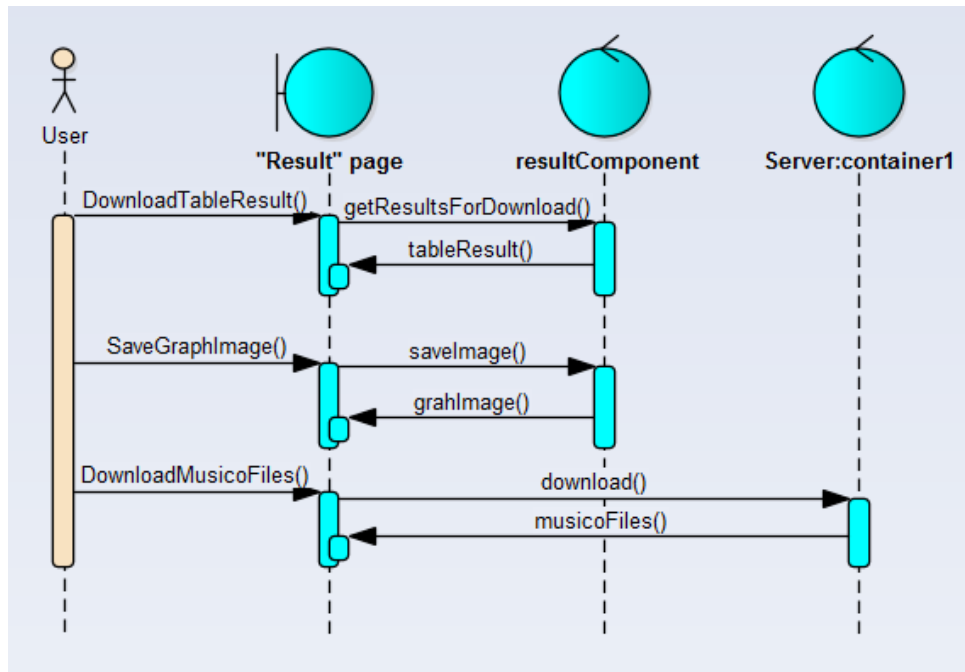
Zahtevi:

Paginacija. Preglednost tabele. Odabir parametara za prikaz.

6.1.11. Čuvanje rezultata proračuna



Slika 21 - Dijagram aktivnosti čuvanja rezultata proračuna



Slika 22 - Dijagram sekvenci čuvanja rezultata proračuna

Kratak opis: Čuvanje rezultata proračuna

Opis slučaja korišćenja:

Korisnik će klikom na dugme „Sačuvaj tabelu rezultata“ moći da preuzme rezultat proračuna u obliku .csv fajla. Korisnik će klikom na dugme „Sačuvaj sliku grafa“ moći da preuzme sliku grafa u njegovom trenutnom stanju. Korisnik će klikom na dugme „Sačuvaj musico fajlove“ moći da preuzme sve fajlove koje generiše softver MUSICO u obliku .zip fajla.

Glavni tok događaja:

Korisnik stranici sa rezultatom proračuna pristupa preko bilo koje od stranica sa izlistanim proračunima klikom na dugme „Prikaži“ ili klikom na ikonicu za obaveštenja pri čemu se otvara padajući meni, u kojem korisnik može da izabere jedan od nepročitanih proračuna ukoliko ih ima. Pošto učitavanje podataka iz baze traje neko vreme, korisnik će biti obavešten o statusu putem progress bara.

Korisnik će klikom na dugme „Sačuvaj tabelu rezultata“ moći da preuzme rezultat proračuna u obliku .csv fajla. Korisnik će klikom na dugme „Sačuvaj sliku grafa“ moći da preuzme sliku grafa u njegovom trenutnom stanju. Korisnik će klikom na dugme „Sačuvaj musico fajlove“ moći da preuzme sve fajlove koje generiše softver MUSICO u obliku .zip fajla.

Slučaj korišćenja se završava preuzimanjem podataka, gašenjem aplikacije, odjavljivanjem ili redirekcijom na drugu stranicu.

Definisanje uslova:

Pred-uslov: Pokretanje aplikacije, prijavljivanje na sitem i pristup stranici sa rezultatom proračuna.

Post-uslov: Regularnost baze podataka.

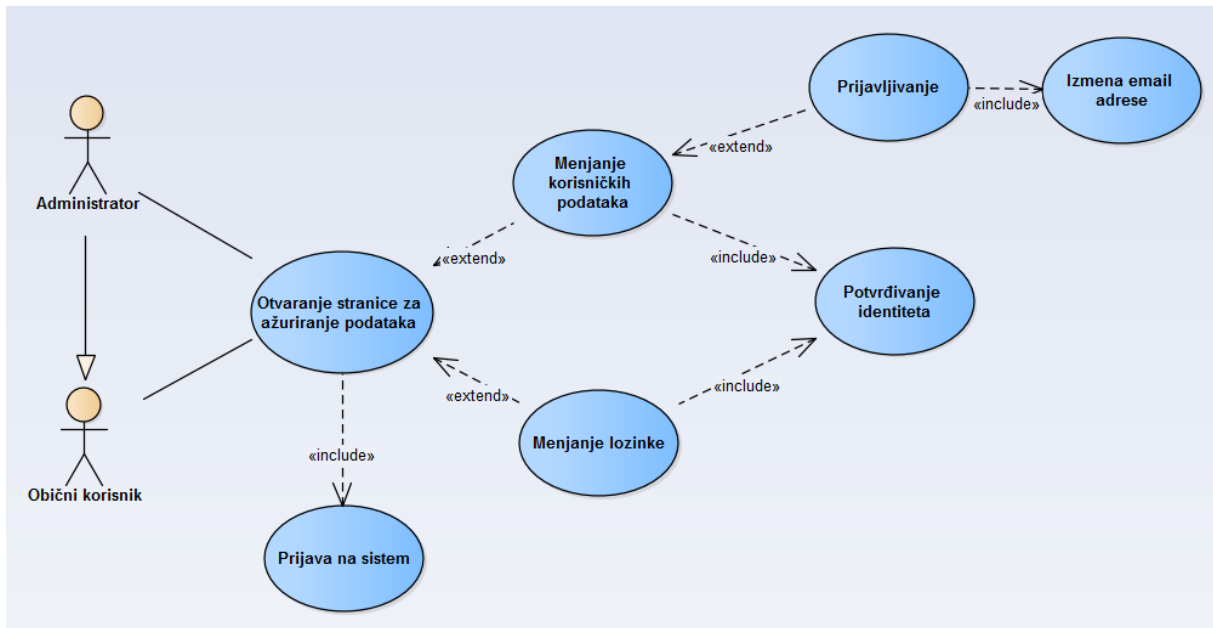
Alternativni tok događaja:

Nemogućnost učitavanja podataka iz baze.

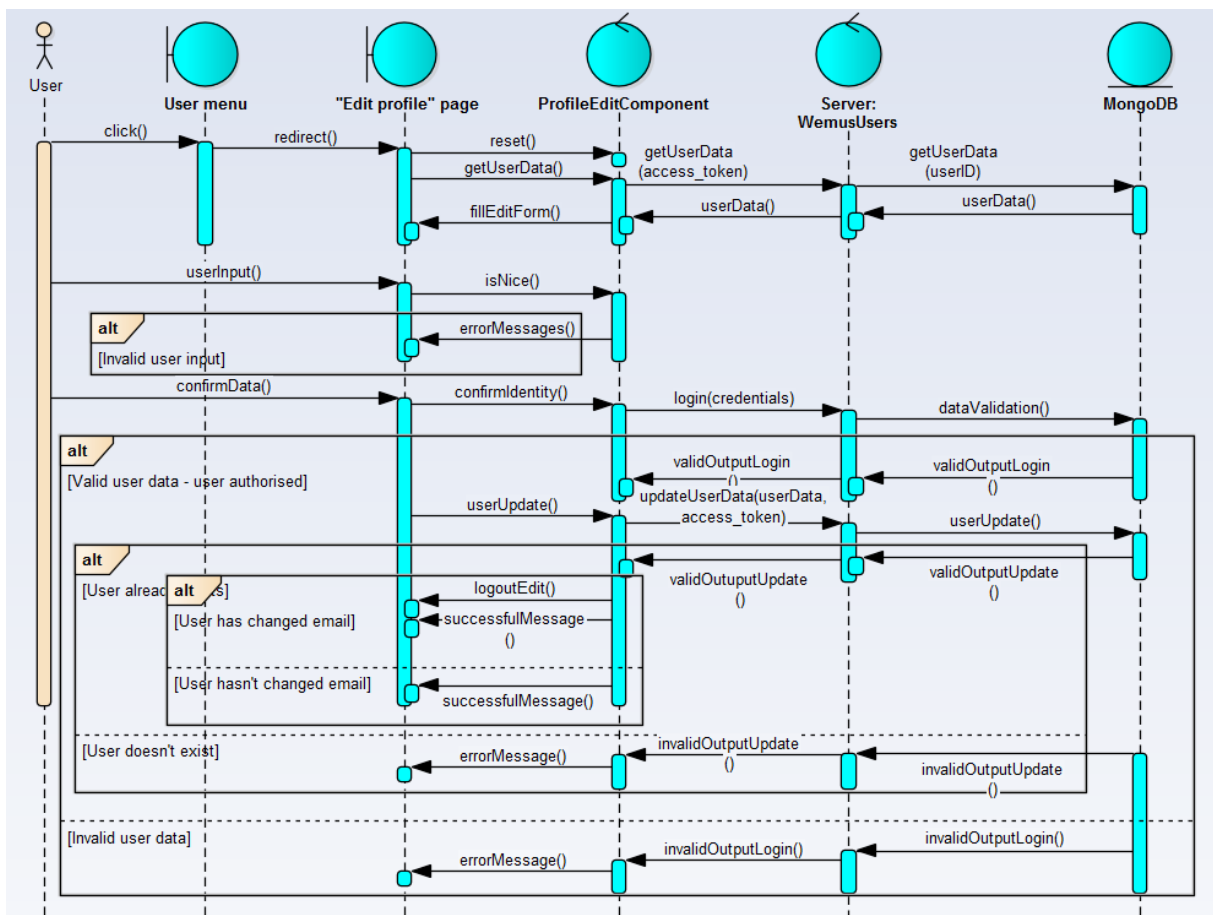
Zahtevi:

Mogućnost preuzimanja slike grafa. Mogućnost preuzimanja svih izlaznih fajlova.

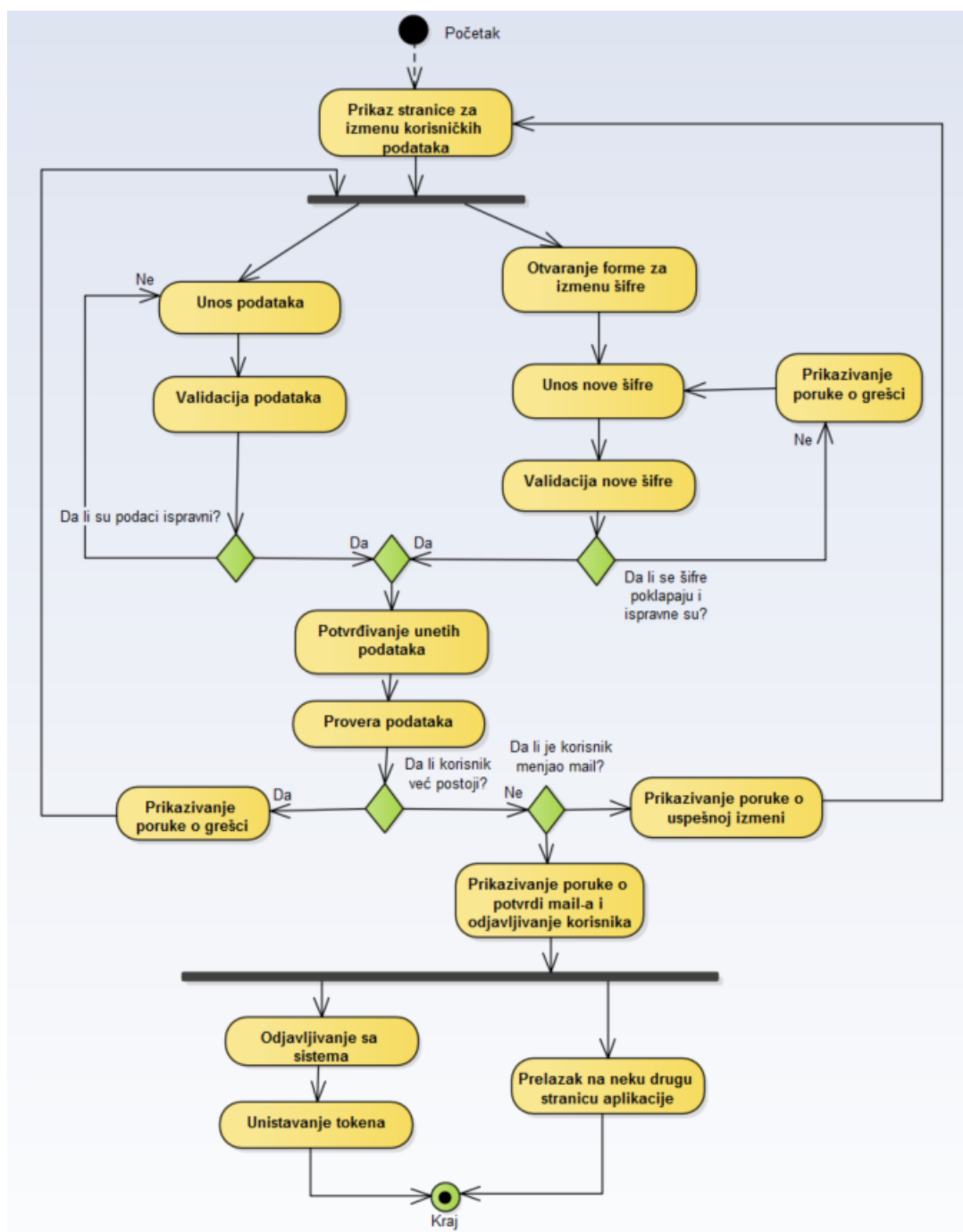
6.1.12. Ažuriranje korisničkih podataka



Slika 23 - Slučaj korišćenja: Ažuriranje podataka korisnika



Slika 24 - Dijagram sekvenci ažuriranja podataka korisnika



Slika 25 - Dijagram aktivnosti izmene korisnikovih ličnih podataka

Kratak opis: Ažuriranje podataka trenutno ulogovanog korisnika.

Opis slučaja korišćenja:

Otvaranjem stranice „Izmenite podatke“ biće prikazana forma koja će biti popunjena korisnikovim podacima. Korisnik će podatke menjati promenom vrednosti polja za unos. Klikom na dugme za promenu lozinke korisnik će dobiti i tu mogućnost. Korisnik mora da unese ime, prezime, elektronsku poštu i lozinku u ispravnom formatu, u suprotnom će porukom

ispod polja za unos biti obavešten o grešci. Klikom na dugme „Sačuvaj promene“ ukoliko korisnički nalog već postoji korisnik će biti obavešten porukom.

Glavni tok događaja:

Otvaranjem stranice „Izmenite podatke“ biće prikazana forma koja će biti popunjena korisnikovim podacima. Korisnik će podatke menjati promenom vrednosti polja za unos. Klikom na dugme za promenu lozinke korisnik će dobiti i tu mogućnost. Korisnik mora da unese ime, prezime, elektronsku poštu i lozinku u ispravnom formatu, u suprotnom će porukom ispod polja za unos biti obavešten o grešci. Pre nego što sačuva promene korisnik će morati da unese svoju trenutnu šifru. Klikom na dugme „Sačuvaj promene“ ukoliko korisnički nalog već postoji korisnik će biti obavešten porukom. Ukoliko je korisnik pogrešio trenutnu šifru takođe će biti obavešten porukom. Ukoliko korisnik nije menjao email, biće obavešten o uspešnoj izmeni podataka. Ukoliko je korisnik menjao email biće odjavljen uz obaveštenje o ponovnoj potvrdi mail adrese.

Slučaj korišćenja se završava odjavljivanjem, gašenjem aplikacije, redirekcijom na drugu stranicu ili uspešnim ažuriranjem podataka.

Definisanje uslova:

Pred-uslov: Pokretanje i prijavljivanje na sistem i stranica za ažuriranje podataka.

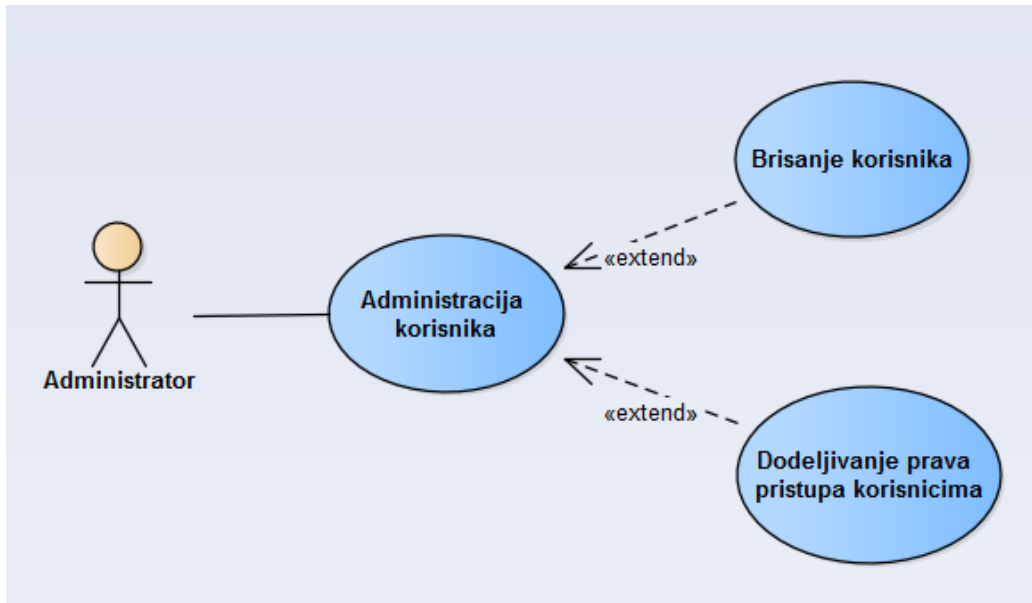
Post-uslov: Regularnost baze podataka.

Alternativni tok događaja:

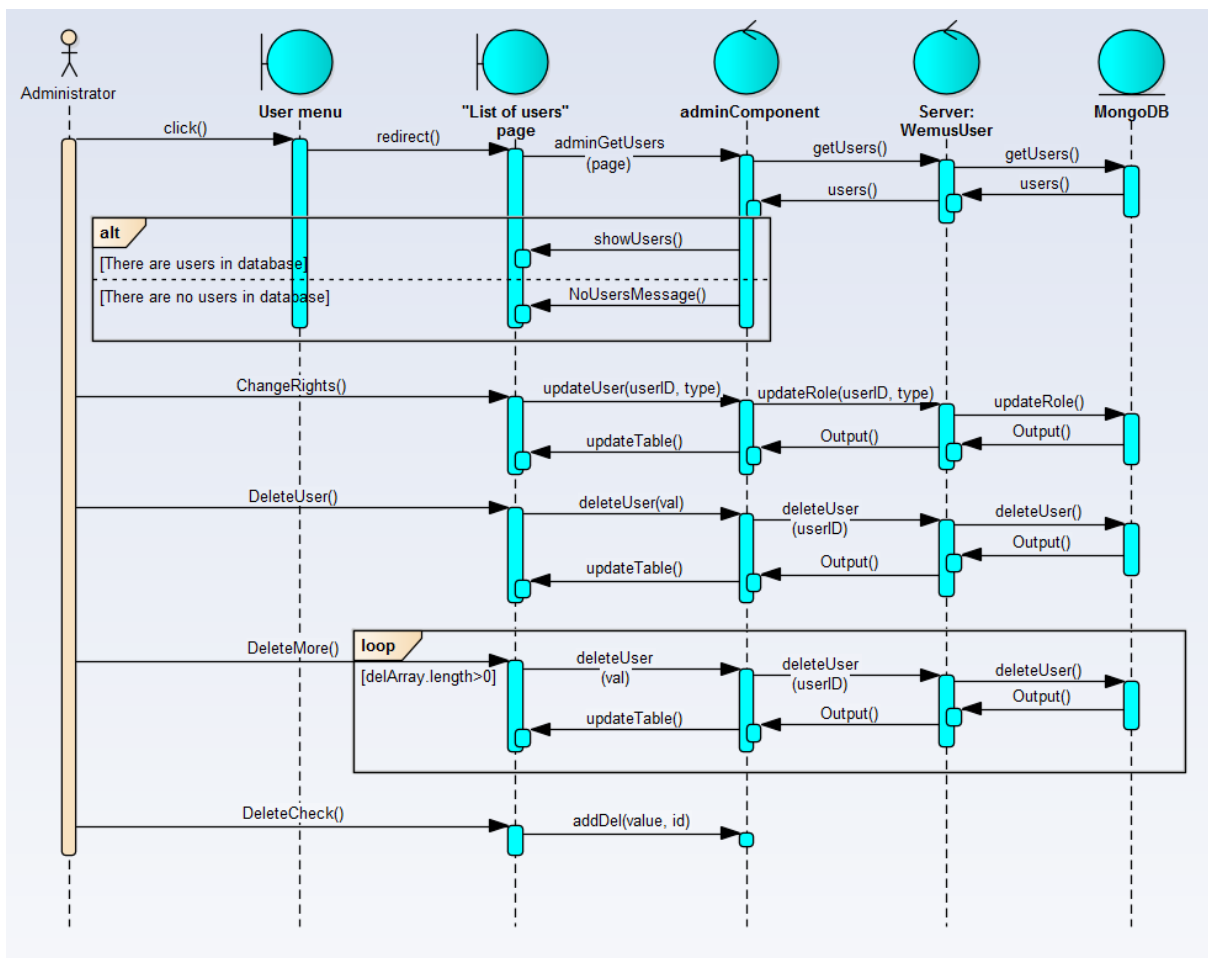
Ukoliko korisnik sa istim korisničkim imenom ili email adresom postoji u bazi korisnik će biti obavešten. Ukoliko polja za unos nisu validna, korisnik će biti obavešten. Ukoliko korisnik nije ispravno uneo šifru za potvrdu promene biće obavešten porukom.

Zahtevi: /

6.1.13. Administracija korisnika



Slika 26 - Slučaj korišćenja: Administracija korisnika



Slika 27 - Dijagram sekvenci administracije korisnika

Kratak opis: Administracija korisnika od strane administratora sistema.

Opis slučaja korišćenja:

Korisnik koji ima administratorske privilegije, imaće pristup stranici za administraciju. Stranica će sadržati tabelarni prikaz korisnika sistema. Svaki red tabele sadržaće nadimak, korisnički mail, prezime, ime i tip korisnika.

Administrator će klikom na dugme „Obriši korisnika“ moći da ukloni korisnika iz sistema.

Administrator će izborom jedne od ponuđenih uloga moći da izmeni prava korisnika u sistemu.

Glavni tok događaja:

Korisnik sa administratorskim pravima pristupa stranici za administraciju korišćenjem korisničkog menija. Klikom na dugme obriši korisnika otvara se prozor za potvrdu, brisanjem korisnika se tabela ažurira a korisnik briše iz sistema. Odabirom uloge i klikom na dugme za potvrdu kao i potvrdnim odgovorom na pitanje da li zaista želi da promeni ulogu korisnika, njegova uloga se menja i tabela korisnika se ažurira. Administrator će moći da izlista sve korisnike sistema sem samog sebe. Ukoliko odgovori odrično na bilo koji prozor za potvrdu tabela se ne menja.

Slučaj korišćenja se završava izmenom tabele, redirekcijom, odjavljivanjem ili gašenjem aplikacije.

Definisanje uslova:

Pred-uslov: Pokretanje aplikacije, prijavljivanje na sitem i pristup stranici za administraciju, što podrazumeva posedovanje administratorskih prava.

Post-uslov: Regularnost baze podataka.

Alternativni tok događaja:

Ukoliko ne postoje korisnici u bazi, administrator će biti obavešten porukom.

Zahtevi: Moguće uloge korisnika u sistemu moraju biti izlistane u padajućem meniju.

6.1.14. Promena jezika na aplikaciji

Kratak opis: Promena jezika aplikacije.

Opis slučaja korišćenja: Korisnik će moći da bira između dva jezika, engleskog i srpskog.

Glavni tok događaja:

Korisnik će koristeći padajući meni u glavnom meniju moći da bira između dva jezika, engleskog i srpskog. Slučaj korišćenja se završava odabirom jezika.

Definisanje uslova:

Pred-uslov: Pokretanje WEMUS aplikacije.

Alternativni tok događaja: Loše prevedene reči.

Zahtevi: /

6.1.15. Odjavljivanje korisnika sa aplikacije

Kratak opis: Odjavljivanje sa sistema

Opis slučaja korišćenja:

Ukoliko je prijavljen na sistem, korisnik se u svakom trenutku može odjaviti korišćenjem glavnog menija. Nakon će biti redirektovan na stranicu za prijavljivanje.

Glavni tok događaja:

Ukoliko je prijavljen na sistem, korisnik se u svakom trenutku može odjaviti korišćenjem glavnog menija. Nakon će biti redirektovan na stranicu za prijavljivanje. Slučaj korišćenja se završava odjavljivanjem.

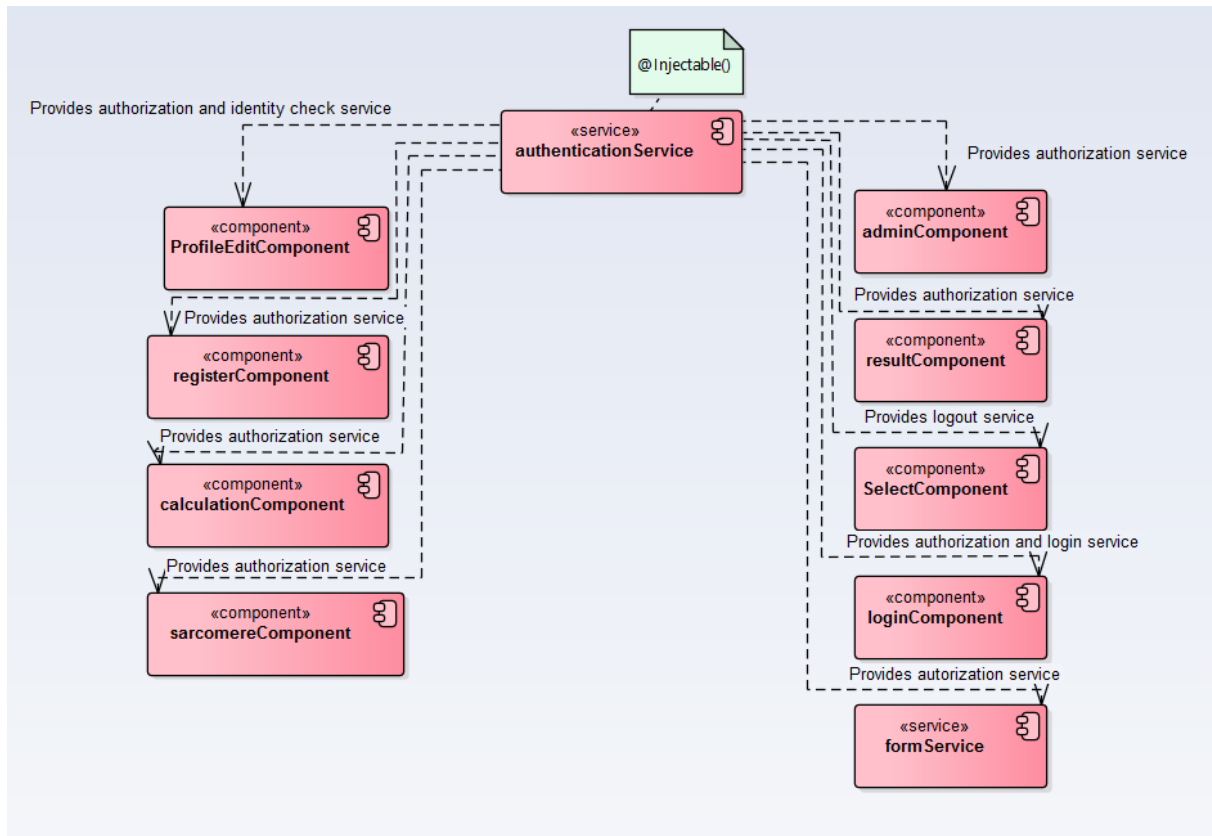
Definisanje uslova:

Pred uslov: Pokretanje aplikacije i prijavljivanje na sistem.

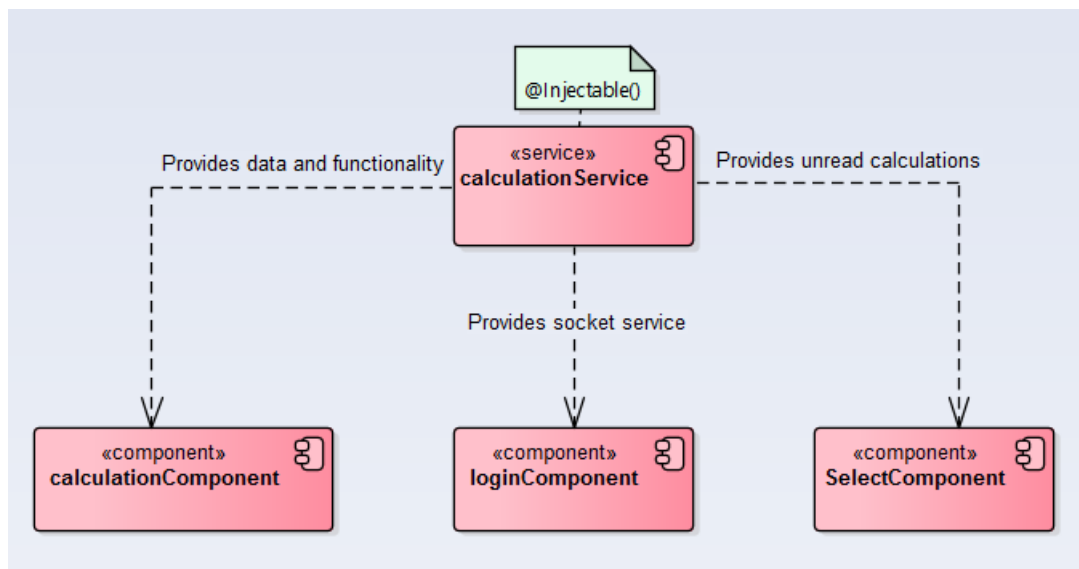
Alternativni to događaja: /

Zahtevi: /

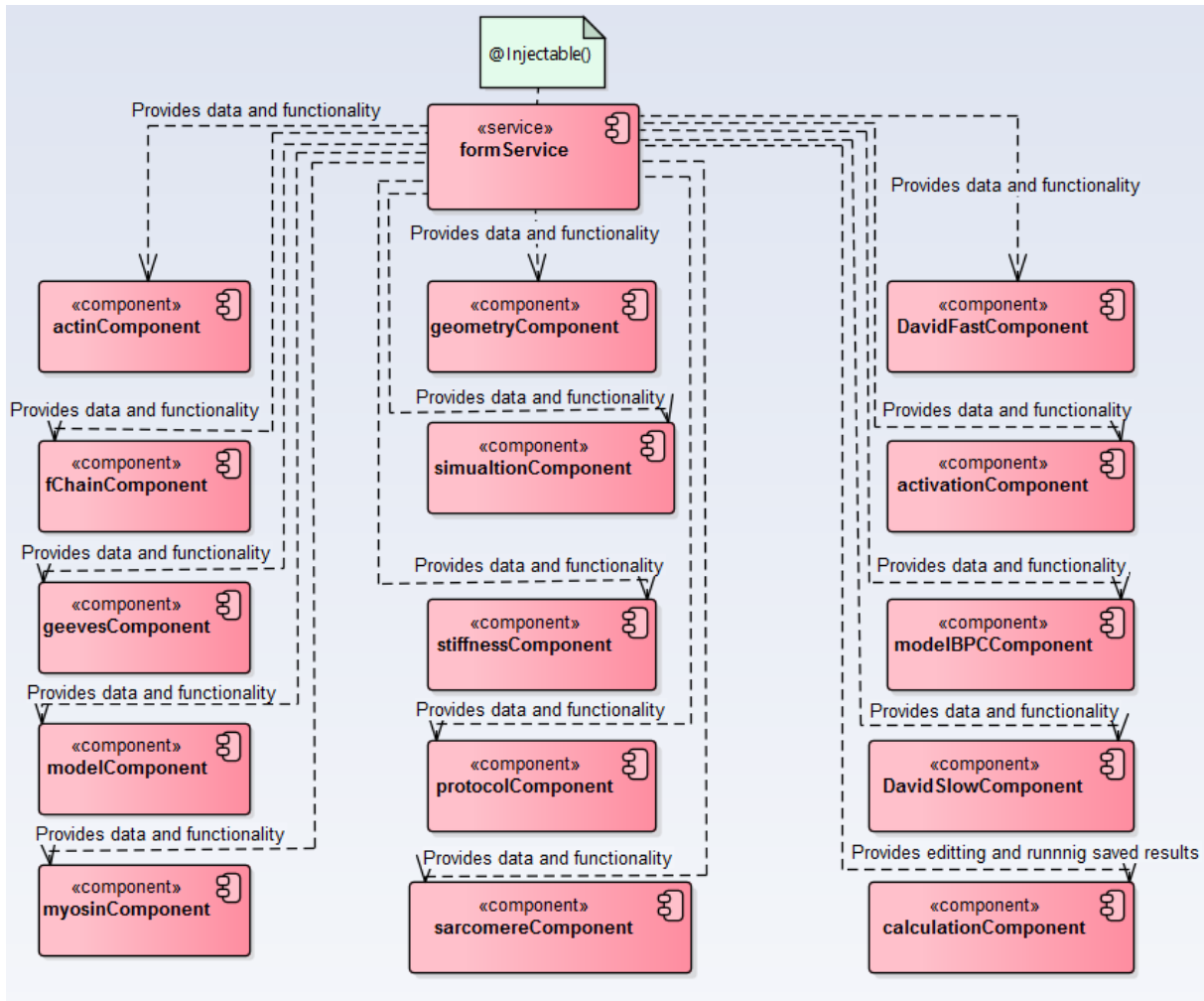
6.2. Zavisnosti između modula klijentskog dela



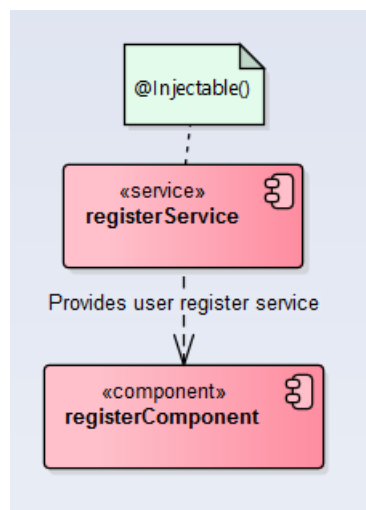
Slika 28 - Dijagram komponenti: authenticationService



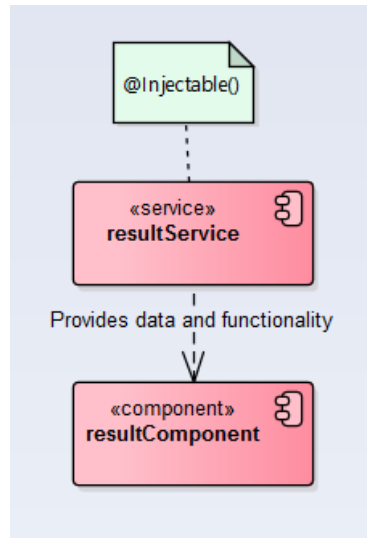
Slika 29 - Dijagram komponenti: calculationService



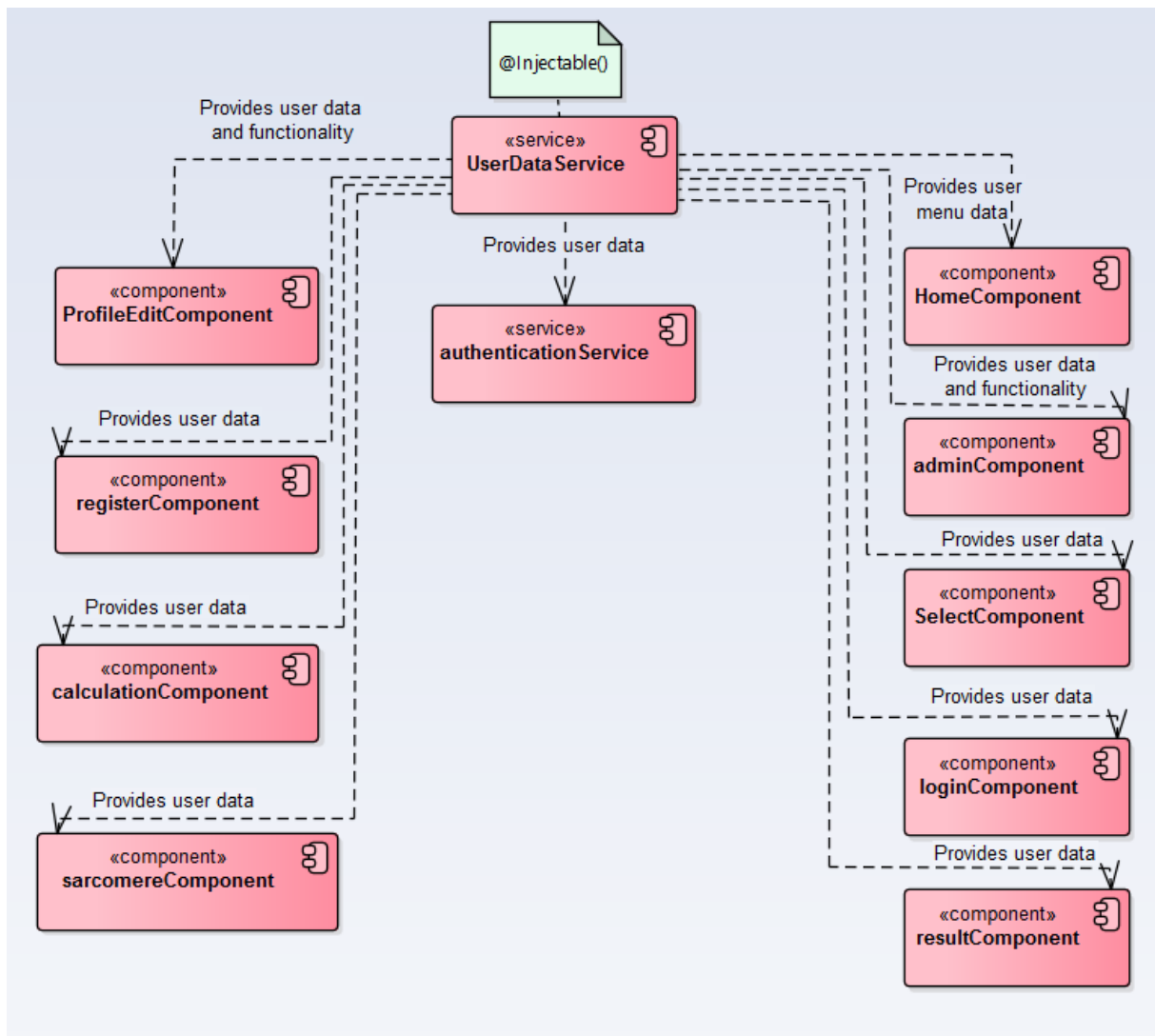
Slika 30 - Dijagram komponenti: formService



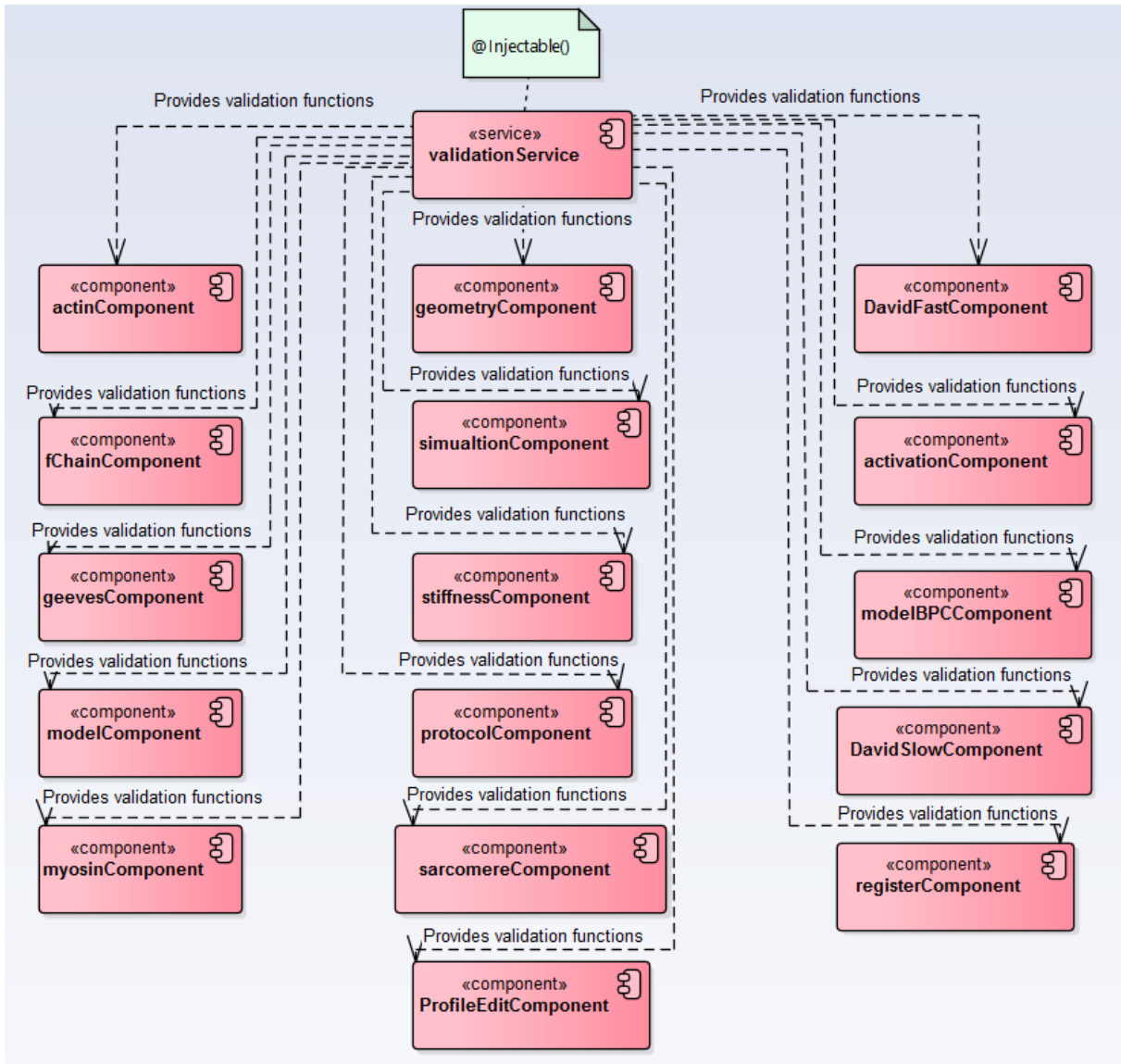
Slika 31 - Dijagram komponenti: registerService



Slika 32 - Dijagram komponenti: resultService

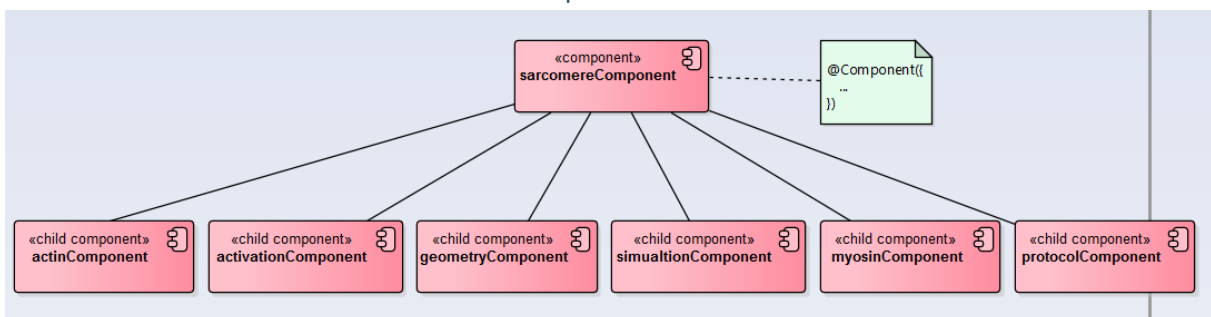


Slika 33 - Dijagram komponenti: UserDataService

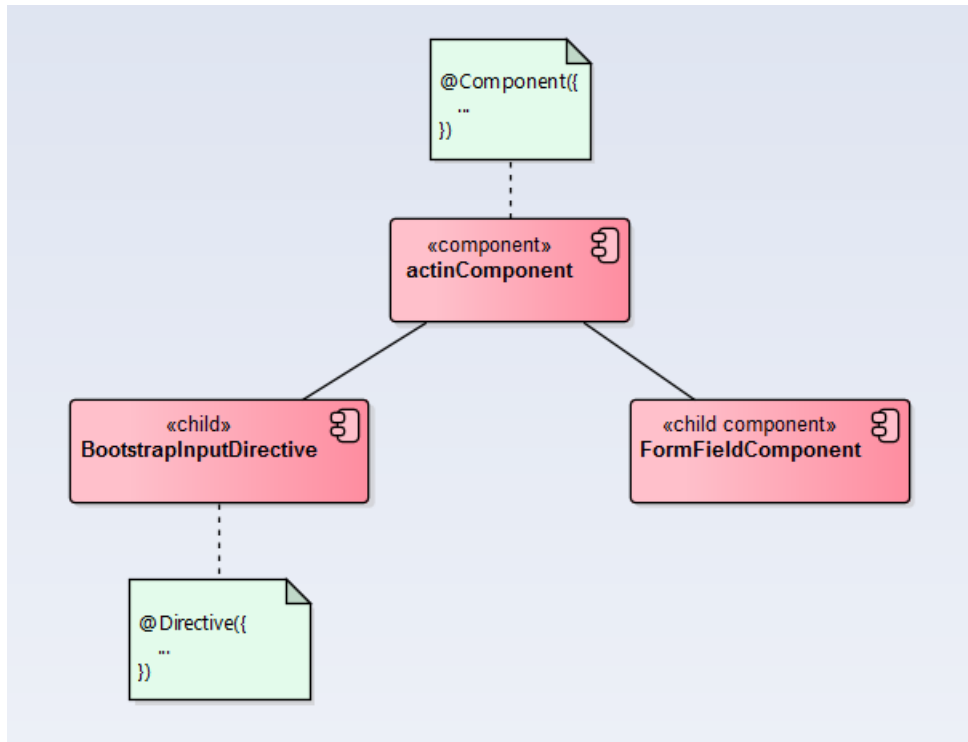


Slika 34 - Dijagram komponenti: validationService

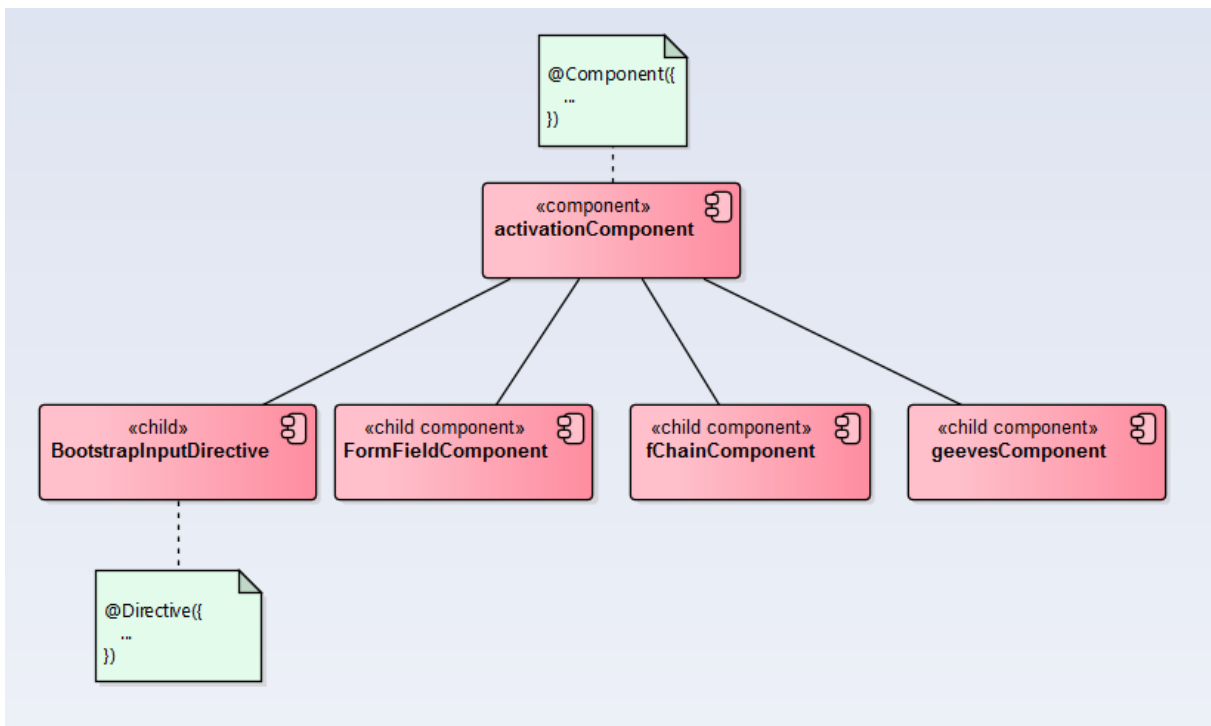
6.2.2. Zavisnosti između komponenti



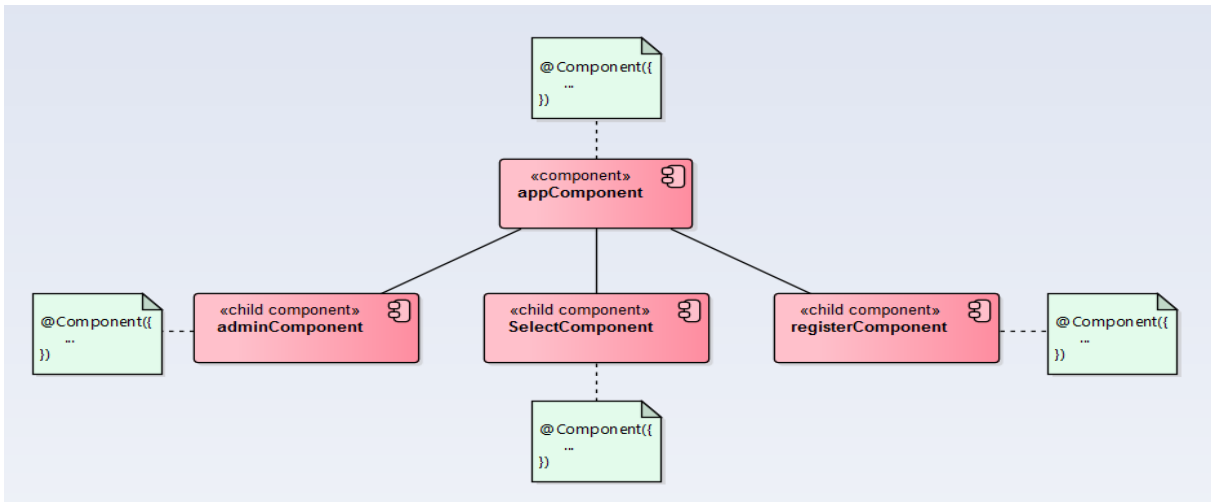
Slika 35 - Dijagram komponenti: sarcomereComponent



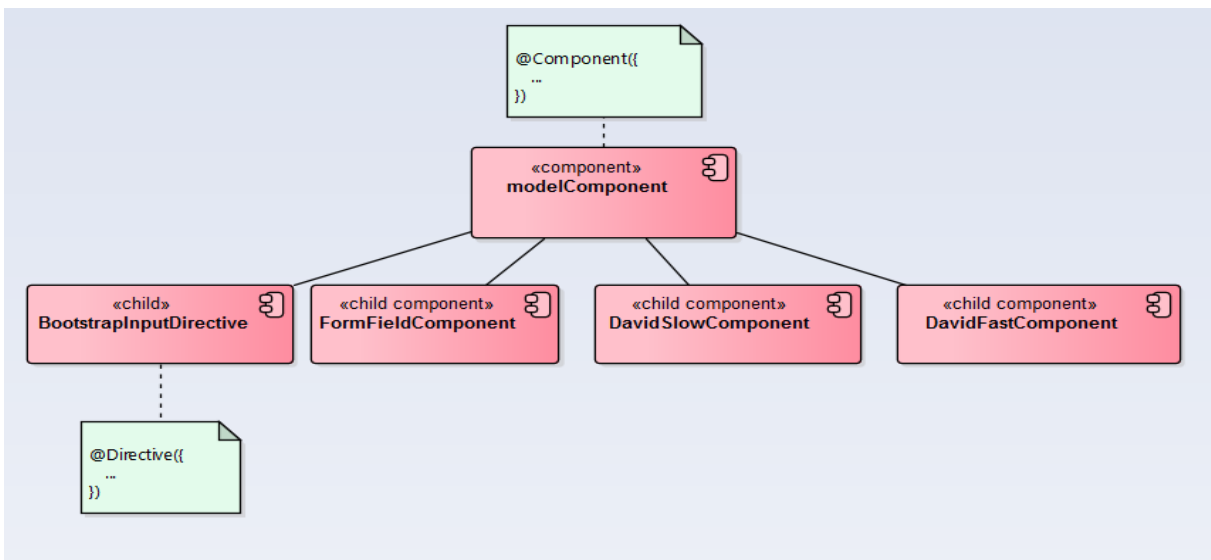
Slika 36 - Dijagram komponenti: actinComponent



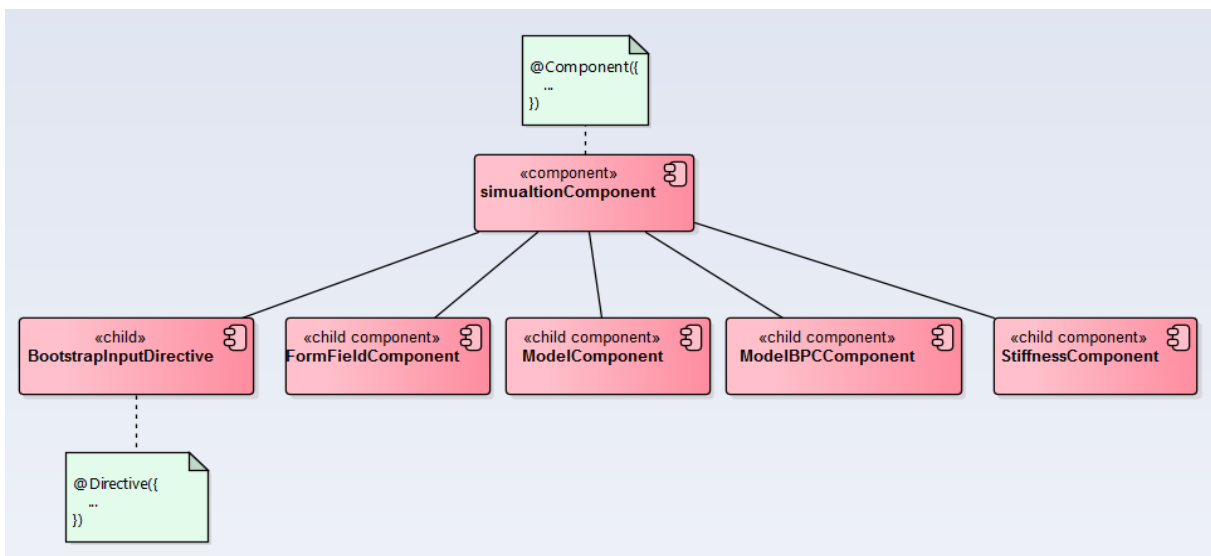
Slika 37 - Dijagram komponenti: activationComponent



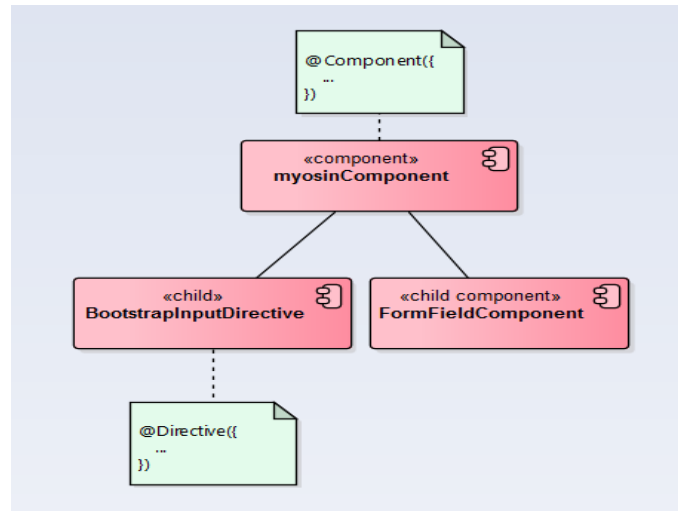
Slika 38 - Dijagram komponenti: appComponent



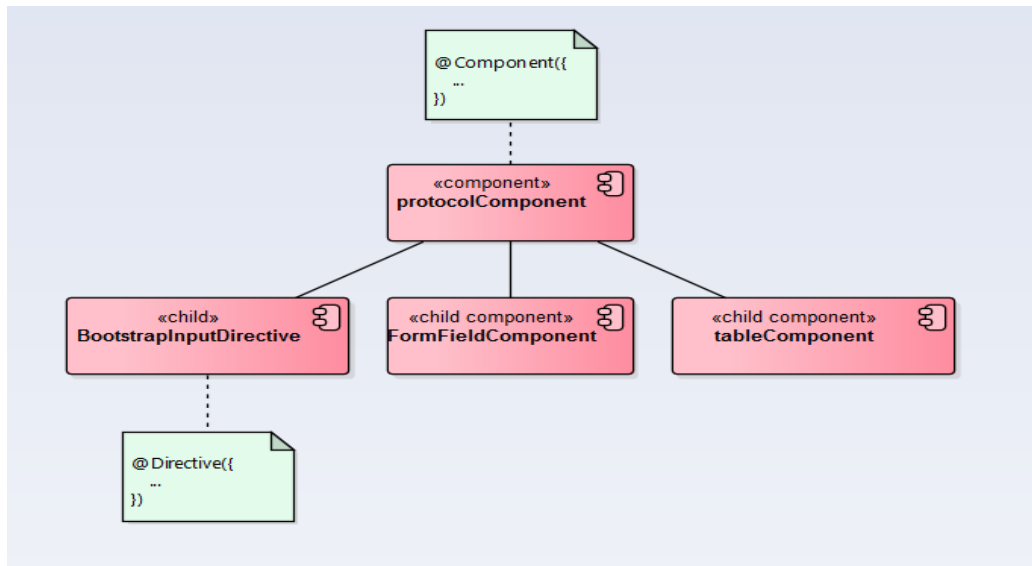
Slika 39 - Dijagram komponenti: modelComponent



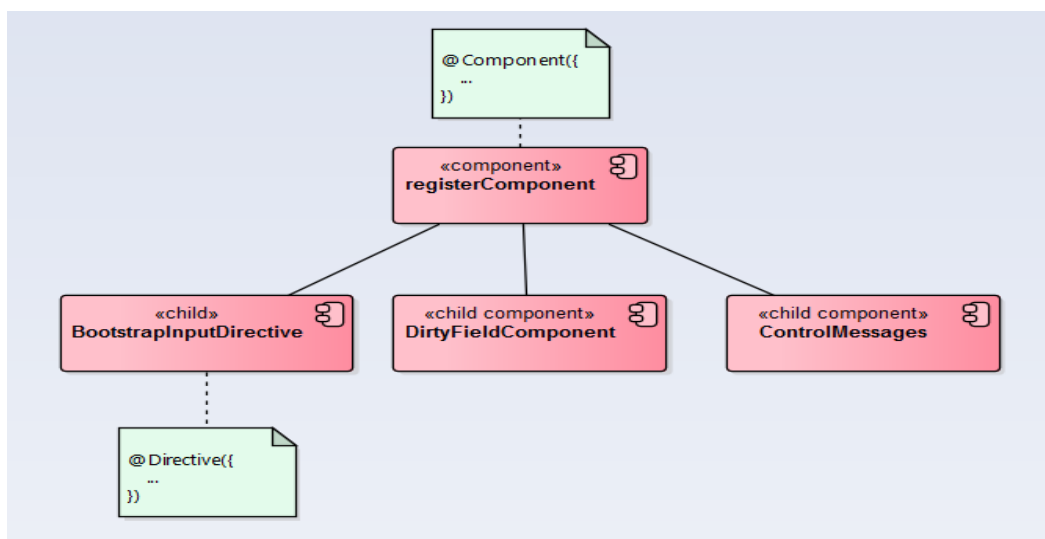
Slika 40 - Dijagram komponenti: simulationComponent



Slika 41 - Dijagram komponenti: myosinComponent



Slika 42 - Dijagram komponenti: protocolComponent



Slika 43 - Dijagram komponenti: registerComponent

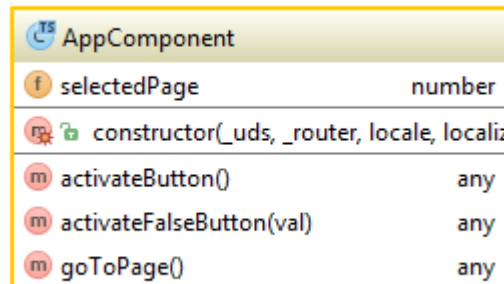
6.3. Dijagrami klasa

Početni paket, u kome se nalaze svi ostali paketi sistema jeste „app“ paket. U okviru njega nalazi se „AppComponent“, polazna komponenta sistema, i sledeći paketi:

- „Package Component“ – sadrži klase koje su zadužene za prikaz podataka u pretraživaču
- „Package Service“ – sadrži servise preko kojih se obavlja komunikacija sa serverskim delom aplikacije, http zahtevi i odgovori na zahteve se šalju putem klasa iz ovog paketa.
- „Package Directives“ – sadrži klase koje utiču na izgled interfejsa.
- „Package Html“ – u okviru ovog paketa se mogu naći html fajlovi koji se prikazuju u pretraživaču.
- „Package service“ – u okviru ovog paketa nalaze se klase neophodne za lokalizaciju.
- „Package Class“ – sadrži klase koje čuvaju podatke o ulaznim parametrima MUSICO softvera.

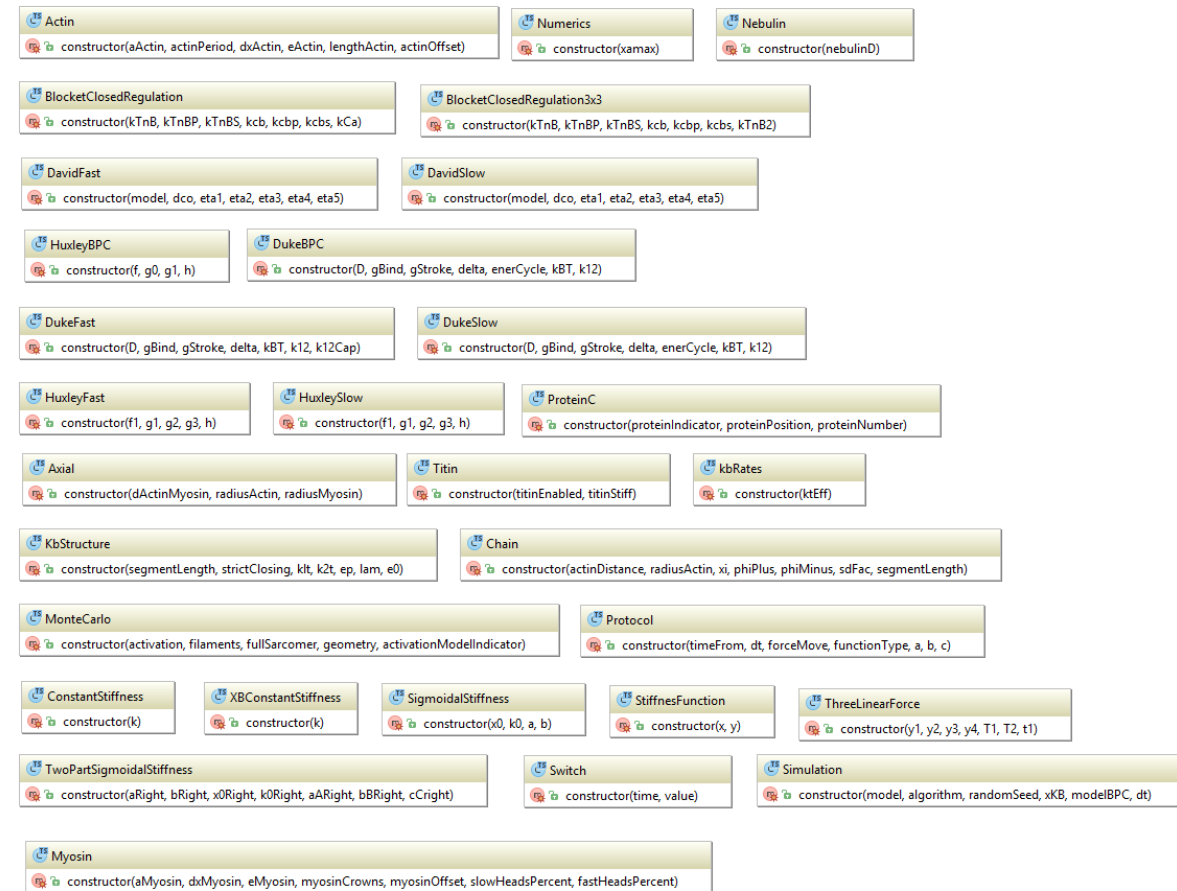


Slika 44 - Dijagram klasa: struktura foldera



Slika 45 - Dijagram klasa: AppComponent

Podaci o ulaznim parametrima se, kao što je već navedeno, mogu naći u paketu „Class“.



Powered by yFiles

Slika 46 - Dijagram klasa: ulazni parametri

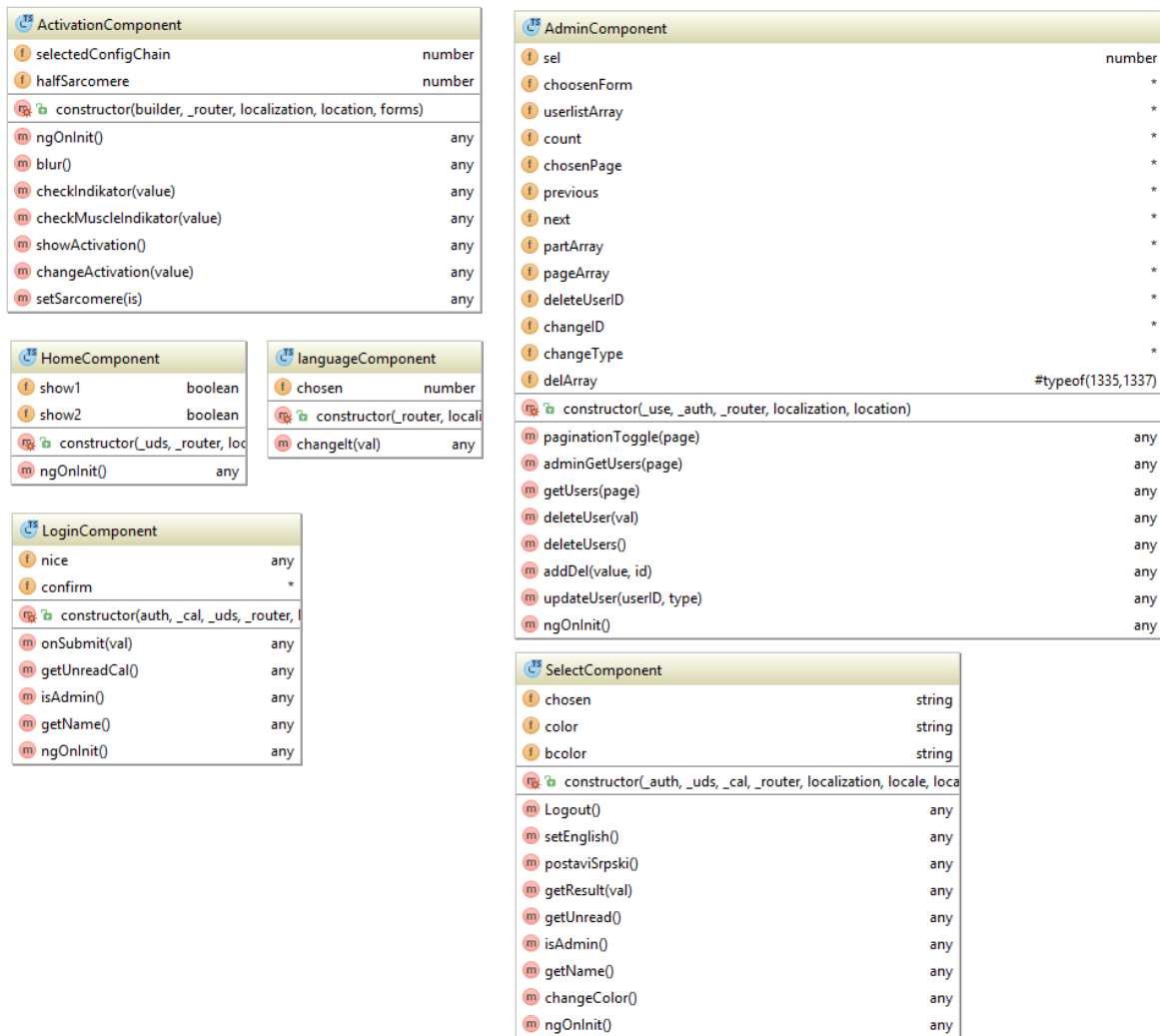
Prikaz ulaznih parametara MUSICO softvera se na neki način moraju prikazati korisniku. Tu ulogu će obavljati paket „Components“ uz pomoć odgovarajućih html datoteka.

The image displays a collection of class diagrams for various components in the MUSICO software. Each diagram shows the class name, its attributes (with types), and its methods. The components shown include:

- DavidFastComponent**: Attributes: partArray (#typeof(1002,1026)), chosenPage (number), previous (number), next (number), pageArray (*), count (number). Methods: constructor, paginationToggle, ngOnInit, discardChangesDukeFast, saveChangesDukeFast.
- DavidSlowComponent**: Attributes: partArray (#typeof(1004,1028)), chosenPage (number), previous (number), next (number), pageArray (*), count (number). Methods: constructor, paginationToggle, ngOnInit, discardChangesDukeSlow, saveChangesDukeSlow.
- SarcomereComponent**: Attributes: chosenForm (*), nice (*), actinTrue (boolean), myosinTrue (boolean), ComTitle (number), protocolSelected (boolean), confirmed (boolean), state (number), titleStatus (boolean). Methods: constructor, change, ngOnInit, makeActin, makeMyosin, makeSimulation, makeGeometry, makeAktivation, makeProtocol, checkTitle, ispis, submitAllForms, redirectFunction.
- ActivationComponent**: Attributes: selectedConfigChain (number), halfSarcomere (number). Methods: constructor, ngOnInit, blur, checkIndikator, checkMuscleIndikator, showActivation, changeActivation, setSarcomere.
- FChainComponent**: Attributes: selected (number), value (number). Methods: constructor, ShowCA, ShowKB, paginationToggle, makeOld, saveChainCa, discardChainCa, saveChainKb, discardChainKb, saveMgOld, discardMgOld.
- ModelBPCCComponent**: Methods: constructor.
- GeometryComponent**: Attributes: selectedConfig (number). Methods: blur, ngOnInit, disableProteinC, disableTitin, ngAfterViewInit, showBPCCModel, changeBPCCModelType, drawFullSarcomere, removeBPCC, discardChangesDukeBPCC, saveChangesDukeBPCC, discardChangesHuxleyBPCC, saveChangesHuxleyBPCC, changeDeficiency.
- GeesesComponent**: Attributes: selected (number), partArray (#typeof(1024,1033)), chosenPage (number), previous (number), next (number), pageArray (*), count (number). Methods: constructor, ShowCA, ShowKB, paginationToggle, ngOnInit, saveGeesesCa, discardGeesesCa, saveGeesesKb, discardGeesesKb.
- ProtocolComponent**: Attributes: chosenProtocol (string), confirmedCa (boolean), confirmedP (boolean), steps (boolean), uploadError (boolean), doesCaExists (boolean), doesPExists (boolean), count (number), chosenPage (number), pageArray (*), next (*), previous (*), partArray (*), falsePlace (boolean), caFalsePlace (boolean). Methods: constructor, ngOnInit, blur, changeFunction, changeFunctionCa, insertDiscretParameter, insertDiscretParameterCa, insertSwitchP, insertSwitchCa, changeProtocol, insertElement, insertElementCa, isThere, isCaThere, checkFloat, checkXlnFile, fileChangeEvent, getDatFile, paginationToggle, addElement, addCaElement.
- SimulationComponent**: Attributes: value (string), isHumberExists (boolean). Methods: constructor, blur, ngOnInit, showStiffness, changeStiffnessModel, changeFormOption, changeBPCCModelType, showModel, insertIntoSteps, removeSteps, dontSave.
- ModelComponent**: Attributes: selected (string). Methods: constructor, setSlow, setFast, discardChangesDukeFast, saveChangesDukeFast, discardChangesDukeSlow, saveChangesDukeSlow, discardChangesHuxleyFast, saveChangesHuxleyFast, discardChangesHuxleySlow, saveChangesHuxleySlow.

Slika 47 - Dijagram klasa: ulazne komponente

U paketu „Components“ će se sem komponente za prikaz ulaznih parametara nalaziti i komponente zadužene za prikaz i ostalih strana aplikacije.

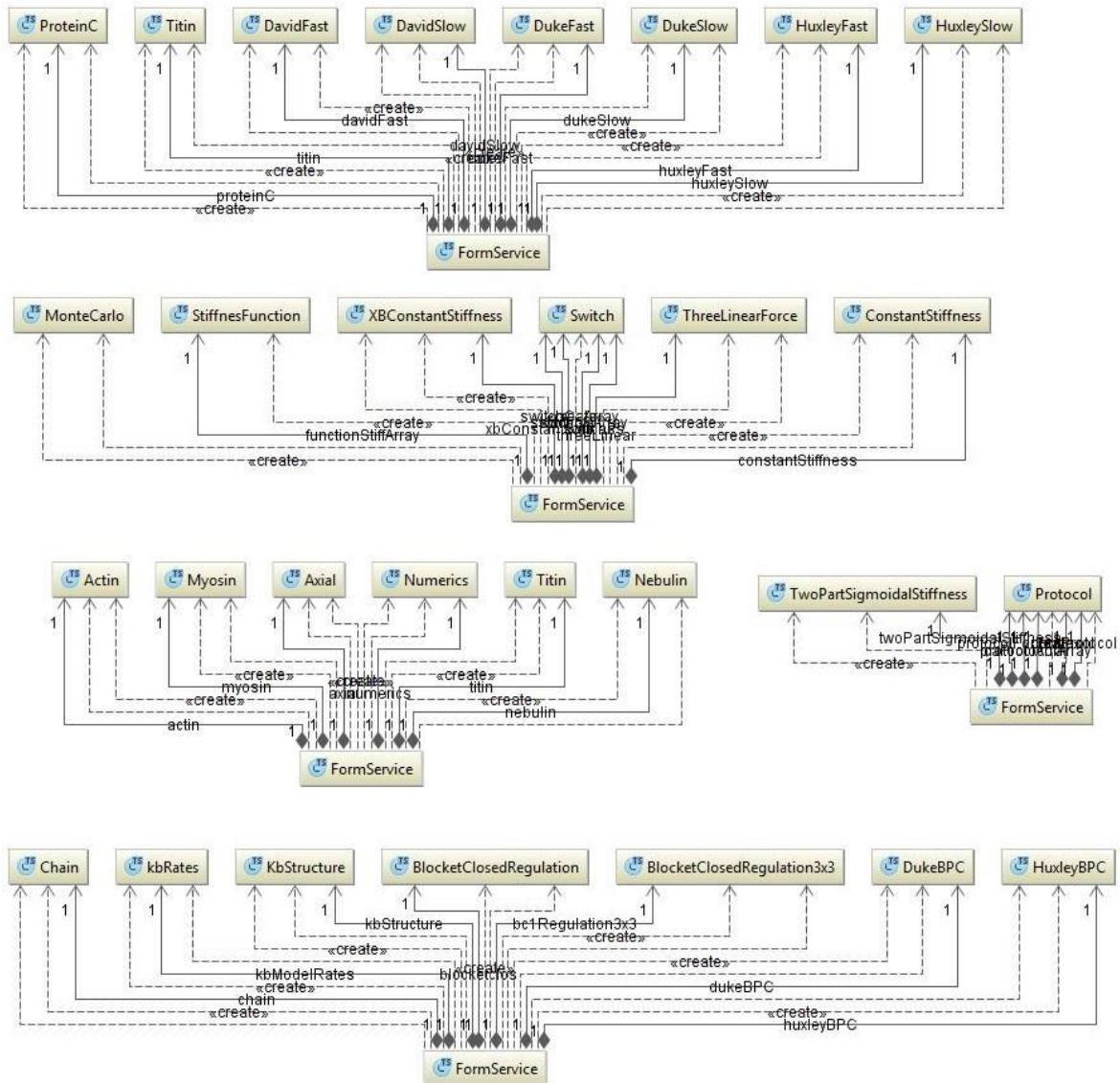


Slika 48 - Dijagram klasa: ostale stranice aplikacije

Kako je osnovni cilj ove aplikacije pokretanje MUSICO proračuna mora se na neki način uspostaviti komunikacija između korisničkog interfejsa i podataka na serveru, na osnovu kojih će MUSICO proračun biti pokrenut. Klase koje će biti zadužene za komunikaciju između klijentskog i serverskog dela aplikacije nalaze se upravo u ovom paketu. Klasa koja brine o slanju ulaznih parametara koje je korisnik popunio jeste klasa „Form Service“.

FormService					
edit	actin	dukeSlowForm	geevesForm	niceStiff	checkTitle()
editPage	axial	dukeBPC	kbModelRates	stepsyNice	ngOnInit()
editIndex	actinForm	dukeBPCForm	kbatesForm	f	handleError(error)
oldProtocol	protocolForEdit	huxleyFast	protocol	StepsNiceForm	createProtocol()
editCa	myosin	huxleyFastForm	protocolForm	_url	createCaProtocol()
editCaPage	myosinForm	huxleyBPC	selectedCaForm	_urlForEdit	paginationToggle(page, arrayType)
editIndexCa	numerics	huxleyBPCForm	caProtocol	_urlForSaved	removeTableRow(index, typeArray)
oldProtocolCa	nebulin	huxleySlow	CaPprotocolForm	disabledLinearP	editProtocol(index)
savingObject	titin	huxleySlowForm	selectedProtocol	disabledLinearCa	editProtocolCa(index)
savingObjectSlow	proteinC	constantStiffness	protocolArray	constructor(http, _auth)	saveProtocolCa()
selectedConfig	geometryForm	constantStiffnessForm	caProtocolArray	isNice(index, val)	saveProtocol()
title	simulation	xbConstantStiffness	selectedMuscleType	isTableNice(index, val)	cancelEdit(type)
comment	simulationForm	xbConstantStiffnessForm	selectedActivationType	isNiceStiff(index, val)	insertSwitchP()
previous	monteCarlo	xbSigmoidalStiffness	switchCa	isCaTableNice(index, val)	insertSwitchCa()
next	blocketclos	xbSigmoidalStiffnessForm	switchP	isStepsNice(index, val)	addStiffnes(x, y)
CaPrevious	CaActivation3x2	threeLinear	switchPArray	isSwitchNice(index, val)	editCaProtocolParameters(i, type)
CaNext	chain	threeLinearForm	switchCaArray	isCaSwitchNice(index, val)	setSelectedCaRow(i)
pageArray	isSarcomereSelected	twoPartSigmoidalStiffness	switchPForm	changeForm(formNumber)	setSelectedRow(i)
resultArray	davidFast	twoPartSigmoidalStiffnessForm	switchCaForm	submitForms(type)	
partArray	davidFastForm	stiffnesFunction	niceSwitch	editResults(id)	
chosenPage	davidSlow	functionStiffArray	niceCaSwitch	changeInputParams(editedParams)	
caChosenPage	davidSlowForm	kbStructure	titinForms	setDefaultInputParams()	
selectedHeads	dukeFast	KbModelForm	writingDistSteps	setDefaultValidationSigns()	
count	dukeFastForm	Activation	selected	fromJSON(json, object)	
	dukeSlow	selectedCaRow	niceSubmit	runSavedCalculation(id)	

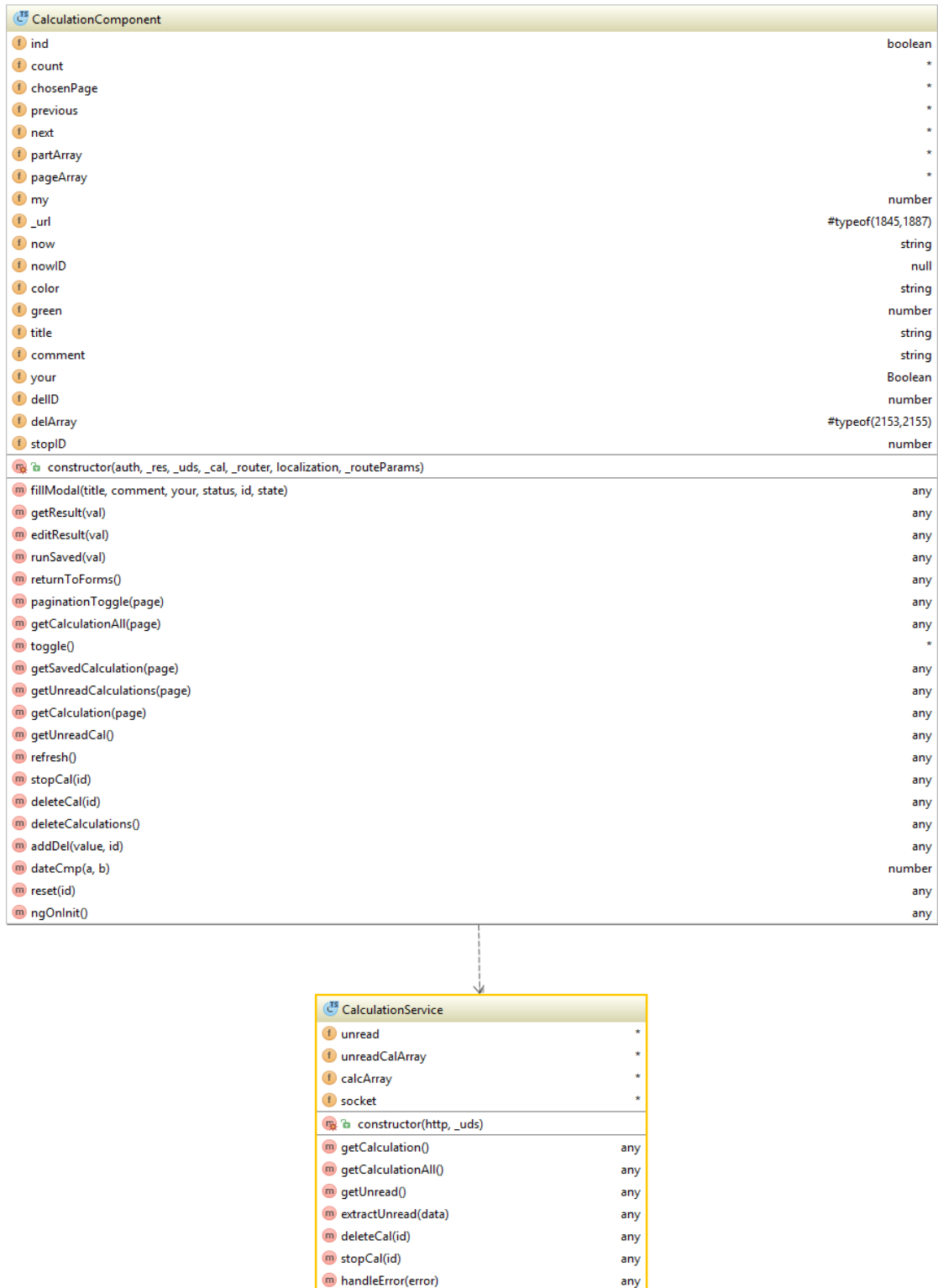
Slika 49 - Dijagram klasa: FormsService



Slika 50 - Dijagram klasa: FormService

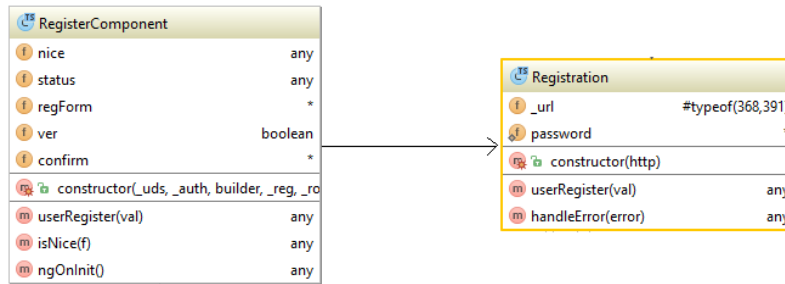
Sem „Form Service“ klase u okviru ovog paketa nalaze se i komponente aplikacije koje sa serverom obavljaju komunikaciju nevezanu za ulazne parametre, kao što su:

- Prikaz svih proračuna,



Slika 51 - Dijagram klasa: prikaz svih proračuna

- Registracija,



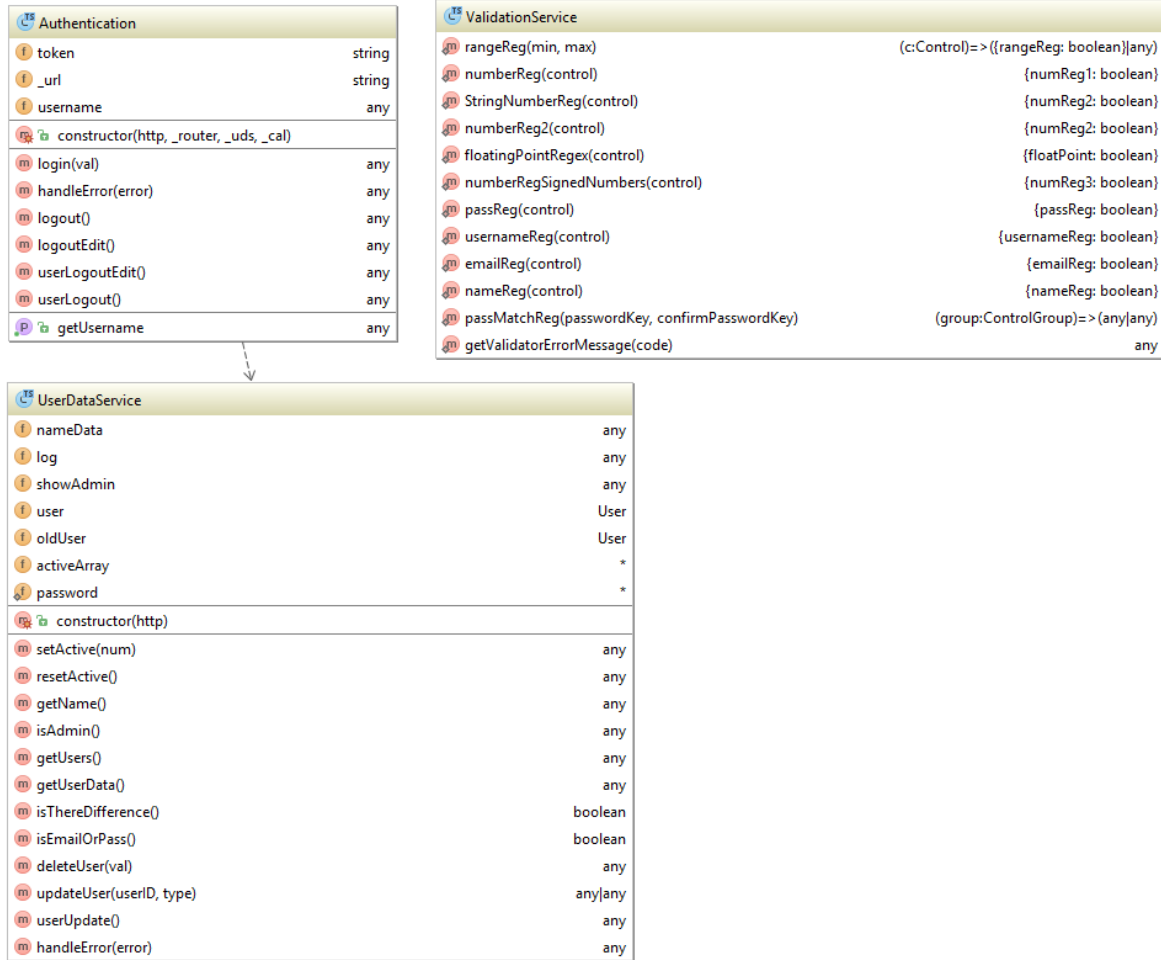
Slika 52 - Dijagram klasa: registracija

- Prikaz rezultata željenog proračuna,



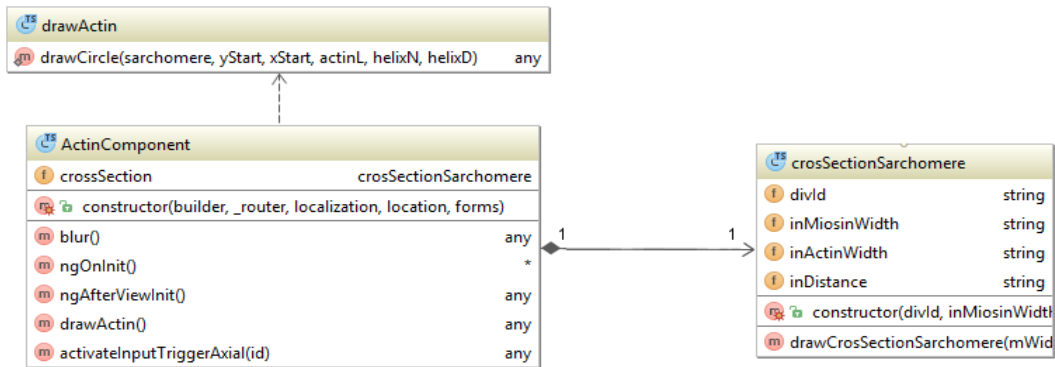
Slika 53 - Dijagram klasa: prikazivanje rezultata proračuna

- Servisi zaduženi za validaciju podataka i autentifikaciju.

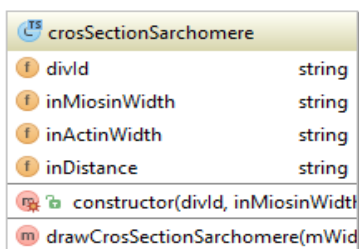
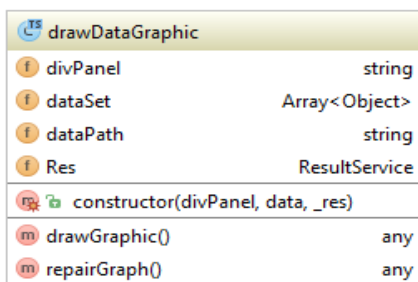
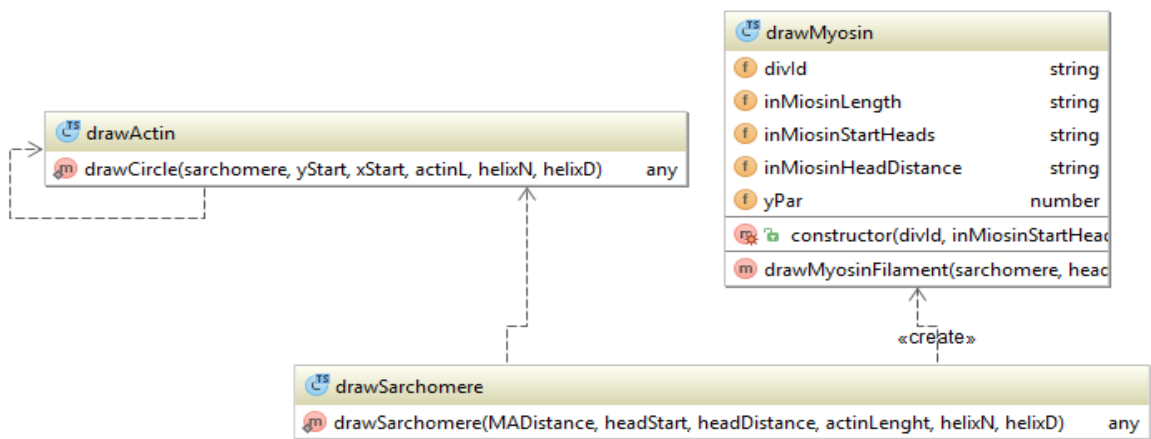


Slika 54 - Dijagram klasa: validacija podataka i autentifikacija

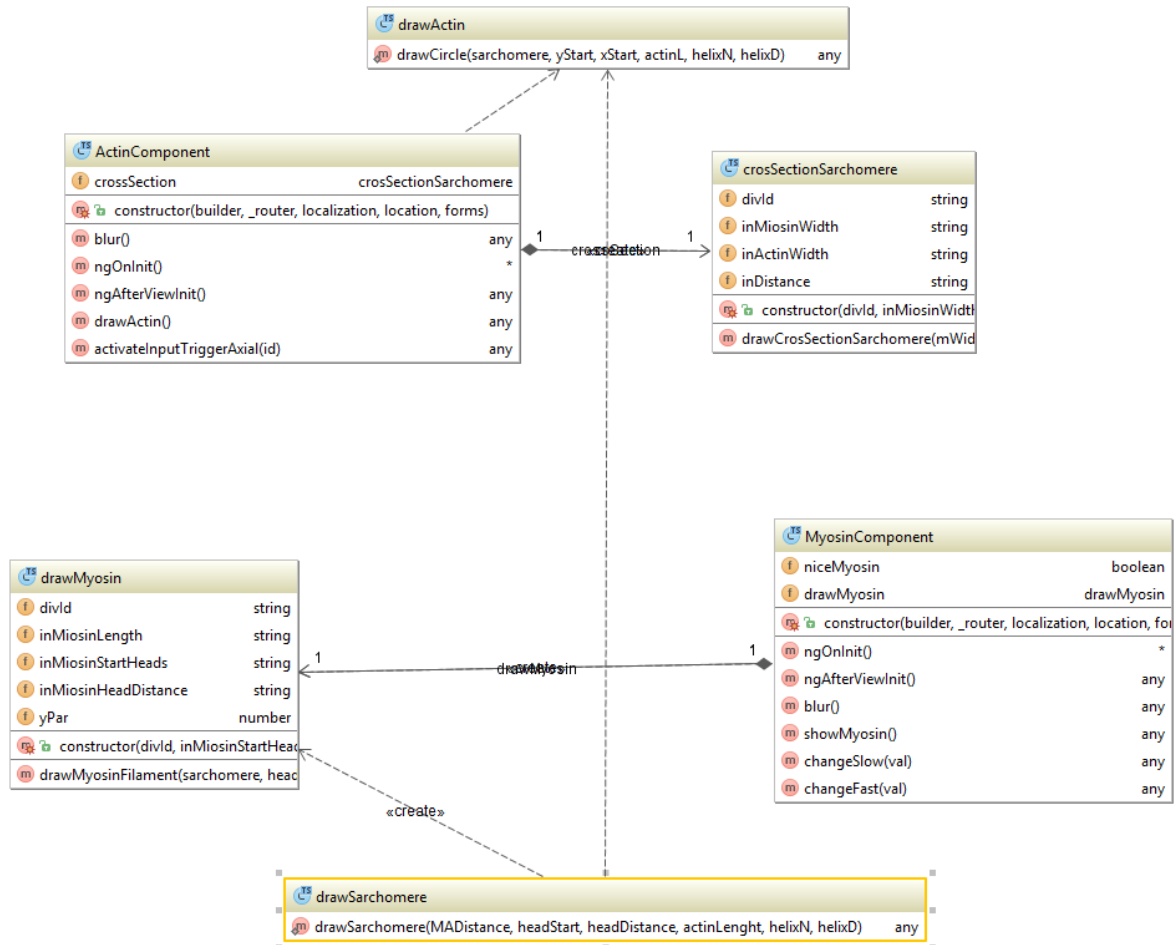
Vizuelizacija podataka obavljaće se preko D3 biblioteka a uz pomoć klasa koje će se nalaziti u „Graph“ paketu unutar „Class“ paketa.



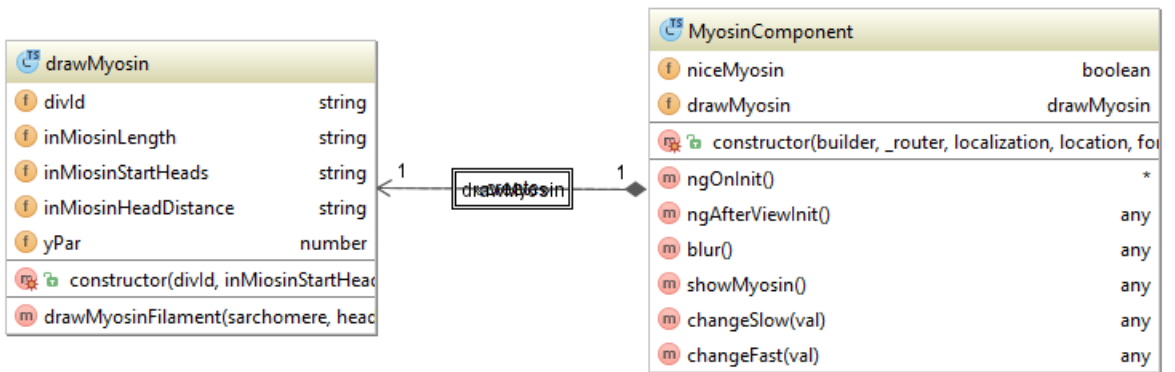
Slika 55 - Dijagram klasa: crtanje aktina



Slika 56 - Dijagram klasa: d3

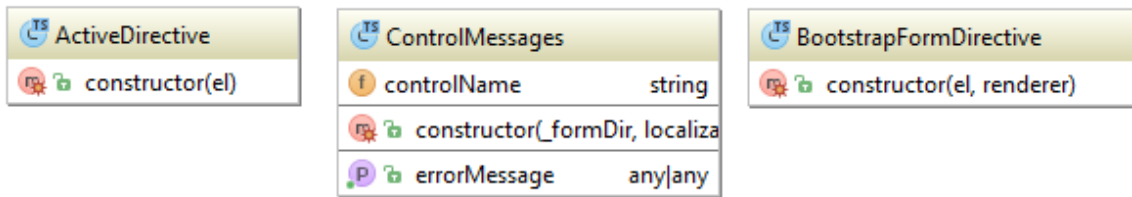


Slika 57 - Dijagram klasa: crtanje sarkomere



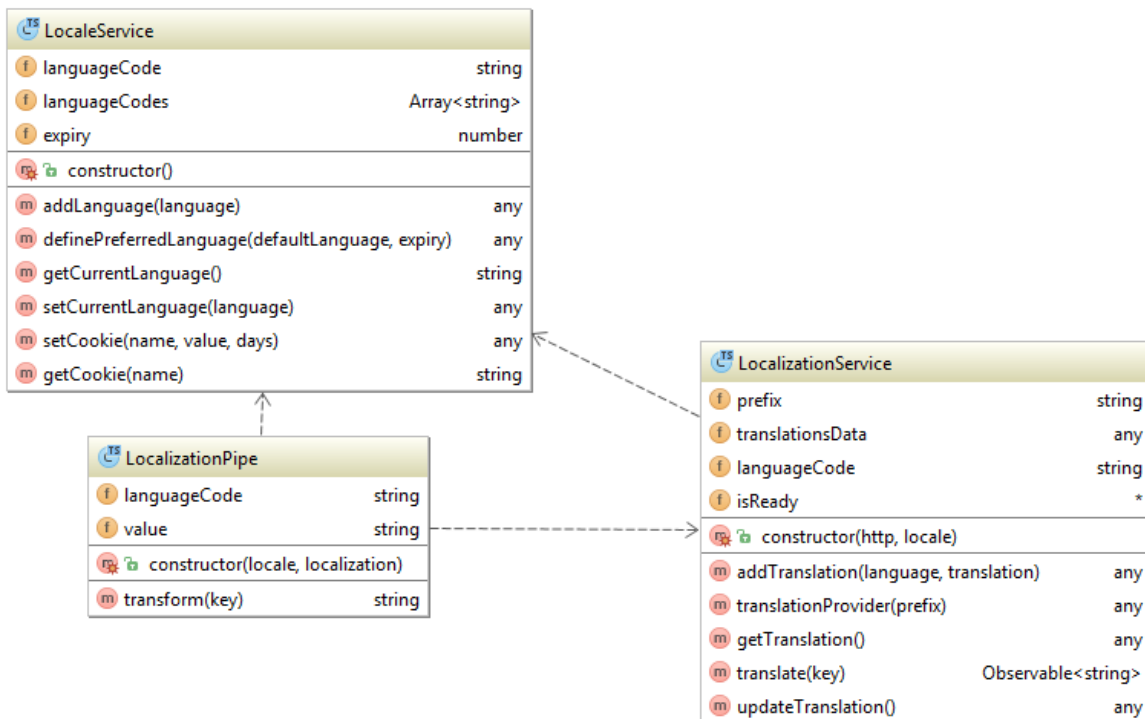
Slika 58 - Dijagram klasa: crtanje miozina

U paketu „Directives“ se, kao što je vec navedeno, nalaze klase koje utiču na promene u okviru korisničkog interfejsa.



Slika 59 - Dijagram klasa: direktive

U paketu „Service“ i „Pipes“ nalaze se klase koje reaguju na želje korisnika za promenom jezika aplikacije i vode brigu o lokalizaciji.



Slika 60 - Dijagram klasa: lokalizacija

6.4. Eksperiment REST API

Experiment model je glavni model i proširuje [PersistedModel](#) model Strongloop-a. Zadužen je za upravljanje proračunima. Model omogućava: čuvanje, čitanje i brisanje proračuna, pokretanje proračuna i njihovo prekidanje. Takođe, uz pomoć [resultChunks](#) modela pruža: upisivanje i čitanje rezultata završenih proračuna. Kako bi se moglo pristupiti bilo kojoj od ovih funkcionalnosti, svakoj metodi se mora proslediti i access token korisnika, koji korisnik dobija pri prijavljivanju na server.

Primer linka sa access token-om:

```
api/experiments/<ime metode>?access_token=<access token>
```

Ukoliko se access token ne prosledi u link, dobija se sledeća greška:

```
{
  error: {
    name: "Error",
    status: 401,
    message: "Authorization Required",
    statusCode: 401,
    code: "AUTHORIZATION_REQUIRED",
    stack: "Error: Authorization Required\n    at...  }
}
```

6.5. Čuvanje proračuna

Čuva podatke o proračunu. U slučaju da se prosledi parametar state sa vrednošću 3, proračun će samo biti sačuvan. Ukoliko se prosledi vrednost 0 za ovaj parametar, proračun će biti sačuvan, a zatim i prosleđen cluster serveru na izvršavanje. Parametar paramers je JSON objekat koji nosi sve ulazne parametre za pokretanje proračuna.

6.5.1. URL i metod zahteva:

```
POST api/experiments/runexperiment
```

6.5.2. Parametri zahteva:

```
{
  data: {
    userID: '',
    title: 'proračun',
    description: 'Komentar proračuna',
    softwareVersion: '1.0.0',
    date: '16/05/2016 16:11:46',
    state: 0,
    parameters: { [ [Object] ] }
  }
}
```

Ovakav zahtev će sačuvati proračun i proslediti ga cluster serveru na izračunavanje. Da se proračun samo sačuva u bazi, u zahtevu za parametar state treba proslediti vrednost 3 (state: 3).

6.5.3. Uspešan zahtev:

Statusni kod odgovora: 200

Sadržaj odgovora: { data: "Experiment saved" }

6.6. Brisanje proračuna

Briše korisnikov proračun iz baze. U zahtevu se prosleđuje ID proračuna koji treba obrisati.

6.6.1. URL i metod zahteva:

POST `api/experiments/destroyexperiment`

6.6.2. Parametri zahteva:

```
{ data: "ID Proračuna za brisanje" }
```

6.6.3. Odgovori na zahtev:

Statusni kod odgovora: 200

Ukoliko je postoji proračun sa prosleđenim ID-om i ako je obrisani

Sadržaj odgovora: { data: True }

Ukoliko ne postoji proračun sa prosleđenim ID-om

Sadržaj odgovora: { data: False }

6.7. Čitanje svih proračuna

Vraća sve proračune iz baze. U zahtevu nije potrebno proslediti ni jedan parametar, sem access token, koji se mora da se prosledi zbog autentifikacije korisnika.

6.7.1. URL i metod zahteva:

POST `api/experiments/getallexperiments`

6.7.2. Odgovor na zahtev

Sadržaj odgovora je niz proračuna iz baze

Statusni kod odgovora: 200

Sadržaj odgovora: { data: [Svi proračuni iz baze] }

6.8. Čitanje korisnikovih proračuna

Vraća sve proračune korisnika čiji je access token prosleđen u zahtevu.

6.8.1. URL i metod zahteva:

POST `api/experiments/getusersexperiments`

6.8.2. Parametri zahteva:

Prosleđeni parametar data može da bude bilo kakav string.

```
{ data: "getUserExperiments" }
```

6.8.3. Odgovor na zahtev:

Sadržaj odgovora je niz proračuna iz baze.

Statusni kod odgovora: 200

Sadržaj odgovora: { data: [Proračuni korisnika] }

6.9. Čitanje nepročitanih proračuna

Vraća korisnikove proračune, čiji rezultat nije pročitani (vrednost parametra state je 1).

6.9.1. URL i metoda zahteva:

POST api/experiments/getunread

6.9.2. Parametri zahteva:

Prosleđeni parametar data može da bude bilo kakav string.

```
{ data: "getNotReadExperiments" }
```

6.9.3. Odgovor na zahtev:

Sadržaj odgovora je niz proračuna iz baze.

Statusni kod odgovora: 200

Sadržaj odgovora: { data: [Proračuni koje korisnik nije pročitao] }

6.10. Pokretanje sačuvanog proračuna

Pokreće proračun koji je korisnik sačuvao za kasnije pokretanje (pri čuvanju proračuna, kao vrednost za parametar state, presleđeno je 3). Za parametar, experimentID, zahteva, prosleđuje se ID proračuna koji želimo da pokrenemo.

6.10.1. URL i metod zahteva:

GET api/experiments/runsavedexperiment

6.10.2. Parametri zahteva:

experimentID:ID_Experimenta

6.10.3. Odgovori na zahtev

Statusni kod odgovora: 200

Sadržaj zahteva: { data: "Experiment Started"}

Ako je proračun već završio sa radom

```
{ data: "Experiment already finished" }
```

Ako je prosleđen pogrešan ID proračuna

```
{ data: "Wrong Experiment ID" }
```

6.11. Prekid pokrenutog proračuna

Prekida izvršavanje pokrenutog proračuna i menja state na sačuvan proračun (state: 3).

6.11.1. URL zahteva:

POST api/experiments/cancelExperiment

6.11.2. Parametri zahteva:

Za parametar data zahteva se prosleđuje ID proračuna za zaustavljanje.

```
{ data: "experimentID" }
```

6.11.3. Odgovori na zahtev:

Statusni kod odgovora: 200

Sadržaj odgovora:

Ako proračun postoji i korisnik koji prekida proračun je njegov vlasnik:

```
{ data: True }
```

Ako nisu ispunjeni ti uslovi, kao odgovor se dobija:

```
{ data: False }
```

Ako ne postoji proračun sa prosleđenim ID-em, kao odgovor se dobija:

```
{ data: null }
```

6.12. Proveravanje postojanja naslova proračuna

Proverava da li korisnik već ima proračun sa određenim naslovom. U zahtevu se, za parametar data, prosleđuje string za koji želimo da proverimo da li već postoji kao naslov nekog korisnikovog proračuna.

6.12.1. URL i metod zahteva:

POST api/experiments/checkTitle

6.12.2. Parametri zahteva:

```
{ data: "Naslov proračuna" }
```

6.12.3. Odgovori na zahtev:

Statusni kod odgovora: 200

Sadržaj odgovora:

Ukoliko postoji proračun sa zadatim naslovom:

```
{ data: True }
```

Ukoliko ne postoji proračun sa zadatim naslovom:

```
{ data: False }
```

Ukoliko naslov nije prosleđen

```
{ data: null }
```

6.13. Uzimanje komada rezultata proračuna

Iz baze vraća komad rezultata proračuna. U zahtevu se, za data parametar, prosleđuje ID proračuna i redni broj komada koji treba uzeti iz baze. Ovaj metod koristi metodu getChunk resultChunks modela, kako bi iz baze mogao da izvuče podatke.

6.13.1. URL i metod zahteva:

POST api/experiments/getchunk

6.13.2. Parametri zahteva:

```
{  
  data: {  
    experimentID: "570ac1bff8471c7030b8b64d",  
    chunkNumber: 1  
  }  
}
```

6.13.3. Odgovor na zahtev:

Statusni kod zahteva: 200

Sadržaj odgovora:

Ukoliko postoji komad sa zadatim parametrima:

```
{
  data: {
    experimentID:"570ac1bfff8471c7030b8b64d",
    chunkNumber:1,
    data:[{Deo rezultata proračuna}]
  }
}
Ukoliko ne postoji komad sa zadatim parametrima:
{ data: null }
```

6.5. ResultChunks API

ResultChunks model je pomoćni model koji takođe proširuje [PersistedModel](#) model Strongloop-a. Ovaj model nema remote metod za pristup, već se njegove funkcije mogu koristiti samo iz sistema, i namenjen je da sistemu doda mogućnost rada sa rezultatima proračuna.

Mogućnosti koje uvodi u sistem su: upis i brisanje rezultata nekog proračuna iz baze i čitanje određenog komada rezultata proračuna iz baze.

6.5.1. Upisivanje rezultata proračuna u bazu

Ova funkcija radi tako što rezultat proračuna prvo izdela na komade, a zatim ih upisuje u bazu. Komadi su uvek iste veličine. Komad je JSON objekat koji u sebi sadrži ID proračuna čiji su rezultati, redni broj tog komada i deo proračuna. Kada funkcija završi sa upisom podataka u bazu, korisniku čiji je proračun na email šalje poruku da je njegov proračun završen i da može da pogleda podatke.

6.5.1.1. Potpis funkcije:

```
insertChunks = function (data, experimentID)
```

Data parametar funkcije predstavlja rezultat proračuna, koji treba upisati u bazu, a experimentID je ID eksperimenta čiji je rezultat.

6.5.2. Brisanje rezultata proračuna iz baze

Ova funkcija briše rezultate proračuna iz baze.

6.5.2.1. Potpis funkcije:

```
deleteChunks = function(experimentID)
```

Parametar experimentID predstavlja ID proračuna čiji rezultat treba da se izbriše.

6.5.3. Čitanje komada rezultata iz baze

Funkcija iz baze uzima komad proračuna i preko callback funkcije, koju dobija od metode [getChunk](#) iz Experiment modela, vraća taj komad korisniku. Ukoliko komad ne postoji, vraća null. ExperimentID i chunkNumber su parametri za pretragu baze, a cb je callback funkcija koja vraća klijentu nađeni komad.

6.5.3.1. Potpis funkcije:

```
getChunk = function (experimentID, chunkNumber, cb)
```

6.6. WemusUser REST API

WemusUser model je model za upravljanje korisnicima aplikacije. Proširuje model [User](#) StrongLoop-a. Ovaj model omogućava: listanja svih korisnika, proveravanja da li je korisnik administrator, uzimanja podataka o korisniku, menjanja informacija o korisniku i njegove role, kao i brisanja korisnika. Registracija i prijava korisnika se uzima od Strongloop User modela. Kao i kod modela Experiment, pri pozivanju pomenutih metoda za upravljanje korisnicima, mora se proslediti i access token kao parametar zahteva. Ukoliko se access token ne prosledi, dobija se greška o autentifikaciji sa kodom 401.

6.6.1. Listanje svih korisnika:

Vraća niz registrovanih korisnika, sem korisnika čiji je access token prosleđen u pozivu. Sem access tokena nije potreban ni jedan drugi parametar za pozivanje.

6.6.1.1. *URL Zahteva:*
GET api/WemusUsers/getUsers

6.6.1.2. *Odgovor na zahtev:*
Statusni kod odgovora: 200

Sadržaj odgovora:

```
{
  data: [
    {
      username: "Korisnik",
      firstName: "Ime korisnika",
      lastName: "Prezime korisnika",
      email: "foo@bar.com",
      emailVerified: true,
      id: "ID Korisnika",
      type: "user"
    }
  ]
}
```

6.6.2. Uzimanje podataka o korisniku

Iz baze vraća sve podatke o korisniku, čiji je access token prosleđen. Pri pozivanju metode, potrebno je proslediti samo access token., ni jedan drugi parametar nije potreban

6.6.2.1. *URL i metod zahteva:*
GET api/WemusUsers/getUser

6.6.2.2. *Odgovor na zahtev:*
Statusni kod odgovora: 200

Sadržaj odgovora:

```
{
  data:{
    username: "Korisnik",
    firstName: "Ime korisnika",
    lastName: "Prezime korisnika",
    email: "foo@bar.com",
    emailVerified: true,
    id: "ID Korisnika",
    type: "user"
  }
}
```

```

    }
  }

```

6.6.3. Provera da li je korisnik administrator

Za korisnika čiji je access token prosleđen u zahtevu, proverava da li ima administratorska prava.

6.6.3.1. URL i metod zahteva:

GET api/WemusUsers/isAdmin

6.6.3.2. Odgovori na zahtev:

Ukoliko korisnik ima administratorska prava:

```
{ data: True }
```

Ukoliko nema administratorska prava:

```
{ data: False }
```

6.6.4. Menjanje informacija o korisniku

Omogućava menjanje ličnih informacija korisnika, kao što su ime, prezime, username, password i email. Kao parametar pri pozivanju metode, prosledjuje se JSON objekat sa navim podacima o korisniku. Kako bi korisnik mogao da

6.6.4.1. URL i metod zahteva:

POST api/WemusUsers/updateUserData

6.6.4.2. Odgovor na zahtev:

Kao odgovor na zahtev se dobja prazan JSON objekat

6.6.5. Menjanje role Korisnika

Korisniku menja privilegije u sistemu. Kao parametar u zahtevu se šalje JSON Objekat koji u sebi ima ID user kojem treba promeniti rolu, i rolu u koju se menja. Role koji mogu da se proslede su: admin I user

6.6.5.1. URL I metod zahteva:

POST api/WemusUsers/updateRole

6.6.5.2. Parametri zahteva:

Ako želimo da korisniku damo rolu administratora:

```

{
  data: {
    userId: "ID Korisnika",
    type: "admin"
  }
}

```

Ako želimo da korisniku damo rolu običnog korisnika:

```

{
  data: {
    userId: "ID Korisnika",
    type: "user"
  }
}

```

6.6.5.3. *Odgovor na zahtev:*

Kao odgovor na zahtev se dobija prazan JSON objekat.

6.6.6. Brisanje korisnika

Iz baze briše korisnika čiji je ID prosleđen u zahtevu.

6.6.6.1. *URL i metod zahteva:*

POST `api/WemusUsers/deleteUser`

6.6.6.2. *Parametar zahteva:*

```
{ data: "ID Korisnika" }
```

6.6.6.3. *Odgovor na zahtev:*

Kao odgovor na zahtev se dobija prazan JSON objekat.

6.7. RunCModel API

RunCModel model je pomoćni model koji proširuje [PersistedModel](#) model strongloop-a. Ovaj model nema remote metode, već su namenjene da se koriste samo iz sistema. Sistemu daje mogućnost pokretanja proračuna, slanje rezultata proračuna wemus serveru na obradu i upisivanje u bazu, kao i pakovanje celog proračuna u zip i slanje na wemus server, odakle može biti preuzet.

6.7.1. Pokretanje proračuna

Ova funkcija radi tako što prvo napravi ulazne fajlove sa parametrima proračuna, a zatim pokreće musico.

6.7.1.1. *Potpis funkcije:*

```
getData = function (callback, params)
```

Parametar callback je funkcija koja se čuva u globalnoj promenljivoj i kasnije će služiti za slanje rezultata proračuna wemus server. Params parametar nosi ulazne parametre za pokretanje proračuna.

6.7.2. Slanje rezultata na wemus server

Funkcija šalje izlazne podatke, musico programa, wemus serveru, koristeći callback funkciju prosleđenu uz funkciju za pokretanje proračuna. Kako bi podaci bili poslani, callback funkciji se mora proslediti socket i podaci koje treba da pošalje preko socket-a.

6.7.2.1. *Potpis funkcije:*

```
sendData = function()
```

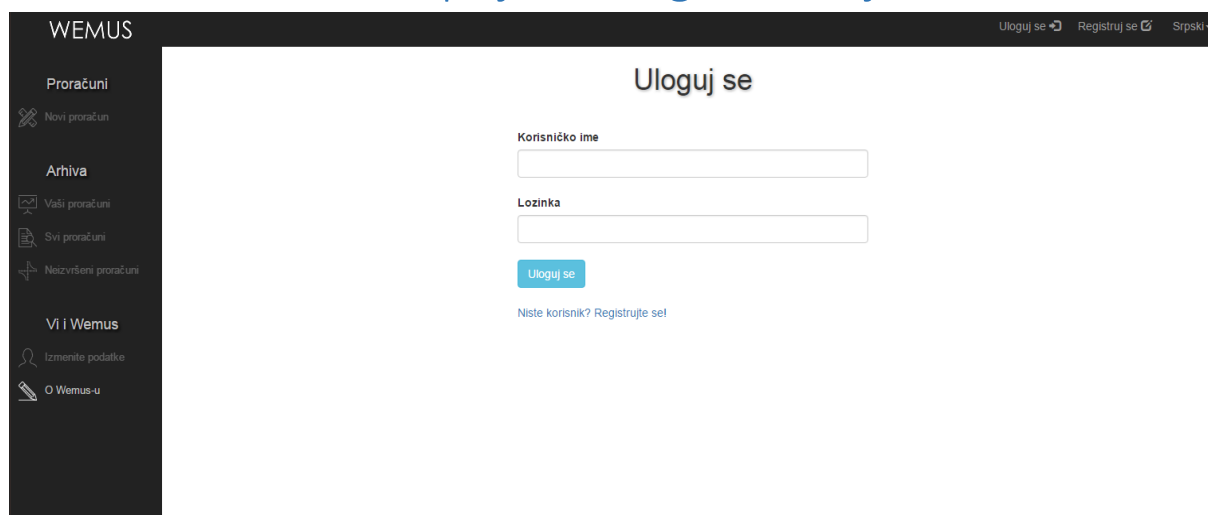
6.7.2.2. *Potpis callback funkcije:*

```
returnData = function(socket,data)
```

7. Dizajn korisničkog interfejsa

U nastavku ovog poglavlja će biti prikazano kako će izgledati delovi aplikacije kao i kratak opis načina njihovog funkcionisanja. Detaljan opis okruženja i načina njihovog korišćenja je dat je u prethodnim delovima dokumenta. Ovde su prikazane stranice koje su zajedničke za sve tipove korisnika aplikacije dok je administrativna strana poseban jedini deo koji vide samo administratori projekta.

7.2. Stranica za prijavu i registrovanje korisnika

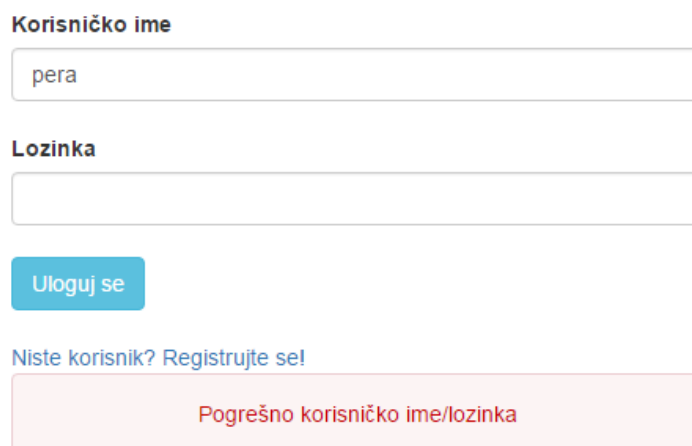


The screenshot shows the WEMUS application interface. On the left is a dark sidebar with the WEMUS logo and navigation items: 'Proračuni' (with sub-items 'Novi proračun', 'Vaši proračuni', 'Svi proračuni', 'Nadzirani proračuni'), 'Vi i Wemus' (with sub-items 'Izmenite podatke', 'O Wemus-u'). The main content area is titled 'Uloguj se' and contains a login form with two input fields: 'Korisničko ime' and 'Lozinka'. Below the fields is a blue 'Uloguj se' button and a link 'Niste korisnik? Registrujte se!'.

Slika 61 - Forma za prijavljivanje na sistem

Kako bi moglo da se pristupi aplikaciji korisnik se mora prijaviti na sistem. Ukoliko nije registrovan član moraće da se registruje klikom na link koji će ga odvesti na novu formu.

Uloguj se



This close-up shows the login form with the following elements: the title 'Uloguj se', the 'Korisničko ime' label and input field containing 'pera', the 'Lozinka' label and empty input field, a blue 'Uloguj se' button, and a red error message box at the bottom that says 'Pogrešno korisničko ime/lozinka'.

Slika 62 - Upozorenje ukoliko je pogrešno unet podatak

Uloguj se



Slika 63 - Gif koji se prikazuje pri pokušaju prijave na aplikaciju

Ukoliko su podaci za prijavu na sistem validno uneti i korisnik je već registrovan moćiće odmah da se prijavi. Pri popunjavanju forme za registraciju, uključena je validacija koju gost mora da poštuje pri unosu inače će se polja boji u crveno. Pri prijavi se pojavljuje gif kao na slici. Isti je i za registraciju. Ukoliko je prijava validna korisnik može nastaviti sa korišćenjem sistema.

Registruj se

Ime	Prezime
<input type="text" value="mike"/>	<input type="text" value="millers "/>
Elektronska pošta	
<input type="text" value="ss"/>	
Neispravna email adresa.	
Korisničko ime	
<input type="text" value="s"/>	
Neispravno korisničko ime. Korisničko ime mora početi slovom i sme da sadrži tačku i donju crtu. Takođe mora da ima najmanje dva karaktera.	
Vaša lozinka	
<input type="password" value="....."/>	
Neispravna lozinka. Lozinka mora da ima najmanje 8 karaktera i da sadrži bar jedno od sledećeg: veliko slovo, malo slovo, broj i specijalni karakter ! # \$ % & ? ili razmak.	
Potvrdite Vašu lozinku	
<input type="password" value="."/>	
Lozinke se ne poklapaju.	
<input type="button" value="Registruj se"/>	

[Već ste član? Prijavite se!](#)

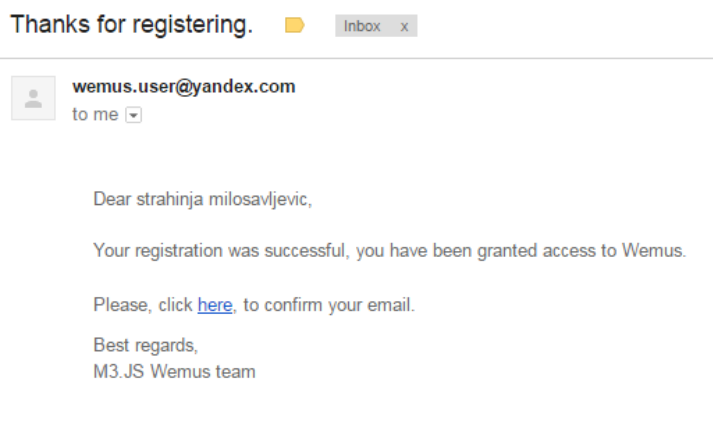
Slika 64 - Forma za registrovanje gosta

Ukoliko gost popunjava formu za registraciju pritiskom na dugme „registruj se“, sva polja moraju proći validaciju tj. da budu označena zeleno, tek tada gost postaje upisan u sistem i imaće mogućnost pristupa aplikaciji nakon što verifikuje navedenu e-mail adresu. Nakon toga može da se prijavi i da počne sa izvršavanjem proračuna i upotrebom sistema. Na slici je dat prikaz validno/nevalidno unetog polja i ispis objašnjenja šta je korisnik loše uneo kako bi mogao adekvatno da ispravi.

Uspešno ste se prijavili!

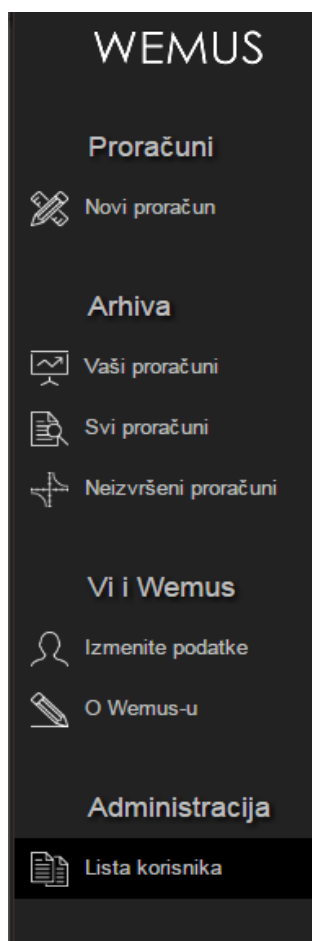
Molimo Vas da proverite Vašu e-poštu i potvrdite verifikacioni link pre [prijavlivanja](#).

Slika 65 - Poruka pri uspešnoj registraciji



Slika 66 - Prikaz mejla za potvrdu

7.3. Fiksirani delovi aplikacije



Slika 67 - Navigacioni panel

Glavni navigacioni meni pri pokretanju aplikacije će se nalaziti sa leve strane i biće fiksiran kako bi korisnik imao lakši i pregledniji način upotrebe i navigiranja.

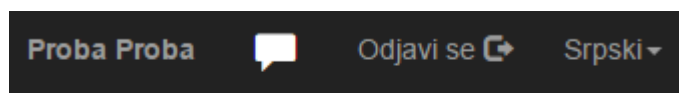
Sastojaće se od 4 kategorije (odeljka):

- Proračuni
- Arhiva
- Vi i Wemus
- Administracija

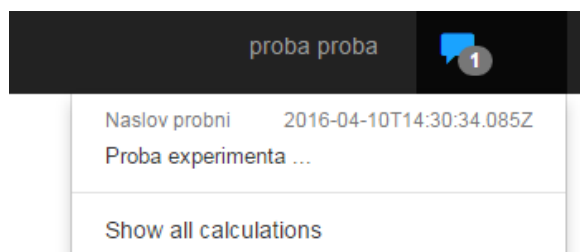
U proračunima se može pokrenuti nov proračun koji se kasnije može editovati, brisati ili pogledati iz odseka arhiva. U delu „Vi i Wemus“ se mogu izmeniti korisnikovi podaci ili pročitati svrha projekta i čemu služi.

Administracija će biti vidljiva samo administratorima sajta koji će moći da je i koriste.

Pored glavnog menija sa leve strane, postojaće i gornji na kome će biti prikazano ime korisnika, notifikacije ukoliko je neki proračun nepročitan, opciju odjavljivanja i promene jezika na engleski ili srpski.



Slika 68 - Gornji desni meni

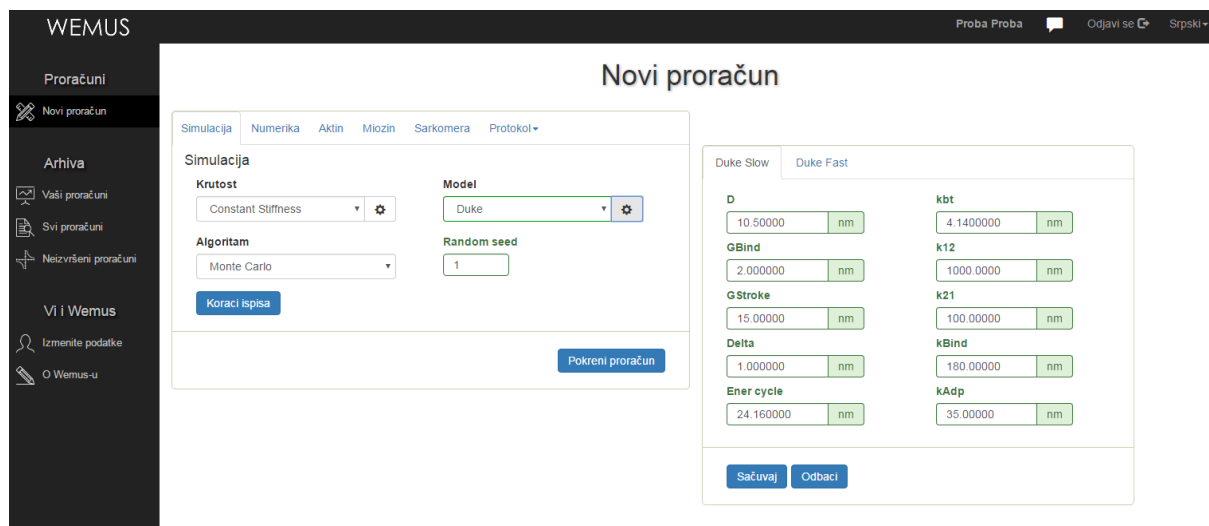


Slika 69 - Notifikacija sa nepročitanim proračunom

Ukoliko ima neki nepregledani proračun korisniku iskače obaveštenje o tome da može ali i ne mora da ga pregleda ili čak sve proračune da pogleda.

7.4. Pokretanje proračuna

Kada se prijavi na sistem otvoriće se stranica za pokretanje novog proračuna. Tu se mogu uneti sve vrednosti ali forme su već popunjene sa predefinisanim vrednostima kako bi korisnik mogao da odmah pokrene neki proračun pa ih može samo izmeniti po svojoj potrebi. Sa desne strane svakog polja za unos će se nalaziti i merna jedinica koja olakšava korisniku da zna koje vrednosti treba da unese. Točkići otvaraju dodatne forme koje će sadržati dodatna polja u zavisnosti sta se odabere iz padajuće liste. Na ovoj stranici se nalaze svi neophodni podaci za pokretanje proračuna samo su oni podeljeni po tabovima radi lakše navigacije po odsecima sarkomere.



Slika 70 - Pokretanje novog proračuna

Koraci ispisa

Vreme (s)		Vreme (s)	Obrisi
<input type="text" value="0.0"/> <input type="button" value="s"/>	<input type="button" value="Dodaj korak ispisa"/>	0.45	<input type="button" value="✘"/>
		12	<input type="button" value="✘"/>

Slika 71 - Koraci ispisa simulacije

Forma za unos numerike sadrži osnovne podatke numerike. Korsnik mora i da čekira opciju koju želi. Sve promene koje unese u polja sa desne strane može da poništi klikom na „Odbaci“ ili da ih sačuva na „Sačuvaj“.

Slika 72 - Forma za unos numerike

Slika 73 - Forma za aktin podatke

U nastavku su dati ostali tabovi koji moraju biti validno svi popunjeni kako bi se proračun pokrenuo. Na slikama je dugme “Pokreni proračun” omogućeno baš iz razloga što postoje predefinisane vrednosti i proračun može da se odmah pokrene sa ovim vrednostima. Ukoliko neka vrednost nije pravilno uneta dugme neće biti aktivno, a korisnik će se obavestiti o neispravnosti parametara i biće označeno koje polje treba da proveri.

The screenshot shows the 'Novi proračun' (New Calculation) form for the 'Miozin' (Myosin) tab. The form is divided into several sections with input fields and sliders:

- Poprečni presek:** 196.070000 nm²
- Elastičnost:** 0.67323000e+003 pN/nm²
- Rastojanje krunica:** 14 nm (slider)
- Udaljenost do prve krunice:** 88 nm (slider)
- Procenat sporih krunica:** 0% (slider)
- Procenat brzih krunica:** 100% (slider)
- Broj krunica:** 50 num

A 3D diagram of a myosin molecule is shown on the right side of the form. The 'Pokreni proračun' (Run Calculation) button is active.

Slika 74 - Forma za podatke o miozinu

The screenshot shows the 'Novi proračun' (New Calculation) form for the 'Sarkomera' (Sarcomere) tab. The form is divided into several sections with checkboxes, input fields, and a dropdown menu:

- Titin:**
 - Postojanje titina:
 - Krutost titina: 0.12707
- Nebulin:**
 - Deficijencija nebulina: Bez nebulina
 - Numerics Xamax: 7.00000
- Protein C:**
 - Postojanje proteina:
 - Startna pozicija: 0
 - Broj krunica: 0
- Protein C stiffness:**
 - K: 1.25 pN/nm
 - BPC Model: DukeBPC

A 3D diagram of a sarcomere is shown on the right side of the form. The 'Pokreni proračun' (Run Calculation) button is active.

Slika 75 - Forma za podatke o sarkomeri

U zavisnosti koji protokol se odabere iz padajuće liste otvara se forma za unos podataka o protokolima. Oba su slična pa ćemo prikazati samo jedan od njih.

Slika 76 - Forma za popunjavanje podataka o protokolu

Parametri diskretne funkcije

Slika 77 - Ukoliko se u protokolu odabere diskretna funkcija

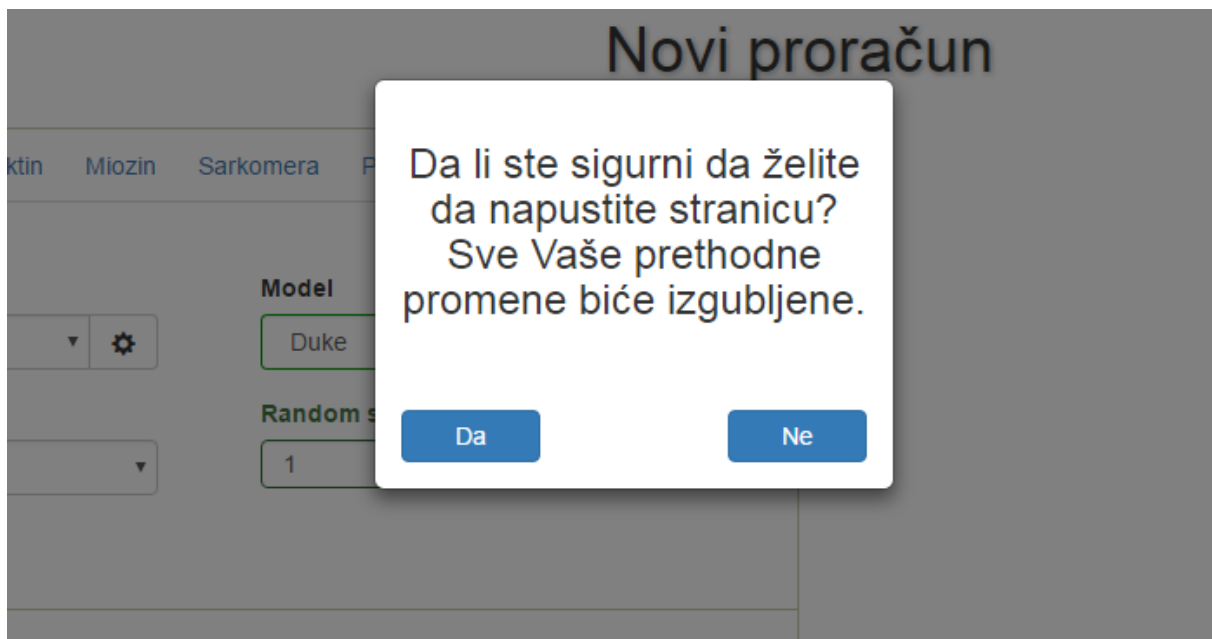
Slika 78 - Podaci za kalcijum protokol

Početno vreme funkcije	Korak simulacije	Zadato	Tip funkcije	a	b	c	d	Obrisi	Izmeni
0.0	1.0e-5	Sila (pN)	Konstantna funkcija	0.0	0.0	0.0	0.0	✕	
0.5	1.0e-5	Sila (pN)	Konstantna funkcija	0.0	0.0	0.0	0.0	✕	

« 1 »

Slika 79 - Izmena i brisanje odabranih protokola

Ako se pri popunjavanju formi za pokretanje novog proračuna pritisne neka opcija iz menija sa strane koja će prekinuti ovo popunjavanje korisnik treba biti obavešten o tome.



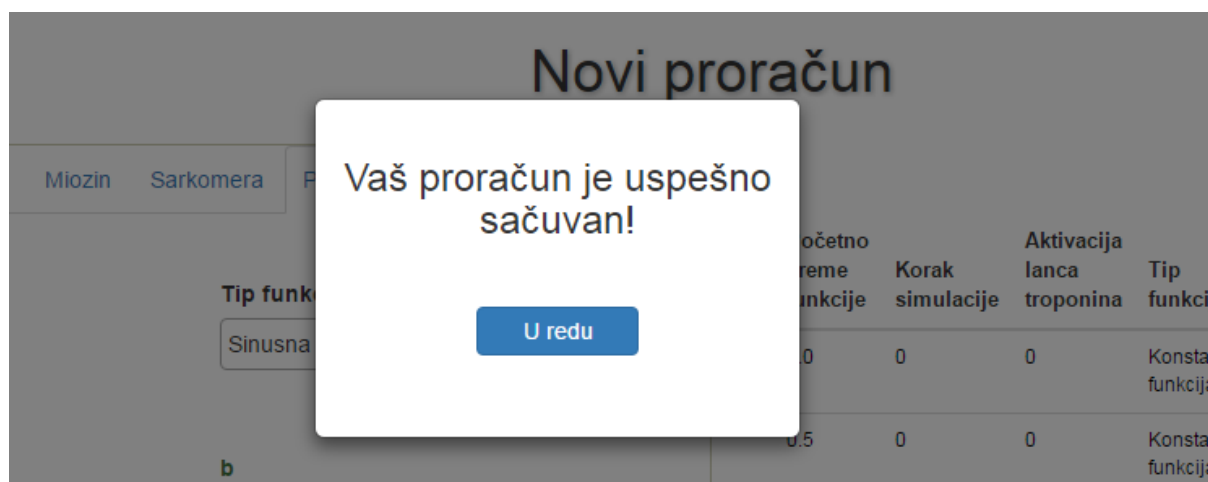
Slika 80 - Prekid popunjavanja proračuna

Kada se klikne na dugme „Pokreni proračun“ otvoriće se prozor u kome se od korisnika traži da unese naslov koji je obavezan i komentar. Zatim se otvaraju mogućnost odabira da li da se proračun pokrene ili samo da se sačuva za kasnije pokretanje.

Slika 81 - Naslov i komentar proračuna

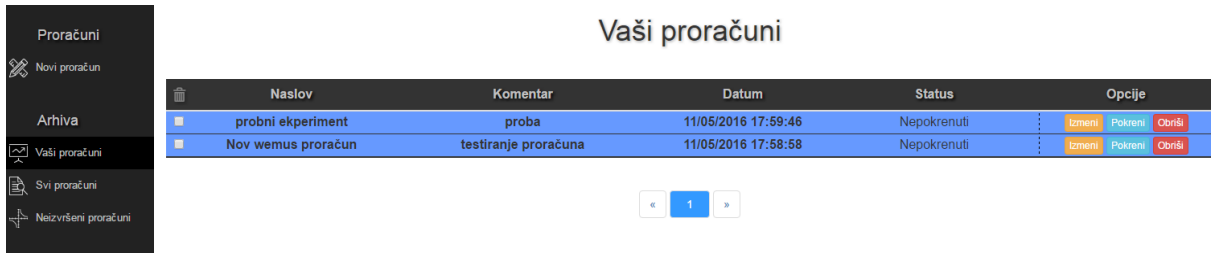
7.5. Čuvanje i prikazivanje proračuna

Ako se odabere opcija iz prethodnog odeljka da se proračun sačuva za kasnije pokretanje onda se otvara iskačući prozor koji o tome obaveštava korisnika.

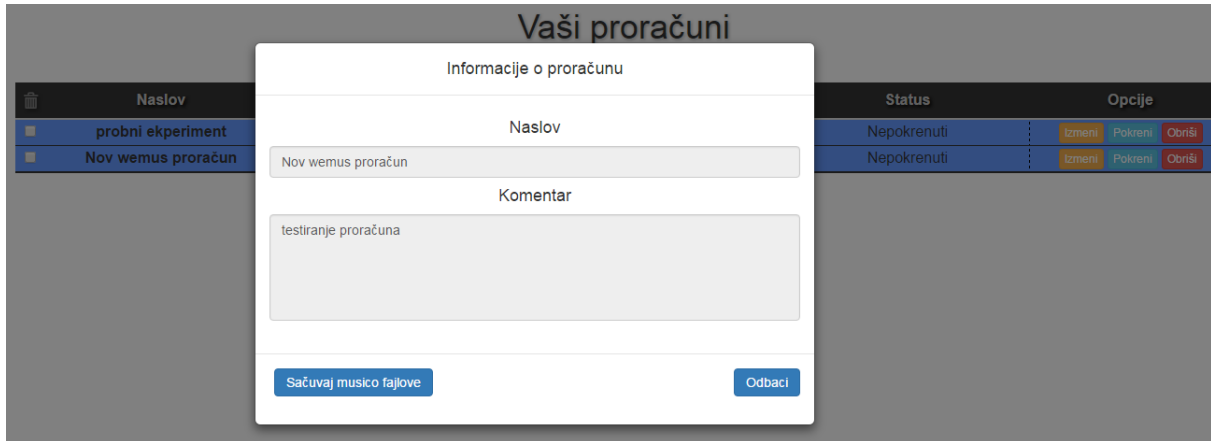


Slika 82 - Sačuvan proračun

U opciji menija sa strane aplikacije se može videti deo za pregled korisnikovih proračuna i tu će moći da ima neke dodatne opcije kao i pregled sačuvanih i izvršenih proračuna. Korisnik može čekirati više svojih proračuna i klikom na korpu ih obrisati. Takođe ako pritisne dugme „Izmeni“ ulazni podaci izabranog proračuna se učitavaju u ulazne forme gde ih on može menjati i ponovo sačuvati, pokrenuti proračun ili odbaciti. U polje „Status“ treba obavestiti korisnika o trenutnom stanju izvršavanja proračuna. Ako se proračun izvršava stajaće progres bar i dugme „Zaustavi proračun“.

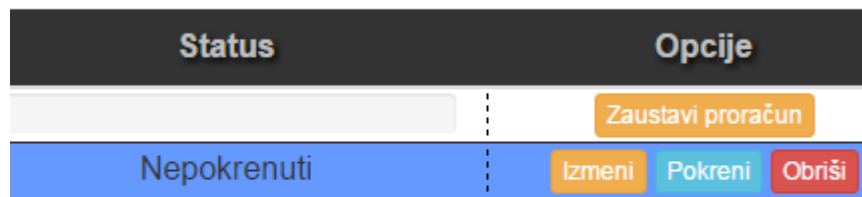


Slika 84 - Prikaz proračuna

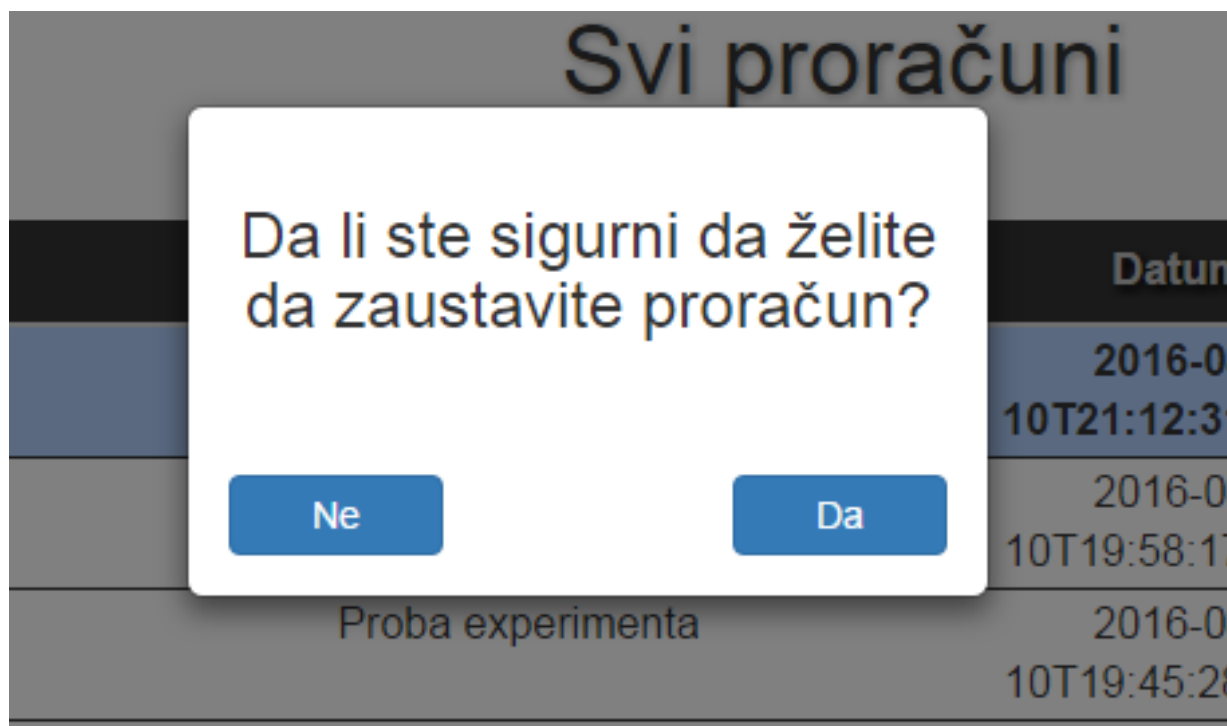


Slika 83 - Podaci o proračunu

Klikom na proračun određeni možemo da vidimo iskaćući prozor koji nas obaveštava o već ispisanim podacima ali i mogućnost da se skinu musico fajlovi za taj proračun.



Slika 85 - Zaustavi proračun



Slika 86 - Zaustavljanje pokrenutog izvršavanja proračuna

Ako se odabere opcija iz menija sa leve strane „Svi proračuni“ onda se mogu videti proračuni drugih registrovanih korisnika i opcija da se obrišu samo proračuni registrovanog korisnika dok jedino administrator ima mogućnost brisanja bilo čijeg proračuna.

Korisnik	Naslov	Komentar	Datum	Status	Opcije
	Naslov probni	Proba experimenta	2016-04-10T21:12:31.285Z	Završen	Prikaži Izmeni
	Naslov probni	Proba experimenta	2016-04-10T19:58:17.005Z	Završen	Prikaži Izmeni
	Naslov probni	Proba experimenta	2016-04-10T19:45:28.685Z	Završen	Prikaži Izmeni
	Naslov probni	Proba experimenta	2016-04-10T19:13:38.249Z	Završen	Prikaži Izmeni
	Naslov probni	Proba experimenta	2016-04-10T17:40:54.567Z	Završen	Prikaži Izmeni
	Naslov probni	Proba experimenta	2016-04-10T14:30:34.085Z	Završen	Prikaži Izmeni
	Naslov probni	Proba experimenta	2016-04-10T14:05:05.943Z	Završen	Prikaži Izmeni

Slika 87 - Svi proračuni

7.6. Pregledanje izvršenih proračuna

Kada se klikne u listi proračuna „pokreni“ otvara se obaveštenje korisniku da se pripremaju izlazni podaci.

Rezultati proračuna

Pripremamo vaše podatke...

27%

Slika 88 - Progres bar učitavanja izlaznih podataka

Rezultati se prikazuju na jedan od dva načina:

- Tabelarno
- Grafički

Korisnik na grafiku može odabrati koje će izlazne parametre prikazati i to silu, broj krosbridževa, pomeranje kraja aktina, pomeranje kraja miozina ili kombinovan prikaz sile i broja krosbridževa.

Rezultati proračuna



Slika 89 - Grafik rezultata proračuna



Slika 90 - Opcije za grafik

Opcije za grafik su redom:

- pomeranje ručno po osi x i y,
- uvećavanje (zumiranje grafika)
- umanjivanje
- zumiranje po y osi
- umanjivanje po y osi
- zumiranje po x osi
- umanjivanje po x osi
- olovčica za crtanje po grafiku
- čišćenje grafa od crtanja
- prikaz preko celog ekrana

Sačuvaj tabelu rezultata Sačuvaj sliku grafa Sačuvaj musico fajlove

F DLAN DLMN CB

Vreme (s)	Sila (pN)	Pomeranje kraja aktina (nm)	Pomeranje kraja miozina (nm)	Mostova po vlaknu (cross-bridges)
0.00001	0.00429	0	0	0.01
0.00002	0.000175	0	0	0.015
0.00003	-0.00381	0	0	0.035
0.00004	-0.0164	0	0	0.04
0.00005	-0.0282	0	0	0.05
0.00006	-0.0282	0	0	0.05
0.00007	-0.0309	0	0	0.06
0.00008	-0.0196	0	0	0.065
0.00009	-0.0196	0	0	0.065
0.0001	-0.0302	0	0	0.07
0.00011	-0.0302	0	0	0.07
0.00012	-0.0302	0	0	0.07
0.00013	-0.0302	0	0	0.07
0.00014	-0.0302	0	0	0.07
0.00015	-0.0302	0	0	0.07
0.00016	-0.0302	0	0	0.07

Pretraži po vremenu
 Pritisni enter

« 1 2 3 4 5 »

Stranica - 467
 Pritisni enter

Slika 91 - Tabelarni prikaz rezultata

Tabelarni prikaz imaće opciju u gornjem desnom uglu šta želi da prikaže kako bi se korisnik snašao lakše. Sa donje desne strane unosi se pretraga po određenom vremenu a sa desne određena strana paginacije proračuna. Korisnik takođe klikom na liniju funkcije u grafičkom prikazu popunjava polje za pretragu tabele po vremenu. U gornjem levom uglu su mogućnosti čuvanja podataka, grafa i musico fajlova.

7.7. Ažuriranje korisničkih podataka

Ažurirajte vaše lične podatke

Ime Prezime

Elektronska pošta

Korisničko ime

Unesite vašu staru lozinku kako biste potvrdili izmene:

Promenite šifru klikom na dugme:

Nova šifra

Potvrdite Vašu lozinku

Slika 92 - Ažuriranje korisničkih podataka

Za ažuriranje je takođe omogućeno pravilno popunjavanje podataka za izmenu preko validacije, ukoliko želi korisnik da promeni lozinku kliknuće na dugme „promeni šifru“. Kada završi izmenu mora da potvrdi lozinku kako bi sistem prihvatio izmene.

7.8. O projektu i aplikaciji Wemus

I registrovani i neregistrovani članovi imaju prisup podsekciji „O wemus-u“ aplikaciji. Tu može da pročitaju nešto o samom projektu kao i da pogledaju tutorijal koji će da ih sprovede kroz opcije gde mogu da i kako da popune podatke .

WEMUS Uloguj se [Registruj se](#) [Srpski](#)

Dobrodošli

O Wemus-u:

Wemus je Klijentsko-serverska web aplikacija za simulaciju ponašanja mišića na molekularnom nivou. Program je zasnovan na simulaciji bioloških i biomehaničkih procesa u sarkomeri, osnovnoj gradivnoj jedinici mišića. Wemus sistem obezbeđuje intuitivno korisničko okruženje za definisanje parametara proračuna zadavanjem numeričkih vrednosti i interaktivnim grafičkim komponentama. Aplikacija omogućava pokretanje proračuna na serverskoj strani, korišćenjem postojećeg MUSICO softvera.

[manje](#)

O Sarkomeri:

Sarkomera je odsečak mišićnog vlakna koji se ponavlja. Osnovna je kontraktilna jedinica poprečno-prugastoga mišićnog tkiva. Mišići se sastoje od cevnastih mišićnih vlakana, miofibrila i miofibrila, koji nastaju u procesu koji nazivamo miogeneza. Mišićne su stanice sastavljene od cevnastih miofibrila. Miofibrile su sastavljene od ponavljajućih odsečaka sarkomera, koji pod mikroskopom izgledaju kao tamne i svetle vrpce. Sarkomere se sastoje od dugih, vlaknastih belančevina koje klize jedna pored druge kad se mišić stežu i opuštaju...

[više](#)

Kako koristiti Wemus




Novi proračun

Slika 93 - O wemus projektu

7.9. Administrativna strana

Administrativnu stranu može da vide samo administratori sajta i tu imaju opciju da dodele ili uklone prava korisnicima kao i mogućnost brisanja istih. Administrator može da selektuje više korisnika od jednom kako bi ih lakše obrisao iz sistema. Ovde se nalaze informacije o korisnicima, tipu korisnika i njihovim podacima.

Administrativna strana

	Korisničko ime	Korisnički mejl	Prezime	Ime	Tip korisnika	Obrisi korisnika	Dodeli Prava
	proba	tmiljevic94@gmail.com	Proba	Proba	user	Obrisi korisnika	Izaberite pravo pristupa Promeni prava
	balmain	stralekg@gmail.com	milosavljevic	strahinja	user	Obrisi korisnika	Izaberite pravo pristupa Promeni prava

Slika 94 - Administracija