RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

Lietišķo datorsistēmu institūts

Lietišķo datorzinātņu katedra

**Jānis Polis**

**RĪGAS ARHITEKTŪRAS CELVEŽA TĪMEKĻA LIETOTNES IZSTRĀDE**

**Bakalaura darbs**

Darba vadītāja

Dr.sc.ing, prof.

Oksana Ņikiforova

Rīga, 2012

###### 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](#_Toc380659054)

[1. zinātniskā daļa 9](#_Toc380659055)

[1.1. Problēmas apraksts 9](#_Toc380659056)

[1.2. Tehnoloģijas izvēle 9](#_Toc380659057)

[1.2.1. Operētājsistēmas izvēle 9](#_Toc380659058)

[1.2.2. Servera izvēle 11](#_Toc380659059)

[1.2.3. Programmēšanas valodas izvēle 14](#_Toc380659060)

[1.2.4. Programmēšanas ietvara izvēle 15](#_Toc380659061)

[2. praktiskā daļa 18](#_Toc380659062)

[2.1. Sistēmas prasības 18](#_Toc380659063)

[2.1.1. Produkta funkcijas 18](#_Toc380659064)

[2.1.2. Funkcionālās prasības 22](#_Toc380659065)

[2.2. Programmatūras projektējuma apraksts 28](#_Toc380659066)

[2.2.1. Datu bāzes struktūra 28](#_Toc380659067)

[Nobeigums 29](#_Toc380659068)

[Pielikumi 30](#_Toc380659069)

[Literatūra 33](#_Toc380659070)

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 informācijai svešvalodās.

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 Rīgas arhitektūras tīmekļa lietotni, piemeklējot tai atbilstošākās tehnoloģijas balstoties uz pasūtītāja izvirzītajām prasībām un bagātināt tās ar autora idejām. Tīmekļa lietotnes pasūtītāji ir no RTU arhitektūras un pilsētplānošanas fakultātes ar ko kopā nepieciešams noteikt lietotnes prasības.

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

* Izpētīt tehnoloģijas interaktīvo karšu izstrādei.
* Apkopot informāciju par ģeogrāfisko datu uzkrāšanas formātiem un to apstrādes tehnoloģijām.
* Definēt prasības lietojumsistēmas izstrādei un aprakstīt tās programmatūras prasību specifikācijā.
* Izveidot sistēmas arhitektūru un izstrādāt programmatūras projektējuma aprakstu.
* Implementēt uzdoto lietojumsistēmu.
* Demonstrēt sistēmas pamata funkcionalitāti strādājošas sistēmas ekrānšāviņos.

<<Satura izklāsts>>

# zinātniskā daļa

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

## 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.

## Tehnoloģijas 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

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

Kā arī Unix 1999. gadā sāka iegūt pozīcijas serveru OS jomā, Linux ieguva 2 [xx]. Vietu aiz Windows NT. 2001.gadā Linux tirgus daļa, bija 25%, citi Unix produkti 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, ka tikko 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
* Unix operētājsistēmās ir iebūvētas vairāk drošības un datņu atļaujas 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. Unix gadījumā 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

Daudzas mitināšanas 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 apache http server. 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 sava datira, vai arī lejupielādēt bināro versiju, ko var lietot daudzās operētājsistēmās (piemēram, Windows, Solaris, Linux, OS / 2, FreeBSD, u.c.). Šim serverim ir pieejami daudz dažādi papildinājumi. 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 Information Services (IIS)

Internet Information Services (IIS) ir Microsoft produkts tīmekļa serveru arēnā. Ja tiek lietots serveris 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 lietototu tīmekļa pakalpojumus.

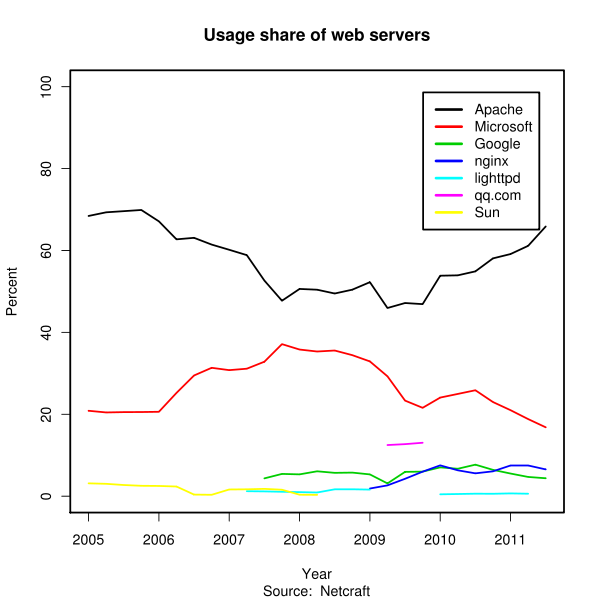
Sun Java Web Server

Trešais būtiskais tīmekļa serveris grupa ir Sun Java Web Server, šobrīd pazīstams kā Oracle iPlanet. Šis ir visbiežāk lietotais serveris korporācijām, kas izmanto Unix tīmekļa 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.

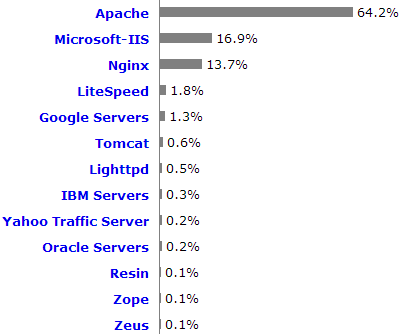
<http://webdesign.about.com/od/servers/a/aa092099.htm>

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 |
| Oracle iPlanet 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 | Ruby on Rails |
| 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.

### Programmēšanas ietvara izvēle

Tā kā tika izvēlēta servera puses programmēšanas valoda PHP, lai atvieglotu izstrādi tiks izvērtēti dažādi ietvari, konkrētāk 5 šobrīd populārākie ietvari tie ir:

1. Yiiframework – uz doto brīdi pats populārākais, bet ir salīdzinoši jauns ietvars ar mazu informācijas daudzumu, tādēļ grūti iemācīties to lietot
2. CodeIgniter – šobrīd arī ļoti populārs izstrādātāju vidū, ļoti ātrdarbīgs, viegli iemācīties, vienkārša saskarne un loģiskā struktūra bibliotēkām, ļauj radoši koncentrēties uz savu projektu, samazinot koda apjomu, kas vajadzīgs konkrētā uzdevuma risināšanai. Izmanto MVC principus.
3. Zend – drošs, tiek lietots izstrādājot lielus projektus, grūti iemācīties īsā laika posmā
4. CakePHP – vienkāršs, izmanto MVC principus, dokumentācija nav tā labākā, bet ir labs lietotāju komūnas atbalsts.
5. Symfony – bagāts ar daudziem instrumentiem un klasēm, kuru mērķis ir saīsināt izstrādes laiku. Tas ir rūpīgi pārbaudīts dažādos reālos pasaules projektos, un ir faktiski izmantots augsta pieprasījuma e-biznesa tīmekļa vietnēs.

<http://www.phpzag.com/top-5-php-frameworks-2012/>

Tā kā ir svarīgi, lai būtu atbalsts google kartei ir jāapskata ko piedāvā šie ietvari:

Apskatot ko piedāvā Yiiframework ir skaidri redzams, ka šis ietvars ir patiešām ļoti jauns, jo google maps papildinājums vēl ir izstrādes stadijā.

<http://www.yiiframework.com/extension/egmap/>

Codeigniter piedāvā pievienot papildus bibliotēku ar kuras palīdzību var īstenot jebkāda veida darbības google maps kartē, šai bibliotēkai ir uzrakstīta ļoti laba dokumentācija kā arī ir iespējams apskatīt demo kartes ar koda paraugiem, kas ļauj ļoti ātri un vienkārši iemācīties lietot šo bibliotēku. Ļoti lielu koda daļu izveido pats ietvars, tam ir nepieciešams tikai ievadīt dažus konfigurācijas parametrus kontrolierī un pēc tam tiek uzģenerēta karte, kuru ar vienu rindiņu ir iespējams parādīt skatā.

<http://biostall.com/codeigniter-google-maps-v3-api-library>

Zend ietvars ļauj lietot google karti tikai izmantojot google API(iepriekš definētu klašu, procedūru, funkciju, struktūru un konstanšu kopums, kas tiek pasniegts kā pielikums, kuru iespējams izmantot ārējiem programmatūras produktiem), ar unikālu reģistrācijas numuru un faktiski visu funkcionalitāti nodrošina pats programmētājs, nav izveidots nekas tāds ar kā palīdzību tiek izveidots kartes kods.

<http://lv.wikipedia.org/wiki/API>

<http://www.bitconsultants.net/2010/geocoding-with-google-maps-and-the-zend-framework/>

Līdzīgi kā Codeigniter ietvarā arī CakePHP izmanto gatavu risinājumu, kuru ir nepieciešams konfigurēt padodot kontrolierī dažas vērtības, kuras tiek apstrādātas un pēc tam parādīts rezultāts.

<http://bakery.cakephp.org/articles/rajibahmed/2010/11/20/cakephp_google_map_v3_helper>

Symfony piedāvā spraudni ar kura palīdzību symfony projektam var pievienot google kartes funkcijas.

<http://www.symfony-project.org/plugins/sfEasyGMapPlugin>

Tā kā CodeIgniter ietvars ir visai populārs un tam ir ļoti laba dokumentācija un lietotāju atbalsts tika izvēlēts šis ietvars.

Sakarā ar to ka ietvari izmanto MVC arhitektūru ir nepieciešams izsprast ko īsti tas nozīmē. Tātad MVC ir trīs vārdu saīsinājums Model(modelis), View(skats), Controller(kontrolieris). Galvenais ir izprast ko katrs no šiem trim dara:

* Modelis ir biznesa loģika, citiem vārdiem sakot, viss, kas saistīts ar datiem no datu bāzes
* Skats ir prezentācijas loģika jeb datu izvade gala lietotājam
* Kontrolieris apstrādā pieprasījumus no lietotāja, saņem datus no datu bāzes un padod tos skatam <http://www.oreillynet.com/onlamp/blog/2007/06/what_is_mvc.html>

# 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 utml pēs PPS struktūras.

### Produkta funkcijas

Tā kā projekta ietvaros ir paredzēts izveidot arhitektūras ceļvedi, kuru apmeklēs arī tūristi, tad būs nepieciešams vairāku valodu atbalsts.

Jāizstrādā administrācijas panelis, kurā ir nepieciešams realizēt datu ievade, rediģēšāna un dzēšana objektiem.

<http://arhitektura.net63.net/> saita adrese

<http://arhitektura.net63.net/login>

lietotajvards:admin

parole:pass

2.tabula. Lietošanas gadījumu īss apraksts

| **Nosaukums un šifrs** | **Aktieri** | **Mērķis** | **Īss apraksts** | **Prioritāte izpildīšanai** | **Atsauces** |
| --- | --- | --- | --- | --- | --- |
| UC-1.1 Ierakstīties sistēmā | Administrators | Ierakstīties sistēmā, lai veiktu darbu | 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 | Galvenais |  |
| UC-1.2 Pievienot valodu | Administrators | Pievienot valodu | Sistēmā tiek pievienota valoda, lai nodrošinātu vairāku valodu atbalstu, tiek ievadīta informācija: valodas nosaukums un kods | Galvenais |  |
| UC-1.3 Pievienot logo | Administrators | Pievienot logo | Sistēmā tiek izvēlēta valoda un pievienots logo katrai valodai atsevišķi, ir jāaizpilda teksta redaktora lauks | Galvenais | Tiek paplašināts ar UC-1.2 |
| UC-1.4 Pievienot tekstus | Administrators | Pievienot statiskos tekstus | Sistēmā tiek pievienoti statiskie teksti, lai katrai valodai tie būtu citi | Galvenais | Tiek paplašināts ar UC-1.2 |
| UC-1.5 Pievienot atslēgas vārdu | Administrators | Pievienot atslēgas vārdu | Sistēmā tiek izvēlēta valoda un ievadīts atslēgas vārds | Galvenais | Tiek paplašināts ar UC-1.2 |
| UC-1.6 Pievienot arhitektu | Administrators | Pievienot arhitektu | Sistēmā tiek pievienots arhitekts | Galvenais | Tiek paplašināts ar UC-1.2 |
| UC-1.7 Pievienot balvu | Administrators | Pievienot balvu | Sistēmā tiek pievienota balva | Galvenais | Tiek paplašināts ar UC-1.2 |
| UC-1.8 Pievienot funkciju | Administrators | Pievienot objekta funkciju | Sistēmā tiek izvēlēta valoda un pievienota funkcija, kuru var veikt konkrētais arhitektūras objekts | Galvenais | Tiek paplašināts ar UC-1.2 |
| UC-1.9 Pievienot arhitektūras objektu | Administrators | Pievienot jaunu arhitektūras objektu | Sistēmā tiek izvēlēta valoda, pēc tam tiek ievadīti dati: objekta nosaukums, adrese, ģeogrāfiskās koordinātes(garums un platums) , tiek izvēlēti dati no iepriekš pievienotajiem arhitektiem, funkcijām, balvām un atslēgas vārdiem, kā arī tiek ievadīta informācija par objektu teksta redaktora laukā, tiek ievadīts ēkas uzcelšanas gads, tiek norādītas saites uz attēliem(maksimāli 10 attēli), kā arī tiek norādīta saite uz papildus informāciju, ja tāda ir | Galvenais | Tiek paplašināts ar UC-1.2,  UC-1.5, UC-1.6, UC-1.7, UC-1.8 |
| UC-1.10 rediģēt vai dzēst valodu | Administrators | Rediģēt valodu | Sistēmā tiek izvēlēta valoda, kuru var rediģēt nospiežot uz rediģēšanas pogu, pēc izvēles tiek parādīts dialoglogs ar šīs valodas atribūtiem, kurus var rediģēt un pēc tam saglabāt, bet ja tiek izvēlēta poga, zem nosaukuma dzēst, tad valoda tiek izdzēsta | Galvenais | Paplašina UC-1.2 |
| UC-1.11 rediģēt logo | Administrators | Rediģēt logo | Sistēmā tiek izvēlēta valoda un pēc tam, ja logo šai valodai ir pievienots, tad ir iespēja mainīt informāciju teksta redaktorā logā un saglabāt | Galvenais | Tiek paplašināts ar UC-1.2 un paplašina UC-1.3 |
| UC-1.12 rediģēt tekstus | Administrators | Rediģēt tekstus | Sistēmā tiek izvēlēta valoda un pēc tam, ja valodai ir pievienoti teksti, tad tos var rediģēt un saglabāt | Galvenais | Tiek paplašināts ar UC-1.2 un paplašina UC-1.4 |
| UC-1.13 rediģēt vai dzēst atslēgas vārdus | Administrators | Rediģēt atslēgas vārdus | Sistēmā tiek izvēlēts atslēgas vārds nospiežot uz rediģēšanas pogas un pēc tam tiek parādīts dialoglogs ar atslēgas vārdu, kuru var rediģēt un saglabāt vai arī nospiežot dzēšanas pogu šis atslēgas vārds tiek izdzēsts | Galvenais | Paplašina UC-1.5 |
| UC-1.14 rediģēt vai dzēst arhitektu | Administrators | Rediģēt arhitektu | Sistēmā tiek izvēlēts arhitekts nospiežot rediģēšanas pogu, tad tiek parādīts dialoglogs, kurā ir iespējams veikt izmaiņas un saglabāt vai arī nospiežot dzēšanas pogu arhitekts tiek izdzēsts no datu bāzes | Galvenais | Paplašina UC-1.6 |
| UC-1.15 rediģēt vai dzēst balvu | Administrators | Rediģēt balvu | Sistēmā tiek izvēlēta balva nospiežot rediģēšanas pogu pie attiecīgās balvas un tad tiek parādīts dialoglogs, kurā var rediģēt šo balvu un saglabāt | Galvenais | Paplašina UC-1.7 |
| UC-1.16 rediģēt vai dzēst funkciju | Administrators | Rediģēt funkciju | Tiek izvēlēta funkcija nospiežot rediģēšanas pogu, pēc tam tiek parādīts dialoglogs, kurā var rediģēt šo funkciju un saglabāt | Galvenais | Paplašina UC-1.8 |
| UC-1.17 rediģēt vai dzēst arhitektūras objektu | Administrators | Atlasīt arhitektūras objektu, lai rediģētu vai dzēstu | Tā kā arhitektūras objektu būs ļoti daudz, tad lai atvieglotu objekta atrašanu ir iespēja meklēt objektu pēc nosaukuma , adreses vai gada kā arī neizvēloties nevienu no šiem kritērijiem tiek parādīti visi objekti. Pēc atlasītās informācijas, atrod konkrēto objektu un nospiežot uz rediģēšanas pogas tiek parādīta papildus sadaļa, kurā var veikt objekta izmaiņas un saglabāt tās, vai arī nospiest dzēšanas pogu un izdzēst konkrēto objektu | Galvenais | Paplašina UC-1.9 |
| UC-1.18 izrakstīties no sistēmas | Administrators | Izrakstīties no sistēmas | Administrators beidzot darbu nospiež uz saites un sesija tiek izdzēsta, tiek parādīta ierakstīšanās forma | Galvenais | Paplašina UC-1.1 |
| UC-1.19 pievienot lietotāju | Administrators | Pievienot administratora tipa lietotāju | Administrators var pievienot vēl kādu lietotāju ievadot lietotājvārdu un paroli | Papildus |  |
| UC-1.20 rediģēt lietotāja datus | Administrators | Rediģēt administratora datus | Administrators var rediģēt savus datus, bet ne cita adminstratora datus | Papildus |  |
| UC-2.1 apskatīt arhitektūras objektu | Lietotājs | Apskatīt objektu | Lietotājs nospiežot ar peli uz kādu no kartē esošajiem marķieriem redz uznirstošu logu uz kartes, kurā ir redzama bilde un objekta nosaukums, kā arī zem kartes ir redzama detalizētāks objekta apraksts | Galvenais |  |
| UC-2.2 apskatīt bildes | Lietotājs | Apskatīt bildes | Zem kartes ir informācija par objektu tai skaitā arī bildes, ir redzama viena neliela bilde, un nospiežot uz tās tiek nomainīta bilde | Galvenais | Paplašina UC-2.1 |
| UC-2.3 apskatīt bildes tuvplānā | Lietotājs | Apskatīt bildes tuvplānā | Zem mazās bildes ir saite, kuru nospiežot tiek parādīta bilde uznirstošajā logā pilnā izmērā | Galvenais | Paplašina UC-2.1 un UC-2.2 |
| UC-2.4 izvēlēties kritērijus atlasei un atlasīt objektus | Lietotājs | Izvēlēties kritērijus atlasei un veikta atlasi | Lietotājs var izvēlēties kritērijus: laika periods, funkcijas, arhitekts un balvas, pēc pogas nospiešanas tiek atlasīti objekti, kas atbilst šiem kritērijiem un tiek parādīti atrastie objekti kartē | Papildus |  |
| UC-2.5 meklēt pēc atslēgas vārda | Lietotājs | Meklēt pēc atslēgas vārda | Nospiežot uz vienu no atslēgas vārdiem tiek veikta objekt atlase un parādīti rezultāti kartē | 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:** |  |

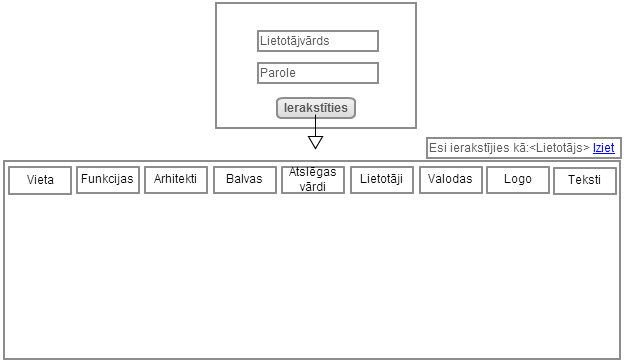
Tipiskā notikumu secība:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **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:** |  |

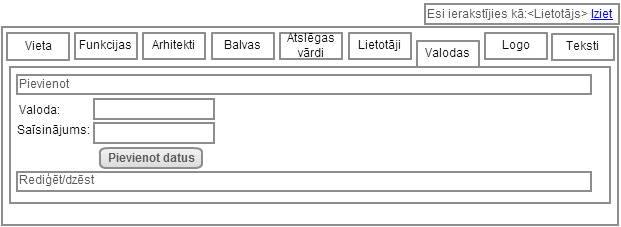
Tipiskā notikumu secība:

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

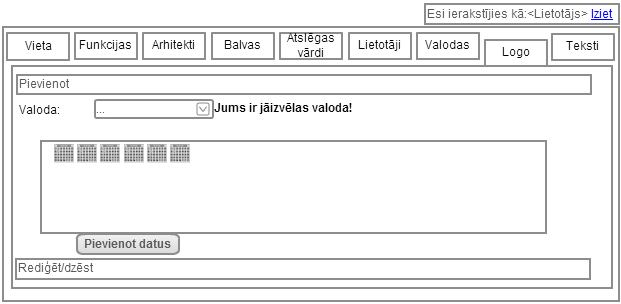
Tipiskā notikumu secība:

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

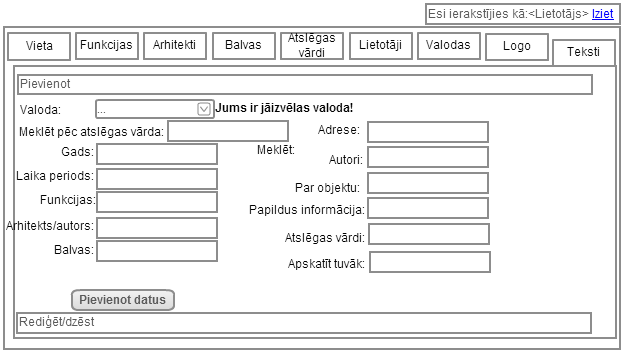
Tipiskā notikumu secība:

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

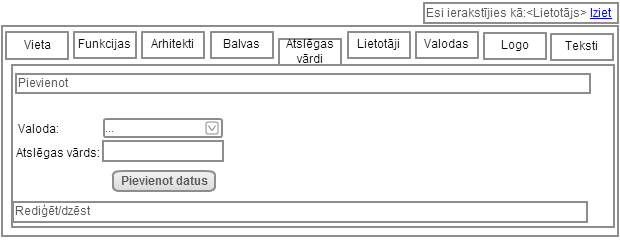
|  |  |
| --- | --- |
| **Aktieri:** | Administrators |
| **Mērķis:** | Pievienot tekstus |
| **Īss apraksts:** | Sistēmā tiek izvēlēta valoda un ievadīts atslēgas vārds |
| **Tips:** | Galvenais |
| **Atsauces:** | Tiek paplašināts ar UC-1.2 |

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 |

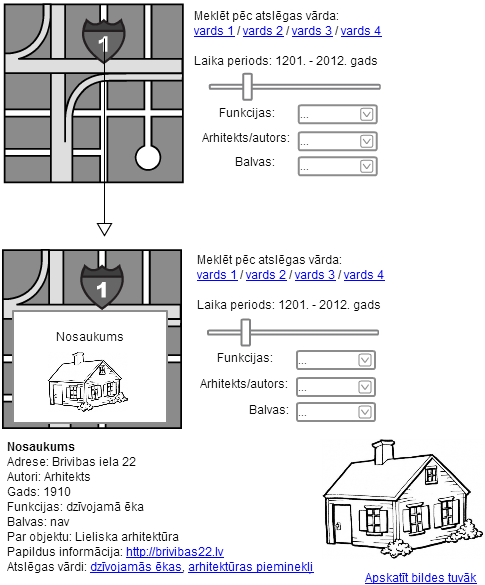
Kļūdu apstrāde:

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:

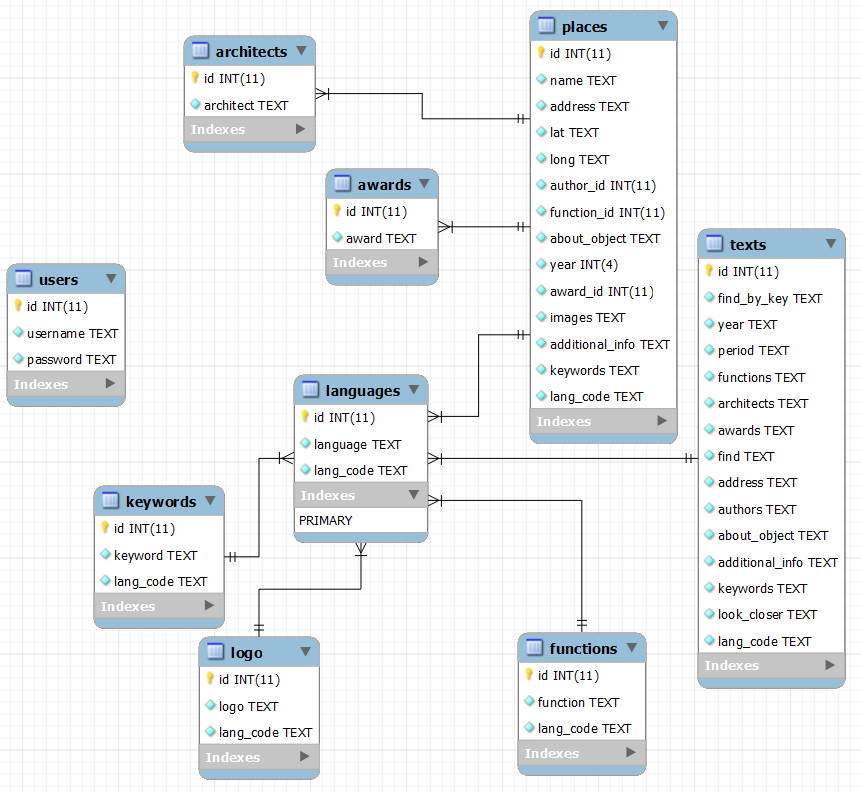


## Programmatūras projektējuma apraksts

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
* Live Riga: 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

<http://codeigniter.com/>

<http://jqueryui.com/>

<http://mottie.github.com/Keyboard/>

<https://github.com/furf/jquery-ui-touch-punch>.

<http://termini.lza.lv/term.php>

<http://www.gliffy.com/gliffy/#d=4193183&t=login>