

INSTITUT ZA MATEMATIKU I INFORMATIKU

PRIRODNO-MATEMATIČKI FAKULTET

UNIVERZITET U KRAGUJEVCU



# IgrANNonica

SPECIFIKACIJA DIZAJNA SOFTVERA

## MENTORI

dr Boban Stojanović

Lazar Krstić

Andreja Živić

Filip Bojović

## STUDENTI

Milutin Aleksić

Jefimija Stamenović

Mina Nikolić

Rijalda Bajraktarević

Milan Jovanović

Nikola Stojanović

JUN 2022.

# Sadržaj

Istorija revizije .....	4
1. Uvod .....	5
1.1 Namena dokumenta .....	5
1.2 Opseg projekta .....	5
1.3 Dodatni materijal .....	5
1.4 Definicije, akronimi, skraćenice .....	5
2. Opis proizvoda .....	7
3. Razmatranje dizajna .....	9
3.1 Pretpostavke .....	9
3.2 Ograničenja .....	9
3.3 Sistemsko okruženja .....	9
4. Arhitektura Sistema .....	11
4.1 Opis sistema .....	11
4.2 Obrazloženje Sistema .....	11
5. Dizajn visokog nivoa .....	12
5.1. Konceptualni pogled .....	12
5.2. Fizički pogled .....	13
6. Dizajn niskog nivoa .....	14
6.1. Dizajn baze podataka .....	14
6.1.1. Tabela aspnetusers (tabela korisnika) .....	15
6.1.2. Tabela experiments .....	16
6.1.3. Tabela models .....	16
6.2. Dijagrami klasa .....	17
6.2.1. Klijentski deo .....	17
6.2.2. Serverski deo .....	19
6.3 Dijagrami slučajeva korišćenja .....	23
6.3.1 Slučaj korišćenja - Enkodiranje podataka .....	23
6.3.2 Slučaj korišćenja - Podešavanje mreže .....	24
6.3.3 Slučaj korišćenja - Prikazivanje grafika .....	25
6.4 Dijagrami sekvenci .....	26
6.4.1 Enkodiranje podataka .....	26
6.4.2 Podešavanje mreže .....	27
6.4.3 Prikazivanje grafika .....	28

7. Dizajn korisničkog interfejsa .....	29
7.1 Slike korisničke aplikacije .....	29
7.1.1 Početna strana .....	29
7.1.2 Prijavljivanje .....	30
7.1.3 Registracija .....	30
7.1.4 Stranica za učitavanje i sređivanje podataka .....	31
7.1.5 Stranica za prikazivanje grafika .....	32
7.1.6 Stranica izbor kolona.....	33
7.1.7 Stranica za podešavanje mreže.....	34
7.1.8 Profil i eksperimenti korisnika.....	37

# Istorija revizije

Član tima	Datum izmene	Opis izmene	Verzija
Milutin Aleksić Jefimija Stamenović Rijalda Bajrakterević		Zapocet rad na dokumentu	0.01
Milutin Aleksić Jefimija Stamenović Rijalda Bajrakterević		Dodata nova poglavlja 3 i 4	0.02
Milutin Aleksić		Preradjen tekst i uredjen dokument, dodata poglavlja 5 i 6 i dodate slike	0.1
Jefimija Stamenović Rijalda Bajrakterević		Napravljene slike za naredna poglavlja	0.11
Milutin Aleksić		Dodat tekst i slike za poglavlja 6 i 7	0.2
Milutin Aleksić		Zavrsena dokumentacija	1.0

# 1. Uvod

## 1.1 Namena dokumenta

Namena ovog dokumenta je da pruži detaljan opis projekta IgraNNonica. Aplikacija IgraNNonica vrši obradu podataka na osnovu kojih omogućava kreiranje i treniranje neuronske mreže. U nastavku dokumenta detaljno će biti opisan dizajn i implementacija zahteva navedenih u dokumentu Specifikacija softverskih zahteva.

## 1.2 Opseg projekta

Projekat IgraNNonica je softversko rešenje za generisanje i treniranje neuronskih mreža realizovan kao klijentsko serverska web aplikacija. Aplikacija radi na osnovu unetih podataka, stoga se korisnicima nudi unos i obrada podataka. Nakon završetka obrade podataka korisniku se nudi opcija pravljenja neuronske mreže. Kreirana neuronska mreža se može trenirati, a zatim grafički prikazati njen rad.

## 1.3 Dodatni materijal

- Specifikacija softverskih zahteva projekta IgraNNonica

## 1.4 Definicije, akronimi, skraćenice

Pojam	Objašnjenje
Softver	skup programa koji daje uputstva računaru kako da izvršava određene zadatke
Veštačka inteligencija	podoblast računarstva koja se bavi istraživanjem i razvojem softvera koji ima sposobnost da samostalno zaključuje i donosi odluke na osnovu tih zaključaka
Neuronska mreža	Neuronska mreža je oblik implementacije sistema veštačke inteligencije, koji predstavlja sistem koji se sastoji od određenog broja međusobno povezanih procesora ili čvorova, ili procesnih elemenata koje nazivamo veštačkim neuronima
Mašinsko učenje, ML – Machine Learning	Mašinsko učenje je podoblast veštačke inteligencije čiji je cilj konstruisanje algoritama i računarskih sistema koji su sposobni da se adaptiraju na nove situacije i uče na bazi iskustva
Hardver	fizički deo računara
Web	sistem međusobno povezanih stranica na Internetu

Framework	univerzalno programsko okruženje koje pruža određenu funkcionalnost kao deo veće softverske platforme za olakšanje razvoja softverskih aplikacija, proizvoda i rešenja
Open source	podrazumeva softver čiji je izvorni kod dostupan svim korisnicima koji ga mogu koristiti i menjati
JavaScript	programski jezik koji se prvenstveno koristi za kreiranje interaktivnih web strana na klijentskoj strani
TypeScript	programski jezik otvorenog koda koji je sintaksički nadskup JavaScript-a i kompajlira se u JavaScript
Angular 13	open source TypeScript framework koji služi za kreiranje jednostraničnih aplikacija
Angular Material	Angular framework za dizajn
.NET 6	.NET 6 je besplatan i open-source framework za Windows, Linux i macOS operativne sisteme.
Python	Python je programski jezik koji se često koristi za pravljenje veb sajtova i softvera, automatizaciju zadataka i sprovođenje analize podataka.
SQLite	relaciona baza podataka koja sve podatke i funkcionalnosti čuva u jednoj datoteci
Npm	node.js paket menadžer
HTML5	jezik za struktuiranje i prikazivanje sadržaja na webu (5.verzija standarda)
HTTP	“HyperText Transfer Protocol” – protokol za razmenu informacija na web-u
JSON	“JavaScript Object Notation” – notacija za predstavljanje JavaScript objekata u obliku koji je razumljiv za ljude
IMI	Institut za Matematiku i Informatiku
FAQ	Često postavljena pitanja
WYSIWYG	“What You See Is What You Get” – odnosi se na predstavljanje teksta na ekranu u obliku koji tačno odgovara primenjenom formatiranju
Interface	Abstraktna klasa gde su sve metode abstraktne i nemaju telo, a klasa ne može imati instancu već samo može da se implementira
Controller	Kontroler određuje koji odgovor da pošalje nazad korisniku, kada korisnik uputi zahtev.

## 2. Opis proizvoda

Danas neuronske mreže predstavljaju veoma atraktivnu oblast istraživanja i postoje brojne oblasti u kojima se koriste. Prednost neuronskih mreža je njihova sposobnost da uče koja im daje prirodniji interfejs ka realnom svetu u odnosu na klasične sisteme koji moraju biti programirani. S obzirom da neuronske mreže imaju i mogućnost tolerancije nedostataka, kao recimo nekompletna skup ulaznih podataka, mreža će ipak biti u stanju da da izlaz.

S obzirom da je glavni cilj obrada podataka i učenje na osnovu njih, aplikacija IgrANNonica omogućava korisniku korišćenje bilo kog skupa podataka. Na osnovu datih podataka za treniranje i izabranih podataka za testiranje, dobija se evaluacija koja vraća metrike modela i tabelu rezultata predikcije nad testnim podacima, ukoliko su prethodno izabrani.

U okviru aplikacije, postoje dve vrste korisnika, registrovani i neregistrovani korisnik (gost).

Korisnik je osoba koja je kreirala nalog u aplikaciji, dok gost predstavlja osobu koja nema nalog i ima pristup podrazumevanim opcijama koje se nalaze na sajtu.

U zavisnosti od tipa nude se različite mogućnosti:

1. Gost (neregistrovani korisnik) ima sledeće mogućnosti:

- Unos fajla u csv, json ili xls formatu.
- Pregled podataka u tabeli.
- Promena originalnih vrednosti podataka.
- Promena ukupnog broja podataka, brisanjem željenih redova i/ili kolona.
- Odstranjivanje podataka koji se detektovani kao izuzeci.
- Sređivanje nedostajućih vrednosti: Brisanje nedostajućih vrednosti ili popunjavanje nedostajućih vrednosti.

2. Registrovani korisnik - ovaj korisnik ima sve mogućnosti kao i gost, s tim što su mu na raspolaganju i dodatne mogućnosti:

- Čuvanje eksperimenata i modela u okviru njega
- Upravljanje profilom
- Manipulacija nad postojećim eksperimentima i modelima

Aplikacija mora da zadovolji sledeću listu funkcionalnosti:

1. Obavezne funkcionalnosti

- Učitavanje podataka iz .csv fajla
- Tabelarna vizuelizacija učitanih podataka
- Filtriranje podataka
- Izmene podataka u tabeli

- Prikaz statistike za kolone ( ukupan broj podataka po kolonama, minimum, maksimum, srednja vrednost, medijana, vrednosti za kvantile, korelaciona matrica, broj jedinstvenih vrednosti, broj nedostajućih vrednosti)
- Izbor ulaznih i izlaznih kolona
- Izbor metoda enkodiranja kategorijskih veličina
- Izbor metoda skaliranja numeričkih vrednosti
- Definisanje hiperparametara mreže (broj slojeva, broj neurona po sloju, aktivacione funkcije, broj epoha, metrike, tip problema - klasifikacija/regresija)
- Kreiranje mreže
- Treniranje mreže
- Vizuelno poređenje testnih podataka i rezultata modela

## 2. Poželjne funkcionalnosti

- Čuvanje problema koji su kreirani i rešavani
- Ponovna upotreba sačuvanih problema
- Upravljanje korisnicima
- Različiti načini kreiranja validacionih skupova
- Prikazivanje eventualnih grešaka nastalih prilikom izvršavanja proračuna

## 3. Opcione funkcionalnosti

- Kreiranje više modela u okviru eksperimenta
- Poređenje rezultata različitih modela
- Rad sa nedostajućim vrednostima



## 3. Razmatranje dizajna

### 3.1 Pretpostavke

Kako je IgrANNonica web aplikacija, korisnici mogu da joj pristupe bilo kada, 24 časa dnevno, 7 dana u nedelji. Aplikacija treba da omogući istovremeni pristup više korisnika. Svaki od korisnika ima mogućnost izvršavanja prethodno navedenih funkcionalnosti, u zavisnosti od tipa korisnika.

Da bi korisnik bio u mogućnosti da koristi aplikaciju, potrebno je da poseduje:

- PC ili laptop računar
- stabilnu internet konekciju
- Web pretraživač (Google Chrome, Firefox, Edge, Opera)

U slučaju kvara ili otkaza web servera, nestanka struje na serveru ili kod korisnika, neaktivan priključak na internet (server ili korisnik) pristup aplikaciji neće biti moguć.

Korisnici aplikacije ne moraju biti eksperti u oblasti informatike ali se podrazumevavosnovni nivo znanja u korišćenju računara. Korisnici aplikacije moraju biti dovoljno stručna lica u oblasti neuronskih mreža kako bi u potpunosti iskoristili potencijal aplikacije.

Klijentski deo aplikacije je predviđen da se pokreće na desktop i prenosivim računarima. Iako se možda može koristiti sa pametnim telefonom, tabletom, aplikacija subjektivno neće pružiti isti doživljaj i lakoću korišćenja kao što je to na uređajima predviđenim za njeno korišćenje.

### 3.2 Ograničenja

Zahtev klijenta je da IgrANNonica razvijena upotrebom savremenih Web tehnologija. Stoga, projekat prate neka tehnička ograničenja.

Projektna ograničenja:

- Za front-end deo aplikacije treba koristiti Angular
- Za čuvanje podataka koristiti MySQL bazu
- Serverski deo aplikacije razviti sa .NET 6 I Python-om

### 3.3 Sistemsko okruženja

Aplikacija IgrANNonica će biti postavljena na IMI serveru koji radi na Linux Ubuntu operativnom sistemu, dok Python mikroservis ( u daljem tekstu ML) i baza podataka mogu ali ne moraju da se nalaze na drugom serveru.

Kako bi aplikacija radila, potrebno je da na serveru bude instalirano:

- Angular: 13.2.5
- Node: 16.14.0
- Package Manager: npm 8.3.1
- NET: 6.0
- MySql: 8.0
- Python: 3.6.2
- Python biblioteke :
  - Pandas, threading, socket, json, sklearn, numpy, requests, time, pandas.api.types, os, shutil, io, random, matplotlib, tensorflow, scipy

# 4. Arhitektura Sistema

## 4.1 Opis sistema

Web aplikacija se sastoji iz klijentskog i serverskog dela. Komunikacija klijentskog dela aplikacije i serverskog dela aplikacije odvija se preko HTTP protokola ili preko soketa, a podaci se razmenjuju u JSON formatu. Takođe, podaci o korisniku i njegovim eksperimentima i modelima se skladište u MySQL bazi podataka.

Klijentski deo aplikacije se sastoji iz više komponenti koje međusobno komuniciraju, ali i preko kojih se vrši komunikacija sa serverskim delom aplikacije putem API zahteva koji koriste HTTP protokol.

Serverski deo aplikacije se sastoji iz kontrolera i dva sloja:

- Sloj za pristup podacima
- Sloj za komunikaciju sa ML serverom

Komunikacije klijent-server i server-ML se odvijaju putem interneta i podaci se razmenjuju u JSON formatu.

## 4.2 Obrazloženje Sistema

Prezentacioni sloj (UI) predstavlja korisnički interfejs i sastoji se od komponenti čije metode prihvataju podatke od korisnika. Implementirane su korišćenjem Angular framework-a. Komponente komuniciraju sa kontrolerima preko HTTP protokola i web soketa za razmenu podataka tokom treniranja.

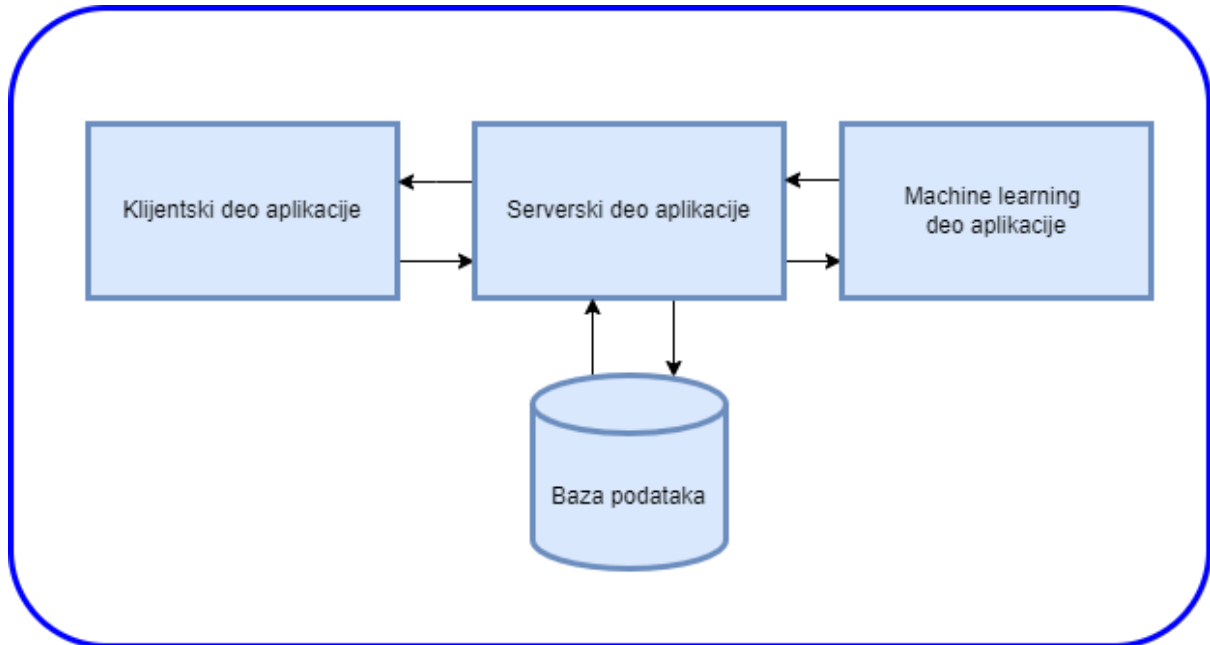
Kontroleri obrađuju pristigle zahteve sa klijentskog dela aplikacije. Ukoliko je potreban pristup bazi (prijavljivanje, registracija ili upravljanje profilom), koristi se sloj za pristup podacima koji dolazi do potrebnih podataka. Ovaj sloj vrši čitanje, izmenu, promenu i brisanje podataka.

Ukoliko je zahtev vezan za obradu podataka ili kreiranje/treniranje neuronske mreže, koristi se sloj za komunikaciju sa ML serverom koji pristigle podatke obrađuje po potrebi i šalje na ML server putem soketa na izvršavanje.

Na ML serveru se za svakog klijenta kreira zasebna nit koja izvršava sve njegove zahteve. Zahtevi se izvršavaju tako što najpre stigne poruka preko soketa koja "kaže" koja komanda treba da se izvrši a odmah zatim i podaci neophodni za izvršavanje te komande. Rezultat izvršavanja (ili poruka greške ukoliko je došlo do greške) se vraćaju putem soketa na server.

## 5. Dizajn visokog nivoa

### 5.1. Konceptualni pogled



Slika 1: Konceptualni prikaz sistema

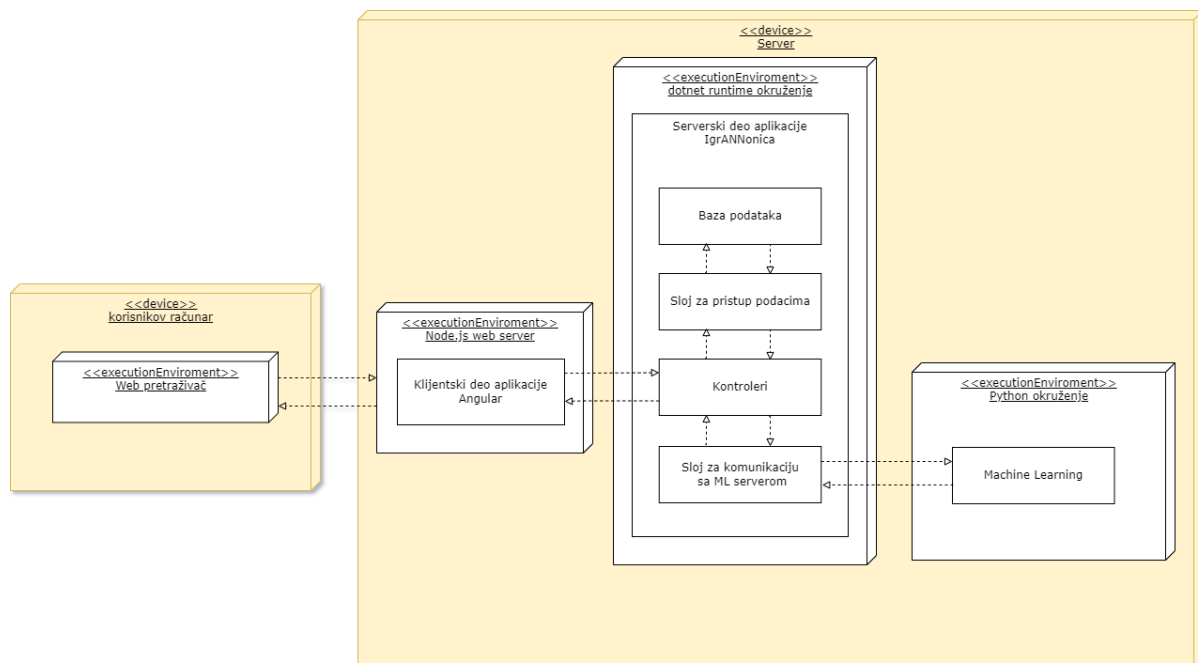
Aplikaciju IgrANNonica čine četiri komponente:

- klijentski deo aplikacije
- web server
- ML servis
- baza podataka

Klijentski deo aplikacije se izvršava u web pretraživaču na računaru korisnika.

Web server se nalazi na IMI serveru i njegova uloga je da komunicira sa ML servisom i bazom podataka, kao i da odgovara na klijentske zahteve (prijavljivanje, registrovanje, učitavanje fajla sa podacima, izmena podataka, kreiranje i treniranje mreže ...)

## 5.2. Fizički pogled



Slika 2: Fizički prikaz sistema

Na serveru na kome se pokreće IgrANNonica mora imati instaliran Node.js, .NET runtime i python okruženja.

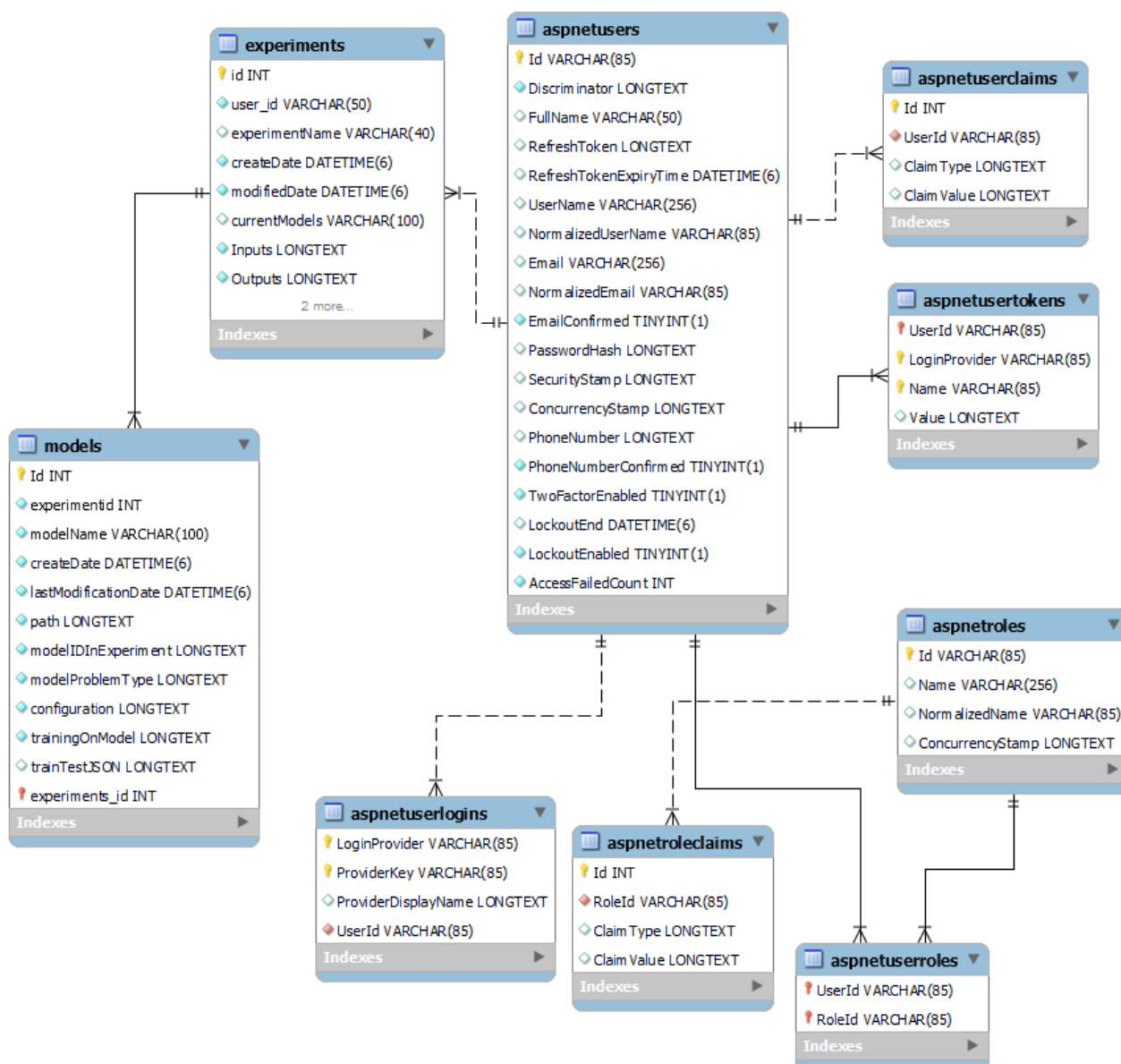
Za korišćenje aplikacije na računaru korisnika, neophodno je da korisnik ima računar sa instaliranim nekim od standardnih modernih web pretraživača i stabilnu internet konekciju.

## 6. Dizajn niskog nivoa

### 6.1. Dizajn baze podataka

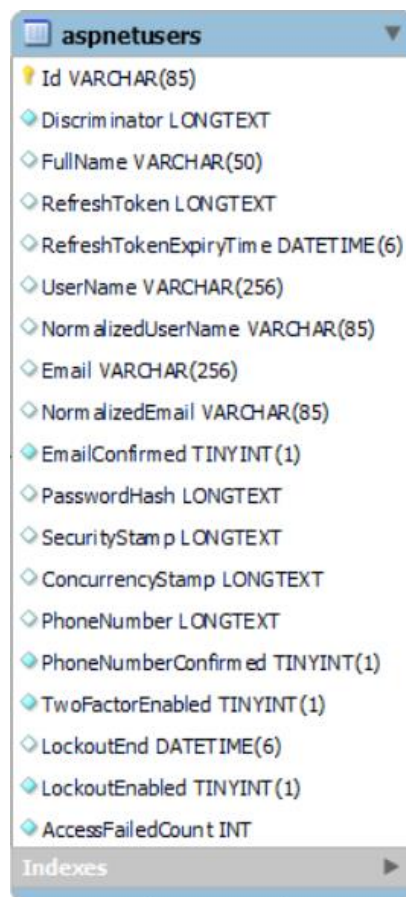
U bazi podataka se čuvaju podaci o:

- Korisnicima,
- Eksperimentima i modelima.



Slika 3: Prikaz tabela i njihovih veza

#### 6.1.1. Tabela aspnetusers (tabela korisnika)



The screenshot shows the structure of the 'aspnetusers' table. The columns are listed with their data types and lengths. The 'Id' column is highlighted with a yellow lightning bolt icon, indicating it is the primary key. Other columns include 'Discriminator', 'FullName', 'RefreshToken', 'RefreshTokenExpiryTime', 'UserName', 'NormalizedUserName', 'Email', 'NormalizedEmail', 'EmailConfirmed', 'PasswordHash', 'SecurityStamp', 'ConcurrencyStamp', 'PhoneNumber', 'PhoneNumberConfirmed', 'TwoFactorEnabled', 'LockoutEnd', 'LockoutEnabled', and 'AccessFailedCount'.

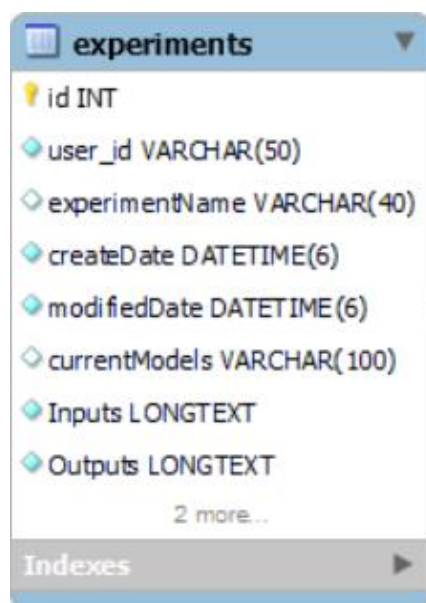
Column Name	Data Type
Id	VARCHAR(85)
Discriminator	LONGTEXT
FullName	VARCHAR(50)
RefreshToken	LONGTEXT
RefreshTokenExpiryTime	DATETIME(6)
UserName	VARCHAR(256)
NormalizedUserName	VARCHAR(85)
Email	VARCHAR(256)
NormalizedEmail	VARCHAR(85)
EmailConfirmed	TINYINT(1)
PasswordHash	LONGTEXT
SecurityStamp	LONGTEXT
ConcurrencyStamp	LONGTEXT
PhoneNumber	LONGTEXT
PhoneNumberConfirmed	TINYINT(1)
TwoFactorEnabled	TINYINT(1)
LockoutEnd	DATETIME(6)
LockoutEnabled	TINYINT(1)
AccessFailedCount	INT

Indexes

Slika 4: Prikaz tabele korisnika

Tabela korisnika sadrži podatke o registrovanim korisnicima pri čemu korisničko ime i email adresa moraju biti jedinstveni. U tabeli se čuva i *refresh token* koji služi za dobijanje novog tokena nakon što istekne.

### 6.1.2. Tabela experiments

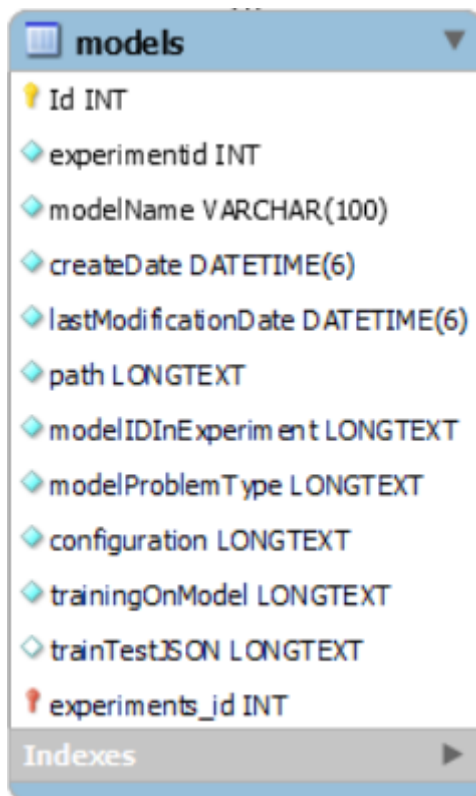


id	user_id	experimentName	createDate	modifiedDate	currentModels	Inputs	Outputs
INT	VARCHAR(50)	VARCHAR(40)	DATETIME(6)	DATETIME(6)	VARCHAR(100)	LONGTEXT	LONGTEXT
2 more...							
Indexes ▶							

Slika 5: Prikaz tabele eksperimenata

U tabeli experiments se čuvaju podaci o sačuvanim eksperimentima korisnika. Kolone “Inputs” i “Outputs” sadrže niz kolona koje se koriste kao ulaz i izlaz u eksperimentu i ovi podaci su zajednički za svaki model iz datog eksperimenta.

### 6.1.3. Tabela models



Id	experimentid	modelName	createDate	lastModificationDate	path	modelIDInExperiment	modelProblemType	configuration	trainingOnModel	trainTestJSON	experiments_id
INT	INT	VARCHAR(100)	DATETIME(6)	DATETIME(6)	LONGTEXT	LONGTEXT	LONGTEXT	LONGTEXT	LONGTEXT	LONGTEXT	INT
Indexes ▶											

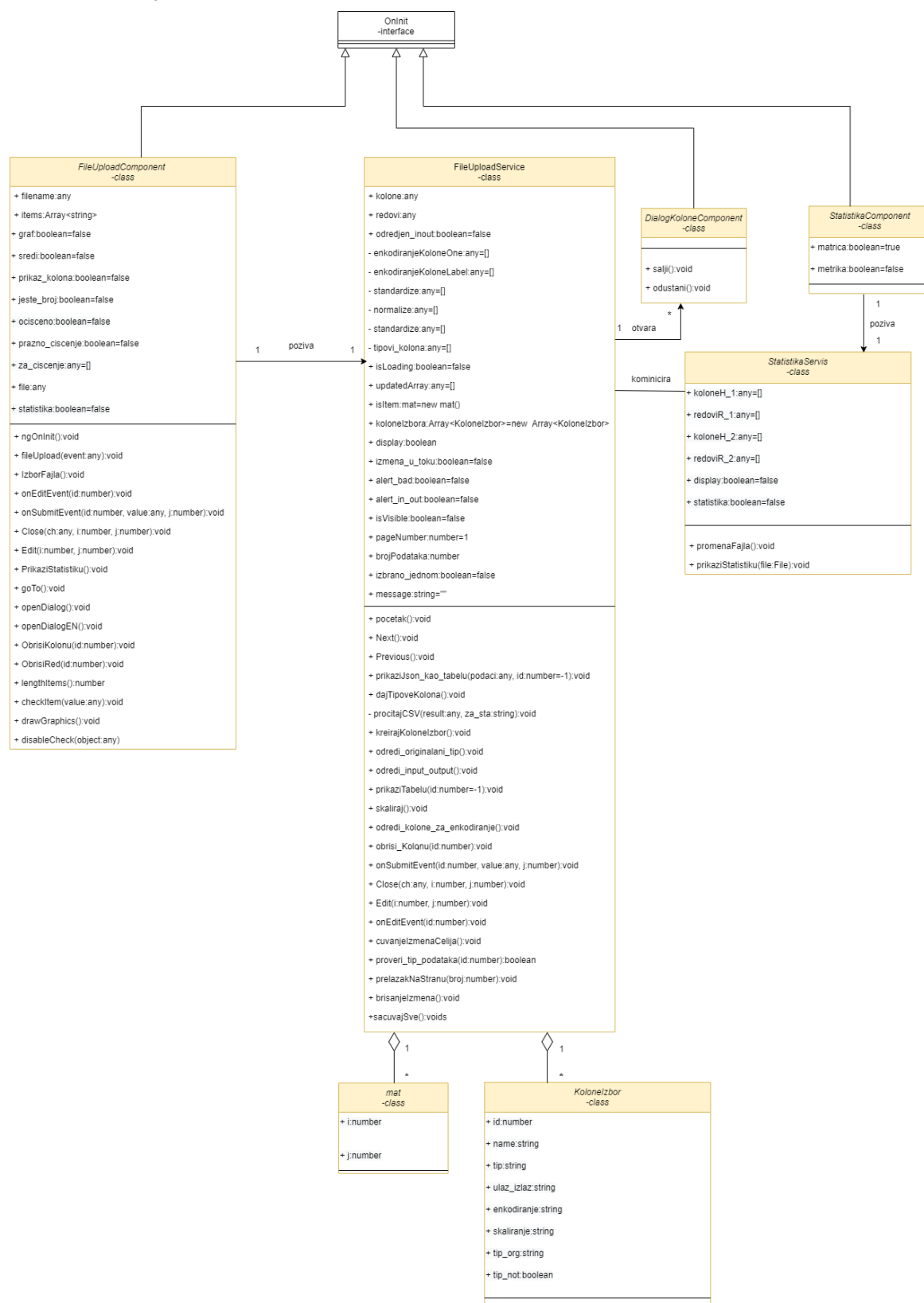
Slika 6: Prikaz tabele modela



U tabeli models se čuvaju podaci o sačuvanim modelima korisnika.

## 6.2. Dijagrami klasa

### 6.2.1. Klijentski deo

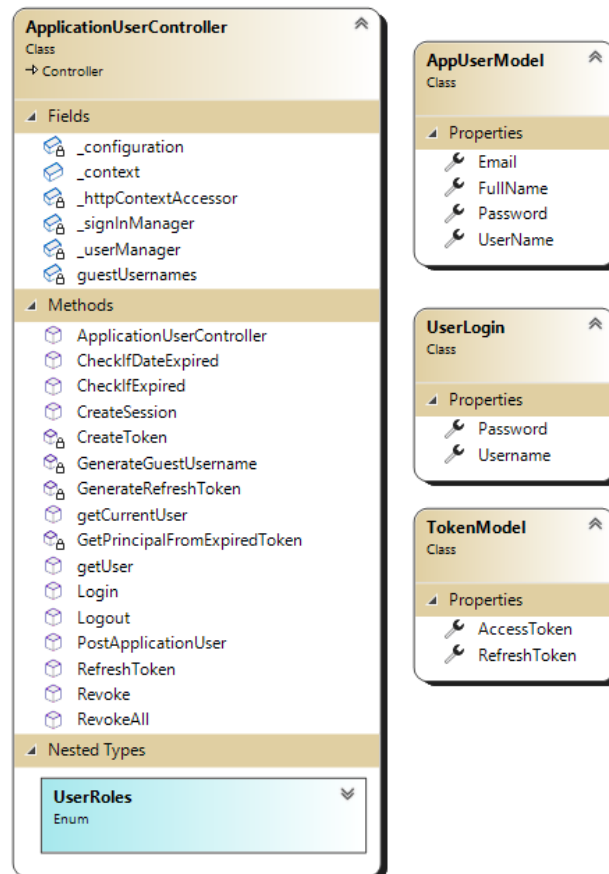


Slika 7: Prikaz komponenti za obradu podataka

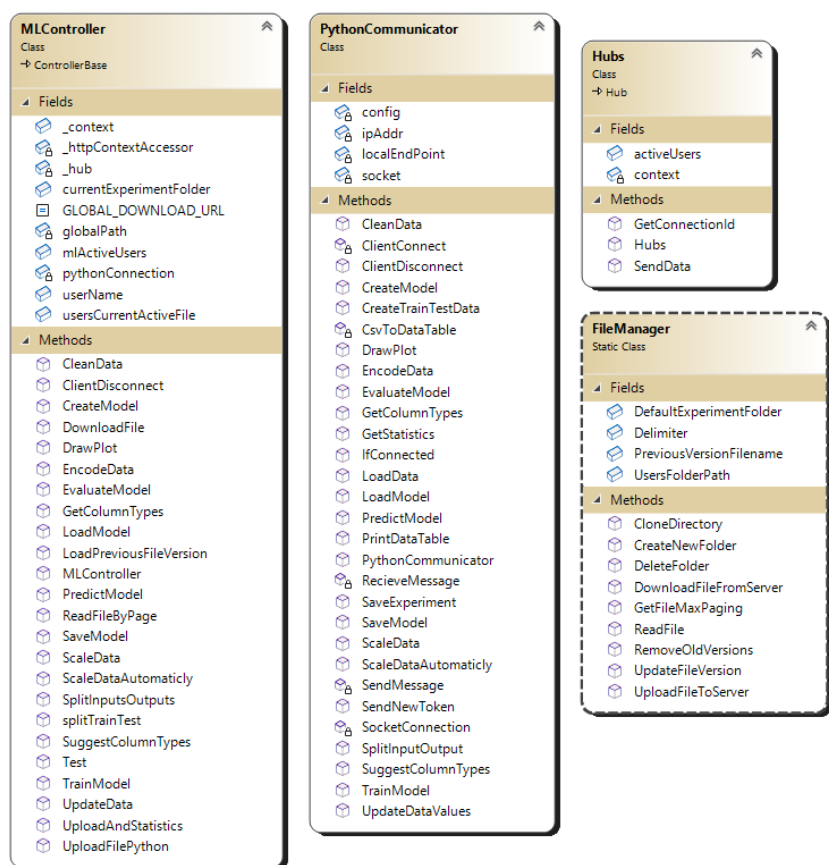


## 6.2.2. Serverski deo

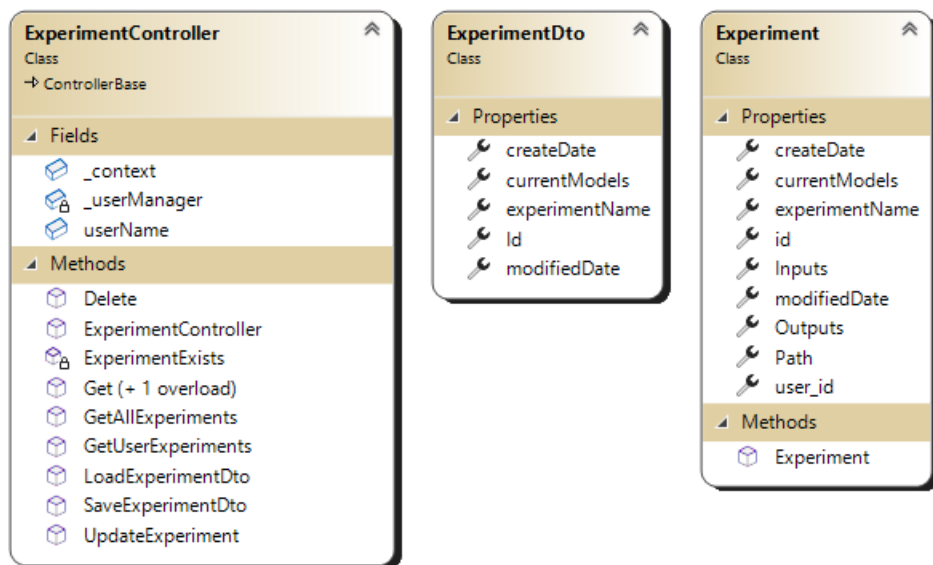
### 6.2.2.1 Kontroleri



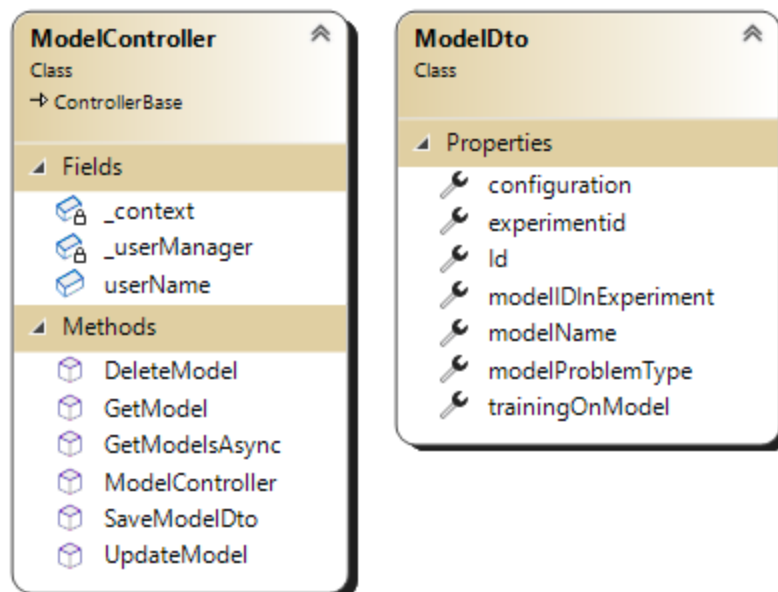
Slika 9: Prikaz kontrolera za prijavljivanje i registraciju



Slika 10: Prikaz kontrolera i pomoćnih klasa za komunikaciju sa ML serverom



Slika 11: Prikaz kontrolera i pomoćnih klasa za rad sa eksperimentima

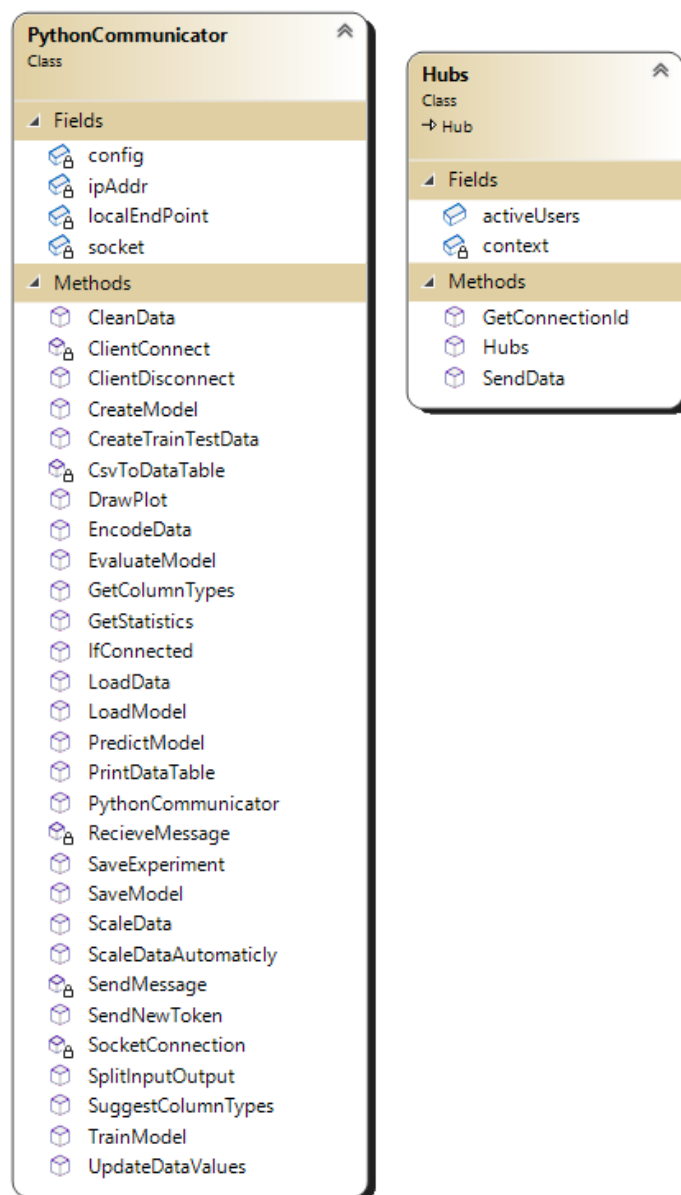


Slika 12: Prikaz kontrolera i pomoćne klase za rad sa modelima

#### 6.2.2.2 Sloj za pristup podacima

Aplikacija koristi code-first pristup što znači da se svaka klasa putem migracije generiše kao tabela, a njene osobine (eng property) kao kolone. Bazi se pristupa putem Controller-a i DbContext-a koji je povezan sa bazom podataka.

### 6.2.2.3 Sloj za komunikaciju sa ML serverom



Slika 13: Klasa PythonCommunicator

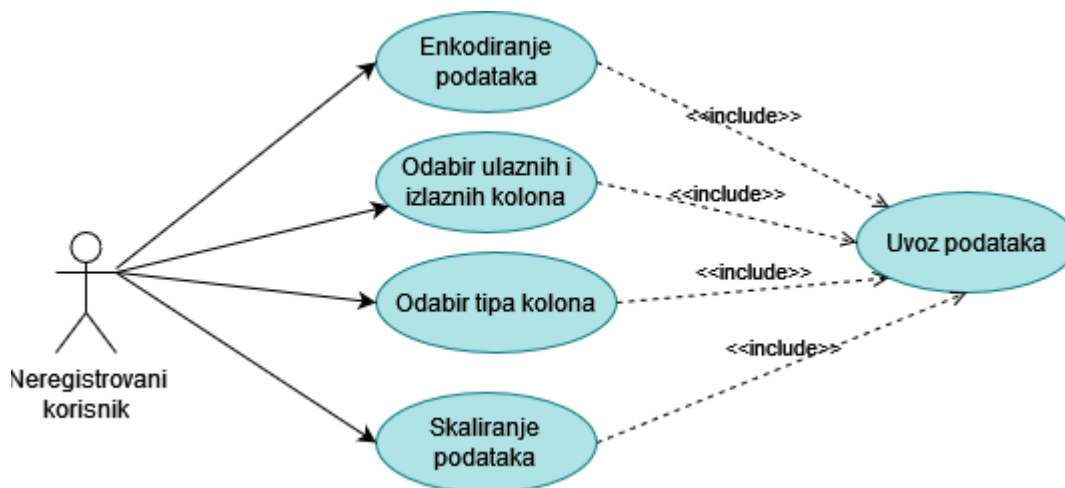
PythonCommunicator klasa se poziva od strane MLController-a i zadužena je prosleđivanje neophodnih podataka ka i od ML server-a putem soketa.

Pomoćna klasa Hubs se koristi tokom treniranja modela (TrainModel funkcija) i zadužena je da prenosi podatke tokom treniranja u realnom vremenu putem web soketa korišćenjem SingalR biblioteke.

## 6.3 Dijagrami slučajeja korišćenja

U nastavku će biti opisano nekoliko načina upotrebe aplikacije.

### 6.3.1 Slučaj korišćenja - Enkodiranje podataka



#### Kratak opis:

Korisnik enkodira podatke jednom od ponuđenih opcija: Label ili OneHot.

#### Opis slučaja korišćenja:

Pred korisnikom se nalazi tabela kolona gde može da odabare različite opcije neophodne za dalji rad. Korisnik nije prinuđen da enkodira kolone koje ne želi da koristi.

#### Glavni tok događaja:

Korisniku se prikazuje tabela kolona, za svaku kolonu korisnik bira da li želi i na koji način da je enkodira.

Ponuđene opcije su: Label encoding i OneHot enkodiranje.

Klikom na padajući meni korisnik bira jednu od pomenutih opcija za svaku kolonu koju želi da enkodira.

Kada je korisnik završio biranjem načina enkodiranja i ostalim podešavanjima u tabeli, pritiskom na dugme "sačuvaj izmene" korisniku iskače obaveštenje da su podešavanja konačna i ne mogu da se promene u toku rada. Korisnik će morati da učitava ceo fajl ponovo i ponovi ceo proces ukoliko želi izmene.

#### Definisanje uslova:

Preduslov: Korisnik mora odabrati iz tabele da li se kolona koristi za ulaz ili izlaz.

#### Alternativni tok događaja:

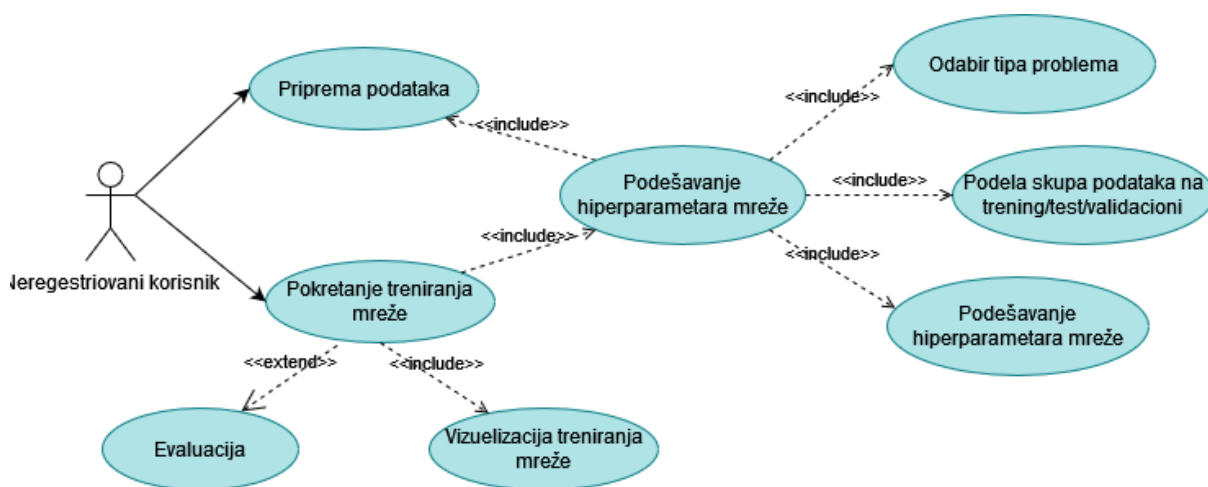
Enkodiranje kolona može da se preskoči.

Ukoliko je korisnik odabrao nenumeričku kolonu kao ulaz ili izlaz a nije je enkodirao, neophodno je ispisati grešku korisniku i onemogućiti dalji rad dok korisnik ne enkodira kolonu ili je ne izbaci iz ulaza/izlaza.

### 6.3.2 Slučaj korišćenja - Podešavanje mreže

#### Kratak opis:

Korisnik podešava parametre neuronske mreže odabirom ponuđenih podešavanja.



#### Opis slučaja korišćenja:

Korisniku se na sredini strane nalazi vizuelni prikaz modela koji pravi i prikaz se ažurira u realnom vremenu dok korisnik dodaje ili briše slojeve i neurone.

#### Glavni tok događaja:

Na vrhu strane korisnik mora da odabere podešavanja mreže. Neka podešavanja su unapred predložena u zavisnosti od toga da li je korisnik odabrao regresioni ili klasifikacioni problem.

Opcije koje korisnik ima su:

- aktivaciona funkcija za skrivene slojeve,
- aktivaciona funkcija za izlazni sloj,
- loss funkcija i
- odabir jedne ili više metrika

Klikom na padajući meni korisnik bira podešavanja parametara svoje mreže. Moguće je odabarati nijednu, jednu ili više metrika. Ispod ovih podešavanja se nalazi prazan deo ekrana gde korisnik vizuelno vidi kako njegova mreža izgleda.



Klikom na "+" korisnik dodaje novi sloj u svoju mrežu sa jednim neuronom, iznad svakog sloja neurona se nalazi "+" i "-" za povećanje ili smanjivanje broja neurona u tom sloju.

Kada je korisnik zadovoljan sa svojom mrežom, pritiskom na dugme u gornjem desnom uglu, korisnik zaključava mrežu.

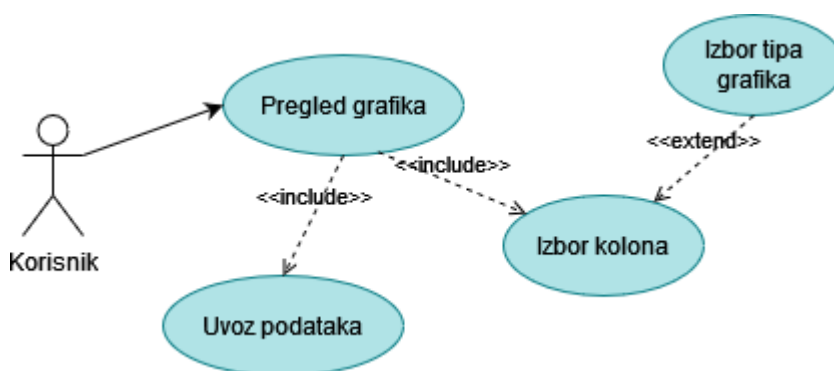
U donjem delu ekrana može da podesi podešavanja za treniranje mreže (omogućeno pre i posle zaključavanja podešavanja mreže).

#### Definisanje uslova:

Preduslov: Korisnik mora da podeli podatke na podatke za treniranje, validaciju i testiranje.

Postuslov: Korisniku se omogućuje početak treniranja mreže

### 6.3.3 Slučaj korišćenja - Prikazivanje grafika



#### Kratak opis:

Korisnik bira jednu ili dve kolone za koje želi da prikaze graf

#### Opis slučaja korišćenja:

Korisnik ima mogućnost da odabere jednu kolonu i vrstu grafa koji može da prikaze (vrsta grafa zavisi od tipa kolone) ili može da odabere dve kolone i automatski prikaže najbolju vrstu grafa za upoređivanje tih kolona

#### Glavni tok događaja:

Pred korisnikom se nalazi tabela sa nazivima kolona na levoj strani ekrana, odabirom jedne kolone, korisniku se omogućuje da odabere tip grafika koji želi da prikaze za odabranu kolonu, nalazi se u padajućem meniju iznad tabele sa kolonama. Klikom na dugme "Iscrtaj grafik" na ekranu se prikazuje grafik.

Odabirom dve kolone, onemogućuje se odabir tipa grafika i pritiskom na dugme "prikaži grafik" prikazuje se grafik upoređivanja odabranih kolona. Nije moguće odabrati više od dve kolone istovremeno

#### Definisanje uslova:

Preduslov: Korisnik mora da učitaj fajl sa podacima

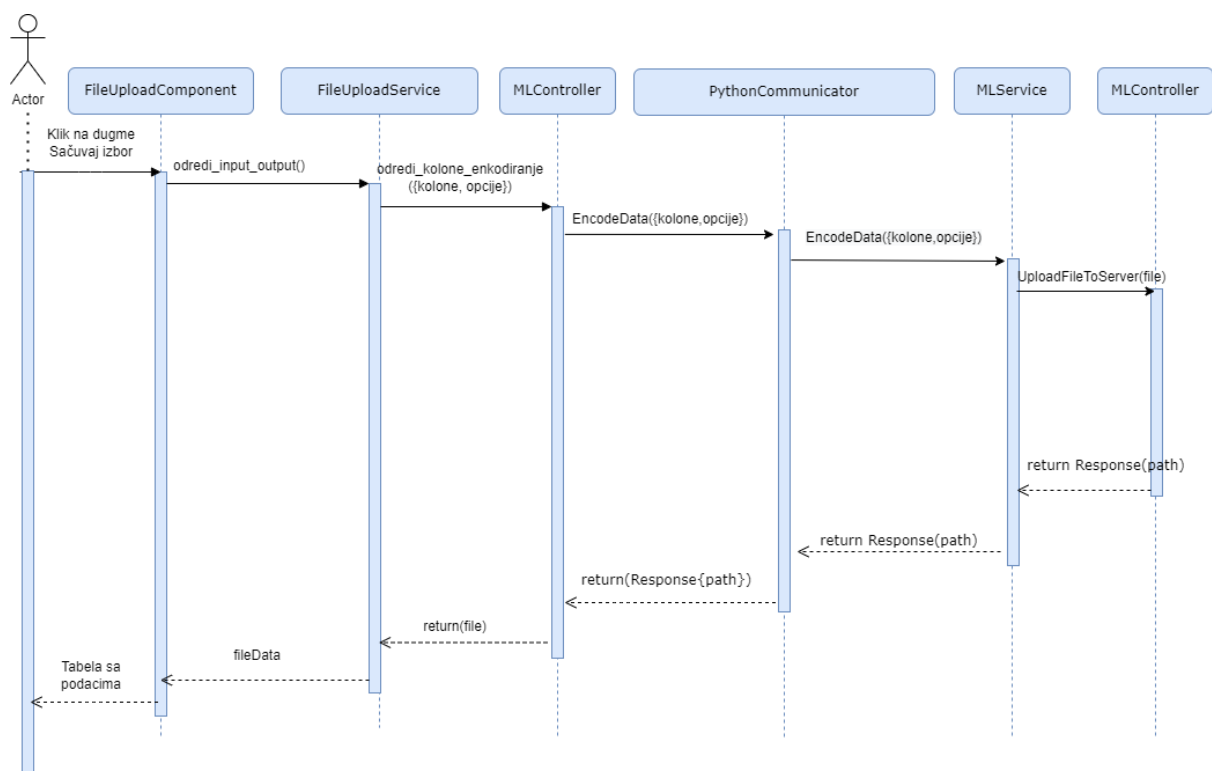
#### Alternativni tok događaja:

Korak prikazivanja grafika može da se preskoči

## 6.4 Dijagrami sekvenci

U ovom poglavlju biće prikazani dijagrami sekvenci za prethodno slučajeve korišćenja opisane u prethodnom poglavlju.

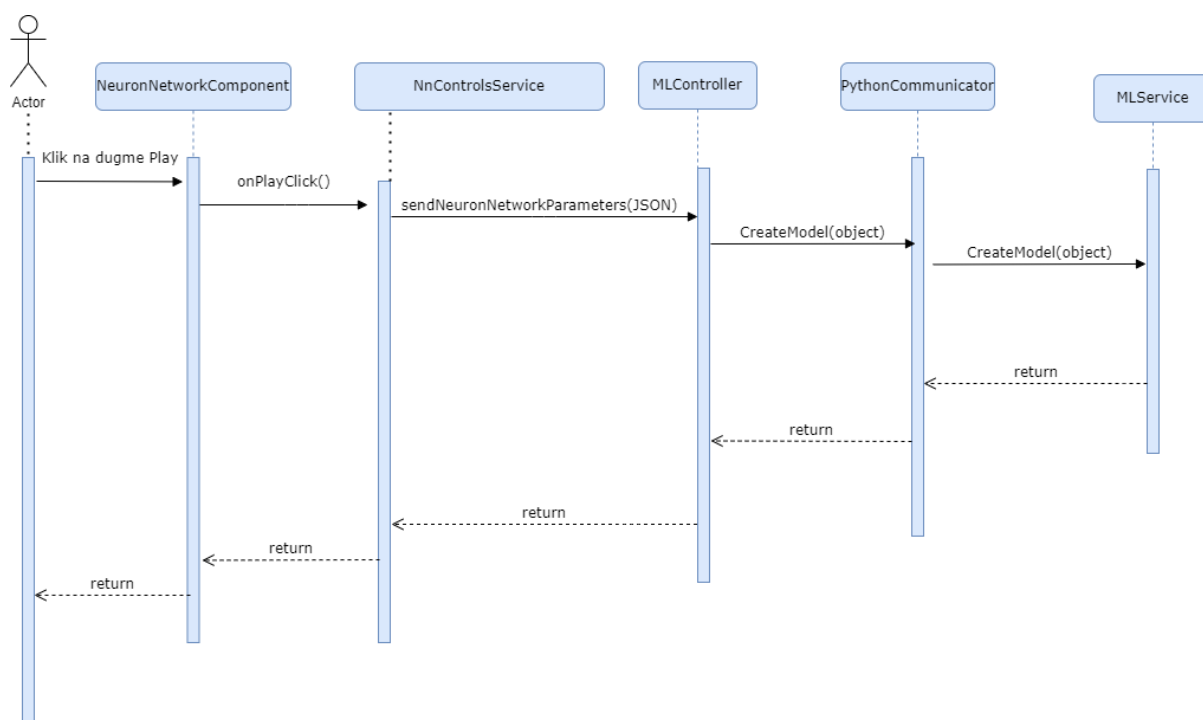
### 6.4.1 Enkodiranje podataka



Slika 14: Dijagram sekvenci – Enkodiranje podataka

Na slici iznad vidimo tok poziva za enkodiranje podataka. Klikom na dugme “Sačuvaj Izbor” šalje se zahtev ka MLController-u koji prosleđuje podatke ML servisu pomoću PythonCommunicator klase. ML servis nakon što enkodira podatke, šalje zahtev MLController-u za učitavanje fajla na server. MLController kao odgovor na zahtev vraća ML serveru putanju gde je fajl sačuvan na serveru. Prvi poziv MLController-a klijentu vraća novu verziju podaka.

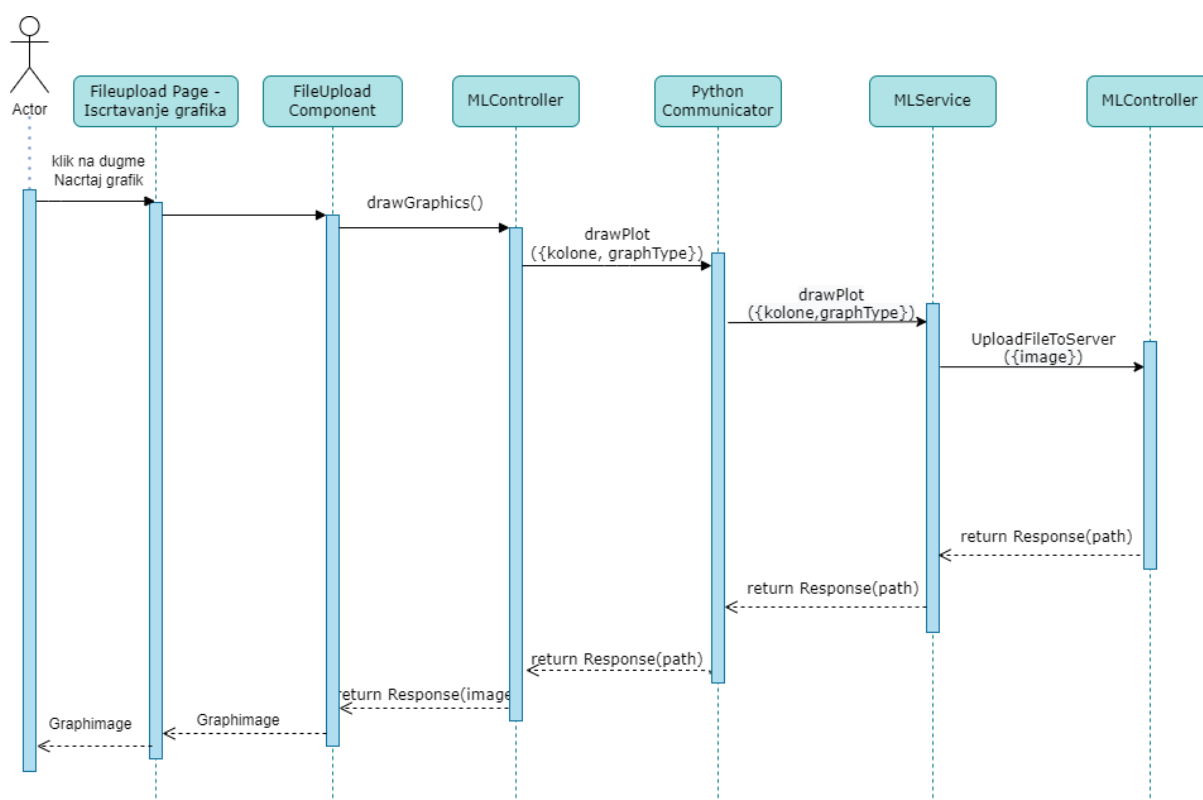
### 6.4.2 Podešavanje mreže



Slika 15: Dijagram sekvenci – Podešavanje mreže

Na slici iznad vidimo tok poziva za podešavanje mreže. Zahtev se šalje putem HTTP protokola MLController-u koji prosleđuje podatke ML servisu pomoću PythonCommunicator klase koja komunicira putem soketa sa ML servisom. Nakon što ML servis uspešno kreira model, vraća poruku o uspešno izvršenom zahtevu.

### 6.4.3 Prikazivanje grafika



Slika 16: Dijagram sekvenci – Prikazivanje grafika

Na slici iznad vidimo tok poziva za iscrtavanje grafika. Nakon što je korisnik odabrao za koju kolonu ili kolone želi da prikaže grafik i klikne na dugme “Iscrtaj grafik”, naziv kolone se šalje FileUpload komponenti koja šalje podatke putem HTTP post zahteva MLController-u na server. MLController prosleđuje podatke na ML server korišćenjem PythonCommunicator klase (sloj za komunikaciju sa ML serverom). Na ML serveru se iscrtava grafik i čuva u lokalu kao slika, zatim se šalje zahtev MLController-u (UploadFileToServer funkcija) za prebacivanje slike na server. MLController kao odgovor na zahtev vraća ML serveru putanju gde je slika sačuvana na serveru. Prvi poziv MLController-a klijentu vraća sliku.

## 7. Dizajn korisničkog interfejsa

### 7.1 Slike korisničke aplikacije

#### 7.1.1 Početna strana



Slika 17: Početna strana

Ulaskom na platformu se prikazuje početna strana. Korisnik je automatski prijavljen kao gost.

Na vrhu svake strane se nalazi navigacioni meni, na levoj strani navigacionog menija se nalazi logo firme, pritiskom na logo firme korisnik se vraća na početnu stranu, na desnoj strani se nalaze "Započnite", "Registruj se" i "Prijavi se". Ukoliko je korisnik prijavljen "Registruj se" i "Prijavi se" se zamenjuju sa karticom koja sadrži korisničko ime i padajući meni koji sadrži dugme koje vodi do profilne strane i dugme za odjavljivanje (pri čemu se korisnik prijavljuje kao gost).

Na sredini strane se nalazi naziv aplikacije i dugme sa strelicom koje je ekvivalentno "Započnite" dugmetu, vodi na stranicu za pripremu podataka.

### 7.1.2 Prijavljivanje

The login form, titled "Prijava", features a back arrow icon at the top left. It contains two input fields: "Unesite korisničko ime" (username) and "Unesite lozinku" (password), each preceded by a corresponding icon (person and lock). Below these fields is a link "Nemate nalog? Registrujte se" (Don't have an account? Register) and a large right-pointing arrow button.

Slika 18: Forma za prijavljivanje

Na slici iznad je prikazana stranica za prijavljivanje. Potrebno je da korisnik unese svoje korisničko ime i lozinku.

### 7.1.3 Registracija

The registration form, titled "Registracija", has a back arrow icon at the top left. It includes four input fields: "Unesite ime i prezime" (name and surname), "Unesite e-mail adresu" (email address), "Jefi" (first name), and a password field with a "Potvrdite lozinku" (confirm password) label. A link "Imate nalog? Prijavite se" (Have an account? Log in) is at the bottom left. The right side of the form features a large right-pointing arrow button.

The second screenshot shows the form after an error. The email field now contains "pera" and displays the message "Uneli ste nepostojeći e-mail!". The first name field contains "Per". Below the password fields, a message states: "Lozinka mora da sadrži: veliko slovo, malo slovo i broj i mora biti dužine 8-24 karaktera!" and "Potvrđena i uneta lozinka se razlikuju!".

Slika 19: Forma za registraciju

Da bi se korisnik uspešno registrovao neophodno je da unese korisničko ime, validan format email adrese i lozinku.

Korisnik će biti obavešten ukoliko je uneo već postojeće korisničko ime, nevalidan format email adrese, imena ili lozinke (pogledati sliku). Ime sme sadržati samo slova.

Nakon uspešne registracije korisnik se automatski prijavljuje na svoj nalog.

#### 7.1.4 Stranica za učitavanje i sređivanje podataka

Unnamed: 0	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration
0	18.0	8	307.0	130.0	3504.0	12.0
1	15.0	8	350.0	165.0	3693.0	11.5
2	18.0	8	318.0	150.0	3436.0	11.0
3	16.0	8	304.0	150.0	3433.0	12.0
4	17.0	8	302.0	140.0	3449.0	10.5
5	15.0	8	429.0	198.0	4341.0	10.0
6	14.0	8	454.0	220.0	4354.0	9.0

Slika 20: Tabela podataka

Unnamed: 0	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration
0	18.0	8	318.0	150.0	3436.0	11.0
1	15.0	8	304.0	150.0	3433.0	12.0
2	17.0	8	302.0	140.0	3449.0	10.5
3	15.0	8	429.0	198.0	4341.0	10.0
4	14.0	8	454.0	220.0	4354.0	9.0

	Unnamed: 0	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration
Unnamed: 0	1.0	0.58647	-0.36034	-0.38714	-0.4229	-0.32	-0.32
MPG	0.58647	1.0	-0.77762	-0.80513	-0.77843	-0.83	-0.83
Cylinders	-0.36034	-0.77762	1.0	0.95082	0.84298	0.89	0.89
Displacement	-0.38714	-0.80513	0.95082	1.0	0.89726	0.93	0.93
Horsepower	-0.4229	-0.77843	0.84298	0.89726	1.0	0.86	0.86
Weight	-0.32184	-0.83224	0.89753	0.93299	0.86454	1.0	1.0
Acceleration	0.29086	0.42333	-0.50468	-0.5438	-0.6892	-0.41	-0.41
Model Year	0.99678	0.58054	-0.34565	-0.36986	-0.41636	-0.36	-0.36
Origin	0.20059	0.56521	-0.56893	-0.61454	-0.45517	-0.56	-0.56

Slika 21: Matrica korelacije

Nakon što je korisnik učitao podatke pritiskom na dugme “Izbor fajla”, prikazuju se tabela podataka i ispod nje matrica korelacije ili tabela sa statistikama kolona (U zavisnosti šta korisnik želi da prikaže).

Sa desne strane se nalaze opcije za čišćenje podataka koje korisnik treba da odabere za svaku kolonu. Pritiskom na ćeliju u tabeli sa podacima, korisnik može da promeni vrednost te ćelije.

Pri vrhu strane se nalaze kartice za naredne korake u procesu sređivanja i vizuelizacije podataka.



### 7.1.5 Stranica za prikazivanje grafika



Slika 22: Prikaz različitih vrsta grafika



Dolaskom na ovu karticu, korisniku sa leve strane se nalaze nazivi kolona učitanoog fajla.

Korisniku je omogućeno da odabere maksimalno dve (2) kolone istovremeno. Odabirom jedne kolone i vrste grafika (nalazi se iznad “Nacrtaj grafik” dugmeta) prikazuje se grafik statistike izabrane kolone. Ukoliko korisnik odabere dve kolone, iscrtava se vrsta grafika koja najbolje prikazuje odnos izabranih kolona.

### 7.1.6 Stranica izbor kolona



Slika 23: Stranica za izbor kolona

Na ovoj stranici korisniku se prikazuje tabela sa nazivima kolona. Svaki red tabele sadrži:

- Promenu tipa kolone,
- Da li se kolona koristi za ulaz ili izlaz u modelu,
- Enkodiranje kolone,
- Skaliranje kolone,

Pritiskom na dugme “Sačuvaj Izbor” korisniku se pojavljuje upozorenje da nije moguće praviti izmene osim ako ne učita fajl ponovo. Nakon uspešnih izmena, korisniku se omogućava da kreira neuronsku mrežu klikom na dugme “Kreiraj neuronsku mrežu”.

### 7.1.7 Stranica za podešavanje mreže



Slika 24: Stranica za podešavanje mreže

Dolaskom na stranu za podešavanje mreže korisniku se prikazuje prozor za podelu podataka na trening, validacione i testne podatke (procentualno). Dodatna mogućnost je opcija za mešanje podataka pre podele. Takođe, na istom prozoru se korisniku daje opcija da odabere vrstu problema koji rešava (regresioni ili kvalifikacioni). Ovaj izbor utiče na izbor funkcija i metrika pri kreiranju modela.

Na vrhu strane se nalaze padajući meniji za izbor podešavanja mreže:

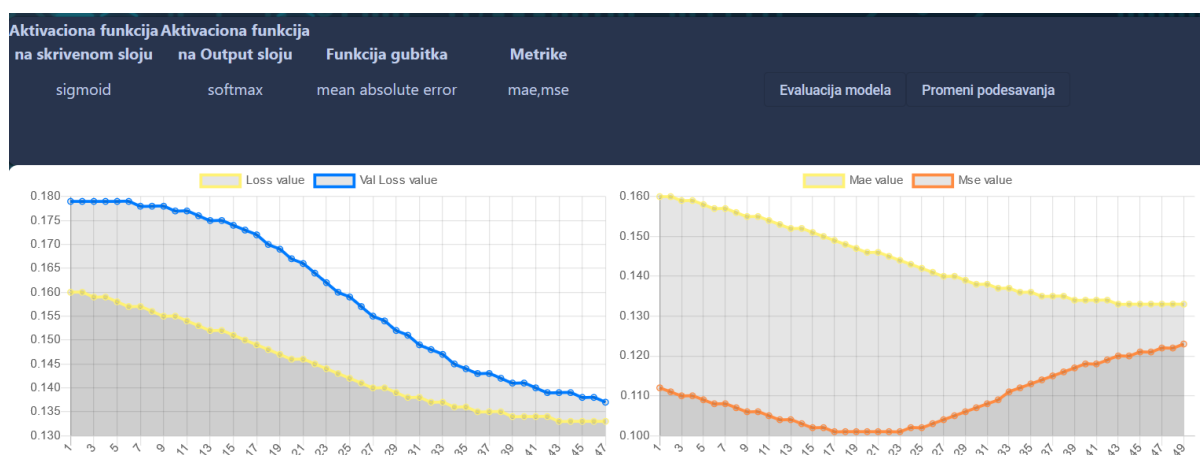
- Aktivaciona funkcija za skrivene slojeve,
- Aktivaciona funkcija za izlazni sloj,
- Loss funkcija,
- Metrike.

Nakon toga, korisnik dolazi do glavnog dela aplikacije, kreiranje neuronske mreže. Klikom na “+” u gornjem levom uglu, korisnik dodaje nov skriveni sloj u mrežu. Iznad svakog skrivenog sloja se nalaze “+” i “-” za dodavanje i brisanje neurona u datom sloju.

Na dnu strane se nalaza podešavanja za treniranje modela koja se mogu izmeniti i nakon zaključavanja mreže (Mreža se zaključava I otključava klikom na dugme u gornjem desnom uglu).

Podešavanja za treniranje sadrže:

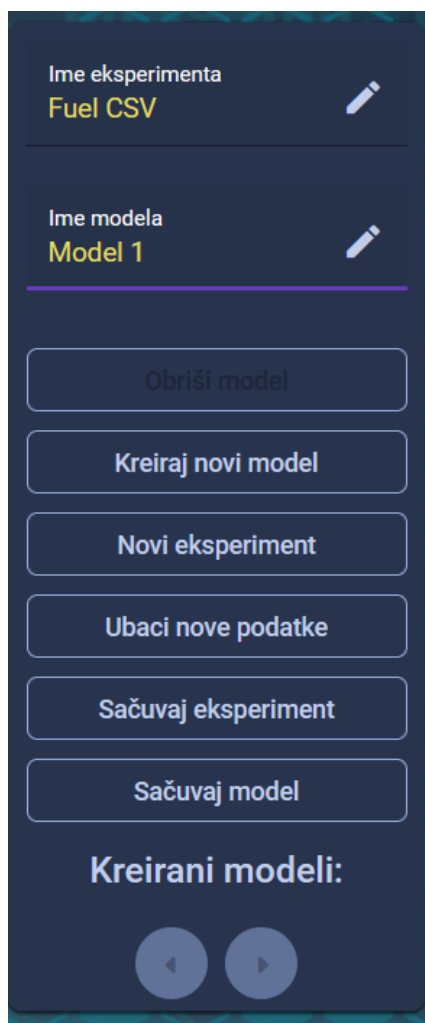
- *BatchSize*,
- Broj epoha,
- Prevrneno zaustavljanje treniranja gde korisnik prosleđuje broj nakon koliko epoha bez promena se treniranje zaustavlja.



Slika 25: Stranica za izbor kolona

Tokom treniranja se u realnom vremenu crtaju grafici koji prikazuju promenu *Loss* funkcije nad trening i validacionim skupom (ukoliko postoji), dok drugi graf prikazuje promenu metrika tokom treniranja.

Nakon završenog treniranja korisnik može da uradi predikciju nad testnim skupom (evaluacija) ili da se vrati na podešavanje mreže.

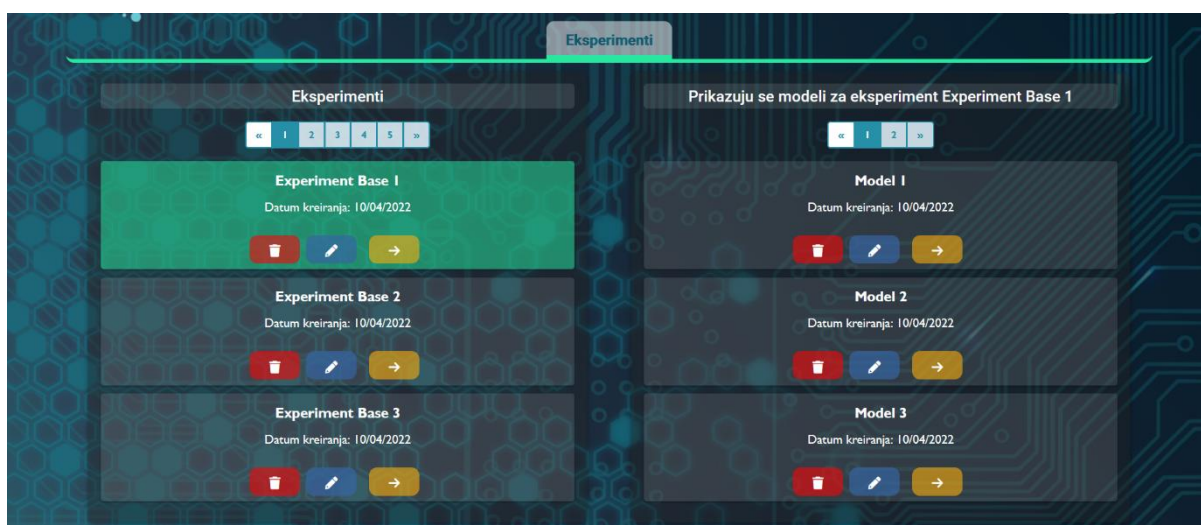


Slika 26: Meni za čuvanje eksperimenata i modela

Sa leve strane se nalazi mogućnost čuvanja ili učitavanja eksperimenta, modela za ponovnu upotrebu.

Ukoliko korisnik želi da sačuva model, neophodno je prvo da sačuva eksperiment. Sve što je potrebno za čuvanje modela ili eksperimenta je da korisnik prosledi naziv.

### 7.1.8 Profil i eksperimenti korisnika



Slika 27: Stranica prikaz i manipulaciju eksperimentata

Prelaskom miša preko imena korisnika u navigacionom meniju, prikazuje se padajući meni. Klikom na “Profil” otvara se profilna stranica korisnika.

Prva kartica prikazuje sve sačuvane eksperimente korisnika, klikom na neki od eksperimentata, prikazuju se svi sačuvani modeli za taj eksperiment.

Na modelu se nalazi žuto dugme sa strelicom, pritiskom na to dugme učitavaju se eksperiment i model pri čemu se model može izmeniti. Učitavanjem eksperimenta se učitavaju poslednja verzija skupa podataka. Izmena podataka nije moguća. Učitavanje modela vodi korisnika na stranicu za podešavanje mreže gde se prikazuje učitani model.

Druga kartica prikazuje informacije o korisniku.