LATVIJAS UNIVERSITĀTE

DATORIKAS FAKULTĀTE

**SENTIMENTU ANALĪZE TEKSTIEM KRIEVU VALODĀ**

KVALIFIKĀCIJAS DARBS

Autors: Pēteris Girgensons

Studenta apl. Nr.: pg13013

Darba vadītājs: M.vad.zin. Pēteris Paikens

RĪGA 2015

# Anotācija

Šī programma sniedz portāla “tvnet.lv” moderatoriem iespēju pamanīt agresīvi un vardarbīgi noskaņotus komentārus.

Sistēma sastāv no 2 moduļiem: servera daļas un pārlūkprogrammas daļas.

Atslēgvārdi: noskaņojuma analīze, mašīnmācīšanās, tekstu apstrāde

# Abstract

This program gives the moderators of “rus.tvnet.lv” portal a possibility to detect aggressive and violent comments.

The sysyem consists of two modules: server side and browser side.

KEYWORDS: sentiment analysis, machine learning, text evaluation

Saturs

[Anotācija 2](#_Toc418588331)

[Abstract 3](#_Toc418588332)

[DEFINĪCIJAS UN SAĪSINĀJUMI 6](#_Toc418588333)

[Ievads 7](#_Toc418588334)

[Sistēmas mērķis 7](#_Toc418588335)

[Darba mērķi un uzdevumi 7](#_Toc418588336)

[Darba struktūra 7](#_Toc418588337)

[PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA 8](#_Toc418588338)

[Ievads 8](#_Toc418588339)

[Nolūks 8](#_Toc418588340)

[Darbības sfēra 8](#_Toc418588341)

[Definīcijas 8](#_Toc418588342)

[Saistība ar citiem dokumentiem 8](#_Toc418588343)

[Vispārējs apraksts 8](#_Toc418588344)

[Lietotāji 8](#_Toc418588345)

[Lietotāja raksturiezīmes 9](#_Toc418588346)

[Vispārējie ierobežojumi 9](#_Toc418588347)

[Funkcionālās prasības 9](#_Toc418588348)

[Ārējā saskarne 12](#_Toc418588349)

[Lietotāja saskarne 12](#_Toc418588350)

[Sakaru saskarne 12](#_Toc418588351)

[Nefunkcionālās prasības 12](#_Toc418588352)

[Izmantojamība 12](#_Toc418588353)

[Drošība 12](#_Toc418588354)

[Pārnesamība 13](#_Toc418588355)

[PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS 14](#_Toc418588356)

[Nolūks 14](#_Toc418588357)

[Saistība ar citiem dokumentiem 14](#_Toc418588358)

[Definīcijas 14](#_Toc418588359)

[KVALITĀTES NODROŠINĀŠANA 20](#_Toc418588360)

[Konfigurācijas pārvaldība 21](#_Toc418588361)

[TESTĒŠANAS DOKUMENTĀCIJA 22](#_Toc418588362)

[PROGRAMMATŪRAS PROJEKTA ORGANIZĀCIJA 24](#_Toc418588363)

[DARBIETILPĪBAS NOVĒRTĒJUMS 25](#_Toc418588364)

[IZMANTOTĀ LITERATŪRA 26](#_Toc418588365)

[Pirmkoda paraugs 27](#_Toc418588366)

[Secinājumi 31](#_Toc418588367)

# DEFINĪCIJAS UN SAĪSINĀJUMI

Lietotājs

Administrators

PHP

CSS

JavaScript

Jquery

Ajax

HTML

SQL

UTF-8

BootStrap

PPS

PPA

Git

GitHub

Repozitorijs

XAMPP

Apache

Treniņdati

# Ievads

Kvalifikācijas darbā aprakstīta programmas funkcionalitāte un prasības.

Programmatūras izveidošanai tika izmantotas PHP un JavaScript valodas.

## Sistēmas mērķis

Šīs programmas mērķis ir atvieglot „rus.tvnet.lv” portāla moderatoru dzīvi, ļaujot viņiem ātri pamanīt naidīgus un pazemojošus komentārus

## Darba mērķi un uzdevumi

Kvalifikācijas darba mērķis ir atspoguļot autora Latvijas Universitātē un praksē

apgūtās zināšanas.

## Darba struktūra

Kvalifikācijas darbs iekļauj programmatūras prasību specifikāciju, programmatūras

projektējuma aprakstu, testēšanas dokumentāciju, kvalitātes nodrošināšanu, konfigurāciju pārvaldību, programmatūras projekta organizāciju, darbietilpības novērtējumu un būtiskākos pirmkoda fragmentus.

# PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

## Ievads

1. **Ievads** - īsi aprakstīts šī dokumenta un programmas mērķis un būtība.

2. **Vispārējs apraksts** - īsumā aprakstītas programmas funkcijas, lietotāji, ierobežojumi un atkarības.

3. **Funkcionālās prasības** - visas programmas prasības, saskaņā ar kurām programmas tālāk tiks projektēta, izstrādāta, testēta, akceptēta un ieviesta ekspluatācijā.

4. **Nefunkcionālās prasības** - programmas nefunkcionālās prasības, saskaņā ar kurām programmas tiks projektēta, izstrādāta, testēta, akceptēta un ieviesta ekspluatācijā, kā arī definē programmas pasūtītāja atbildību.

5. **Ārējās saskarnes prasības** - sadaļa ietver saskarnes prasības un ierobežojumus.

### Nolūks

Programmatūras prasību specifikācija, jeb PPS, tika izstrādāta, lai pasūtītājs varētu precīzi definēt nepieciešamās sistēmas prasības.

### Darbības sfēra

Programmas ir paredzēta lielu portālu moderatoriem, kuri vēlas noteikt, kuri viņu portāla komentāri ir agresīvi, pazemo cilvēkus vai aicina uz vardarbību.

### Definīcijas

Definīcijas un saīsinājumi sīkāk aprakstīti sadaļā *„Vārdnīca”.*

### Saistība ar citiem dokumentiem

Programmatūras prasību specifikācijas noformēšanā ievērotas standarta LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis” prasības.

### Vispārējs apraksts

Programma ir atkarīga tikai no portāla, kurā atrodas komentāri.

### Lietotāji

Portāla moderators

Sistēmas administrators

### Lietotāja raksturiezīmes

Jebkuram sistēmas lietotājam ir jābūt pamatiemaņām darbā ar datoru, izvēlēto interneta pārlūkprogrammu un elementārai izpratnei par tīmekļa vietnēm. Citas specifiskas raksturiezīmes nav nepieciešamas, lai lietotājs spētu pilnvērtīgi izmantot šo programmu.

### Vispārējie ierobežojumi

Programma nav daļa no citas programmatūras, taču, tā kā sistēma ir realizēta

tīmeklī, funkcionalitāti var ierobežot interneta pārlūkprogramma, interneta savienojums un

servera noslodze.

### Funkcionālās prasības

|  |
| --- |
| Autorizācija |
| Lietotājs var autorizēties sistēmā |
| Ievade: |
| Lietotājvārds  Parole |
| Apstrāde: |
| Pārbauda, vai lietotājs ar šādiem datiem ir piereģistrēts  Pārbauda, vai ievadīti dati ir korekti  Pārbauda, vai obligāti aizpildāmi lauki ir aizpildīti  Kļūdas gadījumā izvada kļūdas paziņojumu |
| Izvade: |
| Paziņojums par veiksmīgu autorizāciju |

|  |
| --- |
| Reģistrācija |
| Lietotājs var piereģistrēties, lai varētu pilnvērtīgi lietot sistēmu |
| Ievade: |
| Lietotājvārds  Parole |
| Apstrāde: |
| Saglabā lietotāja datus datu bāzē  Pārbauda, vai ievadīti dati ir korekti  Pārbauda, vai obligāti aizpildāmi lauki ir aizpildīti  Kļūdas gadījumā izvada kļūdas paziņojumu |
| Izvade: |
| Paziņojums par veiksmīgu reģistrāciju |

|  |
| --- |
| Komentāru klases noteikšana ar „Chrome” paplašinājumu |
| „Chrome” paplašinājums iezīmē negatīvus komentārus sarkanus, bet neitrālus kā zaļus. |
| Ievade: |
| Raksta tīmekļa adrese |
| Apstrāde: |
| Dokumenta parsēšana un komentāru izgūšana  Katra komentāra klases noteikšana |
| Izvade: |
| Katra komentāra iezīmēšana sarkanajā vai zaļajā krāsā atkarībā no klases |

|  |
| --- |
| Komentāra klases maiņa |
| Gadījumā, ja lietotājs nepiekrīt programmas rezultātam, ir iespējams manuāli izmainīt komentāra klasi |
| Ievade: |
| Komentāra teksts  Pareizā klase |
| Apstrāde: |
| Komentārs tiek pievienots pareizās klases treniņdatu kopai  Ar papildinātiem treniņdatiem funkcija „Komentāru klases noteikšana ar „Chrome” paplašinājumu” tiek palaista vēlreiz  Kļūdas gadījumā izvada kļūdas paziņojumu |
| Izvade: |
| Katra tekošās tīmekļa adreses raksta komentāra iezīmēšana sarkanajā vai zaļajā krāsā atkarībā no klases |

|  |
| --- |
| Treniņdatu pievienošana |
| Ir iespējams manuāli papildināt treniņdatu kopu |
| Ievade: |
| Jauno datu saturs  Paredzētā klase |
| Apstrāde: |
| Komentārs tiek pievienots dotās klases treniņdatu kopai  Kļūdas gadījumā izvada kļūdas paziņojumu |
| Izvade: |
| Paziņojums par veiksmīgu rezultātu |

|  |
| --- |
| Treniņdatu dzēšana |
| Ir iespējams manuāli nodzēst komentāru no treniņdatu kopas |
| Ievade: |
| Dzēšamie komentāri  Dzēšamā komentāra klase |
| Apstrāde: |
| Komentārs tiek dzēsts no dotās klases treniņdatu kopas  Kļūdas gadījumā izvada kļūdas paziņojumu |
| Izvade: |
| Paziņojums par veiksmīgu rezultātu |

|  |
| --- |
| Populārāko rakstu analīze |
| Ir iespējams atrast visus negatīvus komentārus no šobrīd populārākiem „rus.tvnet.lv” rakstiem |
| Apstrāde: |
| Tiek parsēti komentāri no šobrīd populārākiem rakstiem  Tie tiek saglabāti XML failā  Katra komentāra klase tiek noteikta ar klasifikatora palīdzību  Kļūdas gadījumā izvada kļūdas paziņojumu |
| Izvade: |
| Tiek izvadīti visi negatīvi komentāri no pārbaudītajiem rakstiem |

|  |
| --- |
| Naidīgāko komentētāju atrašana |
| Ir iespējams atrast komentētājus ar vislielāko negatīvu komentāru skaitu no šobrīd populārākiem „rus.tvnet.lv” rakstiem |
| Apstrāde: |
| Tiek parsēti komentāri no šobrīd populārākiem rakstiem  Tie tiek saglabāti XML failā  Katra komentāra klase tiek noteikta ar klasifikatora palīdzību  Tiek atrasti komentētāji ar vislielāko negatīvu komentāru skaitu  Kļūdas gadījumā izvada kļūdas paziņojumu |
| Izvade: |
| Tiek izvadīti visnaidīgākie komentētāji no pārbaudītajiem rakstiem |

### Ārējā saskarne

### Lietotāja saskarne

Visas sistēmas pogas, darbības un paziņojumi ir latviešu valodā. Sistēmai jābūt lietojamai bez tās papildus uzstādīšanas un jāpietiek tikai ar tīmekļa pārlūkprogrammu.

Par vizuālo sistēmas noformējumu atbild kaskadētu stila lapu valodas (CSS) nosacījumi, lietotāja izvēlētā interneta pārlūkprogramma un ierīces skatlauka (viewport) platums. Sistēma tiek veidota pēc responsīvā dizaina pamatnosacījumiem, tātad programmas vietnes daļa spēs pielāgoties un attiecīgi attēlot saturu, atbilstoši ierīces izmēriem. Programmas vietnes daļas responsivitāti pamatā nodrošina „Bootstrap” klienta pušu tehnoloģiju ietvars. Vietnes dizains, visos tai eksistējošajos skatos, ir vienkāršots un nemainīgs.

### Sakaru saskarne

Saskarnes veids, kādā lietotājs un sistēma savā starpā komunicē, tiek nodrošināts ar HTTP protokola palīdzību.

### Nefunkcionālās prasības

### Izmantojamība

### Drošība

Visi dati, kas tiek sūtīti izmantojot „POST” metodi, tiek filtrēti caur CSRF (XSRF) filtriem.

Lietotāju paroles datubāzē tiek glabātas vienīgi šifrētā veidā, izmantojot „MD5” šifrēšanas algoritmu. Tāpat arī visas darbības ar sistēmas lietotāju parolēm, notiek šifrētā veidā.

### Pārnesamība

Sistēmas pārnesamības sarežģītība nav augsta. Nepieciešamības gadījumā, sistēmu būtu viegli pārvietot uz citu serveri

# PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

1. Ievads - īsi izklāstīts šī dokumenta un sistēmas, uz kuru dokuments attiecas, mērķis un būtība.

2. Dekompozīcijas apraksts - sīkāk izklāstīts par sistēmas moduļiem un to savstarpējām sakarībām.

3. Datu plūsmu diagrammas - grafiski attēlotas sakarības starp sistēmas funkcijām, moduļiem un lietotājiem.

4. Datu bāzes projektējums – (iespējams)

5. Lietotāja ekrāna formu projektējums - izklāstīts ekrānformu projektēšanas process.

### Nolūks

Programmatūras projektējuma apraksta, jeb PPA, izveides nolūks ir tehniski konkretizēt programmatūras prasību specifikācijā (PPS) esošo informāciju, atvieglojot sistēmas izstrādes procesu.

### Saistība ar citiem dokumentiem

Programmatūras projektējuma apraksts tika izveidots, cenšoties balstīties uz Latvijas Valsts standarta LVS 72:1996, „Ieteicamā prakse programmatūras projektējuma aprakstīšanai” vadlīnijām.

### Definīcijas

Definīcijas un saīsinājumi sīkāk aprakstīti sadaļā „Vārdnīca”.

Datņu apraksts

Failu saraksts

NaiveBayesClass.php – šajā failā tiek aprakstīta sistēmas galvenā klase „sentiments”

Cross-val.php – kross-validācijas skripts

Negative\_data.txt – šajā failā atrodas negatīvās klases treniņdati

Neutral\_data.txt – šaja failā atrodas neitrālās klases treniņdati

Stop-words.txt – šaja failā atrodas „stopvārdi” – reti lietojami vārdi, kurus var neņemt vērā

Check\_label.php – programma, kas nosaka viena konkrētā komentāra klasi

Added\_negative.txt – lietotāja pievienotie negatīvās klases treniņdati

Added\_neutral.txt – lietotāja pievienotie neitrālās klases treniņdati

„Chrome” paplašinājums

Manifest.json – paplašinājuma galvenais fails

Content.js – paplašinājuma programmas pirmkods

Noskaņojuma analīzes veidus var sadalīt četrās grupās: analīze atbilstoši nosacījumiem, analīze ar vārdnīcu, pārraudzītā mācīšanās, nepārraudzītā mācīšanās.

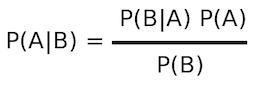
Pirmais veids ir analīze atbilstoši nosacījumiem (rule based analysis). Daudzas komercsistēmas lieto šo pieeju, kaut gan tā pieprasa nozīmīgus laika un resursu patēriņus, jo augstas precizitātes sasniegšanai ir jāapraksta pietiekami daudz noteikumu. Pie tam parasti noteikumi ir ļoti atkarīgi no projekta temata, tādēļ, mainot tēmu programma ir jāizstrādā pilnīgi no jauna. Neskatoties uz to visu, šī pieeja paliek ļoti populāra un, ja ir pietiekams datu apjoms, ļoti precīza, kaut gan no algoritmu viedokļa neļoti interesanta pētīšanai.

Analīze ar vārdnīcu. Tiek pielietotas tonalitātes vārdnīcas (affective lexicons). Vienkāršajā veidā vārdnīca ir vārdu saraksts ar katra vārda sentimentu koeficientu.

Nepārraudzītā mācīšanās (nonsupervised machine learning) - visinteresantākais, bet tajā pašā laikā vismazāk precīzs un veids. Šo metodi ir arī visgrūtāk īstenot. Mēs ļaujam programmai patstāvīgi atrast tekstu likumsakarības.

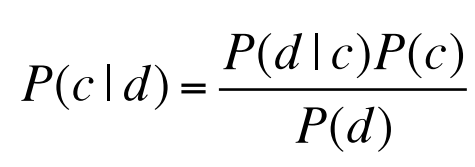
Pārraudzītā mācīšanās (supervised machine learning) ir visizplatītākā pētījumos metode. Lai realizētu šo pieeju, ir nepieciešams savākt treniņdokumentu kopu klasifikatora apmācīšanai. Katrs documents ir jāapzīmē ar pareizo klasi jeb sentimentu. Pēc tam jāizvēlās klasifikācijas algoritmu. Tad būs iespējams izmantot iegūto modeli. Es izmantošu tieši šo metodi.

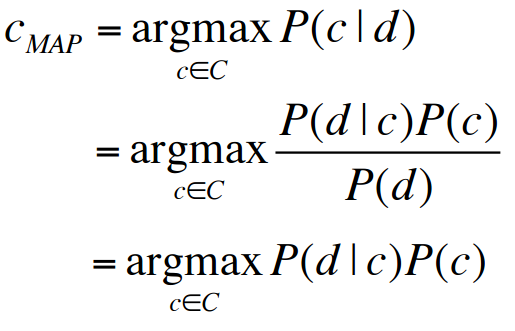
Naivais Baiesa klasifikators ir vienkāršs, bet ļoti efektīvs. Tas ir bieži izmantots tekstu analīzēs, un tas efektīvi strādā gan ar mazām, gan ar lielām datu kopām. Naivais Baiesa klasifikators ir balstīts uz Baiesa teorēmas. Klasifikators ir salīdzinoši viegli pielietojams.



Pēc Baiesa teorēmas var secināt, ka A notikuma varbūtība, ja B notikums ir patiess ir vienāda ar B notikuma varbūtības, ja A notikums ir patiess sareizinātu ar A notikuma varbūtību, un tas viss ir dalīts ar B notikuma varbūtību.

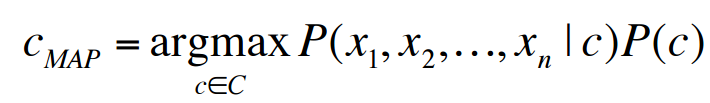
Mūsu gadījumā mēs varam pārrakstīt Baiesa teorēmu šādi: dokumentam d un klasei c, vārbūtība, ka dotais dokuments pieder klasei c ir





MAP – „maximum a posteriori” – visvairāk iespējamā klase

Saucēju var atmest, jo pie visām klasēm tas paliks nemainīgs.



P(d) izteikts, kā vārdu x1, x2, x3 ... xn kopa.

Klasifikators ir balstīts uz „vārdu somas” (bag of words) principa. Tas nozīmē, ka mēs varam uztvert tekstu, kā vārdu kopu, neņemot vērā gramatiku un vārdu kārtību, bet ņemot vērā vārdu biežumu.

Kāpēc naivais?

Klasifikators ir veidots ar diviem pieņēmumiem:

vārdu somas princips

ir pieņemts, ka katra vārda varbūtība ir neatkarīga, ja ir dota klase c

**Algoritma apraskts.**

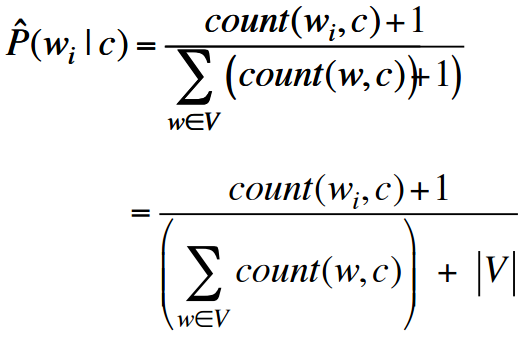
Vispirms ir jāiegūst no interneta nepieciešamie treniņdati – tekstu kopas, pēc kurām mūsu klasifikators varēs noteikt dotā teksta klasi.

Tika izveidota klase „sentiments”. Šai klasei ir definētas trīs galvenās metodes add, classify un tokenize.

Lai pievienotu dokumentu treniņdatiem , ir jāpadod dotais dokuments, kā parametrs funkcijai add. Funkcija add izsauc funkciju tokenize, kura sadala dokumentu atsevišķos vārdos. Pēc tam funkcija add pārtaisa katru vārdu augšējā reģistrā. Tiek apskatīts katrs vārds un ar bibliotēkas phpmorphy palīdzību tiek atrasta katra vārda normālforma. Dotais vārds tiek saglabāts mainīgajā index pie vajadzīgās klases.

Lai noteiktu, pie kādas klases pieder dotais dokuments, ir jāpadod šis dokuments kā arguments funkcijai classify. Tā veic tādu pašu vārdu apstrādi, kā add – tokenizācija, pārveidošana augšējā reģistrā un normālformu iegūšana. Pēc tam tiek iegūta katras klases vārbūtība pēc šāda principa -

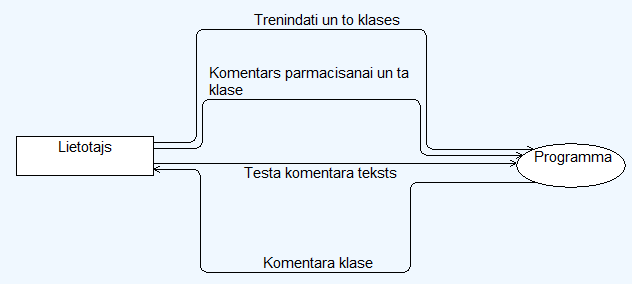
šis vārds šīs klases treniņdatos dalīts ar visiem vārdiem šajā klasē. Vēlāk tiek piereizināti prioru vārbūtības. Dažreiz tiek sastapti tādi vārdi, kas netika sastapti treniņdatos. Tādā gadījumā, lai novērstu reizināšanu ar nulli, tiek pielietota Laplasa izlīdzināšana (Laplace or add-1 smoothing).

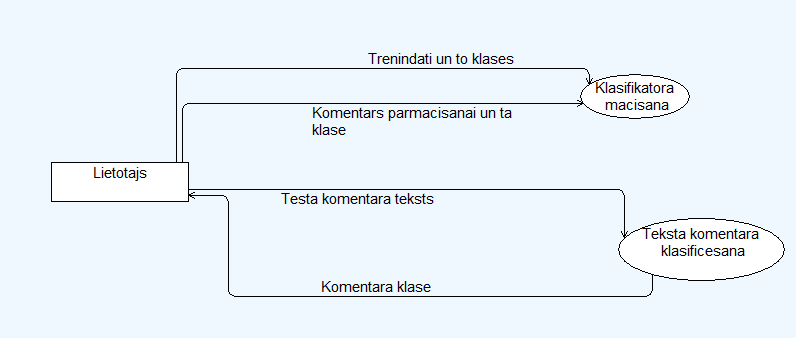


Lai saglabātu datu precizitāti, varbūtības tiek glabātas logaritmu formā.

**Šķērsvalidācija**

Lai noteiktu programmas precizitāti tika izveidota šķērsvalidācija. Treniņdati tika sajaukti un sadalīti 10 daļās. Ejot pēc kārtas viena datu daļa tika pieņemta kā testu datu kopa, bet pārējie 90% tika pieņemti kā klasifikatora treniņdati. Tika noteikta katra pārbaudījuma precizitāte, un pēc tam tika paņemta precizitāšu vidējā aritmētiskā vērtība.





# KVALITĀTES NODROŠINĀŠANA

Lai nodrošinātu augstu programmatūras kvalitāti, tika veiktas sekojošas darbības:

Programmatūras dokumentācijas veikšana pēc Latvijas valsts republikas

standartiem:

LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis

LVS 72:1996 Ieteicamā prakse programmatūras projektējuma aprakstīšanai

Programmas kods tika rakstīts pēc vienota stila, panākot vieglu lasāmību un pārskatāmību.

Veidojot pirmkodu, tika izmantoti Latvijas Universitātes Datorikas fakultātes

kursā „Kvalifikācijas darbs” aprakstītos programmēšanas stila ieteikumi, iekļaujot:

Lietojiet komentārus programmas sākumā, lai dokumentētu programmas būtiskās īpašības

Lietojiet komentārus programmas tekstā, lai uzlabotu lasāmību

Rakstiet īsas programmas, funkcijas (apmēram vienu lappusi garas)

Izvēlieties paskaidrojošus un pielāgotus vārdus mainīgā veidam. Izvēlieties

pēc iespējas īsus mainīgo vārdus, tomēr, lai būtu saprotama jēga;

Lokālie mainīgie, īpaši skaitītāji var būt ar vienburta vārdiem – i, j, k utt.

Funkciju vārdus vēlams veidot no darbības vārda, kam, iespējams, seko lietvārds

Klašu vārdus sāciet ar lielo burtu

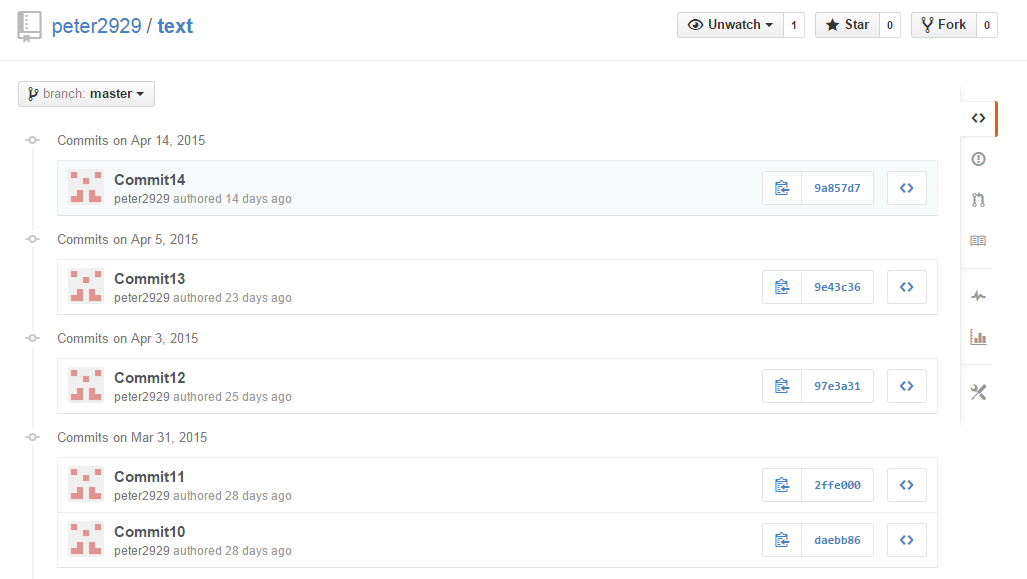
Aizvietojiet izteiksmes, kas atkārtojas, ar funkciju izsaukumiem

# Konfigurācijas pārvaldība

Programmatūras pirmkoda versiju kontrolei tika izmantota versiju kontroles sistēma Git, kodu glabājot savā “github” repozitorijā, papildus, dublējot failus arī uz ārējo cieto disku.

Izmaiņas tika “iestumtas” (git push) apmēram reizi nedēļā.

Katram failam ir iespējams apskatīt visas faila glabātās versijas, kopā ar komentāriem.



# TESTĒŠANAS DOKUMENTĀCIJA

Šis dokuments ir paredzēts, lai aprakstītu sistēmas testēšanas norisi un tās iegūtos rezultātus. Šī detalizētā vienībtestēšana, tika veikta projekta izstrādes beigās, taču, katras funkcijas vienībtestēšana, tika veikta arī projekta izstrādes procesa laikā. Tika mēģināts veikt testēšanu, lielākoties, pamatojoties uz Latvijas valsts standarta LVS 70:1996, “Programmatūras testēšanas dokumentācija” ieteikumiem.

Par sekmīgu testa rezultātu uzskata tādu, kas pilnībā atbilst sagaidāmajam rezultātam. Ja ir novirzes no sagaidāma rezultāta, tās ir jāatzīmē testēšanas rezultātos un radušās kļūdas jāizlabo.

**Testējamās raksturiezīmes**

Veicot sistēmas testēšanu tika pārbaudītas sekojošas raksturiezīmes:

Lietotāju saskarne – elementu izskats, lietotāja puses skriptu izpilde.

Funkcionalitāte – datu ievade, kļūdainu datu un izņēmu situāciju apstrāde.

Integrācija – lietotāja tiesību pārbaude.

Drošība – aizsardzība pret SQL injekcijām un starpvietņu skriptēšanu.

Par sekmīgu testa rezultātu uzskata tādu, kas pilnībā atbilst sagaidāmajam rezultātam. Ja ir novirzes no sagaidāma rezultāta, tās ir jāatzīmē testēšanas rezultātos un radušās kļūdas jāizlabo.

**Vienībtesti**

**Lietotāja reģistrācija sistēmā**

|  |  |  |  |
| --- | --- | --- | --- |
| **Testa** **nr.** | **Apraksts** | **Sagaidāmais rezultāts** | **Rezultāts** |
| 1. | Netiek aizpildīts kāds no obligāti aizpildāmajiem laukiem | Tiek pieprasīts aizpildīt visus laukus | Sekmīgs |
| 2. | Tiek ievadītas paroles, kuras nesakrīt | Tiek izvadīts paziņojums  „Paroles nesakrīt!” | Sekmīgs |
| 3. | Tiek ievadīta parole ar mazāk nekā 8 simboliem | Tiek izvadīts paziņojums „Parolei jāsatur vismaz 8 simbolus!” | Sekmīgs |
| 4. | Dati tiek ievadīti korektos  formātos un garumos | Dati tiek akceptēti, un  lietotāja dati tiek saglabāti  sistēmā | Sekmīgs |
| 5. | Tiek ievadīts lietotājvārds, ar mazāk nekā 3 simboliem | Tiek izvadīts paziņojums „Lietotājvārdam jāsatur vismaz 3 simbolus!” | Sekmīgs |
| 6. | Tiek ievadīts jau eksistējošs lietotājvārds. | Tiek izvadīts paziņojums „Šis lietotājvārds jau ir aizņemts!” | Sekmīgs |

**Lietotāja pieslēgšanās sistēmā**

|  |  |  |  |
| --- | --- | --- | --- |
| **Testa** **nr.** | **Apraksts** | **Sagaidāmais rezultāts** | **Rezultāts** |
| 1. | Netiek ievadīts lietotājvārds vai parole | Paziņojums, ka lietotājvārds  vai parole nav pareizi ievadīti | Sekmīgs |
| 2. | Dati tiek ievadīti pareizi | Dati tiek akceptēti, un  lietotājs tiek pieslēgts sistēmai | Sekmīgs |

**Treniņdatu pievienošana**

|  |  |  |  |
| --- | --- | --- | --- |
| Testa nr. | Apraksts | Sagaidāmais rezultāts | Rezultāts |
| 1. | Tiek ievadīti treniņdati ar klasi “Negatīvs” | Paziņojums, ka dati veiksmīgi pievienoti klases “Negatīvs” datu kopai | Sekmīgs |
| 2. | Tiek ievadīti treniņdati ar klasi “Neitrāls” | Paziņojums, ka dati veiksmīgi pievienoti klases “Neitrāls” datu kopai | Sekmīgs |
| 3. | Ievaddatu garums ir mazāks par 2 simboliem | Tiek izvadīts paziņojums „Lūdzu ievadiet treniņdatus!” | Sekmīgs |

**Komentāru klases noteikšana ar „Chrome” paplašinājumu**

|  |  |  |  |
| --- | --- | --- | --- |
| Testa nr. | Apraksts | Sagaidāmais rezultāts | Rezultāts |
| 1. | Klases noteikšana | Negatīvi komentāri tiek iekrāsoti sarkanā krāsā, bet neitrāli – zaļajā. | Sekmīgs |
| 2. | Komentāra klases maiņa | Nospiežot pogu pie komentāra, lapa tiek atjaunota un dotais komentārs tiek pārkrāsots prētējā krāsā. | Sekmīgs  Piezīme: ja komentārs ir ļoti īss mēdz gadīties, ka klase paliek nemainīta |
| 3. |  |  | Sekmīgs |

# PROGRAMMATŪRAS PROJEKTA ORGANIZĀCIJA

# DARBIETILPĪBAS NOVĒRTĒJUMS

# IZMANTOTĀ LITERATŪRA

# Pielikumi

**Pirmkoda paraugs**

d = "";

function req(c)

{

var xmlhttp = new XMLHttpRequest();

xmlhttp.onreadystatechange = function()

{

if (xmlhttp.readyState == 4 && xmlhttp.status == 200)

{

f = xmlhttp.responseText.trim();

f = f.split(' ');

var k;

var message\_count = document.getElementsByClassName('message').length;

for(i=0; i<message\_count; i++)

{

k = document.getElementsByClassName('message')[i].innerHTML;

if(f[i] == "negative")

{

document.getElementsByClassName('message')[i].style.color = "red";

//if(document.getElementsByClassName('message')[i].id != 'a\_'+i+'\_b') //if(typeof document.getElementById('a\_'+i).value != 'undefined')

//{

/////document.getElementsByClassName('message')[i].innerHTML += "<br><input type=\"submit\" id='a\_"+i+"' class='btn btn-primary' value='Tomēr ir neitrāls'><br>";

document.getElementsByClassName('message')[i].innerHTML += "<form method=post action=http://localhost/text/b.php><input name=change\_to value=\"neutral\" type=hidden><input type=hidden name=cn value='"+i+"'><input name=source\_url value='"+window.location.href+"' type=hidden><br><input name=com value='"+k+"' type=hidden><input type=\"submit\" id='a\_"+i+"' class='btn btn-primary' value='Tomēr ir neitrāls'><br></form>\n";

///document.getElementById('a\_'+i).onclick = chl('a\_'+i);

//////////////////////////////////////////////////document.getElementsByClassName('message')[i].innerHTML += "<form method=post action=http://localhost/text/b.php><input name=change\_to value=\"neutral\" type=hidden><input type=hidden name=cn value='c\_"+i+"'><a name='c\_"+i+"'></a><input name=source\_url value='"+window.location.protocol+"//"+window.location.host+""+window.location.pathname+"' type=hidden><br><input name=com value='"+k+"' type=hidden><input type=\"submit\" id='a\_"+i+"' class='btn btn-primary' value='Tomēr ir neitrāls'><br></form>\n";

document.getElementsByClassName('message')[i].id = "a\_"+i+"\_b";

//}

}

else if(f[i] == "neutral")

{

document.getElementsByClassName('message')[i].style.color = "green";

//if(document.getElementsByClassName('message')[i].id != 'a\_'+i+'\_b')

//{

//document.getElementsByClassName('message')[i].innerHTML += "<br><input type=\"submit\" id='a\_"+i+"' class='btn btn-primary' value='Tomēr ir negatīvs'><br>";

document.getElementsByClassName('message')[i].innerHTML += "<form method=post action=http://localhost/text/b.php><input name=change\_to value=\"negative\" type=hidden><input type=hidden name=cn value='c\_"+i+"'><input name=source\_url value='"+window.location.href+"' type=hidden><br><input name=com value='"+k+"' type=hidden><input type=\"submit\" id='a\_"+i+"' class='btn btn-primary' value='Tomēr ir negatīvs'><br></form>\n";

///document.getElementById('a\_'+i).onclick = chl('a\_'+i);

document.getElementsByClassName('message')[i].id = "a\_"+i+"\_b";

//}

}

}

//alert('fsdfds');

}

}

xmlhttp.open("POST", "http://localhost/text/a.php", true);

xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");

xmlhttp.send("b="+c);

}

**function chl(h)**

{

return function()

{

var xmlhttp2 = new XMLHttpRequest();

xmlhttp2.onreadystatechange = function()

{

if(xmlhttp2.readyState == 4 && xmlhttp2.status == 200)

{

}

}

xmlhttp2.open("POST", "http://localhost/text/b.php", true);

xmlhttp2.setRequestHeader("Content-type","application/x-www-form-urlencoded");

var change\_label\_to;

if(document.getElementById(h).value == 'Tomēr ir neitrāls') change\_label\_to = "neutral";

else if(document.getElementById(h).value == 'Tomēr ir negatīvs') change\_label\_to = "negative";

var k = document.getElementById(h+'\_b').innerHTML.split("<br><input ");

xmlhttp2.send("change\_label\_to="+change\_label\_to+"&b="+k[0]);

//alert(document.getElementById(h+'\_b').innerHTML);

///if(typeof document.getElementById('a\_0').value != 'undefined') document.getElementsByClassName('message')[0].innerHTML += "<br><input type=\"submit\" id='a\_0' class='btn btn-primary' value='Tomēr ir neitrāls'><br>";

//if(document.getElementById(h).value == 'Tomēr ir neitrāls')

//document.getElementById(h).value = 'afdfsfs';

}

}

var s = document.getElementsByClassName('message').length;

for(i=0; i<s; i++)

{

d += document.getElementsByClassName('message')[i].innerText;

d += "DELIMITER";

}

if(s>0) t(d);

# Secinājumi