

Latvijas Universitāte
Datorikas fakultāte

Kvalifikācijas darbs

Interneta – radio vortāls

Darba autore:

Jekaterina Hramešina

Darba vadītājs:

Mg.sc.comp Aleksandrs Papiševs

Rīga 2009

Anotācija

Darba mērķis ir izveidot ērtu, viegli pieejamu Interneta radio vortālu. Interneta radio vortāls ir virtuāls projekts, kas ļauj lietotājiem pašiem sastādīt savu dziesmu sarakstu un augšupielādēt sev tīkamās dziesmas, kā arī klausīties populāras vortāla dziesmas. Lai lietotājs varētu klausīties Interneta radio vortālu, viņam nav obligāti jābūt reģistrētam radio vortālā. Toties katram reģistrētam lietotājam ir savs dziesmu sarakstu. Kā arī lietotājam ir iespējams apskatīties savu klausītāko dziesmu statistiku, kā arī visa vortālā klausītāko dziesmu statistiku.

Abstract

A work purpose is to create comfortable, easy in use and accessible Internet radio portal. Internet radio portal is a virtual project, which allows users to implant its song write, as also listens to popular portal songs. Users can be not registered in portal but they still can listen to music. But only registered users can upload they favorite songs and see they statistics in portal. Also users can watch all internet radio portal statistics for a week or month.

Аннотация

Цель работы была сделать удобный и легкодоступный Интернет радио портал. Интернет радио портал является виртуальным проектом, который дает возможность пользователю самому выбрать песни и составить список песен, а также прослушать самые популярные песни. Для того чтобы пользователь мог прослушать песни на портале, ему не обязательно регистрироваться. Но для каждого зарегистрированного пользователя есть возможность добавить свои песни на Интернет радио портал. У пользователя есть возможность просмотреть статистику самых популярных песен на портале.

Satura rādītājs

1. IEVADS	7
2. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	8
IEVADS.....	8
2.1.1. NOLŪKS.....	8
2.1.2. DARBĪBAS SFĒRA.....	8
2.1.3. DEFINĪCIJAS, AKRONĪMI UN SAĪSINĀJUMI.....	8
2.1.4. SAISTĪBA AR CITIEM DOKUMENTIEM.....	9
2.1.5. DOKUMENTA PĀRSKATS.....	9
2.2. VISPĀRĪGS APRAKSTS.....	9
2.2.1. PRODUKTA FUNKCIJAS.....	9
2.2.2. LIETOTĀJA RAKSTURIEZĪMES.....	10
2.2.3. VISPĀRĒJIE IEROBEŽOJUMI.....	10
2.2.4. PIENĒMUMI UN ATKARĪBAS.....	10
2.3. FUNKCIONĀLĀS PRASĪBAS.....	10
2.3.1. ATSKANOTĀJA FUNKCIJAS UN PRASĪBAS.....	11
2.3.2. LIETOTĀJU FUNKCIJAS.....	12
2.3.2.1. <i>Lietotāja reģistrēšana, profila saglabāšana un labošana</i>	12
2.3.2.2. <i>Ielogošana</i>	13
2.3.3. DZIESMAS PIEVIENOŠANA	13
2.3.4. DZIESMU SARAKSTA IZVEIDE.....	15
2.3.5. DZIESMU STATISTIKAS UZSKAITE.....	15
3. ĀRĒJĀ SASKARNE.....	16
3.1.1. LIETOTĀJA SASKARNE.....	16
3.1.2. APARATŪRAS SASKARNE.....	16
3.1.3. PROGRAMMATŪRAS SASKARNE.....	16
2.5. VEIKTSPĒJAS PRASĪBAS.....	18
2.6. ATRIBŪTI.....	18
2.6.1. DROŠĪBA.....	18
3. UZTURAMĪBA.....	18
3.6.1. PIEEJAMĪBA.....	18
3.1. IEVADS.....	19
3.1.1. NOLŪKS.....	19
3.1.2. DARBĪBAS SFĒRA.....	19
3.1.3. DEFINĪCIJAS UN SAĪSINĀJUMI.....	19
3.1.2. <i>Saistība ar citiem dokumentiem</i>	19
3.2. DATU PLŪSMAS DIAGRAMMAS.....	20
3.3. ER MODEĻI.....	25
3.3.2. REALIZĀCIJAS ER MODELIS.....	26
3.3. DATU BĀZES TABULU DETALIZĒTAIS APRAKSTS.....	27
3.4. PROGRAMMATŪRAS MODUĻU PROJEKTĒJUMS.....	29
3.5.1. INTERFEISS AR FLASH PLEJERI.....	29
3.5.2. DZIESMAS SARAKSTA IZVEIDE, IZMANTOJOT DRAG AND DROP.....	30

3.5.3. STATISTIKA.....	32
3.6.1. NOLŪKS.....	35
3.6.2. FUNKCIJAS.....	35
<u>4. TESTĒŠANA.....</u>	<u>37</u>
4.1. ATSKANOTĀJS:.....	37
4.2. LIETOTĀJU SISTĒMA.....	42
4.3. DZIESMAS PIEVIENOŠANA.....	44
<u>5. PROJEKTA ORGANIZĀCIJA.....</u>	<u>48</u>
<u>6. KVALITĀTES NODROŠINĀŠANA.....</u>	<u>49</u>
<u>7. KONFIGURĀCIJU PĀRVALDĪBA.....</u>	<u>50</u>
<u>8. DARBIETILPĪBAS NOVĒRTĒJUMS.....</u>	<u>51</u>
<u>PROJEKTA REĀLAIS UZSĀKŠANAS LAIKS IR 3. MARTS, PROJEKTS TIKA REGULĀRI IZSTRĀDĀTS, LABOTS UN PAPILDINĀTS.....</u>	<u>51</u>
<u>2. 9. IZMANTOTĀ LITERATŪRA.....</u>	<u>52</u>

1. Ievads

Mūsdienās pastāv dažādi mūzikas vortāli, bet pārsvarā tajos lietotājiem ir jāklausa visiem vienāda mūzika vai arī jāmaksā dalības maksa. Kā arī mūzikas vortālos ir pārāk daudz reklāmas, kas traucē pilnvērtīgi lietot vortālu.

Interneta radio vortāls ir bezmaksas vortāls, kas satur dažādus mūzikas žanrus. Interneta radio vortāls ir izveidots, lai cilvēki, kuriem patīk brīvajā laikā klausīties sev patīkamu mūziku, kā arī interneta radio vortāls dod iespēju iepazīties ar jauniem mūzikas žanriem, māksliniekiem un jaunām dziesmām. Interneta radio vortāls ir ērts lietojamībā, lietotājam nevajag būt pierēģistrētam Interneta radio vortālam, lai noklausītos tajā esošo mūziku. Kā arī Interneta radio portāls ļauj lietotājam ātri un ērti atrast populārākās dziesmas, kas attēlotas grafiskā veidā.

2. Programmatūras prasību specifikācija

2.1. Ievads

2.1.1. Nolūks

Šī dokumenta nolūks ir specificēt interneta radio vortālam.

Pēc saskaņošanas ar pasūtītāju prasību specifikācija kalpos par pamatu turpmākai vortāla izstrādei. Prasību specifikācija domāta lietošanai izstrādātājiem, kas nodarbosies ar projektēšanu, kodēšanu un testēšanu.

2.1.2. Darbības sfēra

Portāla mērķis ir dot cilvēkiem iespēju klausīties populāru mūziku, kā arī padalīties ar savu mūzikas gaumi ar citiem vortāla lietotājiem prognozēt..

Šī dokumenta darbības sfēra nosaka visu portāla funkcionalitāti. Tajā skaitā lietotāju sistēmu, prognozējamo daļu un administrēšanu.

Specifikācija izstrādāta atbilstoši līgumam ar pasūtītāju.

2.1.3. Definīcijas, akronīmi un saīsinājumi

Tabula 1. Dokumentā izmantotie saīsinājumi un definīcijas

Definīcija/Saīsinājums	Skaidrojums
IS	Informācijas sistēma
Apache	Bezmaksas, atklātā pirmkoda tīmekļa serveris ar daudzplatformu atbalstu
MySQL	Populāra bezmaksas DBPS
PHP	Interpretējama programmēšanas valoda, kuru plaši izmanto servera puses lietojumos dinamiska tīmekļa satura ģenerēšanai
Jaucējfunkcija	Algoritms, kas pārveido dažāda garuma tekstu fiksēta garuma neprognozējamā izvadē
DBPS	Datu bāzes pārvaldības sistēma
Use case	Lietošanas piemērs
Playlist	Dziesmu saraksts
Drag and drop	Vilkt un nomest, veic ar peles palīdzību, pārvietojot objektu līdz tā sakrišanai ar tās lietojumprogrammas ikonu, kura izpilda nepieciešamo operāciju.
Radiopoga	Pogu virkne ekrānā. Tā pieļauj tikai vienas pogas izvēli

Ritjosla	Taisnstūrveida josla loga ietvara labajā vai apakšējā malā, ko izmanto loga satura pārvietošanai vertikālā vai horizontālā virzienā
----------	---

2.1.4. Saistība ar citiem dokumentiem

Dokuments izstrādāts saskaņā ar:

1. Standarts LVS 68:1996 “Informācijas tehnoloģija - Programminženierija - Programmatūras prasību specifikācijas ceļvedis”.

2.1.5. Dokumenta pārskats

Dokuments ir sadalīts sešās nodaļās:

1. Dokumenta apraksts
2. Īss sistēmas apraksts
3. Precīzi formulētas konkrētas prasības. Katrā prasība ir vēl sadalīta 4 daļās: ievaddati, apstrāde, izvaddati un kļūdu ziņojumi.
4. Ārējas saskarnes principi un piemēri
5. Veiktspējas prasības
6. Programmatūras kvalitātes atribūti

Nodaļas ir sadalītas apakšnodaļās balstoties uz standartu #1.

2.2. Vispārīgs apraksts

2.2.1. Produkta funkcijas

Portāla mērķis ir ļaut cilvēkiem klausīties un apmainīties ar mūziku. Cilvēki varēs augšupielādēt savus mp3 mūzikas failus un klausīties citu ielādētās mp3 kompozīcijas. Lai nepārkāpt autortiesību likumu, dziesmu lejupielāde netiks ļauta, var tikai iepazīšanas nolūkos noklausīties.

Ir 2 tipu lietotāji: reģistrētie un neregistrētie. Abu tipu lietotāji var veidot savu *playlistu* un klausīties. Reģistrētie papildus var augšupielādēt mūziku.

Vortāls darbosies daudzlietotāju režīmā, kas nozīmē, ka daudzi cilvēki varēs vienlaikus klausīties savu izvēlētu mūziku.

Gan dziesmas, gan artisti tiks sagrupēti pēc žanriem, lai atvieglot meklēšanu. Paredzēts ir klausītāju populārāko dziesmu TOP.

Kāpēc vajadzīgs tāds vortāls?

Jā, mēs visi zinām, ka ir last.fm un citi “monstri” konkurēt ar kuriem ir neiespējami. Bet var arī paskatīties no cita punkta un atrast mūsdienu interneta medija portālu trūkumus. Gandrīz visiem, ir

noderīga funkcija “atskaņot līdzīgus izpildītājus”. Tie, kuri lietoja sapratīs, ka ne vienmēr “līdzīgi izpildītāji” līdzīgi patīk. Un nevar pat redzēt ko tieši atlasīja sistēma, lietotājs redz tikai tekošu dziesmu. Tāpēc mums būs visu laiku redzams playlist, kuru var jebkurā laikā rediģēt. Vēl ir nepatīkami, kad gribi tikai lietot portālu, lai klausīties, bet lietotājam piedāvā veselu sociālu tīklu, nevajadzīgu statistiku, obligātu reģistrāciju, reklāmu, “dzeltenās preses” mūzikas ziņas. Nekā tāda dotais vortāls nepiedāvās!

2.2.2. Lietotāja raksturiezīmes

Portāls ir orientēts uz mūzikas klausītājiem. Tāpēc tiek uzskatīts, ka lietotājs prot lietot mūzikas atskaņošanas programmas. Portāla lietotājiem jābūt minimālām prasmēm darbā ar pārlūkprogrammām. Vēlams, lai lietotājiem būtu pieredze kādā citā WEB-bāzētā daudzlietotāju sistēmā.

2.2.3. Vispārējie ierobežojumi

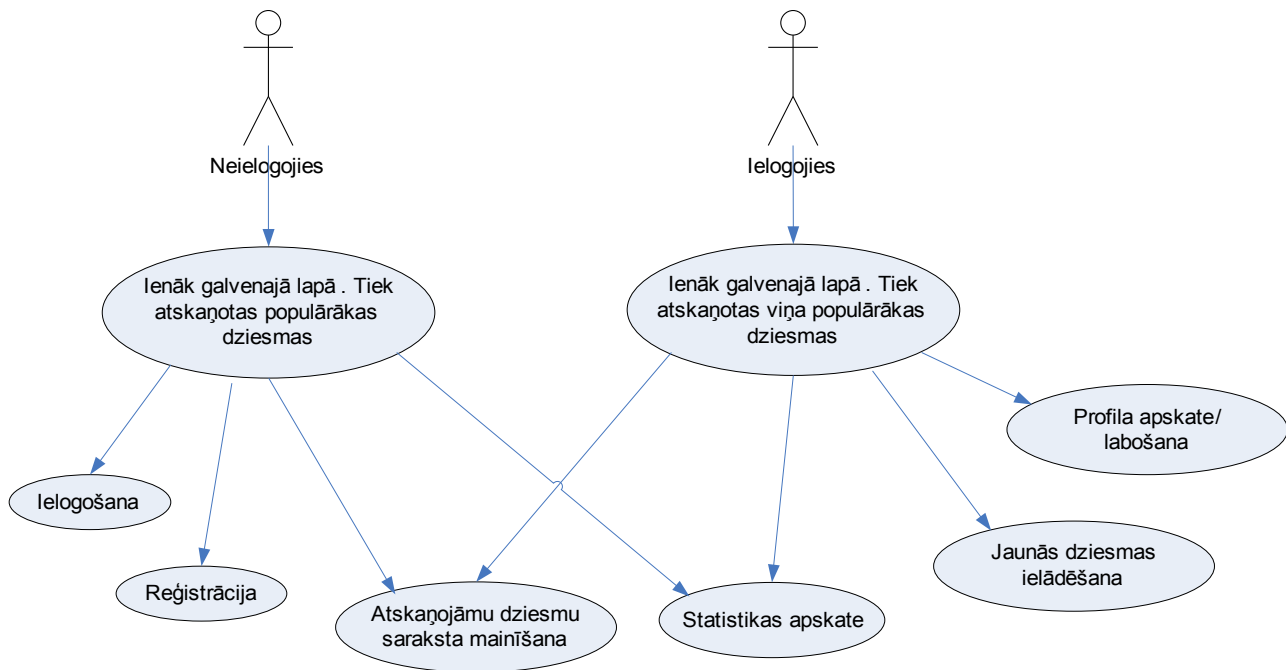
Sistēmā ir paredzētas 3 dažādas lietotāju grupas: reģistrētie lietotāji, neregistrētie lietotāji un administratori. Katram lietotājam var būt vairākas tiesības vienlaikus, t.i. lietotājam var būt gan administratora, gan parasta lietotāja tiesības.

2.2.4. Pieņēmumi un atkarības

Ir pieņemts, ka moderatori ievadīs dziesmu žanrus, rediģēs dziesmas. Sistēmai ir jābūt funkcionējošai, kad tiešsaistē esošo lietotāju skaits nepārsniedz 50. Lietotāja pusē ir jābūt interneta pieslēgums. Tā kā IS atradīsies uz servera, portāla darbība ir atkarīga no servera veiktspējas.

2.3. Funkcionālās prasības

Visas funkcijas var sadalīt atsevišķās daļās: mūzikas atskaņotāja funkcijas, lietotāju funkcijas un citas funkcijas.



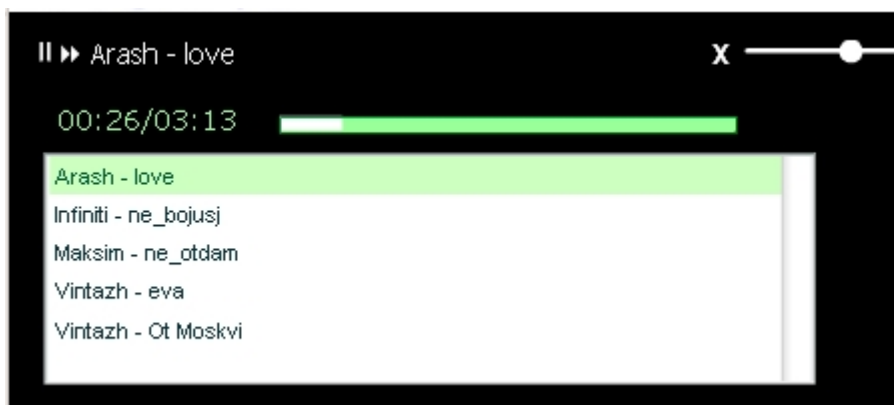
Šī diagramma atspoguļo lietotāju iespējas izmantojot Interneta radio vortāla dziesmu atsakņotāju.

2.3.1. Atskaņotāja funkcijas un prasības

- 2.3.1.1. Pašlaik atskaņotās dziesmas nosaukums. Augšējā kreisajā stūrī lietotājs redz atskaņotās dziesmas nosaukumu un izpildītāja vārdu.
- 2.3.1.2. Play/pause. Pastāv poga, kas dod iespēju lietotājam apstādināt dziesmu jebkurā momentā, kā arī pēc tam atļaut klausīties no tās pašas vietas kur dziesma tika apstādināta.
- 2.3.1.3. Next song. Lietotājam ir iespēja pašrocīgi pāriet uz citu dziesmu, ja atskaņotā dziesma ir pēdējā, tad nospiežot pogu next, dziesmu saraksts tiek pārlīkts uz sākumu.
- 2.3.1.4. Kad dziesma ir beigusies, tad uzreiz tiek atskaņota nākamā dziesma, ja atskaņotā dziesma ir pēdējā, tad nākamā atskaņota dziesma ir pirmā no dziesmu saraksta.
- 2.3.1.5. Visas dziesmas, kas ir iekļautas dziesmu sarakstā visas iekļautās dziesmas.
- 2.3.1.6. Ritjosla. Ja dziesmu sarakstā ir lielāks nekā pats atskaņotāja loga, tad parādās ritjosla, ar kuras palīdzību lietotājs var pabīdīt dziesmu sarakstu uz augšu vai uz leju.
- 2.3.1.7. Volume. Lietotājam ir iespēja regulēt atskaņotās dziesmas skaļumu, kā arī vispār uzlikt klusuma režīmu. Pēc noklusējuma dziesmas atskaņošanas skaļums ir 75 %.
- 2.3.1.8. Dziesmas ilgums un atskaņošanas laiks. Lietotājam ir iespēja redzēt dziesmas garumu, kā arī cik daudz no dziesmas laika jau tika atskaņots.

2.3.1.9. Dzēst dziesmu no saraksta. Lietotājam ir iespēja izdzēst dziesmu no sava atsakņot dziesmu saraksta.

Att. 2 Atskaņotais



2.3.2. Lietotāju funkcijas

2.3.2.1. Lietotāja reģistrēšana, profila saglabāšana un labošana

Mērķis:

Ir paredzēta lietotāju reģistrēšanai, profila datu saglabāšanai, labošanai

Ievaddati:

Jauna lietotāja reģistrācijai obligātie ievaddati ir

- lietotāja vārds Varchar(255)

Neobligātie ievaddati, no kuriem arī veidosies lietotāja profils:

- parole Varchar(255)
- e-pasts Varchar(255)
- attēls attēla fails

Apstrāde:

Ja reģistrēšanas laikā sistēmā jau ir lietotājs ar ievadīto lietotāja vārdu, tad sistēma attēlo kļūdas paziņojumu #1. Mainīt profilu lietotājs var tikai ielogojoties IS. Attēla failam ir jābūt ne lielākam par 64 kB un ar paplašinājumu jpg, gif vai png, ja augšupielādētā bilde pārsniedz 64 Kb apjomu tad tiek parādīts kļūdas ziņojums #2 Pēc reģistrācijas lietotāja augšupielādētu dziesmu skaitītājs ir 0. Parole drīkst būt neobligāta, jo radio vortālā neglabājas lietotāju sensitīvie dati, līdz ar to lietotājama tiek piedāvāta iespēja par paroles izveidi. Kā arī daudziem cilvēkiem sagādā grūtības atcerēties savas paroles, it īpaši, ja cilvēks nelieto vortālu regulāri. Ja lietotājam ir velme padalīties ar savu mūzikas gaumi, tad lietotājam ir jāiedod savas lietotāja vārds citam lietotājam.

Izvaddati:

Informācija par lietotāju tiek saglabāta atbilstoši ievaddatiem.

Kļūdu ziņojumi:

1. «User with this Login already exists. Please choose another one.»;
2. «Only images under 64Kb are accepted for upload»

2.3.2.2. Ielogošana

Mērķis:

Ir paredzēts reģistrēto lietotāju piekļūšanai savam profilam interneta radio vortālā.

Ievaddati:

Reģistrētā lietotāja ielogošanās obligātie ievaddati ir:

- lietotāja vārds Varchar(255)
- parole Varchar(255)

Apstrāde:

Ja ielogošanās laikā lietotājs ir ievadījis nepareizo lietotāju vārdu vai paroli, tad sistēma attēlo kļūdas paziņojumu #1. Lietotāju vārds un parole tiek salīdzinātas ar datu bāzē glabājamos lietotāju vārdiem un parolēm. Ja lietotāju ielogošanās ir notikusi veiksmīgi, tad lietotājs pāriet uz savu profillapu.

Izvaddati:

Pēc ielogošanās lietotājs var apskatīt savu profilu, kurš sastāv no lietotāja vārda, bildes, augšup ielādētās dziesmas un savu sastādīto dziesmu sarakstu, kuru lietotājs var rediģēt.

Kļūdu ziņojumi:

1. Login or password is incorrect;

2.3.3. Dziesmas pievienošana

Mērķis:

Lietotājs var pievienot un saglabāt sev tīkamās dziesmas radio dziesmu bibliotēkā.

Ievaddati:

Dziesmas pievienošanai nepieciešamie dati:

- dziesmas nosaukums Varchar(255)
- dziesmas izpildītāja vārds vai grupas nosaukums Varchar (255)
- dziesmas fails mp3

Katras dziesmas neobligāts lauks:

- žanrs Varchar(255)

Apstrāde:

Ja lietotājs vēlas ielādēt sev vēlamo dziesmu, lietotājam ir jāievada tās nosaukums, ja dziesmas nosaukums netiek ievadīts, tad lietotājs redz ziņojumu #1. Nākamais solis, lietotāja no piedāvātā izpildītāju saraksta ir jāizvēlas atbilstošais dziedātājs vai grupa, ja mākslinieks nav atrodams sarakstā, tad lietotājam ir iespēja no combobox tukšais lauks un jāievada izpildītāja vārds, ja dziedātāja vārds netiek ievadīts, tad tiek parādīts #2 ziņojums. Ja lietotājs nejauši ir izvēlējis mākslinieku gan no saraksta, gan pierakstījis arī savu izpildītāja versiju, tad datu bāzē tiek saglabāts lietotāja ievadītais mākslinieks. Trešajā solī lietotājam no žanru saraksta ir jāizvēlas žanrs, kas atbilst ielādējamai dziesmai. Beidzamajā solī lietotājam no dialoga loga ir jāizvēlas dziesma vai arī lietotājam ir iespējams pašrocīgi ievadīt adresi, kur uz viņa datora glabājas atbilstošā dziesma. Radio datubāzē glabājas tieši dziesmas atrašanas adrese, pati dziesma, t.i. mp3 fails, glabājas uz servera. Lietotājam ir jānorāda korekta dziesmas atrašanās vieta, citādi lietotājam paziņos #3 kļūdas ziņojumu Dziesmai ir jābūt mp3 formātā, pretējā gadījumā tiks parādīts paziņojums #4. Mp3 nedrīkst pārsniegt 15 Mb apjomu, citādi tiek parādīts kļūdas ziņojums #5.

Tabula 2 dziesmas ilguma un kvalitātes attiecība

Mp3 dziesmas ilgums pie faila izmēra 15Mb	Mp3 dziesmas kvalitāte 15360 122880
16 min	128 kb/sek
10:40 min	192 kb/sek
8 min	256 kb/sek
6:24 min	320 kb/sek

Izvaddati:

Datu bāzē tiek saglabāta informācija par dziesmu, kā arī dziesmas atrašanās ceļu.

Kļūdu ziņojumi:

1. "Enter song name please!"
2. "Choose artist or enter artist name, please!"
3. "The file does not exist! Please choose another one!"
4. "Song file should be mp3 format!"
5. "File size should be less than 15Mb!"

2.3.4. Dziesmu saraksta izveide

Mērķis:

Katrs interneta – radio vortāla apmeklētājs var izveidot sev tīkamo dziesmu sarakstu.

Lietotājs no piedāvāta mākslinieku saraksta izvēlas sev tīkamās dziesmas, vai arī lietotājs var izvēlēties dziesmas pēc žanra, tad lietotāja sarakstā parādās visas dziesmas no šī žanra.

Izvaddati:

Tiek izveidots XML fails, kurš satur informāciju par izvēlēto dziesmu sarakstu.

2.3.5. Dziesmu statistikas uzskaitē

Mērķis:

Apskatīt un noklausīties populāras dziesmas, kas visvairāk tika noklausītas portālā.

Ievaddati:

Nav

Apstrāde:

Lietotājs izvēlas dziesmu un noklausās to konkrētu laiku, šo laiku nosaka administrators. Kad dziesmas gabals ir atskaņots, datu bāzē tiek izdarīts ieraksts, kas satur dziesmas nosaukumu, izpildītāju, lietotāju, kurš noklausījies šo dziesmu, kā arī datumu, kad šī dziesma tika atskaņota. Līdz ar to katru dienu dziesmu tops tiek atjaunots.

Izvaddati:

Lietotājs var apskatīties savu klausītāko dziesmu sarakstu pa nedēļu. Kā arī lietotājs var apskatīties populāru visu lietotāju dziesmu sarakstu.

3. Ārējā saskarne

3.1.1. Lietotāja saskarne

Saskarnei ir jābūt angļu valodā. Radio interneta vortāla atskaņotājs ir uztaisīts ar Flash programmatūras palīdzību. Atskaņotājam ir pieejams ritu josla, kas veido atskaņotāju labāk pārskatāmu lietotājam.

Galvenā interneta radio vortāla īpatnība ir tā, ka lietotājam ir ērti strādāt ar radio vortālu. Dziesmu saraksta veidošana strādā ar „drag and drop” palīdzību, tas nozīmē, ka lietotājs pārvieto dziesmu no piedāvātā dziesmu klāsta, savā dziesmu sarakstā. Kā arī dziesmas un izpildītāji ir savietoti šķirķļos, kas dod iespēju lietotājam ātri pārskatīt un izvēlēties dziesmas no esošo dziesmu un mākslinieku saraksta.

3.1.2. Aparatūras saskarne

Serverim ir jābūt tik jaudīgam, lai tas spētu atbalstīt 50 cilvēkus vienlaicīgi. Servera prasības ir:

Tabula 3 Servera prasības

<i>Prasības serverim</i>	Minimālas	Rekomendējamās
Procesors	3.0 GHz divkodola	4.5 GHz divkodola
Operatīvā atmiņa	2GB	8GB
Cieta diska ātrums	10000 rpm	15000 rpm
Cietā diska apjoms	1TB	4TB

Pie minimālām prasībām serveris var uzglabāt aptuveni 69905 dziesmas, bet netiek skaitītas klāt servera operatīvā sistēma un servera programmas.

Klienta datoram ir jābūt interneta pieslēgumam, minimāla pieslēguma kanāla caurlaides spējai ir jābūt atkarīga no izvēlētās dziesmu kvalitātes, praktiski minimālam ātrumam ir jābūt 512kb/sek.

Klienta datora minimālās prasības ir:

- Procesors – 800MHz
- Operatīvā atmiņa – 512 MB

Šīs ir minimālās prasības klienta datoram, jo klienta datoram ir jābūt iespējamam uzinstalēt uz sava datora interneta pārlūkprogrammu un Flash player.

Lai korekti izmantotu interneta radio vortālu, lietotājam ir nepieciešams kustināms kursors.

3.1.3. Programmatūras saskarne

IS ir jābūt pēc iespējas viegli paplašināmai un modificējamai. Serveris būs ar *Ubuntu 7.10 Server Edition* operētājsistēmu. Interneta radio vortāla izstrādes laikā var izmantot Denver -3 2008-01-13 programma.

Izstrādājot interneta radio vortālu tika izmantota PHP5 valoda, MySQL-5.1 datu pārvaldības sistēma, šī sistēma tika izvēlēta sakarā ar to, ka tai ir opcija „Plānotājs” (Scheduler), Apache 2.2.1. Macromedia Flash 8, ar šīs programmas palīdzību tika izstrādāts interneta radio vortāla atskaņotais. Klienta programmatūras prasībās nav svarīga operētājsistēma, galvenais lai uz tās būtu uzinstalēta interneta pārlūkprogramma (minimālās prasības Explorer 6 vai Opera 9, vai Mozila Firefox 2, vai Safari, Google chrom, Konquer).Klienta datorā interneta pārlūkprogramma ir jābūt ieslēgtiem Cookies. Lietotāja datoram ir jābūt uzinstalētam Flash Player 9. Lai klientam būtu iespējams apskatīt statistiku, datorā ir jābūt uzinstalētam vismaz Adobe Flash Player 10.

2.5. Veiktspējas prasības

Sistēmas veiktspēja ir atkarīga no klienta interneta pieslēguma ātruma. Pastāv tādas darbības, kas kavē vortāla darbības laiku, piemēram, dziesmu ielāde. Atverot dziesmu statistiku, tiek izpildīts MySQL pieprasījums.

Pāreja no vienas dziesmas uz nākamo aizņem ne vairāk kā 1 vienu sekundi.

2.6. Atribūti

2.6.1. Drošība

Pie reģistrēšanās lietotājam nevajag obligāti ievadīt paroli. Lietotāju paroles tiek šifrētas ar jaucējfunkcijas (*hash function*) md5 palīdzību un neglabājas atklātā veidā.

3. Uzturamība

Ir nepieciešams cilvēks, kas prot nokonfigurēt servera programmatūru. Administrators uzturēs interneta radio vortālu, administratora pienākumos ietilpst lietotāju pārvaldība, kā arī žanru korekcija.

3.6.1. Pieejamība

IS atļautais nepieejamības laiks ir 20 minūtes mēnesī.

4. Programmatūras projektējuma apraksts

3.1. Ievads

3.1.1. Nolūks

Programmatūras projektējuma apraksta (PPA) nolūks ir aprakstīt, kā realizācijas vidē tiks realizētas programmatūras „Interneta radio vortālā” prasību specifikācijā iekļautās prasības. Dokumentā ir aprakstīts izstrādājamās programmatūras „Interneta radio vortālā” projektējuma apraksts. PPA ir paredzēta programmatūras izstrādātājiem.

3.1.2. Darbības sfēra

Programmatūras prasību specifikācija apraksta programmatūrai „Interneta radio vortāls” izvirzītās prasības.

3.1.3. Definīcijas un saīsinājumi

Tabula 4. Dokumentā izmantotie saīsinājumi un definīcijas

Definīcija/Saīsinājums	Skaidrojums
IS	Informācijas sistēma
Apache	Bezmaksas, atklātā pirmkoda tīmekļa serveris ar daudzplatformu atbalstu
MySQL	Populāra bezmaksas DBPS
PHP	Interpretējama programmēšanas valoda, kuru plaši izmanto servera puses lietojumos dinamiska tīmekļa satura ģenerēšanai
Jaucējfunkcija	Algoritms, kas pārveido dažāda garuma tekstu fiksēta garuma neprognozējamā izvadē
DBPS	Datu bāzes pārvaldības sistēma
Use case	Lietošanas piemērs
Playlist	Dziesmu saraksts
Drag and drop	Vilkt un nomest, veic ar peles palīdzību, pārvietojot objektu līdz tā sakrišanai ar tās lietojumprogrammas ikonu, kura izpilda nepieciešamo operāciju.
Radiopoga	Pogu virkne ekrānā. Tā pieļauj tikai vienas pogas izvēli
Ritjosla	Taisnstūrveida josla loga ietvara labajā vai apakšējā malā, ko izmanto loga satura pārvietošanai vertikālā vai horizontālā virzienā

3.1.2. Saistība ar citiem dokumentiem

Dokuments sagatavots, pamatojoties uz:

- Standarts LVS 72:1996, Ieteicamā prakse programmatūras projektējuma aprakstīšanai;

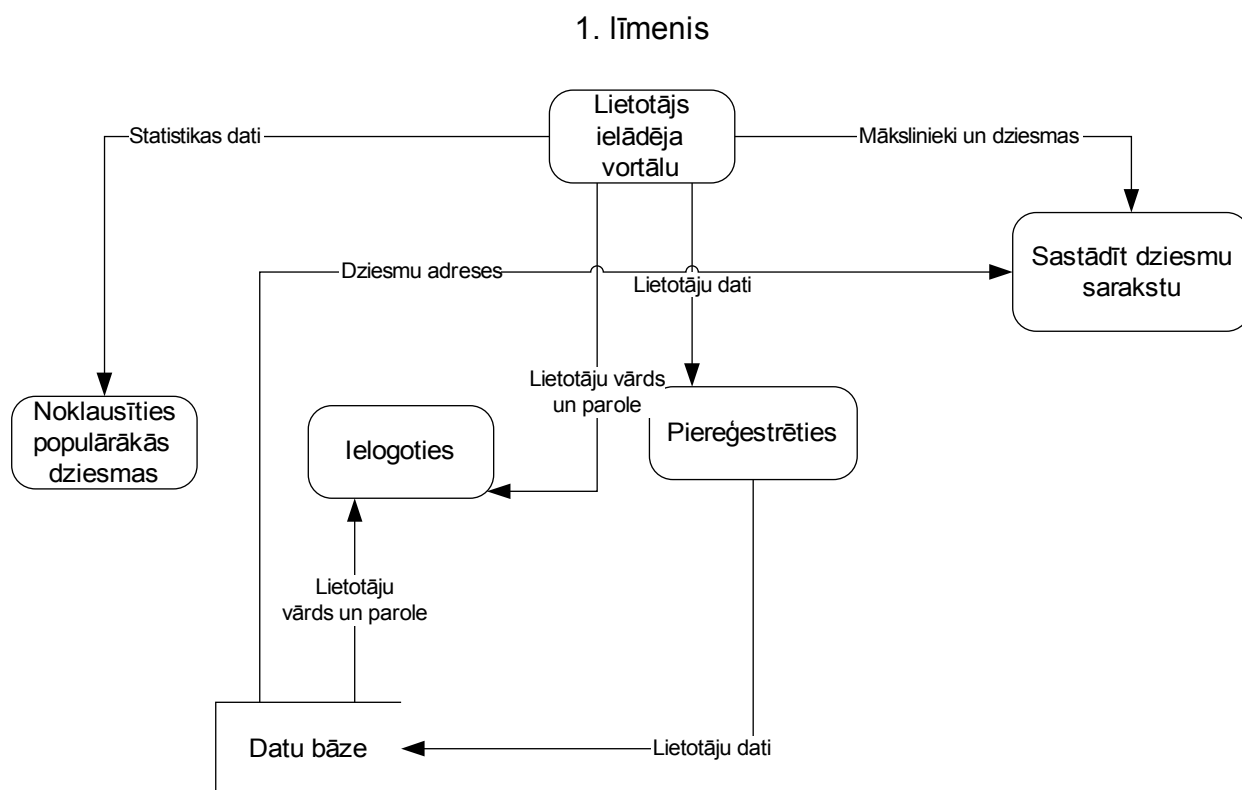
3.2. Datu plūsmas diagrammas

Šajā nodaļā ir aprakstīts programmatūras „Interneta radio vortāla” datu plūsmas:

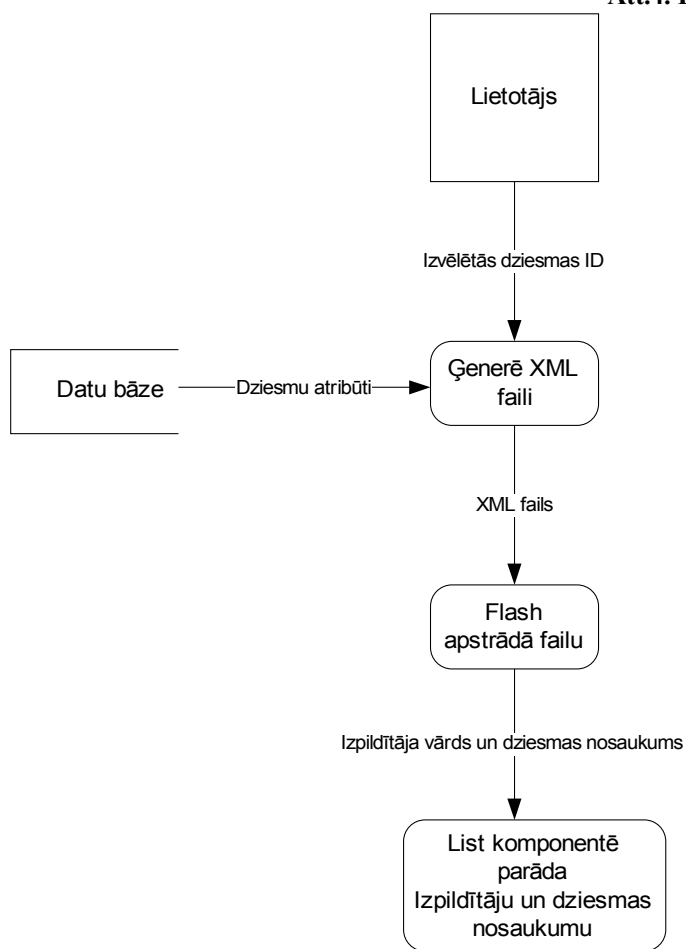
1.1. līmenis (skat. Attēls 3. Datu plūsmas diagramma - 1. līmenis);

2. Dziesmu saraksta veidošana

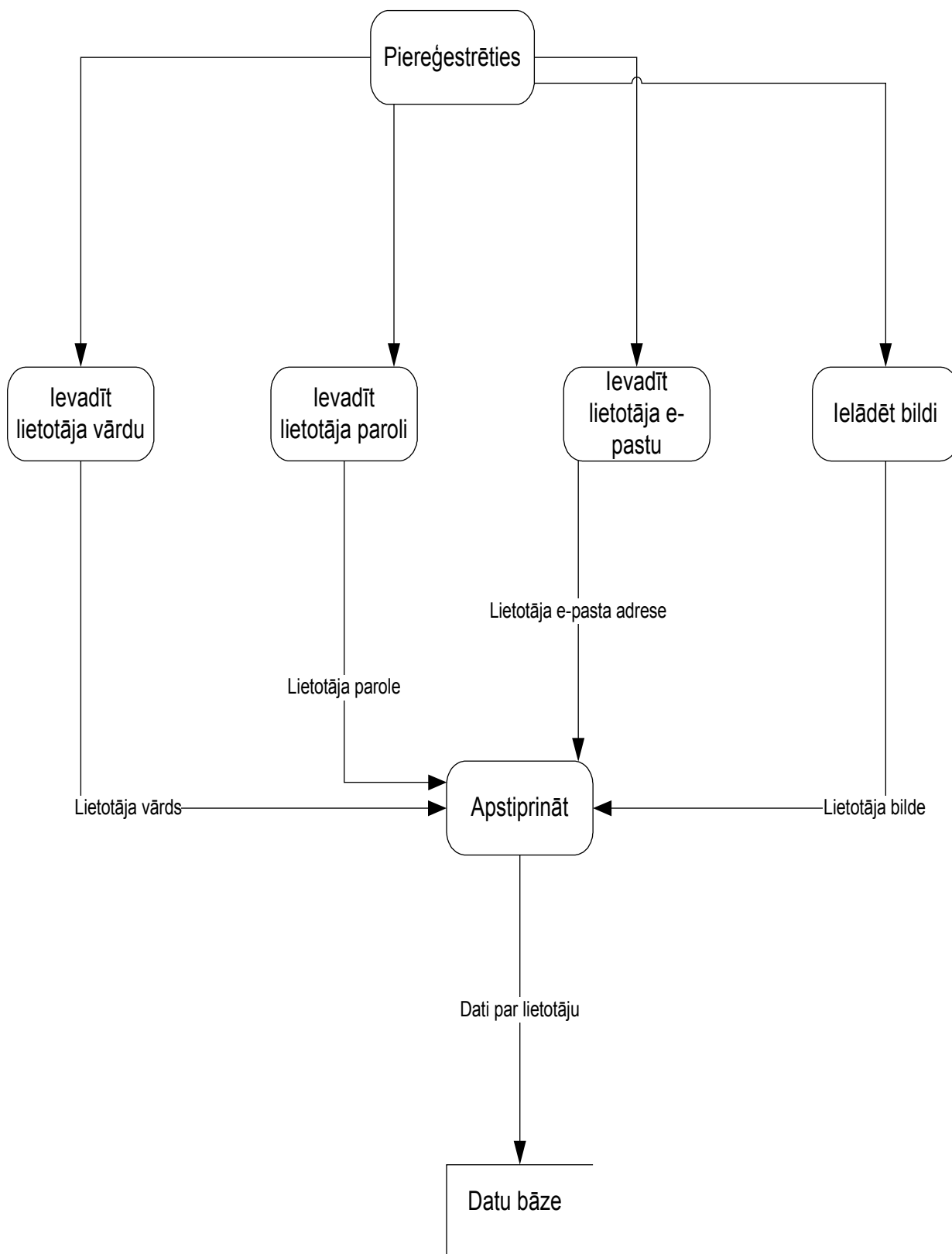
Att. 3 Datu plūsmas diagramma – 1.līmenis



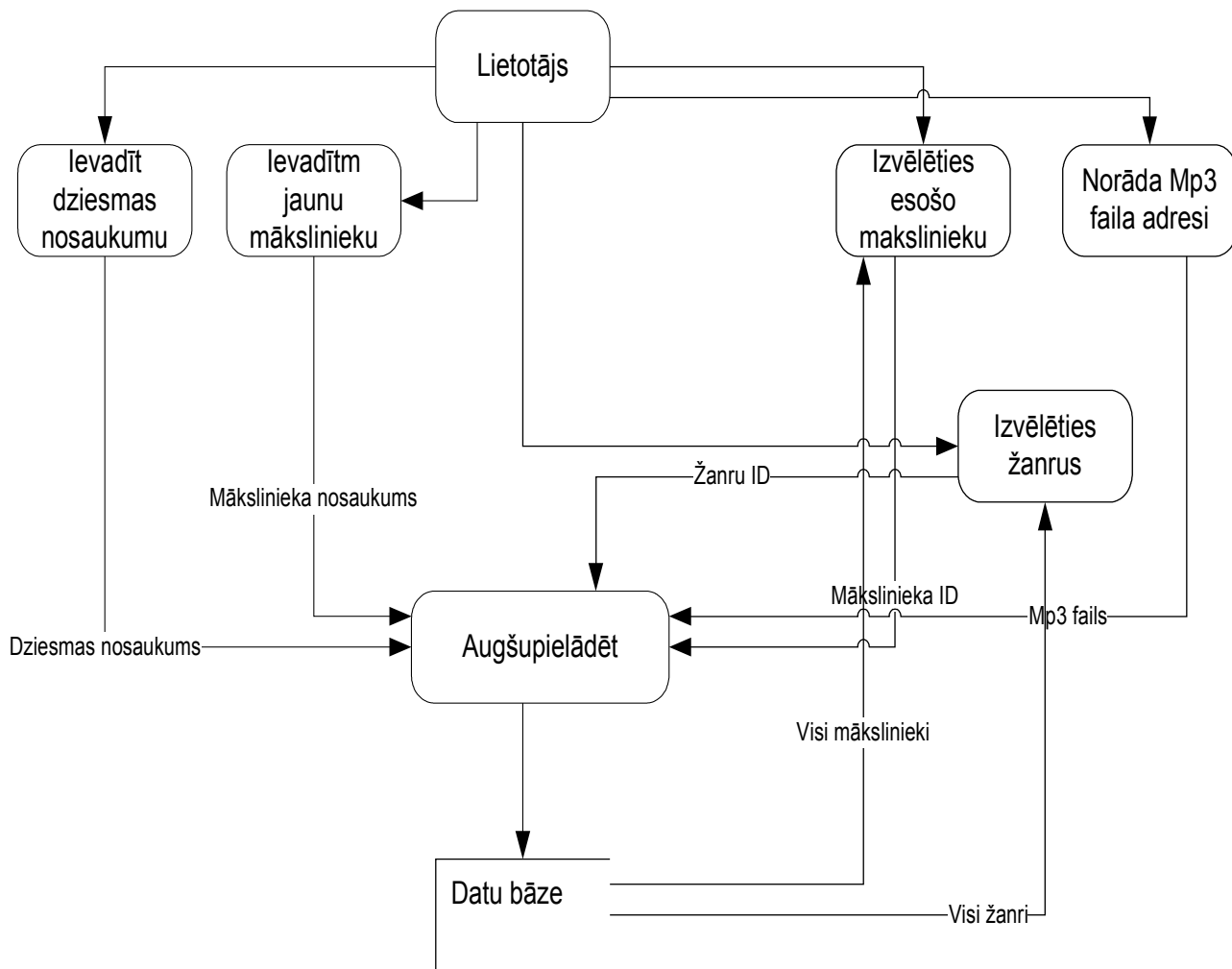
Att.4. Dziesmu saraksta veidošana



Att. 3 2.līmenis



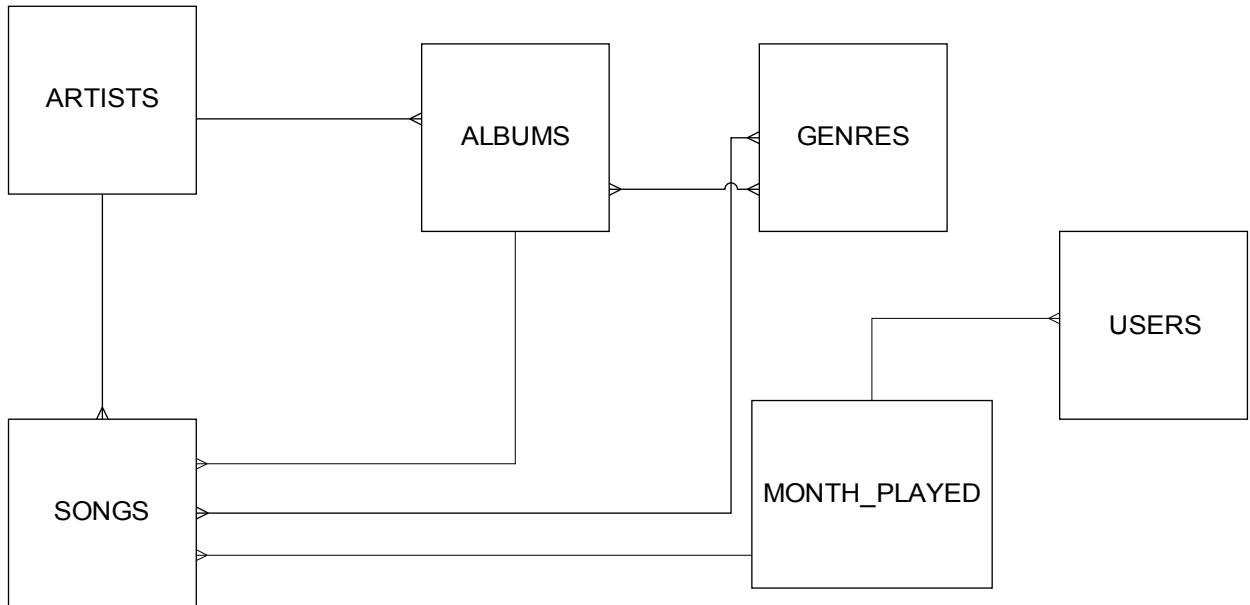
Att. 5 2.līmenis



3.3. ER modeļi

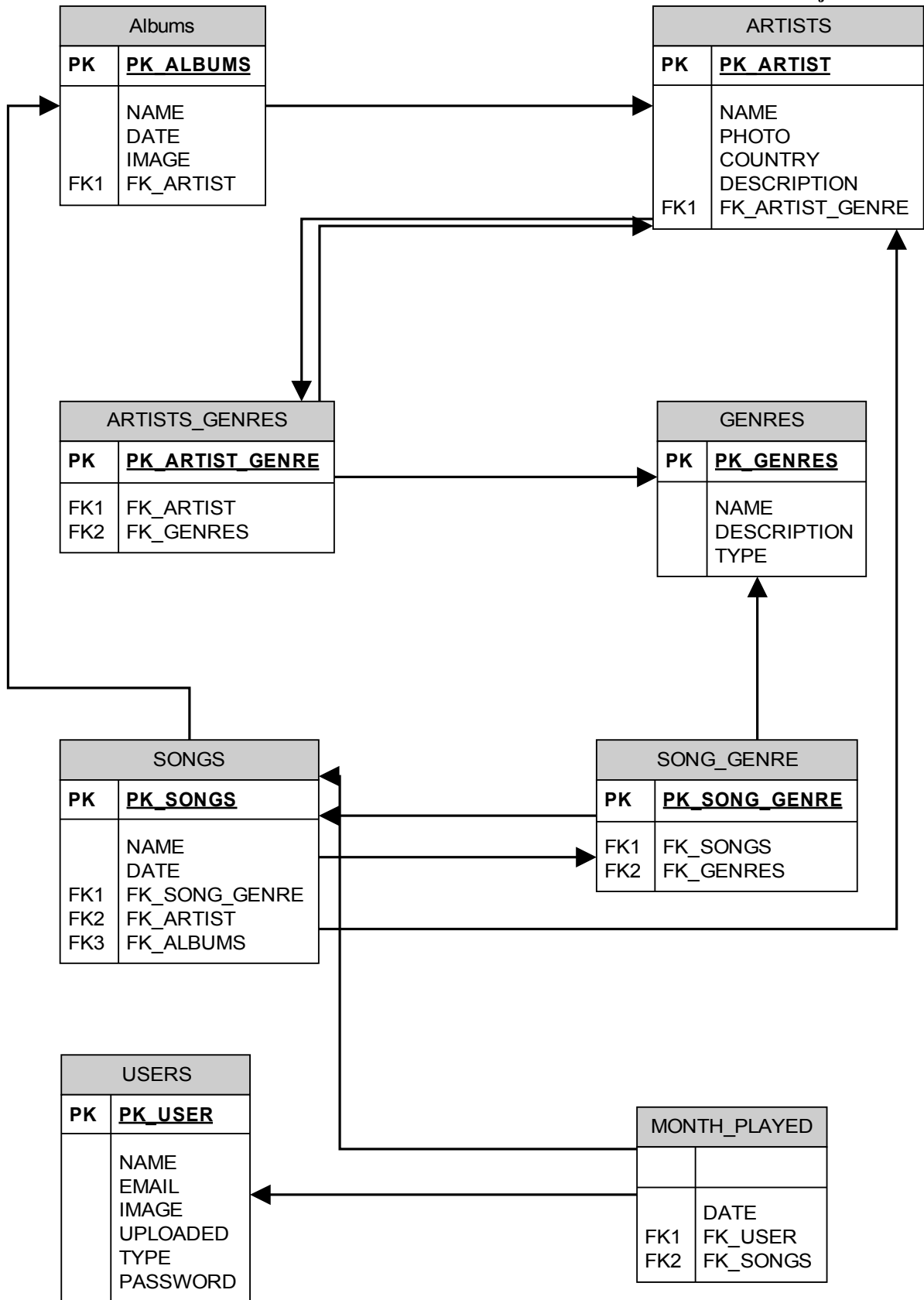
3.3.1. Konceptuālais ER modelis

Att. 6. konceptuālais ER modelis



3.3.2. Realizācijas ER modelis

Att. 7. Realizācijas ER modelis



Tabulas SONG_GENRE un ARTISTS_GENRES kalpo kā starptabulas, kas saista žanrus ar māksliniekiem un žanrus ar dziesmām. Tabulas SONG_GENRE un SONGS ir savienotas ar saiti, kam ir īpašība daudz pret daudzām, tādi pati saite ir starp tabulām ARTISTS_GENRES un ARTISTS.

3.3. Datu bāzes tabulu detalizētais apraksts

Šajā nodaļā ir aprakstītas programmas Interneta radio vortāls datu bāzes tabulas:

1. Tabula „ARTISTS”;
2. Tabula „ALBUMS”;
3. Tabula „SONGS”;
4. Tabula „GENRE”
5. Tabula „USERS”;
6. Tabula „MONTH_PLAED”;
7. Tabula „ARTISTS_GENRES”;
8. Tabula „SONG_GENRES”

3.4.1. Tabula „ARTISTS”

Tabulā glabājas dati par mūzikas izpildītājiem.

Tabula 5. „ARTISTS”

Kolonna	Datu tips	Obligāts	Primārs	Komentārs
PK_ARTIST	Int	Jā	Jā	Ieraksta identifikators
NAME	varchar(255)	Jā	Nē	Mūziķa vārds/nosaukums
PHOTO	blob	Nē	Nē	Izpildītāju foto
COUNTRY	varchar(255)	Nē	Nē	Izpildītāja valsts
FK_ARTIST_GENRE	Int	Nē	Nē	Saista ARTIST tabulu ar starptabulu ARTIST GENRE
DESCRIPTION	varchar(1000)	Nē	Nē	Apraksts par izpildītāju

3.4.2. Tabula „ALBUMS”

Tabulā glabājas dati par mākslinieku albūmu datiem.

Tabula 6. „ALBUMS”

Kolonna	Datu tips	Obligāts	Primārs	Komentārs
PK_ALBUM	Int	Jā	Jā	Ieraksta identifikators
FK_ARTIST	Int	Nē	Nē	Saista tabulas ALBUMS un ARTISTS

NAME	varchar(255)	Jā	Nē	Albūma nosaukums
IMAGE	blob	Nē	Nē	Albūma vāka bilde
DATE	date	Nē	Nē	Albūma izlaišanas datums

3.4.3. Tabula „SONGS”

Tabula glabā datus par dziesmām.

Tabula 7. „SONGS”

Kolonna	Datu tips	Obligāts	Primārs	Komentārs
PK_SONG	Int	Jā	Jā	Ieraksta identifikators
FK_ARTIST	Int	Nē	Nē	Saista tabulas SONGS un ARTIST
FK_ALBUM	Int	Nē	Nē	Saista tabulas SONGS un ALBUM
FK_SONG_GENRE	Int	Nē	Nē	Saista tabulas SONGS un SONG GENRE
NAME	varchar(255)	Jā	Nē	Dziesmas nosaukums
FILENAME	varchar(400)	Jā	Nē	Dziesmas glabāšanas adrese
DATE	date	Nē	Nē	Dziesmas ievietošana radio vortālā

3.4.4. Tabula „GENRE”

Tabula glabā datus par dziesmu žanriem.

Tabula 8. „GENRE”

Kolonna	Datu tips	Obligāts	Primārs	Komentārs
PK_GENRE	Int	Jā	Jā	Ieraksta identifikators
NAME	varchar(255)	Jā	Nē	Žanra nosaukums
DESCRIPTION	varchar(1000)	Nē	Nē	Žanra apraksts

3.4.5. Tabula „USERS”

Tabula glabā datus par vortāla lietotājiem

Tabula 9. „USERS”

Kolonna	Datu tips	Obligāts	Primārs	Komentārs
PK_USER	Int	Jā	Jā	Ieraksta identifikators
NAME	varchar(255)	Jā	Nē	Mūziķa vārds/nosaukums
IMAGE	blob	Nē	Nē	Izpildītāju foto
EMAIL	varchar(255)	Nē	Nē	Izpildītāja valsts
PASSWORD	varchar(255)	Jā	Nē	Lietotāju parole
TYPE	Int	Jā	Nē	Lietotāja grupas tips
UPLOADED	varchar(1000)	Nē	Nē	Apraksts par izpildītāju

3.4.6. Tabula „MONTH_PLAYED”.

Tabulā glabājas dati par atskaņoto dziesmu statistiku.

Tabula 10. „MONTH_PLAYED”

Kolonna	Datu tips	Obligāts	Primārs	Komentārs
---------	-----------	----------	---------	-----------

FK_SONG	Int	Jā	Nē	Saite ar tabulu SONG
FK_USER	Int	Nē	Nē	Saite ar tabulu USER
DATE	date	Jā	Nē	Dziesmas atskaņošanas datums
NUMBER	bigint	Nē	Nē	Dziesmas atskaņoto reižu skaits

3.4.7. Tabula „ARTISTS_GENRES”

Tabulā glabājas dati par saitēm starp tabulām ARTISTS un GENRE.

Tabula 11 „ARTISTS_GENRES”

Kolonna	Datu tips	Obligāts	Primārs	Komentārs
PK_ARTIST_GENRE	Int	Jā	Jā	Ieraksta identifikators
FK_ARTIST	Int	Jā	Nē	Saite ar tabulu ARTIST
FK_GENRE	Int	Jā	Nē	Saite ar tabulu GENRE

3.4.8. Tabula „SONG_GENRES”

Tabulā glabājas dati par saitēm starp tabulām SONGS un GENRE.

Tabula 12 „SONG_GENRES”

Kolonna	Datu tips	Obligāts	Primārs	Komentārs
PK_ARTIST_GENRE	Int	Jā	Jā	Ieraksta identifikators
FK_ARTIST	Int	Jā	Nē	Saite ar tabulu ARTIST
FK_GENRE	Int	Jā	Nē	Saite ar tabulu GENRE

3.4. Programmatūras moduļu projektējums

Šajā nodaļā tiks aprakstītas Interneta radio vortāla moduļi.

- Viens modulis ir interfeiss ar Flash plejeri.
- Dziesmas saraksta izveide, izmantojot drag and drop.
- Statistika izmantojam gatavo rīku, ar selectu no datu bāzes.
- Lietotāju apkalpošana.

3.5.1. Interfeiss ar Flash plejeri.

3.5.1.1. Nolūks

Datu savstarpējā apmaiņa starp PHP un Flash

3.5.1.2. Funkcijas

3.2.1.2.1. Dziesmu saraksta padošana uz Flash plejeri.

Dziesmu saraksts ar dziesmām, kuru izveidoja lietotājs. Katrai dziesmai no izveidotā saraksta tiek atlasīti dati par dziesmu un izpildītāju, tiek uzģenerēts XML fails ar struktūru.

Faila piemērs:

```
<?xml version="1.0" encoding="UTF-8"?>
<songs>
  <song url="mp3/black_eyed_peas_-_boom_boom_pow_2009.mp3" artist="Fergie"
  track="boom" id="1"/>
</songs>
```

Tabula 13. XML koda skaidrojums

XML tag	Apraksts
<?xml version="1.0" encoding="UTF-8"?>	Virsraksts un UTF-8 kodējums
<songs>	Saknes elements
<song url="mp3/black_eyed_peas_-_boom_boom_pow_2009.mp3" artist="Fergie" track="boom" id="1"/>	Konkrēta dziesma: url – faila nosaukums, atrašanās ceļš; artist – mākslinieka vārds; track- dziesmas nosaukums; id – dziesmas identifikators.

Ja lietotājs, kurš sastāda dziesmu sarakstu ir iegājis sistēmā ar savu vārdu un paroli, tad faila nosaukums būs „login”.xml. līdz ar to katram lietotājam, kas ir ielogojies sistēmā un sastādījis savu dziesmu sarakstu, būs savs individuālais dziesmu saraksts.

Ja lietotājs nav ielogojies sistēmā, tad faila nosaukums būs nejauši saģenerēts nosaukums.

Pēc šī faila uzģenerēšanas, faila nosaukums tiek padots Flash, katru reizi, kad mājaslapa tiek pārlādēta Flash plejeris, pieprasa dziesmu saraksta nosaukumu, šis nosaukums tiek ņemts no cookies. Tas tiek realizēts ar actionscript funkciju – sendAndLoad un tipu – LoadVars. Funkcijas parametros ir PHP skripts un LoadVars tipa mainīgais, kas saņem atpakaļ datus.

(actionscript – flash iebūvēta funkcija.). Dziesmas identifikators tiek sūtīts atpakaļ uz „incPlayedNumber.php” failu, kad tā ir bijusi noklausīta atbilstošo laiku, šo laiku uzstāda administrators. PHP skripts saņemot identifikatoru, pievieno ierakstu tabulā MONTH_PLAYED, kas satur tekošo datumu, lietotāju un dziesmas identifikatoru.

3.5.2. Dziesmas saraksta izveide, izmantojot drag and drop

3.5.2.1. Nolūks

Ļauj ērti izvēlēties no visām esošām dziesmām sev tīkamās dziesmas noklausīšanai.

3.5.2.2. Funkcijas

Lietotāju ērtībai vortālā ir divi tabi – Artist un Genre, ja lietotājs nezina kādu mākslinieku, bet vēlas noklausīties konkrētu mūzikas žanru, tad viņam ir iespēja ievietot visas dziesmas no konkrēta žanra savā dziesmu sarakstā. Katra taba saturs veidojas dinamiski, atlasot mākslinieku nosaukumus un žanru nosaukumus. Katrs atsevišķais elements ir kustināms, to var pārvilkt uz dziesmu sarakstu. Ja

mākslinieku atver ar dubultklikšķi, dubultklikšķa funkcija tiek realizēta ar Ajax palīdzību, katram artista elementam ir savs Java skripta funkcija showSongs, kas izveido Ajax pieprasījumu, kas izsauc PHP, PHP atlasa no datu bāzes visu šī mākslinieka dziesmu nosaukumus, tad mākslinieku saraksts tiek paplašināts, pie atbilstoša mākslinieka parādās tā mākslinieka dziesmu saraksts. Tas ļauj lietotājam izvēlēties mākslinieka atsevišķās dziesmas. Katra dziesma ir kustināms elements. Dziesmu saraksts ļauj ievietot sevī mākslinieku dziesmas un žanrus, kā arī atsevišķās dziesmas vienlaicīgi.

Lai izveidotu kustīgu sarakstu tiek izmantota Javascript bibliotēka – Scriptaculous, kurai ir iebūvēta klase Sortable.

Sortable klases izmantošanas piemērs:

```
Sortable.create('artists', {tag: 'div', dropOnEmpty: true, containment: ['playlist'], only: 'draggable_artists'}
```

Tabula 14. Sortable klases izmantošana

Parametrs	Apraksts
'artists'	ID konteiners, ir jābūt iepriekš nodefinēts elements ar šādu ID
Tag	Kura tag elementu vajag taisīt kustināmu
dropOnEmpty	Var pieņemt sevī kustināmus elementus
containment	Tiek padots konteineru elementu saraksts, kuru var pārvilkt uz esošo failu.
only	Parāda, kuras klases kustināms elements tas ir.

Nospiežot pogu Add ģenerē pirmo moduli no otrā moduļa.

Att. .8 Dziesmu saraksta izveide

[Login](#) [Register](#) [Statistics](#)

Artists **Genres**

Arash
Banderos
Fergie
Infiniti
Maksim
Vintazh

Playlist!

Add

3.5.3. Statistika

3.5.3.1. Nolūks

Lietotājam ir iespējams apskatīties statistiku pa mēnesi vai pa nedēļu, kā arī ielogotam lietotājam ir iespēja apskatīties savu klausītāko dziesmu statistiku.

3.5.3.2. Funkcijas

Grafiks atspoguļo piecas klausītākās dziesmas:

3.5.3.2.1. Ielogotā lietotāja statistika:

3.5.3.2.1.1. Pa mēnesi;

3.2.1.1.1. Pa māksliniekiem;

3.2.1.1.2. Pa dziesmām.

3.5.3.2.1.2. Pa nedēļu;

3.2.1.2.1. Pa māksliniekiem;

3.2.1.2.2. Pa dziesmām.

3.5.3.2.2. Neregistrētā lietotāja statistika:

3.5.3.2.2.1. Pa mēnesi;

3.5.3.2.2.1.1. Pa māksliniekiem;

3.5.3.2.2.1.2. Pa dziesmām.

3.5.3.2.2.2. Pa nedēļu;

3.5.3.2.2.2.1. Pa māksliniekiem;

3.5.3.2.2.2.2. Pa dziesmām.

Grafika izveidei tiks lietots XML/SWF Chart 5.08 rīks, kuram tiks padoti XML faili, ar sekojošo XML faila formātu:

```
<?xml version="1.0"?>
```

```

<chart>
  <chart_data>
    <row>
      <null/>
      <string>boom</string><string>ne_bojusj</string>
      <string>love</string><string>ados</string>
      <string>ne_otdam</string>
      <string>Ot_Moskvi</string>
    </row>
    <row>
      <string>Week</string>
      <number>46</number>
      <number>6</number>
      <number>6</number>
      <number>3</number>
      <number>3</number>
      <number>1</number>
    </row>
  </chart_data>
  <chart_label
    position="middle"
    color="FFFFFF"/>
  <series_color>
    <color>51ae1a</color>
  </series_color>
</chart>

```

Tabula 15. XML koda skaidrojums

XML tags	Apraksts
<?xml version="1.0"?>	Virsraksts un UTF-8 kodējums
<chart>	Saknes elements
<chart_data>	Dati, ar kuru palīdzību tiek veidots grafiks.
<row>	<chart_data> sastāvdaļa
<string>boom</string>	Grafika elementu (stabiņu) nosaukumi
<string>Week</string>	Grafika nosaukums
<number>46</number>	Atskaņotās dziesmas reižu skaits
<chart_label position="middle" color="FFFFFF"/>	Grafika datu vizuālā noformēšana: Position="middle", atskaņoto reižu skaits tiks ielikts grafika (stabiņa) vidū Color="FFFFFF", skaitļu krāsa

XML faili tiek pārveidoti SWF formātā. Lai šos datus uztaisītu un atspoguļotu tos mājas lapā, tad tiek izsaukta atbilstoša Javascript funkcija `createChart`, skatīt pielikumā „charts.js”. Funkcijas parametrs ir XML fails, no kura arī tiek parādīts grafiks. Līdz ar to funkciju `createChart` ir jāizmanto astoņas reizes, lai izveidotu atbilstošos grafikus, kā parametrus ir jāpadod sekojošos failu nosaukumus:

Tabula 16. Funkcijas numura un faila nosaukuma attiecība

Funkcijas numurs	Faila nosaukums
Pa mēnesi klausītākie mākslinieki lietotājam	loginArtistsMonthQuery.xml
Pa mēnesi klausītākās dziesmas lietotājam	loginSongsMonthQuery.xml
Pa nedēļu klausītākie mākslinieki lietotājam	loginArtistsWeekQuery.xml
Pa nedēļu klausītākās dziesmas lietotājam	loginSongsWeekQuery.xml

Pa mēnesi klausītākie mākslinieki kopsummā visiem lietotājiem	totalArtistsMonthQuery.xml
Pa mēnesi klausītākās dziesmas kopsummā visiem lietotājiem	totalSongsMonthQuery.xml
Pa nedēļu klausītākie mākslinieki kopsummā visiem lietotājiem	totalArtistsWeekQuery.xml
Pa nedēļu klausītākās dziesmas kopsummā visiem lietotājiem	totalSongsWeekQuery.xml

Datus XML failiem iegūst atlasot visbiežāk no tabulas MONTH_PLAYED un saistītām tabulām, grupējot pēc dziesmām vai māksliniekiem. XML fails tāpat kā Dziesmu saraksta XML tiek veidots ar DOM extension palīdzību.

Att. 9 Statistika



3.6. Lietotāju apkalpošana

3.6.1. Nolūks

Veikt pamatoperācijas ar lietotāju sistēmu, kā arī ierobežot dziesmu augšupielādi, jo tikai reģistrētie lietotāji drīkst augšupielādēt dziesmas.

3.6.2. Funkcijas

Lietotāju reģistrācija, ļauj lietotājam pierēģistrēties Interneta radio vortālam. Reģistrācijas laikā lietotājam ir iespēja augšupielādēt savu bildi, tādi pati funkcija ir realizēts arī dziesmas augšupielādēšana. Tiek izveidots tag, ar speciālu atribūtu - enctype="multipart/form-data.

Reģistrācijas laikā tiek pārbaudīts lietotāja vārda unikalitāte, kā arī lietotāja parole tiek šifrēta md5 formātā, tas tiek realizēts ar PHP funkciju md5, kura šifrē datus.

Ielogošanās, tiek meklēts lietotājs datu bāzē pēc lietotāja vārda, kā arī tiek meklēta arī parole, salīdzina ar ievadītiem datiem, ievadīto paroli uzreiz šifrē md5 formātā un šifrētā veidā tiek salīdzināts ar datu bāzē jau esošo paroli. Lietotāju vārdi datu bāzē tiek glabāti lieliem burtiem, tas nozīmē, kad notiek salīdzināšana, tiek salīdzināts lieliem burtiem. To izdara ar funkcijas strtoupper palīdzību. Lietotājam ir iespēja mainīt savu profilu, profila maiņa tiek realizēta ar tādu pašu funkciju palīdzību, kā pie lietotāja reģistrācijas.

Dziesmas pievienošana, lietotājs ievada dziesmas nosaukumu, ievada mākslinieku vai izvēlas no saraksta, ievada dziesmas glabāšanās adresi un nosūta datus tabulā SONGS, tiek izveidots viens ieraksts, toties starptabulā SONGS_GENRES tiek izveidots tik daudz ierakstu, cik lietotājs ir izvēlējis žanrus atbilstoši dziesmai.

Att. 10 Lietotāja reģistrācija



Name:
Password:
E-mail:
Picture:
(should be jpg and < 64kb)

[Show all registered users](#)

Att. 11 Ielogošanās



Name:
Password:

Att. 12 Jaunas dziesmas pievienošana

Name:

Artist:

Other artist:

Genre:

- Alternative
- Hard rock
- Indie rock
- Pop
- Progressive metal
- Rock

mp3 file:

4. Testēšana

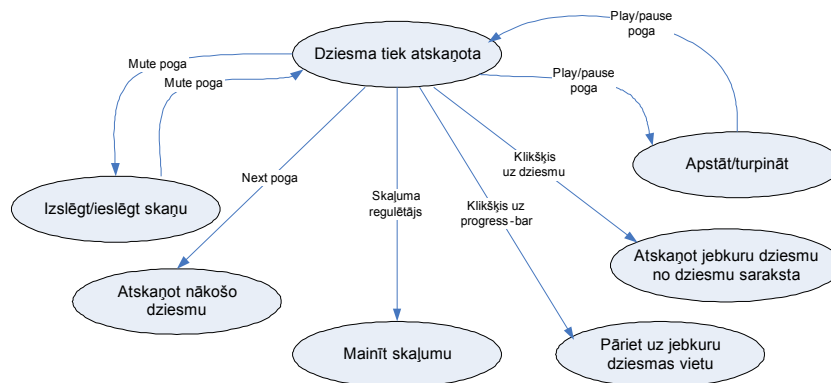
Web aplikāciju testēšana vispār ir ļoti sarežģīts uzdevums. Tīmekļa risinājumi pēc savas būtības ir ļoti neviendabīgi. Programmas komponentes darbojas uz dažādas aparatūras ar dažādām operētājsistēmām un vēl dažādās pārlūkprogrammās. Un arī izstrāde ir veikta izmantojot vairākas programēšanas valodas, apvienojot kopā dažādas tehnoloģijas, lietojot arī trešās puses izstrādātas komponentes.

Taču šajā darbā tika veikta funkcionāla jeb “melnās kastes” testēšana, tāpēc ir vērts abstrahēties no realizācijas.

4.1. Atskaņotājs:

Pielietosim modeļa testēšanas principus, mēģinot uzzīmēt grafu un tad veidot testpiemērus ar domu, lai notestētu katru virsotni, katru saiti un ciklus.

Att. 13 Atskaņotāja funkciju modelis



Tabula 17. Tests#1

Tests #1 Skaņas izslēgšanas/ieslēgšanas pārbaude			vidēji svarīgi dati
Ieejas dati (songs.xml)	Darbības	Rezultāts	Sagaidāmais rezultāts
<pre><?xml version="1.0" encoding="UTF-8"?> <songs> <song url="mp3/01.Sigur Ros - Takk.mp3" artist="Sigur Ros" track="Takk" id="13"/> </songs></pre>	1) Mute poga 2) Mute poga	1) Skaņa izslēdzas, dziesma turpinās. 2) Skaņa ieslēdzas.	1) Skaņa izslēdzas, dziesma turpinās. 2) Skaņa ieslēdzas.
Stauss: OK			

Tabula 18. Tests#2

Tests #2 Dziesmas apstāšana/turpināšana			svāriģi dati
Ieejas dati (songs.xml)	Darbības	Rezultāts	Sagaidāmais rezultāts
<pre><?xml version="1.0" encoding="UTF-8"?> <songs> <song url="mp3/01.</pre>	1) Play/pause poga 2) Play/pause poga	1) Dziesmas atskaņošana apstājas. 2) Atskaņošana	1) Dziesmas atskaņošana apstājas. 2) Atskaņošana turpinās no tās vietas, kur apstājas.

<pre>Sigur Ros - Takk.mp3" artist="Sigur Ros" track="Takk" id="13"/> </songs></pre>		turpinās no tās vietas, kur apstājās.	
Statuss: OK			

Tabula 19. Tests#3

Tests #3 Pāreja uz jebkuru dziesmas vietu			vidēji svarīgi dati
Ieejas dati (songs.xml)	Darbības	Rezultāts	Sagaidāmais rezultāts
<pre><?xml version="1.0" encoding="UTF-8"?> <songs> <song url="mp3/01. Sigur Ros - Takk.mp3" artist="Sigur Ros" track="Takk" id="13"/> </songs></pre>	1) Noklikšīnāt progressbara vidū. 2) Noklikšīnāt progressbara tūvāk dziesmas sākumam.	1) Dziesma atskaņojas no vietas, uz kuru uzklikšīnāja. 2) Dziesma atskaņojas no vietas, uz kuru uzklikšīnāja.	1) Dziesma atskaņojas no vietas, uz kuru uzklikšīnāja. 2) Dziesma atskaņojas no vietas, uz kuru uzklikšīnāja.
Statuss: OK			

Tabula 20. Tests#4

Tests #4 Atskaņot nākošo dziesmu			svāriģi dati
Ieejas dati (songs.xml)	Darbības	Rezultāts	Sagaidāmais rezultāts
<pre><?xml version="1.0" encoding="UTF-8"?> <songs> <song url="mp3/Brainstorm- Online.mp3" artist="Brainstorm" track="Online" id="10"/> <song url="mp3/01- lily_allen- the_fear.mp3" artist="Lily Allen" track="The Fear" id="2"/> </songs></pre>	1) Next poga 2) Next poga	1) Tiek atskaņota 2. dziesma. 2) Atskaņotājs pāriet atkal uz 1. dziesmu.	1) Tiek atskaņota 2. dziesma. 2) Atskaņotājs pāriet atkal uz 1. dziesmu.
Statuss: OK			

Tabula 21. Tests#5

Tests #5 Pāriet uz jebkuru dziesmu no dziesmu saraksta			svarīgi dati
Ieejas dati (songs.xml)	Darbības	Rezultāts	Sagaidāmais rezultāts
<pre><?xml version="1.0" encoding="UTF-8"?> <songs> <song url="mp3/Brainstorm- Online.mp3" artist="Brainstorm" track="Online" id="10"/> <song url="mp3/01- lily_allen- the_fear.mp3" artist="Lily Allen" track="The Fear" id="2"/> </songs></pre>	<ol style="list-style-type: none"> 1) Noklikšīnāt uz dziesmu "The Fear". 2) Noklikšīnāt uz dziesmu "Online". 3) Noklikšīnāt uz dziesmu "Online". 	<ol style="list-style-type: none"> 1) Tiek atskaņota dziesma "The Fear". 2) Pāriet uz dziesmu "Online". 3) Paliel uz dziesmas "Online", tikai sāk atskaņošanu no jauna. 	<ol style="list-style-type: none"> 1) Tiek atskaņota dziesma "The Fear". 2) Pāriet uz dziesmu "Online". 3) Paliel uz dziesmas "Online", tikai sāk atskaņošanu no jauna.
Statuss: OK			

Tabula 22. Tests #6

Tests #6 Skaļuma mainīšana			vidēji svarīgi dati
Ieejas dati (songs.xml)	Darbības	Rezultāts	Sagaidāmais rezultāts
<pre><?xml version="1.0" encoding="UTF-8"?> <songs> <song url="mp3/01. Sigur Ros - Takk.mp3" artist="Sigur Ros" track="Takk" id="13"/> </songs></pre>	<ol style="list-style-type: none"> 1) Samazināt skaļumu no sākotnēja. 2) Palielināt pāri sākotnējam. 	<ol style="list-style-type: none"> 1) Skaļums samazinās. 2) Skaļums krietni palielinās. 	<ol style="list-style-type: none"> 1) Skaļums samazinās. 2) Skaļums krietni palielinās.
Statuss: OK			

Tabula 23. Tests #7

Tests #7 Skaņas izslēgšanas/ieslēgšanas + dziesmas apstāšana/turpināšana			svarīgi dati
Ieejas dati (songs.xml)	Darbības	Rezultāts	Sagaidāmais rezultāts
<pre><?xml version="1.0" encoding="UTF-8"?> <songs> <song url="mp3/Brainstorm- Online.mp3" artist="Brainstorm" track="Online" id="10"/> </songs></pre>	<ol style="list-style-type: none"> 1) Mute poga 2) Play/pause poga 3) Play/pause poga 4) Mute poga 5) Play/pause poga 6) Mute poga 7) Play/pause poga 8) Mute poga 	<ol style="list-style-type: none"> 1) Skaņa izslēdzas, dziesma turpinās. 2) Pauze, tālāk nekas nenotiek. 3) Dziesma turpinās, skaņas nav. 4) Skaņa ieslēdzas, dziesma spēle. 5) Pauze. 6) Skaņa izslēdzas, dziesma nespēle. 	<ol style="list-style-type: none"> 1) Skaņa izslēdzas, dziesma turpinās. 2) Pauze, tālāk nekas nenotiek. 3) Dziesma turpinās, skaņas nav. 4) Skaņa ieslēdzas, dziesma spēle. 5) Pauze. 6) Skaņa izslēdzas, dziesma nespēle.

		dziesma nespēle. 7) Dziesma turpinās, skaņas nav. 8) Skaņa ieslēdzas.	7) Dziesma turpinās, skaņas nav. 8) Skaņa ieslēdzas.
Statuss: OK			

Tabula 22. Tests #8

Tests #8 Skaņas izslēgšanas/ieslēgšanas + nākošā dziesma			vidēji svarīgi dati
Ieejas dati (songs.xml)	Darbības	Rezultāts	Sagaidāmais rezultāts
<pre><?xml version="1.0" encoding="UTF-8"?> <songs> <song url="mp3/Brainstorm- Online.mp3" artist="Brainstorm" track="Online" id="10"/> <song url="mp3/01- lily_allen- the_fear.mp3" artist="Lily Allen" track="The Fear" id="2"/> </songs></pre>	<ol style="list-style-type: none"> 1) Mute poga 2) Next poga 3) Mute poga 	<ol style="list-style-type: none"> 1) Skaņa izslēdzas. 2) Atskaņotājs pāriet uz 2. dziesmu, skaņa ieslēdzas. 3) Skaņa izslēdzas. 	<ol style="list-style-type: none"> 1) Skaņa izslēdzas. 2) Atskaņotājs pāriet uz 2. dziesmu, skaņa paliek izslēgta. 3) Skaņa ieslēdzas.
Statuss: Kļūda			

Tabula 23. Tests #9

Tests #9 Skaņas izslēgšanas/ieslēgšanas + uz jebkuru dziesmu no dziesmu saraksta			Vidēji svarīgi dati
Ieejas dati (songs.xml)	Darbības	Rezultāts	Sagaidāmais rezultāts
<pre><?xml version="1.0" encoding="UTF-8"?> <songs> <song url="mp3/01. Sigur Ros - Takk.mp3" artist="Sigur Ros" track="Takk" id="13"/> </songs></pre>	<ol style="list-style-type: none"> 1) Mute poga. 2) Noklikšīnāt uz to pašu vienīgo dziesmu. 3) Mute poga 	<ol style="list-style-type: none"> 1) Skaņa izslēdzas. 2) Dziesme ir spēlēta no sākuma ar skaņu. 3) Skaņa izslēdzas. 	<ol style="list-style-type: none"> 1) Skaņa izslēdzas. 2) Dziesmas atskaņošana sāka no sākuma bez skaņas. 3) Skaņa ieslēdzas.
Statuss: Kļūda			

Tabula 24. Tests #10

Tests #10 Dziesmas apstāšana/turpināšana + nākošā dziesma			vidēji svarīgi dati
Ieejas dati (songs.xml)	Darbības	Rezultāts	Sagaidāmais rezultāts
<pre><?xml version="1.0" encoding="UTF-8"?> <songs> <song url="mp3/Brainstorm- Online.mp3" artist="Brainstorm" track="Online" id="10"/> <song url="mp3/01- lily_allen- the_fear.mp3" artist="Lily Allen" track="The Fear" id="2"/> </songs></pre>	<ol style="list-style-type: none"> 1) Play/pause poga 2) Next poga 3) Play/pause poga 4) Play/pause poga 5) Next poga 	<ol style="list-style-type: none"> 1) Pauze. 2) Atskaņotājs pāriet uz 2. dziesmas sākumu. 3) Pauze. 4) Atskaņošanas turpināšana. 5) Atgriežas uz 1. dziesmu. 	<ol style="list-style-type: none"> 1) Pauze. 2) Atskaņotājs pāriet uz 2. dziesmas sākumu. 3) Pauze. 4) Atskaņošanas turpināšana. 5) Atgriežas uz 1. dziesmu.
Statuss: OK			

Tabula 25. Tests #11

Tests #11 Skaļuma mainīšana + nākošā dziesma			vidēji svarīgi dati
Ieejas dati (songs.xml)	Darbības	Rezultāts	Sagaidāmais rezultāts
<pre><?xml version="1.0" encoding="UTF-8"?> <songs> <song url="mp3/Brainstorm- Online.mp3" artist="Brainstorm" track="Online" id="10"/> <song url="mp3/01- lily_allen- the_fear.mp3" artist="Lily Allen" track="The Fear" id="2"/> </songs></pre>	<ol style="list-style-type: none"> 1) Samazināt skaļumu ar regulētāju 2) Next poga 	<ol style="list-style-type: none"> 1) Skaņa samazinās. 2) Atskaņo nākošo dziesmu ar uzstādīto skaļumu. 	<ol style="list-style-type: none"> 1) Skaņa samazinās. 2) Atskaņo nākošo dziesmu ar uzstādīto skaļumu.
Statuss: OK			

Tabula 26. Tests #12

Tests #12 Skaļuma mainīšana + pāreja uz jebkuru dziesmas vietu			vidēji svarīgi dati
Ieejas dati (songs.xml)	Darbības	Rezultāts	Sagaidāmais rezultāts
<pre><?xml version="1.0" encoding="UTF-8"?> <songs> <song url="mp3/01. Sigur Ros - Takk.mp3" artist="Sigur Ros" track="Takk" id="13"/> </songs></pre>	1) Palielinām skaļumu ar regulētāju. 2) Noklikšinām progressbara vidū.	1) Skaņa palielinās. 2) Atskaņo dziesmu no vidum ar uzstādīto skaļumu.	1) Skaņa palielinās. 2) Atskaņo dziesmu no vidum ar uzstādīto skaļumu.
Statuss: OK			

Tabula 27. Tests #13

Tests #13 Skaņas izslēgšanas/ieslēgšanas + pāreja uz jebkuru dziesmas vietu			vidēji svarīgi dati
Ieejas dati (songs.xml)	Darbības	Rezultāts	Sagaidāmais rezultāts
<pre><?xml version="1.0" encoding="UTF-8"?> <songs> <song url="mp3/01. Sigur Ros - Takk.mp3" artist="Sigur Ros" track="Takk" id="13"/> </songs></pre>	1) Mute poga 2) Noklikšinām progressbara vidū.	1) Skaņa izslēdzas. 2) Atskaņo dziesmu no vidum ar skaņu.	1) Skaņa izslēdzas. 2) Pāriet uz dziesmas vidu, atskaņo, skaņas nav.
Statuss: OK			

Rezultāts. No 13 testpiemēriem 2 beidzas nepareizi. Abi kļūdaini testi (#7 un #8) attiecas uz vienu lietu. Acīmredzot kļūdas ir tāpēc, ka skaņas izslēgšana darbojas tikai uz tekošo dziesmu. Vajadzētu, lai tā būtu izslēgta visu laiku kāmēr lietotājs atkal to neieslēgs.

4.2. Lietotāju sistēma

Tabula 28. Tests#14

Tests #14 Jauna lietotāja reģistrācija ar pareiziem ieejas datiem		svarīgi dati
Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Homer" Password = "12345" E-mail = "homer.simpson@fox.com" Picture = "C:\Documents and Settings\Administrator\My Documents\avatar.jpg" (attēla fails aizņem 30 kB, kas ir	Parādās paziņojums "You are registered!". Jauna lietotāja dati tiek saglabāti datubāzē.	Lietotāja informācija tiek saglabāta datubāzē.

mazāk nekā maksimāli atļauti 64 kB)		
Statuss: OK		

Tabula 29. Tests#15

Tests #15 Jauna lietotāja reģistrācija ar jau reģistrēto lietotāja vārdu svarīgi dati		
Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Homer" Password = "Trojan" E-mail = "homer@hellas.gr"	Parādās paziņojums "User with this Login already exists. Please choose another one". Reģistrācija beidzas neveiksmīgi.	Kļūdas paziņojums par to, ka tāds vārds jau ir reģistrēts. Datubāzē nekas netiek ierakstīts.
Statuss: OK		

Tabula 30. Tests#16

Tests #16 Reģistrācija ar attēlu, lielāku par maksimāli atļautu mazsvarīgi dati		
Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Rino" Password = "Shredder" Picture = "/home/share/r.gif" (attēla fails aizņem 30 kB, kas ir vairāk nekā maksimāli atļauti 64 kB)	Parādās paziņojums "Only images under 64Kb are accepted for upload! You are registered!". Lietotājs ir izveidots ar tukšo attēla lauku.	Kļūdas paziņojums par to, ka attēls ir pārāk liels. Datubāzē nekas netiek ierakstīts.
Statuss: Kļūda		

Tabula 31. Tests#17

Tests #17 Pareizā ielogošana svarīgi dati		
Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Homer" Password = "12345"	Lietotājs pārvietots galvenajā lapā, kura jau parādās, ka viņš esot ielogojies vortālā.	Veiksmīga ielogošana un rezultāta pāriet galvenajā lapā.
Statuss: OK		

Tabula 32. Tests#18

Tests #18 Nepareiza parole pie ielogošanas svarīgi dati		
Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Homer" Password = "It's me"	Parādās paziņojums "Login or password is incorrect". Paliek ielogošanas lapā.	Kļūdas paziņojums par nepareizo paroli. Piedāvāt ievadīt vēlreiz.
Statuss: OK		

Tabula 33. Tests#19

Tests #19 Neeksistējošais lietotāja vārds pie ielogošanas svarīgi dati		
---	--	--

Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Qwrty" Password = "qwerty"	Parādās paziņojums "Login or password is incorrect". Paliek ielogošanas lapā.	Kļūdas paziņojums par neeksistējošu lietotāja vārdu. Piedāvāt ievadīt vēlreiz.
Statuss: OK		

Tabula 34. Tests#20

Tests #20 Reģistrācija bez paroles		svāriġi dati
Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Cesc" Password = "" E-mail = "fabregas@arsenal.com"	Parādās paziņojums "You are registered! ". Reģistrācija notika.	Lietotāja informācija tiek saglabāta datubāzē.
Statuss: OK		

Tabula 35. Tests#21

Tests #21 Lietotāja ielogošana, kuram ir tukša parole, neievadot paroli		svāriġi dati
Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Cesc" Password = ""	Ielogošana notika, pārvietots uz galveno lapu	Veiksmīga ielogošana un rezultāta pāriet galvenajā lapā.
Statuss: OK		

Tabula 36. Tests#22

Tests #22 Lietotāja ielogošana, kuram ir tukša parole, ievadot kādu paroli		svāriġi dati
Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Cesc" Password = "4"	Parādās paziņojums "Login or password is incorrect". Paliek ielogošanas lapā.	Kļūdas paziņojums par nepareizo paroli. Piedāvāt ievadīt vēlreiz.
Statuss: OK		

Rezultāti: No 9 testpiemēriem 1 atrada kļūdu. Tests #16 atklāja to, ka ja attēla faila izmērs ir lielāks par maksimāli pieļaujamo, tad kaut arī kļūdas paziņojums parādās, skriptā darbība netiek beigta un lietotājs tiek reģistrēts bez attēla.

4.3. Dziesmas pievienošana

Tabula 37. Tests#23

Tests #23 Izpildītājs ir izvēlēts no saraksta		vidēji svāriġi dati
Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Gong"; Artist = "Sigur Ros" (izvēlēts no saraksta); Genre – neviens nav izvēlēts;	Parādās paziņojums "Song is added. Thank you! ". Dziesma ir augšpielādēta uz servera un atrodas mp3/Sigur Ros [Takk]	Dziesma ir augšpielādēta un pievienots ieraksts datubāzē.

mp3 file = "C:\Documents and Settings\Administrator\My Documents\My Music\Sigur Ros [Takk] 08 Gong.mp3"	08 Gong.mp3. Dziesma ir pievienota datubāzei.	
Statuss: OK		

Tabula 38. Tests#24

Tests #24 Jauns izpildītājs		svarīgi dati
Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Bahama_Mama"; Other artist = "Boney M"; Genre – neviens nav izvēlēts; mp3 file = "C:\Documents and Settings\Administrator\My Documents\My Music\Boney_M-Bahama_Mama.mp3"	Parādās paziņojums "Song is added. Thank you! ". Dziesma ir augšpielādēta uz servera un atrodas mp3/Bahama_Mama.mp3 . Dziesma ir pievienota datubāzei. Tabulā ARTISTS ir izveidots artists "Boney M".	Dziesma ir augšpielādēta un pievienots ieraksts datubāzē. Izveidots jauns izpildītājs ar vārdu "Boney M".
Statuss: OK		

Tabula 39. Tests#25

Tests #25 Bez dziesmas nosaukuma		vidēji svarīgi dati
Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = ""; Other artist = "Garbage"; Genre – neviens nav izvēlēts; mp3 file = "D:\My Documents\My Music\garbage\Garbage - Milk.mp3"	Parādās paziņojums "Enter song name please!". Dziesma netika pievienota.	Kļūdas paziņojums, ka dziesmas nosaukums ir obligāts. Piedāvāts ievadīt un pievienot vēlreiz.
Statuss: OK		

Tabula 40. Tests#26

Tests #26 Bez izpildītāja		vidēji svarīgi dati
Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Boys"; Artsit = ""; Other artist = ""; Genre – neviens nav izvēlēts; mp3 file = "D:\My Documents\My Music\04 boys.mp3"	Parādās paziņojums "Choose artist or enter artist name, please!". Dziesma netika pievienota.	Kļūdas paziņojums, ir jābūt izpildītājs. Piedāvāts izvēlēties vai ievadīt un pievienot vēlreiz.
Statuss: OK		

Tabula 41. Tests#27

Tests #27 Dziesmas pievienošana ar norādītiem žanriem	svarīgi dati
--	---------------------

Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Choo Choo"; Artsit = "Arctic Monkeys"; Genre – ir izvēlēti "Indie rock" un "Pop" žanri; mp3 file = "D:\My Documents\My Music\Arctic Monkeys\05 - Choo Choo.mp3"	Parādās paziņojums "Song is added. Thank you! ". Dziesma pievienota. Tabulai SONGS_GENRES pievienoti divi ieraksti ar attiecīgo žanru identifikatoriem un jaunās dziesmas identifikatoru.	Dziesma ir augšpielādēta un pievienots ieraksts datubāzē. Saglabāta informācija par visiem šīs dziesmas žanriem.
Stauss: OK		

Tabula 42. Tests#28

Tests #28 Gan izvēlēts izpildītājs, gan ierakstīts kā jauns		vidēji svarīgi dati
Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Grace"; Artist = "Brainstorm"; Other artist = "Apocalyptica"; Genre – neviens nav izvēlēts; mp3 file = "D:\My Documents\My Music\2007 - Worlds Collide\02-Grace.mp3"	Parādās paziņojums "Song is added. Thank you! ". Dziesma ir augšpielādēta uz servera un atrodas mp3/ 02-Grace.mp3. Kā izpildītājs ir paņemts Apocalyptica, kurš arī ir izveidots.	Dziesma ir augšpielādēta un pievienots ieraksts datubāzē. Kā izpildītājs tiek uztvērts jauns izpildītājs, kurš arī automātiski ir izveidots un pievienots artistu tabulā.
Stauss: OK		

Tabula 43. Tests#29

Tests #29 Faila izmērs lielāks par maksimāli atļautu		svāriigi dati
Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Jesus Dod"; Artist = ""; Other artist = "Burzum"; Genre – neviens nav izvēlēts; mp3 file = "C:\Documents and Settings\Administrator\My Documents\My Music\Filosofem.part1\02 Jesus Dod.mp3" (Faila izmērs ir 19,7 MB)	Dziesma netiek pievienota un augšpielādēta, bet nekāda kļūdas paziņojuma nav, vienkārši tukšs logs.	Kļūdas paziņojums par to, ka faila izmērs pārsniedz maksimāli pieļaujamo. Dziesma netiek augšpielādēta.
Stauss: Kļūda		

Tabula 44. Tests#30

Tests #30 Fails neesot mp3	svāriigi dati
----------------------------	---------------

Ieejas dati	Rezultāts	Sagaidāmais rezultāts
Name = "Jesus Dod"; Artist = ""; Other artist = "Burzum"; Genre – neviens nav izvēlēts; mp3 file = "C:\Documents and Settings\Administrator\My Documents\2a.pdf"	Parādās paziņojums "Song file should be mp3 format!". Dziesma netiek pievienota un augšupielādēta.	Kļūdas paziņojums par to, ka faila paplašinājums neesot mp3. Dziesma netiek augšupielādēta.
Statuss: OK		

Rezultāti. No 8 testiem 1 ir neveiksmīgs. Kļūdains ir apstrādāts maksimālais faila izmērs.

Testēšana nebūt nav pilnīga un pietiekoša. Tā ir veidota, lai ar iespējami mazāku resursu patēriņu un testpiemēru skaitu pārbaudīt pamatfunkcijas. Testēšanas nolūkiem tika veidota atsevišķa datubāze, lai nesabojātu esošus datus.

5. Projekta organizācija

Interneta radio vortāls tika izstrādāts, kā mācību projekts, kas tika izstrādāts ar mērķi uzrakstīt un aizstāvēt šo projektu kā kvalifikācijas darbu, pirmā līmeņa profesionālās augstākās izglītības „Programmēšanas un datortīklu administrēšana” studiju programmas ietvaros.

Interneta radio vortāls ir virtuāls projekts, kas ļauj lietotājiem pašiem sastādīt savu dziesmu sarakstu, kā arī klausīties populāras vortāla dziesmas. Izstrādājot vortālu, tika izmantota programmatūras prasību inženierijas prakse.

Projekta izstrādes sākumā tika izstrādāta programmatūras prasību specifikācijas (PPS). Paralēli PPS izstrādei tika uzsākta programmatūras projektējuma apraksts. Šādā veidā tika atvieglota projekta kodēšana, jo uzsākot kodēšanu projektam jau bija izstrādāts datu bāzes modelis, kā arī tika izplānotas datu plūsmas projektā.

Projekta kodēšana tika sadalīta divās daļās. Sākumā tika izstrādātas visas nepieciešamās radio vortāla funkcijas. Tika izstrādāts HTML kods un izstrādāti PHP skripti.

Otrajā daļā tika izstrādāts interneta radio vortāla dizains, ar HTML un CSS palīdzību. Kodēšanas laikā pēc katras nodaļas izstrādes, programmatūrai tika veikta vienībtestēšana. Pēc kodēšanas darba pabeigšanas tika testēta visa sistēma un izstrādāta Programmatūras testēšanas dokumentācija.

Projekts sastāv no projekta dokumentācijas izstrādātāja, testētāja, programmētāja un projekta vadītāja. Projekta vadītājs šajā projektā bija kvalifikācijas darba vadītājs, bet projekta pārējās funkcijas veica kvalifikācijas darba izstrādātāja.

6. Kvalitātes nodrošināšana

Lai nodrošinātu programmatūras kvalitāti tika veikta dokumentācija, kas rakstīta pēc vispārējās prakses un atbilstoši standartiem. Tika izstrādātas Programmatūras prasību specifikāciju, Programmatūras projektējuma apraksts, kur ir atsauksme uz konkrētām prasībām, kā arī Programmatūras testēšanas dokumentācija. Visi šie dokumenti tika izstrādāti atbilstoši Latvijas Nacionālās standartizācijas un metroloģijas centra izstrādātiem standartiem.

Programmas izstrādes laikā kodam regulāri tika pievienoti komentāri.

Lai nodrošinātu kvalitātei atbilstošo noteikumu izpildi, projekta vadītājs kontrolē koda izstrādi.

7. Konfigurāciju pārvaldība

Programmas koda izstrāde pārsvarā tika veikta php valodā. Atskaņotājs ir paštaisīts Macromedia Flash programmā Actionscript valodā. Projektam tika izveidots repozitorijs. Kā versiju pārvaldības programma tika izmantots rīks TortoiseSVN. Kad kaut kas strādājošs tika uzprogrammēts, tā uzreiz izmaiņas tika ieliktas šajā repozitorijā. Tas ir aktuāls tīmekļa projektos, kad projekts satur ļoti daudz savstarpēji saistītu failu. Mainot vienu, mainās citi. Ar SVN uzreiz pirms augšupielādēšanas ir redzami visi faili, kas ir mainīti un tieši izmaiņas kodā. Tas nodrošināja arī rezerves kopēšanu. Dažreiz ļoti noderēja, ka var vienkārši atgriezties pie failu iepriekšējās versijas. Tā kā datubāzes skripti bija izveidoti pirmajā kārtā tos nācās mainīt. Tiem nolūkiem arī sql skripti tika visu laiku saglabāti uz repozitorijā atsevišķos failos ar unikāliem kārtas numuriem.

8. Darbietilpības novērtējums.

Darbietilpības novērtēšanai tiks pielietotas COCOMO Intermediate formulas. Tā ir vieglāka par funkcijpunktu saskaitīšanas metodi un tajā pašā laikā precīzāka nekā COCOMO Basic formula.

Vispirms ir jāsaskaita koda rindiņu skaits. Programmas koda rindu skaits ir 1181 rindas php+html+javascript koda, kā arī 299 rindu garš actionscripta plejeris. Kopā 1480. Pārveidojot rindu tūkstošos 1480 rindas (angl. *Lines Of Code* jeb *LOC*) = 1.48 KLOC.

Tad ir jāaprēķina projekta sarežģītības koeficientu EAF (angl. *Effort Adjustment Factor*).

Lai to aprēķinātu ir jānovērtē kādi atribūti, un jādabū koeficienti.

Tabula ir tikai tie atribūti, kuru vērtība ir atšķirīga no nominālas vērtības Vidēji un koeficienta 1.

Tabula 44. Darbietilpības novērtējums

Atribūts	Novērtējums	Koeficients
Datubāzes izmērs	Mazs	0.94
Produkta sarežģītība	Augsta	1.15
Programmatūras izstrādes pieredze	Maza	1.13
Programmēšanas valodas pārzināšana	Maza	1.07
Izstrādes rīku izmantošana	Augsta	0.91

$$EAF = 0.94 * 1.15 * 1.13 * 1.07 * 0.91 = 1.19$$

Kā projekta tips visvairāk šim projektam atbilst Organic, jo kaut bija sarežģītības ar flash plejera un pārejas sistēmas integrāciju, projekts ir samērā mazs, izstrādāja viens cilvēks.

Tāpēc koeficienti $a_i = 3.2$ un $b_i = 1.05$.

Tagad varam aprēķināt izstrādei nepieciešamo laiku personmēnešos:

$$E = a_i * (KLOC)^{b_i} * EAF = 3.2 * (1.48)^{1.05} * 1.19 = 3.2 * 1.51 * 1.19 = 5.75$$

Un beidzot, plānotais izstrādes laiks mēnešos:

$$D = c_b * (E)^{d_b} = 2.5 * (5.75)^{0.38} = 4.86$$

Uz rezultātiem var skatīties nedaudz skeptiski, jo visas darbietilpības prognozēšanas formulas praktiski nav pielietojamas maziem projektiem.

Projekta reālais uzsākšanas laiks ir 3. marts, projekts tika regulāri izstrādāts, labots un papildināts.

2. 9. Izmantotā literatūra

1. Кузнецов М., Симдянов И., Голышев С. РНР 5 на примерах, «БХВ-Петербург», Санкт-Петербург 2006 – 573 lpp.
2. Мерсер Д., Кент А., Новицки С., Скуайер Д., Чой В., Мерсер Д. РНР 5 для начинающих «Диалектика», Санкт-Петербург, Киев 2006 – 847 lpp.
3. Соколов С.А. JavaScript в примерах типовых решениях и задачах, «Вильямс», Москва, Санкт-Петербург, Киев 2006 – 589 lpp.
4. Cocomo aprēķināšanas metodes apraksts <http://en.wikipedia.org/wiki/COCOMO>
5. Web aplikācijas testēšanas apraksts <http://www.springerlink.com/content/amu2u6k27rktkxby/>
6. Specializēto vārdu tulkojumi <http://www.termini.lv>

Pielikums

Koda piemēri

Mp3 atskaņotāja izstrādes kods

```
//Setup sound object
var s:Sound = new Sound();
var totalDuration="00:00";
var cps:Number = -1; //Currently playing song
var totalPlayedSecs = 0;
var loaderWidth = 237.9;
var timeToSave:Number = -1; //how much seconds to play before save as played
var filename:String;

var loadVars_out:LoadVars = new LoadVars();
var loadVars_in:LoadVars = new LoadVars();
var xmlFilename_in:LoadVars = new LoadVars();

var loadedInterval = setInterval(loadedStatus,100);
var amountLoaded:Number;

function loadedStatus() {
    amountLoaded = s.position / s.duration;
    loader.loadbar._width = amountLoaded * loaderWidth;
}

s.onSoundComplete = playNext;
s.setVolume(vc.ratio);

//Array of songs (from xml)
var sa:Array = new Array();

//Position of music
var pos:Number;

//Load the songs XML
var xml:XML = new XML();
xml.ignoreWhite = true;
loadVars_in.onLoad = function()
{
    timeToSave = loadVars_in.whenAdd;
}
xml.onLoad = function()
{
    var nodes:Array = this.firstChild.childNodes;
    for(var i=0;i<nodes.length;i++)
    {
        sa.push(new Song(nodes[i].attributes.url, nodes[i].attributes.artist,
            nodes[i].attributes.track, nodes[i].attributes.id));
        //(label, data)
    }
}
```

```

        playlist.addItem(nodes[i].attributes.artist+" - "+nodes[i].attributes.track, i);
    }

    loadVars_out.sendAndLoad("getSettings.php", loadVars_in, "POST");
    playSong(cps);
}

xmlFilename_in.onLoad = function()
{
    filename = xmlFilename_in.filename;
}
//getting xml filename from php into xmlFilename_in
loadVars_out.sendAndLoad("getXMLName.php", xmlFilename_in, "POST");
xml.load(filename);

function duration () {
    timed = s.duration/1000;
    mind = Math.floor(timed/60);
    mind = (mind<10) ? "0"+mind : mind;
    secd = Math.floor(timed%60);
    secd = (secd<10) ? "0"+secd : secd;
    totalDuration = mind+":"+secd;
}

function timer(s) {
    time = s.position/1000;
    totalPlayedSecs = totalPlayedSecs + 1;
    if (totalPlayedSecs == timeToSave) //it's time to save song to db
    {
        loadVars_out.songId = sa[cps].id;
        trace(sa[cps].id);
        loadVars_out.sendAndLoad("incPlayedNumber.php", loadVars_in, "POST");
    }
    min = Math.floor(time/60);
    min = (min<10) ? "0"+min : min;
    sec = Math.floor(time%60);
    sec = (sec<10) ? "0"+sec : sec;
    time_txt.text = min+":"+sec+"/"+totalDuration;
}

//Play the MP3 File
function playSong(num):Void
{
    clearInterval(timeInterval);
    totalPlayedSecs = 0;
    s = new Sound();

    time_txt.text = "00:00/00:00";
    s.onSoundComplete = function() {

```

```

        if (timeToSave == -1) //not configured from php
        {
            loadVars_out.songId = sa[cps].id;
            trace(sa[cps].id);
            loadVars_out.sendAndLoad("incPlayedNumber.php", loadVars_in, "POST");
        }

        playNext();}
s.onLoad = duration;
s.setVolume(vc.ratio);

mute.gotoAndStop('on');
if(num == sa.length-1) //the last song in list
{
    cps=num=0;
    s.loadSound(sa[num].earl,true);
}
else
{
    cps=num;
    s.loadSound(sa[++cps].earl,true);
}

playList.selectedIndex = cps;
timeInterval = setInterval(timer, 1000, s);
trackInfo.text = sa[cps].artist+' - '+sa[cps].track;
playPause.gotoAndStop('pause');

}

function playNext(): Void
{
    playSong(cps);
}

//Music controls
//Pauses the music
function pauseIt():Void
{
    pos = s.position;
    s.stop();
    clearInterval(timeInterval);
}

//unPauses the music
function unPauseIt():Void
{
    timeInterval = setInterval(timer, 1000, s);
    s.start(pos/1000);
}

```

```

//Play/Pause Toggle
playPause.onRollOver = function()
{
  if(this._currentframe == 1)
    this.gotoAndStop('pauseOver');
  else
    this.gotoAndStop('playOver');
}

playPause.onRollOut = playPause.onReleaseOutside = function()
{
  if(this._currentframe == 10)
    this.gotoAndStop('pause');
  else
    this.gotoAndStop('play');
}

playPause.onRelease = function()
{
  if(this._currentframe == 10)
  {
    this.gotoAndStop('playOver');
    this._parent.pauseIt();
  }
  else
  {
    this.gotoAndStop('pauseOver');
    this._parent.unPauseIt();
  }
}

//Next Button
next.onRollOver = function()
{
  this.gotoAndStop('nextOver');
}

next.onRollOut = next.onReleaseOutside = function()
{
  this.gotoAndStop('next');
}

next.onRelease = function()
{
  if(sa[cps+1])
  {
    this._parent.playSong(cps);
  }
}

```

```

else
{
    if ((cps)== sa.length-1) //last song
    {
        this._parent.playSong(cps);
    }
    else
    {
        this._parent.playSong(++cps);
    }
}
}

//Mute Button
mute.onRollOver = function()
{
    if(this._currentframe == 1)
        this.gotoAndStop('onOver');
    else
        this.gotoAndStop('offOver');
}

mute.onRollOut = mute.onReleaseOutside = function()
{
    if(this._currentframe == 10)
        this.gotoAndStop('on');
    else
        this.gotoAndStop('off');
}

mute.onRelease = function()
{
    if(this._currentframe == 10)
    {
        this.gotoAndStop('offOver');
        s.setVolume(0);
    }
    else
    {
        this.gotoAndStop('onOver');
        s.setVolume(vc.ratio);
    }
}

//Progress bar click
loader.onPress = function()
{
    var barPos:Number = Math.round(this._xmouse);
    var offset:Number = Math.round(((barPos/loaderWidth) * s.duration)/1000);
    s.start(offset);
}

```

```

    playPause.gotoAndStop('pause');
}

var playListListener:Object = new Object();
//Playlist on click
playListListener.change = function()
{
    playSong(playList.getItemAt(playList.selectedIndex).data-1);
}

var keyListener:Object = new Object();
keyListener.onKeyDown = function()
{
    if (Key.getCode()==68) //letter 'd' - Delete
    {
        delete sa[playList.getItemAt(playList.selectedIndex).data];
        playList.removeItemAt(playList.selectedIndex);
        playPause.gotoAndStop('play');
        playSong(cps);
        playList.selectedIndex--;
    }
}

//Event registration
playList.addEventListener("change",playListListener);
Key.addListener(keyListener);

```

Statistikas izstrādes kods

```
<!-- Statistic module -->
<html>
<head>
<title>Statistics</title>
<script type="text/javascript" src="tabber.js"></script>
<link rel="stylesheet" href="example.css" TYPE="text/css" MEDIA="screen">
</head>
<script language="javascript">AC_FL_RunContent = 0;</script>
<script language="javascript"> DetectFlashVer = 0; </script>
<script src="AC_RunActiveContent.js" language="javascript"></script>
<script src="charts.js" type="text/javascript"></script>
<body>

<div class="tabber" id="my_total">

<?php
include 'config.php';
include 'dbconnect.php';

/* Function generChartXML generates XML from query according format
   $filename - resulting XML file name
   $query - $query with columns NAME and NUM from which gets data
   $header - chart header
   $color - chart column color*/
function generChartXML($filename, $query, $header, $columnColor)
{
    $sqlres = mysql_query($query);
    $dom = new DOMDocument();
    $root = $dom->createElement("chart");
    $dom->appendChild($root);
        $chartData = $dom->createElement("chart_data");
    $root->appendChild($chartData);
        $chartRow = $dom->createElement("row");
        $null = $dom->createElement("null");
        $chartRow->appendChild($null);
        $chartColumn = $dom->createElement("row");
        $chartHeader = $dom->createElement("string", "$header");
        $chartColumn->appendChild($chartHeader);
        while ($srow = mysql_fetch_array($sqlres))
        {
            $scur_str = $srow['NAME'];
            $string = $dom->createElement("string", "$scur_str");
            $chartRow->appendChild($string);
            $scur_num = $srow['NUM'];
            $number = $dom->createElement("number", "$scur_num");
            $chartColumn->appendChild($number);
        }
    $chartData->appendChild($chartRow);
}
```

```

    $chartData->appendChild($chartColumn);
    $chartLabel = $dom->createElement("chart_label");
    $position = $dom->createAttribute("position");
    $chartLabel->appendChild($position);
    $positionValue = $dom->createTextNode('middle');
    $position->appendChild($positionValue);
    $color = $dom->createAttribute("color");
    $chartLabel->appendChild($color);
    $colorValue = $dom->createTextNode('FFFFFF');
    $color->appendChild($colorValue);
    $root->appendChild($chartLabel);
    $seriesColor = $dom->createElement("series_color");
    $serColor = $dom->createElement("color", "$columnColor");
    $seriesColor->appendChild($serColor);
    $root->appendChild($seriesColor);
    $dom->save("$filename");
}

if(isset($_COOKIE['userLogged'])) //if user is logged in
{
    //getting his own statistics too
    $userName = $_COOKIE['userLogged'];
    $query = "select PK_USER from USERS where NAME='".$userName'";
    $sqlres = mysql_query($query);
    $row = mysql_fetch_array($sqlres);
    $userId = $row['PK_USER'];
    $mySongsWeekQuery = "select count(MP.FK_SONG) as NUM, S.NAME as NAME from
MONTH_PLAYED MP
    join SONGS S on S.PK_SONG=MP.FK_SONG
    where MP.FK_USER='$userId'." and yearweek(MP.DATE) =
yearweek(CURRENT_DATE)
    group by MP.FK_SONG order by
count(MP.FK_SONG) desc limit 5";

    $mySongsMonthQuery = "select count(MP.FK_SONG) as NUM, S.NAME as NAME from
MONTH_PLAYED MP
    join SONGS S on S.PK_SONG=MP.FK_SONG
    where MP.FK_USER='$userId'." and year(MP.DATE) =
year(CURRENT_DATE)
    and month(MP.DATE) =
MONTH(CURRENT_DATE)
    group by MP.FK_SONG order by
count(MP.FK_SONG) desc limit 5";

    $myArtistsWeekQuery = "select count(MP.FK_SONG) as NUM, A.NAME as NAME from
MONTH_PLAYED MP
    join SONGS S on S.PK_SONG=MP.FK_SONG join ARTISTS A on
A.PK_ARTIST=S.FK_ARTIST
    where MP.FK_USER='$userId'." and yearweek(MP.DATE) =
yearweek(CURRENT_DATE)

```

```

                                group by S.FK_ARTIST order by
count(MP.FK_SONG) desc limit 5";

    $myArtistsMonthQuery = "select count(MP.FK_SONG) as NUM, A.NAME as NAME
from MONTH_PLAYED MP
    join SONGS S on S.PK_SONG=MP.FK_SONG join ARTISTS A on
A.PK_ARTIST=S.FK_ARTIST
    where MP.FK_USER='$userId'." and year(MP.DATE) =
year(CURRENT_DATE)
                                and month(MP.DATE) =
MONTH(CURRENT_DATE)
                                group by S.FK_ARTIST order by
count(MP.FK_SONG) desc limit 5";
    generChartXML("mySongsWeekQuery.xml",$mySongsWeekQuery,"Week","51ae1a");
    generChartXML("mySongsMonthQuery.xml",$mySongsMonthQuery,"Month","F86FC6");
    generChartXML("myArtistsWeekQuery.xml",$myArtistsWeekQuery,"Week","51ae1a");
    generChartXML("myArtistsMonthQuery.xml",
$myArtistsMonthQuery,"Month","F86FC6");
    echo '<div class="tabbertab">';

    echo '<h2><a name="tab1">My statistics</a></h2>';
    echo '<div class="tabber" id="songs_artists">';

    echo '<div class="tabbertab">';
    echo " <h3>Songs</h3>";?>
    <script language="JavaScript" type="text/javascript">
    <!--
        createChart('mySongsWeekQuery.xml'); //draw charts from XML data
        createChart('mySongsMonthQuery.xml');
    // -->
    </script>
    <noscript>
        <P>This content requires JavaScript.</P>
    </noscript>
<?php
    echo "</div>";

    echo '<div class="tabbertab">';

    echo " <h3>Artists</h3>";?>
    <script language="JavaScript" type="text/javascript">
    <!--
        createChart('myArtistsWeekQuery.xml'); //draw charts from XML data
        createChart('myArtistsMonthQuery.xml');
    // -->
    </script>
    <noscript>
        <P>This content requires JavaScript.</P>
    </noscript>
<?php

```

```

echo " </div>";

echo "</div>";
echo "</div>";
}
?>
<!-- Total statistics. Always is showed -->
<div class="tabbertab">
<h2>Total</h2>
<?php
$totalSongsWeekQuery = "select count(MP.FK_SONG) as NUM, S.NAME as NAME from
MONTH_PLAYED MP
    join SONGS S on S.PK_SONG=MP.FK_SONG
    where yearweek(MP.DATE) = yearweek(CURRENT_DATE)
    group by MP.FK_SONG order by
count(MP.FK_SONG) desc limit 5";

$totalSongsMonthQuery = "select count(MP.FK_SONG) as NUM, S.NAME as NAME
from MONTH_PLAYED MP
    join SONGS S on S.PK_SONG=MP.FK_SONG
    where year(MP.DATE) = year(CURRENT_DATE)
    and month(MP.DATE) =
MONTH(CURRENT_DATE)
    group by MP.FK_SONG order by
count(MP.FK_SONG) desc limit 5";

$totalArtistsWeekQuery = "select count(MP.FK_SONG) as NUM, A.NAME as NAME
from MONTH_PLAYED MP
    join SONGS S on S.PK_SONG=MP.FK_SONG join ARTISTS A on
A.PK_ARTIST=S.FK_ARTIST
    where yearweek(MP.DATE) = yearweek(CURRENT_DATE)
    group by S.FK_ARTIST order by
count(MP.FK_SONG) desc limit 5";

$totalArtistsMonthQuery = "select count(MP.FK_SONG) as NUM, A.NAME as NAME
from MONTH_PLAYED MP
    join SONGS S on S.PK_SONG=MP.FK_SONG join ARTISTS A on
A.PK_ARTIST=S.FK_ARTIST
    where year(MP.DATE) = year(CURRENT_DATE)
    and month(MP.DATE) =
MONTH(CURRENT_DATE)
    group by S.FK_ARTIST order by
count(MP.FK_SONG) desc limit 5";
    generChartXML("totalSongsWeekQuery.xml",$totalSongsWeekQuery,"Week","51ae1a");
    generChartXML("totalSongsMonthQuery.xml",
$totalSongsMonthQuery,"Month","F86FC6");
    generChartXML("totalArtistsWeekQuery.xml",$totalArtistsWeekQuery,"Week","51ae1a");
    generChartXML("totalArtistsMonthQuery.xml",
$totalArtistsMonthQuery,"Month","F86FC6");
?>

```

```

<div class="tabber" id="songs_artists2">

  <div class="tabbertab">
    <h3>Songs</h3>
    <script language="JavaScript" type="text/javascript">
      <!--
        createChart('totalSongsWeekQuery.xml'); //draw charts from XML data
        createChart('totalSongsMonthQuery.xml');
      // -->
    </script>
    <noscript>
      <P>This content requires JavaScript.</P>
    </noscript>
  </div>

  <div class="tabbertab">

    <h3>Artists</h3>
    <script language="JavaScript" type="text/javascript">
      <!--
        createChart('totalArtistsWeekQuery.xml'); //draw charts from XML data
        createChart('totalArtistsMonthQuery.xml');
      // -->
    </script>
    <noscript>
      <P>This content requires JavaScript.</P>
    </noscript>
  </div>

</div>
</div>
</div>

<?php
  include 'dbdisconnect.php'; ?>

</body>
</html>

```