

**Profesionālās izglītības kompetences centrs
„Rīgas Valsts tehnikums”
Izglītības programma: Programmēšana**

KVALIFIKĀCIJAS DARBS

**Latvijas Nacionālās bibliotēkas programma objektu
importēšanai**

Paskaidrojošais raksts 44lpp.

Audzēknis:

Artūrs Adamovičs

Vadītājs:

Māris Danne

Normu kontrole:

Olga Sabanska

**Rīga
2021**

ANOTĀCIJA

Kvalifikācijas darbā ir aprakstīts masu objekta importēšanas programmas izstrādes process. Programma darbiniekiem ļauj apkopotos un apstrādātos datus par vēsturiskiem objektiem importēt vienlaicīgi sistēmā, kā arī pirms tā veikt informācijas precizitātes pārbaudi un rediģēšanu pirms importēšanas. Programma tika izstrādāta C# programmēšanas valodā, izmantojot WPF ietvaru, un Latvijas Nacionālās bibliotēkas relāciju datubāzi.

Kvalifikācijas darbs ietver ievadu, uzdevumu nostādni, prasību specifikāciju, uzdevuma risināšanas līdzekļu izvēles pamatojumu, programmatūras ceļvedi, nobeigumu un pielikumus. Kvalifikācijas darba ievadā aprakstīts Latvijas Nacionālās bibliotēkas digitālo objektu veidotāju aktuālā problēma un tās risinājums. Uzdevumu nostādnē ir norādītas funkcijas, kuras programmai 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, kādi līdzekļi tiks izmantoti izstrādāšanai, un kādiem nolūkiem tie tiek izmantoti. Programmatūras produkta modelēšanas un projektēšanas apraksts sastāv no sistēmas struktūras modeļa, kas ietver sistēmas arhitektūru un ER modeli, un funkcionālās sistēmas modeļa, kas satur datu plūsmu modeli. Datu struktūru apraksts parāda datu bāzes relāciju shēmu. Kā arī tabulu struktūra ar aprakstu, kas ietver datu tipu un datu garumu norādīšanu. Lietotāja ceļvedī ir norādītas nepieciešamās programmas prasības aparatūrai un programmatūrai, programmas palaišana, kā arī programmas pamācība un apraksts, kas parāda, kā pareizi lietot programmu. Testa piemērā ir detalizēts objektu sagatavošanas un importēšanas apraksts ar vizuāliem attēliem.

Kopumā kvalifikācijas darba apjoms ir 48lpp., kurā ietilpst attēli, tabulas un pielikumi.

ANNOTATION

The qualification job describes the process of developing a program for mass importing objects. The program allows employees to import aggregated and processed data about historical objects simultaneously into the system, and to perform a verification and editing of the accuracy of information prior to import. The program was developed in the C# programming language using the WPF framework and the relationship database of the Latvian National Library.

The qualification work shall include input, task framework, requirement specification, justification for the selection of task resolution features, software guide, final and attachments. The introduction to the qualification work describes the current problem of the Latvian National Library's digital site makers and the solution needed. The Tasks Framework contains the functions that the program will need to perform. The specification of the requirements consists of input and exit information, as well as of the functional and non-functional requirements of the system. The justification for the choice of means of addressing the task shall indicate which means will be used for development and for which they are used. The software product modeling and design description consists of a system structure model that includes the system architecture and the ER model and a functional system model that contains the data flow model. A description of the data structures shows the relationship scheme of the database. As well as a table structure with a description. The User Guide contains the required program requirements for hardware and software, running the program, and a program guide and description that shows how to use the program correctly. The test example contains a detailed description of the preparation and import of objects with visual images.

Overall, qualification work consists of 48 pages, which includes images, tables and attachments.

Saturs

IEVADS	5
1. UZDEVUMA NOSTĀDNE	6
2. PRASĪBU SPECIFIKĀCIJA	7
2.1. Ieejas un izejas informācijas apraksts	7
2.1.2. Izejas informācijas apraksts	8
3. UZDEVUMU RISINĀŠANAS IZVĒLES PAMATOJUMS	11
3.1. Uzdevumu analīze	11
3.2. Izvēlēto rīku apraksts	12
3.3. Aizgūto spaudņu apraksts	13
4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA	14
4.1. Sistēmas struktūras modelis	14
4.1.1. Sistēmas arhitektūra	14
4.1.2. Sistēmas ER modelis	15
4.2. Funkcionālais sistēmas modelis	18
4.2.1. Datu plūsmu modelis	18
5. DATU STRUKTŪRU APRAKSTS	22
6. LIETOTĀJA CEĻVEDIS	27
6.1. Sistēmas prasības aparatūrai un programmatūrai	27
6.2. Programmas apraksts	28
6.3. Testa piemērs	31
NOBEIGUMS	35
INFORMĀCIJAS AVOTI	36
Definīcijas	37
PIELIKUMI	38

IEVADS

Kvalifikācijas darba mērķis ir atvieglot darbu objektu veidotājiem, kā arī padarīt to kvalitatīvāku. Kvalifikācijas darbā ir aprakstīta programma “DOM massingestapp”.

Latvijas Nacionālā bibliotēka nodarbojas ar vienu no vissvarīgākajām lietām Latvijā. Tās vēstures dokumentēšanu un uzturēšanu. Kopš Latvijas Nacionālā bibliotēka sāka strādāt, vācot vēsturiskas un svarīgas grāmatas, bija arī nepieciešamība digitalizēt šīs lietas. Ne tikai arī bibliotēka ar to nodarbojas, arī citi uzņēmumi pieprasa digitalizēt viņu vēstures dokumentus un fotogrāfijas.

Nodarboties ar to sāka jau 2013. gadā, un turpina jau projām. Tagad ne tikai digitalizē vēsturiskas lietas, bet arī mūsdienu notikumus, žurnālus, bildes un arī video.

Veidojot objektus objektu veidotājiem tiešām, ir garš un nogurdinošs process. Sākumā dati ir jāapkopo un jāsaformatē. Pēc tam ir 3 izvēles, vai nu objektu veidotājs pats vada manuāli objektu, kuri var būt tūkstošiem, vai arī lieto 2 novecojušas programmas, kas saformatē objektus metadatu pakotnēs, un tikai tad augšupielādē sistēmā, vai arī ir pēdējais variants nodot šos apkopotos un saformatētos objektus citam darbiniekam Latvijas Nacionālajā bibliotēkā, kas manuāli aizsūta visus datus par objektiem uz sistēmu. Pēc ilgas izpētes ir veikts secinājums, ka nav citu pastāvīgu risinājumu. Lai gan objekta veidotāji ir šādi strādājuši neilgu laiku, tomēr ir bijis pieprasījums izdomāt risinājumu, lai atvieglotu visiem darbu un nenāktos lietot novecojušu programmatūru.

Šis kvalifikācijas darbs ir izdomātais risinājums, lai digitālo objektu veidotāju darbs nebūtu tik sarežģīts, cik tam vajadzētu būt. Programma ir kompakta, ar to ir vienkārši rīkoties, un to ir viegli uzturēt, kā arī veikt atjauninājumus.

1. UZDEVUMA NOSTĀDNE

Kvalifikācijas darba uzdevums ir izveidot programmu, kurā var importēt digitālos objektus. Lietotājs varēs ievietot “Excel” failu, kas vizuāli parādās programmā, un tajā varēs rediģēt, pievienot un nodzēst objektus pirms to rediģēšanas. Pirms objektu importēšanas lietotājs arī var veikt pārbaudi, vai viņa dati atbilst precizitātei, ja ir kāda relācija starp laukiem sistēmā. Pirms lietotājs tiek iekšā programmā lietot to, viņam vajadzēs ievadīt savu uzņēmuma profila lietotājvārdu un paroli, lai tiktu iekšā programmā. Lietotājam jābūt izvēlei, kur importēt objektus, vai nu importēt uz testēšanas vidi, vai arī uz produkcijas vidi. Lietotājam arī jābūt iespējai redzēt, cik objekti ir importēti, un cik nav. Gadījumos, ka nevar importēt objektu jābūt arī vietai, kur lietotājs var saprast, kāpēc viņa objekts neimportējās.

Objektu importa programmai kopumā būs tikai viena lietotāju loma-objektu veidotājs. piekļuve programmai paredzēta tikai digitālā objekta veidotājiem, kuriem ir arī piekļuve objektu sistēmā.

Kvalifikācijas darbam ir vairākas funkcionalitātes:

- Importēt lielu skaitu objektu īsā laika brīdī;
- Pārbaudīt objektu informācijas atbilstību pirms importēšanas;
- Precīzi jāizvada informācija par kļūdainiem objektiem;
- Atbalstīt funkciju, ka objektiem klāt var arī pievienot failus;
- Atļaut veikt objektu informācijas rediģēšanu, ja tie ir kļūdaini;

2. PRASĪBU SPECIFIKĀCIJA

2.1. Ieejas un izejas informācijas apraksts

2.1.1 Ieejas informācijas apraksts

- Informācija par lietotāju, kas sastāv no:
 - lietotājvārda, kas ir simbolu virkne;
 - paroles, kas ir simbolu virkne;
- Informācija par objektiem, ko lietotājs importē, kā “excel” failu, kurā ir objekta mainīgo virsraksti un vērtības :
 - tips, kas ir simbolu virkne, kas norāda objekta tipu;
 - apakštips, kas ir simbolu virkne, kas norāda objekta apakštipu;
 - tituls, kas ir simbolu virkne, kas norāda objekta titulu;
 - titula valoda, kas ir 3 burtu virkne, kas norāda titula valodu;
 - radīšanas datums, kas ir datums, kas norāda oriģināla veidošanas datumu;
 - oriģināldarba formāts, kas ir burtu virkne, kas norāda oriģināldarba formātu dzīvē;
 - oriģināldarba daudzums, kas ir simbolu virkne, kas norāda, cik eksemplāri noteiktajam objektam ir dzīvē;
 - anotācija, kas ir simbolu virkne, kas pasniedz garāku aprakstu par objektu;
 - anotācijas valoda, kas ir 3 burtu virkne, kas norāda anotācijas valodu;
 - avots, kas ir simbolu virkne, kas norāda oriģināldarba avotu;
 - organizācijas lokācija, kas ir simbolu virkne, kas norāda organizāciju, kurai pieder oriģināldarbs;
 - lokācijas kods, kas ir simbolu virkne, kas apraksta Organizācijas kodu;
 - izcelsme, kas ir simbolu virkne, kas apraksta oriģināldarba izcelsmi;
 - līdz radītājs, kas ir simbolu virkne, kas apraksta oriģināldarba līdz radītāju;
 - līdz radītāja loma, kas ir simbolu virkne, kas apraksta līdz radītāja lomu oriģināldarba veidošanā;
 - līdz radītāja tezaurs, kas ir simbolu virkne, kas apraksta līdz radītāja saistošo tezaura vērtību no datubāzes;
 - kolekcija, kas ir simbolu virkne, kas parasti apraksta saistīto notikumu ar oriģināldarba izveidi;
 - kolekcijas daļa, kas ir simbolu virkne, kas detalizētāk precizē saistīto notikumu;

- “URL”, kas ir tīmekļa saite, kas norāda saistīto tīmekļa lapu ar objektu;
 - atslēgvārds, kas ir simbolu virkne, kas apraksta atslēgvārdu, ko pielieto objektu meklēšanā;
 - Autortiesības, kas ir klasifikators, kas apraksta objekta autortiesības citiem lietotājiem;
 - Piekļuve, kas ir klasifikators, kas apraksta objekta informācijas un resursu piekļuvi citiem lietotājiem;
 - Ārējais ID, kas ir unikāla vērtība objektam, kas apraksta objekta ārējo ID, lai ārējās organizācijas varētu identificēt savu objektu;
 - Papildu faili, kas ir simbolu virkne, kas apraksta objekta augšupielādējamo failu nosaukumu kopā ar faila formātu;
 - Arhīva faili, kas ir simbolu virkne, kas apraksta objekta augšupielādējamo arhīva datnes nosaukumu kopā ar faila formātu.
- Informācija par mapes atrašanos direktoriju, kurā atrodas faili, kuri jāaugšupielādē klāt pie objektiem.

2.1.2. Izejas informācijas apraksts

Izejas dati ir no programmas atgrieztie dati, no lietotāja pieprasījuma.

- Informācija par lietotāju piekļuvi sistēmā, kas sastāv no atbildes:;
 - Pareizs, kas atļauj piekļuvi programmā;
 - Nepareizs, kas liedz piekļuvi programmā un atbilstoši izvada logu ar paziņojumu, ka lietotājam piekļuve ir liegta.
- Informācija par objektiem cik ir validēti un cik vajag labot, tas sastāv no:
 - kopējo objektu skaita;
 - validēto objektu skaita;
 - labojamo objektu skaita.
- Detalizēta informācija par objektiem, kurus nevar importēt un to dati ir jālabo, un tie sastāv no:
 - Objekta ārējo ID;
 - Objekta detalizēts saraksts ar kļūdām, kurš lauks, un kāpēc tas nav pareizs.

2.2. Funkcionālās prasības

- Lietotāja piekļūšana programmā:
 - pārbaude, vai lietotāja dati atbilst lietotājam sistēmā;
 - pēc nepareizas datu ievadīšanas jābūt paziņojumam, ka tāda lietotāja nav.
- Objektu importēšana:
 - jānodrošina objekta importēšana, lietojot “Excel” failus no lietotāja;
 - jānodrošina opcija pievienot klāt objektiem nepieciešamos failus;
 - jāizveido vizuāls pārskats, lai lietotājs varētu redzēt, cik objekti ir importēti un cik nav;
 - jānodrošina objektu validēšanas opcija pirms objektu importēšanas;
 - programmā jābūt izvēlei starp 2 vidēm, uz kurām var importēt objektus;
 - Jābūt opcijai sakārtot objektus pēc jebkura mainīgā.
- Kļūdu izvade:
 - Programmai precīzi jāizvada ārā kopējais objektu skaits, importētais objektu skaits un neimportēto objektu skaits;
 - Programmai jāsakārto objekti vizuālajā skatā precīzi, no sākuma nepareizi, un pēc tam pareizie;
 - Objektu kļūdām jābūt opcijai izvadīt tos ārā, lai varētu apskatīt nākotnē;

2.3. Nefunkcionālās prasības

- Prasības procesam:
 - programmas kodam jābūt rakstītam C# programmēšanas valodā, lietojot .NET ietvara 4.6.1 versiju;
 - programmas vizuālajiem elementiem jāveido, lietojot “WPF” ietvaru;
 - testējot mijiedarbību ar sistēmu, uz kuru jāsaņem objekti, jālieto “API” pieprasījumu testētājs;
 - programma nedrīkst pārslogot datora procesoru.
- Prasības produktam:
 - lietotāju saskarnei ar sistēmu jānotiek latviešu valodā;
 - programmai nevajadzētu būt aizkavei ilgāk par 1 minūti, pildot darbietilpīgus procesus;
 - programmai jābūt dinamiski mainīgam izmēram, lai lietotājam būtu ērtāk to lietot.
- Prasības lietotāja interfeisam:
 - lapas pamatkrāsai jābūt gaiši zilai. Krāsai ir jābūt neitrālai;
 - visām programmas pogām jāietur vienāds formatēšanas stils;
 - kļūdu paziņojumiem jābūt vienotā stilā;
 - programmas pogām jābūt saprotamam apzīmējumam;
 - programmā jāiekļauj īsa lietošanas pamācība.

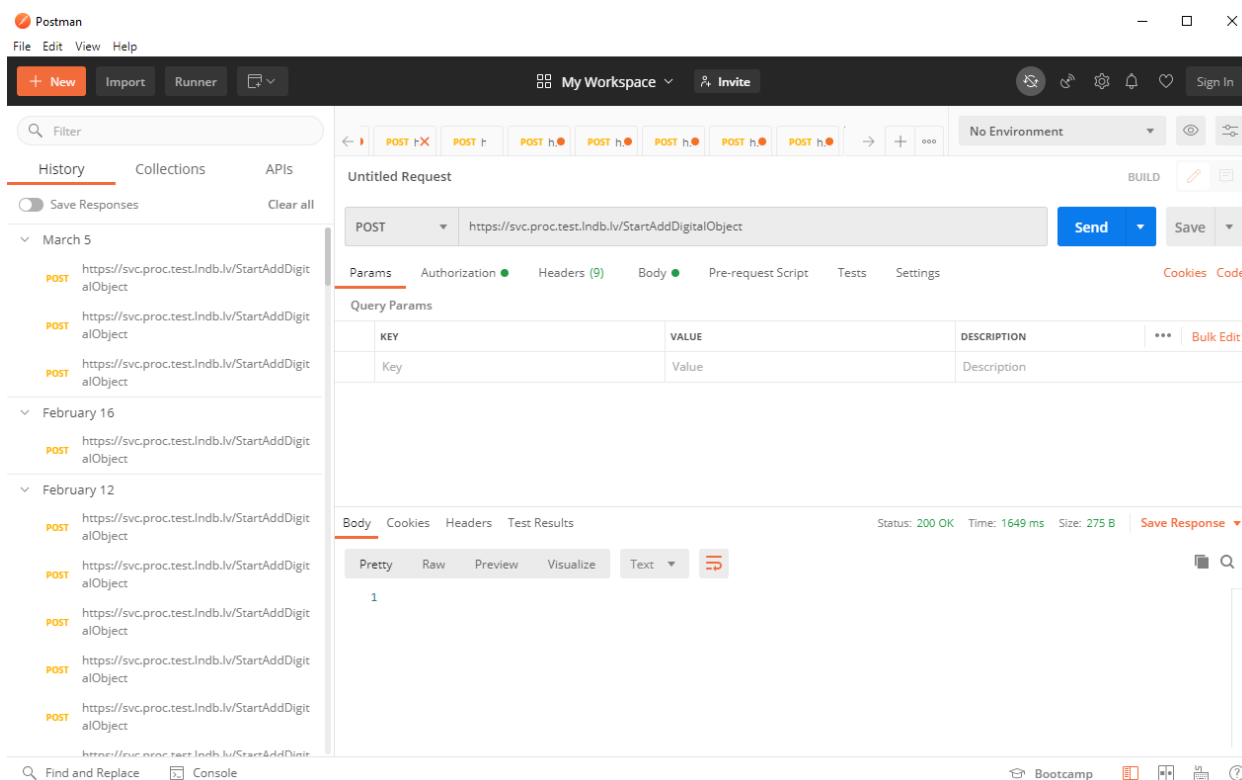
3. UZDEVUMU RISINĀŠANAS IZVĒLES PAMATOJUMS

3.1. Uzdevumu analīze

Kvalifikācijas darba ietvaros izmantoju vienu galveno programmēšanas valodu – C#. Šo programmēšanu izvēlējos, jo tā ir mūsdienīga un aktuāla, kā arī tā bija nefunkcionālā prasība no prakses vadītāja uzņēmumā. Izvējos to arī, jo šķita, ka tā ir līdzīga programmēšanas valodai C++, bet patiesība personīgi līdzību saskatīju ļoti minimālu.

Programmas vizuālajam interfeisam lietoju WPF ietvaru, jo tā bija nefunkcionālā prasība, un ar tās palīdzību var veidot pievilcīgākus interfeisa elementus, kā arī to var implementēt detalizētākas darbības priekš interfeisa elementiem.

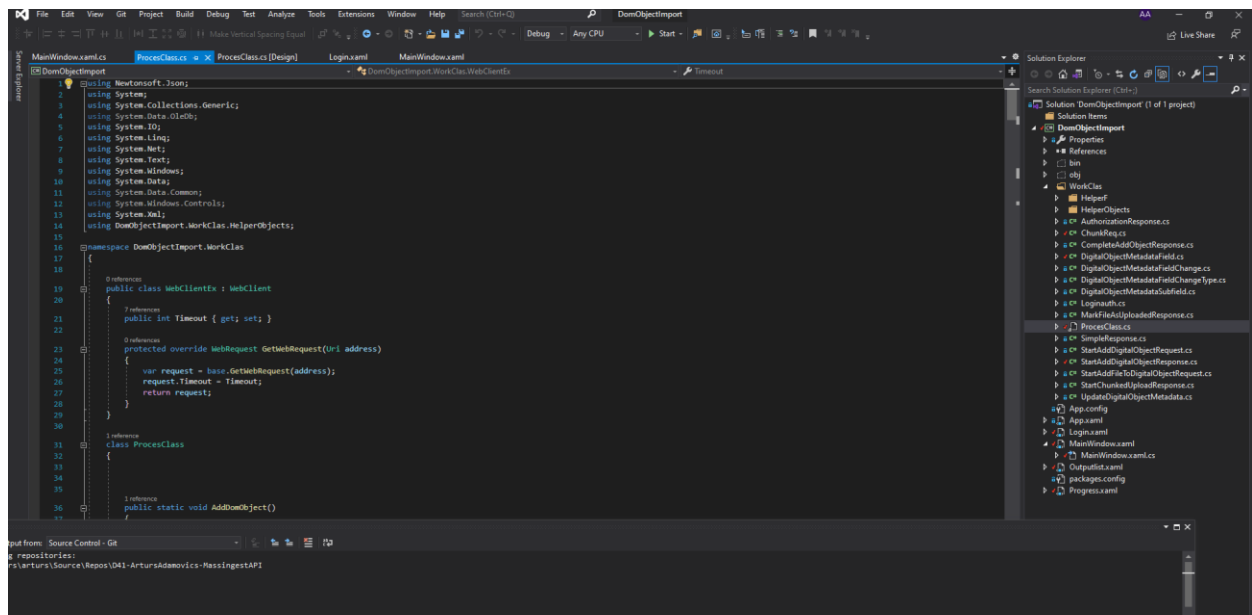
Testējot un strādājot ar objektu sistēmas pieprasījuma saitēm, lietoju lietojumprogrammas saskarnes rīku Postman. Postman ir API (lietojumprogrammu programmēšanas interfeiss) izstrādes rīks, kas palīdz veidot, testēt un modificēt API. Gandrīz jebkura funkcionalitāte, kas varētu būt nepieciešama jebkuram izstrādātājam, ir iekapsulēta šajā rīkā. To ik mēnesi izmanto vairāk nekā 5 miljoni izstrādātāju, lai padarītu savu API izstrādi vienkāršu un vieglu. Tā spēj veidot dažāda veida HTTP pieprasījumus (GET, POST, PUT, PATCH), saglabāt vides vēlākai lietošanai, konvertēt API uz kodu dažādām programmēšanas valodām. To es izvēlējos, jo programmai ir vienkāršs dizains un var viegli veikt bez piepūles, rakstot garu, kodu, lai testētu HTML pieprasījumus.



3.1 att. Postman programma

Veidojot kvalifikācijas darbu ir neizbēgama nepieciešamība lietot programmu, kurā var gan rediģēt kodu, gan arī pārbaudīt un testēt to, tāpēc rakstot programmas pirmkodu es

izmantoju Microsoft Visual Studio Community 2019. Microsoft Visual Studio Community 2019 ir pilnībā pieejams, paplašināms, bezmaksas integrēta attīstības vide, lai radītu modernas lietojumprogrammas priekš Android, iOS, Windows operētājsistēmām, kā arī, lai veidotu tīmekļa lietojumprogrammas un mākoņpakalpojumus. Manuprāt šī integrētā attīstības vidē ir



3.2 att. Visual studio 2019

viegli strādāt, kā iesācējiem, tā arī pieredzējušiem programmētājiem. Šī integrētā attīstības vide ir tik populāra, ka pamācības ir vieglāk atrast šajā integrētā attīstības vidē, nekā jebkurā citā.

3.2 Izvēlēto rīku apraksts

“Microsoft Visual Studio” ir integrēta Microsoft izstrādes vide (IDE). To izmanto datorprogrammu, kā arī tīmekļa vietņu, tīmekļa lietotņu, tīmekļa pakalpojumu un mobilo lietotņu izstrādei. “Visual Studio” izmanto Microsoft programmatūras izstrādes platformas, piemēram, “Windows API”, “Windows Forms”, “Windows Presentation Foundation”. Tas var radīt gan vietējo kodu, gan pārvaldāmo kodu.

“Visual Studio” ietver kodu redaktoru, kas atbalsta programmu “IntelliSense” (koda pabeigšanas komponents). Integrētais atklūdotājs darbojas gan kā avota līmeņa atklūdotājs, gan kā datora līmeņa atklūdotājs. Citi iebūvētie rīki ietver koda profilu, GUI lietojumprogrammu izstrādātāju, tīmekļa noformētāju, klases noformētāju un datu bāzu shēmu noformētāju. Tā pieņem spraudņus, kas paplašina funkcionalitāti gandrīz katrā līmenī, tostarp pievienojot atbalstu avota kontroles sistēmām un pievienojot jaunus rīkus, piemēram, redaktorus un vizuālos dizainerus domēniem specifiskām valodām vai rīku komplektus citiem programmatūras izstrādes dzīves cikla aspektiem

“Postman” ir rīks, ko mēs varam izmantot, lai piekļūtu jebkuram API, ko mēs vēlamies. Lielāko daļu laika mēs izmantosim “Postman”, lai pārbaudītu API, ko mēs veidojam, un produktu, pie kura mēs strādājam. Taču “Postman” var izmantot arī, lai testētu vai “izsauktu” jebkuru API, kas ir pieejama un publicēta jebkurā vietā internetā, piemēram, publiskos API, ar kuriem esat integrējies. Šo programmu es lietoju, lai testētu pieprasījumus LNB sistēmā.

3.3 Aizgūto spraudņu apraksts

“Gitlab for vs” – “Gitlab” ir tīmeklī bāzēts “Git” repozitorijs, kas nodrošina bezmaksas atvērtās un privātās krātuves, problēmu sekošanas iespējas un Vikipēdijas. Tā ir pilnīga “DevOps” platforma, kas ļauj profesionāļiem veikt visus projekta uzdevumus — no projektu plānošanas un avotu kodu pārvaldības līdz pārraudzībai un drošībai. Turklāt tas ļauj darba grupām sadarboties un veidot labāku programmatūru.

“Gitlab” komandas palīdz samazināt produktu dzīves ciklu skaitu un palielināt produktivitāti, kas savukārt rada vērtību klientiem. Lietojumprogramma nepieprasa lietotājiem pārvaldīt katra rīka atļaujas. Ja atļaujas ir iestatītas vienreiz, tad visiem organizācijas dalībniekiem ir piekļuve katram komponentam.

Šo spraudni es lietoju kvalifikācijas darba laikā, lai atjauninātu savu kodu ar LNB veidoto repozitoriju. Sākumā diezgan grūti aptvert konceptu, kā iestatīt un sataisīt, lai ar 2 klikšķiem varētu jau augšupielādēt un atjaunināt kodu repozitorijā.

4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA

4.1. Sistēmas struktūras modelis

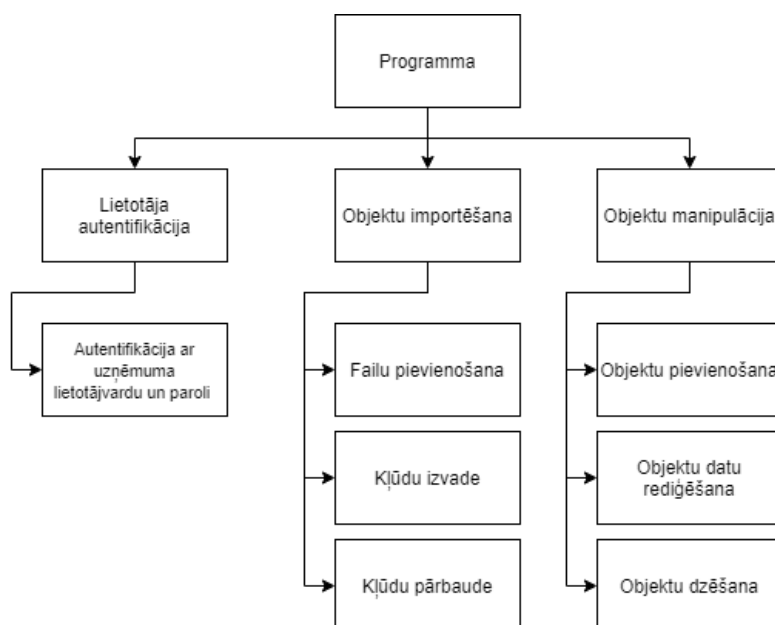
4.1.1. Sistēmas arhitektūra

Objektu importēšanas programma sastāv no 1 daļas.

1. Lietotāja daļa.

Lietotāja daļa sastāv no visām lietotāju plānotajām funkcijām. Lietotāja funkcijas iedalās autentifikācija sistēmā un objektu importēšanas. Lai autentificētos programmā, lietotājs ievada savu lietotājevārdu un paroli, kas ir piešķirta lietotājam uzņēmumā. Objektu importēšanas daļā lietotājs augšupielādē “microsoft excel” failu programmā, un pēc tam atbilstoši var rediģēt objektus, nodzēst nost un pievienot klāt. Nākamajā solī lietotājs atzīmē mapi, kurā ir pievienoti faili, kurus vajadzēs augšupielādēt pie objektiem (ja tādi ir), ja nav papildu faili, tad vienkārši var nepievienot mapi. Tad lietotājs nospiež pogu, lai veiktu datu pārbaudi. Objektu rindas ar kļūdām automātiski krāsojās sarkanā krāsā, līdz tos izlabo un atkal veic datu pārbaudi. Nobeigumā lietotājs izvēlās objektu importa pogu, un gaida izvades logu, kas izvadīs āra visu nepieciešamo informāciju par augšupielādētajiem objektiem.

Sistēmas arhitektūras diagrammu var apskatīt 4.1. attēlā.



4.1. att. Sistēmas arhitektūra

4.1.2. Sistēmas ER modelis

Datu bāzes projektēšanā datu kopu un saišu starp tām attēlošanai tika lietota realitāšu-saišu diagramma, kas sastāv no divu veidu objektiem – entītijām (loģiskais objekts) un relācijām (attiecības).

Datu bāzes ER modelis, kas ir dots pielikumā, sastāv no sekojošām entītijām. Skatīt 1.pielikuma 3.attēlu.

- **„Objekti”** – apraksta objektus sistēmā. Tās atribūtu kopums sevī ietver objekta ID, versijas ID, radīšanas datums, Lokācijas kods, anotācija, nosaukums, Datu avots, Atslēgvārds, ārējais ID, un dokumentu skaits.
- **„Lietotāji”** – apraksta lietotājus sistēmā. Tās atribūtu kopums sevī ietver e-pastu, lietotājevārdu un paroli.
- **„Veicinātājs”** – apraksta veicinātājus pievienotajam objektam. Tās atribūtu kopums sevī ietver ID un vērtību.
- **„Veicinātāja loma”** – apraksta objekta veicinātāja lomu. Tās atribūtu kopums sevī ietver kodu, vērtību un aprakstu.
- **„Tipi”** – apraksta objekta tipu sistēmā. Tās atribūtu kopums sevī ietver kods, ID un nosaukums.
- **„Apakštīpi”** – apraksta objekta apakštīpu, kas ir zem tipiem sistēmā. Tās atribūtu kopums sevī ietver ID, nosaukumu un kodu.
- **„Kolekcijas”** – apraksta kolekciju, pie kura pieder objekts sistēmā (ja ir ietverts kādā kolekcijā). Tās atribūtu kopums sevī ietver aprakstu un kodu.
- **„Kolekciju grupas”** – apraksta kolekciju specifiskāk objektiem sistēmā. Tās atribūtu kopums sevī ietver kodu un aprakstu.
- **„Piekļuve”** – apraksta piekļuvi objektam sistēmā. Tās atribūtu kopums sevī ietver kodu un aprakstu.
- **„Autortiesības”** – apraksta autortiesības objektiem sistēmā. Tās atribūtu kopums sevī ietver kodu un aprakstu.
- **„Ārējās organizācijas”** – apraksta organizācijas, kas sniedz informāciju par objektiem sistēmā. Tās atribūtu kopums sevī ietver nosaukumu un kodu.
- **„Valodas”** – apraksta objekta laukiem valodu sistēmā. Tās atribūtu kopums sevī ietver valodas kodu un valodas pilno nosaukumu.

- **„Fiziskais datu nesējs”** – apraksta, no kurienes radās informācija par objektu sistēmā. Tās atribūtu kopums sevī ietver kodu un nosaukumu.
- **„Avots”** – apraksta objektu izcelsmi sistēmā. Tās atribūtu kopums sevī ietver kodu un aprakstu.

Datu bāzes relācijas parāda, kā savstarpēji ir savienotas divas vai vairākas entītijas:

- starp lietotājiem un objektiem attiecības ir viens pret daudziem, jo viens lietotājs var veidot vairākus objektus, bet vienu objektu ir veidojis viens lietotājs;
- starp veicinātāju lomām un objektiem attiecības ir viens pret daudziem, jo objektu veicinātāju lomas var atkārtoties atšķirīgiem veicinātājiem, bet vienam veicinātājam būs tikai viena loma;
- starp tipiem un objektiem attiecības ir viens pret daudziem, jo objektiem tipi var atkārtoties citos objektos, bet viens tips var būt vienam objektam;
- starp tipiem un apakštipiem attiecības ir viens pret daudziem, jo vienam tipam var būt vairāki apakštipi, bet apakštipi var attiekties tikai uz vienu tipu;
- starp valodām un objektiem attiecības ir viens pret daudziem, jo vairākiem objektiem var būt viena valoda, bet valodas nevar būt vienādas;
- starp kolekcijām un objektiem attiecības ir viens pret daudziem, jo vairākiem objektiem var būt viena kolekcija, bet katrs objekts satur tikai 1 kolekciju;
- starp kolekciju grupām un kolekcijām attiecības ir viens pret daudziem, jo vienai kolekcijai, var būt vairākas kolekciju grupas, kas precizē kolekciju, bet vienai kolekciju grupai var piederēt tikai vienai kolekcijai.
- starp veicinātājiem un objektiem attiecības ir viens pret daudziem, jo objektam var būt vairāki veicinātāji, bet katram veicinātājam ir tikai viens kods;
- starp piekļuvi un objektiem attiecības ir viens pret daudziem, jo vairākiem objektiem var būt vienāda piekļuve, bet viena piekļuve atbilst tikai vienam piekļuves veidam;
- starp autortiesības un objektiem attiecības ir viens pret daudziem, jo vairākiem objektiem var būt vienādas autortiesības, bet autortiesību veidi neatkārtojas;
- starp avotiem un objektiem attiecības ir viens pret daudziem, jo vairākiem objektiem avots var būt vienāds, bet vienam objektam ir viens avots;
- starp ārējām organizācijām un objektiem attiecības ir viens pret daudziem, jo vairākiem objektiem var būt vienādas ārējās organizācijas, bet organizācija pastāv tikai viena;

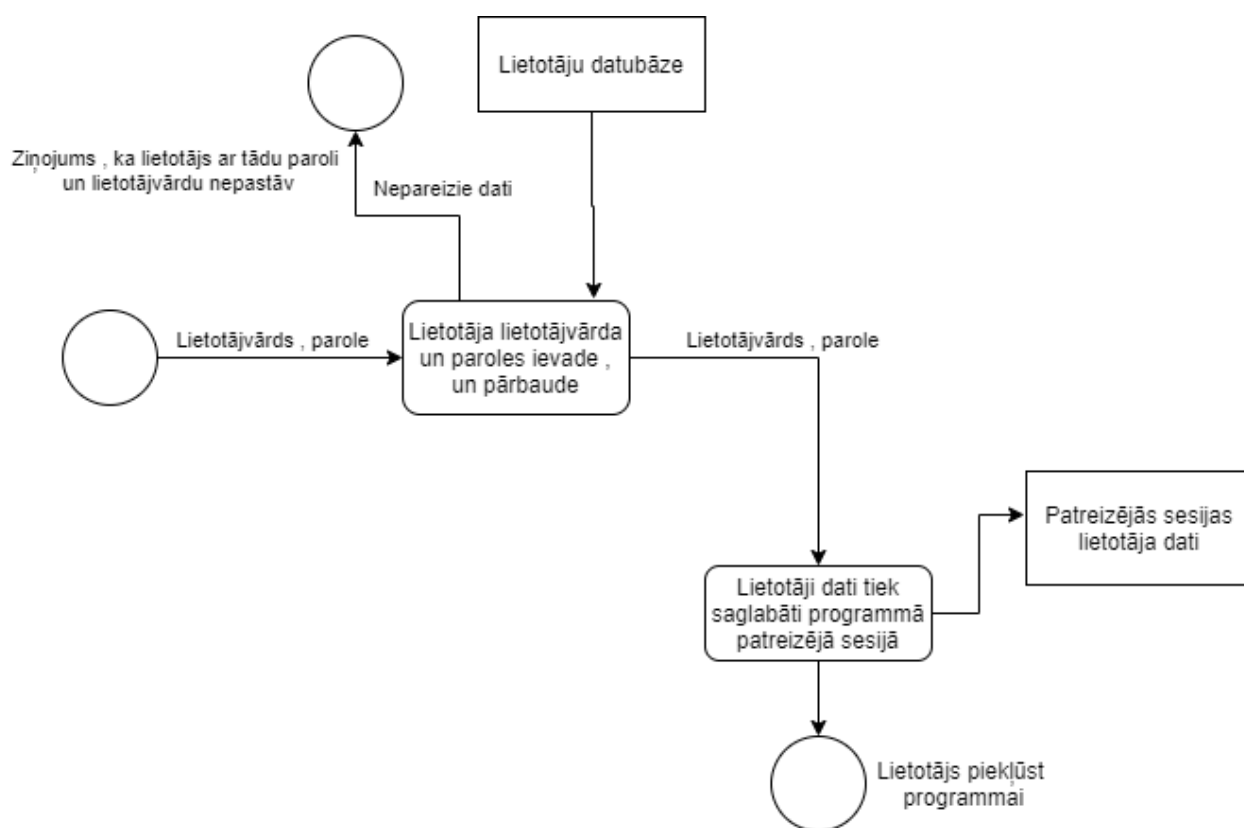
- starp fizisko datu nesēju, un objektiem attiecības ir viens pret daudziem, jo vairākiem objektiem var būt vienāds fiziskais datu nesējs, bet datu nesēji ir vairāki, bet tie neatkārtojas.

4.2. Funkcionālais sistēmas modelis

4.2.1. Datu plūsmu modelis

1. Lietotāja autentifikācija

Palaižot programmu, lietotāja autentifikācija notiek sekojoši skatīt 4.3 attēlu. Lietotājs autentifikācijas logā ievada savu lietotājevārdu un paroli. Tad šos datus programma nosūta uz serveri un salīdzina starp esošajiem lietotājiem datubāzē. Atbilstoši, ja dati ir pareizi, tad lietotājevārdu un paroli programma saglabā esošajā sesijā, un lietotājs piekļūst programmai. Ja dati ir nepareizi, tad atbilstoši programma izvada paziņojumu, ka dati neatbilst lietotājam no datubāzes un lietotājs tālāk nekur netiek.



4.3 att. Autentifikācijas datu plūsma

2. Objektu importēšana

Objektu importēšanas funkcija, iekļūstot programmā, notiek šādi. Skatīt 1. pielikuma 2. attēlu.

Lai lietotājs vispār sākuma varētu veikt jebkāda veida objektu importēšanu vajag izvēlēties iepriekš sagatavotu “Excel” failu ar apkopotiem objektiem. Pēc faila izvēlēšanās arī jāizvēlas specifiskā darba lapa, kurā atrodas objektu kopums.

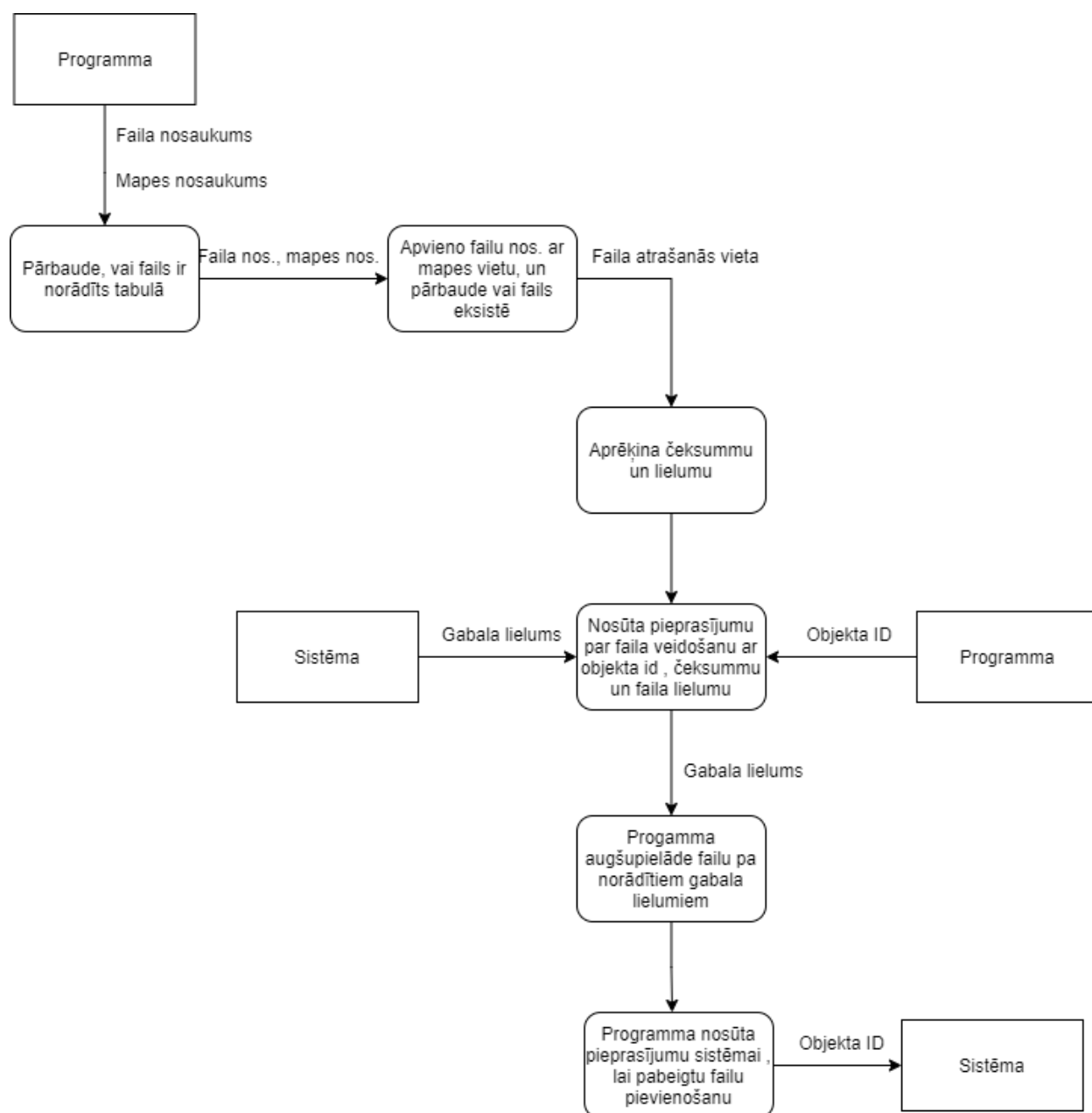
Tad šo tabulu programma saglabā un vizuāli attēlo programmā. Tagad jau var veikt objektu importēšanu, ja objektiem klāt nav nepieciešami faili, un nav nepieciešama pirms pārbaude. Tikai jānorāda, uz kuru vidi vēlas importēt: produkcijas vai testa vidi. Ja tomēr darba lapa nav gatava, tad lietotājs var veikt datu rediģēšanu, pievienot klāt mapes atrašanos vietu, kur atrodas nepieciešamie faili kurus augšupielādē klāt objektam.

Nākamajā datu plūsmā programma cikliski sūta uz sistēmu datus par objektiem un to nepieciešamos failus. Klāt pie failiem programma arī sūta faila izmēru un md5 kontrolsummu, pēc tam sadalīti sūta to pa daļām.

Objekta importa procesa beigās programmai atgriež atpakaļ, vai objekts ir importēts, un vai tajā ir kļūdas. Programma šos datus aprēķina kopā un izvada, cik ir importēti un cik nav. Datus par katra objekta kļūdām programma saglabā sarakstā, un izvada tos ārā, uzspiežot izvades pogu.

3. Objektu formatēšana

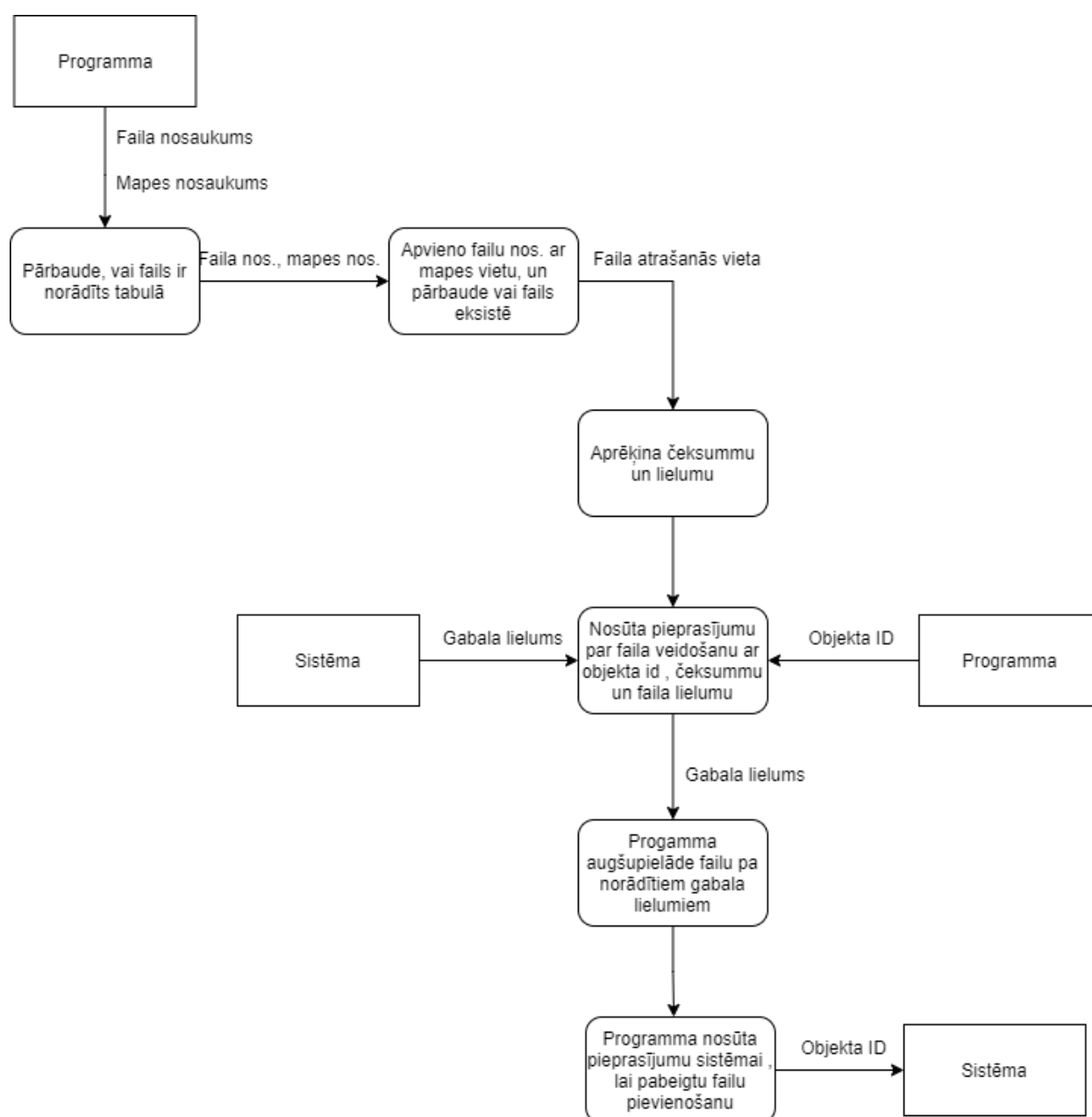
Objektu formatēšanas funkcija pēc objektu importa funkcijas notiek šādi. Skatīt 4.5 attēlu. Neatkarīgi no tā vai lietotājs tikai pārbauda vai veic objektu importu pēc tā seko objektu formatēšana, kas padara objektus ar kļūdām redzamākus un objektus bez kļūdām sakārto zem visiem nepareizajiem objektiem. Objektu importa laikā programma visus objektus ar kļūdām ievieto sarakstā. Sarakstā atrodas visu nepareizo objektu ārējie ID, kuri ir unikāli un tos lieto, lai objektus identificētu ārējās organizācijas. Pēc objektu importa programma salīdzina objektus ar sarakstu, un objektiem, kas sakrīt ar sarakstu tiek atbilstoši krāsota rinda sarkanā krāsā. Pēc rindu krāsošanas objektus atbilstoši sakārto ar sarkano krāsu augšā un ar balto krāsu apakšā.



4.5 att. formatēšanas datu plūsma

4. Failu augšupielāde objektam

Failu augšupielādes funkcija notiek objektu importēšanas laikā starp veidošanas pieprasījuma un beigšanas pieprasījumu. Datu plūsmu skatīt 4.6 attēlu. Sākumā programma pārbauda, vai objektam ir aizpildīta rinda ar faila nosaukumu un paplašinājumu. Ja objektam ir faila nosaukums ar paplašinājumu, tad programma attiecīgi pārbauda, vai fails eksistē. Programma konstatē, ka fails eksistē un aprēķina MD5 kontrolsummu, un atzīmē faila lielumu. Pēc tās programmas nosūta pieprasījumu pievienot failu ar objekta ID, kontrolsummu un faila lielumu. Veiksmīga pieprasījuma gadījumā programmai sistēma atgriež faila augšupielādējamo gabala lielumu, un tiek cikliski augšupielādēts fails. Lai pabeigtu faila pievienošanu, ir jāizsūta pieprasījums ar objekta ID, un fails ir veiksmīgi augšupielādēts.



4.6 att. Augšupielādes datu plūsma

5. DATU STRUKTŪRU APRAKSTS

Datubāzi veido četrpadsmit tabulas. Datubāze sevī ietver informāciju par objektiem, kurš lietotājs tos veidojis, objekta klasifikatorus, anotācijas un nosaukumu, kā arī atbilstoši lauki, kas atzīme valodu citiem laukiem. Datubāze arī sastāv no atkārtojamiem laukiem un unikāliem laukiem. Precīzi dati par katru mainīgā lauku nav zināmi, jo es tikai apzīmēju datus ar kuriem izstrādāju kvalifikācijas darbu. Zemāk ir attēlota datubāzes datu shēma. Zināms arī ir tas, ka datubāzes klasifikatori ir dinamiski, līdz ar to klasifikatorus var mainīt, un pievienot klāt, bet nevar izdzēst, ja klasifikatori ir saistīti ar objektu. Skatīt. 1.pielikuma 4.attēlu.

Tabula „Users” sevī ietver visus nepieciešamos datus par lietotāju, Objektu izveidē. Tā sastāv no lietotājvārda, parole un e-pasta. Lietotāja datus lieto objektu izveidē: pie katra objekta atzīmē, kurš lietotājs veidojis, kā arī datus lieto, lai autentificētos gan sistēmā, gan programmā.

5.1. tabula

Tabulas „Users” struktūra

Nr.	Nosaukums	Piezīme
1.	Username	Norāda lietotājvārdu
2.	Password	Norāda paroli
3.	E-mail	Norāda e-pastu

Tabula “Digital Objects” sevī ietver visus datus izveidotajiem objektiem. Tā sastāv no vairākiem unikāliem un parastiem laukiem, kas var atkārtoties. Digitālo objektu unikālā atslēga ir ID.

5.2. tabula

Tabulas „Digital Objects” struktūra

Nr.	Nosaukums	Piezīme
1.	ID	Norāda lietotājvārdu
2.	DOV ID	Norāda paroli
3.	Name	Apraksta objekta nosaukumu
4.	Name_lang	Apraksta objekta nosaukuma valodu
5.	DateCreated	Apraksta objekta veidošanas datumu
6.	Abstract	Apraksta objekta anotāciju
7.	Abstract_lang	Apraksta anotācijas valodu
8.	Source	Apraksta objekta izcelsmi
9.	Type	Apraksta objekta tipu
10.	Subtype	Apraksta objekta apakštipu
11.	LocationOrganization	Apraksta ārējo organizāciju
12.	locationcode	Apraksta ārējās organizācijas kodu
13.	Origin	Apraksta objekta avotu
14.	Contributor	Apraksta objekta veicinātāju
15.	Contributor_role	Apraksta veicinātāja lomu
16.	isPartOfCollection	Apraksta, kurai kolekcijai pieder
17.	isPartOf	Precizē kolekciju
18.	copyright	Apraksta objekta autortiesības
19.	accessRight	Apraksta objekta piekļuves tiesības
20.	UserCreated	Apraksta lietotāju, kas izveidoja objektu
21.	Keyword	Apraksta objekta atslēgvārdu

22.	Formatextent	Apraksta objekta oriģināla formu
23.	Formatmedium	Apraksta oriģināla eksemplārus

Tabula "Formatmedium" sevī ietver visus datus par objekta īstās dzīves oriģināla formātu. Tā sastāv no laukiem "Code" un "Description".

5.3. tabula

Tabulas „Formatmedium” struktūra

Nr.	Nosaukums	Piezīme
1.	Code	Norāda oriģinālā darba formātu
2.	Description	Norāda formāta aprakstu

Tabula "Access" sevī ietver datus, kurus piešķir objektam par to, kurš var piekļūt objektam, un kurš nevar. Tā sastāv no 2 laukiem. Tabulas "Code" elements nav atkārtojams.

5.4. tabula

Tabulas „Access” struktūra

Nr.	Nosaukums	Piezīme
1.	Code	Norāda objektu piekļuves kodus
2.	Description	Norāda piekļuves apzīmējuma aprakstu

Tabula "Collection" sevī ietver datus par kolekcijām, kurus var pievienot objektam. Tā sastāv no vairākiem unikāliem un parastiem laukiem, kas var atkārtoties. Kolekciju unikālā atslēga ir ID.

5.5. tabula

Tabulas „Collection” struktūra

Nr.	Nosaukums	Piezīme
1.	ID	Norāda kolekcijas ID
2.	Version	Norāda kolekcijas versiju
3.	DOV ID	Norāda kolekcijas versijas ID
4.	Name	Apraksta kolekcijas nosaukumu
5.	CollectionGroup	Norāda kolekciju grupu

Tabula "CollectionPartof" apraksta kolekciju grupas, kas palīdz precizēt kolekcijas notikumu. Tā sastāv no vairākiem unikāliem un parastiem laukiem, kas var atkārtoties.

5.6. tabula

Tabulas „CollectionPartof” struktūra

Nr.	Nosaukums	Piezīme
1.	Collection_ID	ID, kas norāda, kurai kolekcijai pieder
2.	ID	Norāda kolekciju grupas ID
3.	Name	Apraksta kolekcijas grupu

Tabula “Lang” sevī ietver datus par to, kurā valodā ir norādīti objekta lauki. Tā sastāv no 2 laukiem. “Lang” tabulas unikālā atslēga ir “Lang”.

5.7. tabula

Tabulas „Lang” struktūra

Nr.	Nosaukums	Piezīme
1.	Lang	Norāda lietotājbvārd
2.	Name	Norāda paroli

Tabula “Source” sevī ietver datus par objekta izcelsmi un izcelsmes nosaukumu. Tā sastāv no 2 laukiem. “Source” tabulas unikālā atslēga ir ID.

5.8. tabula

Tabulas „Source” struktūra

Nr.	Nosaukums	Piezīme
1.	Name	Norāda izcelsmes nosaukumu
2.	ID	Norāda izcelsmes ID

Tabula “Type” sevī ietver visus datus izveidotajiem objektiem. Tā sastāv no 2 laukiem, un ir saistīta ar tabulu “Subtypes”.

5.9. tabula

Tabulas „Type” struktūra

Nr.	Nosaukums	Piezīme
1.	ID	Norāda datu tipu ID
2.	Name	Norāda objekta datu tipu nosaukumus

Tabula “Subtypes” sevī ietver datus par objektu apakštīpiem. Tā sastāv no 3 laukiem. Tabulas unikālā atslēga ir ID.

5.10. tabula

Tabulas „Subtypes” struktūra

Nr.	Nosaukums	Piezīme
1.	ID	Norāda apakštīpa ID
2.	Type	Norāda tipu, kam pieder apakštīps
3.	Name	Apraksta apakštīpa nosaukumu

Tabula “locationOrganization” sevī ietver datus par piederošo ārējo organizāciju, kam pieder objekts. Tā sastāv no 2 laukiem. Unikālā atslēga ir “Code”.

5.11. tabula

Tabulas „locationOrganization” struktūra

Nr.	Nosaukums	Piezīme
1.	Code	Norāda ārējās organizācijas kodu
2.	Description	Apraksta ārējo organizāciju

Tabula “Copyright” sevī ietver datus par objektu autortiesības līmeņiem. Tā satur 2 laukus. Unikālā atslēga ir “Code”.

5.12. tabula

Tabulas „copyright” struktūra

Nr.	Nosaukums	Piezīme
1.	Code	Norāda lietotājevārdu
2.	description	Norāda paroli

Tabula “Contributor” sevī ietver datus par objekta veicinātājiem. Tā sastāv no 2 laukiem, un unikālā atslēga ir ID.

5.13. tabula

Tabulas „Contributor” struktūra

Nr.	Nosaukums	Piezīme
1.	ID	Norāda veicinātāja ID
2.	Value	Norāda veicinātāja vērtību

Tabula “contributor_role” sevī ietver datus par veicinātāja lomu objekta izveidē. Tā sastāv no 3 laukiem, un unikālā atslēga ir “Code”.

5.14. tabula

Tabulas „contributor_role” struktūra

Nr.	Nosaukums	Piezīme
1.	Code	Norāda lomas kodu
2.	Value	Norāda lomas koda apzīmējuma pilno vērtību
3.	Description	Norāda lomas aprakstu

6. LIETOTĀJA CEĻVEDIS

6.1. Sistēmas prasības aparatūrai un programmatūrai

Lai sāktu efektīvu darbu ar programmu ir nepieciešamas 2 lietas:

- Uztādīta programma;
- fails ar apkopotiem objektu metadatiem.

Uztādīta programma ir programma, kas ir nolikta viegli pieejamā vietā un ir sagatavota, lai to varētu palaist un sākt lietot. Šajā gadījumā programmai nav uzstādīts viens konkrēts instalācijas fails, jo programma jau projām ir nodota ilgākā testēšanā.

Fails ar apkopotiem objektu metadatiem ir fails, kurā atradīsies visi objekti un tās vērtības, kuras nepieciešamas importēt. Veidojot šos datus par programmu ir nepieciešams izmantot programmatūru, kurai ir iespēja failus saglabāt ar “.xlsx” paplašinājumu, šis ir

AutoSave Off Akopoti metadata Search										ahmed velo AV																
File Home Insert Page Layout Formulas Data Review View Help										Share Comments																
Arial 11										General Conditional Formatting Format as Table Normal Bad Good Neutral Insert Delete Format Sum Fill Sort & Filter Find & Select Analyze Data																
Merge & Center										Calculation Check Cell Explanatory ... Followed By																
\$ %										Clear																
U12																										
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	type	subtype	title	title language	dateCreated	formatMedium	formatExtent	abstract	abstract language	location	location	origin	contribu	contribu	contribu	isPartOf	isPartOf	keyword	Copyright	accessRi	external	Imports	file_Acc	file_Acc	file_Access	file_Arch
2	Image	Photo	"Alfa"	lav	1976	Fotopapirs		Fotogrāfi	lav										Public	AllowPub	test_1	ERAF3				
3	Image	Photo	"Alfa"	lav	1976	Fotopapirs			lav	Guļbene									Public	AllowPub	GNB_AF	ERAF3				
4	Image	Photo	"Alfa"	lav	1976	Fotopapirs			lav	Guļbene									Cilvēks	Public	AllowPub	GNB_AF	ERAF3			
5	Image	Photo	"Alfa"	lav	[197-]	Fotopapirs		Fotogrāfi	lav	Guļbene									Cilvēks	Public	AllowPub	GNB_AF	ERAF3			
6	Image	Photo	"Alfa"	lav	1976	Fotopapirs			lav	Guļbene									Cilvēks	Public	AllowPub	GNB_AF	ERAF3			
7	Image	Photo	"Alfa"	lav	[197-]	Fotopapirs			lav	Guļbene									Cilvēks	Public	AllowPub	GNB_AF	ERAF3			
8																										
9																										

6.1 att. Objektu šablons

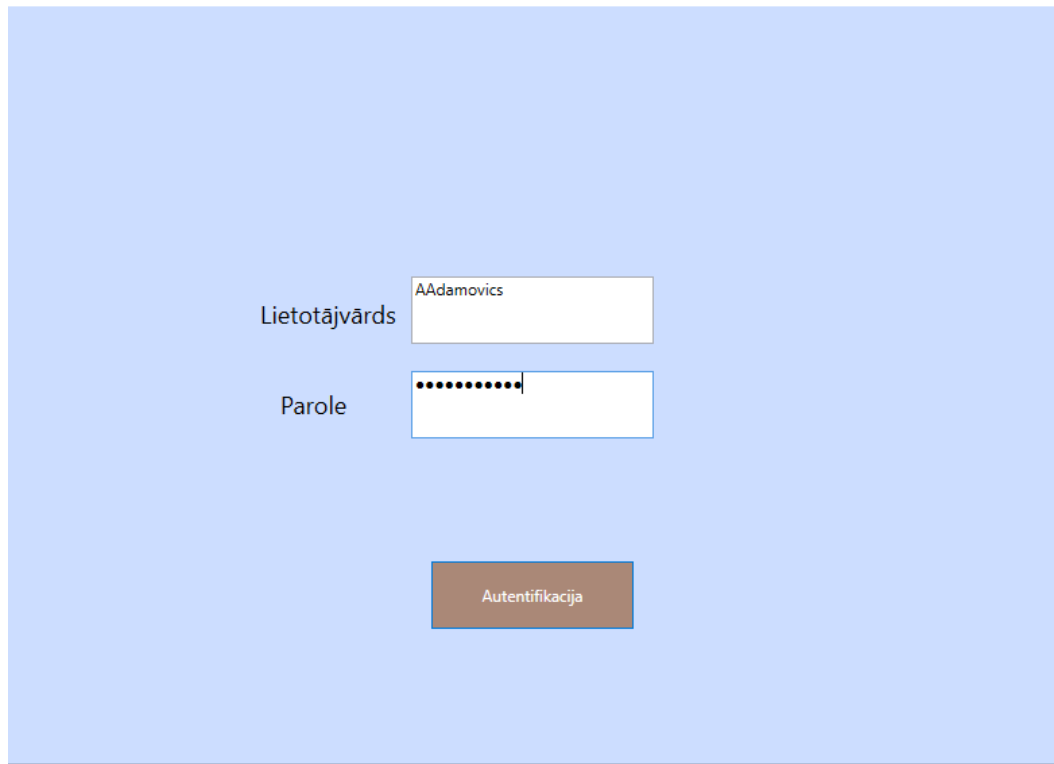
vienīgais paplašinājums, ko programma atbalsta, lai vizuāli varētu attēlot objektu tabulu. Lai programma spētu visus objektus importēt ar precīzām vērtībām ir nepieciešams objektu vērtības aizpildīt pēc šablona. Skatīt 6.1 attēlu.

Sakarā ar to, ka programmas lietotāji visi būs cilvēki, kas strādā LNB, un lieto datorus ar līdzīgām operētājsistēmām: “Windows 10” un “Windows 8”, nav nepieciešams atbalstīt citas darba sistēmas, kā arī programma nav paredzēta lietošanai viedtālrunos. Programmas uzstādīšana pēc testēšanas arī būs daudz atvieglotāka lietotājiem, jo ir jau zināmi programmas lietotāji. Atbilstoši programmas lietotājiem programma būs automātiski uzstādīta ar LNB tīkla aktīvo direktoriju, kurā atrodas visi lietotāji, un pēc tam ar grupu politikas palīdzību šiem lietotājiem var uzstādīt programmu pēc datora restartēšanas.

Aktīvā direktorija ir direktoriju pakalpojumu “LDAP” savietojama “Microsoft” realizācija “Windows NT” saimes operētājsistēmām. Aktīvā direktorija ļauj administratoriem izmantot grupu politiku, lai nodrošinātu lietotāja darba vides unifikāciju, vienkārši instalēt un uzturēt programmnodrošinājumu vienlaicīgi uz daudziem datoriem.

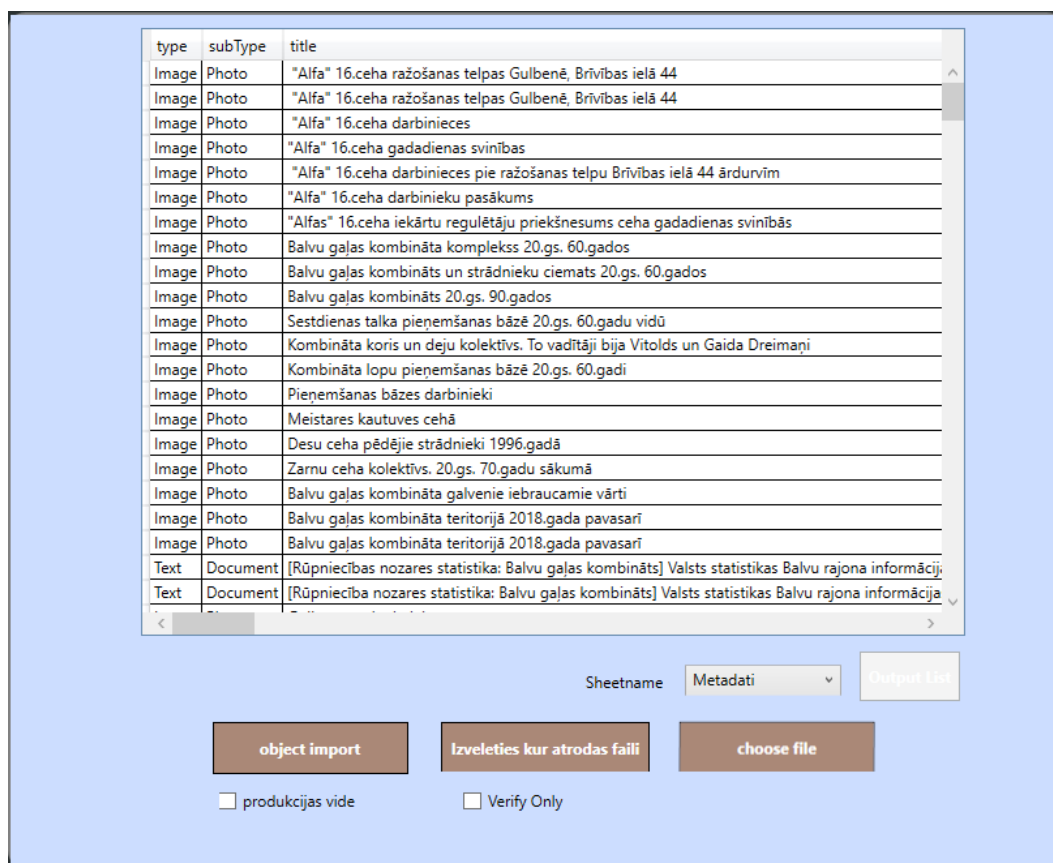
6.2. Programmas apraksts

Kad lietotājs palaiž programmu viņam, uzrādās pirmais logs, kas ir autentificēšanās logs. Skatīt 6.2 attēlu. Nepareizi ievadot lietotāja datus, programma atgriež paziņojumu, ka lietotājs nevar tikt tālāk programmā. Pareizi ievadot lietotāja autentificēšanās datus zem šī loga seko galvenais programmas logs. Skatīt 6.3 attēlu.



The image shows a login window with a light blue background. It contains two input fields: the first is labeled 'Lietotājvārds' (Username) and contains the text 'AAdamovics'; the second is labeled 'Parole' (Password) and contains ten black dots. Below these fields is a brown button with the text 'Autentifikācija' (Authentication).

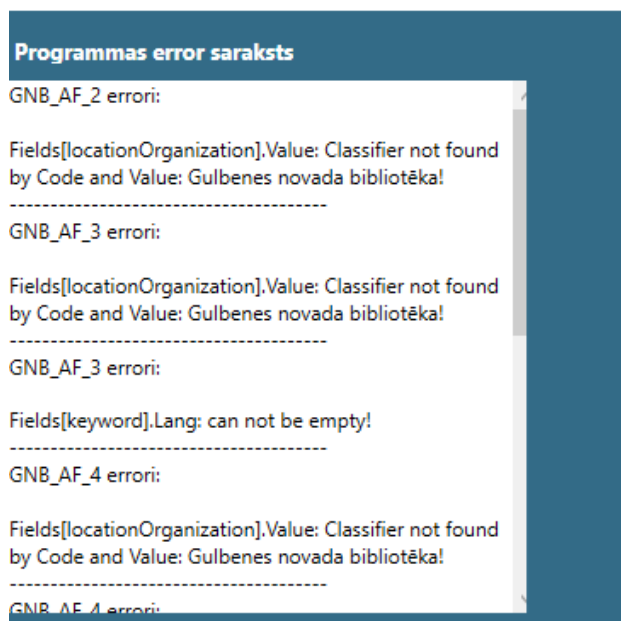
6.2 att. autentifikācijas logs



6.3 att. Galvenais logs

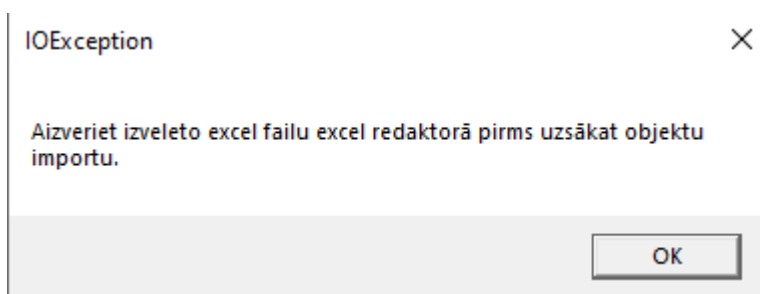
Galvenā loga centrā atrodas vizuālais programmas elements, kas attēlo "excel" faila tabulas. Galvenās lapas apakšā ir 4 galvenās programmas pogas, kurām katrai ir aprakstošs apzīmējums. Labās puses ceturtā poga sākumā nav nospiežama, jo tā aktivizējas tikai tad, kad ir veikts objektu importēšanas process. Galvenajā programmas logā arī atrodas 2 izvēles rūtiņu lauki. Vienu rūtiņu lieto, lai veiktu tikai informācijas pārbaudi, otru rūtiņu lieto, lai norādītu uz kuru vidi importēt objektu tabulu. Nolaižamās izvēlnes elements galvenajā lapā norāda objektu failu darba lapu nosaukumus, un lietotājs izvēlās to, kuru vēlas, lai programma attēlo vizuāli.

Zem galvenā logā vēl atrodas pēdējais logs, salīdzinoši mazāks par pārējiem ir kļūdu izvades logs. Šī loga centrā programma izvada ārā nepareizo objektu ID, un kādas kļūdas apzīmē objektu. Skatīt 6.4 attēlu.



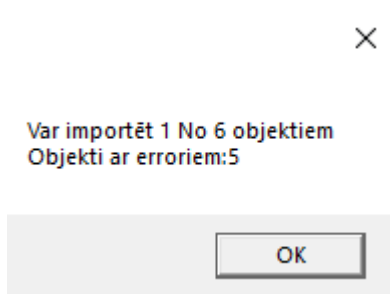
6.4 att. Kļūdu izvades logs

Kad lietotājs atver objektu failu, kas jau ir atvērts citur, parādās šis paziņojums. Skatīt 6.5 attēlu



6.5 att. Aizveriet failu paziņojums

Objektu importa procesa beigās lietotājam izvada šādu paziņojumu. Paziņojums apzīmē, cik kopā ir objekti, cik ir importēti, un cik nav importēti. Skatīt 6.6 attēlu.

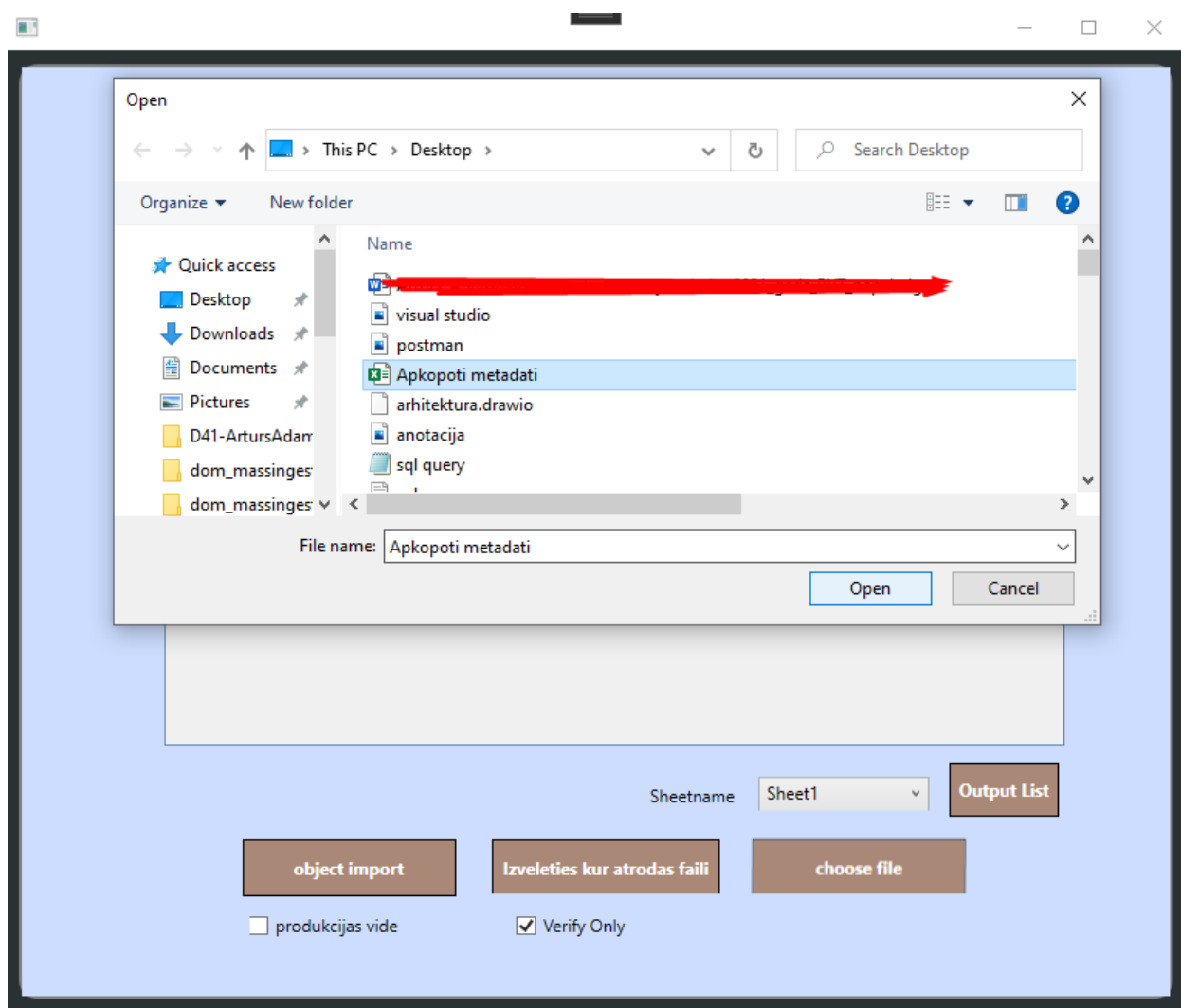


6.6 att. Importa beigu paziņojums

6.3. Testa piemērs

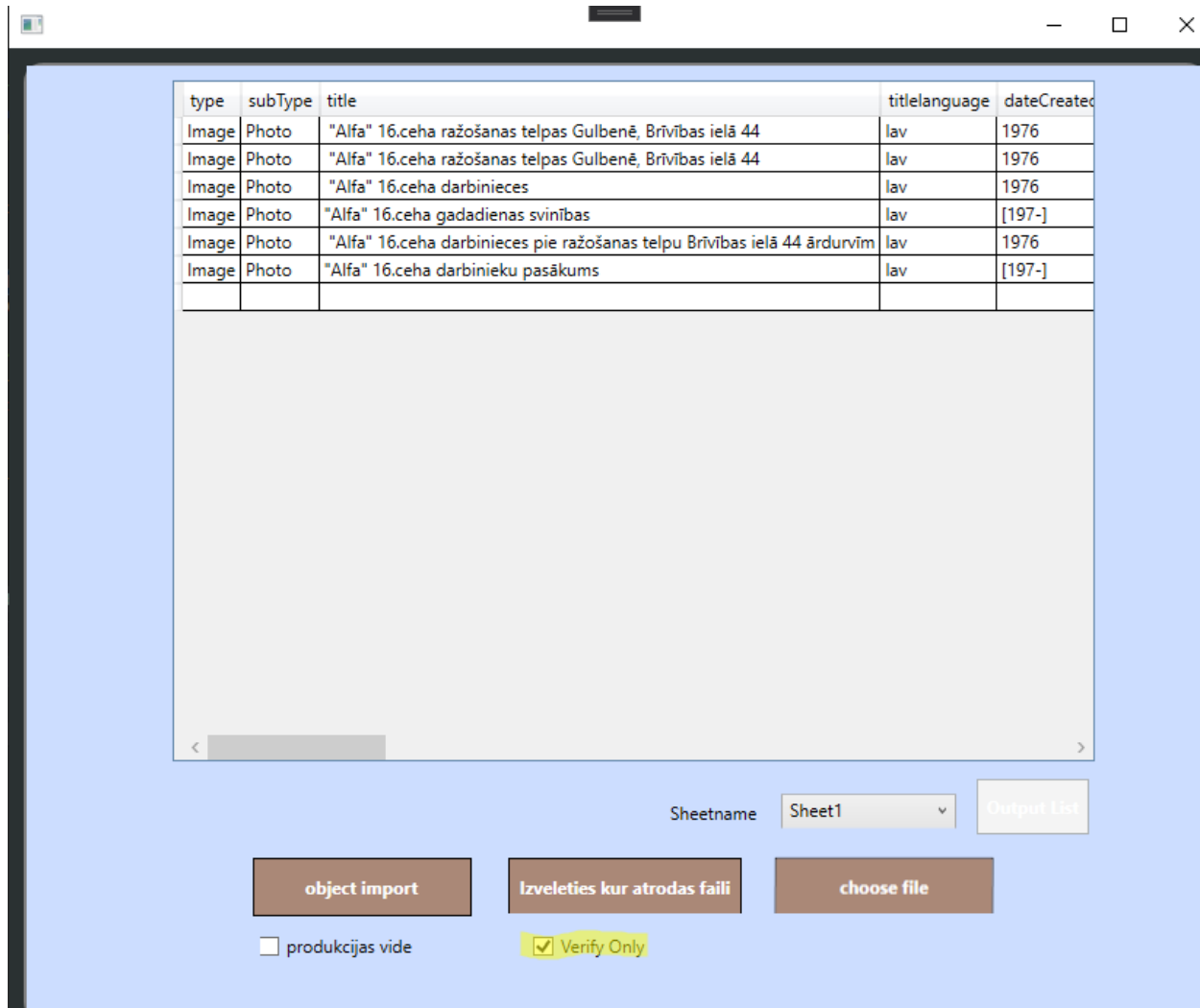
Testa piemēra nolūks ir parādīt, kā izskatās objektu importēšanas process, lai lietotāji paši to varētu atkārtot.

Pirmais solis, lai sasniegtu vēlamu rezultātu ir autentificēšanās programmā. Ievadam savus LNB lietotāja datus un spiežam “Enter” taustiņu. Kad tas ir izdarīts vajadzētu parādīties galvenajam logam. Tagad mēs spiežam pogu “choose file” un izvēlamies savu failu, kurā ir veidoti digitālie objekti pēc iepriekšminētā šablona un spiežam open failu pārlūkā. Skatīt 6.7 attēlu.



6.7 att. Faila izvēlēšanās

Nākamais solis ir pie nosaukuma “Sheetname” izvēlēties attiecīgo darba lapu no nolaižamās izvēlnes, kurā atrodas digitālie objekti. Pirms veicam jebkādu objektu importu ar kļūdām, vienmēr vieglāk ir ielikt ķeksi pie nosaukuma “Verify Only”, lai veiktu objektu vērtību pārbaudi, pirms importējam objektus. Skatīt 6.8 attēlu



6.8 att. Pārbaude pirms importēšanas

Tagad nospiežam pogu “object import”, lai uzsāktu objektu vērtību pārbaudi. Pēc tā programma izvada pretī kopējo objektu skaitu, cik objektus var importēt, un cik nevar. Nospiežam pogu “OK” un pēc tam, objekti vizuālajā logā bez kļūdām būs apakšā, un objektu rinda paliks baltā krāsā, bet objekti ar kļūdām būs augšā, un iekrāsoti sarkanā krāsā. Skatīt 6.9 attēlu.

type	subType	title	titlelanguage	dateCreated
Image	Photo	"Alfa" 16.ceha ražošanas telpas Gulbenē, Brīvības ielā 44	lav	1976
Image	Photo	"Alfa" 16.ceha darbinieces	lav	1976
Image	Photo	"Alfa" 16.ceha gadadienas svinības	lav	[197-]
Image	Photo	"Alfa" 16.ceha darbinieces pie ražošanas telpu Brīvības ielā 44 ārdurvīm	lav	1976
Image	Photo	"Alfa" 16.ceha darbinieku pasākums	lav	[197-]
Image	Photo	"Alfa" 16.ceha ražošanas telpas Gulbenē, Brīvības ielā 44	lav	1976

Sheetname

Sheet1

Output List

object import

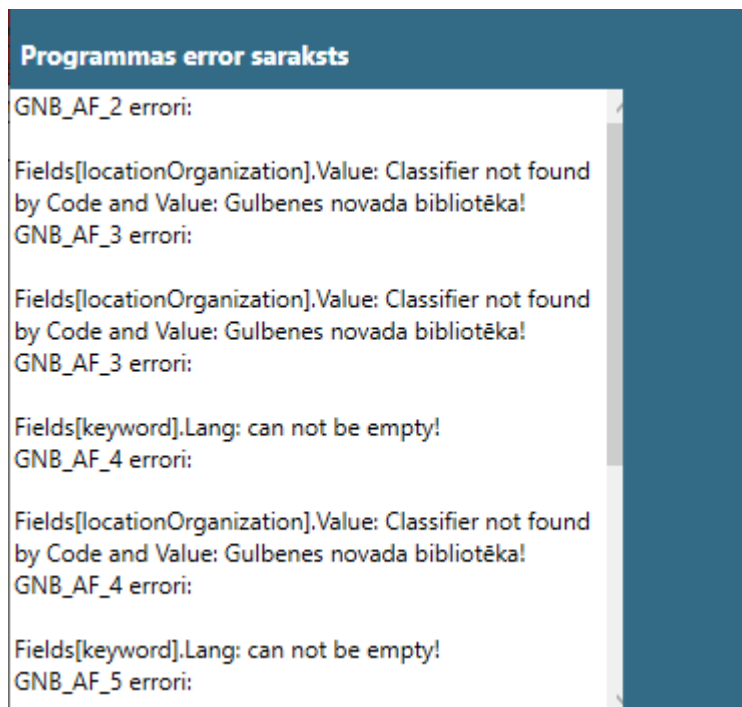
Izvēlēties kur atrodas faili

choose file

☐ produkcijas vide
 ☒ Verify Only

6.9 att. Objektu krāsošana un kārtošana

Nākamais solis ir uzspiest pogu "Output List", un apskatīties kādas kļūdas ir objektiem, kas ir krāsoti sarkanā krāsā. Skatīt 6.10 attēlu. Atbilstoši pēc tam izlabojam objekta vērtības, kuras bija kļūdainas, un nospiežam pogu "Object Import". Atkārtojam šīs darbības, līdz kļūdu vairs nav.



6.10 att. Objektu kļūdu saraksts

Kad kļūdas vairs neuzrādās, tad noņemam ķeksi “Verify only”. Ja objektiem ir pievienojamie faili, tad nospiežam pogu “Izvēlēties, kur atrodas faili”, un norādām mapi, kurā atrodas faili, ja nav pievienojamie faili, tad nespiežam pogu. Tad atliek izvēlēties vidi, uz kuru gribam importēt, ja uz testa vidi tad ķeksis nosaukumam “produkcijas vide” nav jāliek un ja gribam uz produkcijas vidi, tad atbilstoši ķeksis ir jāliek. Pēdējā lieta, kas palikusi ir uzspiest pogu “Object Import” un nu jau skatīties objektus sistēmā. Skatīt 6.11 attēlu.

	Sīktēls	ID	V.	DOV ID	Nosaukums	Tips	Statuss	Izv. liet.	Izv. dat.	Uzdevuma veids	Paņemts	
1.		32094	1	34948	"Alfa" 16.ceha darbinieku pasākums	Fotogrāfija	Sagatave	arturs.adamovics	10.06.2021 16:16:39		Jā	<input type="button" value="v"/>
2.		32093	1	34947	"Alfa" 16.ceha gadadienas svinības	Fotogrāfija	Sagatave	arturs.adamovics	10.06.2021 16:16:38		Jā	<input type="button" value="v"/>
3.		32092	1	34946	"Alfa" 16.ceha ražošanas telpas Gulbenē, Brīvības ielā 44	Fotogrāfija	Sagatave	arturs.adamovics	10.06.2021 16:16:31		Jā	<input type="button" value="v"/>

6.11 att. Kā objekti uzrādās sistēmā

Uzdevums ir sasniegts, un kad vēlamies beigt darbu ar programmu spiežam labajā augšējā pusē sarkano krustiņu lai aizvērtu programmu.

NOBEIGUMS

Kvalifikācijas darba mērķis ir sasniegts. Kvalifikācijas darba laikā ir izveidota programma, kas atvieglo darbu Latvijas Nacionālās bibliotēkas Objektu veidotāja darbiniekiem, kā arī saīsina laiku, pievienojot visus paredzētos digitālos objektus. Tagad programma ir nodota tālākā testēšanā, lai to oficiāli varētu lietot Latvijas Nacionālās bibliotēkas darbinieki, kuriem tā ir nepieciešama.

Programmu izveidoju, lai tā būtu viegli uzturama, kā arī lai būtu viegli pievienot klāt jaunus digitālā objekta laukus un klasifikatorus, lai programma varētu tos atpazīt. Plāns uz nākotni būtu uzturēt šo programmu, kā arī uzklaut lietotāja padomus, ko varētu jaunu pievienot programmai.

Personīgais darba mērķis pašā sākuma bija izveidot programmu pēc iespējas vienkāršu, lai tās lietotāji neapjuku no izskata, tajā ievietot tikai lietotājiem vajadzīgas opcijas, tādēļ arī uztaisīju programmu tā, lai digitālo objektu veidotāji varētu visus savus objektus programma ielikt no "Excel" programmas, jo tajā programmā lietotāji jau bija pieraduši strādāt vairākus gadus.

Kvalifikācijas darba laika beigās secināju, ka esmu ne tikai programmu izveidojis, bet IT nozares kolēģus, un arī ieguvis pamata bāzes zināšanas, kā programmēt C# programmēšanas valodā.

INFORMĀCIJAS AVOTI

- Aktīvās Direktijas nozīme. – https://lv.wikipedia.org/wiki/Active_Directory
- Postman dokumentācija - <https://www.geeksforgeeks.org/introduction-postman-api-development/>
- C# dokumentācija un apraksts - <https://visualstudio.microsoft.com/vs/community/>
- Diagrammu veidošanas rīks - <https://app.diagrams.net/>
- WPF pielietošana - <https://wpf-tutorial.com/>
- Gitlab - <https://www.simplilearn.com/tutorials/git-tutorial/what-is-gitlab>
- Visual studio 2019 - https://en.wikipedia.org/wiki/Microsoft_Visual_Studio
- Postman - <https://www.blazemeter.com/blog/what-is-postman-and-how-to-use-it-for-api-testing>

Definīcijas

Metadati – informācija, kas apraksta kādu citu datu kopu.

IDE – integrēta izstrādes vide.

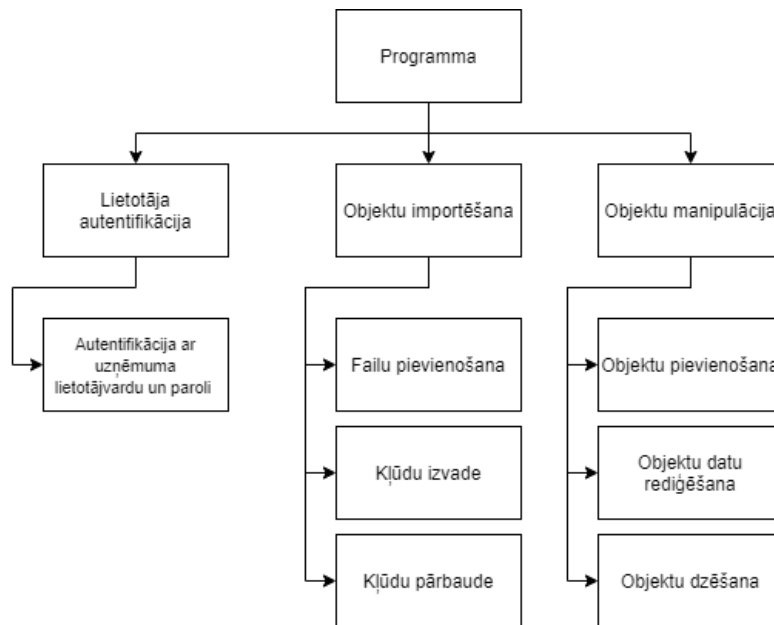
HTTP - hiperteksta transporta protokols ir lietojumslāņa protokols, kas paredzēts datu apmaiņai starp tīmekļa serveriem un pārlūkprogrammām.

API - Lietojumprogrammas saskarne.

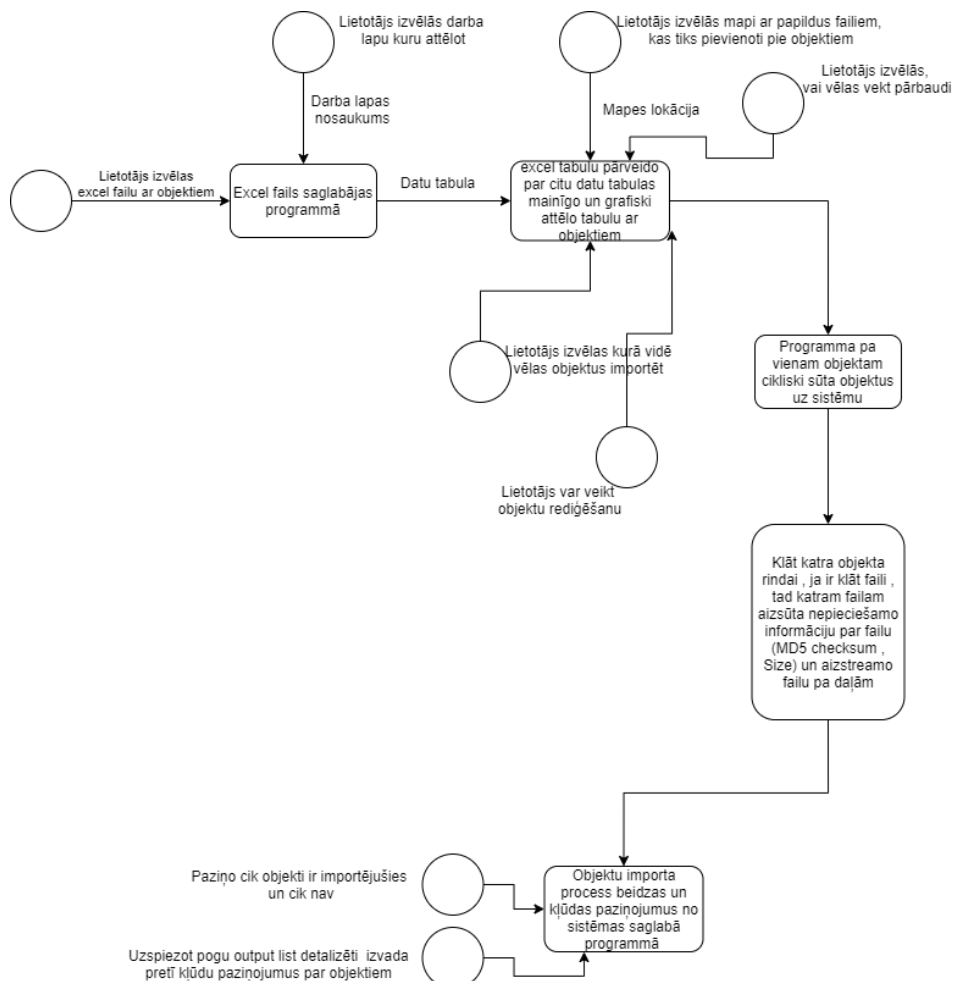
GUI - Grafiskā lietotāja saskarne.

PIELIKUMI

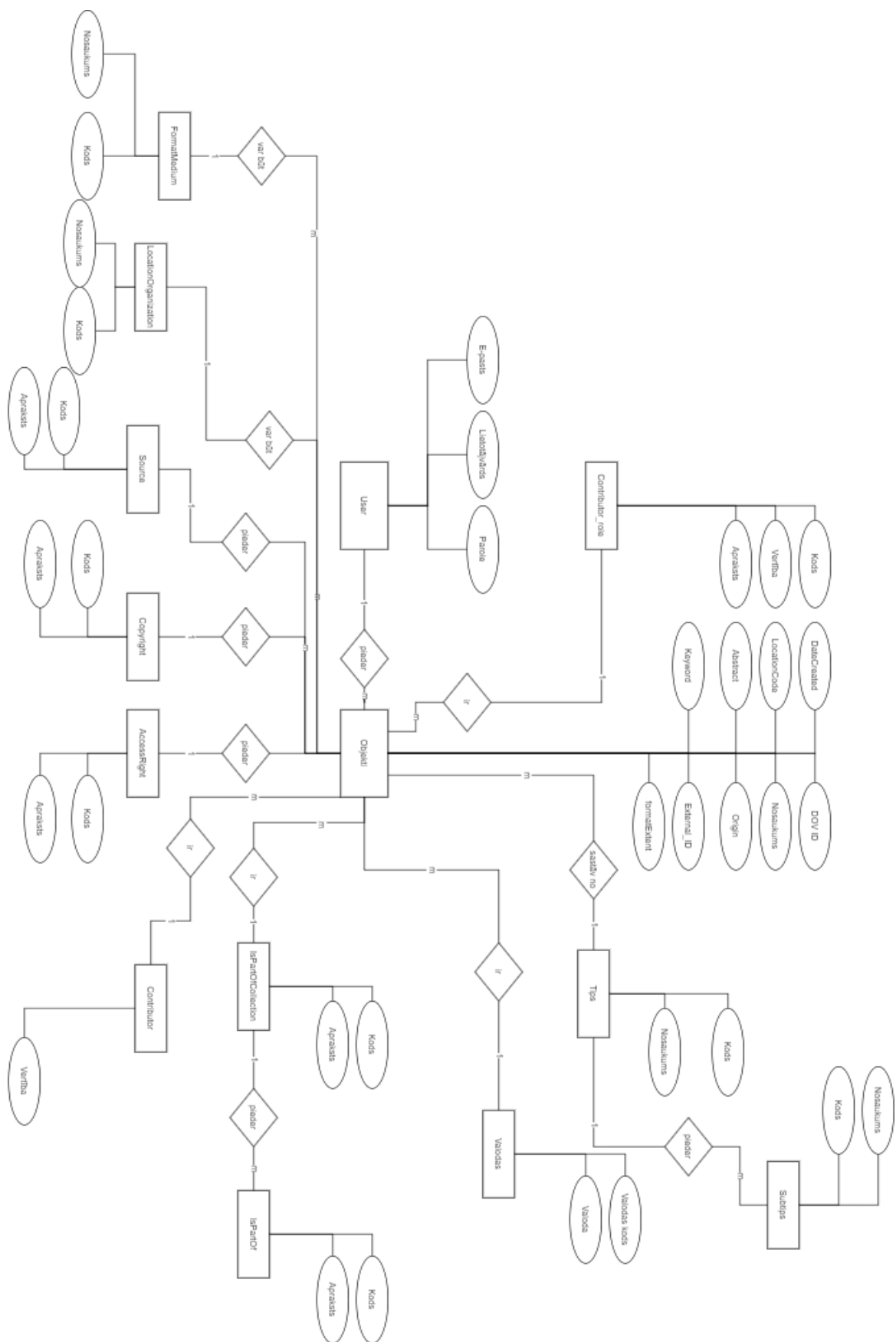
Diagrammas



1. pielikuma 1. att. Programmas arhitektūra

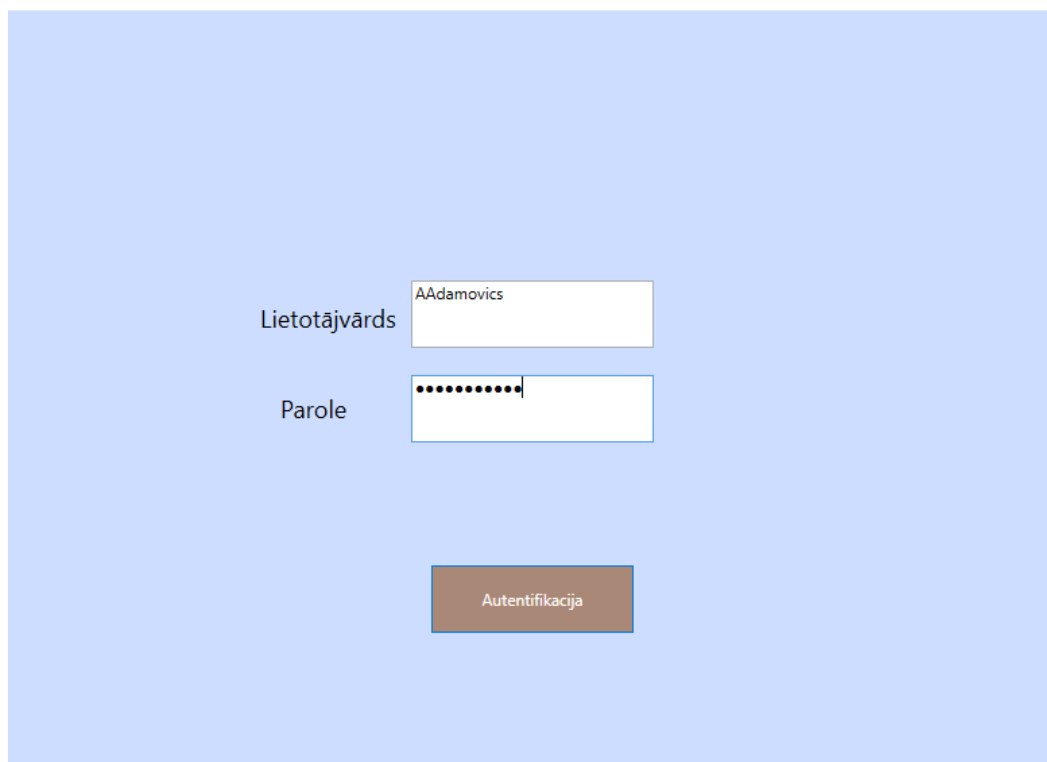


1. pielikuma 2. att. Objektu importa Datu plūsma

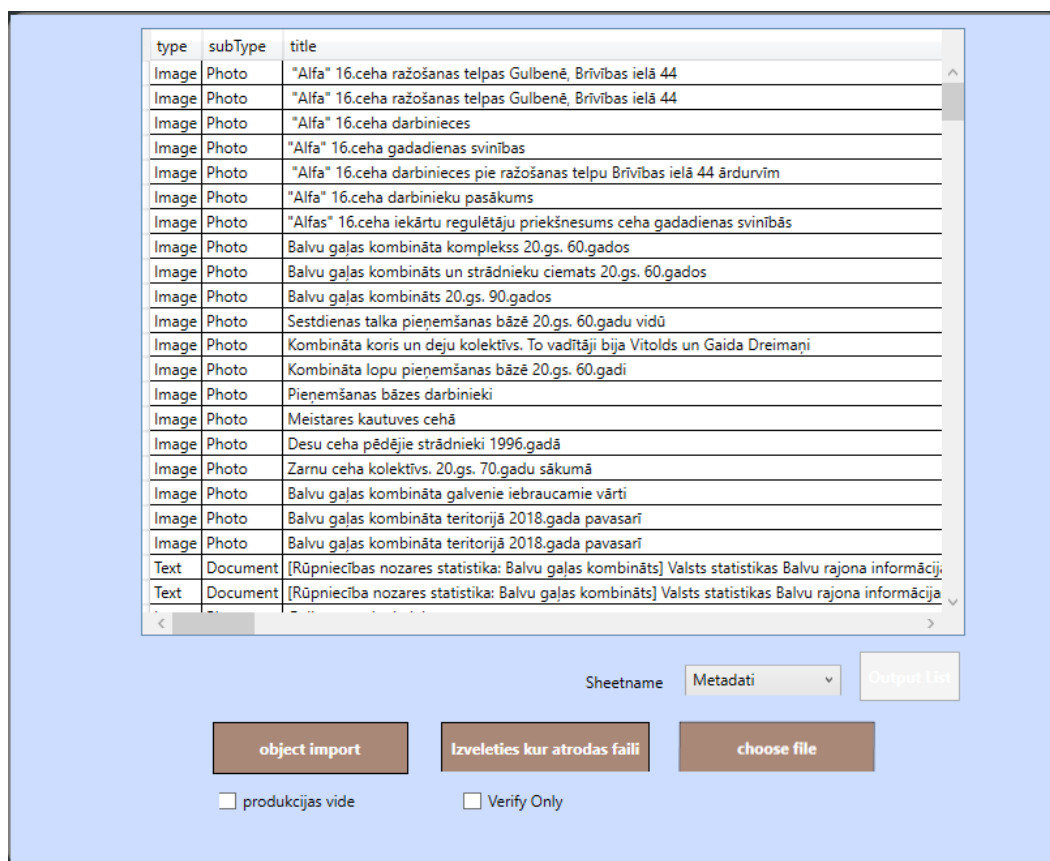


1.pielikuma 3.att. Sistēmas ER diagramma

Programmas izskats

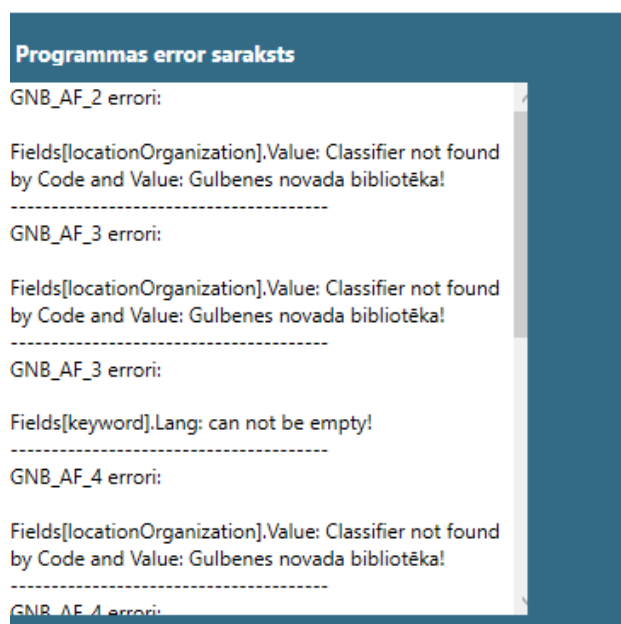


2.pielikuma 1.att. Programmas pirmais logs

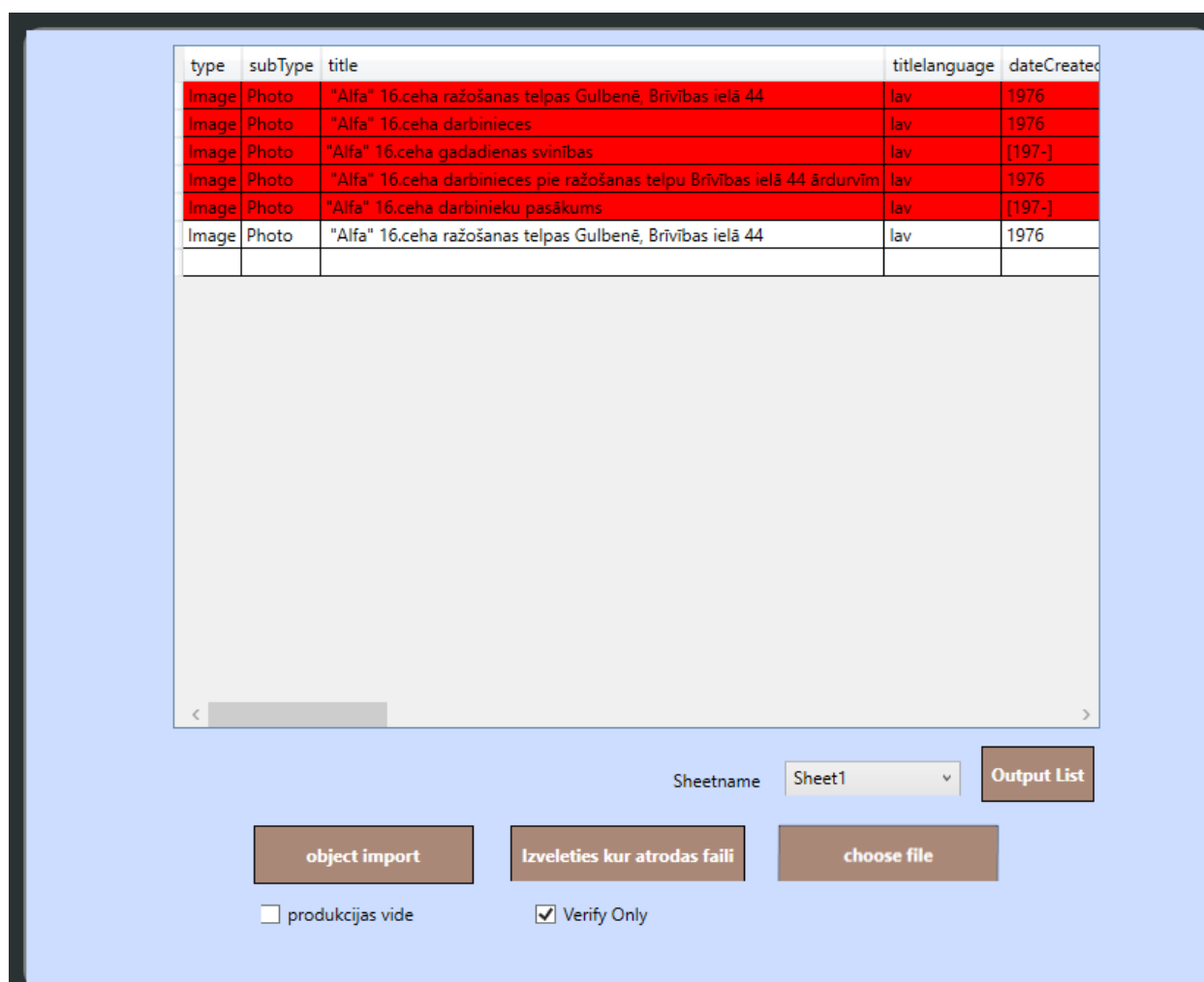


type	subType	title
Image	Photo	"Alfa" 16.ceha ražošanas telpas Gulbenē, Brīvības ielā 44
Image	Photo	"Alfa" 16.ceha ražošanas telpas Gulbenē, Brīvības ielā 44
Image	Photo	"Alfa" 16.ceha darbinieces
Image	Photo	"Alfa" 16.ceha gadadienas svinības
Image	Photo	"Alfa" 16.ceha darbinieces pie ražošanas telpu Brīvības ielā 44 ārdurvīm
Image	Photo	"Alfa" 16.ceha darbinieku pasākums
Image	Photo	"Alfas" 16.ceha iekārtu regulētāju priekšnesums ceha gadadienas svinībās
Image	Photo	Balvu gaļas kombināta komplekss 20.gs. 60.gados
Image	Photo	Balvu gaļas kombināts un strādnieku ciemats 20.gs. 60.gados
Image	Photo	Balvu gaļas kombināts 20.gs. 90.gados
Image	Photo	Sestdienas talka pieņemšanas bāzē 20.gs. 60.gadu vidū
Image	Photo	Kombināta koris un deju kolektīvs. To vadītāji bija Vitolds un Gaida Dreimaņi
Image	Photo	Kombināta lopu pieņemšanas bāzē 20.gs. 60.gadi
Image	Photo	Pieņemšanas bāzes darbinieki
Image	Photo	Meistares kautuves cehā
Image	Photo	Desu ceha pēdējie strādnieki 1996.gadā
Image	Photo	Zarnu ceha kolektīvs. 20.gs. 70.gadu sākumā
Image	Photo	Balvu gaļas kombināta galvenie iebraucamie vārti
Image	Photo	Balvu gaļas kombināta teritorijā 2018.gada pavasarī
Image	Photo	Balvu gaļas kombināta teritorijā 2018.gada pavasarī
Text	Document	[Rūpniecības nozares statistika: Balvu gaļas kombināts] Valsts statistikas Balvu rajona informācija
Text	Document	[Rūpniecība nozares statistika: Balvu gaļas kombināts] Valsts statistikas Balvu rajona informācija

2.pielikuma 2.att. Programmas galvenais logs



2.pielikuma 3.att. Objektu kļūdu saraksts



2.pielikuma 4.att. Objektu kļūdaino objektu formatēšana

Programmas izveides iemesls

Lietotāja ciešanas

Katru dienu lietotājam ir uzdevums : apkopot iedotos datus , sakārtot pēc pareizajiem mainīgajiem excel failā , un pēc tam visus datus nosūtīt uz serveri.

Problēma šajā pēdējā solī ir tāda , ka nav īsti laba starpnieka programmas , kas varētu sazināties ar serveri un nosūtīt visu informāciju kopā ar visām bildēm un datnēm , ja tādas eksistē . lietotājam ir vairākas opcijas kā šo izdarīt , bet tās visas ir laikietilpīgas un neparocīgas. Pirmā opcija būtu tiešsaistē uz servera objektu sistēmā visus šos te objektus rakstīt ar roku un katram objektam pievienot savu aprakstu , gadu , utt. Ja tie būtu pāris objekti ar vienu bildi, tad tas nav tik traki un to var dabūt gatavu 5-10 minūtēs, bet ja tie ir 100 objekti, katrs ar savām bildēm un aprakstu, tad tas ir ļoti laika ietilpīgs process un varētu ilgt vairākas stundas, vai pat dienas. Var būt arī gadījums , ka ir pāris objekti , bet katram objektam klāt var būt 100 faili kas ir jāpievieno klāt , piemēram vecas avīzes. Otrā opcija ir apkopot visu informāciju vienā excel failā un pēc tam lietot programmu , kas sadala katru objektu no excel faila SIP pakās (xml datnē visa informācija par tieši vienu objektu no 100 piem.).Pēc tam būtu jāpielieto vēl viena programma , kas apstrādā šīs te SIP pakas un izveido sistēma nepieciešamos objektus. Problēma ir tāda , ka SIP pakas programma vairs nestrādā , un programma , kas šīs SIP pakas pēc tam sūt projām uz serveri ir novecojusi , un vadība negrib lai lieto šo novecojošo programmu. Trešais variants ir lietot viena darbinieka pašu izveidotu skriptu Excel , kurš arī var šos objektus aizsūtīt uz serveri , bet to oficiāli neviens nelieto , un tam arī ir savi veikspējas traucējumi. Trešo variantu var lietot tikai ,tad kad ir nepieciešams pievienot vecus laikrakstus ar vairākām datnēm. Tā dēļ palūdž man izveidot šo te starpniek-programmu , kas var savākt visus datus un izveidot nepieciešamos objektus sistēmā , kopā ar nepieciešamajām datnēm , un izdarīt to visu vienlaicīgi.

Lietotāja loma

Sagatavot excel failu ar nepieciešamo informāciju , un salikt nepieciešamās galvenes priekš objektiem. Salikt visas datnes , kas ir saistītas ar objektiem vienā mapē kopā ar excel faila apkopoto informāciju. Nospiest import pogu , un pētīt vai ir kādas kļūdas objektu pievienošanā , analizēt tās un pēc tam atbilstoši labot , un mēģināt vēlreiz.

3.pielikuma 1.att. Pierakstītas lietotāja ciešanas pirms programmas