RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

Lietišķo datorsistēmu institūts

Lietišķo datorzinātņu katedra

**Ivars Naglis**

**Tiešsaistes apmācības sistēmas izstrāde**

**Bakalaura darbs**

Darba vadītāja

Dr.sc.ing, prof.

Oksana Ņikiforova

Rīga, 2014

###### Uzdevums

**Anotācija**

Šī darba nosaukums ir Anotācijas teksts. Anotācijas teksts. Anotācijas teksts. Anotācijas teksts. Anotācijas teksts. Anotācijas teksts. Anotācijas teksts. Anotācijas teksts. Anotācijas teksts. Anotācijas teksts. Anotācijas teksts. Anotācijas teksts. Anotācijas teksts. Anotācijas teksts. Anotācijas teksts. Anotācijas teksts.

Dokumentā ir \_\_ lappuses, \_\_\_ attēli, \_\_\_ tabulas, \_\_\_ pielikumi un \_\_\_ nosaukumu informācijas avoti.

**Abstract**

Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.Text of the abstract.

Bachelor (or master) thesis contains \_\_\_ pages, \_\_\_ figures, \_\_\_ tables, \_\_\_ appendixes. Bibliography includes \_\_\_\_ sources.

**Saturs**

[Ievads 7](#_Toc406447005)

[1. zinātniskā daļa 8](#_Toc406447006)

[1.1. Problēmas apraksts 8](#_Toc406447007)

[1.2. Tiešsaistes sistēmu salīdzinājums 9](#_Toc406447008)

[1.2.1. New Horizons Latvia 9](#_Toc406447009)

[1.2.2. Baltijas Datoru akadēmija (BDA) 10](#_Toc406447010)

[1.2.3. Codecademy 11](#_Toc406447011)

[1.2.4. Treehouse 12](#_Toc406447012)

[1.2.5. Lynda.com 13](#_Toc406447013)

[1.2.6. Tuts+ 13](#_Toc406447014)

[1.2.7. KhanAcedamy 14](#_Toc406447015)

[1.2.8. Linux Academy 15](#_Toc406447016)

[1.3. Tehnoloģiju izvēle 16](#_Toc406447017)

[1.3.1. Servera izvēle 16](#_Toc406447018)

[1.3.2. Programmēšanas valodas izvēle 19](#_Toc406447019)

[1.4. Ietvara izvēle 20](#_Toc406447020)

[1.4.1. Laravel 4 ietvars 21](#_Toc406447021)

[1.4.2. Ruby on Rails ietvars 21](#_Toc406447022)

[1.4.3. Django ietvars 21](#_Toc406447023)

[1.4.4. Java Play ietvars 21](#_Toc406447024)

[1.4.5. Ietvaru salīdzinājums 21](#_Toc406447025)

[1.5. Datubāzes izvēlne 22](#_Toc406447026)

[1.5.1. Relācijas datubāzu salīdzinājums 23](#_Toc406447027)

[1.6. Servera tehnoloģiju izvēle 24](#_Toc406447028)

[2. praktiskā daļa 25](#_Toc406447029)

[2.1. Sistēmas prasības 25](#_Toc406447030)

[2.1.1. Produkta funkcijas 25](#_Toc406447031)

[2.1.2. Funkcionālās prasības 27](#_Toc406447032)

[2.2. Programmatūras projektējuma apraksts 41](#_Toc406447033)

[2.2.1. Datu bāzes struktūra 41](#_Toc406447034)

[3. Sistēmas ieviešana un testēšana 45](#_Toc406447035)

[3.1. Git versiju kontrole sistēma 45](#_Toc406447036)

[3.2. Vienību testi 46](#_Toc406447037)

[3.3. TravisCI integrācijas testi 46](#_Toc406447038)

[3.4. Lietotnes ieviešana 47](#_Toc406447039)

[Nobeigums 48](#_Toc406447040)

[Pielikumi 49](#_Toc406447041)

[Literatūra 52](#_Toc406447042)

Ievads

Mūsdienās pateicoties straujai tehnoloģiju attīstībai internets ir pieejams mobilajās iekārtās un datoros. Šīs tehnoloģijas attīstības ir ļāvusi izveidot jauna tipa biznesus un vienkāršot komunikāciju un zināšanu ieguves procesu globāli. Starp jaunajiem biznesu veidiem, kurš tiek veidots balstoties uz tiešsaistes tehnoloģijām ir tiešsaistes apmācība. Šis ir perspektīva nozare, jo tai ir paredzēts popularitātes un pielietojuma kāpums [http://elearningindustry.com/top-10-e-learning-statistics-for-2014-you-need-to-know] [https://www.docebo.com/landing/contactform/elearning-market-trends-and-forecast-2014-2016-docebo-report.pdf].

Tiešsaistes apmācības popularitātes augšanu nodrošina sekojoši faktori:

* ļauj vienkāršot materiālu izdošanas procesu un piedāvāt materiālus ātrāk potenciālajiem klientiem
* ļauj ātrāk iegūt zināšanas un apgūt labākās izstrādes prakses, jo informāciju ir iespējams piegādāt destilēta veidā, kura saturētu tikai galvenās idejas
* šo apmācības veidu stimulē patstāvība mobilo tehnoloģiju attīstības
* klients nav piesaistīts vienai noteiktai platformai un laika, kurā ir iespējams iegūt nepieciešamās zināšanas.

Kā arī apmācības procesu lietotājs var pielāgot pēc saviem mācīšanās paradumiem.

Šis apmācības veiks nav paredzēts, lai aizstātu pilnībā tradicionālo apmācību, bet šīs apmācības metodes mērķis ir sniegt papildus zināšanas un uzspodrināt jau esošās. Tieši šādiem mērķiem lielas kompānijas izmanto tiešsaistes apmācības sistēmas.

Mūsdienās projekta ātrai izstrādei tiek izmantoti ietvari, šie ietvari nodrošina gatavus elementus lietotņu izstrādei.

Django ietvara atpazīstamākie lietotāji ir Instragram, Mozilla, OpenStack

**Bakalaura darba mērķis** ir tiešsaistes apmācības sistēmas izstrāde, izmantojot atbilstošās tehnoloģijas bastoties uz problēmas vidi un ar to saistītām prasībām.

Lai sasniegtu izvirzītu mērķi darbam ir nostādīti šādi **uzdevumi**:

* Aplūkot Latvijā un pasaulē pieejamās tiešsaistes apmācības sistēmas
* Definēt lietotnes prasības izstrādei un aprakstīt tās programmatūras prasību specifikācijā.
* Izveidot sistēmas arhitektūru un izstrādāt lietotnes projektējuma aprakstu.
* Izstrādāt lietotnes prototipu
* Demonstrēt sistēmas pamata funkcionalitāti strādājošas sistēmas ekrānšāviņos.

<<Satura izklāsts>>

# zinātniskā daļa

Šajā daļā būs aprakstīts kāpēc tiek izvēlēti konkrētie tehnoloģiskie risinājumi, lai izveidotu lietotni.

## Problēmas apraksts

Tiešsaistes apmācības sistēmas izstrādei ir nepieciešams izvēlēties tehnoloģijas, kuras ļautu pēc iespējams ātrāk un vienkāršāk izstrādāt, uzturēt lietotni.

Tiešsaistes apmācība ir elektronisko mēdiju, apmācības un informācijas tehnoloģiju apvienojums, kurš tiek izmantots apmācību procesā. Tiešsaistes apmācībai tiek izmantoti dažādi mēdiji, piem., kā teksts, audio, bildes un video.

Tiešsaistes apmācība var notikt klasē un ārpus tās, tā var būt organizēta asinhrona tipa apmācība vai arī instruktora vadīta lineārs apmācības process. Tiešsaistes apmācība galvenokārt ir piemērota tālāk mācībai un dinamiskai apmācībai, bet šo apmācības metodi var izmantot kopā ar klasisku apmācības metodi.

Tiešsaistes apmācība iekļauj un ir sinonīma šādām apmācības metodēm:

* datoru bāzētas instrukcijas
* datora vadītas apmācības
* datoru bāzētas apmācības
* tiešsaistes bāzētas apmācības
* tiešsaistes apmācība
* virtuālā apmācības vide
* mobilā apmācība

Šie alternatīvie nosaukumi uzsvērt konkrētu pieeju, komponentu vai piegādes metodi, bet apzīmē tiešsaistes apmācības metodi. Piemēram, mobilās apmācības uzsver mobilo tehnoloģiju izmantošana, bet citādi šī apmācības metode neatšķiras no tiešsaistes apmācības.

Tiešsaistes apmācību sistēma priekšrocības:

* ļauj samazināt izmaksas, kuras ir saistītas ar personāla piesaistīšanu un apmācību vadīšanu
* ļauj uzkrāt zināšanas pieejamā veidā
* vienkārši paplašināms apmācību loku
* mobilo iekārtu attīstība – iekārtas nodrošina interneta pieslēgumu

izveidotā materiāla ātrāka piegāde mērķauditorijai salīdzinājuma ar iespiestām grāmatām

nodrošina augstāku interaktivitāti ar klientu, lietotāju, viesi salīdzinājumā ar grāmatām un rakstiem, jo video ļauj soli pa solim izpētīt autora darbības.

93 % uzņēmumu vadītāju plāno palielināt un uzturēt apmācību budžetu un 85% jau eksistē mobilo apmācību stratēģija vai arī drīzumā tāda tiks ieviesta [http://www.mvc.astd.org/Resources/Documents/CEO%27s%20and%20Learning.pdf]

84% uzņēmumu izmanto tiešsaistes apmācību uzņēmuma darbinieku apmācību un kvalifikācijas uzlabošanai [http://elearningindustry.com/kineo-e-learning-in-the-enterprise-survey-results-2013-infographic]

10 miljoni studentu ir piedalījušies vismaz vienā lielā tiešsaistes kursā [http://www.bestcollegesonline.org/moocs/]

ASV un Eiropa nodrošina 70% tiešsaistes apmācības pieprasījumu, bet Āzijas un klusā okeāna valstīm ir paredzēta 20% izaugsme [http://elmezine.epubxp.com/title/55545/28]

Uzņēmumu, kuri piekopj apmācību veikšanu ir par 46% lielāka iespēja būt tirgus līderiem un veikt inovācijas [http://www.bersin.com/News/Content.aspx?id=12521]

Latvijas mērogā tiešsaistes apmācību piedāvā:

* BDA (Baltijas Datoru Akadēmija),
* New Horizons Latvia

Dažas no pasaulē populārākajām tiešsaistes apmācības sistēmām:

* Lynda.com
* Treehouse
* Linux Academy
* Codecademy
* Tuts+
* Pluralsight

## Tiešsaistes sistēmu salīdzinājums

Šajā sadaļā tiks salīdzinātas Latvijā un pasaulē pieejamās un populārākās tiešsaistes apmācības sistēmas.

### New Horizons Latvia

Mācību centrs „New Horizons Latvia” pieder starptautiskajam „New Horizon Learinig Centres” tīklam, kurš ir viens no pasaules līderiem apmācību jomā, kuram pieder 300 mācību centri 70 pasaules valstīs.

New Horizon piedāvā apmācības sekojošos virzienos:

* Adobe
* Java
* Oracle
* Cisco
* PHP
* Mobīlo lietotņu izstrāde
* SQL serveris
* Windows serveris
* Linux

Kā papildus iespēju bez apmācības New Horizon piedāvā iespēju iegūt dažādus sertifikātus, piem., MCP, MCITP, MCTS, MCPD, MCSA, MCSE, MCDBA, MCAD, MCSD, .NET, CompTIA A+, Network+, Security+ un citus.

New Horizion Latvi piedāvā vairākas tiešsaistes apmācības metodes:

* Virtuālā klase (Online LIVE)
* Video ar instruktora atbalstu (Online PREMIUM)
* E – kursu bibliotēka (Online ANYTIME)

**Virtuālās klases apmācības** veidā ar interneta starpniecību tiek nodrošinātas interaktīvu nodarbību tiešraides starp studentu un viņa pasniedzēju.

Lai arī pasniedzēji un studenti atrodas dažādos pilsētas, valsts vai pat pasaules nostūros, viņi aktīvi piedalās diskusijās, strādā laboratorijās, vienlaicīgi skatās un analizē vienus un tos pašus dokumentus, kā arī dalās virtuālās grupās.

New Horizons piedāvā vairāk nekā 150 lekciju kursus ar virtuālajām laboratorijām tiešraidē. Lielākā daļa tiešraides piedāvājumu tiešsaistē ir autorizētas programmas, tostarp: Microsoft, Cisco, Citrix, CompTIA, EC Council, Novell and Planet 3.

**Video un instruktora atbalsts** kaut kas par šo apmācības veidu

**E – kursu bibliotēka** nodrošina iespēju patstāvīgi organizēt apmācības procesu

### Baltijas Datoru akadēmija (BDA)

Baltijas Datoru akadēmija jeb BDA ir lielākais datorapmācības centrs Baltijas valstīs, kas ietilpst IT uzņēmuma Lattelecom grupas sastāvā. BDA veiksmīgi darbojas Latvijas tirgū kopš 1994. gada.

BDA piedāvātie pakalpojumi:

* kvalifikācijas celšanas kursus IT profesionāļiem,
* kvalifikācijas celšanas kursus ikvienam datora lietotājam,
* izstrādā pielāgotas mācību programmas un materiālus,
* konsultē klientus par atbilstošo apmācību risinājumu izvēli,
* īsteno darbinieku zināšanu novērtēšanu pirms un pēc apmācībām,
* palīdz izvērtēt darbinieku apmācību izdevumus, kas saistīti ar jaunas programmatūras ieviešanu uzņēmumā / iestādē / organizācijā.

BDA strādā dinamiski un proaktīvi, piedāvājot katra klienta individuālām vajadzībām atbilstošus pakalpojumus.

BDA apmācības virzieni:

* Adobe
* Cisco
* Horizon
* IT drošība
* ITIL
* Java
* Microsoft
* Novell/Linux
* Oracle
* Projektu vadība
* Telekomunikācijas
* VMware

BDA priekšrocības:

* plašs pieejamo apmācību spektrs
* LinkedIn profilu atsauces pasniedzējiem
* BDA tiešsaistes apmācības trūkumi:
* tiešsaistes apmācības notiek pēc noteikta grafika

BDA tiešsaistes apmācības trūkumi:

* Augstas apmācības kursu cenas
* Novecojušu tehnoloģiju video bibliotēkas
* Apmācības notiek pēc iepriekšnoteikta grafika

### Codecademy

Codecademy ir interaktīva tiešsaistes platforma, kura nodrošina iespēju apgūt programmēšanu izmantojot kādu no vairāk atbalstītajām programmēšanas valodām:

* Python,
* PHP,
* jQuery,
* JavaScript,
* Ruby,
* kā arī tiek atbalstīta HTML un CSS

Codecademy tika dibināta 2011. gadā. To dibināja Zach Sims un Ryan Bubinski.

Lietotāju motivēšanai attīstīties, tiek lietota progresa attēlošanas sistēma, šī sistēma ir implementētā izmantojot badges, kuri tiek iegūti pabeidzot kādu noteiktu uzdevumu.

Codecademy piedāvā iespēju sadarboties iesācējiem un profesionāliem izstrādātājiem mijiedarboties, lai viens otram palīdzētu.

### Treehouse

Treehouse ir interaktīva apmācības platforma, kur apmāca studentus izstrādāt web vai mobilās lietotnes izmantojot šāds tehnoloģijas:

* HTML
* CSS
* WordPress
* Java
* Javascript
* PHP
* Ruby
* Objectice-C

Lietotnē ir izvietoti īsi video, interaktīvi koda izaicinājumi viktorīnas un žetonus lietotāja profilam.

Treehouse tika izlaists 2011 novembrī.

Threehouse apmācības pakalpojumu popularizēšanai publicē video www.youtube.com vietnē, kuros tiek apskatītas jaunākās izstrādes tendencēm, rīki un tehnoloģijām.

Treehouse piedāvā divus pakas plānus „Basic” un „Pro”, kuri atšķiras ar piedāvāto funkcionalitāti, tiek piedāvāts arī bezmaksas periods, bet tas ilgst tikai 14 dienas.

Maksājuma plāni:

* bezmaksas 14 dienas (nepieciešams precizēt)
* „Basic”
* „Pro”

„Basic” maksājuma plāns piedāvā:

* piekļuve vairāk nekā 1000 video
* praktiskie uzdevumi
* dalībnieku forums

„Pro” piedāvājums:

* „Basic” nodrošināto piedāvājumu
* industrijas profesionāļu runas
* ekskluzīvas intervijas un semināri

### Lynda.com

Lynda.com ir tiešsaistes apmācības kompānija, kura palīdz apgūt lietotnes, dizainu un biznesa iemaņas. Lynda.com servisa lietotājiem ir pieejama plaša bibliotēka ar augstas kvalitātes video materiāliem. Jauni apmācības kursi tiek pievienoti katru nedēļu.

Lynda.com tika dibināts 1995. gadā.

Lynda.com apmācības sistēmas priekšrocības ir plašā bibliotēka un plašais apmācību materiālu loks, apmācības materiāli ir pieejami ar par tādām tēmām, kā mārketings, fotogrāfijas, biznesa u.c. Lynda.com vietnē regulāri tiek pievienoti jauni materiāli.

Abonementu plāni:

* „Basic” (mēneša/gada maksājumi)
* „Premium” (mēneša/gada maksājumi)

Basic (mēneša / gada maksājumi):

* piekļuve video galerijai
* mobilo iekārtu lietotne

Preminum (mēneša maksājumi):

* Basic maksājuma plānā iekļautais piedāvājums
* projektu failu lejupielāde

Premium (gada maksājums):

* Premium mēneša maksājuma piedāvājums
* projektu failu lejupielāde uz mobilajām iekārtām

### Tuts+

Projekta sākums ir 2007 gadā, kuru uzsāka pamācības par Photosop lietotnes izmantošanu, pēc tām tīkls tika sadalīts 15 apmācības vietnēs, kur katrai ir savs brands un editor komanda. 2011 gadā Tuts+ tika pievienots Preminum piedāvājums, kurš ļauj piekļūt video apmācībām, E-grāmatām. 2014. gadā vietnes tika apvienotas vienā vietnē.

Tuts+ ir vienos no vadošajiem tiešsaistes apmācības un pašorganizētās apmācības izdevējiem, kurš ir orientēts uz tiešsaistes lietotņu izstrādei.

Maksājumi plāns:

* Monthly $15 (mēneša)
* Yearly $180 (gads)
* Yearly Pro $360 (gads ar papildus iespējām)

Monthly (mēneša) maksājuma plāns:

* Pieeja kursiem
* mobilajām iekārtām piemērots video
* jauni kursi katru nedēļu
* netiek izvietotas reklāmas

Yearly (gada) maksājuma plāns:

* Monthly (mēneša) maksājuma plāna piedāvātie pakalpojumi
* Video lejupielāde
* 5 e - grāmata lejupielāde mēnesī
* choose 2 subsciber benefits per year

Yearly Pro maksājumu plāns:

* Yearly (gada) plāna piedāvājums
* pieeja e-grāmatām (bez ierobežojumiem)
* access to all subscriber benefits
* $25 kretīts Envator Market vietnē
* $25 kredīts Envato Studio vietnē

Studentiem ir pieejama 50% atlaide gada abonementam

### KhanAcedamy

Khan Academy ir organizācija, kuras tika izveidota 2006 gadā un tās mērķis ir mainīt apmācības procesu sniedzot bezmaksas augstas kvalitātes apmācības jebkuram lietotājam.

Khan Academy piedāvā interaktīvus testus, šie testi tiek ģenerēti pēc nejaušības principa, kas nodrošina to, ka piedāvātais testa piemērs būs unikāls.

Vietnē izvietotie materiāli ir pieejami jebkuram bezmaksas.

Khan Academy piedāvā iespēju iegūt statistiku par lietotāju, šo informāciju var arī apskatīt klases griezumā. Tas ļauj smalkā iegūt informāciju par to kā studenti apgūst informāciju un problēmu punktus, kuriem būtu nepieciešams pievērst uzmanību.

Khan Academy izmantot adaptīvus novērtēšana vidi, kura ļauj sāk apmācības procesu ar vienkāršāko vielu un apgūt informāciju, kuru nepieciešams atjaunot. Katras problēmas ģenerācija notiek izmantojot nejaušības principu, tā padarot katru apmācības procesu unikālu. Sistēmā ir iestrādāt padomu došanas funkcionalitāte, kura ļauj atrisināt problēmu sadalot to nepieciešamajos soļos, kā arī ir pieejami video materiāli.

Khan Academy piedāvā plašu materiālu bibliotēku, kurā atrodas matemātikas, zinātnes, ekonomikas un finanses, māksla, datorapmācība un sagatavošanās testiem.

Lietotāju motivēšanai Khan Academy pielieto žetonu sistēmu, šos žetonus ir iespējams iegūt pabeidzot sekmīgi apmācības kursus.

### Linux Academy

Maksājumu plāni:

Tiešsaistes apmācības sistēmu salīdzinājums

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | newhorizons.lv | BDA | Lynda.com | Treehouse.com | Pluralsight | Tuts+ | codeacedamy |
| Maksas video | + | + | + | + | + | + | + |
| Bezmaksas video |  |  |  |  |  |  |  |
| Testi |  |  |  |  |  |  |  |
| Blogs |  |  |  |  |  |  |  |
| Kodu piemēri |  |  | + | + |  | + |  |
| Kodu interpretos pārlūkā |  |  |  |  |  |  | + |
| Forums |  |  |  |  |  |  |  |
| Maksas materiāli | + | + | + | + | + | + | + |
| Jauni materiāli |  |  | + | + | + | + | + |

## Tehnoloģiju izvēle

Šajā apakšnodaļā tiks apskatītas un salīdzinātas potenciālas tehnoloģijas, kuras būtu iespējams izmantota lietotnes izstrādei.

### Programmēšanas valodas izvēle

Šajā darba sadaļā tiks aplūkotas izstrādes valodas, kuras būtu iespējams izmantot lietotnes izstrādei.

[Top 10 Programming Languages (Stephen Cass) http://spectrum.ieee.org/computing/software/top-10-programming-languages#]

Kā potenciālas izstrādes valodas tika izraudzītas Java, Python, Ruby, PHP. Sākotnējai izstrādes valodu atlasīšanai tika izmantota interneta vietņu valodu statistikas dati [IEEE Spectrum Web Top 10 (Craig Buckler) http://www.sitepoint.com/best-programming-language-learn-2014-mid-year-update/],

Nepieciešams kaut kas par to, kā pārklājās šīs valodas ar web hosting piedāvājumiem.

## Ietvara izvēle

Šajā nodaļā tiks apskatīti ietvari, kurs potenciāli būtu iespējams izmantot lietotnes izstrādei. Ietvari tiks salīdzināti pēc to piedāvātājam funkcijām, izstrādes valoda šajā gadījumā ir sekundāra, jo mērķis ir izmantot ietvaru, kurš piedāvā vis plašāko funkcionalitāti.

Interneta lietotnes ietvars ir programmatūras ietvars, kurš ir paredzēts interneta vietņu, lietotņu un servisu izstrādei. Ietvara mērķis ir atvieglot atkārtotu aktivitāšu veikšanu izstrādē. Piemēram, daudzi ietvari nodrošina sesiju vadības funkcionalitāti, vienotu projekta izstrādes metodi, abstrakcijas līmeni darba ar datubāzi, lai optimizētu izstrādes procesu un atvieglotu dažādu datubāzu implementācijas detaļas.

Ietvara priekšrocības:

* samazina lietotnes izstrādes laiku un ieguldāmos resursus (ja pielietoto ietvaru pārzina)
* ietvars nodrošina standarta sistēmu, kura ļauj izstrādāt lietotni neuztraucoties par zemākā līmeņa funkcionalitātes izstrādi.
* ļauj ieguldīt vairāk laika lietotnes prasību izstrādē un neveltot to vides un rīku konfigurācijai
* ietvars uzspiež izstrādes šablona izmantošanu, kurš padara kodu vieglāk uztveramu un vienkāršāk paplašināmu nākotnes vajadzībām.
* ietvars nodrošina datu, biznesa loģikas un lietotāja interfeisa nodalīšanu, kas padara kodu vienkāršāku un vienkāršāk paplašināmu

Ietvara pielietošanas trūkumi:

* jauniem lietotājiem ir nepieciešams apgūt ietvara abstrakcijas, uzbūvi un pielietot konceptus lietotņu izstrādei
* viena koda bāze dažādu problēmu risināšanai
* ievari sākotnēji tika atlasīti pēc šādiem kritērijiem, popularitāte/komūnas atbalsts, vienību testu atbalsts, datubāzes migrācijas atbalsts, plaša funkcionalitāte un dokumentācija

Ietvari tiks aplūkoti sekojošām izstrādes valodām:

* PHP
* Ruby
* Python
* Java

Ievērojot sākotnējos izvirzītos kritērijus tika atlasīti šādi ietvari:

* PHP -> Laravel [cc]
* Python -> Django[[1]](#footnote-2)
* Ruby -> Ruby on Rails [cc]
* Java -> Play

### Laravel 4 ietvars

Laravel ir PHP izstrādes valodas ietvars, kurš tika publicēts 2012. gada februārī. Laravel ir bezmaksas atvērtā koda ietvars ar kura palīdzību ir iespējams izstrādāt vietnes. Ietvars ir licenzēts izmantojot MIT licenci. Koda struktūras organizēšanai tiek izmantots MVC šablons.

Laravel 4 ietvara priekšrocības:

* Izstrādāts izmantojot vienu no populārākajām izstrādes valodām PHP [kaut kāda statistika]
* Plašs datubāzu atbalsts
* Plaša un labi organizēta dokumentācija
* Composer pakotņu kontroles sistēma
* Pārbaudītu komponenšu izmantošana

### Ruby on Rails ietvars

Ruby on Rails ir Ruby izstrādes valodas ietvars

Ruby on Rails, sometimes known as "RoR" or just "Rails," ir atvērtā koda ietvars, kurš ir paredzēts interneta vietņu izstrādei izmantojot Ruby izstrādes valodu, kura ir objektorientēta (OOP) izstrādes valoda.

The principle difference between Ruby on Rails and other frameworks for development lies in the speed and ease of use that developers working within the environment enjoy. Changes made to applications are immediately applied, avoiding the time consuming steps normally associated with the web development cycle. According to David Geary, a Java expert, the Ruby-based framework is five to 10 times faster than comparable Java-based frameworks. In a blog posting, Geary predicted that Rails would be widely adopted in the near future.

Rails is made up of several components, beyond Ruby itself, including:

* Active record, an object-relational mapping layer
* Action pack, a manager of controller and view functions
* Action mailer, a handler of email
* Action web services
* Prototype, an implementer of drag and drop and Ajax functionality

Rails can run on most Web servers that support CGI. The framework also supports MySQL, PostgreSQL, SQLite, SQL Server, DB2 and Oracle. Rails is also an MVC (model, view, controller) framework where all layers are provided by Rails, as opposed to reliance on other, additional frameworks to achieve full MVC support. Invented by David Heinemeier Hanss, Ruby On Rails has been developed as an open-source project, with distributions available through rubyonrails.org.

Ruby on Rails pamatprincipi:

* Neatkārtot sevi (DRY)
* Vienošanās nevis konfigurācija
* Lietotne ar viedokli

### Django ietvars

Django ir Python izstrādes valodas ietvars

Django (/ˈdʒæŋɡoʊ/ JANG-goh)[3] is a free and open source web application framework, written in Python, which follows the model–view–controller architectural pattern.[4][5] It is maintained by the Django Software Foundation (DSF), an independent organization established as a 501(c)(3) non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Some well-known sites that use Django include Pinterest,[6] Instagram,[7] Mozilla,[8] The Washington Times,[9] Disqus,[10] and the Public Broadcasting Service.[11]

Django was born in the fall of 2003, when the web programmers at the Lawrence Journal-World newspaper, Adrian Holovaty and Simon Willison, began using Python to build applications.[12] It was released publicly under a BSD license in July 2005. The framework was named after guitarist Django Reinhardt.[12]

In June 2008, it was announced that a newly formed Django Software Foundation (DSF) would maintain Django in the future.

The core Django MVC framework[4][5] consists of an object-relational mapper which mediates between data models (defined as Python classes) and a relational database ("Model"); a system for processing requests with a web templating system ("View") and a regular-expression-based URL dispatcher ("Controller").

### Java Play ietvars

Java Play ir Java izstrādes valodas ietvars

### Ietvaru salīdzinājums

Tiks apkopotas ietvaru iespējas uut.

Ietvaru funkcionalitātes salīdzinājums

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Laravel 4 | Ruby on Rails | Django | Java |
| Izstrādes šablons | MVC | MVC | MTV | MVC |
| Vienības testu funkcionalitāte | + | + | + | + |
| Pakotņu kontroles sistēma | Composer | Gem | Pip | - |
| Automātiska datubāzes struktūru veidošana | + | + | + | + |
| Datubāzes struktūras migrācija | + | + | + | + |
| MySQL | + | + | + | + |
| PostgreSQL | + | + | + | + |
| SQLite | + | + | + |  |
| H2 | - | - | - | + |

## Datubāzes izvēlne

Šajā nodaļa tiks apskatītas un salīdzinātas potenciāli izmantojamās datubāzēs, lai būtu iespējams izvēlēties piemērotāko datubāzi izstrādes procesam. Darbā tiks aplūkotas relāciju datubāzes, jo izvēlētais ietvars atbalsta relāciju datubāzes.

Lietotnes informācijas apjomiem augot ir nepieciešams uzglabāt informāciju, tas ir nepieciešams, lai iegūto informāciju būtu iespējams atkārtoti izmantot, organizēšanai loģiskos apgabalos un saglabāt vēlākai izmantošanai. Informācijas saglabāšanai ir tik izmantotas dažādas metodes:

* informācijas glabāšana atmiņā
* informācijas glabāšana failos
* informācijas glabāšana datubāzē

Datubāzu vadības sistēma (angļu: Database management system (DBMS)), ir programma (vai programmu kopums), kas nodrošina Datubāzu pārvaldību. DBPS ļauj ievietot, nolasīt, modificēt un dzēst ierakstus, kā arī veic citas ar datu glabāšanu un apstrādi saistītas darbības datu bāzē.

Datubāzu pārvaldības sistēmas mērķi:

* nošķirt datu aprasti no datu apstrādes
* Loģiskā un fiziskā datu neatkarība
* Vienkārša datu administrēšana un kontrole
* Minimāla redundance un minimāla aizņemta atmiņā
* Datu integritāte
* Datu koplietošana
* Datu drošība

Relāciju datubāzes

Mūsdienās (2014.04.06) [biežāk] izmanto datubāzu vadības sistēmas. Šo datubāzu mērķis ir sniegt lietotājam iespēju izstrādā, veidot, iegūt datus, atjaunot datus un veikt šo lietotņu administrāciju. Kā izplatītākas datubāzu vadības sistēmas var minēt:

* MySQL
* MariaDB
* Oracle
* PostgreSQL
* SQLite
* Microsoft SQL Server

Relāciju datubāzu priekšrocības

* Vienkāršā datu struktūra
* SQL vaicājumu valoda – šis vaicājumu valoda ir ļoti tuva dabīgajai angļu valodai
* Drošība
* Datu neatkarība
* Web hosting risinājumi piedāvā plašu relāciju datubāzu piedāvājumu

Relāciju datubāzu trūkumi:

* Veiktspēja – datubāzu veiktspēju samazina liels datu apjoms un datu apvienošanas starp tabulām
* Lēna datu apstrāde
* Datu apstrāde

Informācija tiek glabāta tabulu formātā, kur informācija starp tabulām tiek izteikta ar relācijām.

### Relācijas datubāzu salīdzinājums

Šajā darba daļā tiks apskatītas datubāzu vadības sistēmas – MySQL, PostgreSQL, Oracle, SQLite. Šādas datubāzes tika izvēlētas, jo tās ir vienas no populārākajām, tālāk tika apskatītas to piedāvātās iespējas.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | SQLite | MySQL | MariaDB | PostgreSQL | Oracle |
| Atvērtā koda |  | + |  | + | - |
| JSON datu glabāšana | - |  | + | + | - |
| Self contained | + | - | - | - | - |
| ACID |  |  |  | + | + |
| Vairākas rakstīšanas operācijas | - |  |  |  |  |
| Datubāzes lietotāju kontrole | - | + | + | + | + |
| Ārējā atslēga |  | + | + | + | + |
| Glabātās procedūras |  | + | + | + | + |
| SQL vaicājumu valoda | + | + | + | + | + |
| SQL standarti |  |  |  | + |  |
| Datubāzes modelis | RDBMS | RDBMS | RDBMS | ORDBMS | RDBMS |
| Licences | Publiskais domēns | GPL 2 or proprietary | GPL2 | PostgreSQL License | proprietary |

## Servera tehnoloģiju izvēle

Viens no sarežģītākajiem procesiem ir lietotnes uzstādīšana uz publiskā servera, jo ir nepieciešams sabalansēt dažādus faktorus, piem., izmaksas, nodrošināto funkcionalitāti, noturību, piedāvātās iekārtas veiktspēju un uzturēšanas iespējas.

# praktiskā daļa

Prasību specifikācija ir izstrādāta atbilstoši Latvijas Valsts standartu formālajām prasībām. Lai identificētu sistēmas prasības, tiek izmantoti lietošanas gadījumi. Katram lietošanas gadījumam ir parādīta tipiskā notikumu secība, būtiskākās alternatīvas un kritisko kļūdu notikumu apstrāde.

## Sistēmas prasības

Būs vēl nolūks, darbības sfēra, definīcijas utmlpēs PPS struktūras.

### Produkta funkcijas

Šajā darba sadaļa tiks aprakstīti lietošanas gadījumi, kuri ir paredzēti izstrādātajai lietotnei. Lietošanas gadījumi ir paredzēti 4 lietotāju grupām:

* Administrators
* Autors
* Reģistrēts lietotājs
* Viesis

Links uz lietotni

Lietotāja vārds: papillonAdmin

Parole: papillonAdmin

2.tabula. Lietošanas gadījumu īss apraksts

| **Nosaukums un šifrs** | **Aktieri** | **Mērķis** | **Īss apraksts** | **Prioritāte izpildīšanai** | **Atsauces** |
| --- | --- | --- | --- | --- | --- |
|  | Administrators, autors, reģistrēts lietotājs | Ielogošanās sistēmā | Pēc lietotāja norādītās informācijas tiek | Galvenais |  |
|  | Administrators, autors, reģistrēts lietotājs | Profila informācijas labošana |  |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Aptauju apskatīšana |  |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Aptauju aizpildīšana |  |  |  |
|  | Administrators, autors, reģistrēts lietotājs, viesis | Rakstu apskatīšana | Raksta informācijas apskatīšana |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Ziņas nosūtīšana | Ziņas nosūtīšana |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Ziņu atlasīšana | Ziņu meklēšana |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Ziņu saņemšana/lasīšana | Ziņas informācijas apskatīšana |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Ziņu dzēšana | Lietotāja ziņu dzēšana |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Komentāra pievienošana rakstam | Komentāra pievienošana raktam |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Komentāra pievienošana aptaujai | Komentāra pievienošana aptaujai |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Profila informācijas labošana | Lietotāja profila informācijas labošana |  |  |
|  | Administrators, autors | Raksta pievienošana | Raksta pievienošana |  |  |
|  | Administrators, autors | Raksta labošana | Raksta informācijas labošana | Galvenais |  |
|  | Administrators, autors | Raksta dzēšana | Raksta dzēšana | Galvenais |  |
|  | Administrators, autors | Aptaujas pievienošana | Aptaujas pievienošana | Galvenais |  |
|  | Administrators, autors | Aptaujas labošana | Aptaujas informācijas labošana | Galvenais |  |
|  | Administrators, autors | Aptaujas dzēšana | Aptaujas dzēšana | Galvenais |  |
|  | Administrators | Kategoriju pievienošana | Kategorijas pievienošana | Galvenais |  |
|  | Administrators | Kategoriju labošana | Kategorijas informācijas labošana | Galvenais |  |
|  | Administrators | Kategoriju dzēšana | Kategorijas informācijas dzēšana | Galvenais |  |
|  | Administrators | Raksta publicēšana | Publicēt rakstu | Galvenais |  |
|  | Administrators | Aptaujas publicēšana | Publicēt aptauju | Galvenais |  |
|  | Administrators | Publicējamo rakstu saraksts | Attēlot iesniegto rakstu sarakstu, kurus nepieciešams pārskatīt |  |  |
|  | Administrators | Publicējamo aptauju saraksts | Attēlot iesniegto aptauju sarakstu, kurus nepieciešams pārskatīt |  |  |
|  | Administrators | Lietotāju bloķēšana | Lietotāj pieejas tiesību atņemšana | Galvenais |  |
|  | Administrators | Lietotāju atbloķēšana | Lietotāj pieejas tiesību atjaunošana | Galvenais |  |
|  | Administrators | Lietotāju dzēšana | Dzēst lietotājus no sistēmas | Galvenais |  |
|  | Administrators | Tiesību līmeņa mainīšana | Mainīt lietotāja privilēģiju līmeni | Galvenais |  |
|  | Viesis | Rakstu apskatīšana | Pievienotā raksta informācijas iegūšana un attēlošana |  |  |
|  | Viesis | Reģistrācija | Jauna lietotāja reģistrācija sistēmā, lai šim lietotājam tiku piešķirtas papildus | Galvenais |  |

### Funkcionālās prasības

#### UC-1.1 Ierakstīties sistēmā

|  |  |
| --- | --- |
| **Aktieri:** | Administrators, autors un reģistrēts lietotājs |
| **Mērķis:** | Ierakstīties sistēmā |
| **Īss apraksts:** | Administrators, autors vai reģistrēts lietotājs ievada savus datus, kas tiek salīdzināti ar datiem, kas atrodas datu bāzē, kad tie sakrīt, tad tiek iesākta sesija un administrators var sākt darbu |
| **Tips:** | Galvenais |
| **Atsauces:** |  |

Tipiskā notikumu secība:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators ievada autentifikācijas datus | 2 | Sistēma pārbauda datus, verifikācija |
|  |  | 3 | Tiek parādīta darba vide |

Kļūdu apstrāde:

2: Sistēma konstatē, ka nav aizpildīti visi lauki→ Sistēma paziņo par to lietotājam

#### „Ierakstīties sistēmā” lietošanas gadījums

|  |  |
| --- | --- |
| Lietošanas gadījums: | Ierakstīties sistēmā |
| Aktieris: | Administrators, autors, reģistrētais lietotājs |
| Mērķis: | Pārbaudīt lietotāja norādīt informāciju, ja tā ir pareiza, tad piešķirt papildus tiesības |
| Īss apraksts: | Sniegt iespēju lietotājam ielogoties sistēma un iegūt papildus funkcionalitāti |
| Tips: | Galvenais |
| Atsauces: |  |

Tipiskā notikumu secība:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators, reģistrēts lietotājs vai autors ievada autentifikācijas datus | 2 | Sistēma pārbauda datus, verifikācija |
|  |  | 3 | Lietotāj piemeklēšana |
|  |  | 4 | Tiek parādīta darba vide |

Alternatīvie notikumi:

2.Lietotāja dati neatbilst norādītajiem nosacījumiem -> tek izvadīta kļūda par neatbilstošajiem datiem.

3.Lietotāju neizdodas atrast sistēmā -> tiek piedāvāta iespēja reģistrēties.

#### Lietotāja profila labošana

|  |  |
| --- | --- |
| **Lietošanas gadījums:** | **Lietotāja profila labošana** |
| Aktieris: | Administrators, autors, reģistrētais lietotājs |
| Mērķis: | Labot lietotāja norādīto profila informāciju |
| Īss apraksts: | Nodrošināt lietotāja informācijas atjaunošana |
| Tips: | Sekundārs |
| Atsauces: |  |

Tipiskā notikumu secība:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators, reģistrēts lietotājs vai autors izvēlās profila labošanas iespēju | 2 | Sistēma pēc lietotāj identifikatora piemeklē datus un tos ielādē |
|  |  | 3 | Dati tiek attēloti formā |
| 4 | Tiek veiktas izmaiņas datos |  |  |
| 5 | Tiek nospiesta poga „Saglabāt” | 6 | Datu integritātes pārbaude |
|  |  | 7 | Lietotāja informācijas saglabāšana |
|  |  | 8 | Lietotāja pāradresēšana profila pārskatu |

Kļūdas gadījumi:

2.Pēc lietotāja identifikatora neizdodas atrast lietotāju -> tiek attēlota kļūdas ziņa un pārtraukta lietotāja sesija.

7.Datubāzes kļūda saglabājot informāciju -> tiek atgriezta kļūda, ka informācija netika saglabāta.

Alternatīvie notikumi:

6.Ievadītie dati neatbilst nosacījumiem -> tiek attēlota brīdinājuma ziņa

#### „Lietošanas bloķēšana” lietošanas gadījums

|  |  |
| --- | --- |
| Lietošanas gadījums: | Lietotāju bloķēšana |
| Aktieris: | Administrators |
| Mērķis: | Veikt lietotāju bloķēšana |
| Īss apraksts: | Nodrošināt iespēju administratoram iespēju lietotāja iespēju ielogoties sistēmā |
| Tips: | Galvenais |
| Atsauces: |  |

Tipiskā notikumu secība:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators atver formu ar lietotāju sarakstu | 2 | Lietotāju informācijas iegūšana no datubāzes |
|  |  | 3 | Informācijas attēlošana formā |
| 4 | Administrators izvēlās lietotāju un nospiež pogu bloķēt | 5 | Tiek piemeklēts lietotājs pēc norādīta identifikatora |
|  |  | 6 | Uzstāda bloķējuma identifikatoru |
|  |  | 7 | Saglabā informāciju datubāzē |
|  |  | 8 | Pāradresē uz lietotāju sarakstu |

Kļūdu apstrāde:

5.Netiek atrasts lietotājs pēc norādīta identifikatora - > tiek atgriezta kļūdas ziņa, ierakstu neizdevās atrast.

7.Datubāzes kļūda saglabājot informāciju datubāzē -> tiek atgriezta kļūdas ziņa par to, ka nav izdevies saglabāt informāciju datubāzē.

Alternatīvie notikumu gadījumi:

2.Nav atrasts neviens ieraksts datubāzē -> tiek attēlota brīdinājuma ziņa

#### „Lietotāju atbloķēšana” lietošanas gadījums

|  |  |
| --- | --- |
| Lietošanas gadījums: | Lietotāju bloķēšana |
| Aktieris: | Administrators |
| Mērķis: | Veikt lietotāju bloķēšana |
| Īss apraksts: | Nodrošināt iespēju administratoram iespēju lietotāja iespēju ielogoties sistēmā |
| Tips: | Galvenais |
| Atsauces: |  |

Tipiskā notikumu secība:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators atver formu ar lietotāju sarakstu | 2 | Lietotāju informācijas iegūšana no datubāzes |
|  |  | 3 | Informācijas attēlošana formā |
| 4 | Administrators izvēlās lietotāju un nospiež pogu bloķēt | 5 | Tiek piemeklēts lietotājs pēc norādīta identifikatora |
|  |  | 6 | Uzstāda bloķējuma identifikatoru |
|  |  | 7 | Saglabā informāciju datubāzē |
|  |  | 8 | Pāradresē uz lietotāju sarakstu |

Kļūdu apstrāde:

5.Netiek atrasts lietotājs pēc norādīta identifikatora - > tiek atgriezta kļūdas ziņa, ierakstu neizdevās atrast.

7.Datubāzes kļūda saglabājot informāciju datubāzē -> tiek atgriezta kļūdas ziņa par to, ka nav izdevies saglabāt informāciju datubāzē.

Alternatīvie notikumu gadījumi:

2.Nav atrasts neviens ieraksts datubāzē -> tiek attēlota brīdinājuma ziņa

#### UC- Lietošanas gadījums “Rakstu pievienošana”

|  |  |
| --- | --- |
| **Lietošanas gadījums:** | Rakstu pievienošana |
| **Aktieris:** | Administrators, autors |
| **Mērķis:** | Jaunu rakstu pievienošana |
| **Īss apraksts:** | Administrators vai autors veic jauna raksta pievienošanu |
| **Tips:** | Galvenais |
| **Atsauces:** | Jābūt sekmīgi izpildītiem lietošanas gadījumiem “Ierakstīšanās sistēmā” |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators vai autors aizpilda raksta formu | 2 | Sistēma pārbauda datu integritāti, verifikācija |
|  |  | 3 | Informācijas saglabāšana |
|  |  | 4 | Lietotāja pāradresēšana uz pievienotā ieraksta apskati |

Kļūdu apstrāde

7.Kļūda datubāzes līmenī -> lietotājam tiek atgriezta kļūda par to, ka ierakstu neizdevās saglabāt.

Alternatīvie notikumi:

2.Norādīta informācija neatbilst nosacījumiem -> tiek izvadīta ziņa par nekorekti aizpildītiem laukiem

#### UC- Lietošanas gadījums “Raksta publicēšana”

|  |  |
| --- | --- |
| **Lietošanas gadījums:** | **Rakstu publicēšana** |
| Aktieris: | Administrators |
| Mērķis: | Padarīt pievienoto rakstu publisku |
| Īss apraksts: | Lietotājs raksta ievada datus, kuri tiek saglabāti datubāzē, kā raksts |
| Tips: | Galvenais |
| Atsauces: | Jābūt sekmīgi izpildītiem lietošanas gadījumiem “Lietotāja ielogošanās sistēmā” un “Raksta pievienošana” |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators izvēlās rakstu | 2 | Informācija tiek ielādēta datubāzē |
|  |  | 3 | Informācija tiek attēlota formā |
| 4 | Tiek atzīmēta publicēt iespēja |  |  |
| 5 | Nospiež pogu „Saglabāt” | 6 | Informācija tiek saglabāta datubāzē |
|  |  | 7 | Lietotājs tiek pāradresēts uz publicējamo rakstu sarakstu |

Kļūdu apstrāde

2.Neizdodas atrast ierakstu pēc norādīta identifikatora -> tiek atgriezta kļūda par neeksistējošu ierakstu

7.Kļūda datubāzes līmenī saglabājot informāciju -> lietotājam tiek atgriezta kļūda par to, ka ierakstu neizdevās saglabāt.

#### UC- Lietošanas gadījums “Rakstu labošana”

|  |  |
| --- | --- |
| Lietošanas gadījums: | Rakstu labošana |
| Aktieris: | Autors, administrators |
| Mērķis: | Pievienotā raksta informācijas labošana |
| Īss apraksts: | Uzspiežot “Labot” raksta pogu, tiek atvērta raksta forma, kura saturēs informāciju par rakstu. Šo informāciju būs iespējams labot un saglabāt datubāzē. |
| Tips: | Galvenais |
| Atsauces: | Jābūt sekmīgi izpildītiem lietošanas gadījumiem “Lietotāja ielogošanās sistēmā” |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators vai autors izvēlās labojamo ieraktu | 2 | Tiek iegūta raksta informācija un atgriezta formā |
| 3 | Tiek veikti nepieciešamie labojumi | 4 | Informācijas integritātes un korektuma pārbaude |
|  |  | 5 | Informācijas saglabāšana datubāzē |
|  |  | 6 | Pāradresācija uz laboto raksta pārskatu |

Kļūdu apstrāde

2.Neizdodas atrast ierakstu pēc norādīta identifikatora -> tiek atgriezta kļūda par neeksistējošu ierakstu

7.Kļūda datubāzes līmenī -> lietotājam tiek atgriezta kļūda par to, ka ierakstu neizdevās saglabāt.

Alternatīvie notikumi

4.Ievadītā informācija neatbilst nosacījumiem -> tiek izvadīta brīdinājuma ziņa par nekorekti aizpildītu informāciju.

#### UC- Lietošana gadījums “Rakstu dzēšana”

|  |  |
| --- | --- |
| Lietošanas gadījums: | Rakstu dzēšana |
| Aktieris: | Autors, administrators |
| Mērķis: | Raksta dzēšana no datubāzes |
| Īss apraksts: | Raksta ieraksta dzēšana no datubāzes, ieraksts tiek dzēsts pēc norādīta identifikātora |
| Tips: | Galvenais |
| Atsauces: | Jābūt sekmīgi izpildītiem lietošanas gadījumiem “Lietotāja ielogošanās sistēmā” |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators vai autors izvēlās dzēšamo rakstu | 2 | Tiek piemeklēts ieraksts datubāze |
|  |  | 3 | Tiek veikta ieraksta dzēšana no datubāzes |
|  |  | 4 | Lietotājs tiek pāradresēt uz rakstu katalogu |

Kļūdu gadījumu:

2.Datubāzēt netiek atrasts ierakts pēc norādītā identifikatora -> tiek izvadīta kļūdas ziņa.

3.Datubāzes kļūda dzēšot ierakstu -> tiek izvadīta kļūdas ziņa.

#### UC- Lietošanas gadījums “Komentāru pievienošana rakstam”

|  |  |
| --- | --- |
| Lietošanas gadījums: | Komentāru pievienošana rakstam |
| Aktieris: | Autors, administrators, reģistrētais lietotājs |
| Mērķis: | Pievienot lietotāj komentāru rakstam |
| Īss apraksts: | Lietotājs komentāru formā norāda savu komentāru, kurš tiek saglabāts datubāzē ar atsauci uz rakstu. |
| Tips: |  |
| Atsauces: | Jābūt sekmīgi izpildītiem lietošanas gadījumiem “Lietotāja ielogošanās sistēmā” un “Raksta pievienošana” |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators, autors vai reģistrēts lietotājs izvēlās rakstu | 2 | Tiek piemeklēts ieraksts datubāze |
|  |  | 3 | Informācijas un komentāru formas attēlošana |
| 4 | Komentāru formas aizpildīšana | 5 |  |
| 5 | „Pievienot” pogas nospiešana | 6 | Komentāra informācijas integritātes pārbaude |
|  |  | 7 | Komentāra saglabāšana |
|  |  | 8 | Pāradresācija uz rakstu |

Kļūdu apstrāde:

2.Neizdodas atrast ierakstu pēc norādīta identifikatora -> tiek atgriezta kļūda par neeksistējošu ierakstu

7.Kļūda datubāzes līmenī -> lietotājam tiek atgriezta kļūda par to, ka ierakstu neizdevās saglabāt.

Alternatīvie notikumi:

6.Komentāra dati neatbilst nosacījumiem -> tiek atgriezta brīdinājuma ziņa par neatbilstošiem datiem

#### UC- Lietošanas gadījums “Komentāru pievienošana aptaujai”

|  |  |
| --- | --- |
| Lietošanas gadījums: | Komentāru pievienošana aptaujai |
| Aktieris: | Autors, administrators, reģistrētais lietotājs |
| Mērķis: | Komentāra pievienošana aptaujai |
| Īss apraksts: | Reģistrēts lietotājs, autors vai administrators veiks komentāru pievienošanu aptaujai. |
| Tips: | Galvenais |
| Atsauces: | Jābūt sekmīgi izpildītiem lietošanas gadījumiem “Lietotāja ielogošanās sistēmā” un “Aptaujas pievienošana” |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators, autors vai reģistrēts lietotājs izvēlās aptauju | 2 | Tiek piemeklēts ieraksts datubāze |
|  |  | 3 | Informācijas un komentāru formas attēlošana |
| 4 | Komentāru formas aizpildīšana |  |  |
| 5 | „Pievienot” pogas nospiešana | 6 | Komentāra informācijas integritātes pārbaude |
|  |  | 7 | Komentāra saglabāšana |
|  |  | 8 | Pāradresācija uz aptauju katalogu |

Kļūdu apstrāde

2.Neizdodas atrast ierakstu pēc norādīta identifikatora -> tiek atgriezta kļūda par neeksistējošu ierakstu

7.Kļūda datubāzes līmenī -> lietotājam tiek atgriezta kļūda par to, ka ierakstu neizdevās saglabāt.

Alternatīvie notikumi:

3.Rakstam nav atrasts neviens komentārs -> tiek attēlota ziņa, ka pagaidām šai aptaujai nav pievienots neviens komentārs.

6.Komentāra ievadītie dati neatbilst nosacījumiem -> tiek attēlota ziņa par nepareizo datu formātu.

#### UC- Lietošanas gadījums “Aptaujas pievienošana”

|  |  |
| --- | --- |
| Lietošanas gadījums: | Aptauju pievienošana |
| Aktieris: | Autors, administrators |
| Mērķis: | Aptaujas pievienošana |
| Īss apraksts: | Administrators vai autors aizpilda aptaujas formu, kuras satur informāciju par aptauju. |
| Tips: | Galvenais |
| Atsauces: | Jābūt sekmīgi izpildītiem lietošanas gadījumiem “Lietotāja ielogošanās sistēmā” |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators vai autors aizpilda aptaujas formu |  |  |
| 2 | „Pievienot” pogas nospiešana |  |  |
|  |  | 2 | Sistēma pārbauda datu integritāti, verifikācija |
|  |  | 3 | Informācijas saglabāšana |
|  |  | 4 | Lietotāja pāradresēšana uz pievienotā aptauju katalogu |

Kļūdu apstrāde:

2.Neizdodas atrast ierakstu pēc norādīta identifikatora -> tiek atgriezta kļūda par neeksistējošu ierakstu

7.Kļūda datubāzes līmenī -> lietotājam tiek atgriezta kļūda par to, ka ierakstu neizdevās saglabāt.

Alternatīvie notikumi:

2.Ievadīta informācija neatbilst nosacījumiem -> tiek izvadīta brīdinājuma ziņa par nekorekti aizpildītiem datiem.

#### Aptaujas nosaukuma labošana

|  |  |
| --- | --- |
| Lietošanas gadījums: | Aptauju nosaukumu labošana |
| Aktieris: | Autors, administrators |
| Mērķis: | Aptaujas nosaukuma labošana |
| Īss apraksts: | Administrators vai lietotājs izvēlās aptauju, kuru nepieciešams labot un informācija tiek ielasīta formā, kuru var izmantot informācijas labošanai. |
| Tips: | Galvenais |
| Atsauces: | Jābūt sekmīgi izpildītiem lietošanas gadījumiem “Lietotāja ielogošanās sistēmā” un “Aptauju pievienošana” |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Labojamā ieraksta izvēle | 2 | Informācijas iegūšana no datubāzes |
|  |  | 3 | Informācijas attēlošana formā |
| 4 | Informācijas labošana |  |  |
| 5 | „Pievienot” pogas nospiešana |  |  |
|  |  | 6 | Sistēma pārbauda datu integritāti, verifikācija |
|  |  | 7 | Informācijas saglabāšana |
|  |  | 8 | Lietotāja pāradresēšana uz pievienotā aptaujas rakstu |

Kļūdu apstrāde

2.Neizdodas atrast ierakstu pēc norādīta identifikatora -> tiek atgriezta kļūda par neeksistējošu ierakstu

7.Kļūda datubāzes līmenī -> lietotājam tiek atgriezta kļūda par to, ka ierakstu neizdevās saglabāt.

Alternatīvo notikumu apstrāde

6.Dati neatbilst nosacījumiem -> tiek atgriezts brīdinājums par nekorekti aizpildītiem datiem

#### Aptaujas dzēšana

|  |  |
| --- | --- |
| Lietošanas gadījums: | Aptaujas dzēšana |
| Aktieris: | Autors, administrators |
| Mērķis: | Dzēst aptauju |
| Īss apraksts: | Autors vai administrators aizpilda aprakstu, raksta informāciju |
| Tips: | Galvenais |
| Atsauces: | Jābūt sekmīgi izpildītiem lietošanas gadījumiem “Lietotāja ielogošanās sistēmā” un “Aptauju pievienošana” |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators vai autors izvēlās dzēšamo aptauju | 2 | Tiek piemeklēts ieraksts datubāze |
|  |  | 3 | Tiek veikta ieraksta dzēšana no datubāzes |
|  |  | 4 | Lietotāju pāradresācija uz aptauju sarakstu |

Kļūdu apstrāde

2. Tiek piemeklēts ierakstu datubāzē -> pēc norādīta identifikatora neizdodas atrast ierakstu datubāzē, tiek atgriezta kļūda par to, ka ieraksts neeksistē.

3.Kļūda datubāzes līmenī dzēšot ierakstu -> Tiek atgriezta kļūda par to, ka neizdevās saglabāt informāciju datubāzē.

Lietošanas gadījuma scenārija realizācijas grafiskās vides uzmetums:

## Programmatūras projektējuma apraksts

Pēc PPA struktūras.

### Datu bāzes struktūra

Te būs redzama relāciju datu bāzes shēma, lai saprastu kā ir uzkonstruēta DB.

Datubāzes dekompozīcija

Tabula articles\_attachment

|  |  |  |
| --- | --- | --- |
| Lauka nosaukums | Tips | Apraksts |
| Id | integer NOT NULL PRIMARY KEY AUTOINCREMENT |  |
| entity | varchar(100) NOT NULL |  |
| title, | varchar(255) NOT NULL |  |
| created\_at | datetime NOT NULL |  |
| updated\_at | datetime NOT NULL |  |

Tabula articles\_articles

|  |  |  |
| --- | --- | --- |
| Lauka nosaukums | Tips | Apraksts |
| Id | integer NOT NULL PRIMARY KEY AUTOINCREMENT |  |
| user\_id | integer NOT NULL, |  |
| title, | varchar(255) NOT NULL |  |
| description | varchar(255) NOT NULL, |  |
| embeded | text NOT NULL, |  |
| article | text NOT NULL, |  |
| created\_at | datetime NOT NULL |  |
| updated\_at | datetime NOT NULL |  |

Tabula comments\_pollcomments

|  |  |  |
| --- | --- | --- |
| Lauka nosaukums | Tips | Apraksts |
| id | integer NOT NULL PRIMARY KEY AUTOINCREMENT |  |
| user\_id | integer NOT NULL |  |
| poll\_id | integer NOT NULL |  |
| comment | text NOT NULL |  |
| created\_at | datetime NOT NULL |  |
| updated\_at | datetime NOT NULL |  |

Tabula comments\_articlecomments

|  |  |  |
| --- | --- | --- |
| Lauka nosaukums | Tips | Apraksts |
| Id | integer NOT NULL PRIMARY KEY AUTOINCREMENT |  |
| user\_id | integer NOT NULL |  |
| article\_id | integer NOT NULL |  |
| comment | text NOT NULL |  |
| created\_at | datetime NOT NULL |  |
| updated\_at | datetime NOT NULL |  |

# Sistēmas ieviešana un testēšana

Šajā nodaļā tiks aprakstītas tehnoloģijas un metodes, kuras tika izmantotas projekta izstrādei un uzturēšanai. Lietotnes ieviešanai tika izmantoti sekojoši rīki:

* Git – bezmaksas izplatīta versionēšanas sistēma, kura tika izmantota kopā ar GitHub vietni koda glabāšanai
* TravisCI – lietotne tika izmantota, lai veiktu automātiskus integrācijas testus
* Digital Ocean – interneta pakalpojumu sniedzējs, kurš nodrošina lētu, ātru un vienkāršu vietnes izvietošanu

## Git versiju kontrole sistēma

Kā viena no svarīgākajām tehnoloģijām projektu izstrādē ir versiju kontroles sistēmas, darba izstrādei tiek izmantota GIT versiju kontroles sistēma, šīs sistēma ļauj:

* uzturēt koda vēsturi
* decentralizēti veikt projekta izstrādi
* izveidot projekta snapshot <<
* Git priekšrocība:
* zarošanas un apvienošana – Git darba plūsma paredz, ka tiek veidoti zari un tiem nav jābūt savā starpā atkarīgiem
* izplatīts – informācija no repozitorija tiek kopēta, un tā rezultātā tiek iegūts lokāla repozitorija kopija

datu integritāte – šis funkcionalitātes nodrošināšanai Git izveidot kontrolsummu katram failam, kurš tiek pievienots. Datu integritāte arī nodrošina to, ka nav iespējams mainīt repozitorija saturu nemainot izmaiņu identifikatora vērtību. Centralizētās versionēšanas sistēmas piem., kā SVN nenodrošina šādu datu integritāti.

Bezmaksas – Git projekts izmanto GNU 2.0 licenci

konfigurācija

GIT versiju kontroles sistēma bez augstāk minētajiem apgalvojumiem ļauj automatizēt izstrādes procesu. GIT versiju kontroles sistēma ļauj pievienot skriptus, kuri tiek izpildīti dažādos versiju kontroles sistēmas darbības punktos, tā piemēram, pirms izmaiņu ievietošanas repozitorijā, pēc izmaiņu pievienošanas repozitorijā.

Šī iespēja tiek izmantota, lai veiktu testu izpildi pirms jaunā koda pievienošanas repozitorijam, izmaiņas ir iespējams pievienot tikai tajā gadījuma, ja visi testi izpildās veiksmīgi.

Git versiju kontroles sistēma tiek izvēlēta projekta izstrādes, jo tā ļauj integrēt dažādas ārējās sistēmas.

Github tikai izmantots, jo tas nodrošina GIT versionēšans sistēmas pakalpojumus bez maksas (par šiem bezmaksas nosacījumiem). Github.com piedāvātie pakalpojumi tika izmantoti, jo šis serviss tiek plaši pielietots trešās puses lietotnēs, piem., TravisCI šis serviss tiek izmantots, lai norādītu repozitoriju, kuru nepieciešams testēt.

## Vienību testi

Katrā projektā kā viens no svarīgākajiem posmiem ir lietotnes testēšanas proces, jo tas ļauj pārliecināties par korektu lietotnes darbību, tas ļauj izstrādāt lietotni ar augstu kauzalitāti, kā arī samazināt laiku, kurš ir nepieciešams testu veikšanai. Kā vienkāršākā un ātrākā testēšanas metodes, kuru ir iespējams izmantot izstrādātajiem ir vienību testi, jo tos var pievienot izstrādātājs lietotnes izstrādes procesa laikā. Šo testu izpildīšanas ir vienkārša, jo tā parasti ir automatizēta.

Django ietvars ietver sevi testēšanas rīkus, tos ir iespējams izmantot lietotnes funkcionalitātes automātiskai testēšanai, kurš ļauj simulēt HTTP pieprasījums, datu pievienošanu, pārbaudīt lietotnes izvadi un pārliecināties, ka kods izpilda norādītās darbības.

Lai būtu iespējams veikt testus ir nepieciešams vispirms iekļaut testu pamatklāsti, kuras mantošanas ļauj veidot testu gadījumus.

from django.test import TestCase

Veidojot testus priekš Django ietvara failus, kuri satur testu gadījums ir nepieciešams nosaukt tādā veidā, lai tie sāktos ar vārdu test\*.py

Testu palaišana

Testu palaišanai tiek izmantota ./manage.py test komanda, kuru izpilda projekta direktorijā. Izpildot šo komandu tiek izveidoti testu gadījumi visām projekta lietotnēm.

Django testu ietvars nekādā veidā neietekmēs izstrādes vai produkcijas datubāzi, jo testu veikšanai tiek izmantota atsevišķa datubāze, kur tiek izveidota testu sākumā un neatkarīgi no rezultātiem iznīcināta testu beigās.

## TravisCI integrācijas testi

Lai nodrošinātu papildus testus un to automatizāciju projekta izstrādei tiek izmantots TravisCI, tā nepārtrauktās integrācijas sistēma, kuru tiek integrēta ar github.com vietnē izvietoto repozitoriju, tas tiek izdarīts izmantojot hooks, kuri veic projekta veidošanu un testu palaišanu pēc izmaiņu ienešana github.com repozitorijā.

Šī integrēšanas sistēma tika izmantota, lai automatizētu šādus uzdevumus:

testu veikšana

dažādu python un django versiju integrācijas testi

Šo testu veikšana ļauj priekšlaicīgi identificēt problēmas, ar kurām nāksies saskarties aplikācijas dzīves ciklā. Šīs sistēmas izmantošana ļauj priekšlaicīgi identificēt iespējamas problēmas migrējot projektu uz jaunākām Django ietvara versijām.

TravisCI projektu veidošanai un testēšanai norādījumiem izmanto YAML failu, kurš satur nepieciešamo konfigurāciju.

## Lietotnes ieviešana

Šajā nodaļā tiks aprakstīta lietotnes ieviešanas stratēģija.

Server izvēle

Lietotnes ieviešanai tiek izmantoti digitalocean.com piedāvātie pakalpojumi, jo tas nodrošina:

* lētus pakalpojumus
* virtuālo mašīnu pārvietošana
* virtuālās mašīnas ar gatavām instalācijām
* plašus komūnas atbalstu un pamācības

Django instalācijas mezgla konfigurācija

* Ubuntu 14.04
* Python 2.7.6
* Django 1.7.1
* PostgreSQL
* Nginx 1.4.6
* Gunicorn 17.5

Nobeigums

Kopumā secinājums par pētāmo jomu (it kā atgādinājums par ko gāja runa)

1-2 teikumi galvenais darba sasniegums

Ievadā uzstādīti mērķi ir sasniegti (izvērsta teikumā)

Galvenie pētījuma secinājumi (strukturētā formā)

Ja hipotēze nav pareiza, kāda varētu būt atbilde uz sākotnēji uzstādīto jautājumu?

Rezumēt visas grūtības vai problēmas, ar kurām saskārās veicot pētījumu.

**Mērķa grupas:**

* Arhitektūras un mākslas studenti, profesionāļi.
* Pilsētas iedzīvotāji
* Tūristi
* Investori

**Institūcijas, kam tas varētu interesēt:**

* Rīgas Dome: Rīgas tēla popularizēšana, izpratnes veidošana par pilsētas telpu.
* Pilsētas arhitekta birojs : Rīgas arhitektūras kvalitāšu popularizēšana
* Rīgas pašvaldības aģentūras „Rīgas pilsētas arhitekta birojs” (turpmāk tekstā – Rīgas pilsētas arhitekta birojs) mērķi ir sekmēt līdzsvarotu un ilgstpējīgu Rīgas pilsētvides attīstību, uzlabojot pašvaldības darbu arhitektūras kvalitātes pārraudzībā – pilnveidojot pārvaldes instrumentu kopumu un uzturot pastāvīgu, atklātu, savlaicīgu, plašu, profesionālu diskusiju par sabiedrībai nozīmīgām iecerēm un projektiem, kā arī iepazīstināt citu valstu un pilsētu sabiedrību ar labākajiem Latvijas arhitektūras prakses piemēriem.
* Arhitektūras muzejs: Vēsturiskās informācijas pieejamība.
* Arhitektūras informācijas centrs: Diskusijas rosināšanai par ar arhitektūru saistītiem jautājumiem pilsētā
* Anothertravelguide: Tūrisma informācija
* LiveRiga: Tūrisma informācija
* u.c..

Atrisinātas un neatrisinātas problēmas 🡪nākotnes perspektīvas

Ko nākošreiz šāda veida pētījumam būtu jādara citādi?

Sastādīt sarakstu ar citām lietām, ko esat iemācījušies.

Pielikumi

**1.pielikums**

**Pirmā pielikuma virsraksts**

Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts.

**2.pielikums**

**Otrā pielikuma virsraksts**

Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts. Pielikuma teksts.

Literatūra

[Buka 1987] Buka O., Volrāts U. Pilsētbūvniecība. – Rīga: Zvaigzne. – 1987. – 252 lpp.

[Mencis 1990] Vēsture: Māc.grām./ J. Mencis, V. Kārkliņa, E. Krastiņa u.c. – Rīga: Zvaigzne, 1990. – 158 lpp.

[Holcmanis 1999] Holcmanis A. Divas pasāžas// Latvijas arhitektūra. – 1999. – Nr.22. – 8-12.lpp.

[RTU] Rīgas Tehniskā universitāte/ Internets. – http://www.rtu.lv/, skatīts 01.09.2009

[Autos gads] Autori. Grāmatas nosaukums – Izdevējs – gads. Lapas nr.

Python ietvaru salīdzinājums - http://www.skilledup.com/articles/introduction-python-web-frameworks/, skatīts

Ruby ietvaru salīdzinājums - <https://www.ruby-toolbox.com/categories/web_app_frameworks>, skatīts

Top Ten Best PHP Frameworks For 2014 - <http://www.webhostingreviewboards.com/development/top-ten-best-php-frameworks-for-2014/>, skatīts

Best PHP Frameworks for 2014 - <http://www.sitepoint.com/best-php-frameworks-2014/>, skatīts

The 2014 Decision Maker’s Guide to Java Web Frameworks - <http://zeroturnaround.com/rebellabs/the-2014-decision-makers-guide-to-java-web-frameworks/2/>, skatīts

Lietotāju interfeisu bibliotēka – http://semantic-ui.com/, skatīts

<http://devrates.com/>, skatīts

1. <https://github.com/trending?l=python&since=monthly> – Viens no populārākajiem Python projektiem [↑](#footnote-ref-2)