

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

DRAUGIEM CMS
KVALIFIKĀCIJAS DARBS

Autors: **Raivis Gelsbergs**

Stud. Apl. nr.: RG07008

Darba vadītājs: lektors Dainis Dosbergs

RĪGA 2009

ANOTĀCIJA

Draugiem CMS ir satura vadības sistēma, kas piedāvā draugiem.lv sistēmā iekļauto projektu izstrādātājiem viegli integrējamus un modificējamus moduļus ātrākai projektu izstrādei. Papildus priekšrocībām, ko sistēmas izmantošana dod izstrādātājam, tā arī nodrošina gala lietotāja saskarni informācijas apmaiņai ar lietotājiem. Sistēma piedāvā sešus visbiežāk lietotos moduļus, kas ir viegli pielietojami standarta projektiem, kā arī uz to bāzes iespējams veidot sarežģītākas sistēmas.

ABSTRACT

Draugiem CMS is a content management system that gives access to easily modifiable and integrable modules for draugiem.lv system's sub-project developers for developing new projects. In addition to the advantages, which the system gives to the developers, it also provides with an user interface for data exchange with an end-user. The system provides with six of the most commonly used modules that are easy to use for standard projects; similarly, the modules could be used as a base for creating more complex systems.

SATURA RĀDĪTĀJS

APZĪMĒJUMU SARAKSTS.....	7
IEVADS.....	9
1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA.....	10
1.1. Ievads.....	10
1.1.1. Nolūks	10
1.1.2. Darbības sfēra	10
1.2. Vispārējais apraksts	10
1.2.1. Produkta perspektīva	10
1.2.2. Produkta funkcijas	11
1.2.2.1. Sistēmas funkcijas	11
1.2.2.2. Satura vadības moduļi	11
1.2.3. Lietotāja raksturozīmes	12
1.2.4. Vispārējie ierobežojumi	12
1.2.5. Pieņēmumi un atkarības	12
1.3. Konkrētās prasības	13
1.3.1. Funkcionālās prasības	13
1.3.1.1. Projektu pārvaldība.....	13
1.3.1.2. Komentāru pārvaldības modulis.....	14
1.3.1.3. Jaunumu pārvaldības modulis	14
1.3.1.4. Galeriju pārvaldības modulis	15
1.3.1.5. Balsošanas pārvaldības modulis.....	17
1.3.1.6. Foruma pārvaldības modulis	17
1.3.1.7. Projekta atbalstītāju pārvaldības modulis.....	18
1.3.2. Ārējās saskarnes prasības	19
1.3.2.1. Lietotāja saskarne	19
1.3.2.2. Programmatūras saskarne	19
1.3.2.3. Sakaru saskarne	19
1.3.3. Veiktspējas prasības	19
1.3.4. Projekta ierobežojumi	20
1.3.4.1. Atbilstība standartiem	20
1.3.4.2. Aparatūras ierobežojumi	20
1.3.5. Atribūti	20
1.3.5.1. Drošība	20
1.3.5.2. Uzturamība	20
1.3.5.3. Pārnēsāmība	20

2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS.....	21
2.1. Ievads	21
2.1.1. Nolūks	21
2.1.2. Darbības sfēra	21
2.2. Dekompozīcijas apraksts	21
2.2.1. Moduļu dekompozīcija	21
2.2.2. Datu dekompozīcija	22
2.3. Atkarību apraksts	23
2.3.1. Starpmoduļu atkarības	23
2.3.2. Datu atkarības	24
2.4. Detalizēts projektējums.....	25
2.4.1. Moduļu detalizēts projektējums	25
2.4.1.1. Jaunumu pārvaldības klase “News”.....	25
2.4.1.2. Komentāru pārvaldības klase “Comments”.....	27
2.4.1.3. Balsošanas pārvaldības klase “Voting”	28
2.4.1.4. Atbalstītāju pārvaldības klase “Fans”.....	28
2.4.1.5. Forumu pārvaldības klase “Forum”.....	29
2.4.1.5. Galeriju pārvaldības klase “Gallery”.....	31
2.4.2. Datu detalizēts projektējums.....	33
2.4.2.1. Tabula “project”.....	33
2.4.2.2. Tabula “project_comments”.....	34
2.4.2.3. Tabula “project_comments_stats”.....	34
2.4.2.4. Tabula “project_fans”.....	34
2.4.2.5. Tabula “project_forum”.....	35
2.4.2.6. Tabula “project_forum_reply”.....	35
2.4.2.7. Tabula “project_gallery_alb”.....	36
2.4.2.8. Tabula “project_gallery_pics”.....	36
2.4.2.9. Tabula “project_news”.....	37
2.4.2.10. Tabula “project_voting”.....	37
2.4.2.11. Tabula “project_voting_stats”.....	37
3. TESTĒŠANAS DOKUMENTĀCIJA.....	39
3.1. Testēšanas plāns.....	39
3.2. Testēšanas žurnāls.....	39
3.2.1. Projektu pārvaldības panelis.....	39
3.2.2. Komentāru pārvaldības modulis.....	40
3.2.3. Jaunumu pārvaldības modulis.....	41

3.2.4. Balsošanas pārvaldības modulis.....	41
3.2.5. Projekta atbalstītāju pārvaldības modulis.....	42
3.2.6. Foruma pārvaldības modulis	42
3.2.7. Galeriju pārvaldības modulis	43
3.3. Testēšanas rezultātu kopsavilkums.....	44
4. PROJEKTA ORGANIZĀCIJA.....	45
5. KVALITĀTES NODROŠINĀŠANA.....	46
6. KONFIGURĀCIJU PĀRVALDĪBA.....	47
7. DARBIETILPĪBAS NOVĒRTĒJUMS.....	48
8. NOBEIGUMS.....	50
9. IZMANTOTĀ LITERATŪRA UN AVOTI.....	51
PIELIKUMI.....	52
1. Programmas koda fragments.....	52
DOKUMENTĀRĀ LAPA.....	58

APZĪMĒJUMU SARAKSTS

Draugiem CMS – kvalifikācijas darba izstrādātā satura vadības sistēma.

PHP – atklātā pirmkoda programmēšanas valoda, kuru plaši izmanto dinamiskai tīmekļa lappušu ģenerēšanai servera pusē.

MySQL – populāra datu bāzu vadības sistēma.

HTTP – Hyper Text Transfer Protocol, galvenais informācijas pārraides protokols globālajā tīmeklī.

Satura vadības sistēmas modulis – nosacīti neatkarīga satura vadības sistēmas daļa, kas veic noteiktu funkciju.

HTML – Hyper Text Markup Language – iezīmēšanas valoda, ko izmanto teksta un citu datu attēlošanai Interneta pārlūkprogrammā.

XHTML – HTML līdzīga iezīmēšanas valoda, kas seko XML standartam un prasa striktāku korektas sintakses ievērošanu kā HTML.

CSS – Cascading Style Sheets – valoda, ko izmanto tīmekļa lapās esošās informācijas vizuālajai noformēšanai, definējot konkrēta elementa vai noteikta tipa elementu izskatu.

JavaScript – programmēšanas valoda, kuras kods tiek izpildīts klienta pārlūkprogrammā, tādējādi dodot iespēju manipulēt ar tīmekļa lapu pēc tās ielādes.

SQL – datu bāzu vaicājumu valoda.

SQL injekcija – sistēmas drošības caurums, kas atļauj lietotājam modificēt SQL vaicājumus, nosūtot specifiskus ievaddatus, kas netiek pārbaudīti pirms iekļaušanas vaicājumā.

Cross Site Scripting – drošības caurums, kas ļauj lietotājam ievaddatos iekļaut HTML/JavaScript kodu, kas pēc tam tiek attēlots lapā.

PK – primārā atslēga, datubāzes tabulas lauks, kas unikāli identificē tabulas ierakstu.

Forma – informācijas ievades lauku kopa Interneta lappusē, kas ļauj lietotājam ievadīt datus un nosūtīt tos uz serveri.

Sīktēls – attēla samazināta versija, parasti izmanto, lai pārskatāmā veidā vienlaicīgi parādītu daudzus attēlus.

API – application programming interface – programmatūras saskarne, dokumentēts kādas programmatūras vienības ārēji pieejamo un izmantojamo funkciju saraksts.

PHPdoc – PHP koda komentāru pieraksts, kas speciālā formā ļauj kodā ietvert mainīgo, klašu, metožu un funkciju dokumentāciju, no kuras vēlāk, var tikt ģenerēta programmaprodukta API dokumentācija.

Memcached – augstas veiktspējas atmiņas objektu kešservera programmatūra, kas pamatā paredzēta, lai paātrinātu tīmekļa lietotnes ātrumu, samazinot datubāžu pārvaldības sistēmas noslodzi.

HTTP serveris – programmatūra, kuras uzdevums ir atbildēt uz lietotāju HTTP pieprasījumiem, atbildot ar HTTP atbildēm, kas satur dažādus datus. Parasti šie dati ir HTML kods.

Subversion – atvērtā pirmkoda versiju kontroles programmatūra.

IEVADS

Veidojot dažādus reklāmas projektus sociālā portāla draugiem.lv ietvaros tika pamanītas vairākas funkcijas un tipiskas sadaļas, kas projektos atkārtojas. Iesākumā citu cilvēku vadībā tika izstrādāts pirmais mēģinājums šos projektus apvienot vienotā sistēmā, lai katrā jaunizveidotā projektā nebūtu no jauna jāveido vai jādublē iepriekš izstrādātais kods.

Pirmais risinājums bija pilnībā standartizēts, iekļaujot nepieciešamos pamata moduļus satura vadības sistēmai. Lietotājam pieejamā daļa arī tika iekļauta sistēmā, kā arī projekta lapas vizuālais izskats tika standartizēts pieļaujot ļoti mazas atkāpes. Moduļu izmantošana ārpus projekta standarta šablona bija ļoti neērta un apgrūtināša no izstrādātāja puses. Šāda pieeja noveda pie tā, ka līdz ar pat nelielām izmaiņām kādā modulī vai arī visbiežāk projekta lapā kopumā, bija jālabo kods, kas bija globāls visām projekta lapām. Līdz ar projektu skaita pieaugumu sistēmā, sistēmu veidojošais kods "apauga" ar daudz nosacījumiem, kas bija nepieciešami tikai kādam konkrētam projektam.

Secinot, ka šāda sistēma nav ērti lietojama un pamazām kļūst aizvien grūtāk pārvaldāma, tika nopriests veidot jaunu sistēmu, saglabājot iepriekšējās sistēmas labās īpašības. Jaunā sistēma tika veidota pēc citas pieejas – standartizēt tikai pašus moduļus, bet neveidot paša projekta dizainu, kā arī atvieglot moduļu iekļaušanu projekta lapā. Svarīgs aspekts bija arī veidot šos moduļus tā, lai tos būtu ērti modificēt, ja tas ir nepieciešams specifiska projekta ietvaros, neveicot modifikācijas paša moduļa struktūrā. Jaunajā sistēmā tika iekļauti svarīgākie satura vadības moduļi, kas visbiežāk ir sastopami reklāmas projektu ietvaros, maksimāli saīsinot izstrādes laiku. Satura vadības moduļos tika iekļauta arī lietotājam pieejamā funkcionalitāte, kā arī tās vizuālais noformējums, taču katra moduļa HTML kods tika veidots vadoties pēc tā, lai standarta gadījumā būtu nepieciešams tikai veikt izmaiņas standarta vizuālajā attēlojumā, labojot CSS kodu. Labot PHP kodu būtu nepieciešams tikai kādas jaunas funkcionalitātes ieviešanai attiecīgajam modulim.

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Ievads

1.1.1. Nolūks

Šis dokuments ir satura vadības sistēmas “Draugiem CMS” (CMS – Content management system, jeb satura vadības sistēma) prasību specifikācija (PPS). Dokumentā aprakstītas prasības, pēc kurām notiks sistēmas izstrāde. Gatavajā produktā jābūt īstenotām visām dokumentā aprakstītajām prasībām.

1.1.2. Darbības sfēra

Programmatūras prasību specifikācija apraksta “Draugiem CMS” satura vadības sistēmu. Sistēma sevī ietver biežāk nepieciešamos moduļus mājas lapu izstrādē. Sistēma nodrošina daudzu lietotāju piekļuvi ar pielāgojamām lietotāju tiesībām uz dažādām moduļu darbībām.

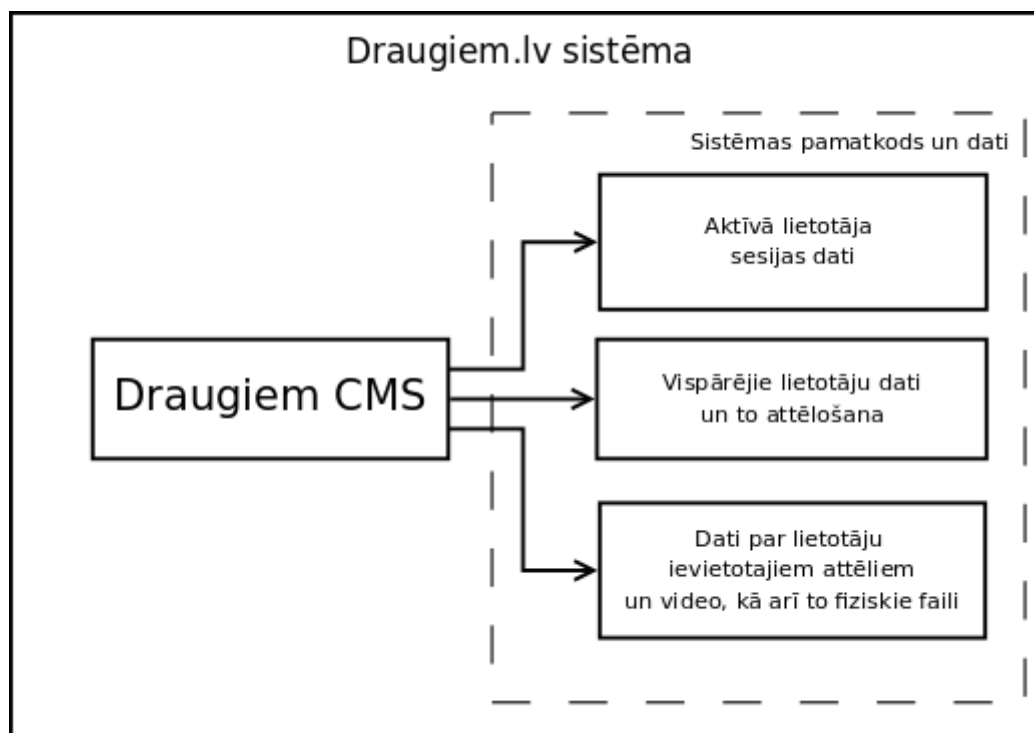
Sistēma sevī ietver ne tikai administratoram un lietotājam pieejamās funkcijas, bet arī standarta moduļu attēlošanu, kuru iespējams modificēt attiecīgā projekta izstrādātājam. Sistēmas mērķauditorija ir projektu izstrādātāji iekš draugiem.lv sociālā tīkla sistēmas, ļaujot viņiem ātri un kvalitatīvi veidot apakšprojektus.

Sistēmas iekšienē ir paredzēti divu veidu lietotāji – administrators un lietotājs, taču ir paredzēts šīs lietošanas tiesības pielāgot pēc izstrādātāja vēlmēm un specifiski šīs funkcijas netiek atdalītas, jo izstrādātājs var brīvi pielāgot noteiktām lietotāju grupām dažādas piekļuves iespējas.

1.2. Vispārējais apraksts

1.2.1. Produkta perspektīva

Sistēma tiks izmantota kā daļa no lielākas sistēmas. Pirms izmantošanas tā tiks integrēta konkrētā draugiem.lv projekta kodā, veicot katram projektam nepieciešamās modifikācijas. Veidojamā sistēma piekļūs vairākām globālās sistēmas daļām, kas attēlots attēlā 3.1.



1.2.2. Produkta funkcijas

Sīkāks funkciju apraksts atrodams nodaļā “Konkrētās prasības”.

1.2.2.1. Sistēmas funkcijas

Ārpus satura vadības moduļiem nepieciešams izveidot projektu pārvaldības paneli, kurā varētu gan pārvaldīt projektus, kas darbojas zem Draugiem CMS sistēmas, gan arī iegūt informāciju un statistiku par jau esošiem projektiem.

1.2.2.2. Satura vadības moduļi

Moduļi, kurus nepieciešams izveidot Draugiem CMS sistēmā, nodrošinot projektiem visbiežāk nepieciešamās iespējas.

- Jaunumu pārvaldības modulis – paredzēts standarta funkciju (labot, pievienot un dzēst) veikšanai ar jaunumiem, kā arī izmantojot komentāru moduli, komentāru pievienošana pie katra ieraksta.
- Komentāru pārvaldības modulis – ļauj pievienot komentāru konkrētam elementa identifikatoram.
- Galeriju pārvaldības modulis – ļauj pārvaldīt albumus un tajos ievietot attēlus un video no draugiem.lv sistēmas lietotāja albumiem. Iespēja pievienot komentārus un balsošanu

par attēliem.

- Balsošanas pārvaldības modulis – ļauj pievienot balsis pie konkrētu elementu identifikatoriem.
- Foruma pārvaldības modulis - ļauj pievienot foruma pavadienus un atbildes uz tiem.
- Projekta atbalstītāju pārvaldības modulis – ļauj lietotājiem pievienoties par atbalstītāju projektam un pēc tam arī atteikties.

1.2.3. Lietotāja raksturiezīmes

Sistēmas galalietotājam ir nepieciešamas pamatiemaņas darbam ar datoru un jāprot rīkoties ar Interneta pārlūkprogrammu.

Programmētājam, kas izmantos sistēmu, izstrādājot jaunus projektus, jāpārzina tās konfigurēšana un darbības principi, kā arī jāspēj izmantot sistēmas piedāvāto datu bāzes struktūru savā produktā vai arī modificēt sistēmu pēc vajadzības.

Sistēma paredzēta pamatā programmētājiem, kas izmantos un pielāgos sistēmu savām vajadzībām, taču ir pieejama arī saskarne gala lietotājiem, kas izmantos satura vadības modulius Interneta pārlūkprogrammā.

1.2.4. Vispārējie ierobežojumi

Sistēmas gala lietotājam nepieciešama Interneta pārlūkprogramma un Interneta pieslēgums, kas ļauj piekļūt serverim uz kura ir uzstādīta sistēma.

Sistēmas funkcijas nedrīkst būt atkarīgas no *JavaScript* atbalsta lietotāja pārlūkprogrammā, sistēmai pilnībā jāfunkcionē gan ar *JavaScript* atbalstu, gan bez tā.

1.2.5. Pieņēmumi un atkarības

Sistēmas darbībai nepieciešams HTTP serveris ar uzstādītu PHP atbalstu. Sistēma darbojas ar PHP 5.x versijām. PHP konfigurācijā jābūt pieejamam MySQL modulis un mbstring paplašinājumam. Sistēmai nepieciešama *MySQL* datubāžu serveris (*MySQL* 4.x vai 5.x versija).

1.3. Konkrētās prasības

1.3.1. Funkcionālās prasības

1.3.1.1. Projektu pārvaldība

Ļauj iegūt informāciju par projektiem, kas ir Draugiem CMS sistēmā un veikt ar tiem standarta operācijas labot informāciju, pievienot un dzēst, kā arī iespēja dzēst konkrētam projektam piesaistītā datus.

Ievade

Labošanas un pievienošanas gadījumā, lietotājam jāizpilda forma, kas satur divus laukus:

1. “Nosaukums” - teksta virkne līdz 200 simboliem
2. “Ceļš uz projekta atrašanās vietu” - teksta virkne līdz 300 simboliem, lietotājs ievada to brīvā tekstā.

Abi aizpildāmie lauki ir obligāti.

Labošanas gadījumā formā tiek ierakstītas vecās lauku vērtības.

Aplūkojot projektu datus, kas izkārtoti tabulas veidā, lai dzēstu konkrēta produkta konkrēta moduļa datus, nepieciešams nospiegt statistikas ikonu projekta rīku joslā un pēc tam, pēc tam nospiegt dzēšanas pogu pie attiecīgā moduļa, kura datus gribam izdzēst.

Apstrāde

Pievienošanas un labošanas gadījumā, tiek pārbaudīts vai dati atbilst prasībām, ja atbilst, tad tiek veiktas izmaiņas datubāzē, pretējā gadījumā parādīts kļūdas paziņojums.

Dzēšanas operācijas izsaukšanas gadījumā, tiek lietotājam parādīts apstiprināšanas paziņojums. Ja tiek saņemts apstiprinājums, projekts tiek dzēsts no datubāzes, kopā ar visiem projekta datiem, kas izveidoti kādā no moduļiem.

Izvaddati

Esošie projekti tiek attēloti tabulā pievienošanas secībā.

Pie visām darbībām, kuru rezultātā tiek veiktas izmaiņu darbības datubāzē, tiek parādīti paziņojumi par to norises rezultātu.

Pie nekorektas ievades, tiek parādīts kļūdas paziņojums.

1.3.1.2. Komentāru pārvaldības modulis

Ļauj pievienot un dzēst komentārus. Programmētājs viena projekta ietvaros var pievienot komentārus vairākiem elementiem, norādot skaitliskus elementu identifikatorus.

Programmētājs var norādīt cik komentāru ierakstus attēlot vienā lapā.

Ievade

Komentāra pievienošanas darbībai nepieciešams tikai ievadīt komentāra tekstu, kas var būt līdz 2000 simboliem un tas ir obligāts lauks.

Komentāru dzēšanas operācija ir pieejama pie katra komentāra un lai to izsauktu, tikai nepieciešams noklikšķināt uz saites ar nosaukumu “[x]”.

Apstrāde

Komentāra pievienošanas gadījumā tiek pārbaudīts vai teksts atbilst prasībām, ja atbilst, tiek izmainīts ieraksts datubāzē, pretējā gadījumā parādīts kļūdas paziņojums.

Dzēšanas operācijas izsaukšanas gadījumā, tiek lietotājam parādīts apstiprināšanas vaicājums. Ja tiek saņemts apstiprinājums, projekts tiek dzēsts no datubāzes.

Izvaddati

Komentāri tiek attēloti tabulas veidā, viens zem otra.

Pievienošanas kļūdas gadījumā tiek parādīts kļūdas paziņojums.

1.3.1.3. Jaunumu pārvaldības modulis

Ļauj veikt standarta operācijas ar jaunumiem: pievienot, labot un dzēst, kā arī pie katra jaunuma ļauj pievienot komentāru.

Jaunumu apskate iespējama divos skatos – lapu skatījumā, kur ir attēloti tikai jaunumu virsraksti un ievadi vai konkrēta ieraksta izvērstajā skatījumā, kur iespējams redzēt visu jaunuma tekstu, kā arī ieraksta komentārus.

Programmētājs var norādīt cik ierakstus vienā lapā attēlot, kā arī iespējot un atspējot iespēju pievienot komentārus.

Ievade

Ierakstu labošanas un pievienošanas gadījumā ir jāaizpilda forma, kas satur laukus:

1. Virsraksts – teksta virkne, līdz 150 simboliem.
2. Ievads – teksta virkne līdz 1000 simboliem
3. Teksts – teksta virkne līdz 5000 simboliem
4. Attēls – lietotājs izvēlas attēlu no kādas savas “draugiem.lv” profila galerijas. Dati par lietotāja galerijām tiek iegūti no jau esošās “draugiem.lv” sistēmas un attēlu augšupielāde netiek veidotā kvalifikācijas darba ietvaros.

Visi lauki, izņemot attēlu ir obligāti.

Noklikšķinot uz konkrēta ieraksta virsrakstu lapu skatījumā, tiek atvērts konkrēta ieraksta izvērtais skatījums.

Ierakstu dzēšanas funkcija ir pieejama pilnajā ieraksta skatījumā, noklikšķinot uz saites “dzēst”.

Tā kā komentāru pievienošanai tiek lietots komentāru modulis, tad šo datu apstrāde un ievada netiks šeit apskatīta, bet to var apskatīt pie komentāru pārvaldības moduļa (punkts 3.3.1.2.).

Apstrāde

Pievienošanas un labošanas gadījumā, tiek pārbaudīts vai dati ievadīti korekti pēc prasībām, ja ir, tad tiek veiktas izmaiņas datubāzē.

Dzēšanas operācijas izsaukšanas gadījumā, tiek lietotājam parādīts apstiprināšanas vaicājums. Apstiprinājuma saņemšanas rezultātā, jaunuma ieraksts tiek dzēsts no datubāzes, kopā ar visiem tā komentāriem.

Izvaddati

Esošie ieraksti lapu skatījumā tiek attēloti tabulas veidā, viens zem otra, dilstošā secībā kārtojot pēc pievienošanas laika.

Izvērstajā skatījumā redzams viss ieraksta teksts.

Pie visām darbībām, kuru rezultātā tiek veiktas izmaiņu darbības datubāzē, tiek parādīti paziņojumi par to norises rezultātu.

1.3.1.4. Galeriju pārvaldības modulis

Ļauj veidot video un attēlu galerijas. Lietotājs var veikt standarta operācijas ar albumiem: izveidot albumu, mainīt albuma nosaukumu, dzēst albumu ar visu to saturu.

Var pievienot attēlu un video vienumus konkrētā albumā.

Galeriju apskate iespējama divos skatos – lapu skatījumā, kur albumi tiek attēloti ar to sīktēliem un nosaukumu vai konkrēta albuma izvērstajā skatījumā, kur iespējams redzēt visus attēlus un video, kas atrodas konkrētajā albumā. Albumā esošo satura secību iespējams mainīt klikšķinot uz saitēm, kas atrodas joslā “Kārtot pēc”, tiek piedāvātas 3 veidu sakārtošanas, pēc jaunākā pievienotā, pēc komentāru skaita, kā arī pēc balsotākā (šī iespēja pieejama tikai, ja ir iespējota balsošanas iespēja) .Pēc tam klikšķinot uz konkrēta sīktēla nonāk uz konkrēta vienuma izvērsto skatījumu, kur ir iespējams aplūkot konkrēto attēlu pilnā izmērā, vai, ja tas ir video, to noskatīties.

Par ievietotajiem attēliem un video ir iespēja balsot un pievienot komentāru.

Ļauj ar “vilkt un nomest” principu mainīt konkrētā albumā esošo vienumu secību.

Ievade

Veidojot albumu tiek norādīts albuma nosaukums, kas ir līdz 200 simbolu garš.

Vienumi albumos tiek pievienoti caur rīku, kas ļauj no “draugiem.lv” sistēmas lietotāja albumiem pievienot attēlus un video iekš projekta galerijām.

Vienumus iespējams pārvietot starp albumiem, kā arī norādīt kādu elementu no albuma kā albuma sīktēlu.

Vienumus ir iespējams “apstiprināt” vai “noraidīt”, ja modulī ir iespējota šī administrēšanas funkcija.

Balsojot par konkrētu albumu lietotājs noklikšķina uz pogas “balsot” vai, ja izvēlēta vērtēšanas sistēma līdz 5 ballēm, tad novērtē, spiežot uz attiecīgā zvaigznīšu skaita.

Mainot attēlu secību, lietotājs ar “vilkt un nomest” palīdzību sakārto jauno secību.

Komentāru pievienošanas ievaddatus un apstrādi skatīt skatīt punktu 1.3.1.1.

Dzēšanas operācijas izsaukšanas gadījumā, tiek lietotājam parādīts apstiprināšanas vaicājums.

Apstrāde

Veidojot jaunu albumu tiek pārbaudīts, vai ievadītais nosaukums atbilst garumam, ja atbilst tiek izveidots ieraksts datubāzē.

Ja tiek pievienots viens vai vairāki ieraksti, tiek pārbaudīts vai iesūtītie identifikatori ir atrodami sistēmā un pievienotie vienumi pieder lietotājam, kas tos iesūta, tādējādi aizliedzot pievienot cita lietotāja albumu saturu.

Balsojot par konkrētu vienumu, tiek pārbaudīts vai lietotājs jau nav sasniedzis balsošanas limitu (sīkāk par limitiem skatīt punktu 1.3.1.5.) šai dienai par noteikto vienumu albumā, ja nav tad tiek pievienots ieraksts datubāzē un atjaunots balsu skaits.

Apstiprināšanas vai noraidīšanas gadījumā, tiek nomainīts vienuma statuss datubāzē, ja vienums tiek apstiprināts, tad albumā publiski šis vienums kļūst pieejams un par to var balsot un pievienot komentārus.

Sakārtošanas operācijas laikā, pēc katras “vilkt un nomest” operācijas tiek veikta secību maiņa visiem iesaistītajiem vienumiem datubāzē.

Ja tiek dzēsts albums, tad tas tiek dzēsts kopā ar visu tā saturu, ja tiek dzēsts konkrēts elements, tad tas tiek dzēsts un tiek atjaunots elementu skaits albumā.

Izvaddati

Albumi tiek attēloti saraksta veidā rindās un kolonās ar tā sīktēlu un nosaukumu.

Albumu saturs tiek izkārtots līdzīgā veidā kā albumi, attēlojot katra vienuma sīktēlu.

Par darbībām, kuru rezultātā tiek veiktas izmaiņas datubāzē, tiek izdrukāti darbību rezultātu statusi.

1.3.1.5. Balsošanas pārvaldības modulis

Ļauj pievienot balsojumu projektā un iegūt datus par lietotāja balsojumiem, kā arī dzēst konkrēta elementa visus balsojuma datus.

Balsojums iespējams ar vienu balsi vai konkrēta skaitļa novērtējumu.

Programmētājs var norādīt 3 veidus kā ierobežot lietotāju balsojumus – var balsot vienu reizi dienā, var balsot vienu reizi dienā par vienu elementu, vai vienu reizi visa projekta ietvaros.

Ievade

Tiek veikts pieprasījums norādot konkrētu objektu un vērtējumu - balsošanas gadījumā.

Lai iegūtu informāciju par lietotāja balsojumiem nepieciešams veikt pieprasījumu ar lietotāja identifikatoru, ir iespējams iegūt visus lietotāja balsojumus.

Lai dzēstu konkrēta elementa balsojumus, nepieciešams veikt pieprasījumu ar elementa identifikatoru.

Apstrāde

Veicot balsojuma pieprasījumu ieraksts tiek pievienots datubāzē.

Informācijas pieprasīšanas gadījumā tiek pieprasīti dati no datubāzes, ņemot vērā programmētāja uzstādītos ierobežojumus.

Dzēšanas gadījumā, tiek veiktas izmaiņas datubāzē.

Izvaddati

Veicot informācijas pieprasījumu tiek atgriezts datu masīvs ar informāciju par lietotāja balsojumiem.

Veicot dzēšanas pieprasījumu tiek atgriezts ietekmēto ierakstu skaits.

1.3.1.6. Foruma pārvaldības modulis

Ļauj pievienot foruma pavadienus un to atbildes, kā arī dzēst, gan atbildes, gan pavadienus.

Foruma apskate pieejama divos režīmos, lapu skatījumā, kur ir pieejams tikai foruma pavadieņu saraksts un izvērstais konkrēta pavadiena skatījums, kur var redzēt konkrēta pavadiena pilno tekstu un tā atbildes.

Programmētājs var norādīt: vai lapu skatījumā rādīt pavadiena ievadtekstu, pēc kā tiks kārtoti ieraksti (pēc jaunākā, pēc vecākā, vai pēc pēdējās pievienotās atbildes, vai lietotājs var pievienot jaunu pavadienam, kā arī vai lietotājs var pievienot atbildi.

Ievade

Pavadiena pievienošanas gadījumā ir jāaizpilda forma, kas satur divus laukus:

1. Virsraksts – teksta virkne, līdz 500 simboliem.
2. Teksts – teksta virkne līdz 5000 simboliem.

Abi lauki ir obligāti jāaizpilda.

Pievienojot atbildi uz kādu pavadienam, ir jāaizpilda tikai teksta lauks, kas ir līdz 2000 simboliem un ir jāaizpilda obligāti.

Noklikšķinot uz konkrēta ieraksta virsraksta lapu skatījumā, tiek atvērts konkrēta ieraksta izvērsta skatījums.

Lai dzēstu konkrētu atbildi vai pavadienam, nepieciešams noklikšķināt uz saites “[x]”, kas atrodas pie katra dzēšamā elementa.

Apstrāde

Pievienojot pavadienam vai atbildi tiek pārbaudīts vai dati ir korekti, ja ir, tiek veiktas izmaiņas datubāzē, pretējā gadījumā tiek izdrukāts kļūdas paziņojums “Lūdzu ievadiet korektus datus, kā norādīts pie ievades formas”.

Dzēšanas operācijas izsaukšanas gadījumā, tiek lietotājam parādīts apstiprināšanas vaicājums. Apstiprinājuma saņemšanas rezultātā, tiek dzēsts pats pavadienam, kā arī visas šim pavadienam pievienotās atbildes, ja tika dzēsts pavadienam, bet atbildes dzēšanas gadījumā, tiek vienkārši dzēsta atbilde.

Izvaddati

Foruma pavadienam virsraksti un atbildes tiek attēloti tabulas veidā, viens zem otra.

Pie visām darbībām, kurās tiek veiktas izmaiņas datubāzē, tiek parādīts paziņojums par to norisi.

1.3.1.7. Projekta atbalstītāju pārvaldības modulis

Ļauj pievienoties par atbalstītāju, kā arī atteikties, ja lietotājs jau ir pievienojies.

Programmētājs var norādīt atbalstītāju skaitu, cik attēlot vienā lapā.

Ievade

Lietotājs noklikšķina uz saites “pievienoties faniem” vai “vairs neesmu fans”.

Atbalstītāja dzēšanas gadījumā, lietotājs noklikšķina uz saites “[x]”, kas piekārtota pie katra lietotāja fotogrāfijas.

Apstrāde

Tiek pārbaudīts vai lietotājs ir attiecīgā projekta atbalstītājs, ja ir, tad lietotājs tiek dzēsts no atbalstītāju saraksta, pretējā gadījumā pievienots atbalstītāju sarakstam.

Atbalstītāju dzēšanas gadījumā (ja lietotājs tiek dzēsts no administratora puses), nevis pats atsakās, tiek parādīts apstiprinājuma vaicājums, ja tas tiek apstiprināts, tad lietotājs tiek dzēsts no atbalstītāju saraksta.

Izvaddati

Apskates režīmā tiek attēloti lietotāji tabulas ierakstu veidā, rindās un kolonnās.

1.3.2. Ārējās saskarnes prasības

1.3.2.1. Lietotāja saskarne

Lietotāja saskarne ir īstenota caur pārlūkprogrammu. Tā kā veidojamā sistēma ir tikai tikai daļa no lielāka projekta, paša projekta lietotāja saskarne ir atkarīga no tā, taču Draugiem CMS piedāvā saviem moduļiem standarta attēlošanu, ko iespējams programmētājam labot pēc saviem ieskatiem izmantojot CSS tehnoloģiju.

Visas formas ir veidotas vienā stilā, lai padarītu lietošanu ērtāku un vienkāršāk saprotamu, kļūdu gadījumos virs ievades formas tiek parādīti kļūdu ziņojumi. Ja datu apjoms pārsniedz programmētāja norādīto skaitu attēlojamajiem ierakstiem vienā lapā, tiek lietota lapu navigācija, lai piekļūtu tālākiem datiem

1.3.2.2. Programmatūras saskarne

Datu apmaiņa ar datubāzi notiek, izmantojot SQL vaicājumus, kas tiek nosūtīti izmantojot PHP iebūvēto MySQL datu bāzes piekļuves moduli.

1.3.2.3. Sakaru saskarne

Datu apmaiņa starp sistēmu un lietotāja datoru notiek caur tīmekļa pārlūkprogrammu, izmantojot HTTP protokolu.

1.3.3. Veiktspējas prasības

Sistēmai jāspēj ģenerēt atbildi uz HTTP pieprasījumiem ne ilgāk kā 1 sekundē vismaz 90 procentos gadījumu, pie nosacījuma, ka sistēmas datubāžu ierakstu skaits vienā tabulā nepārsniedz 1'000'000.

Lai nodrošinātu, optimāli jāizveido datu bāzes indeksi.

1.3.4. Projekta ierobežojumi

1.3.4.1. Atbilstība standartiem

Sistēmas ģenerētajam XHTML kodam jāatbilst *W3C XHTML 1.0 Transitional specifikācijai* [1].

Sistēmas dizaina CSS kodam jāatbilst vismaz *W3C CSS level 1* specifikācijai [2].

Ievērojot šos standartus tiek panākta HTML koda veiksmīga un vizuāli gandrīz vienāda attēlošana Interneta pārlūkprogrammās, kas tos ievēro.

Programmatūras prasību specifikācija izstrādāta atbilstoši standartam *LVS 68:1996* [3].

1.3.4.2. Aparatūras ierobežojumi

Sistēmas veiktspēja un darbības ātrums ir atkarīgi no servera aparatūras veiktspējas un servera noslodzes, kā arī lietotāja Interneta pieslēguma ātruma.

1.3.5. Atribūti

1.3.5.1. Drošība

Lietotāja ievadītie dati pirms ievietošanas datu bāzē vienmēr ir jāpārbauda, lai novērstu *SQL injekcijas* iespēju.

Pie lietotāja ievades visi HTML speciālie simboli ir jāaizstāj ar to attiecīgajām HTML entītijām.

1.3.5.2. Uzturamība

Katrā moduļu failā jābūt norādei, ar kādu Draugiem CMS versiju šis modulis ir savietojams, kā arī veicot jaunas versijas izstrādi, jāpārlicinās par veco moduļu saderību.

1.3.5.3. Pārnesamība

Pārvietojot sistēmu uz citu serveri, tai jādarbojas kā parasti, pēc tam, kad pārkopēti visi faili un datubāze, un projekta failos iekļaujot moduļus norādīti jaunie datu bāzes pieejas parametri.

2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

2.1. Ievads

2.1.1. Nolūks

Šis dokuments ir programmatūras projektējuma apraksts satura vadības sistēmai Draugiem CMS. Tas sastādīts, lai nodrošinātu veiksmīgu sistēmas izstrādi atbilstoši prasību specifikācijā izvirzītajām prasībām. Dokuments paredzēts satura vadības sistēmas izstrādātājiem.

2.1.2. Darbības sfēra

Satura vadības sistēma Draugiem CMS ir paredzēta projektu izstrādātājiem iekš draugiem.lv sociālā tīkla sistēmas, lai samazinātu darba apjomu, kas jāveic izstrādājot apakšprojektus.

Sistēma sastāv no divām daļām - projektu pārvaldības panelis, kas nodrošina Draugiem CMS sistēmas projektu pārvaldību un satura vadības moduļi, kas sevī ietver visbiežāk nepieciešamās funkcijas lietotājiem. Izstrādājot šo sistēmu liela uzmanība arī tika pievērsta, lai moduļi būtu viegli piemērojami un modificējami kādu īpašu projektu vajadzībām, kuriem nepietiek ar moduļu standarta pieejamo funkcionalitāti.

2.2. Dekompozīcijas apraksts

2.2.1. Moduļu dekompozīcija

Sistēma sastāv no divām galvenajām sastāvdaļām:

1. Projektu pārvaldības panelis – ļauj iegūt informāciju par Draugiem CMS sistēmas projektiem, kā arī pievienot, labot un dzēst projektus
2. Satura vadības moduļi – ļauj kontrolēt vietnes apmeklētājiem pieejamo saturu.

Sīkāks moduļu veikto funkciju apraksts atrodams prasību specifikācijā.

- Jaunumu pārvaldība – klase “News”
- Komentāru pārvaldības modulis – klase “Comments”
- Galeriju pārvaldības modulis – klase “Gallery”
- Balsošanas pārvaldības modulis – klase “Voting”
- Foruma pārvaldības modulis – klase “Forum”

- Projekta atbalstītāju pārvaldības modulis – klase “Fans”

Izmantotās funkcijas un dati, kas tiek izmantoti no globālās draugiem.lv sistēmas:

- Lietotāju pieteikšanās sistēmā – Draugiem CMS sistēmā ir pieejam lietotājam izveidotās sesijas dati.

- Dati par lietotāju tiek iegūti no draugiem.lv memcached servera, pieprasot atslēgas ar lietotāju identifikatoriem.

- Lietotāja dati (vārds, uzvārds, attēls, iesauka u.c.) tiek attēloti izmantojot divas metodes. Viena metode ir izmantojot funkciju “dr”, kurai padod iegūtos lietotāja datus no memcached servera un atkarībā no otrā parametra, kas nosaka kādu attēlošanas veidu lietot, tiek atgriezts html saturs ar lietotāja attēlošanu. Otrs pielietotais veids ir izmantojot draugiem.lv sistēmas klases “FriendBoxFactory” metodi “npFactory”, kurai arī padod lietotāju datus no memcache servera.

- Galeriju pārvaldības klasē tiek izmantota globālās sistēmas klase “drGalB”, no kuras tiek izmantotas trīs metodes:

1. “getAlbumsList” - saņemot lietotāja identifikatoru, lapas numuru un skaitu lapā, atgriež datu masīvu ar informāciju par lietotāja albumiem.

2. “getPicList” - saņemot albuma identifikatoru, lapas numuru un skaitu lapā, atgriež datu masīvu ar informāciju par objektiem, kas atrodas konkrētajā albumā.

3. “getItemsMc” - saņemot masīvu ar objektu identifikatoriem, atgriež informāciju par attiecīgajiem video un attēliem.

- Pie tekstu izvades tiek lietota funkcija “wWrap”, kas saņemot tekstu virni un simbolu skaitu, tekstā ievienot ik pēc uzdotā simbolu skaita ievieto specifiskus HTML simbolus, lai novērstu teksta vizuālo attēlošanu ārpus tā ietvara HTML elementa.

- Lapu navigācijai tiek lietota funkcija navigation.

2.2.2. Datu dekompozīcija

Sistēma datus glabā vienā MySQL datubāzē. Datubāze sastāv no 11 tabulām:

1. Tabula “project” - projektu saraksts,
2. Tabula “project_fans” - atbalstītāji,
3. Tabula “project_comments” - komentāri,
4. Tabula “project_comments_stats” - komentāru uzskaitē,
5. Tabula “project_gallery_pics” - galeriju attēli/video,
6. Tabula “project_gallery_alb” - galeriju albumi,
7. Tabula “project_news” - jaunumi,

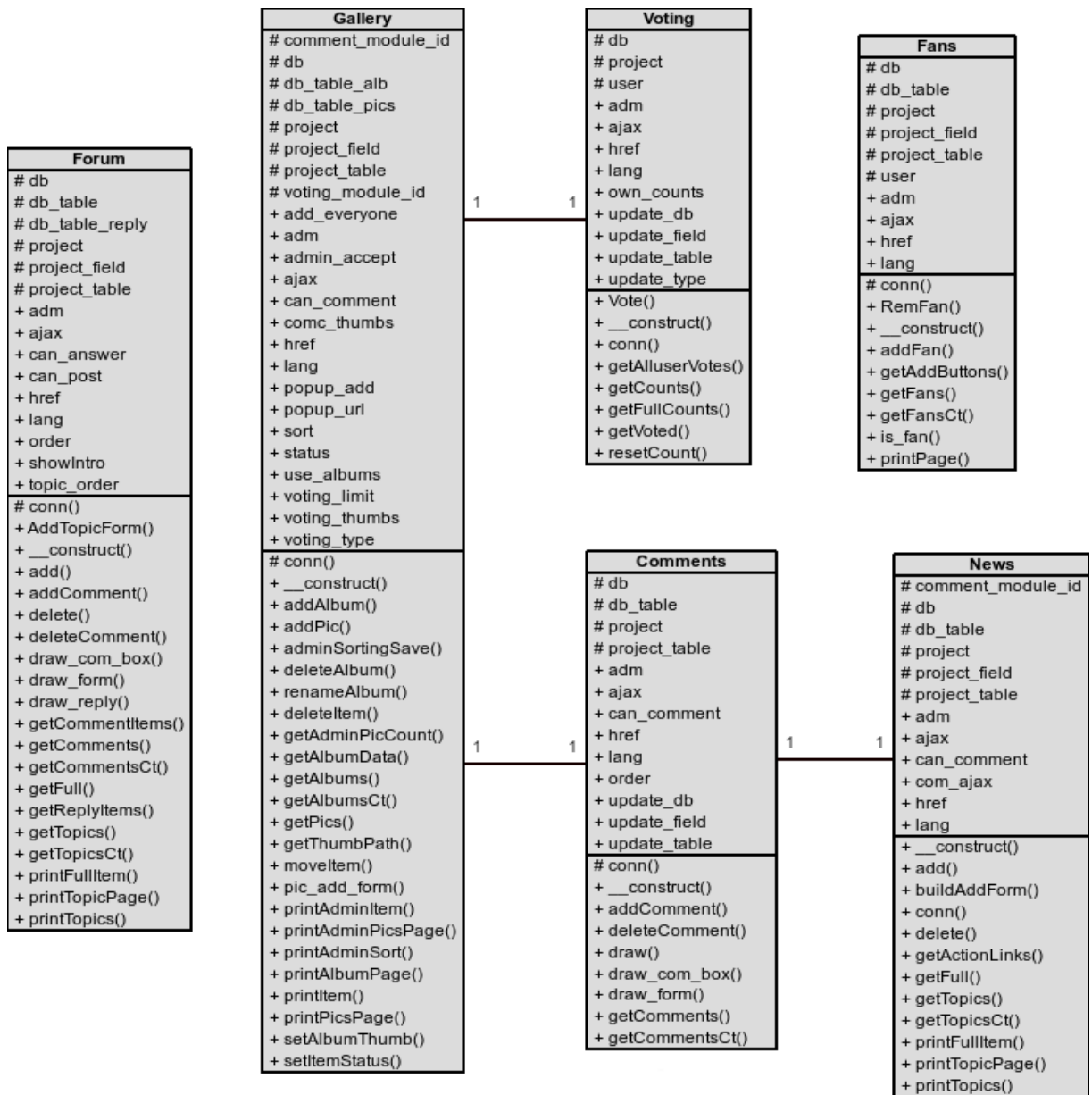
8. Tabula “project_voting” - balsojumu balsis,
9. Tabula “project_voting_stats” - balsu uzskaite,
10. Tabula “project_forum” - foruma pavadieni,
11. Tabula “project_forum_reply” - foruma pavadienu atbildes.

2.3. Atkarību apraksts

2.3.1. Starpmoduļu atkarības

Starpmoduļu atkarības attēlotas klašu diagrammas veidā (attēls 4.1).

Attēls 2.1: “Klašu diagramma”

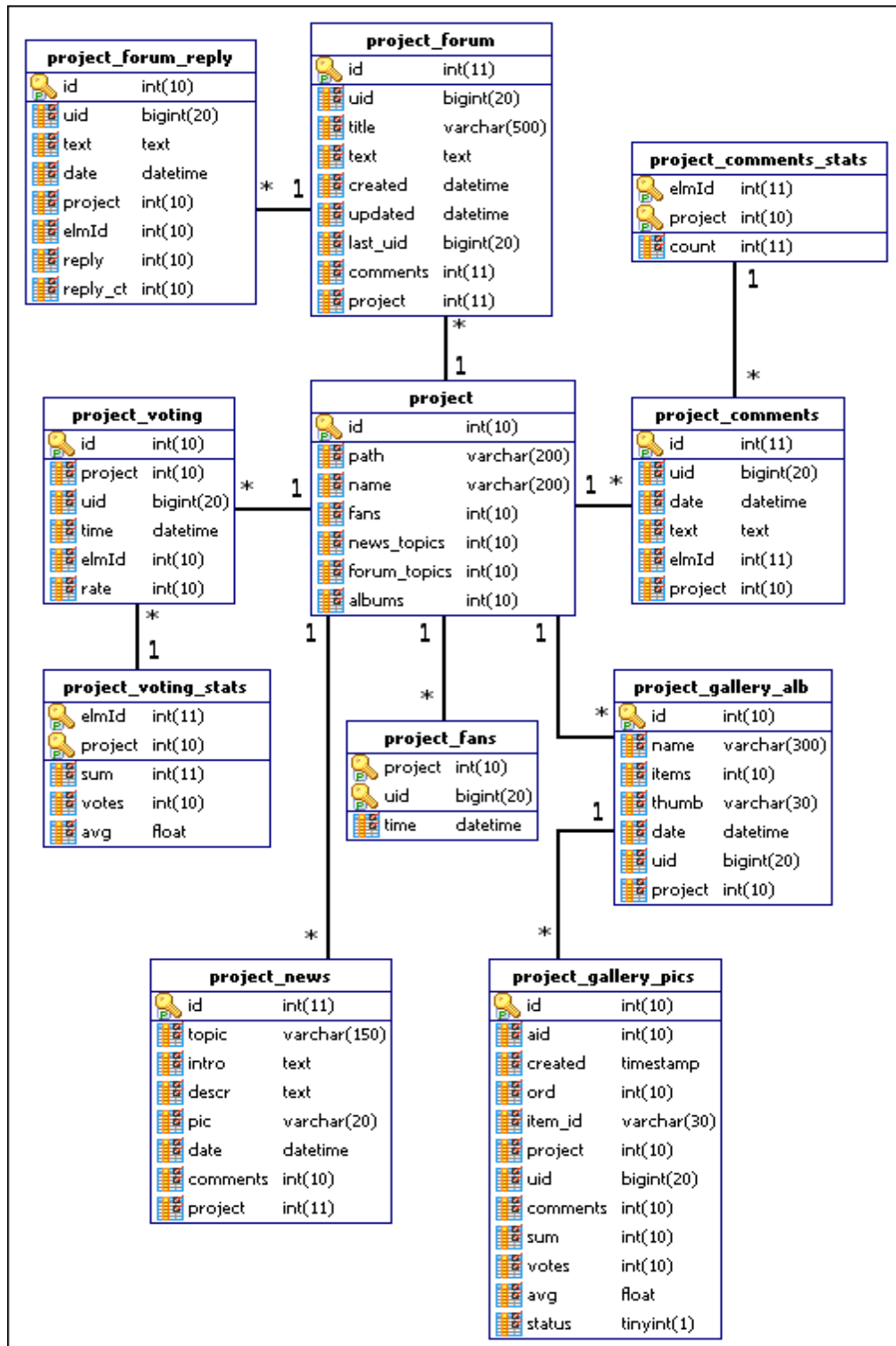


“#” - redzamība “protected”, “+” - redzamība public.

2.3.2. Datu atkarības

Datu atkarības attēlotas ER modelī (skatīt attēlu 4.2).

Attēls 2.2: "Datubāzes ER modelis"



2.4. Detalizēts projektējums

2.4.1. Moduļu detalizēts projektējums

Projektējuma aprakstā ir iekļauti paredzēto klašu un to atribūtu un metožu skaidrojumi. Sīkāks aprakstīto metožu parametru un to atgriežamo vērtību skaidrojums tiks iekļauts programmas kodā PHPdoc komentāru formā. No šiem PHPdoc komentāriem tālāk izmantojot *phpDocumentor* [6] programmatūru tiks uzģenerēta API dokumentācija, kas tālāk kalpos kā ceļvedis programmētājam sistēmas izmantošanā izstrādei. Šajā API dokumentācijā tiks iekļauts precīzs metožu parametru un atgriežamo vērtību apraksts un to datu tipi, kā arī pašu metožu īss apraksts. Tālākā izstrādes gaitā, mainoties metožu atgriežamajām vērtībām vai parametriem, tiks pārģenerēta API dokumentācija, lai tās saturošā informācija nezaudētu aktualitāti.

Atribūti, kas ir vienādi visām klasēm izņemot Balsošanas pārvaldības klasi “Voting”.

1. *\$project* (redzamība: protected) – skaitlis, projekta identifikators,
2. *\$db* (redzamība: protected) – datubāzes savienojuma resurss,
3. *\$href* (redzamība: public) – teksta virkne, kas satur ceļu uz sadaļu, kurā tiek izmantota attiecīgā klase,
4. *\$ajax* (redzamība: public) – teksta virkne, kas satur ceļu uz PHP failu, uz kuru tiks veikti AJAX pieprasījumi,
5. *\$adm* (redzamība: public) – boolean tipa vērtība, kas norāda vai ir iespējotas administratora iespējas,
6. *\$lang* (redzamība: public) – masīvs, kas satur valodas teksta virknes.

Visas šīs vērtības uzstāda attiecīgās klases konstruktors. Klašu konstruktori arī netiks sīkāk aprakstīti pie katras klases, jo pamatā visām ir vienādi – inicializē nepieciešamās vērtības, ko saņem parametros, kas vēlāk būs nepieciešamas darbā ar klases objektu.

Visās klasēs tiek izmantota metode “conn” (redzamība: protected), kas atgriež savienojuma resursu uz datubāzi, tāpēc tā netiks atsevišķi apskatīta pie katras klases, visas pārējās klašu metodes ir ar redzamību public.

2.4.1.1. Jaunumu pārvaldības klase “News”

Klase izmanto tabulas “project_news” un “project”.

Klases atribūti

\$db_table (redzamība: protected) - teksta virkne, kas satur tabulas nosaukumu, kurā glabājas jaunumi.

\$project_table (redzamība: protected) - teksta virkne, kas satur tabulas nosaukumu, kurā glabājas informācija par projektu (tabula tiek izmantota, lai tajā atjaunotu kopējo jaunumu skaitu).

\$project_field (redzamība: protected) - teksta virkne, kas satur lauka nosaukumu tabulā, kurā atjaunot kopējo jaunumu skaitu.

\$comment_module_id (redzamība: protected) – skaitlis, kas satur komentāru moduļa identifikatoru, kurš tiks lietots komentāros.

\$com_ajax (redzamība: protected) – teksta virkne, kas satur ceļu uz PHP failu, uz kuru tiks veikti komentāru AJAX pieprasījumi.

\$scan_comment (redzamība: protected) – boolean tipa vērtība, kas norāda vai ir iespēja komentēt jaunumu ierakstus.

Klases metodes

add – pievieno vai atjauno konkrētu jaunuma ierakstu datubāzē.

delete – dzēš konkrētu jaunuma ierakstu no datubāzes.

getFull – atgriež pilnu informāciju par vienu vai vairākiem jaunumu ierakstiem.

getTopics – atgriež vienu jaunumu ierakstu lapu īsajam skatījumam.

add – pievieno jaunumu datubāzei.

getTopicsCt – atgriež kopējo jaunumu skaitu.

printTopics – saņemot ierakstu datus, atgriež HTML kodu, kas satur konkrētas lapas jaunumu sarakstu.

printFullItem – atgriež HTML kodu, kas satur konkrēta ieraksta izvērsto skatījumu.

printTopicPage – saņemot lapas numuru un ierakstu skaitu lapā, atgriež HTML kodu, kas satur jaunumu sarakstu.

buildAddForm – atgriež HTML kodu ar pievienošanas vai labošanas formu konkrētam ierakstam, kā arī izsauc, pievienošanas vai labošanas darbības, ja notikusi formas saglabšana.

getActionLinks – atgriež HTML kodu, kas satur saites ar darbībām ar ierakstiem.

2.4.1.2. Komentāru pārvaldības klase “Comments”

Klase izmanto tabulas “project_comments” un “project_comments_stats”.

Klases atribūti

\$update_db (redzamība: public) – resurss, kas satur datubāzes pieslēgumu, kurā atrodas tabula, kurā jāveic komentāru skaita izmaiņas.

\$update_table (redzamība: public) – teksta virkne, kas satur tabulas nosaukumu, kurā glabājas informācija par objektiem, kuriem tiks pievienoti komentāri (tabula tiek izmantota, lai tajā atjaunotu komentāru skaitu).

\$update_field (redzamība: public) – teksta virkne, kas satur lauka nosaukumu tabulā, kurā atjaunot komentāru skaitu.

\$scan_comment (redzamība: public) – boolean tipa vērtība, kas nosaka vai ir iespējams pievienot komentāru.

\$order (redzamība: public) – teksta virkne, kas nosaka komentāru izkārtojumu.

\$db_table (redzamība: protected) – teksta virkne, kas satur tabulas nosaukumu, kurā glabājas komentāri.

\$project_table (redzamība: protected) – teksta virkne, satur tabulas nosaukumu, kurā glabājas komentāru statistika.

Klases metodes

getComments – saņemot lapas numuru un ierakstu skaitu lapā atgriež masīvu ar komentāru datiem.

getCommentsCt – atgriež kopējo komentāru skaitu.

addComment – pievieno komentāru.

deleteComment – dzēš komentāru.

draw_form – atgriež HTML kodu, kas satur komentāra pievienošanas formu.

draw_com_box – saņemot komentāru datus, atgriež HTML kodu ar komentāru attēlošanu.

draw – saņemot lapas numuru un ierakstu skaitu lapā atgriež HTML kodu ar komentāru attēlošanu.

2.4.1.3. Balsošanas pārvaldības klase “Voting”

Klase izmanto tabulas “project_voting_stats” un “project_voting”.

Klases atribūti

\$own_counts (redzamība: public) – boolean tipa vērtība, kas nosaka ir vai nav nepieciešams uzskaitīt balsu skaitus pie objektem sistēmas iekšienē.

\$update_db (redzamība: public) – resurss, kas satur datubāzes pieslēgumu, kurā atrodas tabula, kurā jāveic balsošanas skaita izmaiņas.

\$update_table (redzamība: public) – teksta virkne, kas satur tabulas nosaukumu, kurā glabājas informācija par objektiem, kuriem tiks mainīts balsu skaits.

\$update_field (redzamība: public) – teksta virkne, kas satur lauka nosaukumu tabulā, kurā atjaunot balsu skaitu.

\$update_type (redzamība: public) – skaitlis, kas norāda balsošanas tipu.

\$user (redzamība: protected) – lietotāja identifikators, ko izmantot klases funkcijām.

Klases metodes

getCounts – saņemot objektu identifikatorus, tiek atgriezta informācija par balsu skaitu, kas šiem objektu identifikatoriem ir piesaistīts.

getFullCounts – atgriež masīvu ar visiem objektiem un tiem piekārtotajiem balsojumiem.

getAllUserVotes – atgriež visus konkrētā lietotāja balsojumus.

getVoted – atgriež boolean tipa vērtību vai lietotājam pašlaik ir pieejama balsošanas iespēja.

Vote – pievieno balsojumu par konkrētu objektu.

resetCount – ļauj atiestatīt balsojumu skaitu konkrētiem objektiem, vai visiem projekta objektiem.

2.4.1.4. Atbalstītāju pārvaldības klase “Fans”

Klase izmanto tabulas “project_fans” un “project”.

Klases atribūti

\$user (redzamība: protected) – lietotāja identifikators, ko izmantot klases funkcijām.

\$update_db (redzamība: public) – teksta virkne, kas satur tabulas nosaukumu, kurā glabāties atbalstītāji.

\$update_table (redzamība: public) – teksta virkne, kas satur tabulas nosaukumu, kurā glabājas informācija par objektiem, kuriem tiks mainīts atbalstītāju skaits.

\$update_field (redzamība: public) – teksta virkne, kas satur lauka nosaukumu tabulā, kurā atjaunot atbalstītāju skaitu.

Klases metodes

getFans – saņemot lapas numuru un ierakstu skaitu lapā, atgriež masīvu, kas satur masīvu ar lietotāju identifikatoriem.

getFansCt – atgriež kopējo fanu skaitu.

is_fan – atgriež boolean vērtību ir vai nav lietotājs ir pieteicies par atbalstītāju konkrētajam projektam.

getAddButtons – atgriež HTML kodu ar pieteikšanās vai atteikšanās saitēm.

addFan – pievieno lietotāju par atbalstītāju projektā.

RemFan – izdzēš lietotāju no atbalstītāju saraksta.

printPage – saņemot lapas numuru un ierakstu skaitu lapā, atgriež HTML kodu, kas satur lietotāju attēlus.

2.4.1.5. Forumu pārvaldības klase “Forum”

Klase izmanto tabulas “project_forum”, “project_forum_reply” un “project”

Klases atribūti

\$showIntro (redzamība: public) – boolean tipa vērtība, kas nosaka vai tematu skatījumā nepieciešams rādīt ievadtekstu.

\$order (redzamība: public) – nosaka tematu atbilžu secību.

\$topic_order (redzamība: public) – nosaka tematu secību.

\$scan_answer (redzamība: public) – boolean tipa vērtība, kas nosaka vai tiek atļauts pievienot atbildes uz tematiem.

\$scan_post (redzamība: public) – boolean tipa vērtība, kas nosaka vai tiek atļauts pievienot jaunus tematus.

\$db_table (redzamība: protected) – teksta virkne, kas satur tabulas nosaukumu, kurā glabājas foruma temati.

\$db_table_reply (redzamība: protected) – teksta virkne, kas satur tabulas nosaukumu, kurā glabājas foruma tematu atbildes.

\$project_table (redzamība: protected) - teksta virkne, kas satur tabulas nosaukumu, kurā glabājas informācija par projektu (tabula tiek izmantota, lai tajā atjaunotu kopējo tematu skaitu).

\$project_field (redzamība: protected) - teksta virkne, kas satur lauka nosaukumu tabulā, kurā atjaunot kopējo tematu skaitu.

Klases metodes

add – pievieno forumam jaunu tematu.

delete – dzēš konkrētu foruma tematu.

getFull – saņemot ierakstu identifikatorus, atgriež informāciju par vienu vai vairākiem ierakstiem.

getTopics – saņemot lapas numuru un ierakstu skaitu lapā, atgriež informāciju par foruma tematiem.

getTopicsCt – atgriež foruma kopējo foruma tematu skaitu

printTopics – saņemot datus par foruma tematiem, atgriež HTML kodu, kas satur ierakstu attēlošanu.

printFullItem – atgriež HTML kodu, kas satur konkrēta temata izvērsto skatījumu.

printTopicPage – saņemot lapas numuru un ierakstu skaitu lapā, atgriež HTML kodu, kas satur tematu sarakstu.

AddTopicForm – atgriež HTML kodu, kas satur temata izveidošanas formu.

draw_form – atgriež HTML kodu, kas satur atbildes pievienošanas formu.

draw_com_box – saņemot atbilžu datus, atgriež HTML kodu, kas satur temata atbildes attēlojumu.

draw_reply – saņemot lapas numuru, ierakstu skaitu lapā un temata identifikatoru, atgriež HTML kodu, kas satur atbildes uz konkrētu tematu.

addComment – pievieno atbildi konkrētam tematam.

deleteComment – dzēst atbildi.

getCommentItems – saņemot atbilžu identifikatorus, atgriež datus par atbildēm.

getComments – saņemot lapas numuru, ierakstu skaitu lapā un temata identifikatoru, atgriež datus par atbildēm.

getCommentsCt – saņemot temata identifikatoru, atgriež tā kopējo atbilžu skaitu.

2.4.1.5. Galeriju pārvaldības klase “Gallery”

Klase izmanto tabulas “project_gallery_pics”, “project_gallery_alb” un “project”.

Klases atribūti

\$popup_add (redzamība: public) – boolean tipa vērtība, kas nosaka, vai tiek lietota izlecošā loga satura pievienošanas iespēja.

\$popup_url (redzamība: public) – teksta virkne, kas satur adresi uz PHP failu, kurš tiek izsaukt izlecošā loga izsaukšanai.

\$scan_comment (redzamība: public) – boolean tipa vērtība, kas nosaka vai ir iespējams komentēt galeriju saturu.

\$comc_thumbs (redzamība: public) – boolean tipa vērtība, kas nosaka vai sīktēlu skatīšanās režīmā pie attēliem tiek rādīts komentāru skats.

\$voting_thumbs (redzamība: public) – boolean tipa vērtība, kas nosaka vai sīktēlu skatīšanās režīmā pie attēliem tiek rādīts balsu skats.

\$status (redzamība: public) – skaitliska vērtība, kas nosaka ar kādu statusu tiks attēlots albumu saturs (lieto iespējai administratoram apstiprināt attēlu/video konkursa gadījumā).

\$sort (redzamība: public) – skaitliska vērtība, kas nosaka kā tiks izkārtoti sīktēli.

\$voting_type (redzamība: public) – skaitliska vērtība, kas nosaka kāda tipa balsojums par bildēm tiek lietots (-1 – netiek lietots balsojums, 1 – balsojums ar vienu balli, 2 – balsojums ar novērtēšanu līdz 5 ballēm).

\$voting_limit (redzamība: public) – nosaka kādi ir balsošanas ierobežojumi, ko lietot sadarbībai ar balsošanas klasi. (bar balsot vienu reizi dienā, vienu reizi projekta laikā u.c.)

\$use_albums (redzamība: public) – boolean tipa vērtība, kas nosaka vai tiks lietoti albumu attēlu atrādīšanai.

\$add_everyone (redzamība: public) – boolean tipa vērtība, kas nosaka vai ikvienam ir iespēja pievienot saturu galerijām.

\$admin_accept (redzamība: public) – boolean tipa vērtība, kas nosaka vai pievienotie attēli ir jāapstiprina administratoram.

\$db_table_pics (redzamība: protected) – teksta virkne, kas satur tabulas nosaukumu, kurā glabājas pievienotie attēli/video.

\$db_table_alb (redzamība: protected) – teksta virkne, kas satur tabulas nosaukumu, kurā glabājas pievienotie albumi.

\$project_table (redzamība: protected) - teksta virkne, kas satur tabulas nosaukumu, kurā glabājas informācija par projektu (tabula tiek izmantota, lai tajā atjaunotu kopējo albumu skaitu).

\$project_field (redzamība: protected) - teksta virkne, kas satur lauka nosaukumu tabulā, kurā atjaunot kopējo albumu skaitu.

\$comment_module_id (redzamība: protected) – skaitlisks identifikators, kuru izmantot iekļautajam komentāru modulim.

\$voting_module_id (redzamība: protected) – skaitlisks identifikators, kuru izmantot iekļautajam balsošanas modulim.

Klases metodes

pic_add_form – atgriež HTML kodu, kas satur attēlu/video pievienošanas formu.

printAlbumPage – saņemot lapas numuru un ierakstu skaitu lapā, atgriež HTML kodu, kas satur albumu sarakstu.

printItem – saņemot galeriju vienuma identifikatoru, atgriež HTML kodu ar konkrēta video/attēla attēlošanu.

printPicsPage – atgriež HTML kodu ar konkrēta albuma, konkrētas lapas bilžu sīktēlu sarakstu.

printAdminSort – atgriež HTML kodu ar bilžu/video sarakstu, secību maiņai.

printAdminItem – saņemot galeriju vienuma identifikatoru, atgriež HTML kodu ar konkrēta video/attēla attēlošanu administrēšanas režīmā.

printAdminPicsPage – atgriež HTML kodu ar konkrēta albuma, konkrētas lapas bilžu sīktēlu sarakstu administrēšanas režīmā.

getAdminPicCount – atgriež konkrētā albumā esošo neapstiprināto vienumu skaitu.

getThumbPath – saņemot draugiem.lv sistēmas attēlu datus, atgriež katram attēlam/video atbilstošo ceļu uz sīktēlu.

getAlbumData – saņemot albumu identifikatorus, atgriež datus par albumiem

getPics – saņemot lapas numuru, ierakstu skaitu lapā un albuma identifikatoru, atgriež datus par vienumiem.

getAlbums – saņemot lapas numuru un ierakstu skaitu lapā, atgriež datus par albumiem.

getAlbumsCt – atgriež kopējo albumu skaitu.

addPic – pievieno video/attēlu albumam.

addAlbum – pievieno jaunu albumu.

deleteAlbum – dzēš albumu.

renameAlbum – nomaina albuma noaukumu.

deleteItem – dzēš elementu no albuma.

moveItem – pārvieno konkrētu elementu uz citu albumu.

setItemStatus – nomaina konkrēta elementa statusu.

setAlbumThumb – uzstāda albuma sīktēlu.

adminSortingSave – saglabā jaunizveidoto elementu secību albumā.

2.4.2. Datu detalizēts projektējums

2.4.2.1. Tabula “project”

Tabulā glabājas informācija par visi projekti, kuri tiek izmantoti iekš sistēmas.

Primārā atslēga - “id”.

Tabula 2.1: Tabula “project”

Lauks	Tips	Atslēgas	Komentārs
<u>id</u>	int(10)	PK	Automātiski ģenerēts, unikāls projekta identifikators
path	varchar(200)		Relatīvais ceļš, kur projekts ir atrodas.

Lauks	Tips	Atslēgas	Komentārs
name	varchar(200)		Projekta nosaukums
fans	int(10)		Projekta fanu skaits
news_topics	int(10)		Projekta jaunumu skaits
forum_topics	int(10)		Projekta foruma pavedienu skaits
albums	int(10)		Projekta albumu skaits

2.4.2.2. Tabula “project_comments”

Tabulā glabājas informācija par komentāriem.

Primārā atslēga - id. Kombinētais indekss uz trīs laukiem – elmId, project un date.

Tabula 2.2: Tabula “project_comments”

Lauks	Tips	Atslēgas	Komentārs
<u>id</u>	int(10)	PK	Automātiski ģenerēts, unikāls komentāra identifikators
uid	bigint(20)		Lietotāja identifikators
date	datetime	Indekss	Pievienošanas datums
text	text		Komentāra teksts
elmId	int(10)	Indekss	Identifikators objektam, kuram pievieno komentāru.
project	int(10)	Indekss	Projekta identifikators

2.4.2.3. Tabula “project_comments_stats”

Tabulā glabājas informācija par komentu skaitu.

Primārā atslēga uz elmId un project.

Tabula 2.3: Tabula “project_comments_stats”

Lauks	Tips	Atslēgas	Komentārs
<u>elmId</u>	int(10)	PK	Komentāru objekta identifikators
<u>project</u>	int(10)	PK	Projekta identifikators
count	int(10)		Komentāru skaits

2.4.2.4. Tabula “project_fans”

Tabulā glabājas informācija par atbalstītājiem.

Primārā atslēga uz project un uid.

Tabula 2.4: Tabula “project_fans”

Lauks	Tips	Atslēgas	Komentārs
<u>project</u>	int(10)	PK	Projekta identifikators
<u>uid</u>	bigint(20)	PK	Lietotāja identifikators
time	datetime		Pievienošanās laiks

2.4.2.5. Tabula “project_forum”

Tabulā glabājas informācija par foruma pavedieniem.

Primārā atslēga - id. Kombinētais indekss uz project un updated.

Tabula 2.5: Tabula “project_forum”

Lauks	Tips	Atslēgas	Komentārs
<u>id</u>	int(10)	PK	Automātiski ģenerēts, unikāls pavediena identifikators
uid	bigint(20)		Lietotāja identifikators
title	varchar(500)		Pavediena virsraksts
text	text		Pavediena teksts
created	datetime		Pievienošanas laiks
updated	datetime	Indekss	Pēdējās pievienotās atbildes laiks
last_uid	bigint(20)		Pēdējās pievienotās atbildes lietotāja identifikators
comments	int(10)		Atbilžu skaits
project	int(10)	Indekss	Projekta identifikators

2.4.2.6. Tabula “project_forum_reply”

Tabulā glabājas informācija par foruma pavedienu atbildēm.

Primārā atslēga - id. Kombinētais indekss uz project, elmId un reply un atsevišķs indekss uz date.

Tabula 2.6: Tabula “project_forum_reply”

Lauks	Tips	Atslēgas	Komentārs
<u>id</u>	int(10)	PK	Automātiski ģenerēts, unikāls projekta identifikators
uid	bigint(20)		Lietotāja identifikators
text	text		Atbildes teksts

Lauks	Tips	Atslēgas	Komentārs
date	datetime	Indekss	Pievienošanas laiks
project	int(10)	Indekss	Projekta identifikators
elmId	int(10)	Indekss	Pavediena identifikators

2.4.2.7. Tabula “project_gallery_alb”

Tabulā glabājas informācija par galeriju albumiem.

Primārā atslēga - id. Kombinētais indekss uz project un date.

Tabula 2.7: Tabula “project_gallery_alb”

Lauks	Tips	Atslēgas	Komentārs
<u>id</u>	int(10)	PK	Automātiski ģenerēts, unikāls albuma identifikators
name	varchar(300)		Albuma nosaukums
items	int(10)		Vienumu (bilžu un video) skaits, kas atrodas albuma
thumb	varchar(30)		Sīktēla identifikators
date	datetime	Indekss	Izveidošanas laiks
uid	bigint(20)		Lietotāja identifikators
project	int(10)	Indekss	Projekta identifikators

2.4.2.8. Tabula “project_gallery_pics”

Tabulā glabājas informācija par galeriju attēliem un video.

Primārā atslēga - id. Kombinētais indekss uz project, aid, ord, created un status.

Tabula 2.8: Tabula “project_gallery_pics”

Lauks	Tips	Atslēgas	Komentārs
<u>id</u>	int(10)	PK	Automātiski ģenerēts, unikāls attēla identifikators
aid	int(10)	Indekss	Albuma identifikators
created	datetime	Indekss	Izveidošanas laiks
ord	int(10)	Indekss	Kārtas numurs albumā
item_id	varchar(30)		Attēla identifikators draugiem.lv sistēmā.
project	int(10)	Indekss	Projekta identifikators
uid	bigint(20)		Lietotāja identifikators
comments	int(10)		Komentāru skaits

Lauks	Tips	Atslēgas	Komentārs
sum	int(10)		Balsu summa
votes	int(10)		Balsu skaits
avg	float		Vidējais vērtējums
status	tinyint(1)	Indekss	Statuss (0 – neapstiprināta, 1 – apstiprināta, 2 - noraidīta)

2.4.2.9. Tabula “project_news”

Tabulā glabājas informācija par jaunumiem.

Primārā atslēga - id. Kombinētais indekss uz project un date.

Tabula 2.9: Tabula “project_news”

Lauks	Tips	Atslēgas	Komentārs
<u>id</u>	int(10)	PK	Automātiski ģenerēts, unikāls jaunuma identifikators
topic	varchar(150)		Jaunuma virsraksts
intro	text		Jaunuma ievads
descr	text		Jaunuma teksts
pic	varchar(20)		Pievienotā attēla identifikators draugiem.lv sistēmā.
date	datetime	Indekss	Pievienošanas laiks
comments	int(10)		Komentāru skaits
project	int(11)	Indekss	Projekta identifikators

2.4.2.10. Tabula “project_voting”

Tabulā glabājas informācija par galeriju attēliem un video.

Primārā atslēga - id. Kombinētais indekss uz project, time un uid.

Tabula 2.10: Tabula “project_voting”

Lauks	Tips	Atslēgas	Komentārs
<u>id</u>	int(10)	PK	Automātiski ģenerēts, unikāls jaunuma identifikators
project	int(10)	Indekss	Projekta identifikators
uid	bigint(20)		Lietotāja identifikators
time	datetime	Indekss	Pievienošanas laiks
elmId	int(10)	Indekss	Identifikators objektam, par kuru balso.

Lauks	Tips	Atslēgas	Komentārs
rate	int(10)		Balsošanas vērtība.

2.4.2.11. Tabula “project_voting_stats”

Tabulā glabājas balsojumu statistika.

Primārā atslēga – elmId un project.

Tabula 2.11: Tabula “project_voting_stats”

Lauks	Tips	Atslēgas	Komentārs
<u>elmId</u>	int(10)	PK	Balsošanas objekta identifikators
<u>project</u>	int(10)	PK	Projekta identifikators
sum	int(10)		Balsu summa
votes	int(10)		Balsu skaits
avg	float		Vidējais vērtējums

3. TESTĒŠANAS DOKUMENTĀCIJA

3.1. Testēšanas plāns

Vienībtestēšanas procesā jāveic visu satura vadības sistēmas moduļu un projektu pārvaldības paneļa funkcionalitātes pārbaude. Testēšanā jāpārbauda visas prasību specifikācijā norādītās prasības. Testēšanas laikā jāveic sekojošas pārbaudes:

- Vai veiktās darbības ar datiem (pievienošana, labošana, dzēšana) izpildās kā plānots (pārbaude jāveic datubāzē), šeit notiks arī pārbaude ņemot vērā moduļu atkarības, piemēram, ja komentāru moduli izmanto galeriju klase, tad tas jātestē arī galeriju klases ietvaros,
- Vai tiek parādīti statusu paziņojumi lietotājiem par operāciju izpildi,
- Vai tiek izpildīti nosacījumi funkciju izvadei pēc prasību specifikācijas (piemēram: vai funkcionē iespēja pārslēgt attēlošanas skatus jaunumu modulim u.c.).

Testēšanas procesā netiks veikti testi uz statusu paziņojumiem, ja tie nāk no moduļa, kas ir iekļauts kādā modulī (piemēram, ja komentāru modulis ir iekļauts jaunumu modulī, tad jaunumu moduļa ietvaros statusu paziņojumu pārbaude netiks veikta, tiks veikta tikai funkcionalitātes pārbaude, paziņojumu testi notiks viena moduļa ietvaros).

Testēšanu jāveic izmantojot melnās kastes principu, lietojot tīmekļa pārlūkprogrammu, no lietotāja viedokļa, neaplūkojot PHP kodu.

3.2. Testēšanas žurnāls

3.2.1. Projektu pārvaldības panelis

Tabula 3.1: Projektu pārvaldības paneļa testēšanas rezultāti

Testa nr	Testa apraksts	Rezultāts	Komentārs
1.	Attēlošana - esošie projekti tiek attēloti dilstošā secībā, tabulas veidā.	Izpildās	
2.	Pievienošana/labošana – obligāti jāaizpilda lauks “Nosaukums”.	Izpildās	
3.	Pievienošana/labošana – obligāti jāaizpilda lauks “Ceļš uz projekta atrašanās vietu”.	Izpildās	
4.	Pievienošana – ieraksts tiek pievienots datubāzē.	Izpildās	

Testa nr	Testa apraksts	Rezultāts	Komentārs
5.	Labošana – tiek aizpildītas lauka vērtības ar vecajām vērtībām	Izpildās	
6.	Labošana – datubāzes ieraksts tiek atjaunots.	Izpildās	
7.	Izmaiņu (labošana, dzēšana, pievienošana) veikšana - tiek parādīti statusu paziņojumi.	Neizpildās	Netika izdrukāts kļūdas paziņojums pie ieraksta pievienošanas un labošanas, kļūda tika novērsta.
8.	Dzēšana – tiek parādīts dzēšanas apstiprināšanas vaicājums.	Neizpildās	Netika parādīts vaicājums, JavaScript koda kļūda tika novērsta.
9.	Dzēšana – tiek dzēsts projekts kopā ar visiem tā moduļu datiem	Izpildās	
10.	Dzēšana – tiek dzēsti projekta konkrēta moduļa, konkrēti dati.	Izpildās	

3.2.2. Komentāru pārvaldības modulis

Tabula 3.2: *Komentāru moduļa testēšanas rezultāti*

Testa nr	Testa apraksts	Rezultāts	Komentārs
11.	Attēlošana – komentāri tiek attēloti tabulas veidā, viens zem otra.	Izpildās	
12.	Pievienošana – tiek parādīts kļūdas paziņojums, ja netiek aizpildīts teksta lauks.	Izpildās	
13.	Pievienošana – ieraksts tiek pievienots datubāzē..	Izpildās	
14.	Dzēšana – tiek parādīts dzēšanas apstiprināšanas vaicājums.	Izpildās	
15.	Dzēšana – ieraksts tiek dzēsts no datubāzes.	Izpildās	

3.2.3. Jaunumu pārvaldības modulis

Tabula 3.3: Jaunumu moduļa testēšanas rezultāti

Testa nr	Testa apraksts	Rezultāts	Komentārs
16.	Pievienošana/labošana – obligāti jāaizpilda lauks “Virsraksts”.	Izpildās	
17.	Pievienošana/labošana – obligāti jāaizpilda lauks “Ievads”.	Izpildās	
18.	Pievienošana/labošana – obligāti jāaizpilda lauks “Teksts”.	Izpildās	
19.	Pievienošana/labošana – Ir iespējams pievienot ierakstam attēlu un tas ir redzams pēc pievienošanas pie ieraksta..	Izpildās	
20.	Ierakstu labošanas, dzēšanas un pievienošanas rezultātā tiek parādīti statusu paziņojumi.	Izpildās	
21.	Attēlošana – Lapu skatījumā ieraksti izkārtojas tabulas veidā dilstošā secībā pēc pievienošanas laika.	Izpildās	
22.	Attēlošana – Iespējams .pārslēgties no lapu skatījuma uz ieraksta izvērsto skatījumu un atpakaļ.	Izpildās	
23.	Dzēšana – tiek parādīts dzēšanas apstiprināšanas vaicājums.	Izpildās	
24.	Dzēšana – jaunumu ieraksts tiek dzēsts no datubāzes, kopā ar visiem komentāriem.	Izpildās	
25.	Komentāru pievienošana – var pievienot komentāru.	Izpildās	
26.	Komentāru dzēšana – var dzēst komentāru.	Izpildās	

3.2.4. Balsošanas pārvaldības modulis

Tabula 3.4: Balsošanas pārvaldības moduļa testēšanas rezultāti

Testa nr	Testa apraksts	Rezultāts	Komentārs
27.	Balsošana – pievieno balsi, saņemot parametrā vērtējumu un elementa identifikatoru	Izpildās	
28.	Balsošana informācija – pievieno balsi, nododot parametrā vērtējumu un objekta identifikatoru	Izpildās	

Testa nr	Testa apraksts	Rezultāts	Komentārs
29.	Balsošanu dzēšana – dzēš balsis no datubāzes, saņemot parametrā elementa identifikatoru.	Izpildās	

3.2.5. Projekta atbalstītāju pārvaldības modulis

Tabula 3.5: Projekta atbalstītāju pārvaldības moduļa testēšanas rezultāti

Testa nr	Testa apraksts	Rezultāts	Komentārs
30.	Pievienoties/atteikties no atbalstītāju saraksta..	Izpildās	
31.	Dzēšana – tiek parādīts dzēšanas apstiprināšanas vaicājums.	Izpildās	
32.	Dzēšana – lietotās tiek dzēsts no projekta atbalstītāju datubāzes.	Izpildās	
33.	Attēlošana - lietotāji tiek attēloti tabulas ierakstu veidā, rindās un kolonnās.	Izpildās	

3.2.6. Foruma pārvaldības modulis

Tabula 3.6: Foruma pārvaldības moduļa testēšanas rezultāti

Testa nr	Testa apraksts	Rezultāts	Komentārs
34.	Pavediena pievienošana – obligāti jāaizpilda lauks “Virsraksts”.	Izpildās	
35.	Pavediena pievienošana – obligāti jāaizpilda lauks “Teksts”.	Izpildās	
36.	Atbildes pievienošana – obligāti jāaizpilda lauks “Teksts”.	Izpildās	
37.	Pavediena pievienošana – ieraksts tiek pievienots datubāzē	Izpildās	
38.	Atbildes pievienošana – ieraksts tiek pievienots datubāzē	Izpildās	
39.	Attēlošana – Iespējams pārslēgties no lapu skatījuma uz ieraksta izvērsto skatījumu un atpakaļ.	Izpildās	
40.	Ierakstu dzēšanas un pievienošanas rezultātā tiek parādīti statusu paziņojumi.	Izpildās	

Testa nr	Testa apraksts	Rezultāts	Komentārs
41.	Attēlošana – pavelienu un atbilžu saraksts izkārtojas tabulas veidā	Izpildās	
42.	Paveliena/atbildes dzēšana – tiek parādīts dzēšanas apstiprināšanas vaicājums.	Izpildās	
43.	Paveliena dzēšana – Paveliena ieraksts tiek dzēsts no datubāzes, kopā ar visām atbildēm.	Neizpildās	Mysql tabulā tika atstātas Paveliena atbildes, PHP kods tika labots.

3.2.7. Galeriju pārvaldības modulis

Tabula 3.7: Galeriju pārvaldības moduļa testēšanas rezultāti

Testa nr	Testa apraksts	Rezultāts	Komentārs
44.	Attēlošana – albumi izkārtojās tabulas veidā, rindās un kolonās, ar redzamu sīktēlu..	Neizpildās	Tika atklāta problēma, ka netika attēloti korekti albumi, kuriem kā sīktēls tika uzstādīts video. Problēma tika novērsta ar izmaiņām PHP kodā.
45.	Attēlošana – klikšķinot uz albumu sīktēliem iespējams nokļūt uz albuma izvērsto skatu.	Izpildās	
46.	Attēlošana – attēli un video izkārtojās tabulas veidā, rindās un kolonās, ar redzamu sīktēlu..	Izpildās	
47.	Attēlošana – klikšķinot uz albuma vienumu sīktēliem iespējams nokļūt uz to izvērsto skatu.	Izpildās	
48.	Albuma pievienošana/labošana – obligāti jāaizpilda lauks “nosaukums”.	Izpildās	
49.	Albuma pievienošana/labošana – ieraksts tiek pievienots datubāzei	Izpildās	
50.	Attēlu un video pievienošana – ieraksts tiek pievienots datubāzei	Izpildās	
51.	Aptipināšana – ja iespējots konkursa apstiprināšanas režīms ir iespējams apstiprināt pievienoto albumu saturu un tiek veikts labojums datubāzē.	Izpildās	
52.	Aptipināšana – ja iespējots konkursa	Izpildās	

Testa nr	Testa apraksts	Rezultāts	Komentārs
	apstiprināšanas režīms ir iespējams noraidīt pievienoto albumu saturu un tiek veikts labojums datubāzē.		
53.	Dzēšana – dzēšot attēlu / video tiek veiktas izmaiņas datubāzē.	Izpildās	
54.	Dzēšana – dzēšot albumu, tas tiek dzēsts kopā ar visu tā saturu no datubāzes.	Izpildās	
55.	Balsošana – ja iespējots balsošanas režīms, iespējams nobalsot par konkrētu albumā esošu elementu.	Izpildās	
56.	Komentāri – var pievienot komentāru konkrētam albuma elementam.	Izpildās	
57.	Komentāri – var dzēst komentāru no albuma elementa.	Neizpildās	Netika atjaunots elementa tabulā komentāru skaits, šī kļūda nekavējoties tika novērsta.
58.	Var mainīt albuma elementu secību izmantojot “vilkt un nomest” mehānismu.	Izpildās	
59.	Var pārvietot konkrētu elementu no viena albuma uz citu.	Izpildās	
60.	Var uzstādīt konkrētu albuma elementu kā albuma sīktēlu.	Neizpildās	Tika atrasta kļūme SQL vaicājumā, tas tika labots.

3.3. Testēšanas rezultātu kopsavilkums

Netika atrastas kritiskas kļūdas, par lielāko kļūdu daļu lietotājs nemaz neuzzinātu, tādēļ arī visticamāk izstrādes laikā tās tika nepamanītas, taču tās protams tika novērstas, padarot sistēmas darbību korektāku. Tomēr tika atrastas arī divas nopietnas kļūdas testēšanas gaitā, kas vai nu neļāva funkcijai darboties pilnībā vai arī radīja būtisku blakus efektu, kas, kļūdas novēršanas gadījumā, varēja radīt būtiskas problēmas sistēmas darbībā.

Kopumā testēšana uzskatāma par veiksmīgu, jo tika salabotas vairākas līdz šim nepamanītas nepilnības un kļūdas.

4. PROJEKTA ORGANIZĀCIJA

Pirms sistēma tika veidota, bija nepieciešams apzināties iepriekš veidotās sistēmas trūkumus un tādējādi veidojot prasības jaunajai sistēmai. Veicot sīkāku izpēti par konkrētiem moduļiem, kas ietilpa iepriekšējā sistēmā, tika secināts kādi moduļi būtu vajadzīgi jaunajā sistēmā un kuri moduļi tā arī netika pilnvērtīgi izmantoti un jaunās sistēmas pamatā netiktu iekļauti. Tika izvērtēta arī reklāmas projektu specifika un izvērtēts vai ir nepieciešamība pēc kāda jauna moduļa, kas agrāk vēl nebija aktuāls, tika nosecināts, ka seši izraudzītie moduļi: “jaunami”, “komentāri”, “balsošana”, “forums”, “atbalstītāji” un “galerijas” nodrošina visu pamata funkcionalitāti standarta projektiem.

Galvenā doma pretēji iepriekšējai sistēmai, kas centās maksimāli ierobežot programmētāja iespējas uz vizuālā izskata un funkcionalitātes izmaiņām, bija veidot sistēmu maksimāli sadrumstalotu pa moduļiem un ļaut programmētājam modificēt specifiskās lietas, neveicot modifikācijas sistēmas pamata kodā, tādējādi radot unikālus projektus, vadoties tikai pēc pamata iezīmēm, nevis pilnīgas izskata un funkcionalitātes šablona. Programmētājam ir iespēja brīvi modificēt esošās funkcijas un veidot jaunas, atkarībā no konkrētā projekta prasībām. Otrs svarīgs aspekts bija sistēmas veiktspēja, jo sociālais tīkls “draugiem.lv”, kurā sistēma tiks tālāk izmantota, ir ar ļoti lielu ikdienas apmeklētāju skaitu, tādēļ arī bija jāveido sistēma, kas spētu noturēt slodzi pie liela apmeklētāju skaita. Liela uzmanība tika pievērsta datubāzes struktūrai.

Pēc tam, kad tika apkopotas visas prasības, tās tika ņemtas par pamatu izstrādājot prasību specifikāciju, pēc kuras pēc tam tika veidots projektējuma apraksts. Sistēmas izstrāde tika sākta projektu pārvaldības paneli, lai varētu veikt pamata operācijas ar projektiem. Pēc projektu pārvaldības paneļa pabeigšanas tika sākta satura vadības sistēmas moduļu izstrāde. Moduļi tika veidoti pakāpeniski, sākot ar tiem, kuros netiek iekļaut neviens cits modulis. Līdz ar pamata moduļu pabeigšanu, tika uzsākts darbs pie moduļiem, kuros tika iekļauti sākumā veidotie moduļi, līdz ar to, lai atvieglotu savstarpējo moduļu integritāti, tika veiktas izmaiņas un uzlabojumi sākotnējos moduļos. Pēc sistēmas moduļu pabeigšanas tika veikta to testēšana izmantojot prasību specifikācijā norādītajām funkcionālajām prasībām.

Kopumā izstrādes process līdzinās ūdenskrituma modelim, taču ar nelielām atkāpēm no tā ierakstā procesa, jo programmēšanas procesa laikā tika veiktas izmaiņas gan prasību specifikācijā, gan projektējuma aprakstā.

5. KVALITĀTES NODROŠINĀŠANA

Lai nodrošinātu augstu produkta kvalitāti, programmēšanas procesā tika ievēroti vairāki priekšraksti, kas attiecas gan uz programmēšanas stilu, gan projekta realizāciju.

Izstrādes sākotnējā posmā, pirms koda izstrādes, tika veikta programmatūrās prasību specifikācijas un projektējuma aprakste izstrāde. Projektējuma apraksts tika salīdzināts ar prasību specifikāciju, vai tas attēlo specifikācijā minētās prasības. Vēlāk, veidojot testēšanas plānu, tas tika bāzēts uz prasību specifikāciju, lai tiktu veikta testēšana atbilstoši prasību specifikācijai.

Kods tika rakstīts, ievērojot vienotu pieraksta stilu un izvairoties no konstrukcijām, kas apgrūtina lasāmību un koda modificēšanu, tādējādi kods ir vieglāk uztverams un iespējams daudz efektīvāk veikt tajā izmaiņas vai izmantot to citos projektos.

Programmas kods tika detalizēti komentēts, moduļu koda failu komentāri tika pierakstīti PHPdoc formā, kas dod iespēju pēc tam automātiski ģenerēt API dokumentāciju, izmantojot speciālu programmatūru. PHPdoc komentāru veids ļauj arī programmētājam veicot modifikācijas viegli pārskatīt klašu metožu pielietojumu.

Īpaši liela uzmanība tika pievērsta drošībai un veiktspējai. Tā kā sistēmu paredzēts izmantot “draugiem.lv” sistēmā, tad īpaša uzmanība tika pievērsta Cross Site Scripting un SQL injekciju novēršanai, kā arī datubāzes indeksiem un struktūrai. Visi moduļu SQL pieprasījumi, tika pārbaudīti ar MySQL iebūvētajiem pārbaudes līdzekļiem uz to efektivitāti un veiktspēju, mēģinot atrast optimālo struktūru un tabulu atslēgu lietojumu.

HTML un CSS kods tika veidots, ievērojot W3C standartus, un tika rūpīgi pārbaudīts, lai tas strādā vienlīdz korekti uz visām izplatītākajām Interneta pārlūkprogrammām. Standartu ievērošana padara vieglāku lapas koda uzturēšanu, un palīdz izvairīties no dažādām kļūdām lapas attēlojumā, kas izpaustos specifiskos apstākļos uz noteiktām pārlūkprogrammām.

6. KONFIGURĀCIJU PĀRVALDĪBA

Katrā PHP koda failā, kurā atrodas kāds no satura vadības sistēmas moduļiem sākumā ir norādīts kuras Draugiem CMS sistēmas versijā tas pašlaik ir iekļauts. Izstrādes laikā, kamēr vēl nebija pabeigta moduļu izstrāde, versijas tika numurētas ar skaitļiem, kas ir mazāki par 1, formā 0.x, kur x vietā tika ievietots skaitlis no 1 līdz 9. Kad sistēmā iecerēto moduļu izstrāde tika pabeigta, gan pašas sistēmas, gan visu moduļu versijas ieguva vērtību 1.0, kas norāda, ka tā ir pirmā pilnvērtīgi strādājošā šī produkta versija.

Sistēmas tālākā attīstībā versiju kontrole ir paredzēta līdzīgā veidā kā izstrādes gaitā, taču veicot to pakāpeniski. Veicot izmaiņas kādā no moduļiem, tiks palielināts tā versija numurs, formā 1.x. Attiecīgi, ja šis modulis netiek iekļauts nevienā citā modulī un nav nepieciešamas izmaiņas citu moduļu kodā, tad tiek palielināts tikai šī paša moduļa versijas numurs. Taču ja šis modulis ir iekļauts vienā vai vairākos citos moduļos, nepieciešams pārliecināties, vai šie moduļi ir saderīgi. Nesaderību gadījumā, kad ir nepieciešama izmaiņas arī citos moduļos, to versiju numuri arī tiek palielināti, veicot izmaiņas. Atkarībā no izmaiņu nozīmīguma, to var paaugstināt par noteiktu apjomu. Ja modulim tiek pievienota neliela jauna funkcionalitāte, tad standarta gadījumā, versija numuru paaugstina, par 0.1.

Sistēmas globālais versijas numurs mainās tikai gadījumā, ja tiek tiek pievienots jauns modulis. Attiecīgi, ja tiek pievienots viens jauns modulis, tad sistēmas versijas numurs paaugstinās par 0.1.

Papildus iekšējai versiju sistēmai izstrādes laikā tika lietots arī konfigurāciju pārvaldības sistēma Subversion un katra veiktā izmaiņa kodā tika piefiksēta. Attiecīgi versiju atjaunošanas procesā pie atjaunošanas ieraksta komentāriem tika norādīts, ja tika mainīta attiecīgā moduļa versija. Ar šo versiju kontroles mehānismu arī viegli citi programmētāji var uzzināt par atjaunotajām versijām un izmaiņām datnēs.

7. DARBIETILPĪBAS NOVĒRTĒJUMS

Ieguldītā darba novērtēšanai tika izmantots automatizēta programmatūra *SLOCCount* [5]. *SLOCCount* atbalsta vairāku programmēšanas valodu kodu un, saskaitot koda rindas projekta ietvaros, spēj noteikt aptuveno darbietilpību un izmaksas. *SLOCCount* izmanto COCOMO Basic modeli [7].

Iegūtie rezultāti no programmas:

Tabula 9.1 *SLOCCount* aprēķinātā sistēmas izstrādes darbietilpība

Kopējais PHP rindīņu skaits (neskaitot komentārus un HTML)	2250
Aptuvenā darbietilpība – 2,4 * (rindīņu skaits tūkstošos) ^{1,05}	5.62 personmēneši (0,47 persongadi)

Kā redzams, tad iegūtais rezultāts neattēlo reālo situāciju un COCOMO Basic modelis nesniedz reālo novērtējumu. Izmantojot šo pašu *SLOCCount* programmatūru ir iespējams veikt arī precīzāku novērtējumu, ievērojot 17 faktorus.

Lai veiktu precīzāko novērtējumu, no sākuma ir jānovērtē projekta sarežģītība, izvēloties vienu no 3 piedāvātajiem variantiem: Organic (vienkāršs, veidots pazīstamā vidē, neizmantojot sarežģītus algoritmus), embedded (ļoti sarežģītā vidē, ar striktām prasībām), vai semidetached (vidēji sarežģīts projekts). Šeit izvēlējot Organic, jo produkts nesatur ļoti sarežģītus algoritmus un ir izstrādāts pazīstamā vidē.

Tālāk ir jāveic 17 faktoru novērtējums. Katru faktoru novērtē ar skalu no 1 līdz 6, jeb “Very Low” (ļoti zems), “Low” (zems), “Nominal” (nominālvērtība), “High” (augsts), “Very High” (ļoti augsts), “Extra High” (īpaši augsts).

Manis dotie novērtējumi:

Tabula 9.2. *SLOCCount* faktoru novērtējums

Faktors	Novērtējums	Reizinātājs
Required software reliability (nepieciešamais drošums)	Nominal	1
Database size (datubāzes apjoms)	Nominal	1
Product complexity (produkta sarežģītība)	Nominal	1
Execution time constraint (Izpildes laika ierobežojums)	Nominal (<50% lietojums no piešķirtā izpildes laika)	1
Main storage constraint (datu glabātuves ierobežojums)	Nominal (<50% lietojums no pieejamās diska vietas)	1

Faktors	Novērtējums	Reizinātājs
Virtual machine (HW and OS) volatility (virtuālās mašīnas (operētājsistēmas un aparatūras) nepastāvīgums)	Nominal (Lielas izmaiņas ik pēc 6 mēnešiem un mazas izmaiņas ik pēc 2 nedēļām)	1
Computer turnaround time (laiks cik jāgaida līdz programma izdod kādu rezultātu)	Low (pielietojums ir interaktīvs)	0.87
Analyst capability (analītiķa spējas)	High	0.86
Applications experience (pieredze izstrādājot programmatūru)	Nominal (3 gadi)	1
Programmer capability (programmētāja spējas)	Nominal	1
Virtual machine experience (pieredze darbā ar virtuālo mašīnu)	High (3 gadi)	0.9
Programming language experience (pieredze programmēšanas valodā)	High (3 gadi)	0.95
Use of "modern" programming practices (modernās programmēšanas pielietojums)	Nominal (daži pielietojumi)	1
Use of software tools (palīgprogrammu pielietojums)	High	0.91
Required development schedule (izstrādes grafika ierobežojumi)	Nominal	1

Tālāk iegūtie reizināti ir jā sareizina savā starpā, sareizinant arī klāt bāzes faktoru, kas iepriekš izvēlētajai "Organic" sarežģītībai ir 2.3. Aprēķins - $2.3 * 0.87 * 0.86 * 0.9 * 0.91 * 0.95 = 1.3389$.

Ar jauno iegūtu faktoru pārreķinot darbietilpību tiek iegūta darbietilpība 3.14 personmēnešu apjomā, kas ir krietni precīzāks novērtējums.

Aprēķinos tika izmantots tikai autora izstrādātais PHP kods. Sākotnējie rezultāti pārspilēja reālo situāciju, jo reāls izstrādes laiks bija aptuveni 4 mēneši, taču izstrāde nenotika vienmērīgi un izstrādes laikā bija pārtraukumi un reālais izstrādes laiks bija aptuveni 3 mēneši. Veiktais faktoru novērtējums deva daudz precīzāku darbietilpības novērtējumu.

8. NOBEIGUMS

Kvalifikācijas darba izstrādes beigās tika iegūta pilnībā funkcionējoša un viegli modificējama sistēma, atbilstoši izvirzītajām prasībām.

Sistēmas izstrādes laikā autors ieguva padziļinātas zināšanas par objektorientētu sistēmu izstrādi un sistēmas veidošanu, kas būtu viegli modificējama un ērti ievietojama projekta lapā. Iepļānotais mērķis tika sasniegti, jo, lai iekļautu lapā kādu no moduļiem nepieciešams tikai viens PHP fails, kas darbosies ar attiecīgo klasi. Vēl, ja projekta dizains ir komplicēts, tad nepieciešams veikt izmaiņas CSS kodā, lai modulis iekļautos projekta lietotāja saskarnē.

Īsi pēc sistēmas pabeigšanas, tika pilnībā likvidēta iepriekšējā sistēma, kas izrādījās neefektīva un grūti modificējama, un aizstāta ar Draugiem CMS. Sistēma īsā laikā pierādīja savu efektivitāti, aizstājot pilnībā iepriekšējās sistēmas funkcijas un ieviešot jaunas. Jau izstrādes laikā, izmantojot sākotnēji izstrādātos moduļus, dažādos projektos, tika secināts, ka izstrādes laiks ir krietni samazinājies.

Pielietojot sistēmu vairāku projektu izstrādē tika atklātas sistēmas pozitīvās iezīmes, kā arī atklāti nelieli trūkumi sistēmas lietojamībā, kas noteikti tiks laboti sistēmas tālākajās versijās. Radās arī vairākas idejas jauniem moduļiem, kas varētu tikt pievienoti sistēmai.

Pēc sistēma izveides, tā ļoti palīdzēja draugiem.lv projektu izstrādātājiem jau vairākos ar reklāmu saistītos projektos, samazinot izstrādes laiku un līdz ar to arī izmaksas, saglabājot augstu projekta kvalitāti.

9. IZMANTOTĀ LITERATŪRA UN AVOTI

1. W3C Recommendation - *XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition)* [tiešsaiste] W3C, 2000. Pieejams: <http://www.w3.org/TR/xhtml1>.
2. W3C Recommendation - *Cascading Style Sheets, level 1* [tiešsaiste] W3C, 1996. Pieejams: <http://www.w3.org/TR/CSS1>.
3. Latvijas valsts standarts LVS 68:1996 - “*Informācijas tehnoloģija. Programminženierija. Programmatūras prasību specifikācijas ceļvedis*” SIA "Latvijas standarts", 1996.
4. W3Schools Online Web Tutorials. [tiešsaiste]. Pieejams: <http://www.w3schools.com/>
5. David A. Wheeler *SLOCCount - a set of tools for counting physical Source Lines of Code* [tiešsaiste]. Pieejams: <http://www.dwheeler.com/sloccount/>.
6. phpDocumentor: The complete documentation solution for PHP [tiešsaiste]. Pieejams: <http://www.phpdoc.org/>.
7. “COCOMO” [tiešsaiste]. Pieejams: <http://en.wikipedia.org/wiki/COCOMO>.
8. “SLOCCount User's Guide” [tiešsaite]
Pieejams: <http://www.dwheeler.com/sloccount/sloccount.html#cocomo>

PIELIKUMI

1. Programmas koda fragments

Iekļāvu balsošanas klases datnes saturu ar kodu, lai gan tas nav pārāk garš, taču pilnībā attēlo izstrādē lietotos principus un metodes.

```
<?
/**
 * @author Raivis Gelsbergs <gelzis@draugiem.lv>
 * @package DraugiemCMS 1.0
 * @version DraugiemVoting 1.0
 */

/**
 * Klase paredzēta balsošanas iespēju pārvaldībai projektā
 * @package DraugiemCMS 1.0
 */
class Voting{
    protected $project;
    protected $user;
    protected $db;

    public $href;
    public $ajax;
    public $own_counts;
    public $adm;
    public $lang;
    public $update_db;
    public $update_table;
    public $update_field;
    public $update_type;

    /**
     * Klases konstruktors
     * @param int $project Projekta ID
     * @param int $uid Lietotāja ID, kurš izmantos klases f-jas, ja = 0, tad izmanto $_SESSION['id']
     * @param bool $own_counts vai klasei jālieto iekšēja elemntu balsu uzskaita, t.i. vai vajad zīģi dati dabulā elmID | votes
     * @param bool $adm vai lietotājam būs pieejamas administratora funkcijas
     */
    public function __construct ($project, $uid = 0, $own_counts = false, $adm = false) {
        $this->project = (int)$project;
        $this->adm = (bool)$adm;
        $this->own_counts = (bool)$own_counts;
        $this->db = $this->conn();
        if (empty($uid)) {
            $this->user = (int)$_SESSION['id'];
        }
    }
}
```

```

    } else {
        $this->user = (int)$uid;
    }
}

/**
 * Metodi izmanto lai iegūtu konkrēta elementa vai vairāku elementu balsu skaitu
 * @param mixed $elmIds masīvs vai integers ar elementu ID
 * @return mixed array(sum,votes,avg) vai false kļūdas gadījumā
 */
public function getCounts($elmIds) {
    if ($this->own_counts) { // ja tiek izmantots iekšējs balsu skaitītājs
        if (is_array($elmIds) && count($elmIds)) { // ja tiek padots masīvs
            $rs = mysql_query("SELECT `elmId`,`sum`,`votes`,`avg` FROM `project_voting_stats` WHERE `elmId` in (" . implode(',',$elmIds) . ") AND `project`='" . $this->project . "'", $this->db) or trigger_error(mysql_error($this->db));
            if (mysql_num_rows($rs)) {
                $data = array();
                while($r = mysql_fetch_assoc($rs)) {
                    $data[$r['elmId']] = $r;
                }
                return $data;
            } else {
                foreach ($elmIds as $elmId) {
                    $data[$elmId] = array('sum'=>0,'votes'=>0,'avg'=>0);
                }
                return $data;
            }
        } else {
            $rs = mysql_query("SELECT `sum`,`votes`,`avg` FROM `project_voting_stats` WHERE `elmId`='" . intval($elmIds) . "' AND `project`='" . $this->project . "'", $this->db);
            if (mysql_num_rows($rs)) {
                $r = mysql_fetch_assoc($rs) or trigger_error(mysql_error($this->db));
                return $r;
            } else {
                return array('sum'=>0,'votes'=>0,'avg'=>0);
            }
        }
    } else {
        return false;
    }
}

/**
 * Metodi izmanto lai iegūtu visu projekta elementu balsu skaitu
 * @param bool $need_user_vote vai nepieciešams arī lietotāja balsojums
 * @return mixed $data[$elmID] = array(sum,votes,avg) vai false kļūdas gadījumā
 */
function getFullCounts($need_user_vote = false){
    if ($need_user_vote) { // ja nepieciešams lietotāja balsojums, vienojam klāt balsu tabulu
        $join = "left join project_voting as v on v.uid=".(int)$need_user_vote." AND stats

```

```

.elmId = v.elmId AND stats.project = v.project";
    $qry = "SELECT `sum`,`votes`,`avg`,v.uid,stats.elmId FROM `project_voting_stats` a
s stats $join WHERE v.project= " . $this->project . " ";
    $rs = mysql_query($qry,$this->db) or trigger_error(mysql_error($this->db));
    } else {
        $rs = mysql_query("SELECT `sum`,`votes`,`avg`,`elmId` FROM `project_voting_stat
s` WHERE `project`=" . $this->project . "" , $this->db) or trigger_error(mysql_error($this-
>db));
    }
    if (mysql_num_rows($rs)) {
        $data = array();
        while($r = mysql_fetch_assoc($rs)) {
            $data[$r['elmId']] = $r;
        }
        return $data;
    } else {
        return false;
    }
}

/**
 * Metodi izmanto lai iegūtu visus lietotāja balsojumus projekta ietvaros
 * @param int $uid lietotāja ID, neobligāts
 * @return array $data [$selm_id] = array(elmId, time, rate)
 */
function getAlluserVotes($uid = 0){
    if (!$uid) {
        $uid = $this->uid;
    }
    $data = array();
    $rs = mysql_query("SELECT elmId, time, rate FROM `project_voting` WHERE `project
`=" . $this->project . " AND uid=$uid", $this->db) or trigger_error(mysql_error($this->db));
    if (mysql_num_rows($rs)) {
        $data = array();
        while($r = mysql_fetch_assoc($rs)) {
            $data[$r['elmId']] = $r;
        }
    }
    return $data;
}

/**
 * Metodi izmanto lai pārbaudītu vai lietotājs ir balsojis
 * @param int $cond_type pārbaudes nosacījums 1 - lietotājs var balsot vienreiz, 2 - lietotāj
s var balsotu vienu reizi dienā, 3 - lietotājs var balsotu vienu reizi dienā par katru elementu
 * @return bool true - lietotājs jau ir balsojis, false - lietotājs nav balsojis
 */
function getVoted($cond_type,$selmId = 0) {
    switch ($cond_type) { // nosaka kurš nosacījuma tips tiek pieprasīts un izmanto attiecīgo
vaicājumu
        case 1:

```

```

        $qry = "select uid from project_voting where uid=". $this->user." AND `project`=" .
$this->project;
        break;
    case 2:
        $qry = "select uid from project_voting where uid=". $this->user." AND `project`=" .
$this->project . " AND DATE_FORMAT(time,'%Y%m%d')=" . date('Ymd')."";
        break;
    case 3:
        $qry = "select uid from project_voting where uid=". $this->user." AND `project`=" .
$this->project . " AND DATE_FORMAT(time,'%Y%m%d')=" . date('Ymd')." AND elmId = " .
(int)$elmId;
        break;
    default:
        $qry = "select uid from project_voting where uid=". $this->user." AND `project`=" .
$this->project . " AND DATE_FORMAT(time,'%Y%m%d')=" . date('Ymd')."";
        break;
    }
    $rs = mysql_query($qry, $this->db) or trigger_error(mysql_error($this->db));
    if (mysql_num_rows($rs)) {
        return true;
    } else {
        return false;
    }
}

/**
 * Metodi izmanto lai lietotājs balsotu par kādu projekta elementu
 * @param int $elmId elementa identifikators
 * @param int $rate Norāda kāds ir lietotāja vērtējums, standarta balsošanā viena balss = 1 v
ērtējums.
 * @return bool izdevās/neizdevās
 */
function Vote($elmId,$rate){
    $qry = "insert into project_voting(`project`,`time`,`uid`,`elmId`,`rate`) values (".$this->
project.",now(),".$this->user.",",".(int)$elmId.",",".(int)$rate.");";
    $rs = mysql_query($qry, $this->db) or trigger_error(mysql_error($this->db));
    if ($rs && mysql_affected_rows($this->db)) {
        if ($this->own_counts) { // ja tiek lietots iekšēja balsu skaitīšana, tiek pieglabāts skaits
            $qry = "insert into project_voting_stats(`project`,`elmId`,`sum`,`votes`,`avg`)
values
(".$this->project.",",".(int)$elmId.",",".(int)$rate.",'1',".(int)$rate.")
on duplicate key update
sum=sum+".$rate.", votes=votes+1, avg = (sum / votes) + (0.02 * votes)";
            $rs = mysql_query($qry, $this->db) or trigger_error(mysql_error($this->db));
            if ($rs) {
                return true;
            } else {
                return false;
            }
        } else {
            if (isset($this->update_db,$this->update_table,$this->update_field,$this->

```



```

        } elseif (isset($this->update_db,$this->update_table,$this->update_field,$this-
>update_type)) {
            if ($this->update_type==1) {
                $qry = "update ".$this->update_table." set votes=0 where ".$this->update_field."
in (".implode(',',$selmIds).)";
            } else {
                $qry = "update ".$this->update_table." set votes=0,avg=0,sum=0 where ".$this-
>update_field." in (".implode(',',$selmIds).)";
            }
            $rs = mysql_query($qry, $this->update_db) or trigger_error(mysql_error($this-
>update_db));
            return mysql_affected_rows($this->update_db);
        }

    } else {
        $qry = "delete from project_voting where project = ".$this->project;
        $rs = mysql_query($qry, $this->db) or trigger_error(mysql_error($this->db));
        if ($this->own_counts) {
            $qry = "delete from project_voting_stats where project = ".$this->project;
            $rs = mysql_query($qry, $this->db) or trigger_error(mysql_error($this->db));
            return mysql_affected_rows($this->update_db);
        }

    } elseif (isset($this->update_db,$this->update_table,$this->update_field,$this-
>update_type)) {
        if ($this->update_type==1) {
            $qry = "update ".$this->update_table." set votes=0";
        } else {
            $qry = "update ".$this->update_table." set votes=0,avg=0,sum=0";
        }
        $rs = mysql_query($qry, $this->update_db) or trigger_error(mysql_error($this-
>update_db));
        return mysql_affected_rows($this->update_db);
    }
}
}
}

/**
 * Metodi izmanto lai klases savienojuma uzstādīšanai ar datubāzi
 * @return resource datubāzes savienojums
 */
function conn(){
    global $db17;
    if(!isset($db17)) include(ROOT . 'inc/dbcon2.php');
    return $db17;
}
}
?>

```

DOKUMENTĀRĀ LAPA

Kvalifikācijas darbs

Ar savu parakstu apliecinu, ka darbs veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai. Piekrītu sava darba publicēšanai internetā.

Autors: _____

(Autora paraksts)

Ar savu parakstu apliecinu, ka esmu lasījis augšminēto kvalifikācijas darbu un atzīstu to par piemērotu/nepiemērotu (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes studiju programmas „Programmēšana un datortīklu administrēšana” kvalifikācijas pārbaudījuma komisijas sēdē.

Darba vadītājs(-ja): _____

(Vadītāja paraksts)

Darbs iesniegts Datorikas fakultātē _____.

(Iesniegšanas datums)

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.

Metodiķe: _____.

(Metodiķes paraksts)

Recenzents: Pēteris Krastiņš.

Darbs aizstāvēts kvalifikācijas pārbaudījuma komisijas sēdē

_____ prot. Nr. _____, vērtējums _____

(Darba aizstāvēšanas datums)

Komisijas sekretārs: _____

(Sekretāra paraksts)