**Patstāvīgais darbs**

**"Mācību centra vadības sistēmas**

**datubāze un aplikācija"**

**Studiju kursā**

**"Datu bāzu tehnoloģijas"**

Darba autors:

Arnis Freimanis

Darba vadītājs:

Gints Neimanis

2015

Satura rādītājs

[1. IEVADS 3](#_Toc437808997)

[1.1. Datu bāzes un aplikācijas mērķis 3](#_Toc437808998)

[1.2. Datu bāzes un aplikācijas mērķis 3](#_Toc437808999)

[2. DATU BĀZES ENTĪTIJU SAIŠU DIAGRAMMA 4](#_Toc437809000)

[3. DATU BĀZES TABULAS 6](#_Toc437809001)

[3.1. Persona 6](#_Toc437809002)

[3.2. Auditorija 6](#_Toc437809003)

[3.3. AuditorijaNoslogojums 7](#_Toc437809004)

[3.4. Kurss 7](#_Toc437809005)

[3.5. MacibuGrupa 8](#_Toc437809006)

[3.6. PersonasApgutais 8](#_Toc437809007)

[3.7. PersonasNoslogojums 8](#_Toc437809008)

[3.8.Persona\_has\_Kurss 9](#_Toc437809009)

[3.9.Persona\_has\_MacibuGrupa 9](#_Toc437809010)

[3.10 GrupasPlanosana un GrupasPlanosanaStudenti 9](#_Toc437809011)

[4. DATU BĀZES VAICĀJUMI UN SKATI 10](#_Toc437809012)

[5. DATU BĀZES PAPILDUS NOSACĪJUMI 13](#_Toc437809013)

[5.1. Trigeri 13](#_Toc437809014)

[5.2. Funkcijas 13](#_Toc437809015)

[5.3. Procedūras 14](#_Toc437809016)

[6.DATU BĀZES APLIKĀCIJA 15](#_Toc437809017)

[7.SECINĀJUMI 23](#_Toc437809018)

# 1. IEVADS

## 1.1. Datu bāzes un aplikācijas mērķis

Datu bāzes mērķis ir palīdzēt uzņēmuma vadībai tikt galā ar darbinieku komandējumu, apmācību plānošanu, mācību grupu veidošanu, telpu noslogojumu. . Datu bāzē tiek glabāta informācija par darbinieku veiktajām apmācībām, viņu zināšanām, iegūtajiem sertifikātiem

Aplikācijas mērķis ir atspoguļot datu bāzes datus aplikāciju lietotājiem vēlamā, uzskatāmā veidā, kas ļauj iepazīties ar visu svarīgāko informāciju gan par personām, gan auditorijām, gan mācību kursiem.

## 1.2. Datu bāzes un aplikācijas mērķis

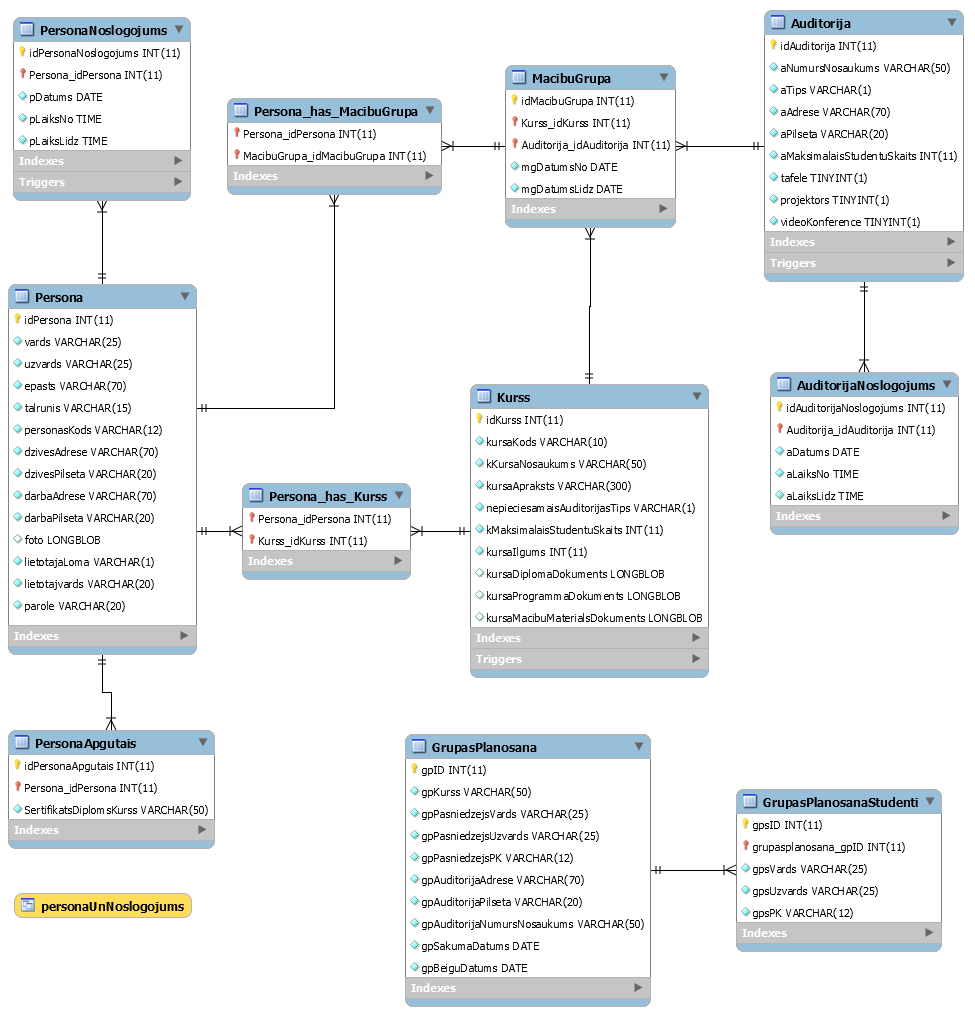
Datu bāzei ir vairāki uzdevumi, kuri ir realizējami aplikācijas līmenī:

* Atspoguļot informāciju par katru darbinieku;
* Nodrošināt iespēju izveidot jaunus aplikācijas lietotājus;
* Nodrošināt iespēju izveidot jaunas auditorijas;
* Nodrošināt iespēju izveidot jaunus kursus;
* Nodrošināt informāciju meklēšanu par personām, auditorijām un kursiem un atspoguļot meklēto informāciju;
* Nodrošināt iespēju izveidot jaunas mācību grupas;

# 2. DATU BĀZES ENTĪTIJU SAIŠU DIAGRAMMA

Mācību centra vadības sistēmas datu bāzes entītiju saišu diagramma sastāv no deviņām tabulām, no kurām septiņas ir pamata tabulas (Persona, PersonaNoslogojums, PersonaApgutais, MacibuGrupa, Kurss, Auditorija, AuditorijaNoslogojums), bet divas () ir starp tabulas, kuras veidojušas starp divām pamata tabulām, kurām saišu attiecības ir bijušas daudzi pret daudziem.

Pēc entītiju saišu diagrammas varam redzēt, ka vienā kursā mācas vairākas personas, bet vienai personai var būt vairāki apgūstamie kursi, tāpēc starp tabulām izveidojas saite daudzi pret daudzi un izveidojas starp tabula. Vairākas personas var būt vairākās mācību grupās, līdz ar to, starp tabulām Persona un MacibuGrupa tika izveidota saite daudzi pret daudzi, kam sekoja arī tabulas Persona\_has\_MacibuGrupa parādīšanās. Tad vēl tabulām PersonasNoslogojums un PersonasApgutais ar tabulu Persona veidojas saites viens pret daudzi, jo viena persona var būt apguvusi vairākas lietas un arī vienai personai pie noslogojuma var būt vairāki darāmi darbi. Tā pat arī tabulai MacibuGrupa bez saites daudzi, pret daudzi ir vēl divas citas saites ar tabulām Kurss un Auditorija, kas daudzi pret viens saites, jo vienai mācību grupai var attiecīgi piesaistīt vienu auditoriju un vienu apgūstamo kursu. Vēl veidojas saite viens pret daudzi ar tabulām Auditorija un AuditorijaNoslogojums, jo viena auditorija var būt aizņemta dažādos datumos un laikos.



# 3. DATU BĀZES TABULAS

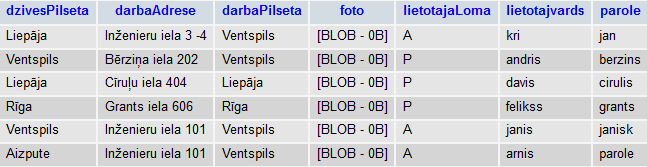
## 3.1. Persona

Kopumā tabula satur informāciju par lietotāju, personu. Katram lietotajam ir savs unikās identifikātors. Protams tike piefiksēts, lietotaja, vārds, uzvārds, epasts, tālrunis, personas kods, kā arī gan dzīves vietas gan darba vietas adreses un pilsētas. Tad vēl šī tabula satur personas bildi, priekš profila lapas aplikācijā, lietotāja lomu (A - administrators, L - lietotājs, P - pasniedzejs) un protams arī litotajvārdu un paroli.



3.1 att. Persona tabulas dati

----->



3.2 att Persona tabulas datu turpinājums

## 3.2. Auditorija

Tabula "Auditorija" satur sevī visas pieejamās auditorijas, katrai auditorijai ir savs unikālais idenfikators un nosaukums, kas ir nosaukums vai numurs. Vēl tabula satur tādu informāciju kā auditorijas tips (A - auditorija, D - datorauditorija), adrese, pilsēta, maksimālais vietu skaits auditorijā un vēl tiek piezīmēta informācija par, tāfeles, projektora un video konferences esamību auditorijā (1 - ir, 0 - nav).



3.3 att Auditorija tabulas dati

## 3.3. AuditorijaNoslogojums

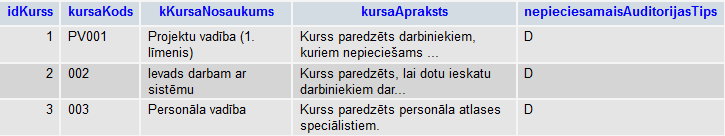
Šī tabula satur informāciju par auditorijas noslogojumu. Katram ierakstam ir savs unikālais idenfikators, tad vēl tiek izvēlēta kāda no esošajām auditorijām pēc to unikāla idenfikatora un ievadīts datums, laika intervāls, kurā auditorija ir aizņemta.



4.4 att AuditorijaNoslogojums tabulas dati

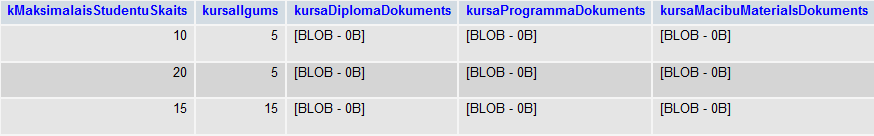
## 3.4. Kurss

Tabulā "Kurss" tiek uzkrāta informācija par kursiem, katram kursam ir savs unikālais idenfikators, kursa kods, nosaukums un neliels apraksts par kursu. Tad vēl tiek uzskaitīti dati par to kāda auditorija būs nepieciešama (A - auditorija, D - datorauditorija), maksimālais studentu skaits grupā un to cik ilgi norisināsies mācības (nedēļās). Tad vel tabula satur trīs dokumentus - kursa diplomu, programmu un mācību materiālus.



3.5 att. Kurss tabulas dati

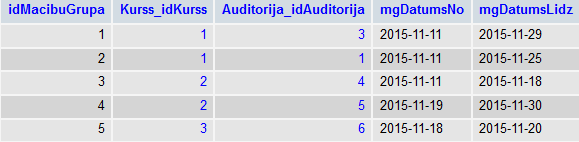
----->



3.6 att. Kurss tabulas datu turpinājums

## 3.5. MacibuGrupa

Pavisam vienkārša tabula, kura satur informāciju par mācību grupām, par to kāds kurs būs apgūstams, kādā auditorijā tas notiks un no kura līdz kuram datumam viss norisināsies, protams, katrai mācību grupai arī ir savs unikālais idenfikators.



3.7 att. MacibuGrupa tabulas dati

## 3.6. PersonasApgutais

Vēl viena vienkārša tabula, kas sevī satur unikālo idenfikatoru ,kādu no sistemā esošajām personām un vēl nosaukumu tam, ko šī persona ir apguvusi.



3.8 att. PersonasApgutais tabulas dati

## 3.7. PersonasNoslogojums

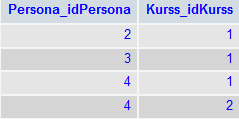
Tabula "PersonasNoslogojums" ir ļoti līdzīga tabulai "Auditoriju noslogojums". Katram ierakstam ir savs unikālais idenfikators, tad vēl tiek izvēlēta kāda no esošajām personām pēc to unikāla idenfikatora un ievadīts datums, laika intervāls, kurā persona ir aizņemta.



3.8 att. PersonasNoslogojums tabulas dati

## 3.8.Persona\_has\_Kurss

Tabula "Persona\_has\_Kurss" satur informāciju par konkrētu personu, kurš kurs viņam ir jāapgūst. Piemēram, pirmajā rindā varam redzēt, ka personai Andrim Bērziņam ir jāapmeklē kurss Projektu vadība.



3.8 att. Persona\_has\_Kurss tabulas dati

## 3.9.Persona\_has\_MacibuGrupa

Tabula "Persona\_has\_MacibuGrupa" izveidojas starp tabulām "Persona" un "MacibuGrupa" un satur informāciju par kādas konkrētas personas mācību grupu, kurai, šī persona ir pievienota. Piemēram, pirmajā rindā varam redzēt, ka persona Fēlikss Grants mācas pirmajā mācību grupā.



3.8 att. Persona\_has\_MacibuGrupa tabulas dati

## 3.10 GrupasPlanosana un GrupasPlanosanaStudenti

Šīs divas tabulas ir domātas, lai uzglabātu informāciju, tad kad tiek veidota jauna mācību grupa, saglabāju informāciju šijās, lai pēc tam vieglāk ierakstīt citās, viņas regulāri ir tukšas, tik līdz tiek piespiesta poga "Izveidot jaunu mācību grupu" viņas tiek iztīrītas.

# 4. DATU BĀZES VAICĀJUMI UN SKATI

Aplikācijā tiek izmantoti vairāk kā desmit dažādi vaicājumi. Šajā darba sadaļā uzrādīšu piecus interesantākos vaicājumus un visus trīs skatus kuri tiek izmantoti.

Pirmajā vaicājumā tiek izkrītošajā sarakstā ielasīti visi iespējamie pasniedzēji, kādam kursam, kas ir izvēlēts iepriekšējā izkrītošajā sarakstā. Vaicājums tiek izmantots failā ar nosaukumu "groupPlanning.php".

SELECT Persona.vards, Persona.uzvards FROM Persona LEFT JOIN Persona\_has\_Kurss ON Persona.idPersona = Persona\_has\_Kurss.Persona\_idPersona WHERE Persona.lietotajaLoma = 'P' AND Persona\_has\_Kurss.Kurss\_idKurss = '$selectedCourseId';

Otrais vaicājums nav sarežģīts bet spēlē lielo lomu visā sistēmā, ar tā palīdzu norisinās pieslēgšanās sistēmai. Vaicājums tiek izmantots failā ar nosaukumu "indexNew.php", kas ir pieslēgšanās skripts.

SELECT parole, lietotajvards, lietotajaLoma FROM Persona WHERE parole='$password' AND lietotajvards='$username';

Trešais vaicājums, nospiežot pogu apstiprināt izkrītošajā sarakstā ielasās visi sistēmā reģistrētie pasniedzēji. Vaicājums tiek izmantots failā ar nosaukumu "groupPlanning.php".

UPDATE GrupasPlanosana SET gpPasniedzejsVards = '$selectedTeacherName', gpPasniedzejsUzvards = '$selectedTeacherSurname', gpPasniedzejsPK = '$selectedTeacherID' WHERE gpID = $maxId;

Ceturtais vaicājums, jeb vaicājumu bloks, sistēmā ir izstrādāta meklēšana un lai tā veiksmīgi strādātu ir nepieciešams izveidot šādu bloku ar ne īpaši sarežģītiem vaicājumiem, tas viss ir nepieciešams, lai meklēšana arī strādātu, ja kāds no meklēšanas laukiem ir atstāts neaizpildīts. Vaicājums tiek izmantots failā ar nosaukumu "search.php".

if($surname == "" && $peronID ==""){

$sql = "SELECT \* FROM Persona WHERE vards='$name'";

$result = $d->query($sql);

} else if($name == "" && $peronID ==""){

$sql = "SELECT \* FROM persona WHERE uzvards='$surname'";

$result = $d->query($sql);

} else if($name == "" && $surname ==""){

$sql = "SELECT \* FROM persona WHERE personaskods='$peronID'";

$result = $d->query($sql);

} else if($peronID == ""){

$sql = "SELECT \* FROM persona WHERE vards='$name' AND uzvards='$surname'";

$result = $d->query($sql);

} else if($name == ""){

$sql = "SELECT \* FROM persona WHERE uzvards='$surname' AND personaskods='$peronID'";

$result = $d->query($sql);

} else if($surname == ""){

$sql = "SELECT \* FROM persona WHERE vards='$name' AND personaskods='$peronID'";

$result = $d->query($sql);

} else{

$sql = "SELECT \* FROM persona WHERE vards='$name' AND uzvards='$surname' AND personaskods='$peronID'";

$result = $d->query($sql);

}

Piektais vaicājums ir INSERT vaicājums, tādi sistēmā ir vairāki, šis viens ir kā piemērs, jo pārējie visi ir ļoti līdzīgi. Ar šī vaicājuma palīdzību tiek ierakstīts jauns lietotājs datubāze. Vaicājums tiek izmantots failā ar nosaukumu "insertUser.php".

INSERT INTO Persona(vards, uzvards, epasts, talrunis, personasKods, dzivesAdrese, dzivesPilseta, darbaAdrese, darbaPilseta, foto, lietotajaLoma, lietotajvards, parole) VALUES('$vards','$uzvards','$epasts', '$talrunis', '$persKods', '$dzivesAdrese', '$dzivesPilseta', '$darbaAdrese', '$darbaPilseta','$foto' ,'$loma', '$lietotajvards', '$parole');

Pirmais skats tiek izmantots lietotāja profila lapā, lai parādītu informāciju par viņu un par to kas personai ieplānots tuvākajā laikā.



----->



----->



4.1 att. PersonaUnNoslogojums skata tabula

CREATE VIEW personaUnNoslogojums AS SELECT \* FROM Persona JOIN PersonaNoslogojums ON idPersona = Persona\_idPersona;

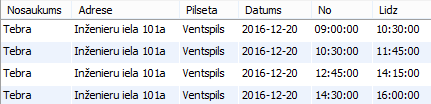
Otrajā skatā esmu apvienojis informāciju par personu un viņa apgūto kursu vau kursu, kurš tam ir jāapgūst.



4.2 att. PersonasApgutaisKurs tabula

CREATE VIEW personasApgustamaisKurs AS SELECT vards, uzvards, kKursaNosaukums FROM Persona JOIN Kurss JOIN Persona\_has\_Kurss ON Persona.idPersona = Persona\_has\_Kurss.Persona\_idPersona AND Kurss.idKurss = Persona\_has\_Kurss.Kurss\_idKurss;

Trešais skats, ko nepieciešams atspoguļot atrodot informāciju par auditoriju atspoguļo laikus kuros auditorija ir aizņemta.



4.3 att. auditorijaAiznemta tabula

CREATE VIEW auditorijaAiznemta AS SELECT aNumursNosaukums as Nosaukums, aAdrese as Adrese, aPilseta as Pilseta, aDatums as Datums, aLaiksNo as No, aLaiksLidz as Lidz FROM Auditorija JOIN AuditorijaNoslogojums ON idAuditorija = Auditorija\_idAuditorija;

# 5. DATU BĀZES PAPILDUS NOSACĪJUMI

## 5.1. Trigeri

No sākuma vēlējos izveidot trigeri, kas izdzēsīs notikumu no tabulas, kad datums būs aizgājis, tā kā tabulā bija FK bija nepieciešami trigeri, kas man ļaus to darī, vispirms izveidoju tos un tad sāku veidot to trigeri ar ko dzēsīšu ārā notikumus, bet tas man neizdevās un tāpēc šie divi ir palikuši datu bāzē.

CREATE TRIGGER fktozero BEFORE DELETE ON PersonaNoslogojums FOR EACH ROW SET FOREIGN\_KEY\_CHECKS=0;

CREATE TRIGGER fktoone AFTER DELETE ON PersonaNoslogojums FOR EACH ROW SET FOREIGN\_KEY\_CHECKS=1;

CREATE TRIGGER deleteAction AFTER INSERT ON personaNoslogojums BEGIN DELETE FROM LOG WHERE DATE(pDatums) > DATE('now', '1 days') END

Vēl sistēmā ir divi šādi trigeri kas neļauj pievienot negatīvus skaitļus, jo ne auditorijā vietu skaits nevar būt negatīvs un tā pat arī ar studentu skaitu kursā.

DELIMITER // CREATE TRIGGER studentuSkaitsPozitivs BEFORE UPDATE ON Kurss FOR EACH ROW BEGIN IF new.kMaksimalaisStudentuSkaits < 0 then SIGNAL SQLSTATE '45000'; END IF; END // DELIMITER ;

DELIMITER // CREATE TRIGGER studentuSkaitsAuditorijaPozitivs BEFORE UPDATE ON Auditorija FOR EACH ROW BEGIN IF new.aMaksimalaisStudentuSkaits < 0 then SIGNAL SQLSTATE '45000'; END IF; END // DELIMITER ;

## 5.2. Funkcijas

Funkcijas bija tās ko es sākumā īsti nesapratu un arī nemācēju uzrakstīt un tā kā mēs viņas sākām mācīties jau kad mana sistēma daudz maz bija izstrādāta un visu ko it kā varēju vieglāk uztaisīt ar funkciju palīdzību biju izdarījis ar php palīdzību, tāpēc arī manā sistēmā ir tikai trīs funkcijas, kas diemžēl netiek lietotas, bet viennozīmīgi būtu palīdzējušas, šī ir funkcija, kas pārbaudītu lietotāja ievadīto lietotājvārdu un paroli pieslēdzoties sistēmai un attiecīgi atgrieztu viens vai nulle.

DELIMITER //

CREATE FUNCTION pieslegsanasParbaude (user CHAR(45), pass CHAR(45))

RETURNS INT

BEGIN

DECLARE pasword CHAR(45);

SELECT parole

FROM Persona

WHERE lietotajvards = user

INTO pasword;

IF pass = pasword THEN

RETURN 1;

ELSE

RETURN 0;

END IF;

END; // DELIMITER ;

Otrā funkcija ko izveidoju ir funkcija, kas skaita cik auditoriju ir sistēmā, to nedaudz uzlabojot varētu skaitīt ccik daudz auditoriju ir vienā un tajā pašā adresē vai kā citādāk. Funkcijai tiek padota ieraksta primārā atslēga, pēc kuras no tabulas tiek atlasīti visi atbilstošie ieraksti un iegūts to kopējais skaits. Funkcija atgriež INT(11) tipa vērtību:

DELIMITER //

CREATE FUNCTION auditorijuSkaits(id INT(11))

RETURNS INT(11)

DETERMINISTIC

BEGIN

DECLARE skaits INT(11);

SELECT COUNT(idAuditorija) INTO skaits FROM Auditorija WHERE idAuditorija=id

GROUP BY idAuditorija;

RETURN skaits;

END

DELIMITER ;

## 5.3. Procedūras

Ar pirmās procedūras palīdzību būtu iespējams dzēst lietotājus pēc viņu unikālā ID.

CREATE PROCEDURE dzest\_lietotaju(id INT (11)) DELETE FROM Persona WHERE idPersona=id;

Ar otrās procedūras palīdzību ir iespējams atrast lietotājus pēc vārda.

CREATE PROCEDURE atarst\_lietotaju (vards CHAR(45)) BEGIN SELECT idPersona, vards FROM Persona WHERE vards like concat('%',vards,'%');

Trešā procedūra palīdz atgriest informāciju par kādu sistēmas lietotāju padodot tā ID.

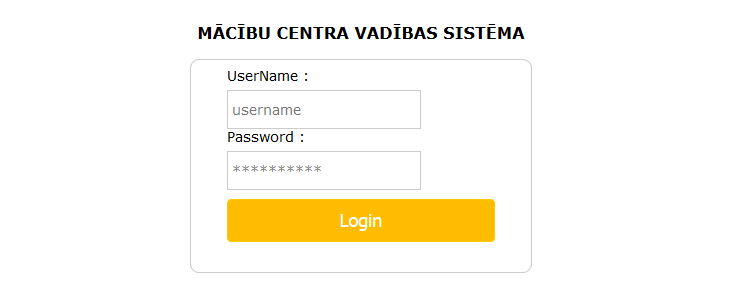
CREATE PROCEDURE LietotajaInfo(idLietInfo INT) BEGIN SELECT vards, uzvards, epasts, talrunis, personasKods FROM Persona WHERE idPersona = idLietInfo;

# 6.DATU BĀZES APLIKĀCIJA

Datu bāzes aplikācija veidota ar mērķi atspoguļot tik pat kā visus datubāzes datus. Aplikācija ir sadalīta lomās, kur ir administrators un lietotājs, lietotājs var tikai apskatīt savu profila lapu un noskaidrot par sev tuvākajām aktivitātēm, bet administrators var veidot jaunus lietotājus, kursus, telpas, mācību grupas un vēl plusā izgūt informāciju no datubāzes pēc dažādiem meklēšanas kritērijiem.

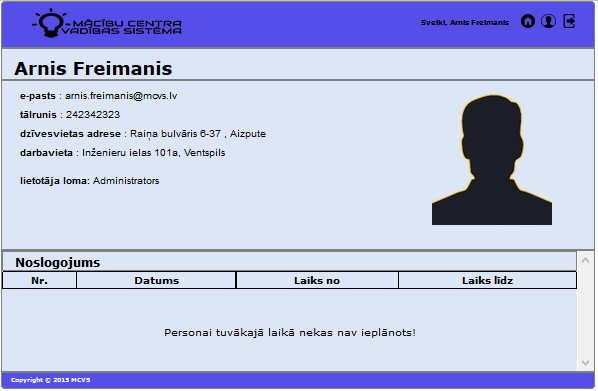
Aplikācija ir domāta kāda uzņēmuma iekšējai lietošanai tāpēc svarīgu lomu spēlē autorizācija. Aplikācija pieejama: http://ovz.venta.lv/15.24/

Pirmais skats ar ko lietotājs saskaras ir autorizācija, kur katrs lietotājs ar savu unikālo lietotājvārdu un paroli var autorizēties un strādāt ar aplikāciju.



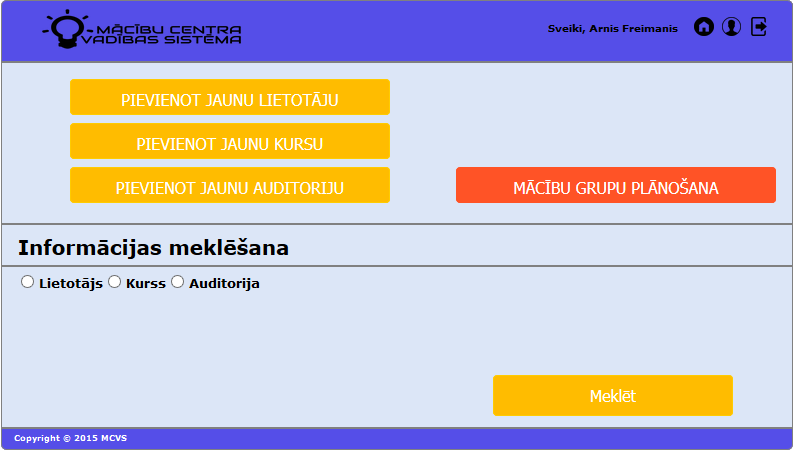
6.1 att Aplikācijas autorizācijas logi

Pēc autorizācijas uzņēmuma darbinieks, ne sistēmas administrators nonāk savā profila lapā, kurā var izlasīt informāciju par sevi un par sev tuvāk ieplānotajām mācībā, bet protams arī administratoram ir pieejama šāda lapa, tikai viņam nākas spiest uz ikonu labajā augšējajā stūrī, cilvēka silueta izskatā. Lietotājam ir ļoti ierobežotas tiesības un tas ir arī viss ko viņš var apskatīt sistēmā.



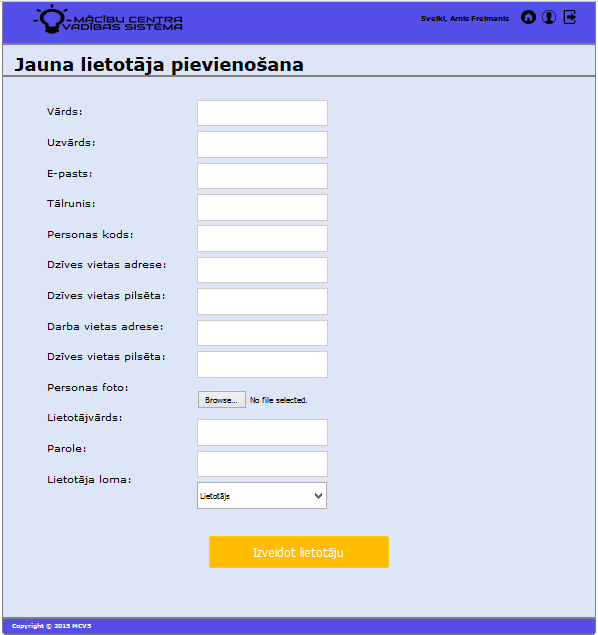
6.2 att.Lietotāja un Administratora profila lapa

Ja sistēmai pieslēdzās kā administrators uzreiz paveras daudz plašāks iespēju logs. Tā tad viņam pēc autorizācijas atveras logs kurā ir četras pogas, pirmā no tām ved uz jauna lietotāja izveidi, otrā uz jauna kursa izveidi, trešā uz jaunas auditorijas izveidi un ceturtā uz jaunas mācību grupas izveidi. Pēc visām šīm iespējām vēl administrators var meklēt informāciju, meklēt var gan personas, gan auditorijas, gan kursus, katru pēc citiem meklēšanas kritērijiem.



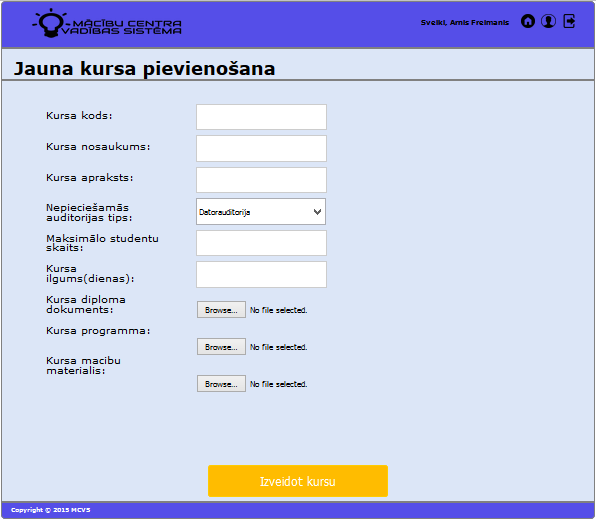
6.3 att. Administratora galvenā lapa

Ja lietotājs izvēlas "PIEVIENOT JAUNU LIETOTĀJU" tad viņš nonāk nākamajā lapā, kur viņam ir jāaizpilda visi nepieciešamie lauki un pēc izvēles jāpievieno personas attēls, ja tas netiek izdarīts, tad profila bildes vietā tiek uzstādīta noklusētā bilde.



6.4 att. Jauna lietotāja pievienošanas logs

Līdzīgi kā jauna lietotāja pievienošanā arī ir jaunas auditorijas un kursa pievienošanā, protams, ir jāievada cita informācija un pie jauna kursa izveidošanas vēl ir iespēja augšupielādēt dažādus dokumentus.



6.5. att. Jauna kursa pievienošanas logs



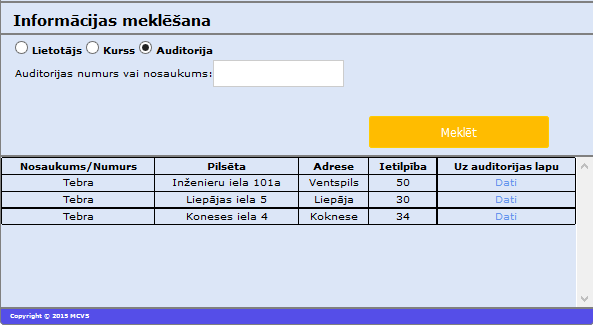
6.5. att. Jaunas auditorijas pievienošanas logs.

Jaunas mācību grupas pievienošana ir nedaudz sarežģītāka, lietotājs vispirms izvēlas mācību kas tiks pasniegta un spiež pogu apstiprināt, pēc pogas nospiešanas, lietotājs var izvēlēties nākamajā laukā visus pasniedzējus, kas šo kursu var pasniegt, tā pat ar auditoriju. Protams vēl ir jāievada laiks, kad notiks šīs mācības un jāpievieno studenti. Studentu pievienošana notiek sekojoši, studentu var meklēt, pēc vārda, uzvārda vai viņa personas koda, pēc pogas meklēt nospiešanas parādās logs ar visiem studentiem kas atbilst ievadītajiem parametriem un ar meklēšanas palīdzību tie tik pievienoti sarakstā un tiek izveidota jauna mācību grupa.

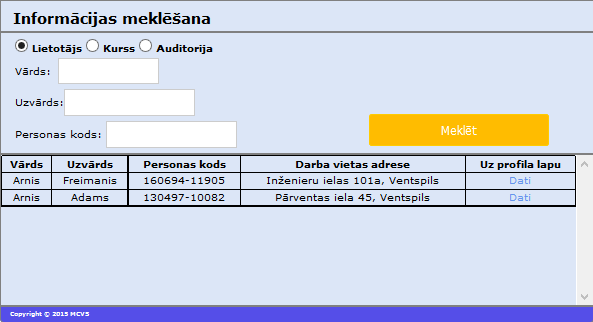


6.6. att.Jaunas mācību grupas logs

Tad vēl galvenajā lapā administrators var veikt meklēšanu, meklēšana notiek tā, ka no sākuma izvēlamies, ko meklēsim uzklikšķinot uz radio pogas, attiecīgi izvēlei parādās parametri pēc kuriem ir iespējams meklēt un piespiežot pogu meklēt parādās tabula ar visiem atrastajiem rezultātiem un nelielu informāciju par tiem, bet ja lietotājs vēlas uzzināt ko vairāk viņam ir jāspiež uz vārdu, pogu "Dati" un tā viņu aizvedīs uz plašāka izklāsta lapu kuras ir ļoti līdzīgas jau iepriekš pieminētajai lietotāja profila lapai.



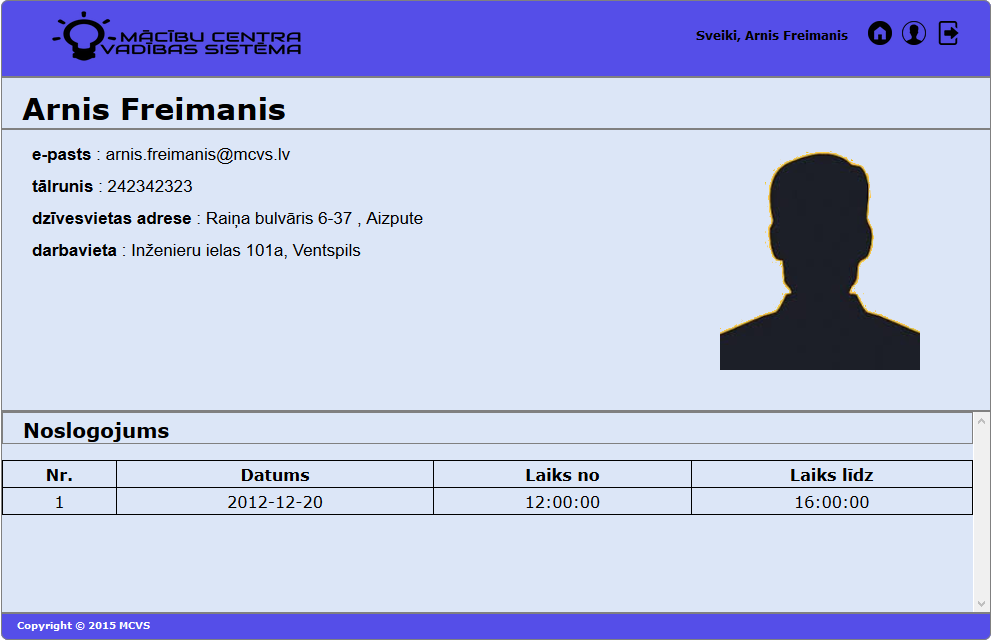
6.8. att. Auditorijas meklēšana ar rezultātiem



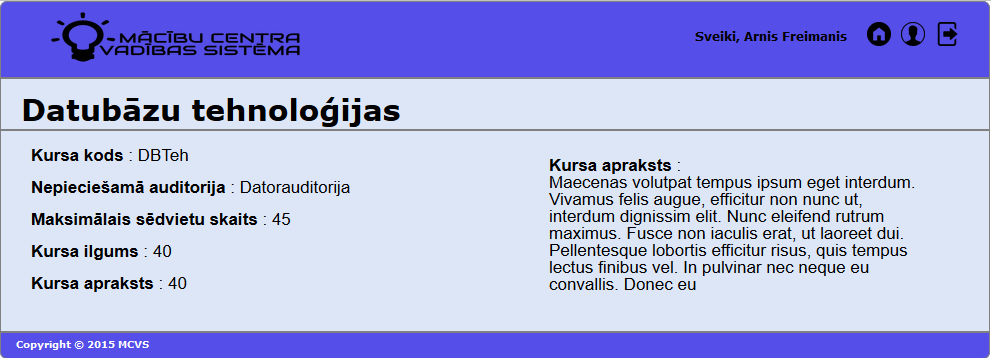
6.9. att. Lietotāja meklēšana ar rezultātiem



6.10. att. Kursu meklēšana



6.11. att. Atrastā lietotāja plašākas informācijas lapa



6.12. att. Atrastās auditorijas plašākas informācijas lapa



6.14. att. Atrastā kursa plašākas informācijas lapa

# 7.SECINĀJUMI

It kā datu bāze varētu būt pielietojama reālitātē un tā spētu veikt visas savas dotās iespējas, bet es uzskatu, ka vēl daudz kas ir jāuzlabo un jāpievieno vēl vairāki padsmiti ļoti nepieciešamas funkcijas, es personīgi teiktu, ka šis ir tikai sistēmas sākums un izstrāde vēl turpināsies.

Tas ko es izdarīju varbūt ne tā ir, ka sāku jau sistēmu izstrādāt ļoti, ļoti laicīgi, kad vēl nebijām mācījušies gan par trigeriem, procedūrām, funkcijām un skatiem, jo dēļ tā, ka es nemācēju tādus tad lietot un nemaz īsti neko nenojautu par tādām lietām lielāko uzsvaru liku uz php, kad ar tā palīdzību arī ļoti daudz ko var realizēt.

Tā kā ir vēl dažas lietas kuru pievienošana un labošana notiek caur phpmyadmin uzskatu ka vēl ir daudz ko pielikt un ļoti gribēju paspēt uztaisīt lietotājiem ērtu un drošu datubāzes informācijas labošanu, bet to nepaspēju izdarīt. Un vēl ir nepieciešams uzlabot jaunas mācību grupas pievienošanu, ir savas nepilnības.

Visā visumā uzskatu, ka šis kurs un visu ko mēs darījām gan lekcijās, gan patstāvīgi, man personīgi ir devis ļoti, ļoti daudz un turpināšu gan datu bāzēs gan php individuāli mācīties un izzināt visu to kārtīgāk.