RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

Lietišķo datorsistēmu institūts

Lietišķo datorzinātņu katedra

**Ivars Naglis**

**RĪGAS ARHITEKTŪRAS CELVEŽA TĪMEKĻA LIETOTNESIZSTRĀDE**

**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 6](#_Toc405493556)

[1. zinātniskā daļa 8](#_Toc405493557)

[1.1. Problēmas apraksts 8](#_Toc405493558)

[1.2. Tehnoloģiju izvēle 8](#_Toc405493559)

[1.2.1. Operētājsistēmas izvēle 8](#_Toc405493560)

[1.2.2. Servera izvēle 10](#_Toc405493561)

[1.2.3. Programmēšanas valodas izvēle 13](#_Toc405493562)

[1.2.4. Ietvara izvēle 14](#_Toc405493563)

[2. praktiskā daļa 17](#_Toc405493564)

[2.1. Sistēmas prasības 17](#_Toc405493565)

[2.1.1. Produkta funkcijas 17](#_Toc405493566)

[2.1.2. Funkcionālās prasības 19](#_Toc405493567)

[2.2. Programmatūras projektējuma apraksts 25](#_Toc405493568)

[2.2.1. Datu bāzes struktūra 25](#_Toc405493569)

[Nobeigums 27](#_Toc405493570)

[Pielikumi 28](#_Toc405493571)

[Literatūra 31](#_Toc405493572)

Ievads

Arhitektūra ir nozīmīgākā pilsētas tēla veidotāja. Dažādu laiku celtnes, to savstarpējā mijiedarbība lielā mērā nosaka konkrētās pilsētas raksturu, tādējādi spēlējot nozīmīgu lomu tās iedzīvotāju identitātes veidošanā.

Kvalitatīva publiskā telpa, augstvērtīga mūsdienu arhitektūra un cieņpilna attieksme pret arhitektūras vēsturiskajām vērtībām ir pamats sabiedrības izpratnes par kvalitatīvas pilsētas telpas veidošanu radīšanai.

Atsevišķas celtnes un pilsētas telpa kopumā ir arī viens no svarīgākajiem pilsētas viesu intereses objektiem. Tādēļ liela nozīme ir ērti pieejamai informācijaisvešvalodās.

Pašreiz informācija par Rīgas (Latvijas?) arhitektūras objektiem lielākoties ir pieejama drukātā formātā. Interneta vidē atrodama informācija par pašām slavenākajām un vēsturiski nozīmīgākajām ēkām – tā izvietota objektu mājaslapās vai tūrisma ceļvežos. Tādēļ kopumā arī pieejamā informācija vērtējama kā sadrumstalota un subjektīva – pārmērīgu uzsvaru liekot uz pāris celtnēm, tiek radīts šaurs skatījums uz Rīgas arhitektoniskajām kvalitātēm.

Lai paplašinātu skatījumu uz pilsētas telpiskajām vērtībām, būtu nepieciešams nodrošināt informācijas pieejamību gan par vēsturisko, gan mūsdienu arhitektūru Rīgā, aptverot plašāku arhitektūras spektru kā tūrisma ceļvežos piedāvātais. Turklāt informācijai jābūt plaši pieejamai un ērti atjaunojamai. Drukātā formātā šo kvalitāšu nodrošināšana ir laikietilpīga un dārga. Mūsdienu informācijas tehnoloģiju attīstības straujais temps un tās piedāvātās iespējas ļauj secināt, ka digitālas datu bāzes veidošana būtu ievērojami veiksmīgāks risinājums šāda veida informācijas atspoguļošanai.

Apkopojot informāciju par nozīmīgākajiem vēsturiskajiem un laikmetīgās arhitektūras objektiem mūsdienīgā formātā, būtu iespējams izveidot visaptverošu digitālu arhitektūras objektu datu bāzi, kas sniegtu pārskatu par pilsētas arhitektūru plašākā kontekstā.

**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**:

* Apkopot informāciju par ģeogrāfisko datu uzkrāšanas formātiem un to apstrādes tehnoloģijām.
* 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

Tātad lai realizētu Rīgas arhitektūras ceļveža tīmekļa lietotni ir nepieciešams izvērtēt kādi tehnoloģiskie risinājumi būs nepieciešami, lai to realizētu pēc iespējas lētāk. Kā arī ir jāveic pašas lietotnes izstrāde, lai varētu apskatīt lielisko un neatkārtojamo Rīgas arhitektūru vispasaules tīmeklī.

## Tehnoloģiju izvēle

### Operētājsistēmas izvēle

<http://www.theonestopwebsiteshop.com/web-hosting/type-of-os.htm>

<http://webdesign.about.com/od/webhosting/a/windows_v_linux.htm>

Operētājsistēma(OS) ir programma, kas ļauj mijiedarboties ar datoru - ar programmatūru un aparatūru datorā. Kā tas tiek darīts?

Pamatā ir divi veidi:

* Ar komandrindas operētājsistēmu (piemēram, DOS), rakstot tekstu komandu un dators atbild saskaņā ar šo komandu.
* Ar grafisko lietotāja saskarnes operētājsistēmu (piemēram, Windows), tā ļauj mijiedarboties ar datoru, izmantojot grafisko saskarni ar attēliem un pogām, izmantojot datorpeli un klaviatūru.

Ar Unix operētājsistēmu ir iespēja, izmantot vai nu komandu rindu (vairāk kontroles un elastība), vai grafisko saskarni, kuru var pielāgot vajadzībām.

Unix un Windows: divas galvenās operētājsistēmu klases

Tāsvēsturiski konkurē savā starpā. UNIX tiek izmantota vairāk nekā četrus gadu desmitus. Kopš 1980. gadaUnix galvenais konkurents Windows ir ieguvis popularitāti sakarā ar pieaugošo pieprasījumu pēc personālajiem datoriem ar Intel procesoriem. Windows, tajā laikā, bija vienīgā galvenāOS, kas paredzēta šāda veida procesoriem. Pēdējos gados, jaunā Unixversija –Linux, kas izstrādāta personālajiem datoriem. To var iegūt par brīvu, un tāpēc ienesīga izvēle privātpersonām un uzņēmumiem.

Kā arīUnix 1999. gadā sāka iegūt pozīcijas serveru OS jomā, Linuxieguva 2[xx]. Vietu aiz Windows NT. 2001.gadāLinux tirgus daļa, bija 25%, citi Unixprodukti bija 12%. Personālo datoru lietotāju vidū, Microsoft pašlaik dominē operētājsistēmu tirgū ar vairāk nekā 90% tirgus daļas.

Sakarā ar Microsoft agresīvo mārketinga praksi, ir miljoniem lietotāju, kuriem nav ne jausmas, katikko iegādātais dators nāk kopā ar Windows OS. Daudziem nav zināms, ka pastāv arī citas operētājsistēmas, kas nav Windows.

Unix priekšrocības[x]:

* Unix ir elastīgāks un to var uzstādīt uz dažāda veida iekārtām, tostarp meinfreima datoriem, superdatoriem un personālajiem datoriem
* Unix ir daudz stabilāka un neplīst tik bieži kā Windows, tāpēc tai nav nepieciešama bieža administrēšana un uzturēšana
* Unixoperētājsistēmās ir iebūvētas vairāk drošības un datņuatļaujas funkcijas nekā Windows
* Unix piemīt daudz lielāka datu apstrādes jauda nekā Windows
* Unix ir līderis tīmekļa risinājumos. Aptuveni 80% no interneta atsaucas uz Unix operētājsistēmām, kur pārsvarā darbojas Apache tīmekļa serveris, pasaulē visplašāk izmantotais tīmekļa serveris
* Programmatūras atjaunināšanas gadījumos Microsoft bieži pieprasa lietotājam iegādāties jaunu vai papildus aparatūru,lai uzstādītu jauno programmatūru. Unixgadījumā tā nav
* Unix arī savā veidā uztur atvērtā koda entuziastus, kas izstrādā arvien jaunus rīkus, kas padara datora lietošanu vienkāršāku un ar mazākām izmaksām

<http://linux.about.com/cs/linux101/a/unix_win.htm>

Piekļūšana tīmekļa serverim

Windows un Linux piedāvā FTP piekļuvi datnēm, bet tikai Linux piedāvā Telnet vai SSH piekļuvi, kas ir daudz drošāks veids kā pārsūtīt datnes, jo savienojums tiek šifrēts. (To ir iespējams arī realizēt uz Windows serveriem, bet ļoti maz hostinga kompāniju to piedāvā.) FTP ir veids, kā pārsūtīt datnes no cietā diska uz tīmekļa serveri. Telnet un SSH ir veids, lai atvērtu logu tieši uz tīmekļa servera un manipulēt datus tieši tur, parasti izmantojot Unix komandrindas komandas[xx].

Servera puses scenāriji

Daudzasmitināšanas sistēmas piedāvā PHP, ASP un ColdFusion. PHP ir biežāk atrodams uz Linux sistēmām, bet ASP ir biežāk atrodams uz Windows. ColdFusion tiek lietots uz abu veidu serveriem.

Datubāzes

Ja tiek gatavota dinamiska satura vietne, tad ir nepieciešama datubāze. Divas populārākās datu bāzes vadības sistēmas ir MySQL un Access. MySQL darbojas gan Linux un Windows, bet biežāk atrodams uz Linux serveriem. Access ir pieejama tikai Windows. Protams, ka ir arī daudz vairāk datu bāzes vadības sistēmas, kuras var lietot šajos serveros.

Drošība

Drošība ir ļoti svarīga tīmekļa vietnēm. Daudzi cilvēki apgalvo, ka Windows serveri ir vairāk ievainojami nekā Linux serveri. Bet realitāte ir tāda, ka viņiem abiem ir drošības problēmas. Ja ir labs administrators, tad serveris būs drošs - neatkarīgi no OS.

<http://webdesign.about.com/od/webhosting/a/windows_v_linux.htm>

Apskatot vēsturi un salīdzinot bezmaksas un maksas risinājumu var secināt, ka īpašas atšķirības nav un ņemot vērā, ka projektam ir nepieciešami bezmaksas risinājumi tiks lietota Linux operētājsistēma.

### Servera izvēle

Te tiks salīdzināti dažādi web serveri, un beigu beigās tiks izvēlēts apachehttpserver. Te būs dažas tabulas ar salīdzinājumiem.

<http://webdesign.about.com/od/servers/a/aa092099.htm>

<http://www.webhostingchat.com/2012/01/apache-vs-internet-information-server-iis/>

<http://www.theonestopwebsiteshop.com/web-hosting/type-of-os.htm>

<http://www.quackit.com/web_servers/tutorial/web_servers_examples.cfm>

<http://rebizz.wordpress.com/2012/01/18/my-take-on-apache-vs-iis-webhost/>

<http://www.techrepublic.com/article/the-web-server-software-choice-iplanet-vs-apache-vs-iis/1034514>

Apache HTTP Server

Tas ir pasaulē populārākais tīmekļa serveris. Tas ir visplašāk izmantotais "atvērtā koda" tīmekļa serveris bezmaksas lietošanai, tam ir pieejami daudz dažādi moduļi, kas dod papildus funkcionalitāti.To var lejupielādēt kā avota kodu, un kompilēt to uz savadatira, vai arī lejupielādēt bināro versiju, ko var lietot daudzās operētājsistēmās (piemēram, Windows, Solaris, Linux, OS / 2, FreeBSD, u.c.). Šim serverim ir pieejami daudz dažādipapildinājumi. Vienīgais, ja gadās nepieciešamība pēc palīdzības, tad var gadīties, ka uzreiz nebūs iespējams sazināties ar kādu pārstāvi no Apache kā tas ir komerciāliem produktiem. Tomēr pastāv maksā-par-atbalstu iespējas kā arī bezmaksas palīdzība forumos, kuros ir liels lietotāju skaits un visu var uzzināt, ja prot atrast meklēto.

Internet InformationServices (IIS)

Internet InformationServices (IIS) ir Microsoft produkts tīmekļa serveru arēnā. Ja tiek lietotsserveris ar Windows Server sistēmu, tas varētu būt labākais risinājums, lai īstenotu tīmekļa servera funkcijas. Tam ir atbilstoša saskarne kā jau tas ir ierasts Windows Server OS, un ir nodrošināts ar atbalstu no Microsoft. Lielākais trūkums šajā tīmekļa serveris ir tas, ka Windows Server ir ļoti dārga programmatūra. Tas nav domāts mazajiem uzņēmumiem, lai lietototutīmekļa pakalpojumus.

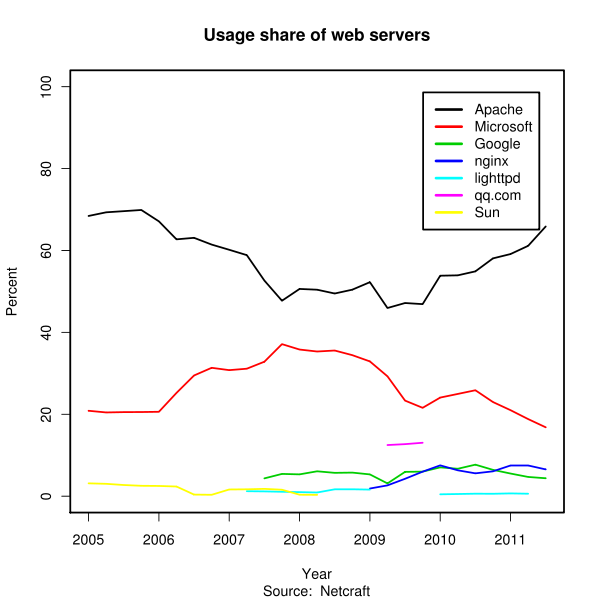
Sun Java Web Server

Trešais būtiskais tīmekļa serveris grupa ir Sun Java Web Server, šobrīd pazīstams kā OracleiPlanet. Šis ir visbiežāk lietotais serveris korporācijām, kas izmanto Unixtīmekļa serveru iekārtas. Sun Java Web Server piedāvā funkcijas no abiem iepriekš aprakstītajiem serveriem, tas ir atbilstošs tīmekļa serveris ar spēcīgu atbalstu, no labi zināma uzņēmuma. Tam arī ir daudz atbalsta ar papildus komponentiem un API, lai serverim būtu vairāk iespēju. Tas ir labs serveris, ja nepieciešams labs atbalsts un elastība uz UNIX platformas.

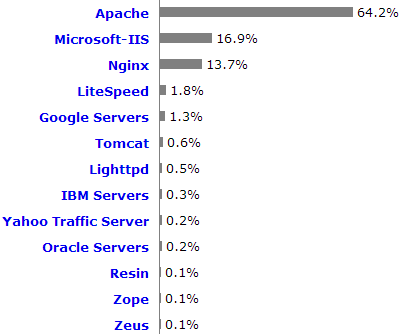
<http://webdesign.about.com/od/servers/a/aa092099.htm>

1.tabula. Servera izvēle, izvērtējot operētājsistēmu atbalstu

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Serveris | Windows | Linux | OS X | BSD | Solaris | eComStation | OpenVMS | AIX | IBM | z/OS | Maksas risinājums |
| Apache HTTP Server | Jā | Jā | Jā | Jā | Jā | Jā | Jā | Jā | Jā | Jā | Nē |
| Internet Information  Services | Jā | Nē | Nē | Nē | Nē | Nē | Nē | Nē | Nē | Nē | Jā |
| OracleiPlanet Web Server | Jā | Jā | Nē | Nē | Jā | Nē | Nē | Jā | Nē | Nē | Jā |



<http://en.wikipedia.org/wiki/File:Usage_share_of_web_servers_(Source_Netcraft).svg>



1.att. Serveru lietojumu statistika (pārņemts no [xx])

<http://w3techs.com/technologies/overview/web_server/all>

Attēlā nr. un nr. var redzēt, ka Apache tīmekļa serveris jau daudzus gadus ir līderis un tiek lietots tīmekļa risinājumu lietošanā. Kā arī ir redzams, ka bijis laika posms, kad savas pozīcijas centies nostiprināt Microsoft IIS tīmekļa serveris, bet tā kā tas ir maksas risinājums, tad ar laiku tas zaudējis savas pozīcijas.Apskatot trīs populārākos tīmekļa serverus ir skaidri redzams, ka šim projektam, kurā galvenais ir bezmaksas risinājumi, tiek izvēlēts Apache tīmekļa serveris.

### Programmēšanas valodas izvēle

Te tiks apskatītas dažas programmēšanas valodas, lai tiktu izvēlēta PHP valoda šim projektam. Te būs tabulas, lai salīdzinātu dažas valodas.

2.tabula. Programmēšanas valodu lietojums pasaulē biežāk apmeklētajām vietnēm [xx]

|  |  |  |
| --- | --- | --- |
| Tīmekļa vietne | Apmeklējumu skaits | Programmēšanas valoda |
| Facebook.com | 870,000,000 | PHP |
| Youtube.com | 790,000,000 | Python |
| Yahoo.com | 590,000,000 | PHP |
| Live.com | 540,000,000 | ASP.NET |
| Wikipedia.org | 460,000,000 | PHP |
| msn.com | 450,000,000 | ASP.NET |
| Blogspot.com | 370,000,000 | Python |
| Baidu.com | 310,000,000 | PHP |
| Microsoft.com | 280,000,000 | ASP.NET |
| qq.com | 250,000,000 | PHP |
| Bing.com | 230,000,000 | ASP.NET |
| Ask.com | 190,000,000 | SSI |
| Taobao.com | 170,000,000 | PHP |
| Twitter | 160,000,000 | RubyonRails |
| Adobe.com | 160,000,000 | ColdFusion |

<http://rogchap.com/2011/09/06/top-40-website-programming-languages/>

Apskatot rezultātus, kas redzami tabulā nr. par pasaulē populārāko vietņu izvēlēto programmēšanas valodu pilnīgi noteikti var apgalvot, ka PHP ir līderis tīmekļa vietņu izstrādē, tādēļ ir vērts izvēlēties šo valodu, jo tā var nodrošināt labu darbību.

## 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.

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 licensi. 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

### Django ietvars

Django ir Python izstrādes valodas ietvars

### Java Play ietvars

Java Play ir Java izstrādes valodas ietvars

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 | - | - | - | + |

# praktiskā daļa

Prasību specifikācija ir izstrādāta atbilstoši Latviajs 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 |  |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Ziņas nosūtīšana |  |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Ziņu meklēšana |  |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Ziņu saņemšana/lasīšana |  |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Ziņu dzēšana |  |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Komentāra pievienošana rakstam |  |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Komentāra pievienošana aptaujai |  |  |  |
|  | Administrators, autors, reģistrēts lietotājs | Profila informācijas labošana |  |  |  |
|  | Administrators, autors | Raksta pievienošana |  |  |  |
|  | Administrators, autors | Raksta labošana |  |  |  |
|  | Administrators, autors | Raksta dzēšana |  |  |  |
|  | Administrators, autors | Aptaujas pievienošana |  |  |  |
|  | Administrators, autors | Aptaujas labošana |  |  |  |
|  | Administrators, autors | Aptaujas dzēšana |  |  |  |
|  | Administrators | Raksta publicēšana |  |  |  |
|  | Administrators | Aptaujas publicēšana |  |  |  |
|  | Administrators | Publicējamo rakstu saraksts |  |  |  |
|  | Administrators | Publicējamo aptauju saraksts |  |  |  |
|  | Administrators | Lietotāju bloķēšana |  |  |  |
|  | Administrators | Lietotāju dzēšana |  |  |  |
|  | Administrators | Tiesību līmeņa mainīšana |  |  |  |
|  | Viesis | Rakstu apskatīšana |  |  |  |
|  | Viesis | Reģistrācija | Jauna lietotāja reģistrācija sistēmā, lai šim lietotājam tiku piešķirtas papildus |  |  |

### Funkcionālās prasības

#### UC-1.1 Ierakstīties sistēmā

|  |  |
| --- | --- |
| **Aktieri:** | Administrators |
| **Mērķis:** | Ierakstīties sistēmā |
| **Īss apraksts:** | Administrators 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:** |  |

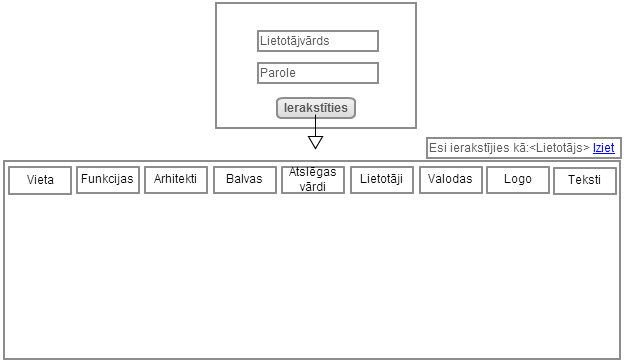
Tipiskā notikumu secība:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **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

Lietošanas gadījuma scenārija realizācijas grafiskās vides uzmetums:



#### UC-1.2 Pievienot valodu

|  |  |
| --- | --- |
| **Aktieri:** | Administrators |
| **Mērķis:** | Pievienot valodu |
| **Īss apraksts:** | Sistēmā tiek pievienota valoda, lai nodrošinātu vairāku valodu atbalstu, tiek ievadīta informācija: valodas nosaukums un kods |
| **Tips:** | Galvenais |
| **Atsauces:** |  |

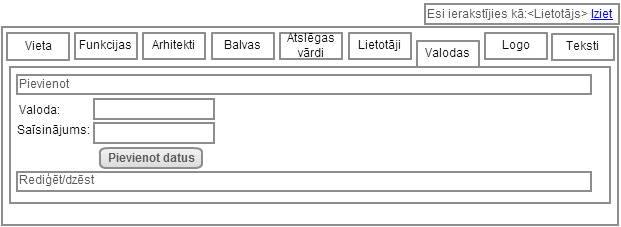
Tipiskā notikumu secība:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators izvēlas sadaļu „Valodas” un aizpilda formu nospiež pogu „Pievienot datus” | 2 | Sistēma pārbauda datus, verifikācija(pārbauda vai aizpildīti visi lauki) |
|  |  | 3 | Dati tiek saglabāti datu bāzē |

Kļūdu apstrāde:

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

Lietošanas gadījuma scenārija realizācijas grafiskās vides uzmetums:



#### UC-1.3 Pievienot logo

|  |  |
| --- | --- |
| **Aktieri:** | Administrators |
| **Mērķis:** | Pievienot logo |
| **Īss apraksts:** | Sistēmā tiek izvēlēta valoda un pievienots logo katrai valodai atsevišķi, ir jāaizpilda teksta redaktora lauks |
| **Tips:** | Galvenais |
| **Atsauces:** | Tiek paplašināts ar UC-1.2 |

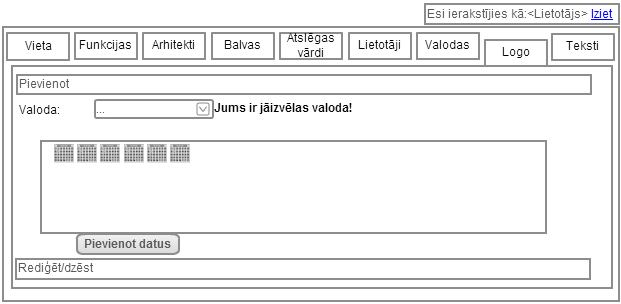
Tipiskā notikumu secība:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators izvēlas sadaļu „Logo” un aizpilda formu nospiež pogu „Pievienot datus” | 2 | Sistēma pārbauda datus, verifikācija(pārbauda vai aizpildīti visi lauki kā arī pārbauda vai šai valodai jau nav pievienots logo) |
|  |  | 3 | Dati tiek saglabāti datu bāzē |

Kļūdu apstrāde:

2: Sistēma konstatē, ka nav aizpildīti visi lauki vai arī šai valodai jau ir logo→ Sistēma paziņo par to lietotājam

Lietošanas gadījuma scenārija realizācijas grafiskās vides uzmetums:



#### UC-1.4 Pievienot tekstus

|  |  |
| --- | --- |
| **Aktieri:** | Administrators |
| **Mērķis:** | Pievienot tekstus |
| **Īss apraksts:** | Sistēmā tiek pievienoti statiskie teksti, lai katrai valodai tie būtu citi |
| **Tips:** | Galvenais |
| **Atsauces:** | Tiek paplašināts ar UC-1.2 |

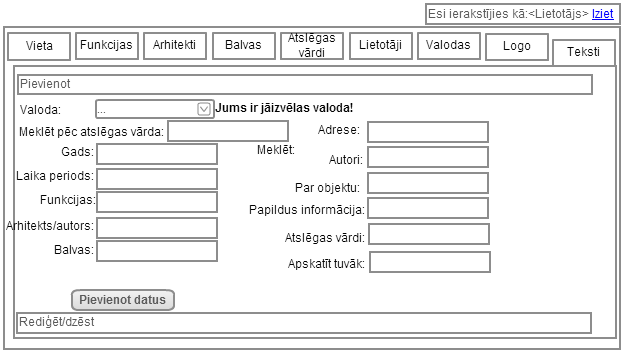
Tipiskā notikumu secība:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators izvēlas sadaļu „Teksti” un aizpilda visus formas laukus un nospiež pogu „Pievienot datus” | 2 | Sistēma pārbauda datus, verifikācija(pārbauda vai aizpildīti visi lauki kā arī pārbauda vai šai valodai jau nav pievienoti teksti) |
|  |  | 3 | Dati tiek saglabāti datu bāzē |

Kļūdu apstrāde:

2: Sistēma konstatē, ka nav aizpildīti visi lauki vai arī šai valodai jau ir pievienoti teksti→ Sistēma paziņo par to lietotājam

Lietošanas gadījuma scenārija realizācijas grafiskās vides uzmetums:



#### UC-1.5 Pievienot atslēgas vārdu

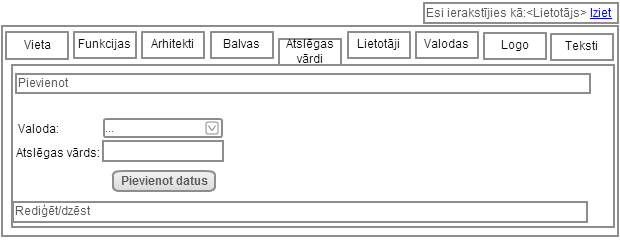
|  |  |
| --- | --- |
| **Aktieri:** | Administrators |
| **Mērķis:** | Pievienot tekstus |
| **Īss apraksts:** | Sistēmā tiek izvēlēta valoda un ievadīts atslēgas vārds |
| **Tips:** | Galvenais |
| **Atsauces:** | Tiek paplašināts ar UC-1.2 |

Tipiskā notikumu secība:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aktiera darbība** |  | **Sistēmas reakcija** |
| 1 | Administrators izvēlas sadaļu „Atslēgas vārdi” un pēc tam izvēlas valodu un ievada atslēgas vārdu , nospiež pogu „Pievienot datus” | 2 | Sistēma pārbauda datus, verifikācija(pārbauda vai aizpildīti visi lauki) |
|  |  | 3 | Dati tiek saglabāti datu bāzē |

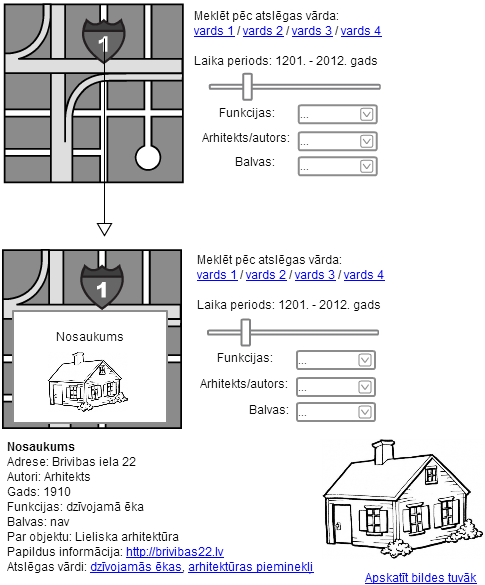
Kļūdu apstrāde:

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

Lietošanas gadījuma scenārija realizācijas grafiskās vides uzmetums:

Tieši tāpat tiek pievienotas arī funkcijas(UC-1.8), tāpēc to vairāk neatkārtosim.

Lietošanas gadījuma scenārija realizācijas grafiskās vides uzmetums:

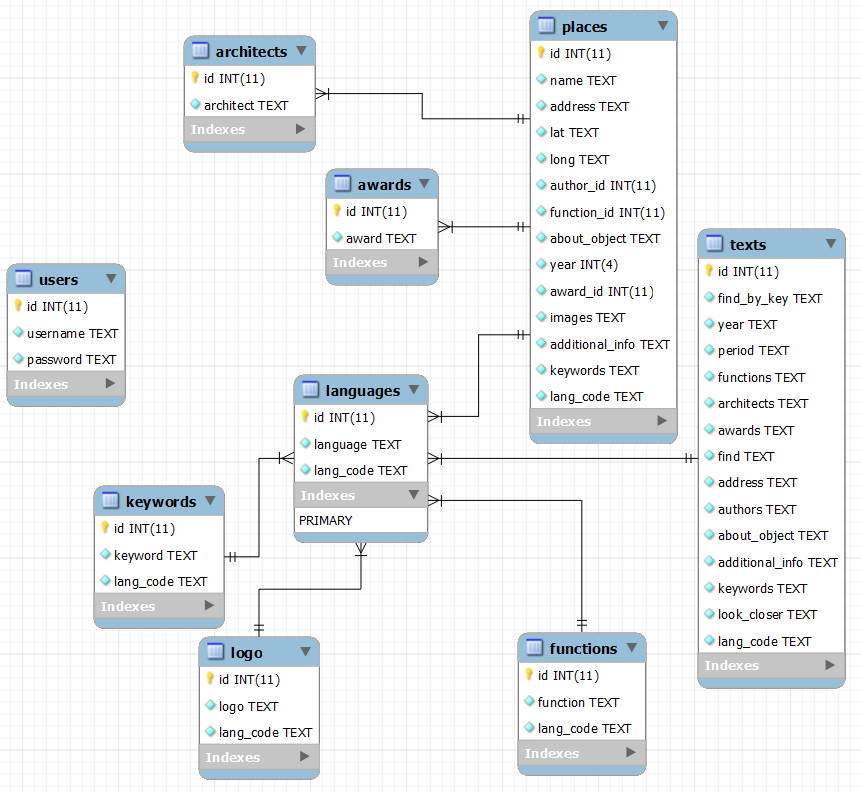


## Programmatūras projektējuma apraksts

Pēc PPA struktūras.

### Datu bāzes struktūra

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



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)