

LATVIJAS UNIVERSITĀTE  
DATORIKAS FAKULTĀTE

NOSAUKUMS

“Rakstu lasīšanas sociālais portāls”

KVALIFIKĀCIJAS DARBS

Autors: Elvijs Pauls Saknītis  
Studenta apliecības Nr. es12074  
Darba vadītājs: Maksims Kravcevs

RĪGA 2018

# Anotācija

Kvalifikācijas darba mērķis ir tīmekļa vietnes izstrāde, kas paredzēta tekstveida rakstu publikācijām un viedokļu, domu izteikšanai komentāru sadaļā zem izvēlētā raksta.

**Atslēgvārdi:** Spring boot, Angular, JWT, Spring boot data, Spring boot security, Spring boot web

# Anotation

(Social article writing website)

The objective is a development of web app suited for text type publications and opinion debates in comment section.

**Keywords:** Spring boot, Angular, JWT, Spring boot data, Spring boot security, Spring boot web

## Saturs

1.	Programmu prasību specifikācija .....	1.
1.1.	Nolūks .....	1.
1.1.2.	Darbības sfēra.....	1.
1.1.3.	Saistība ar citiem dokumentiem .....	1.
1.2.	Vispārējs apraksts.....	1.
1.2.1.	Produkta perspektīva .....	2.
1.2.2.	Funkcijas pa moduļiem.....	2.
1.2.5.	Lietotāju raksturiezīmes.....	2.
1.2.6.	Vispārējie pieņēmumi .....	2.
1.2.7.	Pieņēmumi un atkarības.....	2.
1.3.	Funkcionālās prasības .....	3.
1.3.1.	Lietotāju pārvaldības funkcijas .....	3.
1.3.2.	Rakstu modulis.....	6.
1.3.3.	Komentāru modulis .....	9.
1.4.	Ārējās saskarsmes prasības.....	9.
1.4.1.	Lietotāju saskarne .....	9.
1.4.2.	Programmatūras saskarne .....	10.
1.4.3.	Aparatūras saskarne .....	10.
1.5.	Nefunkcionālās prasības .....	10.
1.5.1.	Veiktspējas prasības .....	10.
1.5.2.	Uzturamība .....	10.
2.	Programmatūras projektējuma apraksts.....	11.
2.1.	Ievads .....	11.
2.1.1.	Nolūks .....	11.
2.1.2.	Sfēra .....	12.
2.1.3.	Saistība ar citiem dokumentiem .....	12.
2.1.4.	Definīcijas .....	12.
2.2.	Datu plūsmu diagramma .....	13.
2.2.1.	DPD vispārejs .....	13.
2.2.2.	DPD1 .....	14.
2.3.	Autorizācijas tiesību kontrole.....	15.
2.4.	Datubāzes projektējums.....	16.
2.5.	Datubāzes fiziskais modelis .....	17.
2.6.	DB tabulu apraksti.....	17.
3.	Projekta organizācija.....	18.
4.	Konfigurāciju pārvaldība .....	19.
5.	Darbietilpības novērtējums.....	20.
6.	Drošība .....	20.
7.	Izmantotā literatūra .....	21.
10.	Pielikumi .....	



# **1. Programmas prasību specifikācija**

## **1.1. Ievads**

### **1.1.1. Nolūks**

Šis dokuments ir paredzēts produkta “Rakstu Lasīšanas Sociālais Portāls” funkcionālo un nefunkcionālo prasību specifikācijas dokumentācijai, kā skaidram un viennozīmīgam prasību uzskaitījumam, pēc kura jāvadās izstrādes laikā.

### **1.1.2. Darbības sfēra**

Ar ‘Rakstu lasīšanu’ uzsvars ir likts uz primāro veicamo uzdevumu, kas ir - nodrošināt ērtu un veiktspējīgu publicēto rakstu rindēšanu, pēc noteiktās kategorijas, rakstu skatīšanu (ieiešana un iziešana no skata), un komentēšanu izvēlētā raksta skatā – zem raksta. Jābūt iespējai skatīt raksta vai komentāra autora profilu ar klikšķi uz spiedlaukuma (lietotājvārda). Lietotāja profila skatā tiks izlikta informācija par pašu lietotāju, un zemāk, pēc izvēles (pogas), rindoti šī lietotāja raksti vai komentāri.

Reģistrētiem lietotājiem jāvar labot vai dzēst savu lietotnes profilu, savus rakstus un komentārus. Administrators var dzēst visus lietotājus, rakstus un komentārus, bet nevar rediģēt šo informāciju.

### **1.1.3. Saistība ar citiem dokumentiem**

Dokuments noformēts saskaņā ar LVS 68:1996 “Programmatūras prasību specifikāciju ceļvedis” standartu.

## **1.2. Vispārējs apraksts**

### **1.2.1. Produkta perspektīva**

Programmai jābūt neatkarīgai un pašpietiekamai no ārējām saskarnēm (neskaitot e-pastu). Programmai jābūt pilnībā izmantojamai no mūsdienīgas interneta pārlūkprogrammas, neprasot nekādu citu programmu palīdzību.

### **1.2.2. Produkta funkcijas pa moduļiem**

#### **1. Lietotāju pārvaldības modulis**

1. Jauna lietotāja reģistrācija
2. Lietotāja pieslēgšanās sistēmai

3. Lietotāja atslēgšanās no sistēmas
4. Esoša lietotāja datu rediģēšana
5. Lietotāju profilu skatīšana
6. Lietotāja dzēšana

## **2. Raksta modulis**

1. Raksta publicēšana
2. Raksta dzēšana
3. Raksta skatīšana

## **3. Komentēšanas modulis**

1. Komentāra izveide

### **1.2.3. Lietotāju raksturierzīmes**

Esošu sociālo saitu elki.

### **1.2.4. Vispārējie ierobežojumi**

Lietotāju reģistrācijai būs nepieciešams derīgs e-pasts. Pieeja internetam un pārlūkprogrammas HTML5 un Javascript atbalsts.

Ir jābūt iespējai kategorizēt rakstus pēc izvēlētās kategorijas (ir jāiekļauj arī noklusējuma ‘visi raksti’ kategorija), un jābūt rakstu meklēšanas teksta ailei, kura atgriezīs rindotas sakrītības ar rakstu virsrakstu nosaukumiem no datu glabātuves. Jābūt ērtai lapošanas funkcionalitātei gan rakstiem, gan komentāriem. Ir jānodrošina sesijas saglabāšana un drošība.

### **1.2.5. Pieņēmumi un atkarības**

Tiek pieņemts, ka galvenā mērķauditorija izmantos programmu no sistemātiski spējīgākām ierīcēm kā galda un portatīvie datori, taču jāvar tīmekļa lapu izmantot no zemākas spējas iekārtām kā androīdu telefoni.

## 1.3. Funkcionālās prasības

Šajā nodaļā ir aprakstītas funkcionālās prasības, kuru darbība izstrādātajai sistēmai ir jānodrošina.

### 1. Lietotāju pārvaldības modulis

#### 1.3.1. Jauna lietotāja reģistrācija

##### 1.3.1.1. Mērķis

Funkcijas mērķis ir pieteikt un saglabāt jauna lietotāja datus sistēmā kā identificējamu un atgūstamu datu kopu.

##### 1.3.1.2. Ievaddati

Reģistrācijas lauki:

- Unikāla e-pasta adrese, garumā no 3 līdz 30 simboliem
- Vismaz 6 simbolus gara parole
- Unikāls lietotājvārds, garumā 3 līdz 30 simboliem

##### 1.3.1.3. Apstrāde

Apstrādājot veikto pieprasījumu, sistēma veic šādas pārbaudes un darbības:

- Pārbaudī datu atbilstību garuma ierobežojumiem
- Pārbauda e-pasta un lietotājvārda unikalitāti sistēmā
- Datubāzē tiek pievienots jaunā lietotāja ieraksts

##### 1.3.1.4. Izvaddati

Pie veiksmīgas reģistrācijas izvada paziņojumu - "Registration successful".

##### 1.3.1.5. Kļūdu paziņojumi

- "This username is already in use"
- "This email is already in use"
- "Email has to be of length 3 to 55"
- "Email has to be of length 3 to 30"
- "Password has to be of length 6 to 30"

### 1.3.2. Lietotāja pieslēgšanās sistēmai

#### 1.3.2.1. Mērķis

Funkcijas mērķis ir identificēt sevi sistēmai.

#### 1.3.2.2. Ievaddati

Ievaddatu lauki:

- Lietotājvārds
- Parole

#### 1.3.2.3. Apstrāde

Sistēma apstrādā:

- Pārbauda vai datubāzē ir reģistrētā lietotāja vārds
- Salīdzina iegūtu paroli ar atgūto no datubāzes

#### 1.3.2.4. Izvaddati

Pie veiksmīgas autentifikācijas izvada paziņojumu - "You are logged in".  
Ievaddatu lauki tiek slēpti.

#### 1.3.2.5. Kļūdu paziņojumi

- "Incorrect username or password"

### 1.3.3. Lietotāja atslēgšanās no sistēmas

#### 1.3.3.1. Mērķis

Funkcijas mērķis ir nepieļaut atkārtotu konkrētā lietotāja pieeju sistēmai bez atkārtotas pieslēgšanās

#### 1.3.3.2. Ievaddati

- Poga

#### 1.3.3.3. Apstrāde

Atkarībā no implementācijas

#### 1.3.3.4. Izvaddati

Paziņojums ar vai bez pārvirzīšanu uz citu saiti - "You just logged out".

#### 1.3.3.5. Kļūdu paziņojumi

-

### 1.3.4. Esoša lietotāja datu rediģēšana

#### 1.3.4.1. Mērķis

Funkcijas mērķis ir nodrošināt lietotājam iespēju mainīt savus datus.

#### 1.3.4.2. Ievaddati

Reģistrācijas lauki:

- Info

#### 1.3.4.3. Apstrāde

Apstrādājot veikto pieprasījumu, sistēma veic šādas pārbaudes un darbības :

- Pārbaudī datu atbilstību garuma ierobežojumiem
- Datubāzē tiek pievienots jaunā lietotāja ieraksts

#### 1.3.4.4. Izvaddati

Paziņojums -“Your profile has been modified”

#### 1.3.4.5. Kļūdu paziņojumi

- “There was a problem”
- “Info has to be at least 6 long”

### 1.3.5. Lietotāju profilu skatīšana

#### 1.3.5.1. Mērķis

Funkcijas mērķis ir nodrošināt iespēju skatīt lietotāju profilus, rindojojot izvēlētā lietotāja augšuplādēto ‘Rakstu’ vēsturi, ar iespēju pēc tam doties uz kādu no tiem.

#### 1.3.5.2. Ievaddati

Nodod sistēmai rīkojumu skatīt lietotāja profilu ar klikšķi uz :

- Autora lietotājvārda pie kāda no ‘Rakstiem’
- Autora lietotājvārda pie kāda no ‘Komentāriem’

#### 1.3.5.3. Apstrāde

Apstrāde :

- Nodod sistēmai lietotāja slēpto kārtasnumuru
- Pieprasa sistēmai lietotāja profila datus

- Pieprasa sistēmai lietotāja ‘Rakstus’

#### 1.3.5.4. Izvaddati

Sistēma veic pāradresāciju uz citu logu un :

- Tiek attēlota lietotāja profila informācija
- Ja lietotājam ir publicēti ‘Raksti’, tad tie tiek rindoti

#### 1.3.5.5. Kļūdu paziņojumi

- “Something went wrong”

### 1.3.6. Lietotāja dzēšana

#### 1.3.6.1. Mērķis

Funkcijas mērķis ir dot iespēju administratoram dzēst lietotāju.

#### 1.3.6.2. Ievaddati

Ar klikšķināmu pogu savā profilā :

- Parole

#### 1.3.6.3. Apstrāde

Apstrādājot veikto pieprasījumu, sistēma veic šādas pārbaudes un darbības :

- Pārbauda vai datubāzē ir reģistrētā lietotāja vārds
- No datubāzes tiek izņemts lietotāja ieraksts

#### 1.3.6.4. Izvaddati

Paziņojums - “Your profile has been deleted”

- Paziņojums – “Profile has been deleted”

#### 1.3.6.5. Kļūdu paziņojumi

- “\_”

## **2. Rakstu modulis**

### 1.3.7. Raksta publicēšana

#### 1.3.7.1 Mērķis

Funkcijas mērķis ir ļaut lietotājiem publicēt ‘Rakstus’.

#### 1.3.7.2 Ievaddati

Klikšķināms laukums, kas redirekto lietotāju uz raksta lapu :

- ‘Raksta’ nosaukums (garumā no 4 līdz 40)
- ‘Raksta’ teksts (garumā no 35)

#### 1.3.7.3. Apstrāde

Apstrāde :

- Pārbauda datu atbilstību garuma ierobežojumiem
- Datubāzē tiek atrasts ‘Raksta’ autors
- ‘Raksts’ tiek ierakstīts datubāzē

#### 1.3.7.4. Izvaddati

Paziņojums:

- Paziņojums - “Article created”

#### 1.3.7.5. Kļūdu paziņojumi

- “Article has to be at least 35 letters”
- “The length of title has to be within 4 - 40”

### 1.3.8. Raksta dzēšana

#### 1.3.8.1 Mērķis

Funkcijas mērķis ir dot iespēju administratoram dzēst ‘Rakstus’.  
Kā arī dot iespēju lietotājam dzēst pašam savu ‘Rakstu’.

#### 1.3.8.2. Ievaddati

Ar klikšķināmu pogu iekš raksta:

- Parole

#### 1.3.8.3. Apstrāde

Apstrādājot veikto pieprasījumu, sistēma veic šādas pārbaudes un darbības :

- Pārbauda vai datubāzē ir reģistrētā lietotāja vārds
- Salīdzina iegūtu paroli ar atgūto no datubāzes
- Dzēš ‘Rakstu’

#### 1.3.8.4. Izvaddati

Paziņojums:

- Paziņojums - “Article deleted”

#### 1.3.8.5. Kļūdu paziņojumi

- “Something went wrong”
- “Incorrect password”

### 1.3.9. Raksta skatīšana

#### 1.3.9.1. Mērķis

Funkcijas mērķis ir dot iespēju lietotājiem skatīt rakstus.

#### 1.3.9.2. Ievaddati

Ar klikšķa notikumu uz izvēlēta raksta nosaukuma :

- Raksta kārtasnumurs

#### 1.3.9.3. Apstrāde

Apstrādājot veikto pieprasījumu, sistēma veic šādas pārbaudes un darbības :

- Atrast ‘Raksta’ lauku datubāzē

#### 1.3.9.4. Izvaddati

Pāradresācija uz jaunu lapu :

- ‘Raksta’ autora izvade
- ‘Raksta’ nosaukuma izvade
- ‘Raksta’ teksta izvade

#### 1.3.9.5. Kļūdu paziņojumi

- “Something went wrong”
- “User has been deleted”

### 3. Komentāru modulis

#### 1.3.8.6. Komentāra publicēšana

##### 1.3.8.7. Mērķis

Funkcijas mērķis ir ļaut lietotājiem publicēt 'Komentārus' zem izvēlētā 'Rakstiem', rakstu lasīšanas logā.

##### 1.3.8.6. Ievaddati

'Komentāra' forma ar klikšķināmu pogu :

- Netukšs 'Komentāra' teksts

##### 1.3.8.5. Apstrāde

Apstrāde :

- Sistēmai tiek nodots 'Komentārs'
- Sistēmai tiek nodots 'Komentāra' autora identifikators
- Sistēmai tiek nodots komentētais 'Raksts'

##### 1.3.8.6. Izvaddati

Paziņojums:

- Pievienots komentārs

##### 1.3.8.7. Kļūdu paziņojumi

- "Something went wrong"

## 1.4. Ārējās saskarnes prasības

### 1.4.1. Lietotāju saskarne

Sistēmas interfeisam jābūt pašizprotamam, pogām, laukiem, informācijas detaļām jābūt viegli atpazīstamām un ērti izkārtotām, pēc plaši pieņemtajiem tīmekļu programmu standartiem.

Lietotājam visām pogām un visai sistēmas funkcionalitātei ir jābūt strādājošai arī uz mazāka ekrāna ierīcēm, kuras paredzētas tīmekļu programmu izmantošanai kā planšetdators vai androīds.

Ir jābūt savi skati :

- noklusējuma lapai
- lietotāja profila skatīšanai
- raksta skatīšanai
- raksta rakstīšanai

### **1.4.2. Programmatūras saskarne**

Ierīcei jābūt nodrošinātai ar kādu no interneta pārlūkprogrammām, kas atbalsta HTML5 un javascript un interneta pieslēgumu.

### **1.4.3. Aparatūras saskarne**

Lietotājam ir nepieciešams dators, planšetdators vai androīda viedtālrunis.

## **1.5. Nefunkcionālās prasības**

### **1.5.1. Veiktspējas prasības**

Servisam jāvar nodrošināt 1000 tiešsaites klientu, bez aizķērumiem.

## 2. Programmatūras projektējuma apraksts

### 2.1. Ievads

Produkta “Rakstu lasīšanas sociālais portāls” programmas prasību specifikācija tiks implementēta kā divu fiziski atdalītu sistēmu arhitektūras kopums. Klienta puses javascript aplikācija (Angular5) un aizmugurējā sistēma (Spring Boot), kas nodrošinās nepieciešamos servissus ar REST pieprasījumiem.

#### 2.1.1. Dokumenta nolūks

Programmatūras projektējuma apraksts (PPA) paredzēts izstrādājamās programmatūras “Rakstu lasīšanas sociālais portāls” projektējuma aprakstīšanai.

#### 2.1.2. Darbības sfēra

PPA apraksta sistēmas projektējuma, t.i., izvirzīto prasību realizāciju programmatūras izstrādes vidē. Sekojošas prasības tika izpildītas sekojoši:

1) Aizmugurējās sistēmas ‘REST’ atbalstu nodrošina ‘Spring boot starter web’, kurš ir ‘MVC’ bāzēts, konfigurējams ietvars, ar kuru tiek specificēti pieprasījumu paraksti un pārvaldīti datu izejpunkti. Datu apmaiņa ar Angular ‘client’ pusi notiek caur ‘JSON’ pierakstu, ienākošā teksta rindotne tiek pa taisno mapēta uz pieteiktā objekta rasējuma ‘blueprint’. ‘Servleti’ nodrošina procesu zarošanu ienākošajiem pieprasījumiem.

2) Datu saglabāšanai glabātuvē tiek izmantota mysql datubāze un objektu saistīto mapēšanu ‘ORM’ nodrošina ‘Spring boot data starter jpa’ ietvars, kurš noteikumu ‘JPA’ implementācijai izmanto Hibernate un Spring data konfigurāciju. Bez jau ieregulētajām glabātuvēm ‘repositories’, kuras jau nodrošina lietotāju ar standarta noklusējuma metodēm, ir izmantots arī pašrakstītas rindotnes ‘queries’ relāciju mapēšanai pa taisno ar objektiem un tipiēm. Šis ietvars iekļauj arī lapuotājus ‘pageable’, kas ļauj gan no datubāzes pieprasīt noteiktu datu apjomu, gan arī nenoslogot klienta pusi ar pārlietu daudz informāciju.

3) Ienākošo pieprasījumu drošību un lietotāju autentificēšanu, un autorizāciju nodrošina ‘spring boot starter security’ ietvars, ‘Java.WebSecurityConfig’ pārraksta noklusējuma bāziski vienkāršo drošības sistēmu, nodrošinot plašas konfigurēšanas iespējas. Piemēram, servlet filtru ķēdošana, izvadpunktu pieejamības noteikumu un tiesību definēšanai.

4) Sesijas drošībai ir izmantots ‘JWT’ (jason tīmekļa tokens), kurš tiek ģenerēts pie autentifikācijas apstiprināšanas un ar kura paraksta atšifrēšanu turpmāk tiek noskaidrots pieprasītājs. Angular (klienta puse) izmanto pārķeršanas ‘interceptor’ rīkojumu šī tokena paraksta ievērošanai, saglabāšanai un turpmākai pieprasījumu izsūtīšanai. Šo tokenu pārķet viens no filtriem, kuriem jāiziet cauri pieprasījumam, līdz viņi var tikt pie datiem.

Piezīme!

Šajā projektā resursu un autentifikācijas serveris nav atdalīti, kā arī sistēmā netiek izmantoti trešo personu autentifikācijas serveru pakalpojumi. JWT tiek ģenerēts, parakstīts un atšifrēts uz vietas.

5) Angular ir statiskais javascript pakojums klienta pusē, proti, tas pats datus neglabā, bet izmanto mūsu aizmugurējās sistēmas sniegtos 'REST' servisu. Kā sistēmas prezentācijas skatījumam, Angular ir 'MVC' cieši nesaistītu komponentu bāzēts ietvars.

### ***2.1.3. Saistība ar citiem dokumentiem***

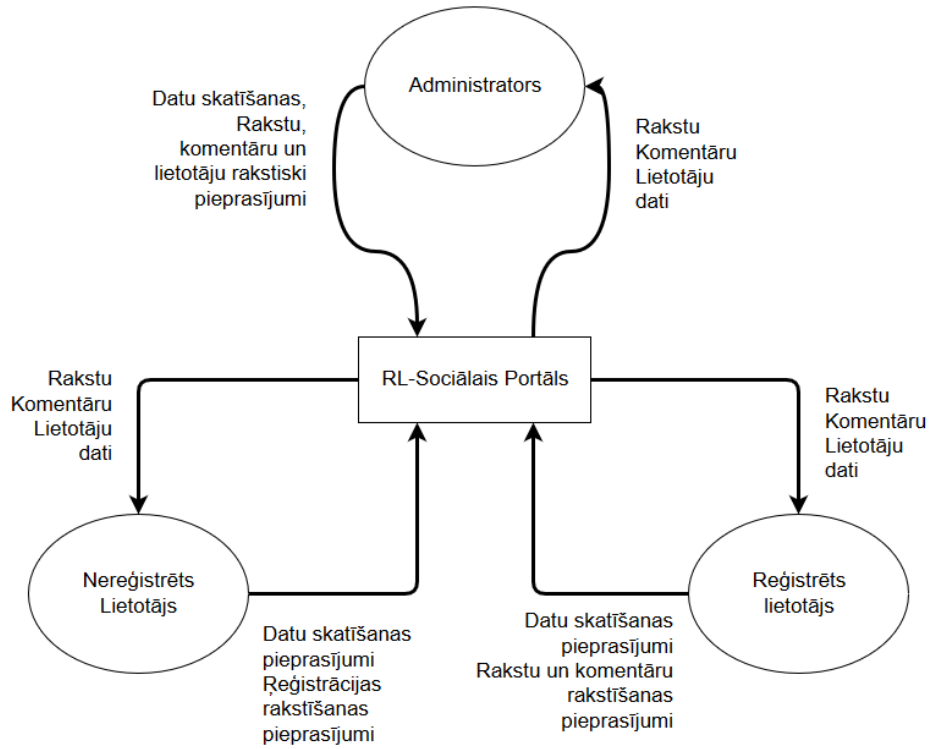
PPA apraksta sistēmas projektējuma, t.i., izvirzīto prasību realizāciju programmatūras izstrādes vidē.

### ***2.1.4. Definīcijas***

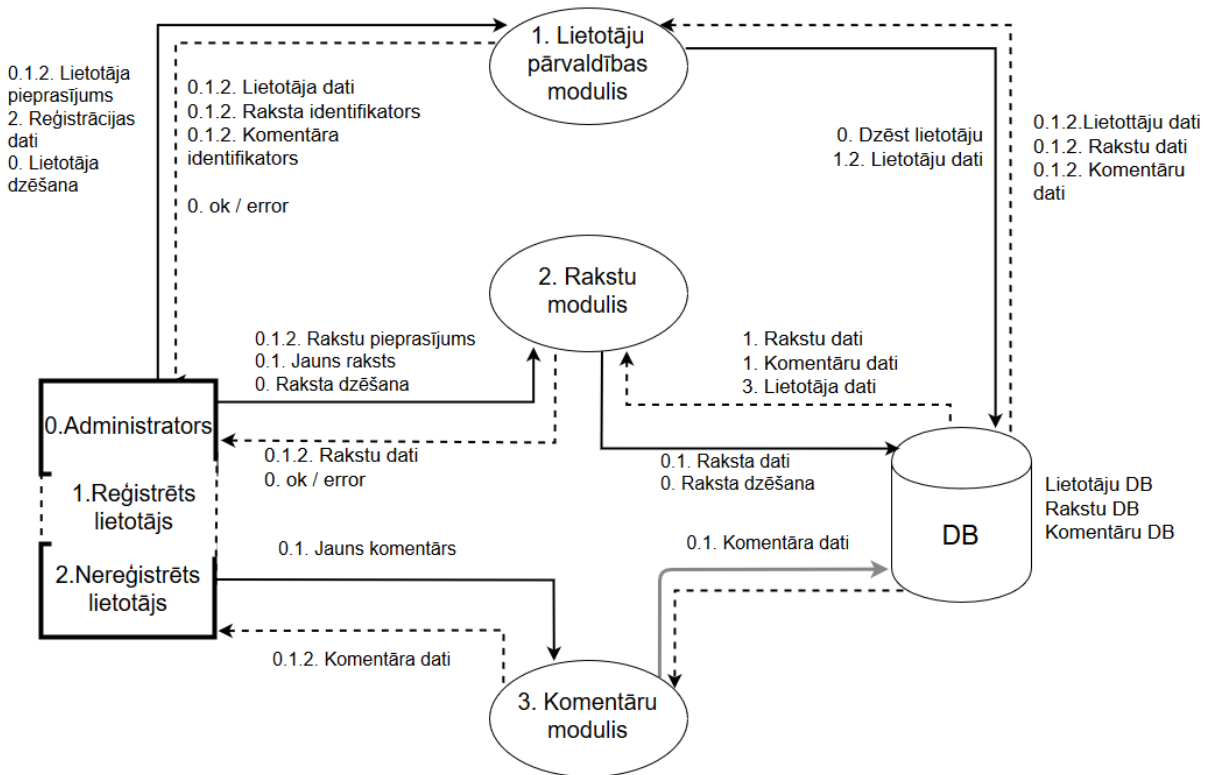
REST	- reprezentatīvā stāvokļa transportēšanas protokols
Spring Boot	- Java Spring ietvara autokonfigurēts blakusietvars
Angular	- Javascript klienta puses ietvars
JWT	- Jason tīmekļu tokeni
MVC	- Modeļa skatījuma kontroliera arhitektūras tips
Javascript	- Skriptējamā valoda, kuru lasa pārlūkprogrammas
HTML5	- Pārlūkprogrammas DOM taisīšanas valoda
Java[.].	- Java klases projektā
JSON	- Vienas ailes pieraksts Javascript sintaksē
Spring boot[.].	- Spring boot ietvars ar optionāli ietveramām komponentēm (security, data-jpa ..)

## 2.2. Datu plūsmu diagrammas

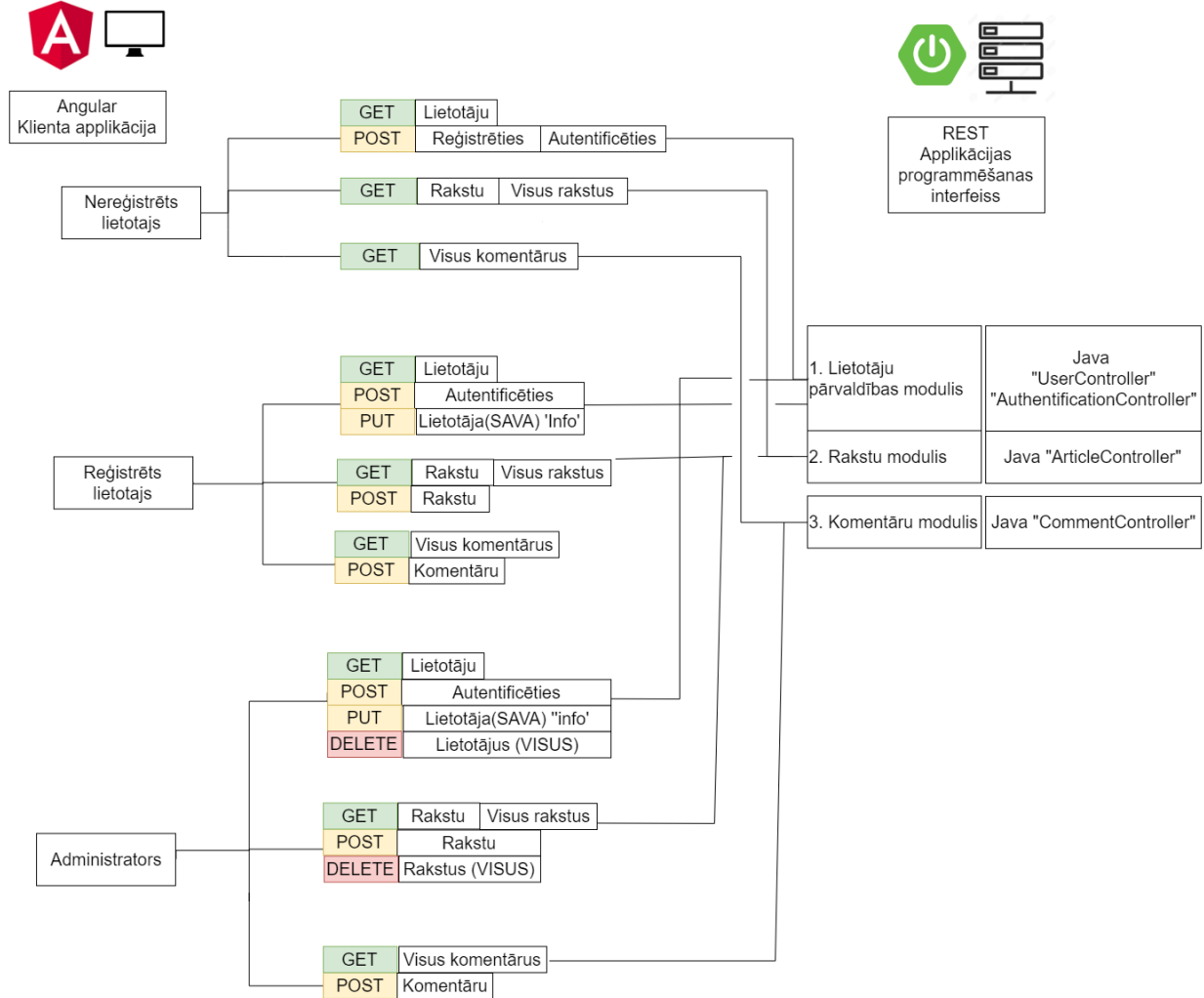
### 2.2.1. DPD vispārējs



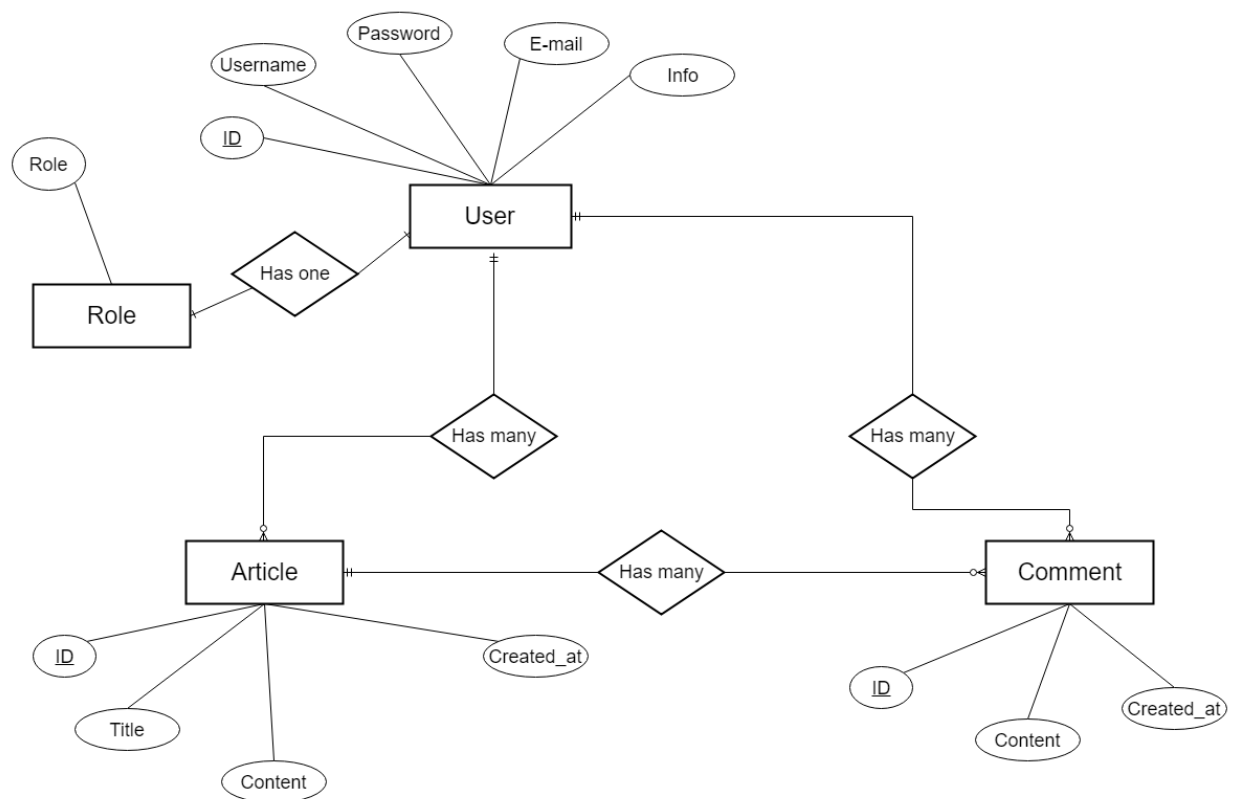
### 2.2.2. DPD 1



### 2.3. Autorizācijas tiesību kontrole REST izejpunktiem

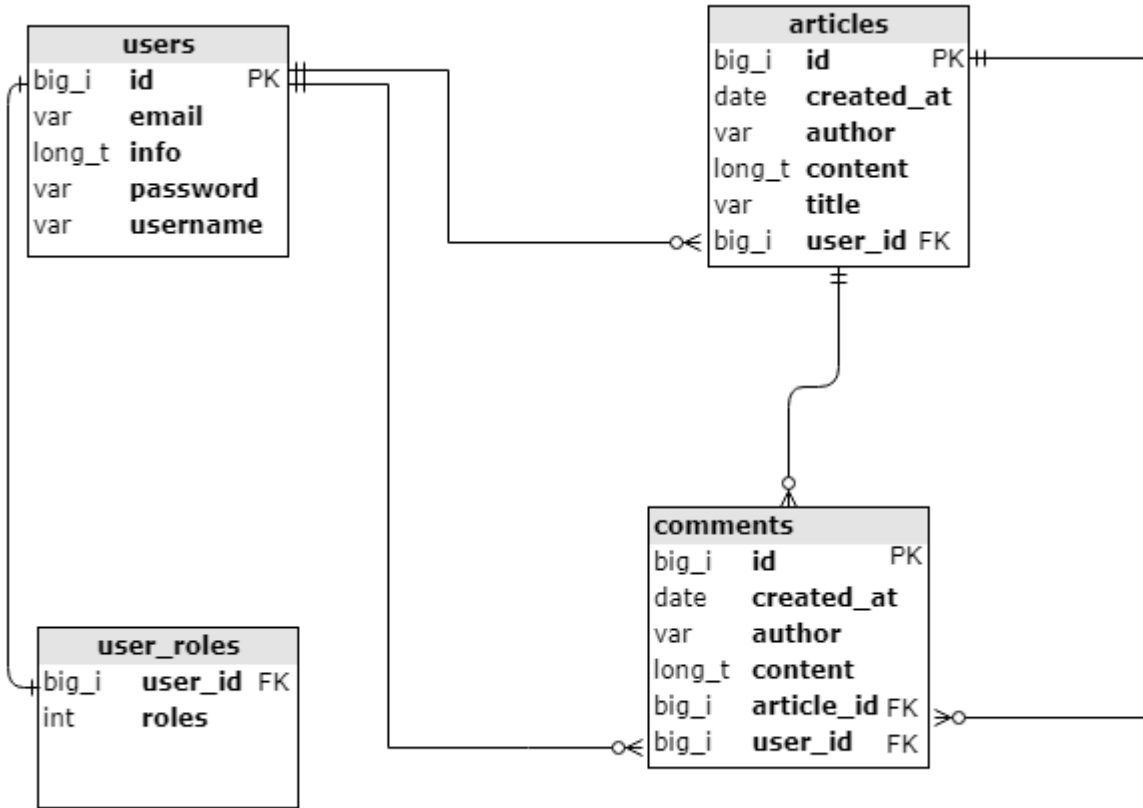


## 2.4. Datubāzes projektējums



## 2.5. Fiziskais db modelis, db tabulu apraksti

### Fiziskais db modelis



## 2.6. DB tabulu apraksti

### Lietotāji

<i>Lauks</i>	<i>Tips</i>	<i>Obligāts</i>	<i>Īpašs ('Unique')</i>	<i>Atslēga</i>	<i>Apraksts</i>
<i>id</i>	<i>bigint(20)</i>	<i>jā</i>	<i>jā</i>	<i>jā</i>	<i>automātiski pieaugošs</i>
<i>email</i>	<i>varchar(30)</i>	<i>jā</i>	<i>jā</i>	<i>nē</i>	<i>lietotāja e-pasts</i>
<i>info</i>	<i>longtext</i>	<i>nē</i>	<i>nē</i>	<i>nē</i>	<i>autora apraksts</i>
<i>password</i>	<i>varchar(30)</i>	<i>jā</i>	<i>nē</i>	<i>nē</i>	<i>parole</i>
<i>username</i>	<i>varchar(30)</i>	<i>jā</i>	<i>nē</i>	<i>nē</i>	<i>lietotājvārds</i>

### Raksti

<i>Lauks</i>	<i>Tips</i>	<i>Obligāts</i>	<i>Īpašs ('Unique')</i>	<i>Atslēga</i>	<i>Apraksts</i>
<i>id</i>	<i>bigint(20)</i>	<i>jā</i>	<i>jā</i>	<i>jā</i>	<i>automātiski pieaugošs</i>
<i>created_at</i>	<i>datetime</i>	<i>jā</i>	<i>nē</i>	<i>nē</i>	<i>kad ieraksts veikts</i>
<i>author</i>	<i>varchar(30)</i>	<i>jā</i>	<i>nē</i>	<i>nē</i>	<i>komentāra autors</i>
<i>content</i>	<i>longtext</i>	<i>jā</i>	<i>nē</i>	<i>nē</i>	<i>komentāra teksts</i>
<i>title</i>	<i>varchar(40)</i>	<i>jā</i>	<i>nē</i>	<i>nē</i>	<i>raksta nosaukums</i>
<i>user_id</i>	<i>bigint(20)</i>	<i>jā</i>	<i>nē</i>	<i>jā</i>	<i>pieder autoram</i>

### Komentāri

<i>Lauks</i>	<i>Tips</i>	<i>Obligāts</i>	<i>Īpašs ('Unique')</i>	<i>Atslēga</i>	<i>Apraksts</i>
<i>id</i>	<i>bigint(20)</i>	<i>jā</i>	<i>jā</i>	<i>jā</i>	<i>automātiski pieaugošs</i>
<i>created_at</i>	<i>datetime</i>	<i>jā</i>	<i>nē</i>	<i>nē</i>	<i>kad ieraksts veikts</i>
<i>author</i>	<i>varchar(30)</i>	<i>jā</i>	<i>nē</i>	<i>nē</i>	<i>komentāra autors</i>
<i>content</i>	<i>longtext</i>	<i>jā</i>	<i>nē</i>	<i>nē</i>	<i>komentāra teksts</i>
<i>article_id</i>	<i>bigint(20)</i>	<i>jā</i>	<i>nē</i>	<i>jā</i>	<i>pieder rakstam</i>
<i>user_id</i>	<i>bigint(20)</i>	<i>jā</i>	<i>nē</i>	<i>jā</i>	<i>pieder autoram</i>

### Lomas

<i>Lauks</i>	<i>Tips</i>	<i>Obligāts</i>	<i>Īpašs ('Unique')</i>	<i>Atslēga</i>	<i>Apraksts</i>
<i>role</i>	<i>Int(11)</i>	<i>nē</i>	<i>jā</i>	<i>nē</i>	<i>0 vai 1 vērtība resp. admins vai lietotājs</i>
<i>user_id</i>	<i>bigint(20)</i>	<i>jā</i>	<i>nē</i>	<i>jā</i>	

### 3. Projekta organizācija

Sistēmas izstrāde tika bāzēta uz ūdenskrituma modeļa principu. Kad tiek noteiktas prasības un jāvadās pēc tām. Tomēr projektēšanas un vispār sistēmu programmēšanas pieredzes trūkuma dēļ, tas ātri vien drīzāk pārtapa par agile veida pieeju, tikai bez kodā programmēšanas testēšanas.

Strādāšana pie sistēmām (Angular un REST) notika vienlaicīgi, lai varētu saglabāt kaut kādu organizētību, novērojot jau kaut kādus beigu rezultātus.

Projekta izstrāde noritēja pa soļiem

- 1) Implementēt REST izejpunktus
  - Pārbaudīt tos no pārlūkprogrammas izsaukumiem
- 2) Implementēt Spring data ORM, pieslēgt MYSQL
  - Novērot pārlūkprogrammas izsaukumus un mysql komandu terminālu
- 3) Uzstādīt Angular, iepazīt 'typescript' valodu un jēgu
  - Veikt pirmos izsaukumus uz REST,
- 4) Strādāt ar filtriem, cors aizsardzību, angular komponentu arhitektūru
- 5) Mācīties darboties ar Spring Security
- 6) Piesaistīt JWT tokenu kā sesiju (nesesijas no servera puses) drošību
- 7) Būvēt gala arhitektūru
  - Katru iznākumu novērot no datubāzes līdz klientam

#### ***4. Konfigurāciju pārvaldība***

Izstrādātās programmatūras versiju pārvaldībai un organizācijai tika izmantota versiju kontrole Git, kura nodrošināja

- pakāpenisku koda versijas datu saglabāšanu
- iepriekšējo versiju datu skatīšanu pēc izvēles
- versiju punktu zarošanu
- ērtu izmaiņu salīdzināšanu ar izvēlēto versijas punktu

Izstrādātās programmatūras konfigurāciju pārvaldību nodrošināja Maven atkarību pārvaldības rīks, kurš nodrošināja

- projektā izmantoto bibliotēku kopošanu
- izvēlēto bibliotēku resursu automātisku ielādi
- pašu bibliotēku atkarību no citām bibliotēkām pārvaldību
- 'pluginu' automātisku piesaisti sistēmai
- projekta opciju specificēšanu

Programmas aizmugurējā 'back-end' daļā REST-API kods tika izstrādāts uz IntelliJ, kas nodrošināja pilnvērtīgu Javas koda konfigurēšanu un pārvaldību.

Programmas 'front end' Angular5-App kods tika izstrādāts uz Visual Studio 2017.

## **5. Darbietilpības novērtējums**

- 2 personmēneši programmatūras koda izstrādei,
- 1 personnedēļa PPS izveidei;
- 1 personnedēļa PPA izveidei;
- 2 personnedēļas testēšanai

## **6.. Drošība**

Priekš izvēlētas ( REST-api servisu un Angular5 aplikācijas ) sistēmas arhitektūras, tās drošību nodrošinās JWT (Jason Web Token) ar Spring Boot Security autorizācijas pārvaldību ‘aizmugurējā servisu sistēmā’.

JWT tokens pēc izvēlēta algoritma un atslēgvārda tiek ģenerēts aizmugurējā sistēmā, pēc piekļuves personas autentifikācijas apstiprinājuma un padots atpakaļ klientam. Tālāk klienta aplikācijai ir tikai jānoglabā šis tokens pārlūkprogrammas lokālajā glabātuvē un pie tālākiem pieprasījumiem jāpievieno JWT tokens pieprasījuma ‘headeram’.

Aizmugurējā sistēma filtrē visus pieprasījumus (neskaitot reģistrāciju un autentifikāciju) un pārlicinās par tokena pareizību, atšifrējot to. (exp data etc)

Pats tokens sevī var nest informāciju, kura ir publiska un privāta, norādot autorizācijas tiesības, bet pats sevī tas paroli neietver, jo nav ieteicams.

Ar jwt tokenu aizmugurējai sistēmai nav jānoglabā nekas, šī drošības metode ir bez-stāvokļa (sessionless).

## **7. Izmantotā literatūra**

<http://jasonwatmore.com/post/2016/09/29/angular-2-user-registration-and-login-example-tutorial>

<http://www.baeldung.com/security-spring>

<http://www.devglan.com/spring-security/spring-boot-security-rest-basic-authentication>

<https://www.linkedin.com/pulse/traditional-authentication-systems-vs-token-based-angular-singh>

<http://www.devglan.com/spring-security/angular-jwt-authentication>

## 10. Pielikumi

### 10.1.Kods

#### Intellij Java Spring Boot REST

The screenshot displays the IntelliJ IDEA interface. On the left, the Project Structure view shows a Maven project with the following structure:

- src/main/java/net.nesley.eyassum
  - controller
    - ArticleController
    - AuthenticationController
    - CommentController
    - UserController
  - dao
    - entity
      - Article
      - Comment
      - TimeStamps
      - User
    - ArticleRepository
    - CommentRepository
    - UserRepository
  - model
    - AuthToken
    - LoginUser
    - Role
    - UserDto
  - security
    - Constants
    - CorsFilter
    - JwtTokenFilter
    - JwtTokenProvider
    - SecurityConfig
  - service
    - servicImpl
      - CustomException
      - UserService2
      - UserService
    - EyassumApplication
  - resources
    - static
    - templates
    - application.properties

On the right, the source code for the `User` entity is shown:

```
1 package net.nesley.eyassum.dao.entity;
2
3 import ...
4
5 @Entity
6 @Table(name="users")
7 public class User {
8
9     @Id
10    @GeneratedValue(strategy = GenerationType.AUTO)
11    private Long id;
12
13    @NotBlank
14    @Column(name="username", length=30, unique=true)
15    @Size(min=3)
16    private String username;
17
18    @NotBlank
19    @JsonIgnore
20    @Size(min=6)
21    private String password;
22
23    @NotBlank
24    @Column(name="email", length=30, unique=true)
25    private String email;
26
27    @ElementCollection(fetch = FetchType.EAGER)
28    List<Role> roles;
29
30    public Long getId() { return id; }
31
32    public void setId(Long id) { this.id = id; }
33
34    public String getUsername() { return username; }
35
36    public void setUsername(String username) { this.username = username; }
37
38    public String getPassword() { return password; }
39
40    public void setPassword(String password) { this.password = password; }
41
42    public String getEmail() { return email; }
43
44    public void setEmail(String email) { this.email = email; }
45
46    public List<Role> getRoles() { return roles; }
47
48 }
```

The screenshot displays an IDE window with the following components:

- Project Explorer (Left):** Shows the project structure for 'eyassum'. The 'entity' package is expanded, and 'Article' is selected.
- Code Editor (Right):** Shows the source code for 'Article.java'. The code is as follows:
 

```

2  import ...
3
19
20 @Entity
21 @Table(name="articles")
22 public class Article extends TimeStamps {
23     @Id
24     @GeneratedValue(strategy= GenerationType.IDENTITY)
25     private Long id;
26
27     @NotBlank
28     @Size(max=100)
29     @Column(unique=true)
30     private String title;
31
32     @NotBlank
33     @Lob
34     private String content;
35
36     @ManyToOne(fetch= FetchType.LAZY)
37     @JoinColumn(name="user_id", nullable = false)
38     // @OnDelete(action = OnDeleteAction.CASCADE)
39     @JsonIgnore
40     private User user;
41
42     public Long getId() { return id; }
43
44
45
46     public void setId(Long id) { this.id = id; }
47
48
49
50     public String getTitle() {
51         return title;
52     }
53
54     public void setTitle(String title) { this.title = title; }
55
56
57     public String getContent() { return content; }
58
59
60
61     public void setContent(String content) { this.content = content; }
62
63
64
65
66
67 }
68

```
- File Tabs (Top):** Shows 'SecurityConfig.java', 'Article.java', and 'JwtTokenProvider.java'.

```
<version>0.0.1-SNAPSHOT</version>
<packaging>jar</packaging>

<name>eyssun</name>
<description>Demo project for Spring Boot</description>

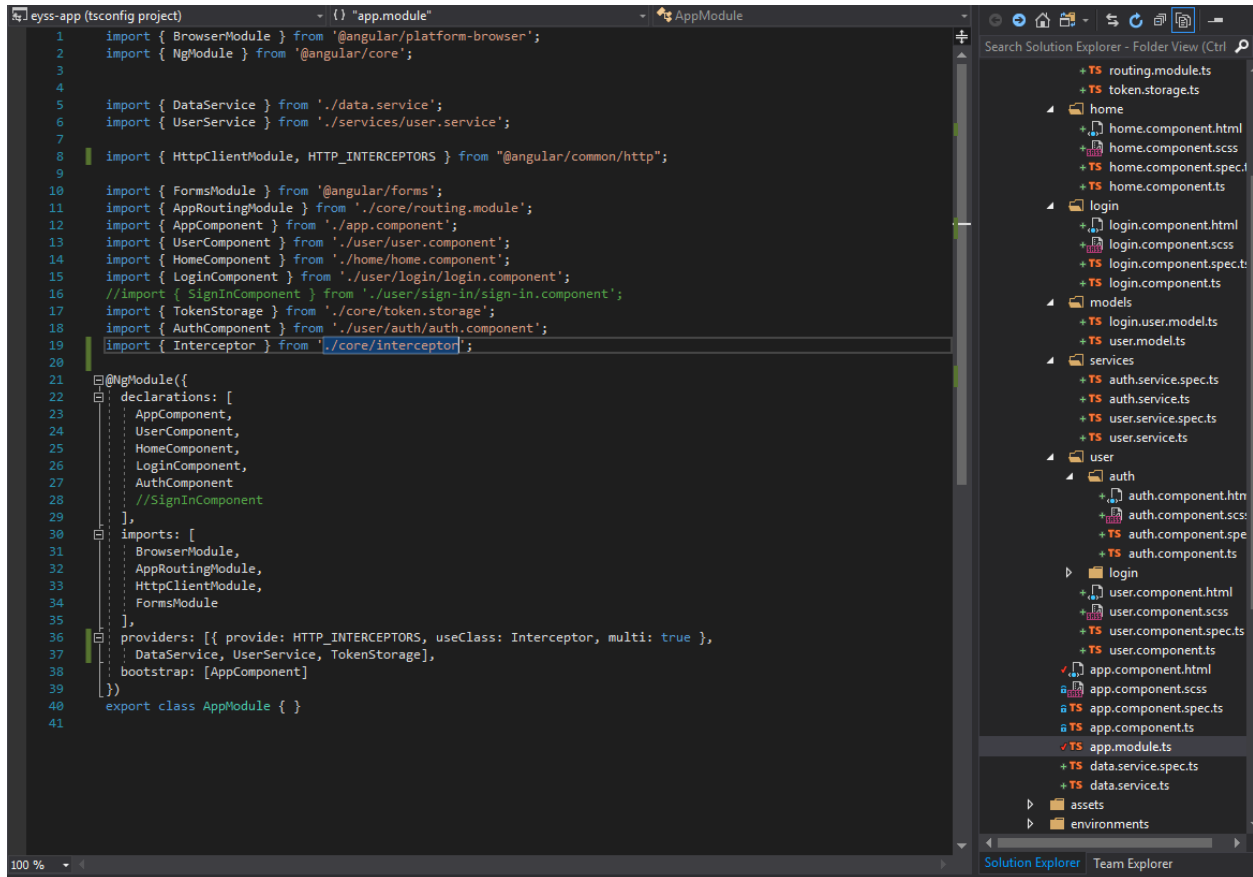
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.0.2.RELEASE</version>
  <relativePath/>
</parent>

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
  <java.version>1.8</java.version>
</properties>

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
  </dependency>
  <!-- Drošibai -->
  <!--<dependency>-->
  <!--<groupId>org.springframework.boot</groupId>-->
  <!--<artifactId>spring-boot-starter-actuator</artifactId>-->
  <!--</dependency>-->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
  </dependency>
  <dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt</artifactId>
    <version>0.9.0</version>
  </dependency>
  <!-- Vienibu Testiem -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
  </dependency>
</dependencies>
```

project

# Angular 5



Kvalifikācijas darbs „*Rakstu lasīšanas sociālais portāls*” izstrādāts Latvijas Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: *Elvijs Pauls Saknītis* \_\_\_\_\_ .05.2018.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs/a: *doc. Maksims Kravcevs* \_\_\_\_\_ .05.2018.

Recenzents: *M.dat. Jānis Mārtužs*

Darbs iesniegts 28.05.2018.

Kvalifikācijas darbu pārbaudījumu komisijas sekretāre: *Darja Solodovņikova* \_\_\_\_\_

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

\_\_\_\_.06.2018. prot. Nr. \_\_\_\_\_

Komisijas sekretārs(-e): \_\_\_\_\_