

# Fitness aplikacija za praćenje dnevni aktivnosti treninga

Završni rad

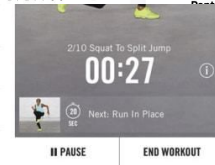
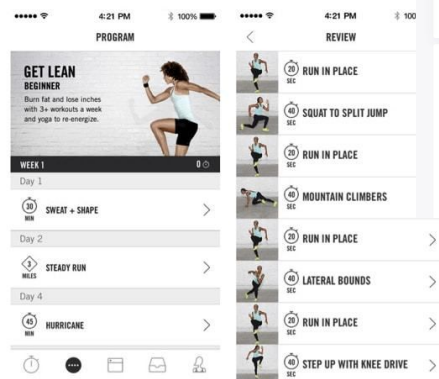
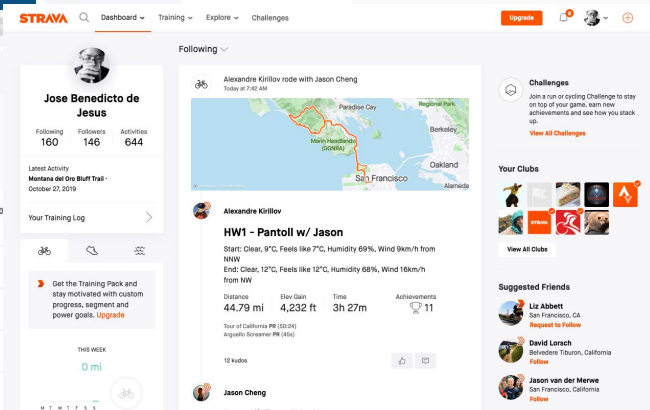
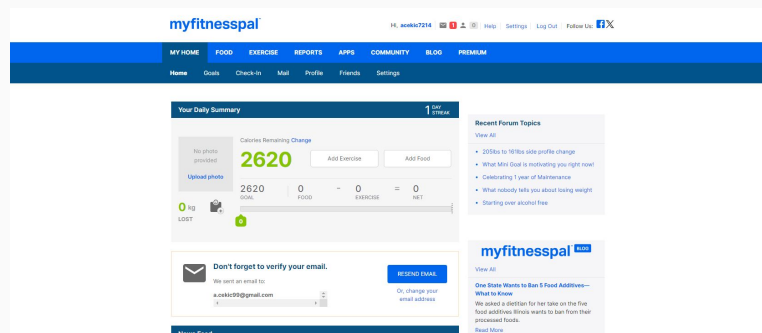
Aleksa Cekić 4173

# Sadržaj

1. Pregled postojećih fitness aplikacija
2. Tehnologije
3. Zahtevi sistema
4. Implementacija
5. Demonstracija

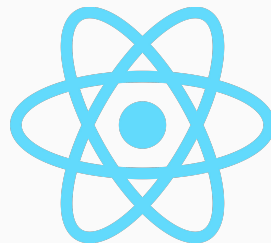
# 1. Pregled postojećih fitness aplikacija

1. MyFitnessPal
2. Lifesum
3. Nike training Club
4. Strava
5. Fitbit



## 2. Tehnologije

- ReactJs
- Spring Boot
- MySQL



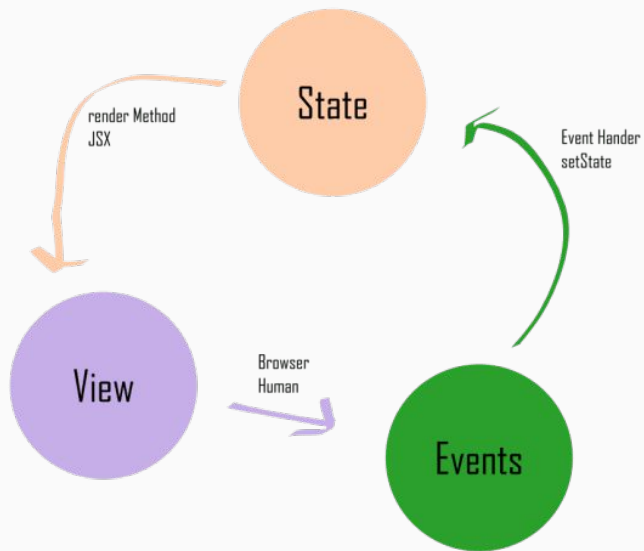
spring®

# ReactJs

- Popularni Javascript radni okvir.
  - Razvijen od strane Facebook-a.
  - Efikasan, brz, reaktivan.
  - Koristi se za razvoj veb aplikacija.
  - Koristi komponente da “razbije” aplikaciju.
  - **Virtualno** reprezentuje **DOM** radi poboljšavanja performansi.
  - **JSX (JavaScript XML)** jezik za opisivanje korisničkog interfejsa.
  - Jednosmeran protok podataka
- Koristi **Hooks** za prikaz stanja aplikacije.
  - **React Router** upravlja rutama aplikacije.



## Unidirectional Data Flow

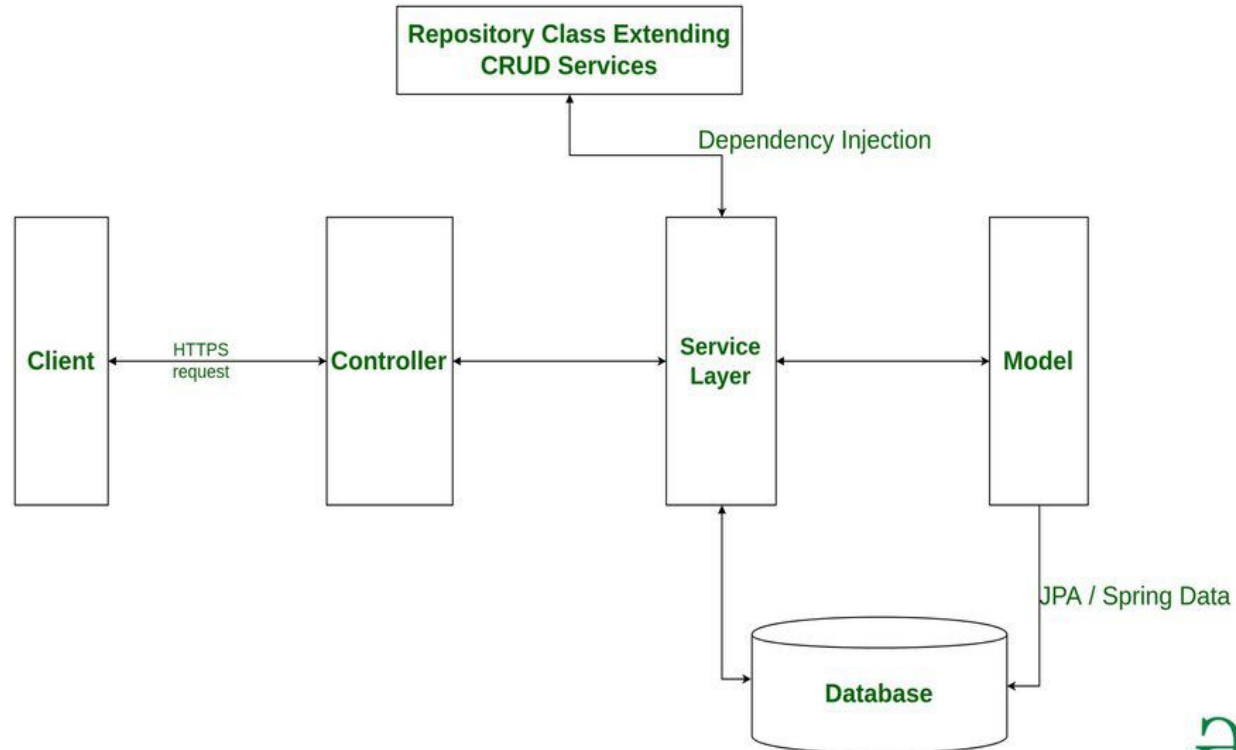


Podaci se prenose od nadređene do podnadređene komponente.

# Spring Boot

- Radni okvir za izradu Java veb aplikacija.
- Kreiran od strane Rod Johnson-a.
- Brz i olakšan razvoj uz jednostavnu konfiguraciju
- Deo Spring ekosistema (Spring Framework, Spring Data, Spring Security, itd...)
- Brza implementacija
- Ugrađen Tomcat server
- Podržava razvoj različit tip arhitektura (npr. Mikroservisna Arhitektura)
- Upravljanje zavisnostima (***Maven*** i ***Gradle***)
- Testiranje

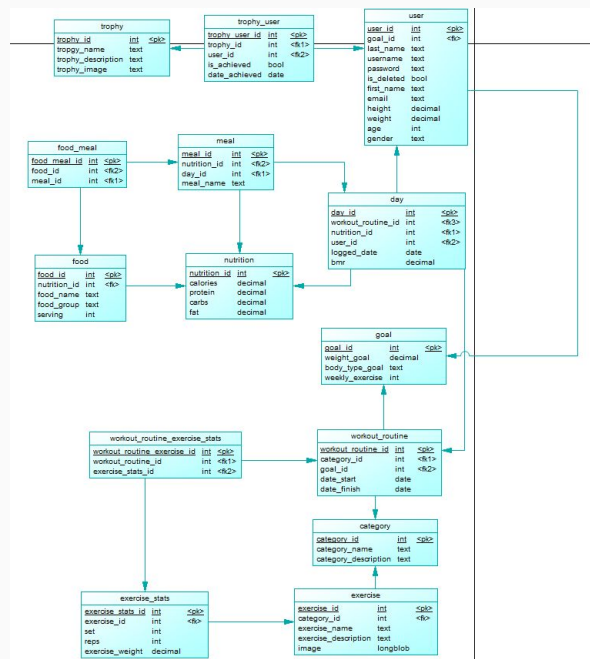
## Spring Boot flow architecture





# MySQL baza podataka

- Popularni open-source sistem za upravljanje bazama podataka (RDBMS).
- Koristi SQL (Structured Query Language) za upravljanje podacima.
- Organizovan skup podataka sa definisanim relacijama.
- Brze performanse
- Definisanje indeksa



# 3. Zahtevi sistema

# Funkcionalni zahtevi

	Title ID	Full Description	Code	Priority
→	1.	<b>Korisnik</b>	REQ_0001	5
2	1.1	<b>Autentifikacija korisnika</b> Korisnici moraju imati nalog da bi pristupili aplikaciji.	REQ_0002	5
3	1.1.1	<b>Logovanje korisnika</b> Korisnici se prijavljuju na aplikaciju sa svojim korisničkim kredencijalima. (username, password).	REQ_0003	5
4	1.1.2	<b>Registrowanje korisnika</b> Ukoliko korisnici ne poseduju vec kreiran nalog, moraju se registrovati na platformi. Prilikom registracije korisnici moraju da unesu podatke o sebi (ime, prezime, email, korisnicko ime, sifru, visinu, težinu, broj godina i pol), takodje moraju i da unesu svoje ciljeve. (željena težina, koliko puta nedeljno bi trenirao kao i konacnog cilja)	REQ_0004	5
5	1.2	<b>Pregled komandne table</b> Korisniku je dostupan pristup komandne table gde imaju sumirane informacije.	REQ_0005	4.5
6	1.2.1	<b>Pregled ukupne ishrane</b> Korisniku je dostupan pregled ukupne ishrane do sada koju je uneo.	REQ_0013	4
7	1.2.2	<b>Pregled unete hrane</b> Korisniku je dostupan pregled svake unete hrane po obrocima.	REQ_0008	5
8	1.2.3	<b>Izmena ličnih ciljeva</b> Korisnik ima mogućnost da izmeni svoje ciljeve ukoliko to želi.	REQ_0006	5
9	1.2.4	<b>Pregled telesne mase</b> Korisniku ima mogućnost prikazu telesne mase (Body Mass Index) na osnovu podataka koje je uneo prilikom registracije.	REQ_0007	5
10	1.3	<b>Kreiranje dana sa planom vežbanja</b> Korisnik ima mogućnost kreiranje danova sa planom vežbanja (treninzima) koje može raditi.	REQ_0009	5
11	1.3.1	<b>Automatsko kreiranje vežbi</b> Korisnik ima mogućnost automatskog kreiranje vežbi na osnovu ciljeva koje je uneo.	REQ_0022	5
12	1.3.2	<b>Ručno kreiranje vežbi</b> Korisnik ima mogućnost kreiranja vežbi ručno, unosom kategorije, datuma i vežbi za tu kategoriju.	REQ_0023	5
13	1.3.3	<b>Pregled kreiranih treninga</b> Korisnik ima mogućnost pregleda svih treninga koje je kreirao po danu.	REQ_0010	5
14	1.3.4	<b>Ažuriranje treninga</b> Korisnik ima mogućnost ažuriranja treninga klikom na "Završi trening" dugme, gde ažurira trening sa datumom završetka.	REQ_0011	5
15	1.3.5	<b>Brisanje treninga</b> Korisnik ima mogućnost brisanja treninga.	REQ_0012	5
16	1.4	<b>Pregled ishrane po kreiranim danima</b> Korisnik ima mogućnost pregleda ishrane za dane sa treninzima koje je prethodno kreirao.	REQ_0015	5

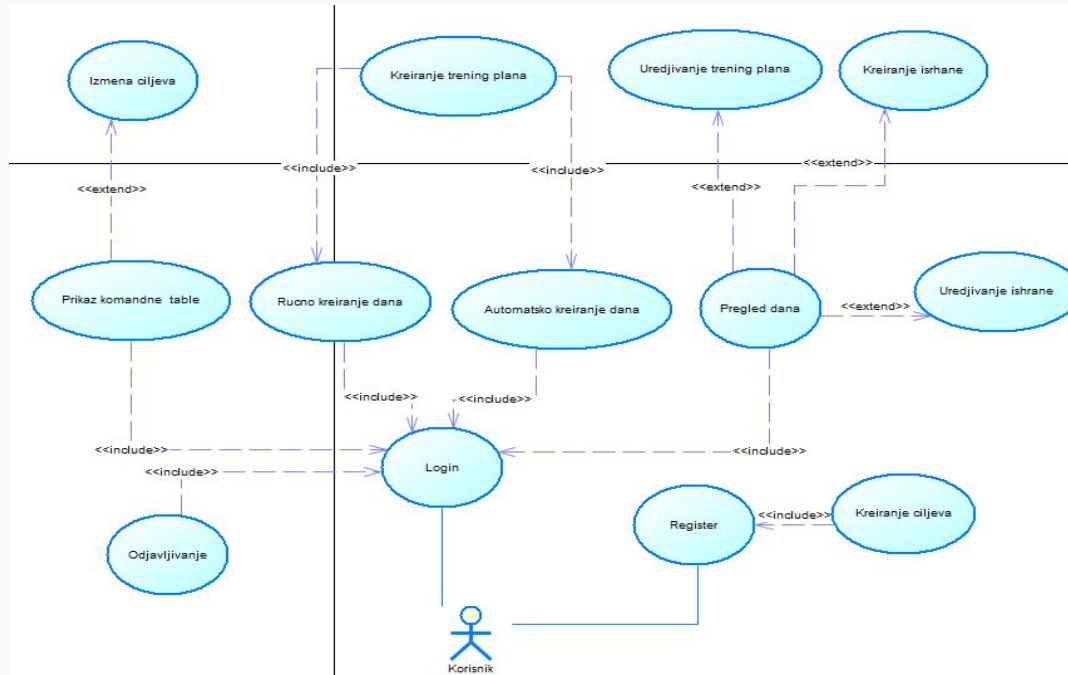
16	1.4	<b>Pregled ishrane po kreiranim danima</b> Korisnik ima mogućnost pregleda ishrane za dane sa treninzima koje je prethodno kreirao.	REQ_0015	5
17	1.4.1	<b>Prikaz metabolizma</b> Korisnik ima mogućnost pregleda metabolizma i dnevnih kalorija koje treba da unese.	REQ_0020	5
18	1.4.2	<b>Kreiranje obroka</b> Korisnik ima mogućnost kreiranje obroka unosom naziva za taj obrok kao i dodavanje hrane.	REQ_0016	5
19	1.4.3	<b>Brisanje hrane iz obroka</b> Korisnik ima mogućnost brisanja hrane iz prethodno kreiranog obroka.	REQ_0019	5
20	1.4.4	<b>Brisanje obroka</b> Korisnik ima mogućnost brisanja obroka.	REQ_0017	5
→	1.5	<b>Odjavljivanje korisnika</b> Korisnik ima mogućnost odjavljivanja sa aplikacije u bilo kom trenutku nakon uspešnog logovanja.	REQ_0021	5

# Nefunkcionalni zahtevi

	Title ID	Full Description
→	1.	<b>Dostupnost</b> Aplikacija mora biti dostupna korisnicima 24/7 bez značajnih prekida. Period održavanja aplikacije trebaju se sprovoditi samo 1-2 puta mesečno, tokom vikenda, posle 16h, kako bi se minimizovao uticaj na korisnike.
2	2.	<b>Sigurnost podataka</b> Svi lični podaci korisnika moraju biti dobro zaštićeni. Pristup važnim podacima kao što su korisničke šifre, mora biti strogo kontrolisan i enkriptovan.
3	3.	<b>Pouzdanost</b> Aplikacija mora biti pouzdana kako bi se sprečilo neodobreno dođavanje ili brisanje podataka.
4	4.	<b>Raspoloživost</b> Aplikacija mora uvek biti dostupna kako bi korisnici mogli nesmetano pristupati svojim podacima i funkcionalnostima.
5	5.	<b>Performanse</b> Aplikacija mora imati brz odziv prilikom učitavanja i navigacije. Optimalno iskustvo korisnika treba biti prioritet i zbog toga minimizovati vreme odziva korisničkih zahteva. Aplikacija treba da učita glavni ekran unutar 2 sekunde od trenutka kada korisnik otvori aplikaciju.
6	6.	<b>Bezbednost</b> Aplikacija mora biti pouzdana i sigurna za upravljanje kako bi se minimizirali potencijalni rizici po zdravlje korisnika. Sve funkcionalnosti vezane za vežbanje moraju biti pažljivo projektovane i implementirane. Ovo uključuje pravilno praćenje i izbegavanje potencijalno opasnih aktivnosti. Prikaz slika treba biti dobro učitao, pregledan i jasan tako da korisnik može dobro ispratiti vežbu.

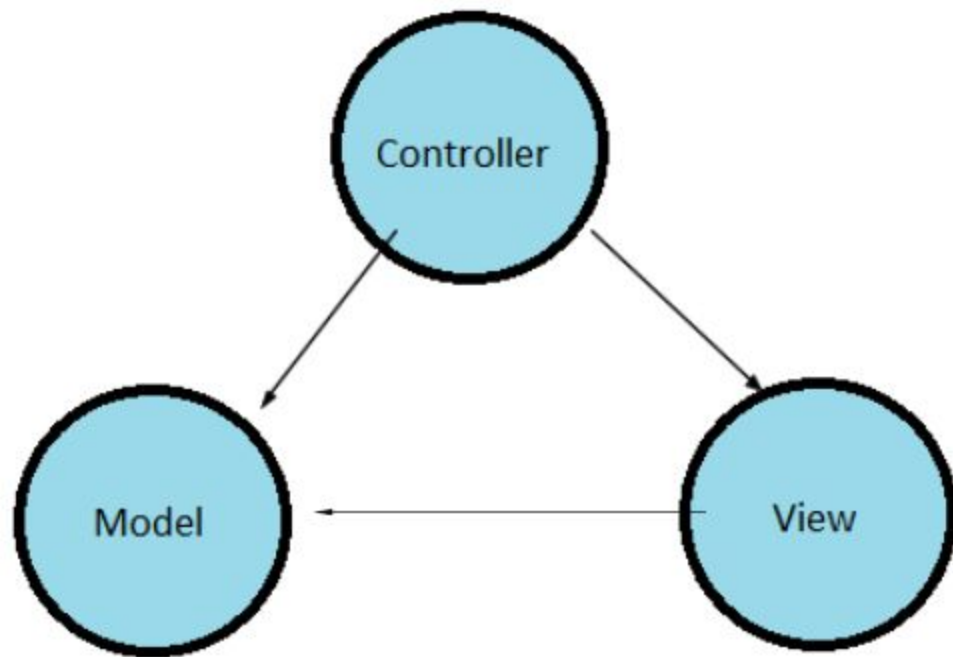
## 4. Implementacija

# Slučajevi korišćenja



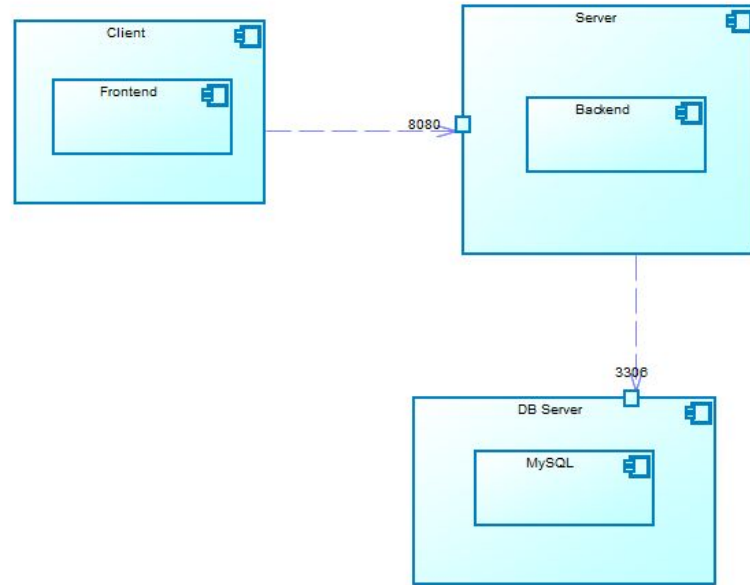
# Arhitektura projekta

- REST (Representational State Transfer) predstavlja arhitektonski stil softvera koji vodi dizajn i razvoj arhitekture različitih veb servisa.
- REST API (Representational State Transfer Application Programming interface) je interfejs koji omogućava komunikaciju između sistema putem HTTP metoda.
- Restful aplikacije su one koje poštuju principe i prakse definisane REST arhitekturom.
- Aplikacija je Restful ako pravilno primenjuje REST principe.
- Oslanja se na principe REST arhitekture.
- MVC (Model-view-Controller) je šablon u softverkom dizajnu koji se koristi za organizaciju komponenti u aplikacijama sa akcentom na one koje imaju UI.

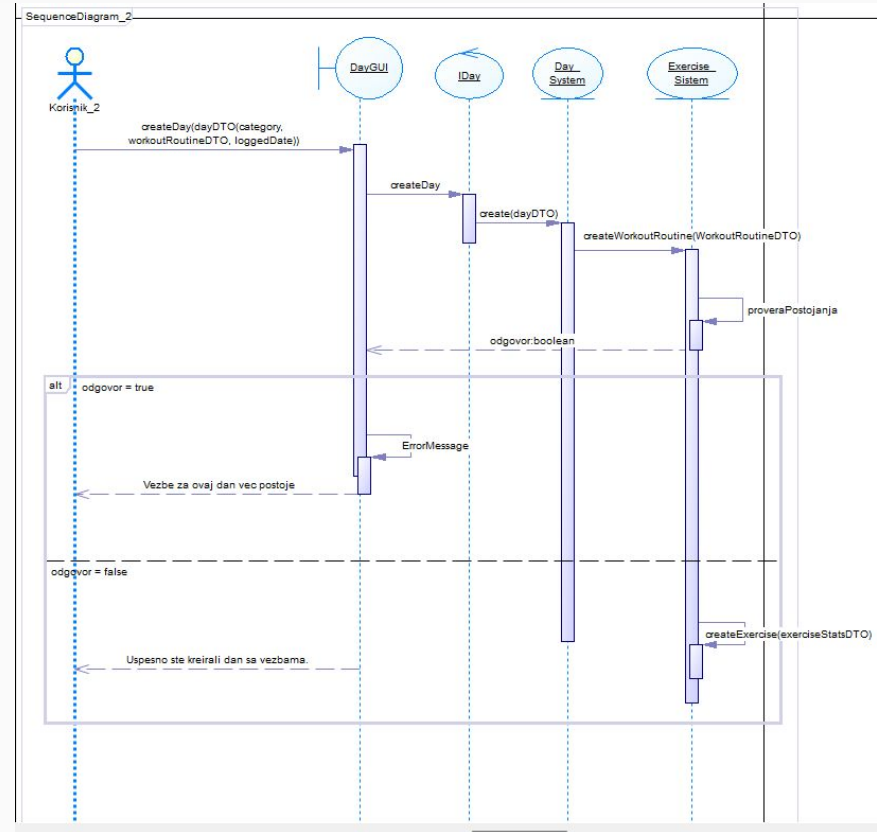
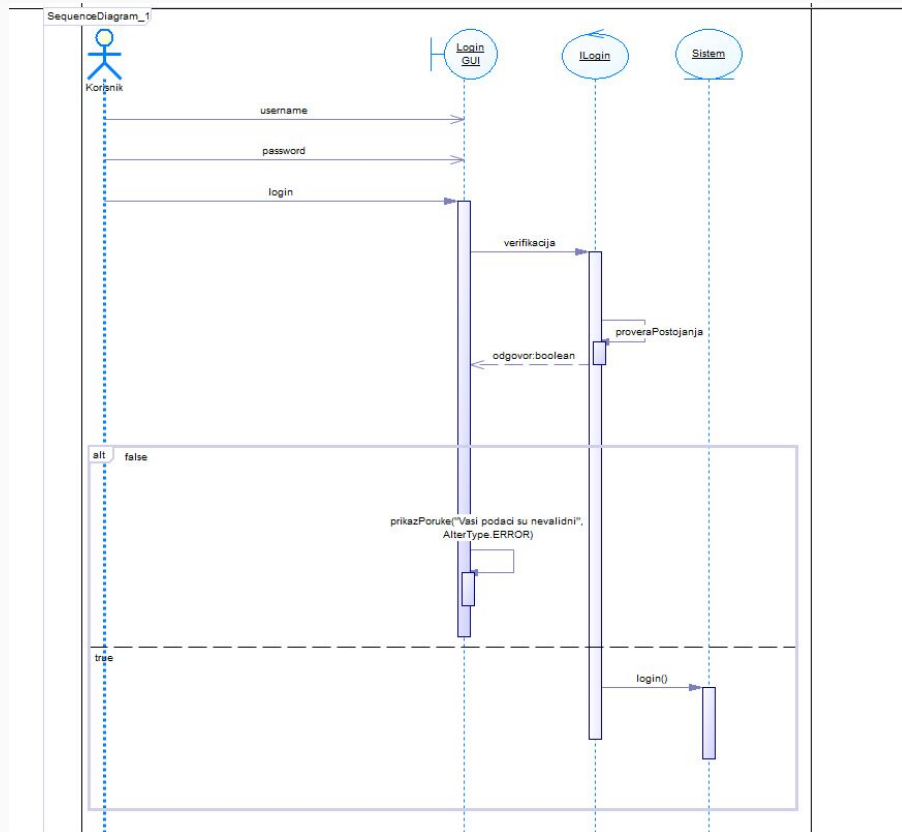




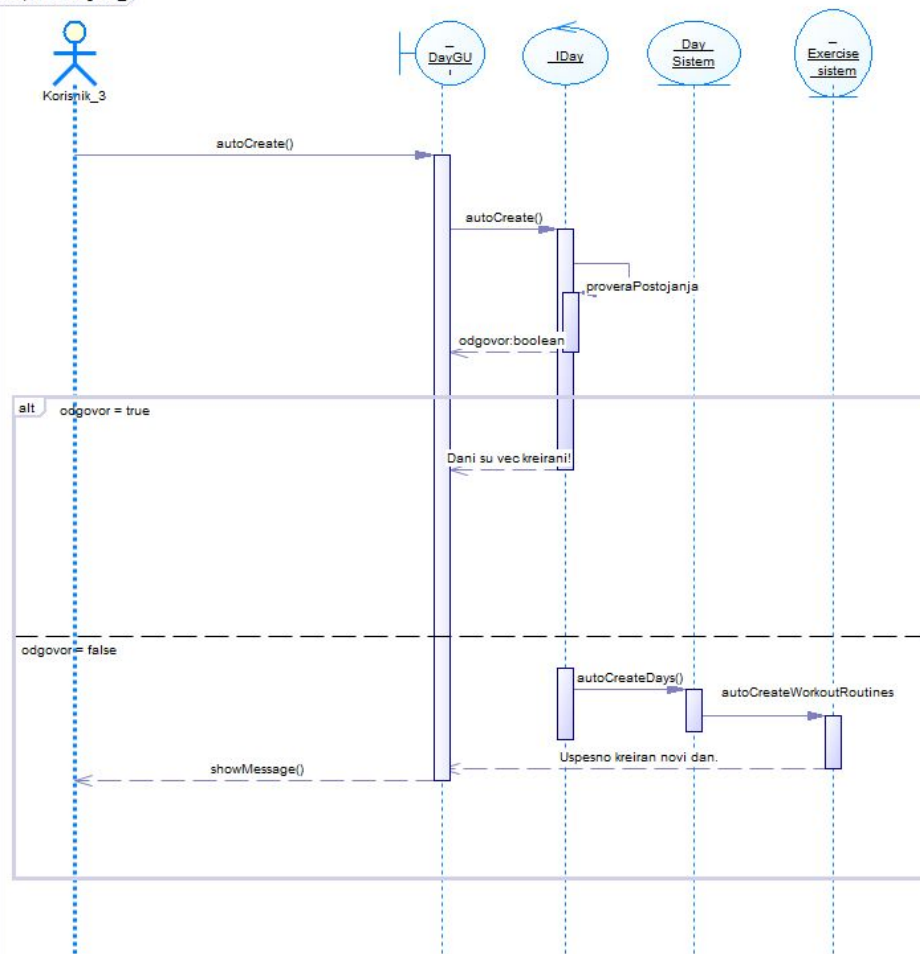
# Klijent-server arhitektura



# Sistem sekvencijalnih dijagrama



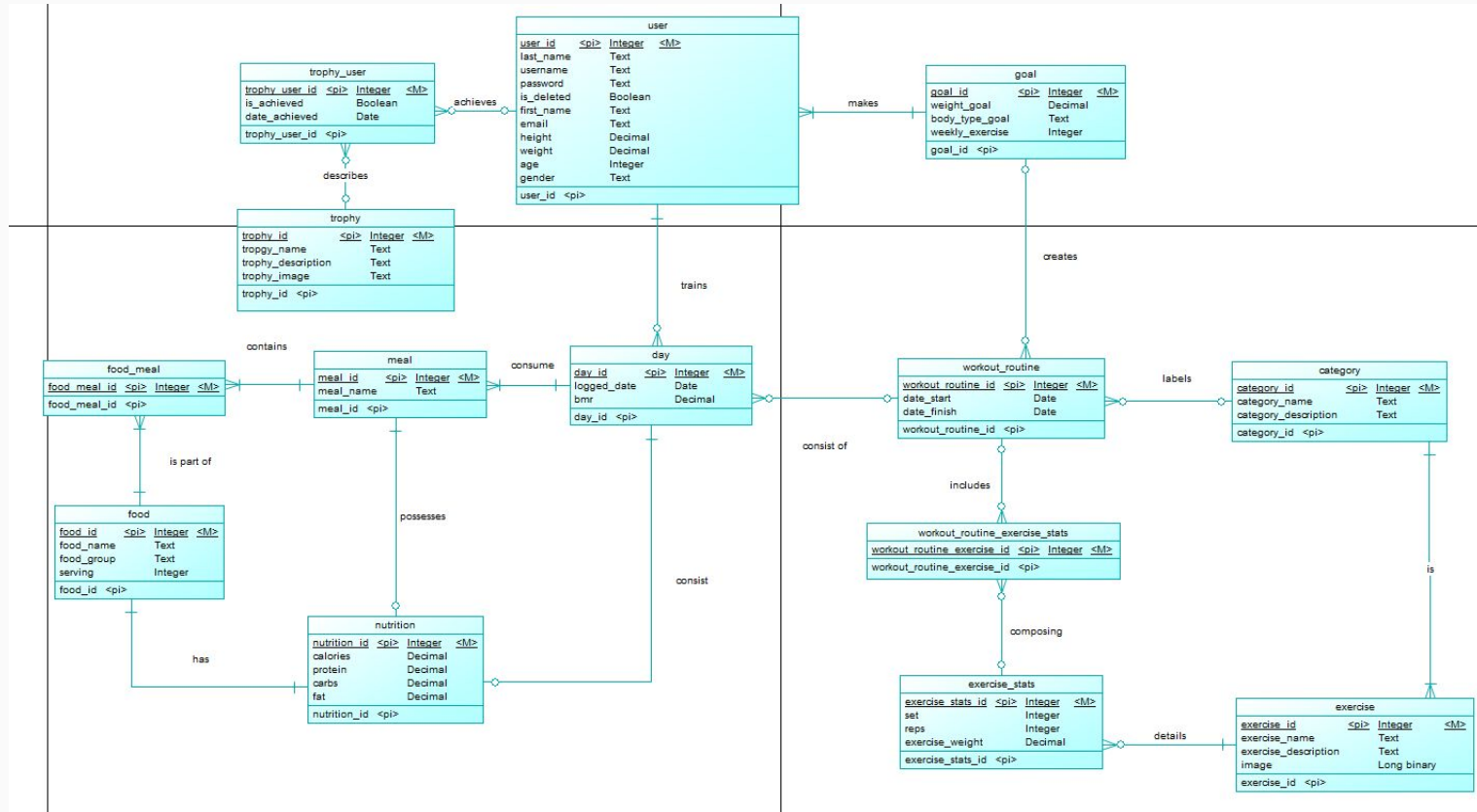
SequenceDiagram\_3



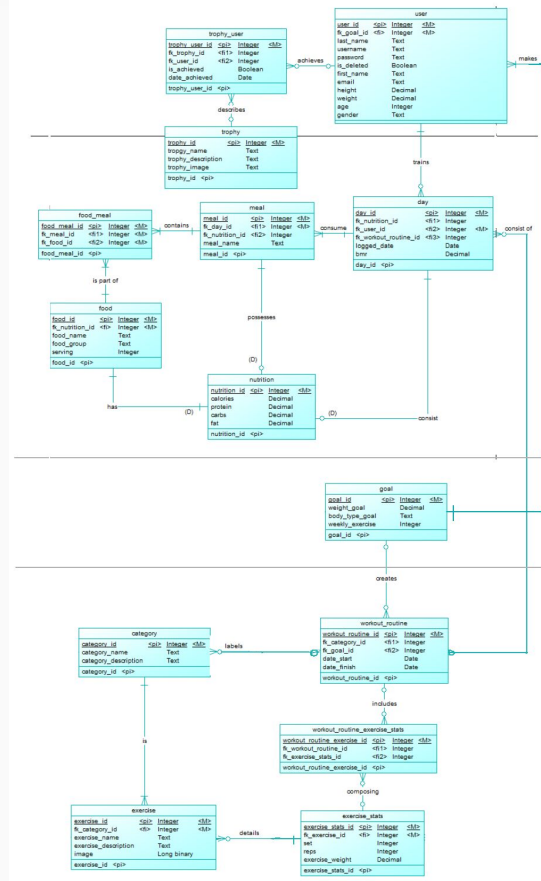
# Baza podataka

- Prikaz dijagrama baze podataka delimo na **konceptualni**, **logički** i **fizički** dijagram.
- Konceptualni dijagram se fokusira na prikazivanje visokog nivoa apstrakcije i konceptualnih veza između entiteta.
- Logički dijagram prelazi na detaljniji nivo i odnosi se na organizaciju podataka u samoj bazi.
- Prikazuje entitete, attribute i veze, dodajući detalje o ključevima
- Fizički dijagram prikazuje konkretnu strukturu baze podataka, najdetaljniji je od sva tri i usredsređen je na tehničke aspekte implementacije.
- Direktno opisuje podatke i kako će oni biti organizovani i sačuvani u bazi.
- Opisuje specifičnosti kao što su tipovi podataka i ograničenja

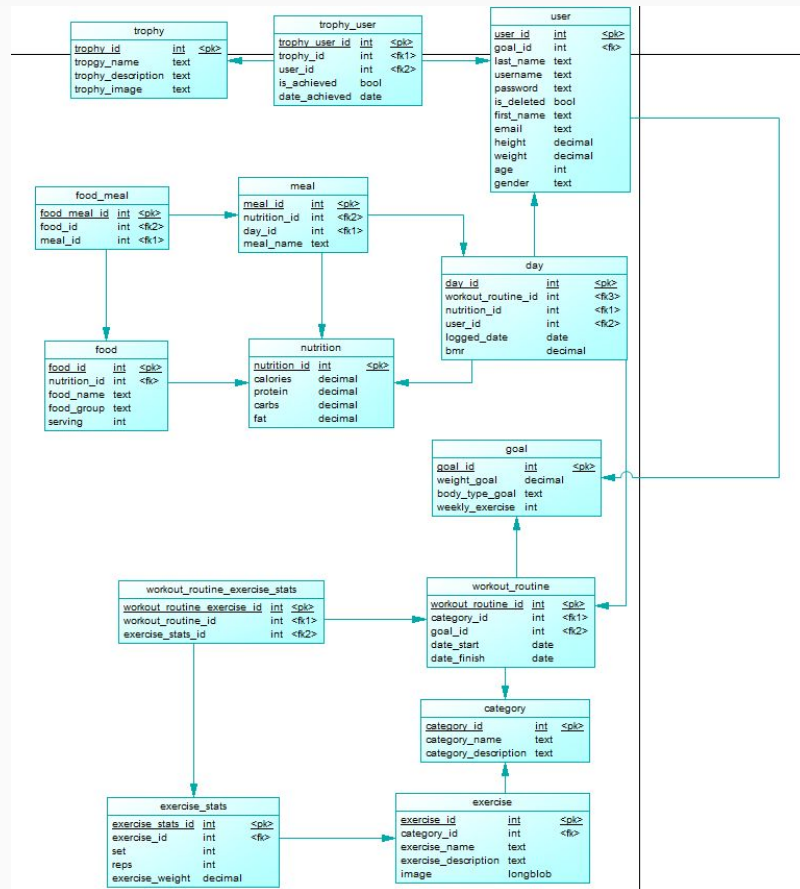
# Konseptualni diagram



# Logički dijagram

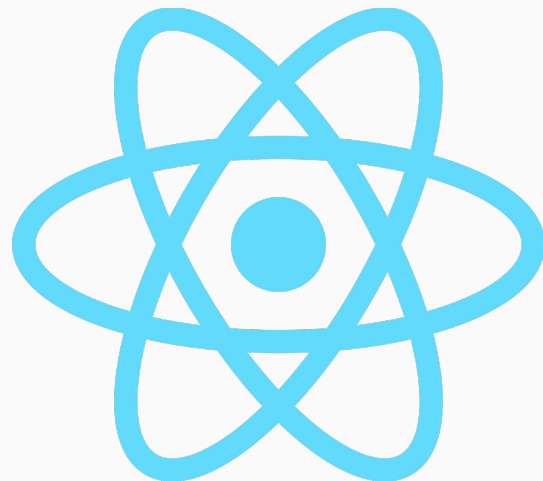


# Fizički dijagram



# Struktura projekta u ReactJs-u

- /src
- /public
- /node-modules
- /package.json
- /package-lock.json
- /assets
- /components
- /pages
- /api
- /tsconfig.json
- /tsconfig.node.json
- /postcss.config.js
- /tailwind.config.js
- /vite.config.ts
- .gitignore





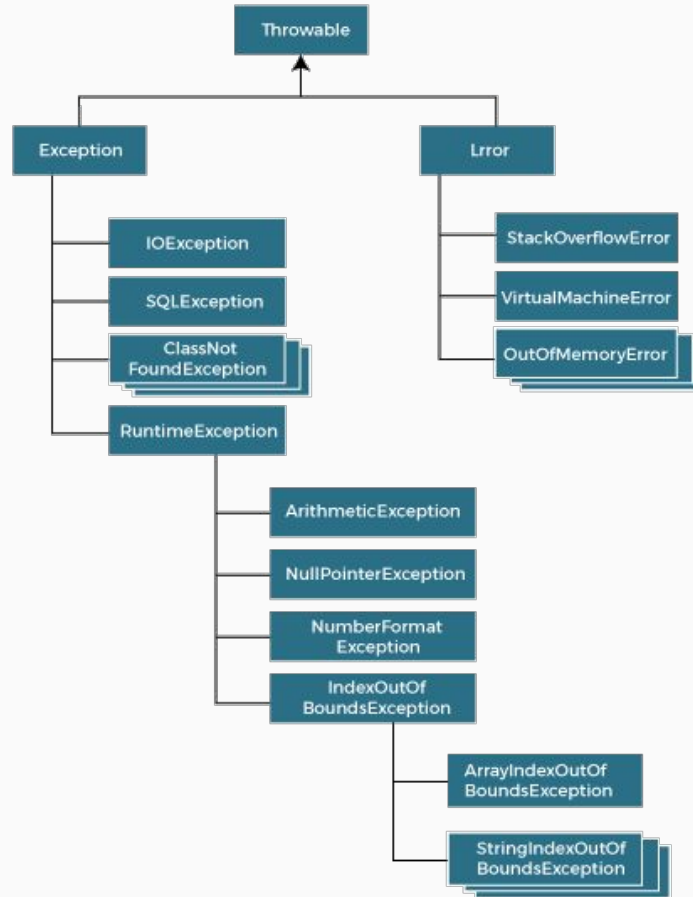
# Struktura projekta u Spring Boot-u

- /config
- /controller
- /converter
- /exception
- /model
- /repository
- /security
- /service
- /util
- /validation



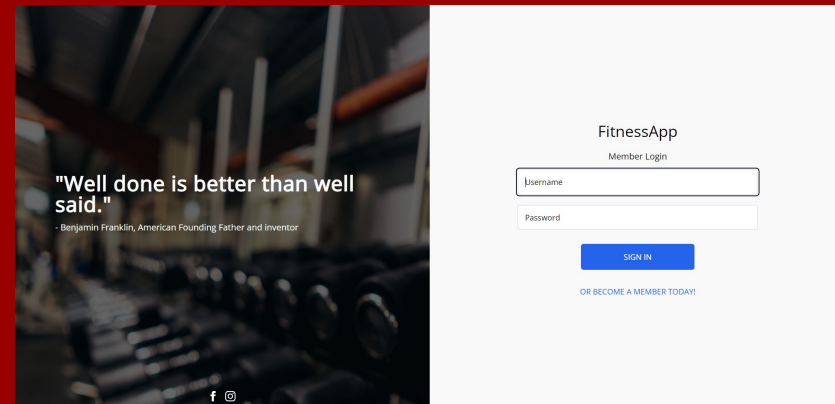
# Obrada izuzetaka

- Obrada izuzetaka predstavlja ključni aspekt robusnosti i pouzdanosti softvera.
- Nepredviđeni ulazi ili nepostojanje resursa kao i raznovrsni faktori mogu izazvati greške koje ugrožavaju normalno funkcionisanje aplikacije.
- Izuzeci su mehanizam koji omogućavaju programima da reaguju na takve neočekivane situacije čime se sprečava nepredvidivo ponašanje i omogućava brz i lak oporavak od potencijalnih problema.
- Java i Spring Boot radni okviri pružaju snažne mehanizme za njihovo efikasno hvatanje i obradu.



# 6. Demonstracija

Potpuna demonstracija  
funkcionalnosti aplikacije



Hvala na pažnji!

