



Rīgas Valsts tehnikums

Datorikas nodaļa

KVALIFIKĀCIJAS DARBA SATURS

kvalifikācijai “Programmēšanas tehniks”

Izstrādāja:

Ilona Demčenko, Jeļena Matvejeva

Oksana Roslova, Igors Litvjakovs

*paraugi: **Katrīna Gandzjuka***

1. KVALIFIKĀCIJAS DARBA IZSTRĀDE

Kvalifikācijas darbs (KvD) specialitātei “Programmēšanas tehniķis” tiek izstrādāts kvalifikācijas prakses laikā un tiek izmantots kā praktiskā daļa kvalifikācijas eksāmenā atbilstoši kvalifikācijas eksāmena programmai^[1]. Prakses beigās notiek kvalifikācijas darba prezentēšana, kurā ar Datorikas nodaļas profesionālo priekšmetu skolotāju līdzdalību tiek vērtēts kvalifikācijas darbs un tiek ierosināti uzlabojumi, kas jāveic pirms kvalifikācijas eksāmena.

1.1. Kvalifikācijas darba uzbūve

Kvalifikācijas darbs sastāv no:

- **programmprodukta** – programmas kods un izpildāmais modulis, kas tiek saglabāts GitHub repozitorijā;
- **paskaidrojošā raksta** – konkrētas struktūras dokuments, ko noformē atbilstoši Datorikas nodaļas mācību darbu noformēšanas norādījumiem^[2]. Tiek iesiets ar spirāli, kopējais apjoms ~100-120 lpp. (līdz pielikumiem ~60-80 lpp.);
- **prezentācijas** – tiek izmantota aizstāvēšanas laikā.

1.2. Kvalifikācijas darba tēmas izvēle

Kvalifikācijas darba tēmu un darba veidošanas rīkus izvēlas audzēknis, saskaņo tos ar kvalifikācijas darba vadītāju kvalifikācijas prakses sākumā. Kvalifikācijas darba tematu audzēknis var izvēlēties jebkurā programmēšanas jomā atbilstoši savām interesēm un studiju programmas prasībām. Kvalifikācijas darbu ir ieteicams saistīt ar aktuālām un modernām programmēšanas tehnoloģijām, ka arī vēlams izvēlēties interesantu un mūsdienās aktuālu projekta izstrādes priekšmeta jomu. Tēmas nosaukumā:

- nedrīkst būt salikts teikums.
- jāsaturs informācija par sistēmas darbības un pielietojšanas jomu, kā arī sistēmas veidu (tīmekļa lietojumprogramma, darbvirsmas lietojumprogramma, mobilā aplikācija, spēle utt).
- var pievienot izvēlētās izstrādes vides vai platformas nosaukumu.

¹Centralizēto profesionālās kvalifikācijas eksāmenu programmas. <https://www.visc.gov.lv/lv/media/23847/download?attachment>
(Resurss apskatīts 08.10.2024.)

² I.Demčenko, O.Sabanska. Norādījumi mācību darbu noformēšanai 2024.

2. PASKAIDROJOŠĀ RAKSTA IZSTRĀDE

Kvalifikācijas darba paskaidrojošais raksts atspoguļo kvalifikācijas darba programmaprodukta izstrādes procesu un rezultātu. Paskaidrojošā raksta saturu nosaka kvalifikācijas eksāmena praktiskās daļas vērtēšanas kritēriji. Paskaidrojošā raksta sadaļas ir hronoloģiski un pēctecīgi sakārtotas, sākot ar sistēmas izstrādes ideju, prasību noteikšanu, projektēšanu, līdz gatavā programmaprodukta aprakstam. Specifisko KvD programmaprodukta uzdevumu gadījumā saturs var būt mainīts, saskaņojot to ar Datorikas nodaļas Metodisko komisiju. Bez ierobežojumiem drīkst ievietot paskaidrojošā rakstā papildus nodaļas, saglabājot noteiktās.

Obligātās paskaidrojošā raksta sastāvdaļas ir: titullapa, anotācijas (*valsts valodā un svešvalodā*), saturs, ievads, uzdevuma nostādne, prasību specifikācija (*ieejas un izejas informācijas apraksts, funkcionālās prasības, nefunkcionālās prasības*), uzdevuma risināšanas līdzekļu izvēles pamatojums, programmatūras produkta modelēšana un projektēšana (*sistēmas struktūras modelis - sistēmas arhitektūra, sistēmas ER modelis, funkcionālais sistēmas modelis - datu plūsmu modelis*), datu struktūru apraksts, lietotāja ceļvedis (*sistēmas prasības aparatūrai un programmatūrai, sistēmas instalācija un palaišana, programmas apraksts*), programmatūras lietojamības testēšana, nobeigums, informācijas avoti, pielikumi (*diagrammas, ekrāna attēli, programmas pirmkods*).

2.1. Titullapa

Kvalifikācijas darba titullapu noformē atbilstoši paraugam (skat. 1. pielikums). Drīkst, bet nav obligāti, izmantot rāmīti. Pirms kvalifikācijas darba aizstāvēšanas darbu paraksta visas titullapā minētās personas. Titullapā ir jānorāda paskaidrojošā raksta lappušu skaits bez pielikumiem.

2.2. Anotācija

Valsts un svešvalodā rakstīto anotāciju saturam ir jāsakrīt. Svešvalodu izvēlas audzēknis no Eiropas savienības oficiālām valodām (angļu, vācu, spāņu, ... valodas). Anotācija sastāv no trim rindkopām. Pirmajā teikumā ir jāformulē kvalifikācijas darba tēma. Pirmajā rindkopā ir īsi jāskaidro izstrādājamās programmsistēmas būtība, jāuzrāda izstrādāšanas rīki. Otrajā rindkopā jānodod īss paskaidrojošā raksta nodaļu saturs. Trešajā rindkopā ir saskaitīts, cik paskaidrojošā rakstā kopā ir lappušu, attēlu, tabulu un pielikumu. Anotācijas piemērs valsts valodā ir dots 2. pielikumā. Anotāciju lapās netiek uzrādīta lappušu numerācija.

2.3. Saturs

Saturu izveido atbilstoši paraugam (skat. 3. pielikums), ievērojot trekno, slīpu burtu un atkāpju noformēšanu. Saturā iekļauj visus teksta virsrakstus līdz 3. līmenim ieskaitot. Ja satura rādītāju veido, izmantojot MS Word Insert Index rīku, pielikumu pārskaitījumu parasti papildus pieraksta ar roku. Saturam ir jāiekļaujas vienā lapā, kuru nenumurē.

2.4. Ievads

Ievadā apraksta programsistēmas veidošanas nepieciešamību, vajadzību, raksturo sistēmas aktualitāti, pamato tās veidošanas lietderīgumu, veicot nelielu tirgus izpēti ar dažu esošo alternatīvo sistēmu raksturojumu. Var aprakstīt pasūtītāja organizācijas situāciju, kas pastāv pirms programmas veidošanas; izvirzīt problēmu tā, lai labākais risinājums būtu KvD programmas veidošana. Uzrāda mērķauditoriju, kam paredzēta sistēma, aprakstot ieinteresēto personu klāstu. Apjoms ir no $\frac{3}{4}$ lapas līdz 2 lapām. Ievads ir pirmā lapa, kas tiek numurēta ar skaitli “5”. Ievada sākuma piemērs ir dots 4. pielikumā.

2.5. Uzdevuma nostādne

Uzdevuma nostādnes pirmajā rindkopā ir jādod īss un konkrēts uzdevuma formulējums. Otrajā rindkopā ir jāapraksta galvenās sistēmas funkcionalitātes (ap ~7), sākot ar svarīgākām, izmantojot aizzīmēto sarakstu (bullets). Funkcionalitātes var atdalīt pēc lietotāju lomām (piem., klients, administrators). Uzdevuma nostādne parasti aizņem $\frac{1}{2}$ – $\frac{2}{3}$ lapas. Ilustrējot iespējamo lietotāju un funkcionalitāšu mijiedarbību ir papildus jāizmanto lietojumgadījumu diagramma (use case diagramm), kurā jāizvieto šajā pašā lapā, vai pirmajā pielikumā, ja tā neietilpst. Lietojumgadījuma diagrammā jāattēlo sistēma ar īso nosaukumu, visas lietotāju lomas (aktori), 5-7 lietojumgadījumi ar saitēm. Obligāti ir jādod atsauce uz diagrammu.

Uzdevuma nostādnes piemērs ir dots 5. pielikumā.

2.6. Prasību specifikācija

2.6.1. Ieejas informācijas apraksts

Ieejas informācijas aprakstā ir jābūt visam programmā ievadāmās informācijas aprakstam ar formātu, ierobežojumiem no lietotāja viedokļa un ievada informācijas piemēriem. Ieteicams informācijas vienības sagrupēt pēc datu kopām (iespējams, tās pārvērsīsies par entitijām un, turpmāk, par tabulām), izceļot datu kopu nosaukumus (piem., “Jāievada šāda informācija par

frizētavas klientu:”). Katrai ievadāmās informācijas vienībai ir jāuzrāda nosaukums, būtības apraksts, izmērs (piem., “līdz 20 simboliem”), datu tips (piem., “veselie skaitļi”), bet ne konkrētas programmēšanas valodas sintakses termini; ja ir, atļautie/neatļautie simboli, datu formāts (piem., “Iepirkuma datums – dd.mm.yyyy formāta ciparu kombinācija”). Jāuzrāda ievadāmās informācijas piemērs. Informācijas vienības jāapraksta tikai vienu reizi, pat, ja programmā būs paredzēts ievadīt tās vairākos logos dažādiem lietotājiem.

Ja sistēmā ir paredzēts informāciju iegūt ne tikai no ekrānformām, bet arī no citiem avotiem (datu failiem, ierīcēm, citām lietotnēm spraudņu formātā), šeit ir jādod detalizēts šo datu struktūras apraksts. Nevajag uzrādīt ID, dienesta laukus, atsaukties uz tabulām!!!

Parasti Ieejas informācijas apraksts aizņem 1½ - 3 lapas. Ieejas informācijas piemērs ir dots 1. attēlā.

PRASĪBU SPECIFIKĀCIJA

2.1. Ieejas un izejas informācijas apraksts

2.1.1. Ieejas informācijas apraksts

Sistēmā tiks nodrošināta šādas ieejas informācijas, ko ievadīs lietotājs no tastatūras, apstrāde.

1. Informācija par **lietotājiem** sastāvēs no šādiem datiem.

- Personas kods – lietotāja personas kods – burtu teksts ar izmēru 12 rakstzīmes (piem. 160906-27839).
- Vārds - lietotāja vārds - burtu teksts ar izmēru līdz 30 rakstzīmēm (piem. Alise).
- Uzvārds - lietotāja uzvārds - burtu teksts ar izmēru līdz 30 rakstzīmēm (piem. Silīņa).

1.att.Ieejas informācijas apraksta piemērs

2.6.2. Izejas informācijas apraksts

Šajā sadaļā apraksta visu programmprodukta rezultējošo informāciju. Izejas informācija – tā ir programmas darbības mērķa realizācija! Tas nav Ieejas informācijas atkārtojums, bet PĀRSTRĀDĀTA Ieejas informācija!

Izejas informācija var būt: 1 – uz printeri (pārskats, izdrukai noformētā informācija), 2 – uz failu (datu bāzes fails, e-pasts, .pdf fails, ...), 3 – uz ekrāna.

Uz ekrāna - tas var būt:

- datu apstrādes rezultāts (aprēķini, statistika, atvasinātā informācija),

- datu atlases rezultāts (sakārtotā, filtrētā informācija, vaicājumu rezultāts),
- ja nekas augstāk minēts nav – tad arī īpaši noformēta ievadinformācija.

Pēc skaita – minimāli trīs vienības. Katru izdrukas vai faila veidu nepieciešams nosaukt, aprakstīt struktūru un uzrādīt paraugu, kuru var izvietot pielikumos. Izejas informācijas piemērs ir dots 2. attēlā.

2.1.2. Izejas informācijas apraksts

1. **PDF izdruka** ar grupas audzēkņu sarakstu, kur lapas augšējā daļā ir norādīts grupas nosaukums, nodarbību grafiks, pasniedzējs un audzēkņu skaits, zemāk jāatrodas audzēkņu sarakstam tabulas formātā. Tabulā jābūt audzēkņa vārdam, uzvārdam un telefona numuram, kā arī audzēkņa vecāka vārdam, uzvārdam un telefona numuram.
2. **PDF izdruka** ar noliktavā pieejamu tērpu sarakstu, kur lapas augšējā daļā ir jābūt norādītam pieejamo tērpu skaitam, kam sekos tērpu saraksts tabulas formātā ar tērpu nosaukumiem, krāsām, izmēriem un attēliem.

2.att. Izejas informācijas apraksta piemērs

2.6.3. Ārējās informācijas apraksts

Ja programmas sistēmā ir paredzēta datu apmaiņa ar ārējām sistēmām, tās ir pilnībā jāapraksta. Tas varētu būt: pierakstīšanās programmā caur sociāliem tīkliem vai citām sistēmām (Facebook, Twitter, Google, internetbankas, www.latvija.lv, ...). Šādā gadījumā jāapraksta kādi dati tiek nodoti no/uz šīm sistēmām, kā notiek sistēmu savienošanās. KvD sistēma var būt daļa no kādās lielākas sistēmas. Tad ir jāuzrāda visu saistīto sistēmu savienošanās diagramma, jāapraksta tā, jāapraksta kopējo datu struktūra. Jā saites ar ārējām sistēmām nav – šīs paskaidrojošā raksta punkts izpaliek.

2.6.4. Funkcionālās prasības

Funkcionālām prasībām jābūt strukturētām, numurētām, viennozīmīgām un pabeigtām. Prasība jāformulē tā, lai to nekādā veidā nevarētu pārprast. Jāapraksta pilnīgi visas funkcijas, kas būs realizētas sistēmā, visas lietotāja iespējamās darbības, sistēmas reakcija, nosacījumi un ierobežojumi. Kopējais funkcionālo prasību skaits ir atkarīgs no sistēmas lieluma (apt. 3-5 lappuses).

Sadaļas sākumā detalizēti ir jāapraksta 3-5 sistēmas galvenās funkcionālās prasības (*piemēram, pasūtījuma noformēšana, sludinājuma publicēšana, viesnīcas numura rezervēšana utt.*) atbilstoši 3.attēlā dotajam paraugam.

2.2. Funkcionālās prasības

1. Pieteikšanās sistēmai

Identifikators	PS 2.2.1
Ievads	
Funkcija paredzēta, lai lietotājs varētu pieslēgties sistēmai, un varētu izmantot dotās sistēmas funkcijas	
Ievaddati	
<ol style="list-style-type: none">1. Lietotājvārds – obligāts lauks, kurš sastāv no sistēmas lietotāja vārda un uzvārda, kurš ir reģistrēts aktīvajā direktorijā.2. Parole – obligāts lauks, kura ir reģistrēta aktīvajā direktorijā.	
Apstrāde	
<ol style="list-style-type: none">1. Ja lietotājvārds un parole ir apstiprināta, lietotājs piekļūst sistēmai.2. Ja ir notikusi kļūda ievaddatos, tiek paziņota kļūda.	
Izvaddati	
<ol style="list-style-type: none">1. Ja lietotājvārds ar tādu paroli eksistē, tiek dota piekļuve sistēmai.2. Ja notikusi kļūda ievaddatos, tiek paziņota kļūda.	
Kļūdas paziņojumi	
<ol style="list-style-type: none">1. Ievadīts nepareizs lietotājvārds vai parole.	
Izsaucamās prasības	
Var veikt jebkurš lietotājs sistēmā, kuram ir piekļuve internetam.	

3.att. Galveno funkcionālo prasību apraksta piemērs

Pārējās funkcijas, kas būs realizētas sistēmā, var aprakstīt saīsinātā veidā (sk.4.attēlā)

3. Lietotāju kontu rediģēšana un dzēšana:

- 3.1. Jānodrošina, ka lietotājs, kuram ir pedagoga vai audzētāja tiesības var rediģēt tikai savu profilu.
- 3.2. Lietotājs ar vecāka tiesībām var rediģēt tikai savu vai sava bērna profilu.
- 3.3. Rediģēt un dzēst informāciju par jebkuru lietotāju var tikai administrators.
- 3.4. Lietotāju profilu rediģēšanai ir jāpiedāvā forma, kur jāievada jauni dati.
- 3.5. Ja formāts neatbilst ieejas informācijas apraksta 1. punktā minētajam, neļaut pievienot informāciju un izvadīt ziņojumu.
- 3.6. Pirms jaunas informācijas profila dzēšanas jāizvada brīdinājuma ziņojums ar iespēju saglabāt vai atcelt veiktās izmaiņas.

4.att. Funkcionālo prasību apraksta piemērs

2.6.5. Nefunkcionālās prasības

Nefunkcionālo prasību sadaļā jāapraksta visas sistēmas izstrādes prasības, kas neattiecas uz lietotājam sniegto servisu. Tās varētu būt prasības izstrādes procesam, ātrdarbībai, drošumam, pārnēsāmībai, u.t.t, kā arī lietotāja interfeisam. Šeit ir jādod vismaz 2 ekrānlogu skicē piemēri ar aprakstu. Skicēm jābūt uzzīmētām ar kādu zīmēšanas vai projektēšanas rīku, bet nedrīkst būt gatavi programmas ekrāna uzņēmumi (screenshot). Nefunkcionālo prasību piemērs ir dots 6. pielikumā.

2.7. Uzdevuma risināšanas līdzekļu izvēles pamatojums

Šajā sadaļā ir jāapraksta visas sistēmas izstrādāšanas procesā izmantotās programmēšanas valodas, izstrādes vides, čaulas, palīgprogrammas (piem., PHP, Laravel, Git, JetBrains PhpStorm, HTML, CSS, jQuery, Bootstrap, Less, Balsamiq Mockup, MySQL, Javascript, C#, Amazon S3, AngularJS, Isotope, .Net Framework, ASP.NET MVC, ...). Jāuzrāda rīku versijas un jādod īss to raksturojums.

Pirmajā rindkopā jāsniedz sistēmas īsa analīze: skaidrojums kāda tipa ir uzdevums un, attiecīgi, realizējamā sistēma, no kādām daļām tā sastāvēs (vai paredzēta izmantošanai pārlūkprogrammās, mobilajās ierīcēs, vai tā ir konkrēta žanra datorspēle, kas uzreiz nosaka savas izstrādes prasības, vai ir paredzēta video plūsmas apstrāde, mākslīgā intelekta elementi, nopietna datu apstrādes algoritmizācija, datu uzglabāšana/organizēšana datu bāzē, u.tml..., vai būs divas daļas: lietotāja puse (frontend) un servera puse (backend).

Tālāk jāseko pamatojumam, kāds rīks tiek izvēlēts, lai realizētu katru no apakšdaļām (nosaukums, versijas numurs, īss raksturojums).

Sadaļā nepieciešams uzrādīt pilnīgi visus aizgūtos programmas moduļus (plugin), piem. galerija, forums, izvēlne, ..., norādot to atrašanās vietas informācijas avotos. Aizgūto moduļu īpatsvars (programmas koda rindās) nevar pārsniegt $\frac{1}{3}$ no kopēja koda rindu skaita.

Uzdevuma risināšanas līdzekļu izvēles pamatojuma piemērs ir dots 7. pielikumā.

2.8. Programmatūras produkta modelēšana un projektēšana

Šajā sadaļā tiek aprakstīts pamata sistēmas izveidošanas process – tā projektēšana dažādos līmeņos. Visas shēmas jāizvieto uz gaiša fona un jāizmanto latviešu valoda.

2.8.1. Sistēmas arhitektūra

Jāuzrāda sistēmas iekšējā struktūra un sadalījums apakšsistēmās un moduļos. Ieteicams izmantot ilustrācijai Funkcionālās dekompozīcijas hierarhisko diagrammu (skat. 5. att.). Uzreiz pēc

virsraksta jābūt ievadtekstam ar minimālo aprakstu par sistēmas sadalījumu apakšsistēmās un atsauci uz diagrammas attēlu.

Apakšsistēmas ieteicams sadalīt pēc darbības sfērām (piem., Darbības ar precēm, Darbības ar grāmatām, Darbības ar Klientiem, ...). Var arī pēc gala lietotāja lomām (piem., Viesa darbības, Klienta darbības, Moderators darbības, ...). Apakšsistēmas jānosauc ar īsu (2-3 vārdu) nosaukumu, kas raksturotu visas šīs moduļu grupas funkcijas.

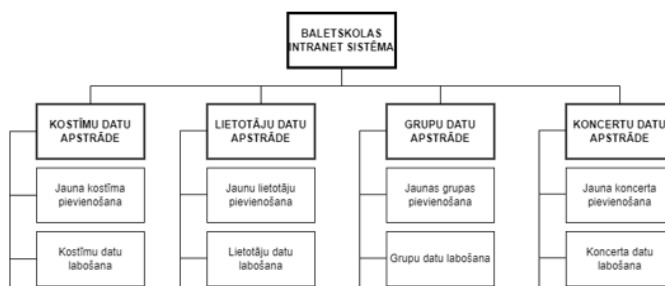
Pirmo un otro moduļu līmenis (sistēmu un apakšsistēmas) ir jāizceļ ar lielāka izmēra šriftu, trekno rakstu un augšējo reģistru (Bold, Caps). Var ar krāsu vai ēnu izcelt moduļus, kam turpmāk būs zīmētas detalizētas projektēšanas diagrammas DPD.

PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA

4.1. Sistēmas struktūras modelis

4.1.1. Sistēmas arhitektūra

Sistēma tiks veidota no četrām apakšsistēmām (skat. 7.att): kostīmu datu apstrādes apakšsistēmas, lietotāju datu apstrādes apakšsistēmas, grupu datu apstrādes apakšsistēmas un koncertu datu apstrādes apakšsistēmas.



5.att. . Funkcionālās dekompozīcijas diagrammas piemērs

Pēc diagrammas jāseko tās aprakstam. Jānosauc visas apakšsistēmas un tajās ietilpušie moduļi ar ļoti īsu aprakstu.

2.8.2. Sistēmas ER modelis

Jāsniedz sistēmas apstrādājamo datu konceptuālais modelis. Ieteicams izmantot Čena notācību ar saišu kardinalitāšu uzrādīšanu.

Pamatprasības diagrammai: Entitijas – lietvārds vienskaitlī, izceltas Bold, Caps, Kontūra 3* biezāka, nekā savienošanas līnijas, Atribūti – lietvārds vienskaitlī, saite – darbības vārds vai

lietvārds, kas apzīmē darbību. Iespējamās saites 1-1, 1-N, N-M. Ieteicams atstāt saiti daudzi pret daudziem. Nav ieteicams izmantot atribūtu ID – tas ir konceptuālais datu modelis, nevis fiziskais tabulu projektēšanas līmenis. (ER diagrammas paraugs ir pieejams 8.pielikumā).

Obligāti ir jāapraksta saites, kas ir starp entitijām diagrammā, attiecinot uz konkrētu būtību (skat. 6. att.). Aprakstīt, kādi atribūti ietilpst katrā entitijā nav obligāti, jo tas viennozīmīgi ir skaidrs no shēmas.

ER-modelis demonstrē entītijū savstarpējo saistību:

- starp entītijām “Lietotājs” un “Loma” ir attiecība daudzi pret daudziem, jo viena loma var būt vairākiem lietotājiem, bet, no otras puses, vienam lietotājam var būt vairākas lomas;
- starp entītijām “Publikācija” un “Komentārs” ir attiecība viens pret daudziem, jo viens komentārs var piederēt tikai vienai publikācijai, bet, no otras puses, vienai publikācijai var būt vairāki komentāri;

6.att. . ER-diagrammas saišu apraksta piemērs

Ja tiek izmantota cita notācija, tās notācijas nosaukums ir jāuzrāda, kā arī jāuzrāda informācijas avots, no kurienes ir ņemts tās apraksts.

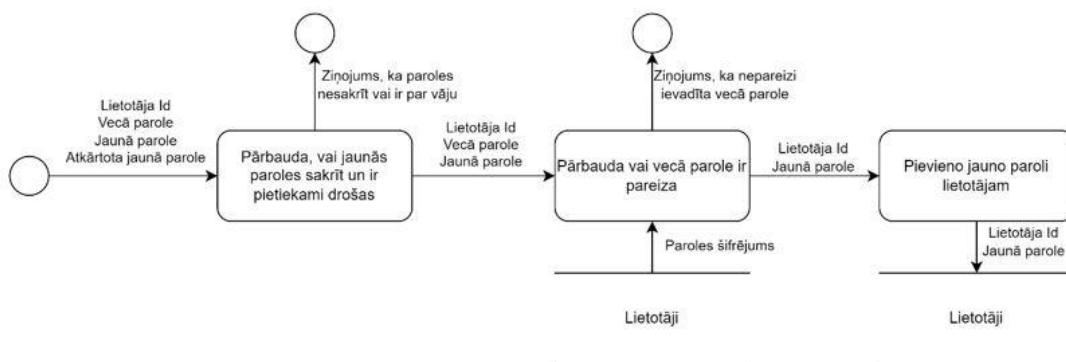
2.8.3. Datu plūsmu modelis

Šeit ir jādod detalizēts komponentu projektējumu apraksts. No moduļu saraksta, kas tiek noteikts sistēmas arhitektūras punktā, tiek izvēlēti daži (ap 7-10) svarīgāki procesi, kas ir būtiski datu apstrādē izstrādājamā sistēmā un kas var raksturot programmēšanas pieejas oriģinalitāti un prasmes. Tiem tiek detalizēti aprakstīti visi datu transformācijas un apstrādes procesi, kas notiek moduļos. Datu plūsmas diagrammas tiek zīmētas tikai detalizētā (apakšējā) līmenī un ir ieteicamas ar procesu skaitu no 3 līdz 6. Obligāti jābūt detalizētam datu apstrādes procesa tekstuālam aprakstam, kas ir neatņemama diagrammas sastāvdaļa.

Diagrammas ir jānumurē, jāuzrāda moduļa nosaukums un atsauce uz diagrammas attēlu (skat. 7. att.).

5. Paroles labošanas modulis (skat. 4.5. att.).

Lietotājs ievada veco paroli, jauno paroli un atkārotu jauno paroli. Tad tiek pārbaudīts, vai jaunā parole abas reizes ievadīta vienādi un vai tā ir pietiekami sarežģīta. Pēc tam tiek pārbaudīta, vai vecā parole ir ievadīta pareizi. Ja kāds no procesiem bija neveiksmīgs, tiek izvadīts kļūdas ziņojums, citādi lietotājam tiek saglabāta jaunā parole.



4.5. Paroles labošanas moduļa datu plūsmas diagramma

7.att. Datu plūsmas diagrammas piemērs

Datu apstrādes algoritma apraksts (diagrammas aprakstošā daļa) ir jāizveido visiem vai lielākajai daļai funkcionālās dekompozīcijas diagrammas moduļiem, bet vizuālā daļa ir jāveido tikai izvēlētiem 7-10 moduļiem.

2.9. Datu struktūru apraksts

Šajā punktā ir jāapraksta sistēmā izmantojamo datu organizācija loģiskā līmenī. Ja tiek izmantota datu bāze – tad ir jādod datu bāzes struktūras apraksts. Ja ir paredzēta cita veida datu glabāšana – tad ir jāizmanto citas metodes (ieskaitot diagrammas un detalizētu aprakstu), lai aprakstītu datu organizāciju. Šeit ir dots piemērs relāciju datu bāzes datu struktūras aprakstam.

Vispirms ir jāsniedz īss apraksts no cik tabulām sastāv datu bāze. Jābūt atsaucei uz tabulu saišu shēmas attēlu. Jādod īss tabulu pārskaitījums ar tabulu nosaukumiem un glabājamo datu būtību (skat. 8.att.).

5. DATU STRUKTŪRU APRAKSTS

Datubāzes izveidošanai ir nepieciešama datubāzes fiziskā struktūra. Tā tika iegūta normalizējot ER diagrammu, kuras rezultātā no entitijām izveidoja relācijas. 5.1. attēlā ir redzama datubāzes tabulu saišu shēma. Kā redzams datubāze sastāv no 11 tabulām, kas satur informāciju par lietotājiem, publikācijām, publikāciju veidiem, birkām, komentāriem, programmēšanas valodām un lomām.

- Tabula “**Users**” glabās informāciju par reģistrētiem lietotājiem.
- Tabula “**Posts**” glabās informāciju par publikācijām.
- Tabula “**PostTypes**” glabās informāciju par publikāciju veidiem.
- Tabula “**Comments**” glabās informāciju par publikāciju komentāriem.
- Tabula “**ProgrammingLanguages**” glabās informāciju par programmēšanas valodām.

8.att. Datu plūsmas diagrammas piemērs

Tālāk katrai tabulai jāuzrāda “tabuliņu” ar detalizētu informāciju par tabulas lauku nosaukumiem, tipiem, garumiem un glabājamās vērtības aprakstu. Var uzrādīt vai lauka vērtībai jābūt/nav jābūt unikālai, nav jābūt tukšai (required). Obligāti jāuzrāda tabulas atslēgas lauks. Starp tabuliņām jābūt tekstam, kas apraksta konkrētas tabulas saistību ar citām tabulām, uzrādot caur kuriem laukiem tās tiek realizētas. Ja tabula nav saistīta ne ar vienu citu tabulu, tad tā arī jāuzrāda. Piemērs tabuliņu noformēšanai ir dots 9. attēlā.

Tabula “pedagogsgrupa” satur informāciju par lietotājiem ar lomu “pedagogs” piešķirtajām grupām: identifikatoru, nodarbību skaitu nedēļā, pedagoga personas kodu un grupas nosaukumu.

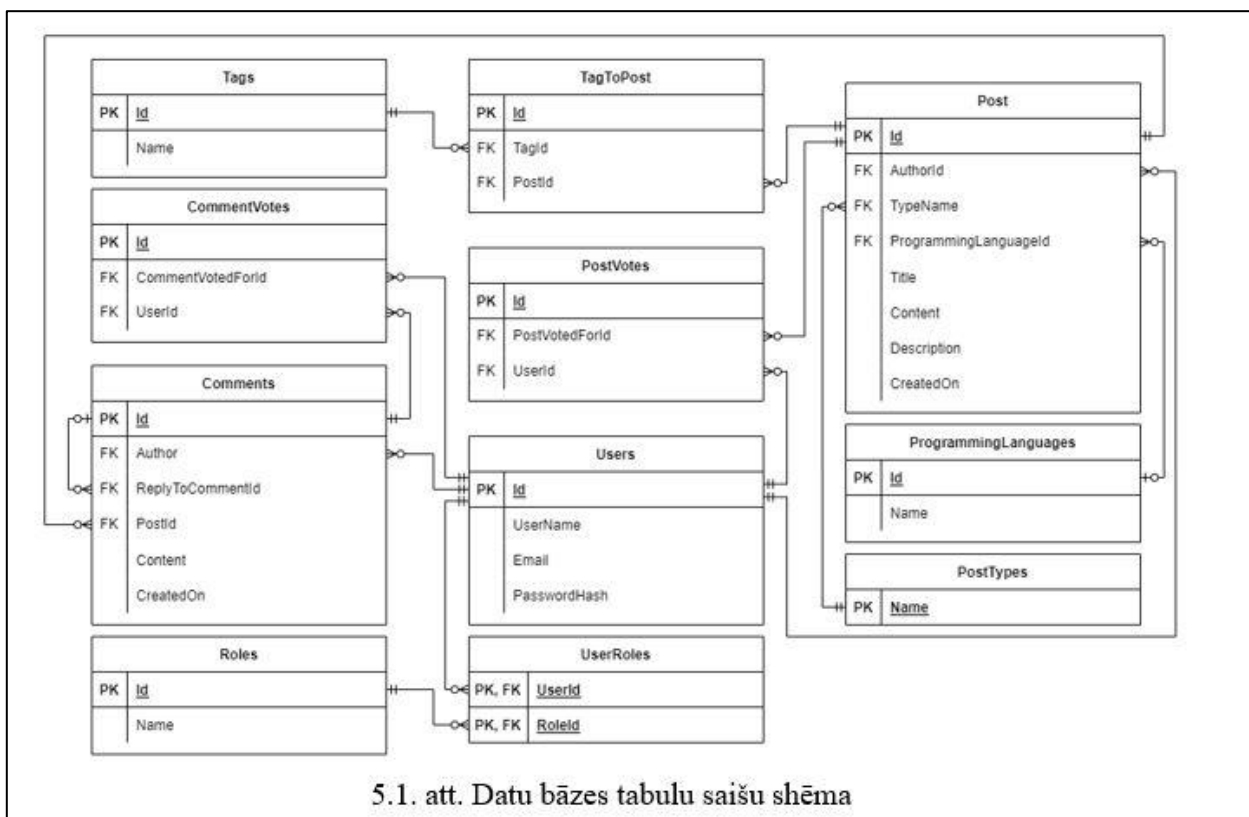
7.tabula

Tabulas “**PedagogsGrupa**” struktūra

Nr	Lauka nosaukums	Tips	Izmērs	Apraksts
1.	PedagogsGrupaID	INT	-	Ieraksta identifikators
2.	NodSk	INT	-	Nodarbību skaits, ko pedagogs novada konkrētajai grupai vienā nedēļā
3.	PedagogaPersonasKods	CHAR	12	Pedagoga personas kods
4.	GrupasPedagogaNosaukums	VARCHAR	5	Grupas nosaukums

9. att. Datu bāzes “tabuliņu” noformēšanas piemērs

Tabulu saišu shēmai vēlams jāatrodas uzreiz pēc tabulu pārskaitījuma. Piemēru var redzēt 10. attēlā. Ja tā ir ļoti liela un objektīvi neietilpst lapā, to ir iespējams izvietot pielikumā. Ja starp “tabuliņām” bija dots katras tabulas saišu ar citām tabulām apraksts, tad to var uzskatīt par tabulu saišu shēmas aprakstu. Citādāk tās ir jādod atsevišķi. Saitēm starp tabulām jābūt precīzi lauks pret lauku ar skaidri saprotamu kardinalitāti (saišu svaru).



5.1. att. Datu bāzes tabulu saišu shēma

10.att.Tabulu saišu shēmas piemērs

2.10. Lietotāja ceļvedis

2.10.1. Sistēmas prasības aparatūrai un programmatūrai

Ir jāapraksta visas prasības datorā vai mobilajā ierīcē, lai varētu veiksmīgi palaist un darbināt jūsu izstrādātu programmsistēmu (*paraugu sk. 9.pielikumā*). Atsevišķi var aprakstīt izmantojamo operētājsistēmu, tās versijas, ka arī uzrādīt nepieciešamo brīvo vietu cietajā diskā vai procesora ātrumu.

2.10.2. Sistēmas instalācija un palaišana

Ir jādod sistēmas instalācijas un palaišanas apraksts (*paraugu sk. 9.pielikumā*). Ja programmas nodošanas periodā (2 nedēļas pirms Kvalifikācijas eksāmena) tā būs izvietota web

domēnā konkrētā adresē, tad šajā sadaļā var uzrādīt šo adresi un loginu un paroli no visām sistēmā izmantojamām lomām.

2.10.3. Programmas apraksts

Ir jāapraksta visas sistēmas katrs lietotāja saskarnes (GUI) elements ar visām tajā iespējamām darbībām un serviss. Katram GUI elementam jādod ekrāna paraugs ar ievadinformācijas piemēriem. Tas ir garākais punkts dokumentā. Ekrāna uzņēmumi nevar sekot viens aiz otra, starp tiem obligāti jābūt tekstam. Katram attēlam jābūt atsaucei tekstā. Jābeidz šī sadaļa ar norādījumiem, kā pabeigt darbību ar jūsu programmu. Programmas apraksta paraugs ir pieejams 10.pielikumā.

2.11. Programmatūras lietojamības testēšana

Šajā sadaļā ir jāuzrāda 3-5 funkcionālo prasību testēšanas procesa pārskati. Katrai prasībai ir jāuzrāda 2-4 testi ar pareiziem/paredzētiem un nepareiziem/neparedzētiem ieejas/izejas datiem. Testēšanas rezultātus uzrādīt tabulā (piemēru skat. 11.attēlā).

7. PROGRAMMATŪRAS LIETOJAMĪBAS TESTĒŠANA		
Lietotāja loma:	lietotājs ar administratora tiesībām	
Prasības identifikators :	PS 2.2.2	
Prasības apraksts		
Jānodrošina, ka jaunu lietotāja kontu var pievienot tikai lietotājs ar administratora lomu.		
Ievaddati/situācijas apraksts	Sagaidāmais rezultāts	Statuss
1.Jaunā lietotāja kontu pievieno lietotājs ar administratora tiesībām.	konts tiek veiksmīgi pievienots	pareizi
2.Jaunā lietotāja kontu pievieno lietotājs ar audzēkņa tiesībām.	konts nav pievienots	pareizi
3.Jaunā lietotāja kontu pievieno lietotājs ar vecāku tiesībām.	konts nav pievienots	pareizi

11.att.Testēšanas pārskata piemērs

Katrā testēšanas pārskatā jābūt uzrādītam prasības numuram no Funkcionālās prasības (2.2.) punkta, prasības īsam nosaukumam, Ievadāmās informācijas aprakstam vai citu manipulāciju aprakstam, sagaidāmam rezultātam, testēšanas statusam (vai sagaidāmais rezultāts sakrīt ar iegūto).

2.12. Nobeigums

Parasti ir divās rindkopās:

- konstatācija, kas ir realizēts Kvalifikācijas darba izstrādes laikā (laikam, tas, kas bija sākumā formulēts, kā uzdevums ☺), kādās valodās, cik labi;
- kur tas tiek pašlaik izmantots. Kur to varēs izmantot turpmāk. Vai varēs attīstīt uz kaut ko lielāku?

Apjoms līdz ½ lpp. Sadaļas paraugs ir pieejams 11.pielikumā.

2.13. Informācijas avoti

Ir jāapraksta vismaz 9 oficiāli informācijas avoti, ko jūs izmantojāt sistēmas izstrādes laikā. Sākumā tiek ievietotas drukātas grāmatas vai drukātie periodiskie izdevumi latviešu valodā, pēc tam angļu valodā, pēc tam citās valodās.

INFORMĀCIJAS AVOTI

1. L.Zaiceva. Programmatūras izstrādes tehnoloģija. Rīga : RTU, 2002. – 204 lpp.
2. Problēmas analīze, programmatūras specifikācija un darba plānošana. – <https://skolo.lv/course/view.php?id=19577§ion=5>. – (Resurss apskatīts 23.10.2023.).
3. [ANG] Unity 3D dokumentācija - <https://docs.unity3d.com/Manual/index.html>. – (Resurss apskatīts 24.05.2023.).

12. att. Informācijas avotu piemērs

Indormācijas avotus noformē saskaņā ar [Hārvardas sistēmu \(Harvard system\)](#).

Titullapas paraugs

RĪGAS VALSTS TEHNIKUMS

DATORIKAS NODAĻA

Izglītības programma: Programmēšana

KVALIFIKĀCIJAS DARBS

“Tēma”

Paskaidrojošais raksts ____ lpp.

Audzēknis:

Jānis Bērziņš

Prakses vadītājs:

Anna Ozoliņa

Nodaļas vadītājs:

Normunds Barbāns

Rīga 2025

Anotācijas paraugs latviešu valodā

ANOTĀCIJA

Baletscolas kostīmu uzskaites sistēma ir izstrādāta, lai efektīvi pārvaldītu skolēnu, vecāku un pedagogu datus, kā arī organizētu un vadītu koncertus un to numurus. Sistēmas izstrādes laikā tika izmantotas tādas tehnoloģijas kā HTML5, CSS Bootstrap 3.4, JavaScript, PHP 8.1, Laravel 10 un MySQL. Papildus tika izmantota DomPDF bibliotēka, lai ģenerētu PDF dokumentus no HTML un CSS koda.

Kvalifikācijas darbs ietver ievadu, uzdevumu nostādni, prasību specifikāciju, uzdevuma risināšanas līdzekļu izvēles pamatojumu, programmatūras produkta modelēšanas un projektēšanas aprakstu, datu struktūru aprakstu un secinājumus. Ievadā ir aprakstīta baletscolas kostīmu uzskaites sistēmas aktualitāte. Uzdevumu nostādnē ir norādīti uzdevumi, kurus sistēmai būs nepieciešams veikt. Prasību specifikācija sastāv no ieejas un izejas informācijas, kā arī no sistēmas funkcionālajām un nefunkcionālajām prasībām. Uzdevuma risināšanas līdzekļu izvēles pamatojumā ir norādīti izstrādāšanai izmantotie līdzekļi un to pielietojums. Programmatūras produkta modelēšanas un projektēšanas apraksts ietver sistēmas struktūras modeli un funkcionālo sistēmas modeli. Datu struktūru aprakstā tiek parādīta datu bāzes relāciju shēma, kā arī tabulu struktūra ar datu tipu un datu garuma norādīšanu. Lietotāja ceļvedī ir norādītas sistēmas prasības aparatūrai un programmatūrai, instalācija un palaišana, kā arī programmas lietošanas pamācība. Testa piemērā ir dots detalizēts autentificēšanās un jauna konta izveides apraksts ar vizuāliem attēliem.

Kvalifikācijas darbs sastāv no 120 lappusēm, kurās ietilpst 81 attēli, 15 tabulas un 1 pielikuma. Pielikums satur programmas pirmkodu fragmentus.

Satura rādītāja paraugs

SATURS	
IEVADS.....	5
1. UZDEVUMA NOSTĀDNE	6
2. PRASĪBU SPECIFIKĀCIJA	8
2.1. Ieejas un izejas informācijas apraksts	8
2.1.1. Ieejas informācijas apraksts	8
2.1.2. Izejas informācijas apraksts.....	9
2.2. Funkcionālās prasības	10
2.3. Nefunkcionālās prasības	15
3. UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS	16
4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA	17
4.1. Sistēmas struktūras modelis	17
4.1.1. Sistēmas arhitektūra	17
4.1.2. Sistēmas ER modelis	19
4.2. Funkcionālais sistēmas modelis	21
4.2.1. Datu plūsmu modelis	22
5. DATU STRUKTŪRU APRAKSTS	28
6. LIETOTĀJA CEĻVEDIS	34
6.1. Sistēmas prasības aparatūrai un programmatūrai	34
6.2. Sistēmas instalācija un palaišana	34
6.3. Programmas apraksts	35
7. PROGRAMMATŪRAS LIETOJAMĪBAS TESTĒŠANA.....	45
NOBEIGUMS	48
INFORMĀCIJAS AVOTI	49
PIELIKUMI	50
1. pielikums. Pārskatu piemēri	51
2. pielikums. Ekrānattēlu piemēri	54
2. pielikums. Programmas pirmteksts	60

Ievada sākuma paraugs

IEVADS

Šobrīd ir ļoti svarīgi izmantot vienoto digitalizēto risinājumu, kas ļauj viegli pārvaldīt datus un procesus iekš kādas iestādes, tajā skaitā deju skolās, kolektīvos un teātros. Uz doto brīdi iepriekš minētajām organizācijām nav pieejama sistēma, kas minimizētu sarežģījumus tērpu izsniegšanas un savākšanas posmā, grupu komplektēšanas laikā, kā arī koncerta programmas sastādīšanas un pasākuma organizācijas periodā. Lai novērstu šīs problēmas un izvairītos no cilvēciskajām kļūdām informācijas apstrādes un uzglabāšanas laikā, ir nepieciešams izstrādāt WEB aplikāciju, kas ļaus darbiniekiem un audzēkņiem piekļūt viņiem piemērotas informācijas un funkcionalitātes klāstam. Piekļuve centralizētai informācijas datu bāzei veicinās efektīvāku sadarbību starp skolotājiem, audzēkņiem un vecākiem, kā arī uzlabos vispārējo organizācijas darba plūsmu.

Tirgus izpētes un teātriem, deju kolektīviem un sporta komandām pieejamo risinājumu analīzes laikā, tika secināts, ka lielam kolektīvam piemērota tērpu pārvaldes tehnoloģija šobrīd nav pieejama tirgū. Izpētot pieredzes stāstus un atsauksmes, ar kurām teātru direktori un kolektīvu vadītāji dalījās tīmeklī, tika atklāts, ka lielākā daļa ar skatuves mākslu saistīto institūciju seko līdzi savām kostīmu krātuvēm, izmantojot Excel tabulas vai papīra žurnālus, retos gadījumos tiek izmantota kāda inventāra pārvaldes sistēma. Šāda situācija norāda uz būtisku robu digitālajos risinājumos, kas varētu palīdzēt skatuves mākslas iestādēm efektīvāk pārvaldīt savus resursus. Tradicionālie uzskaites veidi bieži vien rada papildu administratīvo slogu un palielina kļūdu iespējamību, īpaši lielās organizācijās, kur ir daudz dalībnieku un inventāra vienību.

Viens no tādas sistēmas piemēriem, ko izmanto kādā no ASV teātriem, ir StageStock. Šī ir viena no visveiksmīgāk izveidotajām inventarizācijas sistēmām, kas šobrīd ir pieejama lietošanai. Mākoņu risinājums StageStock ļauj administratoram samērā viegli veikt kādas institūcijas īpašumā esošo tērpu, aksesuāru un rekvizītu uzskaiti, nodrošinot ērtu vienību pievienošanu, pārvaldīšanu un meklēšanu. Starp stiprajām pusēm ir jāpiemin arī iespēja pievienot jebkāda veida failus vienības aprakstam un apskatīt tās vēsturi. Šo funkcionalitāti ir mērķis ieviest arī projektā - skolas sistēmā. Vislielākais trūkums ir tas, ka pirms lietošanas sistēmas lietotne ir jālejupielādē un tā ir pieejama tikai veicot ikmēneša maksājumu. Turklāt, sistēmas lietošanas izmaksas var kļūt par būtisku šķērslī, īpaši mazām organizācijām ar ierobežotiem budžetiem, kas ne vienmēr var atļauties regulāras abonēšanas maksas.

Uzdevuma nostādnes paraugs

1. UZDEVUMA NOSTĀDNE

Kvalifikācijas darba uzdevums ir izveidot baletskolas kostīmu uzskaites sistēmu. Sistēmā nepieciešams realizēt iespēju glabāt un pārvaldīt datus par skolas pedagogiem, audzēkņiem un viņu vecākiem, kā arī nodarbību grafikiem un kostīmiem. Pedagogiem un administrācijai ir jābūt iespējai ērti piekļūt sarakstiem ar saviem audzēkņiem, koncertiem, nodarbībām un noliktavā esošiem kostīmiem, audzēkņiem un vecākiem jābūt nodrošinātai piekļuvei informācijai par saviem pedagogiem, izpildāmiem numuriem un tiem nepieciešamiem tērpiem. Papildus iepriekš minētajām prasībām skolas direktoram ir jānodrošina iespēja pievienot un pārvaldīt informāciju par koncertiem, numuriem, tiem piesaistītiem audzēkņiem, pedagogiem un tērpiem. Skolas personālam ir jābūt iespējai veikt nepieciešamās informācijas izdrukas ērtai piekļuvei vajadzīgiem datiem.

Ir plānotas vairākas funkcijas:

- skolas direktoram (*administratoram*) būs iespēja pārvaldīt informāciju par pedagogiem, audzēkņiem, viņu vecākiem, koncertiem utt. ;
- skolas pedagogam būs iespēja:
 - piekļūt informācijai par savu grafiku, grupām, to audzēkņiem un viņu vecākiem;
 - piekļūt informācijai par koncertiem, tajos iekļautiem numuriem, izpildītājiem un nepieciešamiem kostīmiem;
- skolas audzēknim būs iespēja:
 - piekļūt informācijai par savu grupu, nodarbību sarakstu, pedagogu un kostīmiem;
 - apskatīt informāciju par koncertiem un numuriem, kuros audzēknis piedalās un tiem nepieciešamiem tērpiem.

Informācija par sistēmas lietotājiem un to mijiedarbību ar sistēmu tiek apkopota

Lietojumgadījumu diagramma (USE CASE diagramm), kas ir pieejama 3.pielikumā, atspoguļo detalizēto informāciju sistēmas lietotāju mijiedarbību ar sistēmu.

Nefunkcionālo prasību apraksta paraugs

2.3. Nefunkcionālās prasības

1. Sistēmas saskarnes valodai ir jābūt latviešu valodai.
2. Jānodrošina tas, ka tīmekļa lietojumprogramma automātiski pielāgojas un saglabā ērtu lietojamību dažādos ekrāna izmēros.
3. Sistēma ir jābūt saderīgai ar populārākajiem tīmekļa pārlūkiem, piemēram, Chrome, Firefox, un Safari.
4. Dizainam ir jābūt zilos toņos.
5. Sistēmai jāsniedz lietotājam informatīvus paziņojumus par veiksmīgiem vai neveiksmīgiem procesiem, nodrošinot skaidru lietojamību.
6. Lietotāja pieslēgšanās formai jāizskatās sekojoši (skat. 2.att).

The image shows a sketch of a login form. It is a rounded rectangle with a light blue background. At the top, the title 'Pieslēgšanās' is centered in a bold, dark blue font. Below the title, there are two input fields, each with a light blue border and a light blue background. The first field contains the text 'personas kods' in a light gray font. The second field contains the text 'parole' in a light gray font. Below these two fields, there is a button with a light blue border and a light blue background, containing the text 'ienākt' in a bold, dark blue font.

2.att. Lietotāja pieslēgšanas formas skice

Uzdevuma risināšanas līdzekļu izvēles pamatojuma piemērs

3. UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS

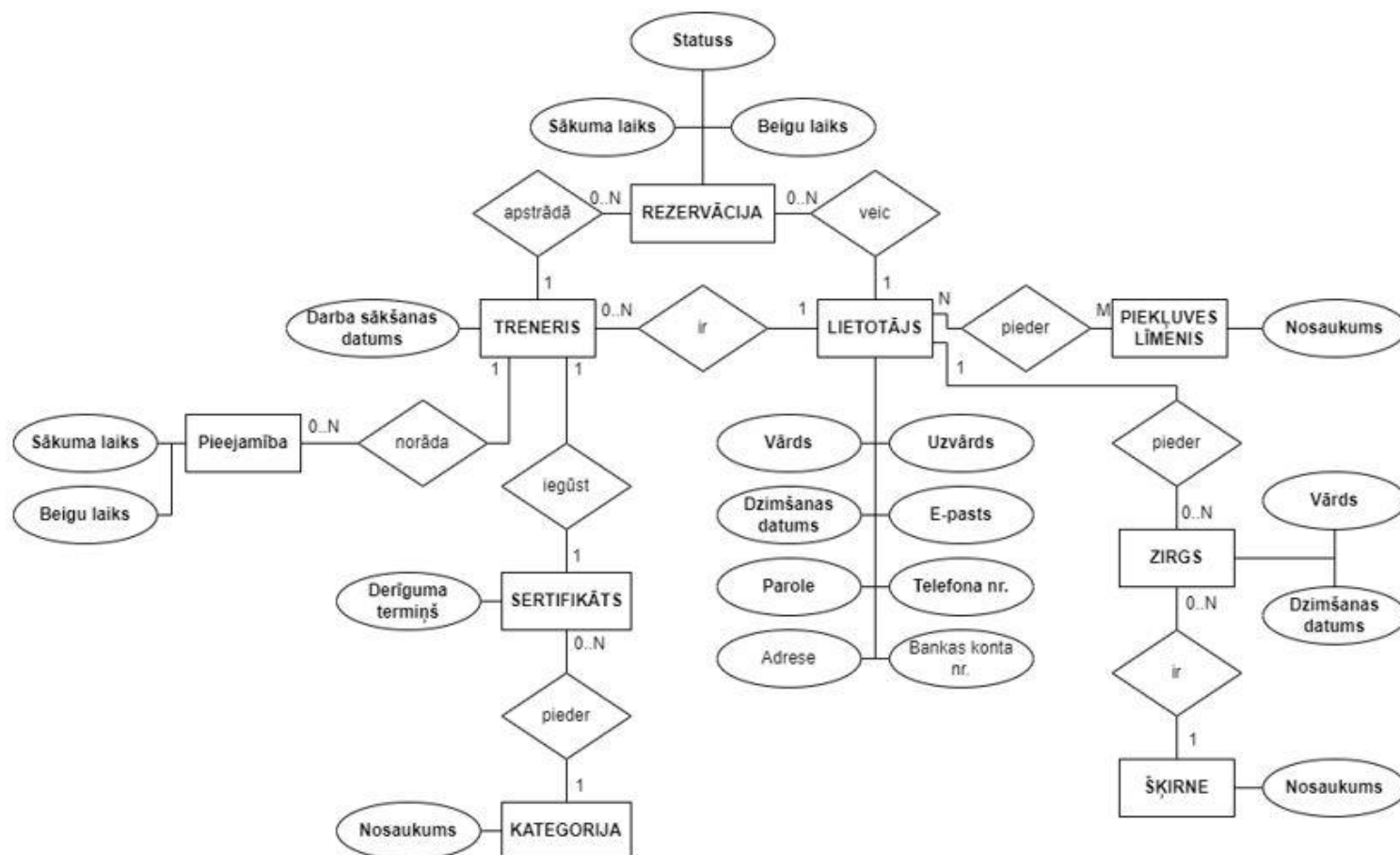
Baletskolas kostīmu uzskaites sistēma ir web aplikācija, kas paredzēta dalībnieku, kostīmu un koncertu datu uzskaitēi baletskolas ietvaros. Tā sastāv no divām galvenajām daļām: lietotāja daļas un servera daļas.

Lietotāja daļas izstrādē tiks pielietotas trīs galvenās tehnoloģijas: **HTML5**, kas tiek izmantota modernās un dinamiskās tīmekļa lapas struktūras nodrošināšanai, **CSS Bootstrap 3.4**, kas ļauj viegli pielāgot izskatu un izkārtojumu, piedāvājot responsīvu dizainu un saglabājot konsistentu izskatu visās ierīcēs, **DomPdf**, kas ir PHP bibliotēka, kura ļauj ģenerēt PDF dokumentus no HTML un CSS koda, nodrošinot vienkāršu un efektīvu veidu, kā izveidot drukājamās dokumentācijas tieši no tīmekļa lapām, kā arī **JavaScript**, kas nodrošina interaktivitāti un ērtu ziņojumu sūtīšanu un saņemšanu, ietekmējot sistēmas dinamiku, kopumā šīs tehnoloģijas ļauj veidot patīkamu lietotāju pieredzi.

Servera puses izstrādē, savukārt, tiks izmantotas četras sekojošās tehnoloģijas: **PHP8.1** programmēšanas valodas jaunākā versija, kas izmanto jaunākās tehnoloģijas, lai veiksmīgi un droši apstrādātu datus servera pusē, PHP ietvars **Laravel 10**, kas ļauj ātri izstrādāt dinamiskās tīmekļa lietotnes, izmantojot iebūvētās funkcijas, kā arī **MySQL**, kas tiek izmantots kā datubāzes pārvaldības sistēma, nodrošinot efektīvu un drošu datu uzglabāšanu, kopā minētās tehnoloģijas nodrošina modernu lietotāja pieredzi un efektīvu datu pārvaldību.

Sistēmas izstrādes procesā tiks izmantotas divas **izstrādes vides**: **Visual Studio Code**, kas ir atvērta koda izstrādes vide, paredzēta tīmekļa lietojumprogrammu izveidei, tā nodrošina daudzas funkcijas un papildinājumus, kas atvieglo koda rakstīšanu, atklāšanu un projektu pārvaldību, kā arī **Laragon**, kas ir lokāla izstrādes vide MySQL un PHP, tā ļauj ātri iestatīt lokālu vidi izstrādei un testēšanai, pirms sistēma tiek publicēta tiešsaistē, šī vide nodrošina vienkāršu konfigurāciju un drošu darbību, padarot izstrādes procesu ērtu un efektīvu.

ER diagrammas piemērs



1.1. att. Sistēmas ER-diagramma

Lietotāja ceļveža piemērs

6. LIETOTĀJA CEĻVEDIS

6.1. Sistēmas prasības aparatūrai un programmatūrai

Baletscolas kostīmu uzskaites sistēma ir tīmekļa lietotne, kas ir piemērota ērtai lietošanai lielākajā daļā ierīču, ja tās atbilst sekojošām prasībām:

- Operētājsistēma:
 - Windows 10 vai jaunāka versija;
 - macOS 10.15 vai jaunāka versija;
 - jebkura moderna Linux versija.
- Interneta Pārlūks:
 - Google Chrome 89 vai jaunāka versija;
 - Mozilla Firefox 86 vai jaunāka versija;
 - Safari 14 vai jaunāka versija;
 - Microsoft Edge 89 vai jaunāka versija.

Ir jāņem vērā, ka lai ierīces spētu efektīvi darboties ar baletscolas kostīmu uzskaites sistēmu, nodrošinot lietotājiem optimālu lietotāja pieredzi ir jāparliecinās, ka izmantojamais pārlūks atbalsta JavaScript.

6.2. Sistēmas instalācija un palaišana

Lai instalētu un palaistu sistēmu lietotājam vispirms ir jāparupējās par sekojoša programmnodrošinājuma instalēšanu uz izmantojamās ierīces:

- Visual Studio Code
- Laragon (vai cits lokālās izstrādes serveris)
- Node.js
- Composer
- Git

1. Projekta klonēšana no GitHub:

1.1. Atveriet terminālu un izpildiet komandu (skat. 14.att. Git Clone);

```
git clone https://github.com/pikcrvt-prakse/dvb-2024-KatrinaGandzjuka.git
```

14.att. Git Clone

1.2. Ieejiet projekta mapē (skat. 15.att. Projekta mape).

```
cd dvb-2024-KatrinaGandzjuka/BaletscolasIntranets
```

15.att. Projekta mape

Programmas apraksta piemērs

6.3. Programmas apraksts

Autentifikācija.

Atvērot tīmekļa lietotni pārlūkprogrammā, lietotājam tiek piedāvāts ievadīt savu personas kodu un paroli, ko viņam piešķir administrators (skat. 22.att.).

22.att. Pieslēgšanās forma

Pēc paroles un personas koda ievades lietotājam jānospiež poga "Ienākt", ja visi dati tika ievadīti korekti, lietotājs tiks pāradresēts uz viņa lomas funkcionalitātei piemērotu lapu.

Administrātora funkcionalitāte, jauna lietotāja pievienošana.

Veiksmīgās autentifikācijas gadījumā lietotājs ar administrātoru lomu tiek pāradresēts uz jauna lietotāja pievienošanas lapu (skat. 23.att.).

23.att. Jauna lietotāja pievienošanas forma

Pēc jauna lietotāja informācijas ievades administratoram ir nepieciešams uzpiezt pogu "Pievienot lietotāju", lai lietotājs tiktu pievienots sarakstam.

Nobeiguma piemērs

NOBEIGUMS

Kvalifikācijas darba rezultātā ir izveidota daudzfunkcionāla sistēma lietotāju kontu pārvaldībai, grupu un kostīmu administrēšanai, kā arī koncerta numuru plānošanai un izpildei. Sistēma piedāvā plašu funkcionalitāti, nodrošinot iespēju lietotājiem reģistrēties, autorizēties, rediģēt un dzēst profilus, pievienot un pārvaldīt grupas, kostīmus un koncertus, kā arī nodrošināt piekļuvi informācijai par lietotājiem, kostīmiem un numuriem. Sistēma tika izstrādāta, izmantojot programmēšanas valodas, piemēram, PHP un JavaScript, kā arī HTML un CSS priekš lietotāja saskarnes izstrādes. Datu bāzes pārvaldībai tika izmantots MySQL. Realizētās funkcijas ietver jaunu lietotāju kontu pievienošanu, lietotāju autorizāciju, profilu rediģēšanu un dzēšanu, grupu un kostīmu pievienošanu un pārvaldību, koncerta numuru plānošanu un izpildītāju pievienošanu, kā arī detalizētu atskaišu un izdruku ģenerēšanu PDF formātā.

Sistēma pašlaik var tikt izmantota izglītības iestādē, lai efektīvi pārvaldītu skolēnu, vecāku un pedagogu datus, kā arī organizētu un vadītu koncertus un to numurus. Turpmāk to varēs izmantot citās izglītības iestādēs vai organizācijās, kur nepieciešama līdzīga funkcionalitāte. Sistēmu iespējams attīstīt tālāk, pievienojot papildus funkcionalitāti, kā piemēram, sekot līdzi kostīmu skaitam noliktavā, ļaut veikt zāles rezervāciju, izmantojot šo sistēmu, pielāgot pdf failu saturu savam vajadzībām, kā arī uzlabotu lietotāju saskarni un mobilo aplikāciju integrāciju. Šādi paplašinājumi ļaus vēl vairāk palielināt sistēmas efektivitāti un pielietojuma iespējas.